БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ
ИНСТИТУТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

**Светлана Тодорова Топалова**

# КЛАСИФИКАЦИЯ НА ДИЗАЙНИ И СРОДНИ НА ТЯХ КОМБИНАТОРНИ СТРУКТУРИ

## ДИСЕРТАЦИЯ

за придобиване на научната степен
"доктор на науките"
Област 4. Природни науки, математика и информатика
Професионално направление 4.6. Информатика и компютърни науки

София, 2019

# Съдържание

**Увод** **6**

i.1 **Смисъл и цели на класификационните задачи** . . . . . 6

i.2 **Методология и алгоритми** . . . . . 7

i.3 **Структура на изложението и обзор на съдържанието** . . . . 8

i.4 **Апробация на резултатите** . . . . . 15

i.5 **Авторска справка за приносите** . . . . . 17

i.6 **Съответствие на изискванията** . . . . . 19

i.7 **Благодарности** . . . . . 20

**1 Основни понятия и предварителни резултати** **21**

1.1 **Дефиниции, означения и условия за съществуване** . . . . . 21

1.1.1 Дизайни и свързани с тях матрици . . . . . 21

1.1.2 Резолюции . . . . . 24

1.1.3 Спредове и паралелизми на $PG(n, q)$ . . . . . 24

1.1.4 Оптични ортогонални кодове (OOC) . . . . . 27

1.2 **Предварителни алгоритми** . . . . . 28

1.2.1 Сложност (ефективност) на алгоритми . . . . . 29

1.2.2 Изчерпващо търсене с връщане с отхвърляне на частични решения . . . . . 31

1.2.3 Базирани на търсенето с връщане методи за класификация, при които се генерират само нееквивалентни решения . . . . 34

1.2.4 Паралелна реализация на изчерпващото търсене с връщане 35

1.2.5 Класификация на дизайни със зададени автоморфизми с локалния подход . . . . . 38

1.2.6 Класификация на оптимални оптични ортогонални кодове . 41

1.2.7 Отделяне на неизоморфните дизайни . . . . . 43

1.2.8 Намиране реда на пълната група от автоморфизми на дизайн 45

1.2.9 Намиране на неизоморфните резолюции на дизайн . . . . . 46

**2 Симетрични дизайни и Адамарови матрици със зададени автоморфизми** **48**

2.1 **Дефиниции, означения и предварителни резултати** . . . . . 48

2.2 **Алгоритъм за конструиране на симетрични дизайни със зададени автоморфизми** . . . 49
2.2.1 Промени в алгоритъма от раздел 1.2.5 . . . 49
2.2.2 Сравнение с други алгоритми . . . 50
2.3 **Симетрични 2-(69,17,4) дизайни с автоморфизми от ред** 13 50
2.3.1 Въведение . . . 50
2.3.2 Особености на орбитната матрица и класификационния алгоритъм . . . 50
2.3.3 Класификация. Остатъчни дизайни . . . 52
2.4 **Алгоритъм за установяване на Адамарова еквивалентност и за намиране на групата от автоморфизми на Адамарова** $n \times n$ **матрица** . . . 54
2.4.1 Използване на разширение ($2n \times 2n$ матрица) . . . 54
2.4.2 Инварианти на Адамарови матрици . . . 54
2.5 **Адамарови матрици от ред 44** . . . 55
2.5.1 Въведение . . . 55
2.5.2 Свойства и начин на конструиране на 2-(43,21,10) дизайните 56
2.5.3 Класификация на Адамаровите дизайни . . . 58
2.5.4 Класификация на Адамаровите матрици и дизайните, които пораждат . . . 58
2.5.5 Двоични екстремални самодуални кодове с дължини 86 и 88 59
2.6 **Адамарови матрици от ред 64** . . . 60
2.6.1 Въведение . . . 60
2.6.2 Особености на конструктивния автоморфизъм и алгоритъма 61
2.6.3 Класификация на Адамаровите дизайни - отсяване на неразширимите орбитни матрици . . . 66
2.6.4 Класификация на Адамаровите матрици - намиране на класовете на еквивалентност . . . 67
2.6.5 Съответните афинни 2-(64,16,5) дизайни и хипотезата на Хамада . . . 67
2.6.6 Алгоритъм за намиране на 2-(64,16,5) дизайни с минимален ранг от Адамарови 2-(63,31,15) дизайни . . . 69

**3 Дизайни с автоморфизми от малки редове при изброяване без конструиране** **70**
3.1 **Изброяване без конструиране** . . . 70
3.2 **Уилсонови Щайнерови системи от тройки от редове** 19 **и** 21 71
3.2.1 Въведение . . . 71
3.2.2 Уилсонови $STS(19)$ и $STS(21)$ . . . 72
3.2.3 Алгоритми и особености на програмната реализация . . . 75
3.2.4 Класификационни резултати . . . 79

3.3 **Двойни на проективната равнина от ред 4** . . . . . 81
3.3.1 Въведение . . . . . 81
3.3.2 Двойни на уникално разделим дизайн . . . . . 82
3.3.3 2-(21,5,2) дизайните са уникално-разделими . . . . . 84
3.3.4 Алгоритъм за конструиране на двойни дизайни . . . . . 87
3.3.5 Разделими 2-$(21, 5, 2)$ с автоморфизми от ред 2 . . . . . 88
3.3.6 Класификация на двойните на проективната равнина от ред 4 91

**4 Спредове на $PG(n, 2)$ 93**
4.1 **Дефиниции, означения и предварителни резултати** . . . . . 93
4.2 **Алгоритъм за конструиране на всички (с точност до проективна еквивалентност) спредове на $PG(n, 2)$** . . . . . 94
4.2.1 Множество на търсене . . . . . 95
4.2.2 Конструиране на спреда . . . . . 95
4.2.3 Ефективност . . . . . 95
4.2.4 Тест за минималност на частичните решения . . . . . 95
4.2.5 Модификации . . . . . 96
4.3 **Класификация на всички спредове на $PG(5, 2)$** . . . . . 97
4.3.1 Въведение . . . . . 97
4.3.2 Особености на групата от автоморфизми на $PG(5, 2)$ и използването им в класификационния алгоритъм . . . . . 97
4.3.3 Класификация - модели по подпространства . . . . . 99
4.4 **Класификация на всички книжни спредове на $PG(5, 2)$** . . . 102
4.4.1 Въведение . . . . . 102
4.4.2 Свойства на книжните спредове на $PG(5, 2)$ и алгоритмите за конструирането им . . . . . 102
4.4.3 Класификация - инварианти и свойства . . . . . 104
4.5 **Класификация на някои видове книжни спредове на $PG(7, 2)$** 105
4.5.1 Въведение . . . . . 105
4.5.2 Видове книжни спредове в $PG(7, 2)$ и алгоритмите, с които ги конструираме . . . . . 106
4.5.3 Класификация - книжни спредове с изоморфни страници, инварианти . . . . . 110

**5 Разрешими дизайни и $t$-паралелизми на $PG(n, q)$ 116**
5.1 **Дефиниции, означения и предварителни резултати** . . . . . 116
5.2 **Алгоритъм за класификация на резолюции със зададени автоморфизми от прост ред** . . . . . 118
5.2.1 Цели . . . . . 118
5.2.2 Множество на търсене . . . . . 118
5.2.3 Конструиране на паралелните класове . . . . . 118

5.2.4 Тест за минималност на частичните решения . . . . . 118
5.3 **Алгоритъм за намиране на пълната група от автоморфизми на резолюция** . . . . . 119
5.4 **Алгоритъм за намиране на пълната група от автоморфизми на резолюция с даден автоморфизъм** . . . . . 120
5.4.1 Спрегнатите елементи в групата от автоморфизми . . . . . 120
5.4.2 Конструиране на спрегнати пермутации . . . . . 122
5.4.3 Спрегнати пермутации, които можем да не конструираме . . 122
5.4.4 Заключителни бележки . . . . . 124
5.5 **Резолюции на циклични дизайни с малки параметри** . . . . 125
5.5.1 Въведение . . . . . 125
5.5.2 Особености, които касаят конструктивния алгоритъм . . . . 126
5.5.3 Класификация - типове според късата орбита на дизайна, точково циклични резолюции . . . . . 126
5.6 **Резолюции на $STS(21)$ с автоморфизми от ред 3 с 3 фиксирани точки и 7 фиксирани блока** . . . . . 131
5.6.1 Въведение . . . . . 131
5.6.2 Построяване на дизайните и резолюциите им . . . . . 131
5.6.3 Класификация . . . . . 133
5.7 **2-паралелизми на $PG(5,2)$ с автоморфизъм от ред 31** . . . . . 135
5.7.1 Въведение . . . . . 135
5.7.2 Конструиране на паралелизмите . . . . . 135
5.7.3 Намиране на множества от взаимно ортогонални паралелизми 136
5.7.4 Класификация - първите транзитивни $t$-паралелизми за $t > 1$. 138

**6 Циклично-пермутационни константно-тегловни кодове, циклични дизайни, антиконфликтни и силни антиконфликтни кодове 140**
6.1 **Дефиниции и означения** . . . . . 140
6.2 **Алгоритъм за класификация на кодове с циклична структура при сравнително малък брой решения** . . . . . 144
6.2.1 Допълнителен тест за разширимост: съществуване на следваща кодова дума . . . . . 145
6.2.2 Приложимост . . . . . 145
6.2.3 Други алгоритми за кодове с циклична структура . . . . . 146
6.3 **Оптимални $(v,k,1)$ двоични циклично - пермутационни константно - тегловни кодове и циклични дизайни** . . . . . 146
6.3.1 Въведение . . . . . 146
6.3.2 Особености на използваните алгоритми . . . . . 148
6.3.3 Циклични 2-$(73,4,1)$, 2-$(76,4,1)$ и 2-$(127,7,1)$ дизайни - класификация и изоморфизъм . . . . . 148

6.3.4 Класификация на оптимални $(v, k, 1)$ двоични циклично - пермутационни константно - тегловни кодове с $k = 4, 5, 6$ и 7 и малки дължини. . . . . 150
6.4 **Антиконфликтни кодове** . . . . . 154
6.4.1 Въведение . . . . . 154
6.4.2 Особености на антиконфликтните кодове, които са отразени в алгоритмите . . . . . 157
6.4.3 Класификация - обща и за плътни кодове . . . . . 158
6.5 **Силни антиконфликтни кодове** . . . . . 167
6.5.1 Въведение . . . . . 167
6.5.2 Допълнителните условия и използваните алгоритми . . . . . 170
6.5.3 Класификация - сравнение с известните конструкции и граници 171

**Възможности за бъдеща работа** **176**

**Библиография** **177**

**Публикации по дисертацията** **192**

**Цитирания на публикациите по дисертацията** **194**

**Означения и съкращения** **200**

# Увод

## i.1 Смисъл и цели на класификационните задачи

Предмет на настоящата дисертация са задачи за конструиране, изброяване и класификация (според дадени свойства) на дизайни и сродни на тях обекти, в това число - резолюции на дизайни, Адамарови матрици, спредове и паралелизми на крайни проективни пространства, самодуални кодове, циклично-пермутационни константно-тегловни кодове, антиконфликтни и силни антиконфликтни кодове. Както изброяването, така и класификацията, касаят обекти, които се разбиват на класове на еквивалентност спрямо дадена релация на еквивалентност. При изброяването определяме само броя на класовете на еквивалентност, докато при класификацията описваме и някои свойства на обектите от различните класове. Понякога изброяване или класификация са възможни и без построяване на самите обекти. Два такива примера са дадени в глава 3. Във всички останали случаи, описани в дисертацията, първо намираме по един представител от всеки клас на еквивалентност, а след това описваме характерните свойства. Такава класификация се нарича конструктивна.

Получените класификационни резултати са за зададени сравнително малки параметри и представляват интерес, както с оглед на възможни директни приложения (в шумозащитното кодиране, в криптографията, за статистически експерименти, в комуникационни системи с множествен достъп и др.), така и като необходима информация за решаването на по-общи задачи или като евентуална съставна част в конструкции за по-големи параметри. За тези цели може да се наложи подбор на структури с конкретни свойства и това е особено лесно, ако всички възможни решения са класифицирани и достъпни.

Ако броят на решенията е твърде голям, то дори да е възможно получаването им за разумно време, съхраняването е проблем, а и търсенето сред тях също. В такива случаи обикновено се прибягва до класификацията само на тези обекти, които притежават някакви допълнителни свойства, които считаме за интересни и полезни. Такива са, например, инвариантността спрямо дадени групи от автоморфизми и геометрични свойства като регулярност и книжност (на спредове и паралелизми).

## i.2 Методология и алгоритми

В дисертацията са разгледани примери за решаването на интересни отворени задачи за конструктивна класификация с помощта на компютър. Решенията включват конструиране на всички или част от структурите с дадени параметри, разбиването им на класове на еквивалентност и намиране на някои характеристики на представители от всеки от тези класове (например, реда на пълната група от автоморфизми). Резултатите са получени чрез прилагане, както на подходящи компютърни алгоритми, така и на редица теоретични съображения.

Ако се предполага, че съществува комбинаторна структура с определени параметри и, евентуално, със зададени допълнителни свойства, може да се напише програма, която да анализира пространството от възможни решения докато се намери такова, което удовлетворява съответните критерии. Ако съществуват много решения, то програмата трябва да отдели еквивалентните и да класифицира получените структури според свойствата им. В голямата част от случаите не е тривиално да се намерят ефективни алгоритми, както за конструиране, така и за анализ на получените решения. Затова компютърните методи, използвани в конструктивната комбинаторика, представляват също толкова голям интерес, колкото и получените от прилагането им резултати.

Алгоритмите за конструиране на разглежданите в дисертацията обекти се базират на разновидност на метода на Faradžev [86] и Read [206], известен под името ***подредено генериране (orderly generation)***. Те се основават на изчерпващо търсене с връщане и са с експоненциална сложност. Използването им за параметри, които представляват реален интерес, става възможно с помощта на теоретични съображения, които позволяват намаляване на пространството на търсене и/или бързо отхвърляне на значителен брой еквивалентни или непригодни частични решения. Основно значение при тези алгоритми имат представянето и подредбата в множеството на търсене, бързодействието на тестовете за минималност и разширимост и подборът на частичните решения, върху които те се прилагат.

Всички компютърно-зависими класификационни резултати, представени в дисертацията, са получени със софтуер на автора, който реализира описаните алгоритми и е написан на C++. В случаите когато е употребяван и друг софтуер, това е обяснено на съответните места. Използвани са персонални компютри със сравнително добри за времето си параметри (съвсем различни в началото и в края на 18-годишния период, за който става въпрос). На тях най-тежките задачи са отнемали 20 - 30 дни. В голямата част от случаите всички или част от пресмятанията са правени и от съавторите с помощта на друг софтуер, а понякога и на други алгоритми. Това е един от начините да сме по-сигурни в достоверността на резултатите. Винаги е разглеждана и съвместимостта на полученото с предварително известните теоретични изследвания и граници, както и с класификации на сродни обекти.

## i.3 Структура на изложението и обзор на съдържанието

Дисертацията съдържа увод и 6 глави. Всяка глава се състои от няколко части, а всяка част от няколко раздела.

В първата глава са представени предварителни резултати във вид, който е удобен за по-нататъшното им използване в дисертацията.

Част 1.1 включва дефиниции, означения и резултати, които се използват в повече от една от следващите пет глави. Те касаят преди всичко комбинаторните дизайни и/или техните резолюции, защото дизайните са свързани по един или друг начин с всички обекти, разглеждани в дисертацията. Понятия и резултати, които се използват само в една от следващите глави, са дефинирани в нейната първа част, а такива, които се срещат само в една от частите на дадена глава, са дефинирани в първия раздел на тази част.

В част 1.2 са описани предварително известните алгоритми, които са използвани, модифицирани или надграждани в дисертацията - в раздел 1.2.2 - изчерпващо търсене с връщане с отхвърляне на частични решения [101], [151], в 1.2.5 - класификация на дизайни със зададени автоморфизми с локалния подход [14], в 1.2.6 - класификация на оптимални OOC [1], в 1.2.7 - тест за изоморфизъм на дизайни, в 1.2.8 - определяне реда на пълната група от автоморфизми на дизайн и в 1.2.9 - намиране на неизоморфните резолюции [14].

Глави 2 до 6 съдържат получените нови резултати и подробно описание на разработените за нуждите на всяка задача методи и алгоритми.

Глава 2 включва класификационни резултати за симетрични дизайни и за Адамарови матрици (тях получаваме от Адамарови симетрични дизайни). Равният брой на точките и блоковете на симетричните дизайни води до редица допълнителни свойства и приложения (част 2.1). Алгоритмите за конструирането им също използват техните специфични свойства.

В част 2.2 е описан разработеният алгоритъм за конструиране на симетрични дизайни със зададени автоморфизми от прост ред, който е модификация на алгоритъма от раздел 1.2.5.

Тема на част 2.3 са 2-(69,17,4) дизайните. Преди настоящото изследване бяха известни два симетрични 2-(69,17,4) дизайна [182] - единият е от безкрайния клас, конструиран от Shrikhande и Singhi [212], а другият е намерен от Božikov [40] като инвариантен относно група на Frobenius от ред 39. В този раздел конструираме всички неизоморфни 2-(69,17,4) дизайни с автоморфизми от ред 13 и техните остатъчни 2-(52,13,4) дизайни.

Адамаровите матрици са тема на останалите части от тази глава. Те имат разнообразни приложения [73], [74], [210] и са напълно класифицирани за редове до 32 [145], [119], [153], [154], [220], докато за по-големите редове има само частични класификации (например [34], [75], [98], [99], [221], [229]).

Част 2.4 представя разработените алгоритми за установяване на Адамарова

еквивалентност и намиране на групата от автоморфизми на Адамарова $n \times n$ матрица, които свеждат задачите до съответните задачи за бинарни матрици с размер $2n \times 2n$ .

В част 2.5 са разгледани Адамарови матрици от ред 44. Преди резултатите, представени в него, бяха известни два Адамарови 2-(43,21,10) дизайна [182] с автоморфизми от ред 43 [105], [198]. В дисертацията е представена класификацията на всички нееквивалентни Адамарови матрици от ред 44, които имат автоморфизми от ред 7, и съответните им Адамарови 3-(44,22,10) и Адамарови 2-(43,21,10) дизайни. От 2-(43,21,10) дизайните с методи, описани в [29] и [233] намираме двоични двойно-четни самодуални кодове с дължина 88. Два от тях са екстремални, единият от които беше нов към момента на публикуване на резултата. Конструираме и много самодуални кодове с дължина 86, сред които няма екстремални.

Част 2.6 е посветена на Адамарови матрици от ред 64. Hamada формулира хипотеза [107], според която дизайните, получени от точките и подпространствата с дадена размерност в $AG(n, p^m)$ ($p$ е просто) имат минимален $p$-rank, а всички други дизайни със същите параметри имат по-голям $p$-rank. Сред няколкото намерени контрапримера на тази хипотеза са три 2-(64,16,5) дизайна [110], [185], [235], които посредством конструкцията на Rahilly [205] съответстват на три Адамарови 2-(63,31,15) дизайна. И трите са инвариантни относно диедралната група от ред 10. Това мотивира представената тук класификация на всички Адамарови 2-(63,31,15) дизайни, инвариантни относно диедралната група от ред 10 и на съответните им нееквивалентни Адамарови матрици от ред 64.

От 2-(63,31,15) дизайните с 2-ранг не по-голям от 16 чрез конструкцията на Rahilly [205] построяваме 2-(64,16,5) дизайни. Така получаваме точно предварително известните (от [110] и [185]) три 2-(64,16,5) дизайна с 2-ранг 16. Описан е разработеният алгоритъм за намиране на 2-(64,16,5) дизайни с минимален ранг от Адамарови 2-(63,31,15) дизайни, свеждащ задачата до построяването на спредове от прави на Адамаровите дизайни, като ни интересуват само спредовете с минимален ранг.

Изследванията от глава 2 са публикувани в [T1], [T2] и [T6].

Глава 3 представя решението на поредица от задачи за конструиране на дизайни, инвариантни относно групи от автоморфизми от ред 2 или 3, които са използвани след това за изброяване на дизайните с дадените параметри и с тривиална група от автоморфизми.

Част 3.1 разглежда възможността за определяне броя на всички неизоморфни дизайни с дадени параметри и евентуални допълнителни свойства, ако са конструирани дизайните с нетривиални автоморфизми и е известен броят на всички дизайни (заедно с изоморфните).

Щайнеровите системи от тройки от ред $n$ ($STS(n)$) са сред най-изучаваните и най-широко използвани дизайни. До момента са класифицирани всички $STS(n)$

за $n \leq 19$ [150] и следователно $STS(21)$ са системите от най-малък ред, от които са изследвани само някои класове с допълнителни свойства [62], [143], [179], [180], [231]. Те представляват особен интерес и поради това, че още не е известно дали сред тях има двойно-разрешими.

Тема на част 3.2 е класификацията на Уилсоновите Щайнерови системи от тройки от редове 19 и 21. Те съдържат три Щайнерови подсистеми от тройки от ред 7 и са разглеждани за пръв път от Wilson [249], който с помощта на известни резултати за латински квадрати определя общия им брой и извежда долни граници за броя на техните класове на изоморфизъм. В настоящата част 3.2 е определен точният брой на тези класове на изоморфизъм, като за целта е използвана конструктивната класификация на всички дизайни с нетривиални автоморфизми.

В раздел 3.2.3 са разгледани начините за построяване на дизайните при различни случаи на предварително зададени автоморфизми от ред 2 или 3, при каквито локалният подход, описан в раздел 1.2.5, не винаги е достатъчно ефективен. Затова най-често използваме особеностите на конкретната задача за разработване на алгоритъм за изчерпващо търсене за конкретния случай.

В раздел 3.2.4 са събрани класификационните резултати за всички STS(21) от Уилсонов тип и за Уилсоновите STS(19). Получените STS(21) са изследвани за двойна разрешимост.

Част 3.3 съдържа класификация на двойните на единствения с точност до изоморфизъм 2-(21,5,1) дизайн. Той може да бъде получен от инцидентността на точките и правите на проективната равнина от ред 4. Както и в предишната част, класификацията е конструктивна само за дизайните с нетривиални автоморфизми. Нашите разглеждания от раздел 3.3.2 показват, че това е възможно, ако двойните дизайни са уникално разделими, а в раздел 3.3.3 доказваме, че това е в сила за двойните на 2-(21,5,1) дизайна и извеждаме долна граница за техния брой.

В раздел 3.3.4 е разработен алгоритъм за конструиране на двойни дизайни, при който тестът за минималност съществено използва групата от автоморфизми на дизайна, чиито двойни търсим. Ако конструираните двойни дизайни са уникално-разделими, с прилагането на този тест построяваме само неизоморфни дизайни.

В раздел 3.3.5 са разгледани модификациите на алгоритъма от раздел 3.3.4, с които са конструирани двойни дизайни с различни видове автоморфизми от ред 2, а в раздел 3.3.6 са представени и обсъдени получените класификационни резултати за двойните на 2-(21,5,1) дизайна. Техният брой е доста близък до долната граница, получена в раздел 3.3.3.

Изследванията от глава 3 са публикувани в [T4] и [T5].

Глава 4 е посветена на конструктивната класификация на всички или избрана част от спредовете на $PG(n,2)$. Освен като геометрични обекти, спредовете

в проективни пространства представляват интерес и заради приложенията си в теорията на кодирането [83], [84], [173], [185] и криптографията [202]. Те са обект, както на много теоретични разглеждания [17], [123], [124], [130], [131], така и на компютърни класификации [33], [185], [217].

В част 4.2 е описан разработеният алгоритъм за конструиране на всички (с точност до проективна еквивалентност) спредове на $PG(n,2)$, който съществено използва свойствата на групата от автоморфизми на проективното пространство, както при теста за минималност, така и за предварително подреждане по удобен начин на елементите от множеството на търсене и фиксиране на част от тях без загуба на общност.

Преди настоящото изследване $PG(5,2)$ е най-малкият отворен случай за пълна класификация на спредове в $PG(n,2)$, като частична класификация на такива спредове е направена в [185]. В част 4.3 на дисертацията е представена конструктивната класификация на всички (131044) спредове на $PG(5,2)$. За построяването им е използвана модификация на алгоритъма от част 4.2, при която фиксираме 3 прави и предварително намираме и запазваме конкретни автоморфизми на $PG(5,2)$. Частта завършва с обобщение в раздел 4.3.3 на получените резултати за групите от автоморфизми, за броя на правите от спреда, съдържащи се в различните тримерни подпространства и за ранга на 2-(64,16,5) дизайна, получен с конструкцията на Rahilly [205].

Книжните спредове, дефинирани от Shaw, представляват самостоятелен интерес заради допълнителните си геометрични свойства, които позволяват по-обстоятелствено теоретично разглеждане. Част 4.4 разглежда конструирането на книжните спредове на $PG(5,2)$. В раздел 4.4.2 е описана използваната модификация на алгоритъма от част 4.2 за дадения случай, при който са фиксирани шест конкретно подбрани прави от спреда. Получените резултати са обобщени в раздел 4.4.3. Те са в съответствие с теоретичните разглеждания, направени от Shaw.

Част 4.5 е посветена на книжните спредове на $PG(7,2)$. Въпреки, че съставляват малка част от всички спредове в това проективно пространство, тяхната пълна конструктивна класификация засега е невъзможна. Затова в тази част от дисертацията са построени само книжните спредове, отговарящи на някои допълнителни условия. Тези условия, както и съобразената с тях модификация на алгоритъма от раздел 4.2 са описани в раздел 4.5.2, а в раздел 4.5.3 са обобщени получените класификационни резултати.

Изследванията от глава 4 са публикувани в [T7], [T10] и [T11].

Предмет на глава 5 са задачи, които включват класификация на всички резолюции на даден дизайн или само на тези от тях, които са инвариантни относно избрана група от автоморфизми от прост ред. Част от резултатите касаят $t$-паралелизми на $PG(n,q)$, защото те могат да бъдат разглеждани като резолюции на дизайна от точките и $t$-мерните подпространства на $PG(n,q)$.

Алгоритъмът за класификация на резолюции със зададени автоморфизми е описан в част 5.2. Ако броят на автоморфизмите на дизайна не е прекалено голям, в процеса на конструиране намираме и автоморфизмите на резолюцията. При много голяма група от автоморфизми на дизайна, обаче, за да ускорим конструирането използваме друг тест за минималност и в такъв случай се налага допълнително определяне на пълната група от автоморфизми на резолюцията. В глава 5 са представени и двата разработени за целта алгоритъма - единият е приложим за всякакви резолюции и е описан в раздел 5.3, а другият е предназначен само за резолюции, които притежават даден автоморфизъм. Той се основава на предварителното генериране на някои спрегнати на зададения автоморфизъм и е предмет на част 5.4.

Резолюции с богати групи от автоморфизми представляват особен интерес с оглед на някои приложения [104], [135]. Затова резолюциите на циклични дизайни са обект на множество статии [68], [97], [139], [159], [160], [179], [190], [191], [237].

В част 5.5 са разгледани резолюции на циклични Щайнерови системи от тройки ($STS(v)$) с малки параметри. Целта е от една страна конструирането на нови резолюции ($KTS(v)$), а от друга изследване на свойствата на всички известни резолюции на циклични $STS(v)$ с $v \leq 39$, както и събирането им и предоставянето на web-достъп до тях с оглед на възможни бъдещи приложения и теоретични изследвания.

Специфичните особености на задачата са разгледани в раздел 5.5.2, а в раздел 5.5.3 е представена класификацията и някои инварианти на резолюциите на цикличните $STS(15)$, $STS(21)$ и $STS(27)$, на резолюциите с нетривиални автоморфизми на цикличните $STS(33)$ и на резолюциите с автоморфизми от ред 13 на цикличните $STS(39)$. Сред тях особен интерес представляват *точково циклични* (point-cyclic) резолюции. Те притежават автоморфизъм, който пермутира точките в един цикъл.

Резолюциите на $STS(21)$ представляват особен интерес с оглед на нерешения въпрос за съществуване на двойно разрешим дизайн с тези параметри и са обект на много изследвания [62], [143], [179], [180], [231]. В част 5.6 са конструирани всички $STS(21)$ с автоморфизми от ред 3 с 3 фиксирани точки и 7 фиксирани блока и са класифицирани техните резолюции.

Изследването на $t$-паралелизми в $PG(n, q)$ представлява интерес понеже те имат много връзки с други математически обекти и разнообразни приложения [42], [83], [129], [216], [222]. Затова са обект, както на множество теоретични изследвания [24], [28], [77], [78], [126], [127], [134], [201], [253], така и на класификации с помощта на компютър [27], [203], [204], [208] [209], [223], [238], [240], [241], [259].

В част 5.7 са класифицирани всичките 12312 неизоморфни 2-паралелизми на $PG(5, 2)$ с автоморфизъм от ред 31. Задачата беше от особен интерес, защото преди настоящото изследване въпросът за съществуване на транзитивни $t$-паралелизми в

$PG(n,q)$ за $t>1$ беше отворен. С решаването на тази задача са намерени първите и единствени до момента 92 примера на такива паралелизми. Това са 2-паралелизми в $PG(5,2)$ с пълна група от автоморфизми от ред 155, която е транзитивна върху спредовете. От тях са построени и множества от по 10 взаимно-ортогонални паралелизми.

Изследванията от глава 5 са публикувани в [T3], [T8], [T12] и [T13].

Глава 6 разглежда двоични циклично-пермутационни константно-тегловни кодове (CPCW), циклични дизайни, антиконфликтни кодове (CAC) и силни антиконфликтни кодове (SCAC). Тук са обединени класификационни резултати за обекти, които са различни, но имат сходна циклична структура и множество връзки помежду си, което определя и сходния подход към класификационната задача в тези случаи. В главата са разработени модификации на алгоритъма от раздел 1.2.6. Едната от тях касае проверката за несъществуване на код с дадени параметри и размер, а останалите отчитат специфичните особености на разглежданите обекти. От особена важност са начините за минимизиране и подреждане на множеството на търсене (възможните кодови думи) и някои допълнителни тестове за разширимост на частичните решения.

Предмет на част 6.3 са оптималните $(v,k,1)$ двоични циклично - пермутационни константно - тегловни кодове (CPCW) с малки дължини и свързани с тях циклични дизайни. CPCW кодовете намират приложение в комуникационни канали без обратна връзка, използвани едновременно от много потребители [31], [66]. CPCW кодове с параметри $(v,k,1)$ са еквивалентни на $(v,k,1)$ оптични ортогонални кодове и са обект на редица статии - [15], [32], [41], [46], [48], [53], [57], [175], [194], [196], [244]. Преди нашите изследвания не бяха известни класификационни резултати за оптималните $(v,k,1)$ CPCW кодове и за цикличните 2-$(73,4,1)$ и 2-$(76,4,1)$ дизайни. Имаше само класификации на циклични 2-$(v,k,1)$ дизайни с някои по-малки параметри, а $q=127$ беше първият неизяснен случай за съществуване на $(q,7,1)$ циклична разностна фамилия (цикличен 2-$(q,7,1)$ дизайн) за $q$, което е степен на просто число и $q\equiv 1 \pmod{42}$ [55].

В част 6.3 са класифицирани с точност до мултипликативна еквивалентност оптималните $(v,k,1)$ CPCW кодове с $k=4,5,6$ и 7 и с точност до изоморфизъм цикличните 2- $(73,4,1)$, 2-$(76,4,1)$ дизайни и е установено, че не съществува цикличен 2-$(127,7,1)$ дизайн. Особеностите на използваните алгоритми са описани в раздел 6.3.2, а резултатите са обобщени в раздел 6.3.4.

Част 6.4 е посветена на $(n,k)$ антиконфликтните кодове (CAC). Това са кодове, които гарантират, че в комуникационните системи с множествен достъп без обратна връзка всеки измежду $s$ (колкото е броят на кодовите думи) потенциални участници в комуникацията може да изпрати успешно поне един пакет с данни в рамките на определен времеви период от $n$ еднакви интервала, ако е активен през този период и ако най-много $k$ участници са активни едновременно.

Съществуването на оптимални CAC е обект на интензивни изследвания. Случаят $k = 3$ е напълно решен [92], [93], [125], [166], [167], [189], [192], [250], а конструкции на оптимални CAC с тегла 4 и 5 са дадени в [193]. Известни са и някои горни граници за размера на CAC [214], [215]. Примери за CAC с тегла 3, 4 и 5 и малки дължини могат да бъдат намерени в [234], но на автора не са известни класификационни резултати преди настоящото изследване.

В част 6.4 са определени множество неизвестни преди стойности на максималния брой кодови думи $M(n, k)$ на $(n, k)$ CAC с малки дължини и $k \leq 7$. За повечето разглеждани дължини е представена и класификация. Използваният класификационен алгоритъм и спецификата на задачата са описани в раздел 6.4.2. Получените резултати са обобщени в раздел 6.4.3, където е представена и отделна класификация на плътните CAC с $k \leq 7$ и малки дължини.

Докато CAC се използват в комуникационните системи с множествен достъп без обратна връзка, но с налична синхронизация по време (слот), то в случаите, когато такава синхронизация няма, се използват кодове, наречени *силни антиконфликтни кодове (SCAC)* [254]. Те са предмет на част 6.5. Някои горни граници за размера на такива кодове са изведени в [252] и [255].

В част 6.5 са класифицирани оптимални SCAC с малки параметри. Използваните алгоритми са описани в раздел 6.5.2 и представляват модификации на алгоритъма от част 6.2 и на този от раздел 6.4.2, като промените касаят основно *възможните кодови думи* (които генерираме и подреждаме преди да започне търсенето), допълнителни проверки при добавянето на кодова дума и теста за минималност. За разлика от CAC и CPCW кодовете, класът на еквивалентност на един SCAC не е затворен относно автоморфизмите на $Z_n$. По тази причина тест за мимималност не може да бъде прилаган на частичните, а само на пълните решения. Класификационните резултати за оптимални SCAC с $k \leq 5$ и малки дължини са представени в раздел 6.5.3.

Изследванията са публикувани в [T9], [T14], [T15], [T16], [T17] и [T18].

## i.4 Апробация на резултатите

Включените в дисертацията резултати са получени от автора в периода от 2000 до 2017 година. Част от тях са самостоятелни изследвания – [T1], [T2], [T3], [T12], [T13]. Останалите статии са в съавторство с:

- Z. Mateva – [T6] и [T7];
- S. Zhelezova – [T8];
- T. Baicheva – [T9], [T14], [T15], [T16], [T17], [T18];
- R. Shaw – [T11];
- T. P. McDonough , R. Shaw – [T10];
- V. Fack, J. Winne, R. Zlatarski – [T4];
- P. Kaski, P. Östergård, R. Zlatarski – [T5].

В дисертацията са описани само резултатите, които са получени от автора. В много от случаите за проверката на пресмятания с помощта на компютър, голяма част от тях са направени едновременно от още един или повече съавтори с различни алгоритми и/или програмни реализации, а в [T10] и с теоретични разглеждания. В статиите [T5] и [T10] е описан приносът на отделните автори.

Публикациите са в следните научни списания:

- Journal of Statistical Planning and Inference (ISI IF) – [T1];
- Discrete Mathematics (ISI IF) – [T2], [T4], [T5], [T6], [T11];
- Journal of Combinatorial Designs (ISI IF) – [T7];
- Graphs and Combinatorics (ISI IF) – [T8];
- Problems of Information Transmission (ISI IF) – [T9], [T14];
- IEEE Communications Letters (ISI IF) – [T17];
- Designs Codes and Cryptography (ISI IF) – [T18];
- Note di Matematica (SJR) – [T10];
- Electronic Notes in Discrete Mathematics (SJR) – [T15];
- Cybernetics and Information Technologies (SJR) – [T16];
- Journal of Algebra Combinatorics Discrete Structures and Applications –[T13];

- Mathematica Balkanica – [T3];
- Serdica Journal of Computing – [T12];

Изследванията по дисертацията са представяни на различни научни конференции:

- Международните конференции *Algebraic and Combinatorial Coding Theory*, Русия (2002, 2010), България (2000, 2004, 2012, 2016);
- Международните конференции *Optimal codes and Related Topics*, България (2001, 2005, 2017);
- Международна EWM конференция *Groups and Graphs*, Варна, България (2002);
- Международна конференция *Congress MASSEE*, Боровец, България (2003);
- Международен *Workshop on Combinatorial Algorithms and Algorithmic Graph Theory*, Ghent University, Гент, Белгия ( 2003);
- Международна конференция *European Conference on Combinatorics, Graph Theory and Applications (EuroComb '05)* , Берлин, Германия (2005);
- Международна конференция *Pioneers of Bulgarian Mathematics*, София (2006);

Доклади по отделни задачи, които са решени в дисертацията, са изнасяни на:

– Семинара по Кодиране, Институт проблем передачи информации на Руската академия на науките, Москва (2001);

– Националния семинар по теория на кодирането (2002, 2010, 2016);

– Семинара по комбинаторика към Института по математика на Унгарската академия на науките – Будапеща (2002);

– Семинара на Department of Applied Mathematics and Computer Science, Ghent University, Гент, Белгия (2004);

– Семинар Математически основи на информатиката - съвместен регулярен семинар на Секция "Математически основи на информатиката" към ИМИ – БАН и Факултет "Математика и информатика" на ВТУ Св. Св. Кирил и Методий (2009).

## i.5 Авторска справка за приносите

По мнение на автора основните приноси на дисертационния труд са:

Разработени са поредица от алгоритми за конструктивна класификация, както и алгоритми за допълнително изследване на построените обекти, в това число:

– алгоритъм за конструиране на симетрични дизайни със зададени автоморфизми от прост ред [T1];

– алгоритми за установяване на Адамарова еквивалентност и за намиране на групата от автоморфизми на Адамарова $n \times n$ матрица [T2], [T6];

– алгоритъм за конструиране на двойни дизайни [T4];

– алгоритъм за конструиране на всички (с точност до проективна еквивалентност) спредове на $PG(n, 2)$ [T7];

– алгоритъм за класификация на резолюции със зададени автоморфизми от прост ред [T8], [T13];

– алгоритъм за намиране на пълната група от автоморфизми на резолюция с даден автоморфизъм [T12];

– алгоритми за класификация или установяване на несъществуване на CAC [T14], [T15], SCAC [T17] и CPCW кодове [T9], [T18];

– алгоритми, които са приложими само в контекста на конкретна задача и параметри.

Тези алгоритми се основават на специфичните свойства на разглежданите структури. С програмните им имплементации са направени приноси в изучаването на дизайни и техните резолюции, Адамарови матрици, спредове и паралелизми на проективни пространства, самодуални кодове, двоични циклично-пермутационни константно-тегловни кодове, антиконфликтни и силни антиконфликтни кодове. По-точно, в дисертацията е направена конструктивна класификация на:

- Четирите неизоморфни 2-(69,17,4) дизайна с автоморфизми от ред 13 и техните 16 остатъчни 2-(52,13,4) дизайни [T1];

- Всичките 384 нееквивалентни Адамарови матрици от ред 44, които имат автоморфизми от ред 7 и съответните им 1683 Адамарови 3-(44,22,10) и 57932 Адамарови 2-(43,21,10) дизайна. От 2-(43,21,10) дизайните са построени двоични самодуални кодове с дължини 86 и 88, сред които има екстремални и нови [T2];

- Всичките 8330 Адамарови 2-(63,31,15) дизайни, инвариантни относно диедралната група от ред 10 и съответните им 1691 нееквивалентни Адамарови матрици от ред 64. От 2-(63,31,15) дизайните с 2-ранг не по-голям от 16 чрез конструкцията на Rahilly са намерени всички съответстващи им 2-(64,16,5) дизайни с 2-ранг 16, които представляват особен интерес във връзка с хипотезата на Hamada за ранга на геометричните дизайни [T6];

- Всички Уилсонови Щайнерови системи от тройки от ред 19 и 21, които притежават нетривиални автоморфизми (1192 STS(19) и 10165 STS(21)). От резултатите следва, че броят на неизоморфните Уилсонови STS(21) е 2166351. Конструираните STS(21) са изследвани за двойна разрешимост, защото това са Щайнеровите системи от най-малък ред, за които не се знае съществуват ли двойно-разрешими [T5];

- Всичките 1028899 двойни на 2-(21,5,1) дизайна, които притежават нетривиални автоморфизми. С тяхна помощ и след доказване на уникалната разделимост на двойните на 2-(21,5,1), е установено, че броят на всички двойни е 1746461307 [T4];

- Всичките 131044 спреда на $PG(5,2)$. За тях са пресметнати инварианти, включващи групата от автоморфизми, броя на правите от спреда, съдържащи се в различните тримерни подпространства и ранга на 2-(64,16,5) дизайна, получен с конструкцията на Rahilly [T7];

- Деветте книжни спреда на $PG(5,2)$. Получените резултати са в съответствие с теоретичната класификация, направена от Shaw [T10];

- Книжни спредове на $PG(7,2)$ (над 2 милиона), отговарящи на някои допълнителни условия. За тях са пресметнати инварианти, включващи групата от автоморфизми и броя на тримерните и петмерните подпространства, чиито точки са изцяло покрити с прави от спреда [T11];

- Всички 2963 неизоморфни $STS(21)$ с автоморфизъм от ред 3 с 3 фиксирани точки и 7 фиксирани блока и техните резолюции [T3];

- Резолюции на циклични Щайнерови системи от тройки ($STS(v)$) с малки параметри, сред които особен интерес представляват *точково цикличните* (point-cyclic) резолюции. Представени са инварианти на 19364 резолюции на цикличните $STS(v)$ с $v \leq 27$, на 844346 резолюции с нетривиални автоморфизми на цикличните $STS(33)$ и на 2827 резолюции с автоморфизми от ред 13 на цикличните $STS(39)$ [T13];

- Всичките 12312 неизоморфни 2-паралелизми на $PG(5,2)$ с автоморфизъм от ред 31. С решаването на тази задача са намерени първите и единствени до момента 92 примера на транзитивни $t$-паралелизми в $PG(n,q)$ за $t > 1$. Това са 2-паралелизми в $PG(5,2)$ с пълна група от автоморфизми от ред 155, която е транзитивна върху спредовете. От тях са построени и множества от по 10 взаимно-ортогонални паралелизми [T8];

- Всички с точност до изоморфизъм оптимални $(v,k,1)$ двоични цикличнопермутационни константно-тегловни кодове (CPCW) с $k = 4,5,6$ и 7 и малки

дължини (за някои дължини има милиони кодове); От класификационните резултати следва несъществуването на $(127, 7, 1)$ циклична разностна фамилия и, респективно, цикличен 2-$(127, 7, 1)$ дизайн [T9], [T18];

- Всички неизоморфни циклични 2-$(73, 4, 1)$ и 2-$(76, 4, 1)$ дизайни, чиито брой е, съответно, 1426986 и 1113024 [T9];

- Мултипликативно нееквивалентните оптимални $(n, k)$ антиконфликтни кодове (CAC) с $k = 3, 4, 5, 6$ и 7 и малки дължини (милиони кодове за някои дължини). За други дължини е определен само максималният брой кодови думи $M(n, k)$ [T14], [T15];

- Мултипликативно нееквивалентните плътни оптимални $(n, k)$ антиконфликтни кодове (CAC) с $k = 3, 4, 5, 6$ и 7 и малки дължини [T16];

- Мултипликативно нееквивалентните оптимални $(n, k)$ силни антиконфликтни кодове (SCAC) с $k = 3, 4$ и 5 и малки дължини [T17].

## i.6 Съответствие на изискванията

**Точки по националните изисквания**:

- от публикациите по дисертацията: 373 точки (при изискване за минимум 100)

  - 15 от представените публикации са в списания, индексирани в Web of Science или Scopus (12 с импакт фактор и 3 с SJR) и носят общо 313 точки (по $40/n$ точки всяка, където $n$ е броят на съавторите)
  - 3 публикации са в рецензирани списания и носят 60 точки (по $20/n$ точки)

- от цитирания на публикациите по дисертацията: 192 точки (при изискване за минимум 100)

  - 150 точки са от 30 цитирания в публикации в списания, индексирани в Web of Science или Scopus
  - 24 точки са от 8 цитирания в монографии или колективни томове с научно рецензиране
  - 18 точки са от 9 цитирания в публикации в рецензирани списания (7 от тях реферирани в ZbMath)
  - цитирания, които не носят точки: 4 в сборници от международни конференции, 10 в дисертации в чужбина и 10 в дисертации в България.

Нито една от представените тук 18 публикации не е използвана в дисертацията на автора за придобиване на образователната и научна степен *доктор*. Съвместните публикации с Цонка Байчева не са използвани в нейната дисертация за придобиване на научната степен *доктор на науките*.

Въпреки, че няма изисквания за припокриване с публикациите, представени на конкурса за заемане на академичната длъжност професор, този въпрос често се задава и беше зададен и на мене. Осемте публикации от 2014 година насам не са използвани в конкурсите за академичните длъжности доцент (2001) и професор (2013). От тях две са самостоятелни (останалите са с по един съавтор), 4 са с импакт фактор и 2 – с SJR.

## i.7 Благодарности

Със задачи за конструктивна класификация, сродни на разгледаните в настоящата дисертация, са се занимавали на даден етап не малко от българските специалисти, работещи в областта на комбинаториката и алгебричната теория на кодирането – Тончев [11], [12], [231], [120], [142], [163], [227], [228], [229], [230], [232], [233], [237] , Капралов [4], [5], [6], [7], [140], [143], [144], [142], [163], Ланджев [162], [142], [163], [164], Буюклиев [2], [34], [35], [36], [37], [38], [39], Буюклиева [37], Байчева [1], [197], Матева [9], [176], [177], Железова [3], [258], [259], [260], [261], [262], Монев [39], Джумалиева [38], [39] и др. Работила съм съвместно с някои от българите със сходни научни интереси (Стоян Капралов, Николай Манев, Иван Ланджев, Росен Златарски, Златка Матева, Стела Железова, Цонка Байчева), други (например Стефан Додунеков, Владимир Тончев, Стоян Капралов, Иван Ланджев, Стефка Буюклиева, Илия Буюклиев, Евгения Великова, Ася Русева) са откликвали и отговаряли (наживо или по е-мейл) на различни мои запитвания и са ме насочвали към интересни задачи, а с трети просто сме дискутирали по отворени въпроси, най-често повдигнати при срещите ни на Националния семинар по теория на кодирането или на международните конференции Algebraic and combinatorial coding theory (ACCT) и Optimal codes and related topics (OC). В годините, когато все още беше много трудно да разбереш какво вече е направено по даден конкретен проблем, при визитите си в чужбина колегите (най-често Додунеков и Тончев, но и много други) са ми изпращали (носили) статии и справочници, които ме интересуват. Изключително приятно е да се работи в такава силна и добронамерена група. Благодаря на всичките си съавтори за плодотворната и приятна съвместна работа. Благодаря на цялата колегия за помощта, която са ми оказвали, и за разбирането, което винаги са проявявали. А на всички колеги от Велико Търново благодаря и затова, че с удоволствие съм ходила на работа през тези години.

За съжаление, Стефан Додунеков - човекът, на когото най-много бих искала да благодаря, вече не е между живите. Основаването и оцеляването на Велико Търновския филиал на БАН стана възможно само благодарение на неговите нестандартни идеи и на размаха, оптимизма, смелостта и всеотдайността, с които ги осъществяваше, на дарбата му да ръководи и насърчава останалите, на способността му да открие и насочи в правилна посока заложбите на всеки един от колектива, на готовността му да рискува, за да помогне с всичко, с каквото може, и на всестранната му подкрепа. Благодаря на Николай Манев за неоценимата помощ, която му оказваше при прохождането на Велико Търновското звено, а на ръководствата на БАН, на ИМИ и на община Велико Търново – за усилията да съхранят и доразвиват създаденото от професор Додунеков.

# Глава 1

# Основни понятия и предварителни резултати

Комбинаторните дизайни са свързани по един или друг начин с всички обекти, разглеждани в дисертацията. Затова в тази глава са събрани понятия, означения и известни алгоритми и резултати, касаещи дизайните и резолюциите им, като е посочена и връзката им с изследванията в следващите глави, а за някои от тях и конкретния начин, по който са използвани по-нататък.

Понятия и предварителни резултати, които се използват само в една от следващите глави, са дефинирани в началото на съответната глава.

## 1.1 Дефиниции, означения и условия за съществуване

### 1.1.1 Дизайни и свързани с тях матрици

**Дефиниция 1.1.1.** *Нека* $V = \{P_i\}_{i=1}^{v}$ *е крайно множество от* ***точки****, а* $\mathcal{B} = \{B_j\}_{j=1}^{b}$ *– крайна фамилия от* $k$*-елементни подмножества на* $V$*, наречени* ***блокове****. Казваме, че* $D = (V, \mathcal{B})$ *е* ***дизайн*** *с параметри* $t$*-(*$v$*,*$k$*,*$\lambda$*), ако всяко* $t$*-подмножество на* $V$ *се съдържа точно в* $\lambda$ *блока на* $\mathcal{B}$*.*

Традиционно с буквите $v, k, \lambda, b$ и $r$ означаваме параметрите на разглеждан дизайн, т.е. $v$ е броят на точките на дизайна, $k$ – броят на точките, съдържащи се в произволен блок, $\lambda$ – броят на блоковете, съдържащи дадена двойка точки, $b$ – броят на блоковете и $r$ – броят на блоковете, съдържащи дадена точка. Самия 2-($v$,$k$,$\lambda$) дизайн обикновено означаваме с $D$.

**Дефиниция 1.1.2.** ***Матрица на инцидентност*** *на дизайна наричаме (0,1) матрица с* $v$ *реда и* $b$ *стълба, в която елементът от* $i$*-ия ред и* $j$*-ия стълб е равен на* 1*, ако* $P_i \in B_j$ $(i = 1, 2, ..., v,\ j = 1, 2, ..., b)$ *и на* 0 *в противен случай.*

Дизайнът се определя напълно от своята матрица на инцидентност. На множеството от точките на дизайна съпоставяме множеството от естествените числа от 1 до $v$, а на множеството от блоковете – това на естествените числа от 1 до

$b$, т.е. говорим за $i$-тата точка и $j$-я блок или за точка с номер $i$ и блок с номер $j$, $i = 1, 2, ..., v,\ \ j = 1, 2, ..., b$. Ясно е, че $i$-та точка съответствува на $i$-я ред от матрицата на инцидентност на дизайна, а $j$-ят блок – на $j$-я ѝ стълб. На фигура 1.1.а е дадена матрицата на инцидентност на един $2-(9, 3, 3)$ дизайн.

**Дефиниция 1.1.3.** *Един дизайн се нарича* ***симетричен****, ако броят на точките е равен на броя на блоковете му ($v = b$).*

При симетричен дизайн $k = r$ и всеки два блока имат точно $\lambda$ общи точки. Транспонираната на неговата матрица на инцидентност е матрица на инцидентност на дизайн със същите параметри, наречен ***дуален*** на дадения. Ако един симетричен дизайн е изоморфен на дуалния си, той се нарича ***самодуален***.

**Дефиниция 1.1.4.** *Ако от симетричен 2-($v$,$k$,$\lambda$) дизайн премахнем един блок и точките, които той съдържа, се получава 2-($v-k$,$k-\lambda$,$\lambda$) дизайн, наречен* ***остатъчен***.

**Дефиниция 1.1.5.** *Под* ***Щайнерова система*** *$S(t,k,v)$ разбираме $t$-($v$,$k$,1) дизайн, а под* ***Щайнерова система от тройки от ред $v$*** *– 2-($v$,3,1) дизайн. Ако последният е разрешим, резолюциите се наричат* ***Киркманови системи от тройки от ред $v$***.

Щайнеровите системи от тройки от ред $v$ се означават $STS(v)$, а Киркмановите – $KTS(v)$.

**Дефиниция 1.1.6.** *Два дизайна са* ***изоморфни****, ако съществува взаимно-еднозначно съответствие между точките и блоковете на единия дизайн и, съответно, точките и блоковете на другия, запазващо инцидентността.*

От тази дефиниция следва, че два дизайна са ***изоморфни***, тогава и само тогава, когато матрицата на инцидентност на единия се получава от матрицата на инцидентност на другия чрез пермутация на редове и стълбове.

**Дефиниция 1.1.7.** ***Автоморфизъм*** *наричаме всеки изоморфизъм на дизайна със себе си.*

Автоморфизмите представяме като пермутация на точките, при която блоковете се изобразяват в блокове. Множеството от всички автоморфизми на дизайна образува група по отношение на операцията композиция на изображения. Наричаме я ***пълна група от автоморфизми***. Всяка нейна подгрупа наричаме група от автоморфизми на дизайна.

**Дефиниция 1.1.8.** ***Циркулантна матрица*** *от ред $n$ е квадратна матрица $M = (m_{i,j})_{n\times n}$ с $n$ реда и стълба, такива че $m_{i+1,j+1} = m_{i,j}$, където $i, j = 0, 1, \ldots, n-1$ и събирането на индексите е по модул $n$.*

Всеки дизайн, който притежава автоморфизъм от прост ред $p$, е изоморфен на дизайн с матрица на инцидентност с циркулантна структура (фигура 2.1).

**Дефиниция 1.1.9.** *Насочен граф с $n$ върха, които съответстват на елементите на $Z_n$, се нарича* ***циркулантен граф****, ако пермутацията $(0, 1, 2, \ldots, n-1)$ е автоморфизъм на графа.*

Матрицата на съседство на циркулантен граф е циркулантна (0,1) матрица. Ето защо терминът ***циркуланта*** се използва както за циркулантен граф, така и за циркулантна матрица.

**Дефиниция 1.1.10.** *Нека $D$ е дизайн с група от автоморфизми $G$ от прост ред $p$. Да означим с $O_1$, $O_2$, ..., $O_m$ точковите и с $O'_1$, $O'_2$, ..., $O'_n$ блоковите орбити спрямо $G$. Нека $m_{ij}$ е броя на точките от $O_i$, съдържащи се в кой да е блок от $O'_j$ $(i = 1, 2, ..., m,\ j = 1, 2, ..., n)$.* ***Орбитна матрица*** *на дизайна спрямо тази група от автоморфизми наричаме матрицата $\mathcal{M} = (m_{ij})_{m \times n}$.*

**Дефиниция 1.1.11.** *$t$-$(v,k,\lambda)$ дизайн е* ***цикличен****, ако притежава автоморфизъм $\alpha$, който пермутира точките му в един цикъл.*

**Дефиниция 1.1.12.** ***Квазикратен*** *на 2-$(v,k,\lambda)$ дизайн наричаме дизайн с параметри 2-$(v,k,m\lambda)$, където $m$ е цяло число, по-голямо от единица.*

**Дефиниция 1.1.13.** *Един квазикратен дизайн е* ***разделим*** *до $m$ дизайна с параметри 2-$(v,k,\lambda)$, ако съществува разбиване на блоковете му на $m$ подмножества, всяко от които е 2-$(v,k,\lambda)$ дизайн.*

Следващите три теореми и техните доказателства могат да бъдат намерени във всяка монография, посветена на комбинаторните дизайни, например [11], [12], [26], [106], [115].

**Теорема 1.1.1.** *Параметрите на 2-$(v,k,\lambda)$ дизайн са свързани с равенствата:*

$$vr = bk,$$
$$bk(k-1) = v(v-1)\lambda.$$

**Теорема 1.1.2.** *[89] (Неравенство на* ***Fisher****)*

*За всеки 2-$(v,k,\lambda)$ дизайн с $k < v$ е в сила:*

$$v \leq b.$$

**Теорема 1.1.3.** *[174] (Неравенство на* ***Mann****)*

*Ако съществува 2-$(v,k,\lambda)$ дизайн с $k < v$, съдържащ $s$ еднакви блока, то:*

$$s.v \leq b.$$

**Теорема 1.1.4.** *[43], [58] (Теорема на* ***Bruck-Ryser-Chowla****) За симетричен 2-$(v,k,\lambda)$ дизайн е в сила:*

∘ *Ако $v$ е четно, то $k - \lambda$ е точен квадрат.*

∘ *Ако $v$ е нечетно, то съществуват цели числа $x, y, z$, не всичките равни на 0, които са решение на уравнението $x^2 = (k-\lambda)y^2 + (-1)^{\frac{v-1}{2}}\lambda z^2$.*

### 1.1.2 Резолюции

**Дефиниция 1.1.14.** ***Паралелен клас*** *се нарича множество от блокове на дизайна, такова, че всяка точка се съдържа точно в един блок.*

**Дефиниция 1.1.15.** ***Резолюция*** *наричаме всяко разбиване на блоковете на дизайна на паралелни класове. Един дизайн е* ***разрешим****, ако има поне една резолюция.*

Пример за разрешим дизайн е даден на фигура 1.1.

**Дефиниция 1.1.16.** *Две резолюции на дизайна $D$ са* ***изоморфни****, ако съществува автоморфизъм на $D$, който изобразява всеки паралелен клас на едната резолюция в паралелен клас на другата.*

**Дефиниция 1.1.17.** *Две резолюции на един и същ дизайн са* ***ортогонални****, ако всяка двойка паралелни класове, единият от едната, а другият от другата резолюция, имат най-много един общ блок.*

**Дефиниция 1.1.18.** *Един 2-$(v,k,\lambda)$ дизайн е* ***двойно разрешим****, когато притежава поне две ортогонални резолюции.*

На фигура 1.1 е даден пример за две ортогонални резолюции.

### 1.1.3 Спредове и паралелизми на $PG(n,q)$

**Дефиниция 1.1.19.** *Нека $\mathcal{P}$ и $\mathcal{L}$ са две множества, чиито елементи наричаме съответно* ***точки*** *и* ***прави****. Структура на инцидентност $\mathcal{G} = (\mathcal{P}, \mathcal{L}, \mathrm{I})$, $\mathrm{I} \subset \mathcal{P} \times \mathcal{L}$ наричаме* ***проективно пространство****, ако тя удовлетворява следните три аксиоми:*

- *С всеки две различни точки е инцидентна точно една права.*
- *С всяка права са инцидентни поне три различни точки.*
- *(Аксиома на Veblen – Young). Четири различни прави не могат да се пресичат в пет различни точки.*

**Дефиниция 1.1.20.** *Проективно* ***подпространство*** *на проективно пространство $\mathcal{G}$ се нарича множество от точки $\mathcal{W} \subset \mathcal{P}$ такива, че за всеки две точки $K, H \in \mathcal{W}$, $\mathcal{W}$ съдържа и всяка точка $X$, инцидентна с правата, определена от $K$ и $H$.*

На векторното пространство $V_{n+1}(q)$ с размерност $n+1$ над крайно поле с $q$ елемента съответства крайно проективно пространство с размерност $n$ и ред $q$, чиито точки съответстват на нормализираните вектори от $V_{n+1}(q)$. Означаваме го с $\boldsymbol{PG(n,q)}$. Всяко крайно проективно пространство с $n \geq 3$ е изоморфно на някое $PG(n,q)$, получено по описаното по-горе съответствие [225, теорема 2.22].

Фигура 1.1: Две ортогонални резолюции на 2-$(9,3,3)$ дизайн

а) матрица на инцидентност на дизайна

$v = 9$, $b = 36$ – матрицата има 9 реда и 36 стълба,

$k = 3$, $r = 12$ – има 3 единици във всеки стълб, 12 единици във всеки ред,

$\lambda = 3$ – скаларното произведение на всеки два реда е 3.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 2 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 3 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 5 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 6 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 7 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 8 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 9 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |

б) разбиване на блоковете на дизайна от резолюцията $\mathcal{R}$

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 2 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 3 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 5 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 6 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 7 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 8 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 9 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |

в) разбиване на блоковете на дизайна от резолюцията $\mathcal{T}$

| | 1 | 15 | 9 | 4 | 27 | 3 | 7 | 33 | 6 | 10 | 35 | 18 | 13 | 32 | 2 | 16 | 29 | 24 | 19 | 11 | 30 | 22 | 20 | 12 | 25 | 14 | 5 | 28 | 23 | 36 | 31 | 26 | 8 | 34 | 17 | 21 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 2 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 3 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 4 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 5 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 6 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 7 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 8 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 9 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |

г)множество от две взаимно-ортогонални резолюции:

$$\mathcal{R} = \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \\ 13 & 14 & 15 \\ 16 & 17 & 18 \\ 19 & 20 & 21 \\ 22 & 23 & 24 \\ 25 & 26 & 27 \\ 28 & 29 & 30 \\ 31 & 32 & 33 \\ 34 & 35 & 36 \end{matrix} \qquad \mathcal{T} = \begin{matrix} 1 & 15 & 9 \\ 4 & 27 & 3 \\ 7 & 33 & 6 \\ 10 & 35 & 18 \\ 13 & 32 & 2 \\ 16 & 29 & 24 \\ 19 & 11 & 30 \\ 22 & 20 & 12 \\ 25 & 14 & 5 \\ 28 & 23 & 36 \\ 31 & 26 & 8 \\ 34 & 17 & 21 \end{matrix}$$

**Дефиниция 1.1.21.** ***Автоморфизъм*** *на $PG(n,q)$ е взаимно еднозначно съответствие, което изобразява точките в точки, правите в прави и запазва инцидентността.*

**Дефиниция 1.1.22.** *Множество от $t$-мерни подпространства на $PG(n,q)$, което задава разбиване на множеството от точките ѝ, се нарича* ***$t$-спред****.*

**Дефиниция 1.1.23.** ***Частичен*** *$t$-спред е множество от непресичащи се $t$-мерни подпространства на $PG(n,q)$. Частичен $t$-спред в $PG(n,q)$ е* ***максимален****, ако не се съдържа в друг частичен $t$-спред на $PG(n,q)$ с повече елементи.*

**Дефиниция 1.1.24.** *Два (частични) $t$-спреда са* ***изоморфни****, ако съществува автоморфизъм на проективното пространство, изобразяващ единия в другия.*

Ако $t = 1$, говорим за 1-спред и го наричаме просто *спред*.

**Дефиниция 1.1.25.** *разбиване на фамилията от всички $t$-мерни подпространства на $PG(n,q)$ на $t$-спредове се нарича* ***$t$-паралелизъм****.*

Необходимо условие за съществуването на $t$-спредове и $t$-паралелизми е $(t+1)|(n+1)$. Обикновено 1-паралелизмите се наричат само *паралелизми*.

**Теорема 1.1.3.1.** *( [11], [225]) Инцидентността на точките и $t$-мерните подпространства на $PG(n,q)$ дефинира 2-дизайн с параметри:*

$$v = (q^{n+1}-1)/(q-1),\ k = (q^{t+1}-1)/(q-1),$$

$$\lambda = \begin{cases} \frac{(q^{n+1}-q^2)(q^{n+1}-q^3)\ldots(q^{n+1}-q^t)}{(q^{t+1}-q^2)(q^{t+1}-q^3)\ldots(q^{t+1}-q^t)} & t > 1, \\ 1 & t = 1. \end{cases}$$

Следователно съществува взаимно еднозначно съответствие между паралелен клас от резолюция на съответния дизайн и $t$-спред от $t$-мерни подпространства на проективното пространство, както и между резолюция на дизайна от $t$-мерните подпространства и $t$-паралелизъм. Изоморфизъм и ортогоналност при $t$-паралелизмите се дефинира както за резолюциите.

Автоморфизмите на $PG(n,q)$ изобразяват $t$-мерните подпространства в $t$-мерни подпространства. Поради това, всички, свързани със съответното проективното пространство дизайни, притежават неговата пълна група от автоморфизми.

**Дефиниция 1.1.26.** ***Автоморфизъм*** *на $t$-паралелизма $P$ е автоморфизъм на $PG(n,q)$, който изобразява всеки $t$-спред на $P$ в $t$-спред на $P$.*

**Дефиниция 1.1.27.** ***Транзитивен*** *се нарича $t$-паралелизъм, притежаващ група от автоморфизми, която действа транзитивно върху $t$-спредовете му.*

**Дефиниция 1.1.28.** *Един $t$-паралелизъм е* ***цикличен****, ако притежава автоморфизъм, който пермутира $t$-спредовете му в един цикъл.*

Цикличните $t$-паралелизми са транзитивни, но обратното не винаги е вярно.

### 1.1.4 Оптични ортогонални кодове (OOC)

За основните дефиниции и означения следваме [47], [60], [65] и [194].

Със $Z_v$ означаваме цикличната адитивна група от целите числа по модул $v$. Груповата операция означаваме с $\oplus$. Ако целите числа $a$ и $v$ са взаимно прости ($gcd(a, v) = 1$) и $a \in Z_v$, елементът $a$ генерира цялата група, а умножението на всички елементи на групата с него задава автоморфизъм на $Z_v$. Всеки автоморфизъм на $Z_v$ се получава с умножение на елементите на групата с елемент, който я генерира. Автоморфизмите на $Z_v$ означаваме с $\varphi_0, \varphi_1, \ldots, \varphi_{m-1}$, където $m$ е броят на всички генератори, а $\varphi_0$ е идентитета.

**Дефиниция 1.1.29.** *Един $(v, k, \lambda_a, \lambda_c)$* ***оптичен ортогонален код*** *$\mathcal{C}$ е колекция от $\{0, 1\}$ последователности с дължина $v$ и тегло по Хеминг $k$, за която е в сила:*

$$\sum_{i=0}^{v-1} x(i)x(i \oplus j) \leq \lambda_a, \ 1 \leq j \leq v-1 \tag{1.1}$$

$$\sum_{i=0}^{v-1} x(i)y(i \oplus j) \leq \lambda_c, \ 0 \leq j \leq v-1 \tag{1.2}$$

*за всяка двойка различни последователности $x, y \in \mathcal{C}$.*

Всеки $(v, k, \lambda_a, \lambda_c)$ OOC може да бъде дефиниран и по следния начин:

**Дефиниция 1.1.30.** *Оптичният $(v, k, \lambda_a, \lambda_c)$ ортогонален код е колекция $\mathcal{C} = \{C_1, \ldots, C_s\}$ от $k$-елементни подмножества (кодови думи) на $Z_v$, такива че всеки две различни транслации на кодова дума имат най-много $\lambda_a$ общи елемента, а всеки две транслации на две различни кодови думи имат най-много $\lambda_c$ общи елемента:*

$$|C_i \cap (C_i + t)| \leq \lambda_a, \quad 1 \leq i \leq s, \quad 1 \leq t \leq v-1 \tag{1.3}$$

$$|C_i \cap (C_j + t)| \leq \lambda_c, \quad 1 \leq i < j \leq s, \quad 0 \leq t \leq v-1. \tag{1.4}$$

Условието (1.3) се нарича ***автокорелационно свойство***, а (1.4) - ***кроскорелационно свойство***. *Размер* на кода $\mathcal{C}$ наричаме броя $s$ на кодовите му думи. Когато $\lambda_a = \lambda_c$, кодът се означава с $(v, k, \lambda)$.

За всяка кодова дума $C = \{c_1, c_2, \ldots, c_k\}$ на оптичен ортогонален код означаваме с $\triangle' C$ мултимножеството от стойности на разликите $c_i - c_j$, $i \neq j$, $i, j = 1, 2, \ldots, k$. Автокорелационното свойство означава, че най-много $\lambda_a$ разлики са еднакви. Нека да означим с $\triangle C$ съответното на мултимножеството $\triangle' C$ множество.

***Тип*** на $C$ е броят на елементите в $\triangle C$, т.е. броят различни стойности на разликите. Върху кодовите думи на ООС дефинираме ***лексикографска наредба*** по следния начин: Приемаме, че $c_1 < c_2 < \ldots < c_k$ за всяка кодова дума $C = \{c_1, c_2, \ldots, c_k\}$. Тогава $C' = \{c'_1, c'_2, \ldots, c'_k\}$ е лексикографски по-малка от $C'' = \{c''_1, c''_2, \ldots, c''_k\}$, ако типа на $C'$ (т.е. броят на разликите в $\Delta(C')$) е по-малък от типа на $C''$, или ако типовете на двете кодови думи са равни, но $c'_i = c''_i$ за $i < a$, а $c'_a < c''_a$. Без загуба на общност, приемаме, че всяка кодова дума е лексикографски по-малка от всички свои транслации.

Ако $\lambda_c = 1$, кроскорелационното свойство означава, че $\Delta C_1 \bigcap \Delta C_2 = \emptyset$ за две кодови думи $C_1$ и $C_2$ от $(v, k, \lambda_a, 1)$ оптичен ортогонален код.

**Дефиниция 1.1.31.** *Оптичният* $(v, k, \lambda_a, 1)$ *ортогонален код е* ***съвършен****, ако*

$$|\bigcup_{i=1}^{s} \Delta C_i| = v - 1,$$

*което означава, че всички възможни ненулеви разлики са покрити.*

**Пример 1.1.1.** *Съвършен* $(17, 4, 3, 1)$ *оптичен ортогонален код* $\mathcal{C}$

| $\mathcal{C}$ | $\triangle C$ | *Тип* |
|---|---|---|
| *{1100110000000000} или {0, 1, 4, 5}* | *{ 1,3,4,5,12,13,14,16}* | *8* |
| *{1010000010100000} или {0, 2, 8, 10}* | *{ 2,6,7,8,9,10,11,15}* | *8* |

**Дефиниция 1.1.32.** *Един оптичен ортогонален код е максимален, ако има максималния възможен (за дадените параметри) брой кодови думи.*

**Дефиниция 1.1.33.** *Два* $(v, k, \lambda_a, \lambda_c)$ *оптични ортогонални кода* $C$ *и* $C'$ *са* ***мултипликативно еквивалентни****, ако могат да бъдат получени един от друг чрез прилагане на автоморфизъм на* $Z_v$ *и замяна на кодова дума с някоя от нейните транслации.*

## 1.2 Предварителни алгоритми

В тази част са описани предварително известни алгоритми, които са използвани или модифицирани в дисертацията. Комбинаторните структури, които са обект на настоящата дисертация, са конструирани с използване на изчерпващо търсене с връщане (backtrack search) с ранно отхвърляне (early pruning) на еквивалентни и неразширими частични решения. Генерирането на лексикографски минимални представители на класовете на еквивалентност (което постигаме с тест за минималност на частичното решение) е добре известна конструктивна техника, описана и използвана за класификация на различни комбинаторни обекти, като дизайни, кодове, графи, резолюции на дизайни и др. [101], [151].

Раздел 1.2.1 представя основни понятия и факти, касаещи сложността на използваните алгоритми. В раздел 1.2.2 ще се спрем на най-общите неща при

търсенето с връщане, а в раздел 1.2.4 представяме една негова паралелна реализация. В 1.2.3 са разгледани особеностите на известните методи за класификация, при които се генерират само нееквивалентни решения, и е описан най-често използвания в дисертацията подход. Всъщност, най-голямо значение за успешните решения на описаните задачи има специфичното: начинът на представяне на обектите, изборът на подходяща лексикографска наредба, бързодействието и местата на прилагане на конкретните тестове за еквивалентност, както и математическите съображения, водещи до ефективни тестове за разширимост на частичните решения. Подробности за всичко това при различните задачи са дадени в съответните глави.

В разделите 1.2.5 и 1.2.6 са разгледани два предварително известни алгоритъма за класификация – единият за конструиране на дизайни с локалния подход, а другият – за класификация на оптични ортогонални кодове.

Алгоритмите от раздели 1.2.7, 1.2.8 и 1.2.9 са за допълнително изследване на конструираните комбинаторните структури, за да бъдат определени някои интересни и полезни техни свойства. Те не са предмет на настоящата работа. Известни са много други алгоритми и софтуер, с които могат да бъдат заменени и някои от тях са използвани от съавторите. Тези алгоритми са описани за пълнота поради това, че са в основата на софтуера на автора, с който са определяни свойствата на конструираните в дисертацията обекти и поради това, че в дисертацията са разработени и модификации на някои от тях за различни сродни обекти.

Всички останали методи и алгоритми, които са използвани или разработени в дисертацията, както и някои особености на програмната им реализация, са описани в съответните глави.

### 1.2.1 Сложност (ефективност) на алгоритми

Основните понятия в този раздел представяме според [82] и [151]. Сложността е мярка за изменението на необходимото време (или памет) за изпълнение на алгоритъма с изменяне на размера на входните данни. Най-често ни интересува сложността по време, но понякога и тази по памет може да се окаже критична.

За сравняване на функции е възприета следната терминология. Казваме, че:

- $f(x)$ е $O(g_1(x))$, ако съществува константа $M_1$, такава, че $|f(x)| \leq M_1 |g_1(x)|$ за достатъчно големи стойности на $x$.
- $f(x)$ е $\Omega(g_2(x))$, ако съществува константа $M_2$, такава, че $|f(x)| \geq M_2 |g_2(x)|$ за достатъчно големи стойности на $x$.
- $f(x)$ е $\Theta(g(x))$, ако съществуват константи $M$ и $M'$, такива, че$|f(x)| \leq M |g(x)|$ и $|f(x)| \geq M' |g(x)|$ за достатъчно големи стойности на $x$.

Времето за изпълнение на алгоритъма е пропорционално на броя $N$ на елементарните операции, които трябва да бъдат извършени, а те нарастват с нарас-

тването на размера $n$ на входните данни. Ако съществува полином $p(n)$, такъв, че $N$ е $O(p(n))$, казваме, че алгоритъмът е полиномиален. Ако $N$ е $O(log_2 n)$ алгоритъмът е логаритмичен, а ако $N$ е $O(a^n)$, алгоритъмът е експоненциален. За най-добри се считат логаритмичните алгоритми, защото при тях броят на елементарните операции нараства много слабо с обема на входните данни. Приложимостта на полиномиалните алгоритми е сравнително добра, докато експоненциалните са най-нежелани.

Изследването на задачите, за които не са известни полиномиални алгоритми, е от специален интерес. Ако можем да проверим с полиномиален алгоритъм дали едно евентуално решение удовлетворява условията на такава задача, тя се нарича $\mathcal{NP}$ - трудна. Обект на настоящата дисертация са именно $\mathcal{NP}$ - трудни класификационни задачи.

Тези $\mathcal{NP}$ - трудни задачи, за които се счита, че не допускат решаване с полиномиален алгоритъм, се наричат $\mathcal{NP}$ - пълни. За една задача може да се докаже, че е $\mathcal{NP}$ - пълна чрез свеждането до нея (по начин, който е описан, например, в [8]) на някоя $\mathcal{NP}$ - пълна задача. Тази техника е илюстрирана в [146], където е представен първият списък от $\mathcal{NP}$ - пълни задачи. Сред тях са задачите за

А. съществуване в граф на клика със зададен размер;

Б. съществуване на разбиване на графа на даден (или по-малък от дадения) брой клики.

На пръв поглед задачата за съществуването на дизайни, кодове или спредове с дадени параметри изглежда близка до задача А, а въпросът за разрешимостта на един дизайн - до частен случай на задача Б (при еднаква фиксирана дължина на кликите). За тези задачи, обаче, не е доказано, че са $\mathcal{NP}$-пълни. Доказано е, че въпросът за вложимостта на дадена Щайнерова система от тройки в Щайнерова система от тройки с по-големи параметри е $\mathcal{NP}$-пълна задача [64].

На алгоритмите за класификация на дизайни и кодове е посветена книгата *Classification algorithms for codes and designs* на Kaski и Östergård [151]. В нея те разглеждат задачи за намиране (listing) на всички решения и задачи за класификация, като при последните се търси само по един представител от всеки клас на еквивалентност. В книгата е обърнато специално внимание на това, че действителната ефективност на класификационните алгоритми зависи и от обема на изходните данни, от начина, по който се съхраняват, от това дали вече конструираните решения са необходими при намирането на още такива, както и от методите за определяне дали две решения са еквивалентни. Също така е доказано, че ако са известни част от редовете (стълбовете) на матрицата на инцидентност на 2-$(v,k,\lambda)$ дизайн, то задачата за разширяването на това частично решение по редове (стълбове) е $\mathcal{NP}$-пълна.

Доколко разглежданите $\mathcal{NP}$ - трудни класификационни задачи са $\mathcal{NP}$ - пълни не е обект на настоящата дисертация. Разработените алгоритми са оптимизи-

рани така, че да работят добре за някакво крайно множество от параметри и да доведат до решаването на конкретни задачи. Този подход се различава от теоретичния, защото понякога алгоритъм с много голяма асимптотична сложност може да се окаже най-подходящ за дадена задача. Такъв пример са алгоритмите, реализиращи търсене с връщане (backtrack search), при което по някакви критерии се отхвърлят част от частичните решения (early pruning). Те са с експоненциална сложност, но се използват успешно за построяване и (или) класификация на най-различни комбинаторни конфигурации, в това число дизайни, кодове, графи, Адамарови матрици [148], [151]. Целта на настоящата работа е решаването на конкретни класификационни задачи и затова ни интересува преди всичко поведението на използваните алгоритми за дадените параметри и приложимостта им при други близки задачи.

### 1.2.2 Изчерпващо търсене с връщане с отхвърляне на частични решения

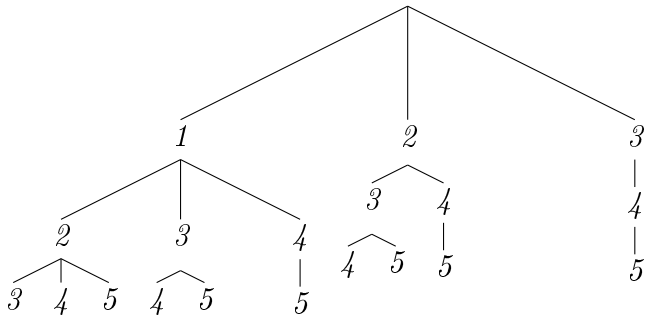
#### Множество на търсене и базова задача

Искаме да построим $m$-елементни подмножества на множество от $n$ еднотипни елемента (числа, вектори, циркулантни матрици, блокове на дизайн и т.н.), записани в масива $A$, който е сортиран във възходящ ред според зададена лексикографска наредба. Това $n$-елементно множество ще наричаме множество на търсене. Всяко негово подмножество конструираме на $m$ стъпки, като на $k$-тата от тях избираме подред всички възможни стойности за съответния елемент $S[k]$ с рекурсивната функция PutElement. Текущо построеното подмножество се получава подредено в лексикографски нарастващ ред в масива $S$, който има $m+1$ елемента и $S[0]=0$. Елементите на $S$ не са елементи на $A$, а номерата на елементите от $A$.

```
PutElement(int k)
{
    for i = S[k-1]+1, S[k-1]+2, ... n-(m-k)
    {
        S[k] = i
        if k < m then PutElement(k+1)
        else save or print the set S
    }
}
```

**Пример 1.2.1.** *Нека $n=5$, $m=3$ и $A=\{1,2,3,4,5\}$. С извикването на PutElement(1) ще бъдат конструирани в лексикографски ред* 10*-те* 3*-елементни*

*подмножества на $A$, т.е.* 123, 124, 125, 134, 135, 145, 234, 235, 245 *и* 345. *Функцията PutElement реализира обхождане в дълбочина на съответното дърво на решенията.*




Броят на елементарните операции е $N = Mn(n-1)\dots(n-m+1) = O(n^m)$, където $M$ е целочислена константа, т.е. алгоритъмът е експоненциален. В действителност, обаче, ефективността му съществено зависи от наличието на допълнителни конструктивни условия, както и от включването на тестове за минималност и разширимост на частичните решения.

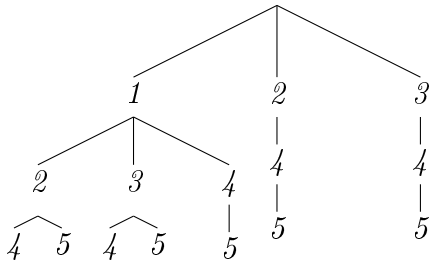
**Конструктивни условия**

Обикновено в задачите се търсят не всички подмножества, а само тези от тях, отговарящи на някакви условия, които по-нататък ще наричаме конструктивни условия. Нека Condition(k,i) е функция, която проверява дали ще са изпълнени конструктивните условия, ако към частично решение с $k-1$ елемента добавим $i$-тия елемент на масива $A$. С нея PutElementC (модификация на PutElement) ще намери само тази част от подмножествата, която отговаря на условията.

```
PutElementC(int k)
{
    for i = S[k-1]+1, S[k-1]+2, ... n-(m-k)
    {
        if Condition(k,i)
        {
            S[k] = i
            if k < m then PutElement(k+1)
            else save or print the set S
        }
    }
}
```

**Пример 1.2.2.** *Нека $n = 5$, $m = 3$ и $A = \{1,2,3,4,5\}$ и конструктивното условие да изисква никое подмножество да не съдържа едновременно и втория, и третия*

*елемент на A. С PutElementC(1) ще бъдат конструирани 7-те подмножества на A, т.е.* 124, 125, 134, 135, 145, 245 *и* 345. *Конструктивното условие отрязва част от клоните на дървото.*




**Тест за минималност на частичните решения**

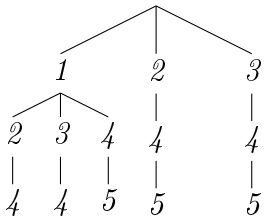
При реалните задачи, за които става въпрос, броят на всички решения, които удовлетворяват конструктивното условие, е твърде голям и построяването и съхранението на всичките обикновено е невъзможно. Класификацията им се прави с точност до еквивалентност.

Тестът за минималност на частичните решения проверява дали някое еквивалентно преобразование не изобразява текущото частично решение в частично решение, което е лексикографски по-малко от него [151, раздел 7.1.2]. Нека New(k,i) е функция, която проверява дали ще получим ново решение (което не е еквивалентно на лексикографски по-малко от него), ако към частично решение с $k-1$ елемента добавим $i$-тия елемент на масива $A$. С нея PutElementCN ще намери само нееквивалентните подмножествата.

```
PutElementCN(int k)
{
      for i = S[k − 1] + 1, S[k − 1] + 2, . . . n − (m − k)
      {
            if Condition(k,i) if New(k,i)
            {
                  S[k] = i
                  if k < m then PutElement(k+1)
                  else save or print the set S
            }
      }
}
```

**Пример 1.2.3.** *Нека* $n = 5$, $m = 3$ *и* $A = \{1, 2, 3, 4, 5\}$ *и конструктивното условие да изисква никое подмножество да не съдържа едновременно и втория, и третия елемент на A и нека две решения да са еквивалентни, ако се получават едно от друго чрез замяна на четвъртия елемент на A с петия и обратното, т.е. при пермутацията (1)(2)(3)(4,5). С PutElementCN(1) ще бъдат конструирани само*

*лексикографски най-малките представители на* 5*-те класа на еквивалентност, т.е.* 124, 134, 145, 245 *и* 345.




**Тест за разширимост на частичните решения**

В някои случаи е възможно с проверка на прости условия да се установи, че частичното решение е неразширимо до пълно решение (виж, например, раздели 6.2.1 и 6.5.2). Както и теста за минималност, ранният отрицателен тест за разширимост може да спести построяването на множество по-големи неразширими частични решения (по отрязаните клони) и съществено да ускори работата на алгоритъма.

### 1.2.3 Базирани на търсенето с връщане методи за класификация, при които се генерират само нееквивалентни решения

В книгата *Classification algorithms for codes and designs* [151] са изброени няколко метода. Общото между тях е, че на различните стъпки по някакъв начин се отхвърлят или изобщо не се генерират еквивалентни частични решения. Подходът, който най-често е използван в настоящата дисертация, е разновидност на метода, наречен ***подредено генериране (orderly generation)***, който за пръв път описват независимо един от друг Faradžev [86] и Read [206]. Принципът на подреденото генериране е, че на всяка стъпка се отхвърлят всички частични решения, които не са в зададена канонична форма. На практика тестът за минималност прави същото, защото можем да приемем, че в канонична форма е само лексикографски най-малкото решение. Такъв тест е използван в голямата част от разгледаните тук задачи и на него е обърнато специално внимание, но за разлика от подреденото генериране, в алгоритмите от дисертацията този тест обикновено ***не се използва на всяка стъпка***, а се прилага само на някои от частичните решения и на всички пълни решения.

Ако тестът за минималност е отрицателен при малки частични решения, това спестява построяването на множество частични решения (по отрязаните клони), които се разширяват до еквивалентни на лексикографски по-малки решения и по този начин бързодействието на програмите може значително да се увеличи. Има случаи, обаче, когато отхвърлените малки частични решения не са много, а тестът за минималност бави и в резултат на това бързодействието може да намалее, вместо да се увеличи. Затова този тест е изключително важна част от алгоритъма. От една страна той трябва да бъде максимално бърз, от друга - изключително преци-

зен, защото при обекти, за които са наложени множество допълнителни условия, вероятността за погрешно изхвърляне на частични решения е голяма. Освен това, трябва да се прецени на кои частични решения да се прилага теста с оглед на постигането на максимално бързодействие.

Приоритет при разработването на конкретните алгоритми от дисертацията е и намирането на подходящи бързи тестове за разширимост на някои от частичните решения. Тестовете за минималност и разширимост, както и минимизирането и удобната подредба на множеството на търсене, са основните инструменти, с които постигаме желаната практическа ефективност на алгоритмите.

### 1.2.4 Паралелна реализация на изчерпващото търсене с връщане

**Въведение**

Разглежданията от настоящия раздел не се използват в следващите глави на дисертацията. Те са включени тук, за да отговорят на естествено възникващият въпрос относно възможностите за използване на по-голям изчислителен ресурс при решаването на подобни задачи. В продължение на няколко години авторът имаше отдалечен достъп до паралелния компютър Blue Gene на Националния център за суперкомпютърни приложения в режим, в който потребителят разполага с 512 независими процесора, като всеки от тях изпълнява едно копие на програмата, има 512 MB памет и ползва общо дисково пространство.

Опити да се атакуват изчислително тежки параметри с помощта на паралелна реализация на софтуера са правени за два вида от разглежданите в дисертацията комбинаторни структури: при класификацията на книжни спредове на $PG(7,2)$ с дадени свойства (част 4.5) и при тази на максимални оптични ортогонални кодове [23]. Тук ще опишем използвания начин за разпаралелване на търсенето с връщане, който е принципно един и същ в двата случая.

Основните изисквания към софтуера за мултипроцесорен компютър са да минимизира времето за комуникация между процесорите и времето, когато някой от процесорите не работи (idle time). Ускорението (speed-up) $S(n)$ на паралелен алгоритъм се дефинира като

$$S(n) = \frac{t_s}{t_n},$$

където $t_s$ е времето, необходимо на съответния последователен алгоритъм, а $t_n$ – времето на работа на паралелния алгоритъм на $n$ процесора.

Една паралелна реализация се нарича *скалируема* с точност до $n$ процесора, ако на $n$ процесора работи по-бързо отколкото на по-малко от $n$ процесора. Подобра скалируемост се постига чрез равномерно натоварване на процесорите и минимизиране на комуникациите между ядрата.

В [147] е показано, че търсенето с връщане може да бъде разпаралелено с ускорение, което е оптималното с точност до малка константа и са разгледани два

типа паралелни алгоритми за търсене с връщане - случаен (randomised) и с глобален контрол (global control). При глобален контрол един или няколко процесора се използват единствено за управление на комуникацията, докато при случайния начин свободният процесор се обръща за работа към произволно избран процесор.

В нашата реализация използваме глобален контрол като избираме за *мениджър* процесора с номер 0. Този процесор изпълнява само функции, свързани с комуникацията с останалите. За предаване на съобщенията между процесорите използваме MPI( Message Passing Interface).

**Разпаралелване на кода**

Да разгледаме търсене в дълбочина, което вместо с рекурсивна функция е реализирано чрез стек. При последователния алгоритъм винаги работим с последния елемент, който сме записали в стека, т.е. използваме го в текущото решение, премахваме го от стека и поставяме на негово място наследниците му. Нищо няма да се промени, ако заменим стека с *deque* (double ended queue). На C++ използването на дек е просто. Предварително дефинираме структура, която в нашия случай е достатъчно да има два члена: $k$ - номерът на поредния елемент от текущо конструираното подмножество $S$ и $e$ - номера на елемента от $A$, който му е присвоен ( $S[k] = e$). Текущото съдържание на $S$ е променлива, чиито стойности един процесор изпраща на всеки процесор, на който дава работа. Конструирането на подмножествата е илюстрирано с псевдокода на фигура 1.2.

Винаги обработваме най-последния елемент, записан в дека (с най-голяма стойност на $k$), а даваме на друг процесор първия, защото той е с най-малка стойност на $k$ и неговата обработка се очаква да продължи най-дълго, а ние искаме да дадем на другия процесор работа за възможно най-дълго време, за да минимизираме комуникацията.

***Комуникация***: Във всеки момент мениджърът знае кой процесор има в своя дек елемент с текущо най-малката стойност на $k$ и затова е най-подходящ да раздава работа на другите. Когато някой процесор завърши задачата си, мениджърът му нарежда от кой процесор да получи работа, или, евентуално, да приключи. Всеки от останалите процесори изпраща съобщение на мениджъра когато остане без работа, или ако се смени най-малката стойност на $k$ в дека му. Той изпраща работа на друг процесор, само ако мениджърът му каже.

***Инициализация***: В програмата за класификация на спредове в началото само процесор с номер 1 започва търсенето и постепенно раздава работа на други, а те на други и т.н. При това, обаче, към времето, необходимо на програмата, се добавя забележима константа. При класификацията на оптични ортогонални кодове това е избегнато, като в началото на работата на алгоритъма, всички върхове на дървото на решенията от първо ниво се конструират от всички процесори.

Фигура 1.2: Конструиране на подмножествата

```
struct Selement
{
   int k;
   int e;
};
Selement C, Cb, Cf;
deque<Selement> D;
while(!D.empty())
{
      C = D.back( ); //присвояваме на C посл. елемент от дека
      D.pop_back(); //изтриваме посл. елемент от дека
      k=C.k+1;
      if(C.k==m) WriteRes(); //записваме резултат
      else for(i=n-(m-k); i>S[k-1]+1; i - -) // възможностите за S[k]
      {
            if Condition(k,i) if New(k,i)
            {
                  Cb.k = k;
                  Cb.e = i;
                  D.push_back(Cb); //записваме Cb най-отзад в дека
            }
      }
      ...
      if(flag) //нареждане да дадем работа на друг процесор
      {
            Cf = D.front( ); //присвояваме на Cf първия елемент от дека
            D.pop_front(); //изтриваме първия елемент от дека
            ... // изпращаме Cf и масива S на другия процесор
      }
}
```

Нека техният брой да е $F$. Така всеки процесор с номер $0 < p \leq F$ започва да разширява $p$-тия връх от първо ниво.

***Обработка на резултатите***: Всеки процесор записва получените резултати в отделен файл. В последствие тези файлове архивираме и прехвърляме на PC за допълнителна обработка и обобщаване на полученото. Само някои кратки данни (брой на конструираните обекти и т.н.) се изпращат от всеки процесор на мениджъра и той ги обобщава и записва накрая в един заключителен файл.

Описаният подход води до много добро ускорение спрямо паралелния алгоритъм. За съжаление обаче, тежките случаи, които бяха атакувани с него, останаха въпреки всичко нерешени. И това бързодействие се оказа недостатъчно за класификацията на всички книжни спредове с параметри $(7, 2, 5, 3)_{1(2)}$ (раздел 4.5.2), а при оптичните ортогонални кодове възникнаха проблеми с недостиг на памет,

както и със съхранението на получените решения и в резултат на това с паралелните програми добавихме само няколко набора параметри към класификацията на максималните $(v, 5, 2, 1)$ оптични ортогонални кодове [20]. Едновременно с това вече нямаме достъп до Blue Gene, а се появиха персонални компютри с голяма изчислителна мощ. Това, обаче, не значи, че паралелният подход няма бъдеще. Първите резултати са много обнадеждаващи, но успеваемостта зависи и от конкретните задачи. Понякога следващият отворен случай може да е хиляди пъти по-тежък от този за по-малките съседни параметри и тогава и разпаралелването няма как да помогне.

### 1.2.5 Класификация на дизайни със зададени автоморфизми с локалния подход

Този алгоритъм също е експоненциален и базиран на търсене с връщане, но е доста ефективен, защото разделя проблема на няколко последователни подзадачи. Тук представяме накратко само основните идеи според [14]. Локалният подход се използва за построяване на всички дизайни с фиксирани параметри, които имат предварително зададени автоморфизми. Той е разработен за пръв път в [16] и се базира на факта, че за матрицата на инцидентност на дизайн с автоморфизъм от прост ред съществува специфично представяне с помощта на циркуланти (виж, например фигура 2.1). Някои работи, в които е използван този метод са: [4], [5], [6], [7], [121], [140], [141], [142], [152], [156], [162], [178], [179], [180], [183], [184], [213], [218], [219], [227], [228], [229], [231], [232], [242], [247]. Неразривно свързано с локалния подход е понятието *орбитна матрица*.

Нека $D$ е дизайн с група от автоморфизми $G$ от прост ред $p$. Да означим с $O_1, O_2, ..., O_m$ точковите и с $O'_1, O'_2, ..., O'_n$ блоковите орбити спрямо $G$. Нека $m_{ij}$ е броя на точките от $O_i$, съдържащи се в кой да е блок от $O'_j$ $(i = 1, 2, ..., m,\ j = 1, 2, ..., n)$.

**Орбитна матрица** на дизайна спрямо тази група от автоморфизми наричаме матрица $\mathcal{M} = (m_{ij})_{m \times n}$.

Нека $O_1, O_2, ..., O_{m'}$ са нетривиалните точкови и $O'_1, O'_2, ..., O'_{n'}$ – нетривиалните блокови орбити спрямо $G$, $m' \leq m$ и $n' \leq n$.

**Съкратена орбитна матрица** на дизайна спрямо $G$ наричаме матрицата $M' = (m_{ij})_{m' \times n'}$.

**Съкратена матрица на инцидентност** на дизайна спрямо дадена група от автоморфизми $G$ наричаме матрицата, която се получава като премахнем от матрицата на инцидентност редовете и стълбовете, съответстващи на фиксираните от $G$ точки и блокове.

За да приложим локалния подход за конструиране на 2-$(v,k,\lambda)$ дизайни, трябва първо да изясним какви автоморфизми могат да притежават дизайните с такива параметри и да изберем каква ще е зададената група от автоморфизми. После

трябва да установим на какви условия отговарят орбитните матрици спрямо група от автоморфизми $G$ от прост ред $p$, която е подгрупа на зададената група от автоморфизми. Условията са повече, ако зададената група не съвпада с $G$. След тези специфични за дадения случай разглеждания, преминаваме към построяването на дизайните, което става на няколко етапа:

- Намиране на всички нееквивалентни съкратени условни орбитни матрици (Наричат се *условни*, защото е възможно някои от тях да не са разширими до дизайни. По-нататък ще изпускаме думата *условни*);
- Конструиране на нееквивалентните съкратени матрици на инцидентност;
- Добавяне на фиксираните точки и блокове по всевъзможните начини;
- Отсяване на неизоморфните дизайни.

Основно достойнство на разработения в [14] алгоритъм е, че отсява значителна част от еквивалентните решения още в процеса на получаването им.

**Намиране на съкратените орбитни матрици**

Входните данни, необходими за работата на алгоритъма, включват условията, на които трябва да отговарят елементите на съкратената орбитна матрица $M$ и информация за това кои от нейните редове са взаимно-заменими. Два реда на $M$ наричаме *взаимно-заменими*, ако при размяна на местата на съответните редове от условната орбитна матрица, условията за $M$ остават непроменени. Решенията за съкратената орбитна матрица получаваме по следния начин:

• Първоначално намираме всички нееквивалентни възможности за всеки един от редовете на матрицата $M$.

• Първия ред на $M$ подреждаме в лексикографски нарастващ ред.

• Ако сме намерили $i$ реда от матрицата, трябва да проверим кои от пермутациите на всяка от възможностите за $i+1$-вия ред отговарят на условията за $M$. Пермутациите генерираме в лексикографски нарастващ ред.

• Тест за минималност: След генериране на поредната възможност за матрицата от първите $i$ реда на $M$ (Нека я означим $M^i$), проверяваме дали тя не е еквивалентна на някоя от вече разглежданите комбинации от $i$ реда. За целта за всяка пермутация на взаимно-заменимите измежду първите $i$ реда проверяваме дали след подреждане на стълбовете в лексикографски нарастващ ред не се получава матрица, по-малка от $M^i$.

**Намиране на съкратените матрици на инцидентност**

Замяната на елементите на съкратената орбитна матрица $M$ с циркуланти от ред $p$ (или *разширяването* на $M$) до получаване на всички нееквивалентни решения за съкратената матрица на инцидентност $A$ представлява най-съществената

част от пресмятанията. Обикновено тя е и най-бавната, въпреки че има и изключения (например, генерирането на всички нееквивалентни орбитни матрици спрямо автоморфизъм от ред 5 на Адамаров 2-(63,31,15) дизайн в раздел 2.6 се оказа много по-бавно от разширяването им спрямо предположения автоморфизъм от ред 10).

Ще наричаме *i-ти хоризонтален слой* на съкратената матрица на инцидентност тези $p$ реда от нея, които съответстват на точките от $i$-тата точкова орбита, а *j-ти вертикален слой* – стълбовете, съответствуващи на блоковете от $j$-тата блокова орбита.

Алгоритъм за разширяване на орбитните матрици до дизайни:

1. Входните данни включват матрицата $M$ и таблица със скаларните произведения на всеки два реда на съкратената матрица на инцидентност.

2. Всички циркуланти от ред $p$, с които ще заместваме (нека са $s$ на брой), са номерирани така, че циркулантата с номер $lp+a$ се получава от тази с номер $lp+1$ чрез циклично завъртане на дясно или нагоре $a-1$ пъти, $l=0,1,...,\frac{s}{p}-1$, $a=1,2,...,p$.

Особено важно за бързодействието на програмата е, че вместо със самите циркуланти работим с техните номера, т.е. строим целочислена матрица (нека я означим $Q=(q_{ij})_{\frac{v-f}{p}\times\frac{b-h}{p}}$) с размерите на $M$, а не на съкратената матрица на инцидентност $A$, което много облекчава изчисленията.

3. Елементите на $Q$ намираме последователно. За поредния елемент $q_{ij}$ опитваме всички възможни номера на циркуланти с $m_{ij}$ единици в ред или стълб, като с всяка от тях правим опит за доизграждане на матрицата.

4. За намаляване броя на изоморфните решения фиксираме спрямо циклично завъртане първия ненулев елемент от всеки ред и стълб на $M$, т.е. този елемент се замества само с едната (тази с най-малък номер) измежду $p$ циркуланти, които се получават една от друга чрез циклично завъртане.

5. Тест за минималност - прилагаме го след построяване на цял ред от $Q$. Той се състои от две части:

**А**) Проверяваме дали намерената част от матрицата не е еквивалентна на лексикографски по-малка от нея. За целта генерираме всички пермутации на редовете на $Q$, при които $i$-тият ред отива в $j$-тия само ако $i$-тият ред на орбитната матрица е еквивалентен на $j$-тия с точност до пермутация.

**Б**) В резултат на **А** решенията за $Q$ са нееквивалентни едно на друго, но сред решенията за съкратената матрица на инцидентност, които се получават непосредствено от тях, в общия случай има и немалко еквивалентни. Затова прилагаме всевъзможните пермутации на редовете в рамките на хоризонталните слоеве и на стълбовете в рамките на вертикалните слоеве, които довеждат всички циркуланти в циркуланти (тези пермутации са, всъщност, автоморфизмите на $Z_p$). Тогава проверяваме както в **А** дали матрицата от новополучените циркуланти не е екви-

валентна на някое предишно, лексикографски по-малко решение за $Q$.

#### Добавяне на фиксираните точки и блокове

Често пъти при една и съща фиксирана част с помощта на пермутации на хоризонталните и вертикални слоеве на съкратената матрица на инцидентност $A$ можем да получим няколко неизоморфни дизайна. За удобство вместо пермутации на хоризонталните и вертикални слоеве на $A$ генерираме възможните пермутации на редовете и стълбовете на частта от матрицата на инцидентност, която съответствува на фиксираните точки и блокове. Получените по този начин дизайни проверяваме за изоморфизъм помежду си.

Алгоритъмът и съответният софтуер на автора [14], както и техни модификации са използвани в глава 2, в някои от описаните подзадачи от част 3.2 и в част 5.6 на дисертацията.

### 1.2.6 Класификация на оптимални оптични ортогонални кодове

Алгоритъмът е описан в [1] и [18].

#### Цели

Класификация с точност до мултипликативна еквивалентност на $(v, k, \lambda, 1)$ ООС, които имат $s$ кодови думи

#### Множество на търсене

Думите на кода избираме измежду всички *възможни кодови думи*, които предварително подреждаме по следния начин: Нека $\varphi_0, \varphi_1, \ldots, \varphi_{m-1}$ да са автоморфизмите на $Z_v$, като $\varphi_0$ е идентитета. Конструираме масив от всички множества от $k$ елемента на $Z_v$, които биха могли да са кодови думи, т.е. които удовлетворяват автокорелационното свойство и са по-малки от всички свои транслации. Групата от автоморфизми на $Z_v$ ги разбива на орбити. Лидер на орбита наричаме най-малкия нейн елемент. Орбитите подреждаме в масива според лексикографската наредба на лидерите им. След всеки лидер в масива има $m-1$ множества, в които той се изобразява при $\varphi_1, \varphi_2, \ldots, \varphi_{m-1}$. Така получаваме масива $L$ от фигура 1.3.

#### Конструиране на кодовете

За да изберем кодовите думи на оптичния ортогонален код измежду всички възможни думи, записани в масива $L$, прилагаме търсене с връщане. При описаната подредба думи от орбити с дължина по-малка от $m$ се срещат повече от един път в $L$, но ние отбелязваме повторените и ги прескачаме при търсенето. Работим с номерата на възможните кодови думи от $L$, затова в кода всяка следваща кодова

дума е с по-голям номер от предходната. При търсенето на първата кодова дума избираме само измежду лидерите на класовете на еквивалентност.

Фигура 1.3: Масив с възможните кодови думи

$$
\begin{array}{lcl}
\mathbf{L_0} & & \\
L_1 & = & \varphi_1 L_0 \\
L_2 & = & \varphi_2 L_0 \\
\vdots & & \\
L_{m-1} & = & \varphi_{m-1} L_0 \\
\mathbf{L_m} & & \\
L_{m+1} & = & \varphi_1 L_m \\
L_{m+2} & = & \varphi_2 L_m \\
\vdots & & \\
L_{2m-1} & = & \varphi_{m-1} L_m \\
\vdots & & \\
\mathbf{L_{im}} & & \\
L_{im+1} & = & \varphi_1 L_{im} \\
L_{im+2} & = & \varphi_2 L_{im} \\
\vdots & & \\
L_{(i+1)m-1} & = & \varphi_{m-1} L_{im}
\end{array}
$$

Алгоритъмът е експоненциален, но бързината на теста за минималност на частичните решения и възможността за прилагане на ефективен тест за разширимост го прави приложим в много случаи, които представляват практически интерес.

**Тест за минималност на частичните решения**

С него проверяваме дали текущото решение може да бъде изобразено в лексикографски по-малко при прилагане на някой от автоморфизмите на $Z_v$. Нашата подредба на възможните кодови думи е удобна за ефективното изпълнение на теста за минималност, защото при нея е ясно как те се изобразяват една в друга при даден автоморфизъм на $Z_v$.

**Тест за разширимост на частичните решения**

Да приемем, че вече сме избрали $r$ кодови думи от кода. Нека $T$ да е типът на $r$-тата кодова дума и нека $d$ да е броят различни разлики, покрити от избраните $r$ кодови думи. Тъй като търсим единствено кодове с $s$ кодови думи, то, заради наредбата в масива $L$, типът $T$ на оставащите да бъдат избрани кодови думи трябва да е не по-малък от типа $T_r$ на избраната $r$-та кодова дума. Понеже $d + (s-r)T \leq v-1$, за $T$ получаваме $T_r \leq T \leq (v-d-1)/(s-r)$. Ако това условие не се изпълнява, то намереното частично решение е неприложимо, защото не може да бъде допълнено до код с $s$ кодови думи. Затова тестът за разширимост

е отрицателен. В следствие на това алгоритъмът се връща една стъпка назад и търси следваща възможност за $r-1$-вата кодова дума.

Всички класификационни алгоритми от глава 6 могат да бъдат считани за модификации на този алгоритъм.

### 1.2.7 Отделяне на неизоморфните дизайни

Една от основните трудности при конструирането на комбинаторни структури по каквито и да е методи, е да бъдат премахнати еквивалентните решения. В повечето случаи, разгледани в настоящата дисертация, те се отсяват напълно още в процеса на конструирането им с помощта на теста за минималност на частичните решения. Но понякога разглеждаме и свързани с тях дизайни, от които трябва да се отделят изоморфните. Освен ако специално не е обяснено нещо друго, в такива случаи използваме софтуер на автора, който е описан в [14]. Ще разгледаме и тук основните идеи на този алгоритъм, понеже в следващите глави ще става въпрос и за негови модификации за други сродни обекти.

#### Използване на инварианти

Разглеждаме дизайни с параметри 2-$(v,k,\lambda)$. Инварианти на точките (блоковете) наричаме такива техни характеристики, които не се променят при изоморфизъм, т.е. всеки изоморфизъм между два дизайна изобразява точките (блоковете) в точки (блокове) с еднакви инварианти.

Много удачни се оказаха следните инварианти ($I_1$), взаимствани от [12]:

- За всяка точка $P$ намираме вектора ($m_0$, $m_1$, ..., $m_\lambda$), където $m_j$ ($j = 0,1,...,\lambda$) е броя на двойките точки $(Q,R)$, различни от $P$ и такива, че $P$, $Q$ и $R$ се съдържат едновременно точно в $j$ блока.

- За всеки блок $B$ пресмятаме вектора ($n_0^{(a)}$, $n_1^{(a)}$, ..., $n_{v-3}^{(a)}$), където $n_i^{(a)}$ ($i = 0,1,...,v-3$, $a = 1,2,...,\lambda$) е броят на двойките блокове $(V,W)$, различни от $B$ и такива, че съществуват точно $i$ други блока, съдържащи поне $a$ общи точки с всеки от блоковете $B$, $V$, $W$.

Ако сортираме по някакъв начин инвариантите на точките и блоковете, получаваме инварианта на дизайна, т.е. ако броят на точките и блоковете с дадени инварианти не е един и същ за два дизайна, то те са неизоморфни и по-нататъшен тест за изоморфизъм не е необходим. Това е особено важно, защото инвариантите намираме с полиномиален алгоритъм.

#### Прескачане на пермутации

Нека два дизайна $D_1$ и $D_2$ имат едни и същи инварианти. Според дефиницията за изоморфизъм, за да докажем, че са неизоморфни, трябва да генерираме всевъзможните пермутации на редовете на матрицата на инцидентност на $D_1$ и за

всяка от тях да проверим дали не съществува пермутация на стълбовете ѝ, която да води до матрицата на инцидентност на $D_2$. Добре е известно, че генерирането на всички пермутации на повече от 12 елемента е вече толкова бавно, че изчисленията стават безсмислени. Затова е изключително важно да бъдат пропуснати част от пермутациите, които очевидно не водят до изоморфен дизайн. От начина, по който се прави това, зависи ефективността на програмата.

Броят на разглежданите пермутации на точките на $D_1$ може да намалее значително, ако генерираме само тези от тях, за които е в сила следното *условие за инвариантите*: Ако пермутацията изобразява точката $P_i$ в точката $P_j$, то точките $P_i$ от $D_1$ и $P_j$ от $D_2$ трябва да имат едни и същи инварианти.

Нека $D_1$ и $D_2$ имат матрици на инцидентност съответно $L_1$ и $L_2$. Генерираме пермутации на редовете на $L_1$ в лексикографски ред. Нека $\varphi_n$ е текущата пермутация, която трансформира $L_1$ във $\varphi_n L_1$. Ако множеството от стълбовете на $\varphi_n L_1$ съвпада с това на $L_2$, то двата дизайна са изоморфни. Ако това не е така, трябва да генерираме следващата пермутация $\varphi_{n+1}$. Дали тя да бъде първата, следваща $\varphi_n$ в лексикографски ред и отговаряща на условието за инвариантите, или можем да прескочим известен брой пермутации? За да отговорим на този въпрос, разглеждаме матриците $L_2^i$ и $\varphi_n L_1^i$, състоящи се от първите $i (i = 1, 2, ..., v)$ реда на $L_2$ и $\varphi_n L_1$ и определяме най-голямото $i$, за което множеството от стълбовете на $L_2^i$ съвпада с това на $\varphi_n L_1$. Тогава за $\varphi_{n+1}$ избираме първата в лексикографски ред пермутация след $\varphi_n$, за която $\varphi_n L_1^{i+1}$ се различава от $\varphi_{n+1} L_1^{i+1}$. Така прескачаме $(v - i - 1)!$ ($i$ е между 1 и $v$) пермутации при генерирането на всяка нова и това прави изчисленията възможни.

Инвариантите на блоковете са от полза, когато определяме дали един стълб на $\varphi_n L_1$ е стълб и на $L_2$, защото можем да пропуснем сравненията между стълбове, съответствуващи на блокове с различни инварианти.

**Пример 1.2.4.** *Започваме проверката за изоморфизъм на 2-(7,3,3) дизайните $D_1$ и $D_2$ с матрици на инцидентност $A_1$ и $A_2$.*

| $A_1$ | $A_2$ |
|---|---|
| 111111111000000000000 | 111111111000000000000 |
| 111000000111111000000 | 111000000111111000000 |
| 111000000000000111111 | 110100000100000111110 |
| 000111000111000111000 | 001011000011000111001 |
| 000111000000111000111 | 000100110011100000111 |
| 000000111111000000111 | 000011001000111100110 |
| 000000111000111111000 | 000000111100011011001 |

*Матриците от първите им два реда са еквивалентни помежду си, но от първите три реда не са. Затова първата пермутацията, която разглеждаме, не е $\varphi = (1)(2)(3)(4)(5)(6,7)$, а $\varphi = (1)(2)(3,4)(5)(6)(7)$. Проверяваме дали точки 3 и 4 на първия дизайн имат същите инварианти като съответно точки 4 и 3 на втория. Ако това не е така, преминаваме към пермутацията $(1)(2)(3,5)(4)(6)(7)$. В противен случай проверяваме дали множеството от стълбовете на $\varphi A_1$ не е*

*същото като множеството от стълбовете на $A_2$... Продължаваме по същия начин, като всяка следваща пермутация е лексикографски по-голяма от предишната разгледана и накрая установяваме, че вече няма по-големи пермутации, които да разглеждаме, т.е. двата дизайна са неизоморфни.*

Алгоритъмът е експоненциален, но в задачите от дисертацията проблеми с него възникват само при някои случаи на много големи матрици на инцидентност. Начини, чрез които е намерено решение за конкретни такива случаи са показани в раздели 2.6.4 и 6.3.3. Тук отново трябва да обърнем внимание на това, че за проверката за изоморфизъм съществуват много други алгоритми и софтуер. Някои от съавторите, например, използват софтуера *Nauty* [187] на McKay.

### 1.2.8 Намиране реда на пълната група от автоморфизми на дизайн

Както при проверката за изоморфизъм, така и за намирането на пълната група от автоморфизми на дизайни е използван софтуер на автора, реализиращ алгоритъм с експоненциална сложност и описан в [14]. Алгоритъмът се базира на следната теорема от теорията на крайните групи.

**Теорема 1.2.1.** *([11],[30],[248]): Ако $G$ е крайна група от пермутации, $G_x$ – стабилизаторът на точката $x$, а $|x^G|$ – дължината на орбитата ѝ, то*

$$|G| = |x^G||G_x|.$$

Групата от автоморфизми намираме с неколкократното прилагане на тази теорема. Отначало определяме дължината на орбитата на първата точка на дизайна, т.е. колко са точките, в които тя отива под действието на групата. След това определяме дължината на орбитата на втората точка спрямо стабилизатора на първата, на третата точка спрямо стабилизатора на първите две, ..., на последната точка спрямо стабилизатора на всички останали. Редът на пълната група от автоморфизми на дизайна е произведението на дължините на всички тези орбити.

Функцията CalcOrb намира дължината на орбитата на точката $cn$ при фиксирани точки $1, 2, \ldots, cn-1$. Изчистена от подробностите, тя е представена на фигура 1.4.

Функцията SetNextPerm(cn, a) намира лексикографски най-малката пермутация, при която $cn$ отива в $a$. Ако, обаче, се окаже, че $cn$ и $a$ имат различни инварианти, или, че вече има намерен автоморфизъм, изпращащ $cn$ в $a$, се търси пермутация, при която $cn$ отива в $a+1$ и т.н. Функцията връща точката, в която се изобразява $cn$.

Прескачането на неперспективни пермутации (като в раздел 1.2.7) е реализирано с функциите SameDes и PermRows. Функцията SameDes(i) проверява дали текущата пермутация довежда матрицата от първите $i$ реда на матрицата на инцидентност в себе си. Използването в нея на подходящи инварианти (виж раздел 1.2.7) може значително да я ускори. Функцията PermRows(cn, i) генерира след-

Фигура 1.4:

```
int CalcOrb(int cn) // where can row cn go to
{
	int i,a,orb;
	a = SetNextPerm(cn, cn+1);
	while(a <= ROWS)
	{
		for (i = cn + 1; i <= ROWS; )
		{
			if (SameDes(i))
			{
				i++;
				continue;
			}
			i = PermRows(cn, i);
			if (i == cn) break;
		}
		if (i != cn) orb=AddAut();
		a = SetNextPerm(cn, a+1);
	}
	 return orb;
}
```

ващата в лексикографски ред пермутация, която променя мястото на $i$-я или по-малък от него ред и връща номера на най-малкия преместен ред.

Функцията AddAut добавя намерен автоморфизъм към генераторите на групата и нови точки към орбитата на $cn$, като връща текущата дължина на тази орбита.

Софтуерът е използван в раздели 1.2.9 и 2.5.3 и в част 2.4, а за намиране на пълната група от автоморфизми на резолюция в част 5.3 е разработена модификация на този алгоритъм. Отново трябва да обърнем внимание на това, че за намиране на групата от автоморфизми съществуват много други приложими алгоритми и софтуер. В раздел 2.6.4, например, използваме софтуер на Bouyukliev [36], а някои от съавторите намират групите от автоморфизми с *Nauty* [187] на McKay.

### 1.2.9 Намиране на неизоморфните резолюции на дизайн

В раздели 3.2.4, 5.5.2 и 5.6.2 за намиране на неизоморфните резолюции на дизайн е използван софтуер на автора, описан в [14]. Ще се спрем накратко на алгоритъма.

За да определим дали един дизайн е разрешим, подреждаме блоковете му в лексикографски ред и започваме да изграждаме паралелните класове. Нека вече сме избрали част от блоковете в даден паралелен клас. Да означим с $x$ точката с

най-малък номер, която участващите до момента в този паралелен клас блокове не съдържат. Следващия блок трябва да изберем измежду тези, които все още не са избрани в никой паралелен клас и които съдържат точката $x$ и още $k-1$ точки с по-големи номера. Заради това, че съдържат точката $x$, тези блокове са съседни в лексикографската наредба и проверката дали някой от тях може да бъде включен в съответния паралелен клас става бързо. Ако се окаже, че няма подходящ блок, се връщаме назад и избираме следващата възможност за предходния блок от паралелния клас.

Тест за минималност прилагаме след конструирането на цяла резолюция. Той проверява дали някой от автоморфизмите на дизайна не преобразува решението в лексикографски по-малко от него. Ето защо, след конструиране на първата резолюция, генерираме всички автоморфизми на дизайна, като намираме пораждащо множество на тази група с алгоритъма от раздел 1.2.8. Тест за минималност на частичните решения може, според случая, да се прилага и след всеки конструиран паралелен клас.

Алгоритъмът е експоненциален, като ефективността му зависи от бързината на теста за минималност на частичните решения. Затова възникват проблеми при дизайни с много големи групи от автоморфизми, при които е невъзможна проверката по този начин за изоморфизъм на получените резолюции. Начини за справяне с проблема са дадени в глава 5.

Основните идеи в този алгоритъм са използвани в раздел 5.2.3 за конструиране на резолюции със зададени автоморфизми.

**Сравнение с други алгоритми.**

Друг доста популярен подход (използван от съавторите ми) към тази задача е първо да бъдат конструирани всички възможни паралелни класове и след това измежду тях да бъдат избрани класовете на резолюцията. При разглежданите тук задачи ефективността на двата алгоритъма е близка.

# Глава 2

# Симетрични дизайни и Адамарови матрици със зададени автоморфизми

Интересът към симетричните дизайни се основава на допълнителните свойства, които те притежават и на многобройните им приложения. Освен това тези дизайни са сравнително малко и сравнително трудни за построяване. Не случайно точно за симетричните дизайни е изведено такова силно необходимо условие за съществуване, каквото се явява теоремата на *Bruck-Ryser-Chowla* (1.1.4). Адамаровите дизайни, от своя страна, са симетрични дизайни с допълнителни свойства. Те могат да бъдат допълвани до Адамарови матрици. Резултатите в тази глава са получени с модификации на метода за конструиране на дизайни със зададени автоморфизми с локалния подход, описан в раздел 1.2.5.

## 2.1 Дефиниции, означения и предварителни резултати

Броят на блоковете на симетричен дизайн е равен на броя на точките му. *Дуален* на симетричния дизайн $D$ наричаме дизайн със същите параметри, точките на който съответстват на блоковете на $D$, а блоковете - на точките на $D$. Един симетричен дизайн е *самодуален* ако е изоморфен на своя дуален.

**Теорема 2.1.1.** *(Brauer 1941, Parker 1957, виж [26], p.36, I.4.8)*
*Автоморфизъм на симетричен дизайн фиксира равен брой точки и блокове.*

*Адамарова матрица от ред* $n$ наричаме $n \times n$ $(\pm 1)$-матрица $H$, за която $HH^t = nI$ (т.е. редовете са взаимно ортогонални). Две Адамарови матрици са *еквивалентни* ако едната може да бъде получена от другата чрез пермутация на редовете и стълбовете и умножение с $-1$ на някои редове или стълбове. Една Адамарова матрица е *самодулна* ако е еквивалентна на своята транспонирана. Под *автоморфизъм* на Адамарова матрица разбираме нейна еквивалентност със себе си.

Всяка Адамарова матрица може да бъде нормализирана, т.е. заменена с еквивалентна Адамарова матрица, чиито първи ред и стълб съдържат само 1-ци. Ако отстраним първия ред и стълб на нормализирана Адамарова матрица от ред $4m$ и заменим $-1$-те с 0-ли, получаваме симетричен 2-$(4m-1,2m-1,m-1)$ дизайн, наречен Адамаров 2-дизайн. От всеки Адамаров 2-дизайн $D$ можем да получим по единствен начин Адамарова матрица, както и Адамаров 3-$(4m,2m,m-1)$ дизайн, който съдържа блоковете на $D$ и тези на допълнителния му 2-$(4m-1,2m,m)$ дизайн (всеки негов блок съдържа точките, които не се съдържат в съответния блок на $D$) и една нова точка, която е инцидентна с блоковете на $D$.

Адамаровите матрици имат изключително интересни комбинаторни свойства и разнообразни приложения [73], [74], [210] и са напълно класифицирани за редове до 32 [145], [119], [153], [154], [220]. Само частични класификации са известни за по-големите редове (например [34], [75], [98], [99], [221], [229]), защото изчислителната сложност на класификацията нараства експоненциално. Класификационни методи, подобни на използваните в тази глава, са разработени, например, в [34], [152], [228], [229].

## 2.2 Алгоритъм за конструиране на симетрични дизайни със зададени автоморфизми

Симетрични дизайни, разбира се, могат да бъдат конструирани с използване на алгоритъма, описан в раздел 1.2.5. В случаи, когато това води до твърде дълги пресмятания, за задачите от настоящата глава са разработени негови модификации, чиято сложност също е експоненциална, но благодарение на използването на допълнителните свойства на симетричните дизайни, с тях стават решими някои конкретни задачи. Тук изреждаме само основните различия в тези модификации.

### 2.2.1 Промени в алгоритъма от раздел 1.2.5

- Вместо да опитваме всички начини за заместване на поредния елемент на съкратената орбитна матрица с циркуланта, отначало разширяваме всеки от нейните редовете до всевъзможните хоризонтални слоеве на съкратената матрица на инцидентност и ги запазваме.

- Съкратената орбитна матрица конструираме хоризонтален слой по хоризонтален слой, като за всяко от решенията за този слой проверяваме дали е съвместимо с предходните конструирани слоеве. Важното тук е, че за симетричен дизайн тази проверка включва пресмятане на скаларните произведения не само на редовете, но и на стълбовете на получената матрица, а това води до отхвърляне на много повече частични решения.

- Ако орбитната матрица съдържа еднакви редове, то без загуба на общност

приемаме, че съответните им хоризонтални слоеве са подредени в лексикографски ред.

### 2.2.2 Сравнение с други алгоритми

Най-популярен е алгоритъмът на Janko [122] за конструиране на симетрични дизайни с избрани автоморфизми, който се основава едновременно на локалния подход и на намирането на десните съседни класове на зададената група. Методът е особено ефективен при групи от непрост ред, докато в тази глава разглеждаме прости редове или диедралната група от ред 10, за която използваме лека модификация на алгоритмите за прост ред.

## 2.3 Симетрични 2-(69,17,4) дизайни с автоморфизми от ред 13

### 2.3.1 Въведение

Преди настоящото изследване бяха известни два симетрични 2-(69,17,4) дизайна [182] - единият принадлежи на безкрайния клас, конструиран от Shrikhande и Singhi [212], а другият е намерен от Božikov [40] с конструиране на всички (2 на брой) 2-(69,17,4) дизайни, инвариантни относно група на Frobenius от ред 39. Групата от автоморфизми и на двата дизайна е от ред 156. От симетричен 2-(69,17,4) дизайн могат да бъдат получени остатъчни дизайни с параметри 2-(52,13,4), каквито не бяха известни преди тази работа [182].

В тази част конструираме всички (четири) неизоморфни 2-(69,17,4) дизайна с автоморфизми от ред 13 и 16 остатъчни 2-(52,13,4) дизайна. Двата от намерените симетрични дизайни са изоморфни, съответно, на построените от Shrikhande и Singhi, и от Božikov. Другите два са нови.

### 2.3.2 Особености на орбитната матрица и класификационния алгоритъм

**Теорема 2.3.1.** *Ако $\alpha$ е автоморфизъм от ред 13 на 2-(69,17,4) дизайн, то $\alpha$ фиксира 4 точки и 4 блока.*

**Доказателство.** Автоморфизъм на симетричен дизайн фиксира равен брой точки и блокове [200]. Нека $D$ е 2-(69,17,4) дизайн с автоморфизъм $\alpha$ от ред 13, фиксиращ $f$ точки и $f$ блока. Понеже $\lambda = 4$, а нефиксиран блок не може да съдържа повече от една фиксирана точка, то всяка фиксирана точка се съдържа или в 4, или в 17 фиксирани блока. Да означим с $n$ броя на фиксираните точки, които са

в 4 фиксирани блока. Тогава

$$\left|\begin{array}{rcl} n\binom{4}{2}+(f-n)\binom{17}{2} & = & 4\binom{f}{2} \\ n & \leq & \dfrac{69-f}{13} \\ f & = & 4,17,30,43,56. \end{array}\right. \tag{2.1}$$

Системата има единствено решение в цели положителни числа, т.е. $f = 4$, $n = 4$. □

Нека $D$ е 2-(69,17,4) дизайн с автоморфизъм $\alpha$ от ред 13, фиксиращ 4 точки и 4 блока. Без загуба на общност приемаме, че $\alpha$ действа на точките и блоковете на $D$ както следва:

$\alpha = (0,\ 1,\ ...\ ,\ 12)(13,\ 14,\ ...,\ 25)\ ...\ (52,\ 53,\ ...,\ 64)(65)(66)(67)(68)$.

Тогава матрицата на инцидентност $A$ на $D$ е:

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} & A_{15} & z^t & z^t & z^t & z^t \\ A_{21} & A_{22} & A_{23} & A_{24} & A_{25} & u^t & z^t & z^t & z^t \\ A_{31} & A_{32} & A_{33} & A_{34} & A_{35} & z^t & u^t & z^t & z^t \\ A_{41} & A_{42} & A_{43} & A_{44} & A_{45} & z^t & z^t & u^t & z^t \\ A_{51} & A_{52} & A_{53} & A_{54} & A_{55} & z^t & z^t & z^t & u^t \\ z & u & z & z & z & 1 & 1 & 1 & 1 \\ z & z & u & z & z & 1 & 1 & 1 & 1 \\ z & z & z & u & z & 1 & 1 & 1 & 1 \\ z & z & z & z & u & 1 & 1 & 1 & 1 \end{pmatrix}$$

където $A_{ij}$, $i$=1, 2, ..., 5, $j$=1, 2, ..., 5 са циркуланти от ред 13, $u = (\underbrace{1, 1, ..., 1}_{13})$ и $z = (\underbrace{0, 0, ..., 0}_{13})$. Нека означим с $m_{ij}$ броя на 1-ците в ред на $A_{i,j}$. Тогава

$$\left|\begin{array}{lll} \sum\limits_{j=1}^{5} m_{1j} = 17, & \sum\limits_{j=1}^{5} m_{1j}^2 = 65 & \\ \sum\limits_{j=1}^{5} m_{ij} = 16, & \sum\limits_{j=1}^{5} m_{ij}^2 = 52, & i = 2,3,4,5 \\ & \sum\limits_{j=1}^{5} m_{i_1 j} m_{i_2 j} = 52, & i_1, i_2 = 1,2,...,5. \end{array}\right. \tag{2.2}$$

С точност до еквивалентност има само една орбитна матрица, за която (2.2) е в сила.

$$M = \begin{pmatrix} 1 & 4 & 4 & 4 & 4 \\ 4 & 3 & 3 & 3 & 3 \\ 4 & 3 & 3 & 3 & 3 \\ 4 & 3 & 3 & 3 & 3 \\ 4 & 3 & 3 & 3 & 3 \end{pmatrix}$$

Симетрията на тази матрица значително улеснява заместването на нейните елементи с циркуланти с използване на алгоритъма от раздел 2.2. Вторият, третият, четвъртият и петият ред (стълб) на $M$ са същите, така, че първо намираме всички възможни начини за разширяване на един такъв ред (стълб) и тогава сглобяваме от тях решения за цялата матрица, като считаме, че съответните им четири хоризонтални слоя от съкратената матрица на инцидентност са подредени в лексикографски ред.

### 2.3.3 Класификация. Остатъчни дизайни

Базисните блокове (без фиксираните) на четирите 2-(69,17,4) дизайна $D_1$, $D_2$, $D_3$ и $D_4$ са представени в Таблица 2.1. Дизайни $D_1$ и $D_2$ са дуални един на друг, затова $D_2$ не е даден. Неговата матрица на инцидентност е транспонираната матрица на инцидентност на $D_1$.

**Таблица 2.1: Базисни блокове на 2-(69,17,4) дизайни с автоморфизми от ред 13**

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $D_1$ | | | | | | | | | | | | | | | | |
| $B_0$ | 0 | 13 | 20 | 21 | 24 | 26 | 33 | 34 | 37 | 39 | 46 | 47 | 50 | 52 | 59 | 60 | 63 |
| $B_{13}$ | 0 | 7 | 9 | 12 | 13 | 14 | 15 | 31 | 35 | 38 | 43 | 45 | 49 | 55 | 60 | 63 | 65 |
| $B_{26}$ | 0 | 7 | 9 | 12 | 18 | 22 | 25 | 26 | 27 | 28 | 42 | 47 | 50 | 56 | 58 | 62 | 66 |
| $B_{39}$ | 0 | 7 | 9 | 12 | 17 | 21 | 23 | 29 | 32 | 37 | 41 | 44 | 48 | 52 | 53 | 64 | 67 |
| $B_{52}$ | 0 | 7 | 9 | 12 | 16 | 19 | 24 | 30 | 34 | 36 | 39 | 40 | 51 | 54 | 57 | 61 | 68 |
| | $D_3$ | | | | | | | | | | | | | | | | |
| $B_0$ | 0 | 13 | 20 | 21 | 24 | 26 | 33 | 34 | 37 | 39 | 46 | 47 | 50 | 52 | 59 | 60 | 63 |
| $B_{13}$ | 0 | 7 | 9 | 12 | 13 | 22 | 25 | 27 | 28 | 31 | 42 | 47 | 49 | 56 | 58 | 63 | 65 |
| $B_{26}$ | 0 | 7 | 9 | 12 | 14 | 15 | 18 | 26 | 35 | 38 | 43 | 45 | 50 | 55 | 60 | 62 | 66 |
| $B_{39}$ | 0 | 7 | 9 | 12 | 16 | 21 | 23 | 30 | 32 | 37 | 40 | 41 | 44 | 52 | 61 | 64 | 67 |
| $B_{52}$ | 0 | 7 | 9 | 12 | 17 | 19 | 24 | 29 | 34 | 36 | 39 | 48 | 51 | 53 | 54 | 57 | 68 |
| | $D_4$ | | | | | | | | | | | | | | | | |
| $B_0$ | 0 | 13 | 19 | 23 | 24 | 26 | 32 | 36 | 37 | 39 | 45 | 47 | 48 | 52 | 58 | 60 | 61 |
| $B_{13}$ | 0 | 7 | 9 | 12 | 19 | 22 | 23 | 27 | 29 | 34 | 40 | 49 | 50 | 52 | 54 | 59 | 65 |
| $B_{26}$ | 0 | 7 | 9 | 12 | 14 | 16 | 21 | 32 | 35 | 36 | 39 | 41 | 46 | 53 | 62 | 63 | 66 |
| $B_{39}$ | 0 | 7 | 8 | 11 | 15 | 16 | 25 | 26 | 32 | 37 | 44 | 49 | 51 | 55 | 56 | 59 | 67 |
| $B_{52}$ | 0 | 7 | 8 | 11 | 13 | 19 | 24 | 28 | 29 | 38 | 42 | 43 | 46 | 57 | 62 | 64 | 68 |

$D_3$ е еквивалентен на дизайна на Shrikhande/Singhi [212], а $D_4$ - на конструирания в [40]. Всеки от дизайните $D_3$ и $D_4$ е самодуален. Генераторите на групите от автоморфизми на четирите дизайна са представени в Таблица 2.2.

Остатъчните 2-(52,13,4) дизайни получаваме от симетричните 2-(69,17,4) дизайни чрез отрязване по всевъзможните начини на един блок и точките, които той съдържа. Отрязването на блокове от една и съща блокова орбита спрямо пълната група от автоморфизми води до изоморфни остатъчни дизайни. Дължините на точковите (блоковите) орбити на симетричните дизайни са дадени в Таблица 2.3.

Всички остатъчни дизайни, получени с отрязване на блокове от различни орбити, се оказват неизоморфни. Поради това четирите симетрични 2-(69,17,4) ди-

**Таблица 2.2: Автоморфизми**

| дизайн | генератори на пълната група от автоморфизми | $\|AutD\|$ |
|---|---|---|
| $D_1$, $D_2$ | $\alpha$,<br>(0)(1)...(12)(13,26)(14,27)...(25,38)(39,52)(40,53)...(51,64)(65,66)(67,68). | 26 = 13.2 |
| $D_3$ | $\alpha$,<br>(0)(1,3,9)(2,6,5)(4,12,10)(7,8,11)(13)(14,16,22)(15,19,18)(17,25,23)(20,21,24)(26)(27,29,35)<br>(28,32,31)(30,38,36)(33,34,37)(39)(40,42,48)(41,45,44)(43,51,49)(46,47,50)(52)(53,55,61)<br>(54,58,57)(56,64,62)(59,60,63)(65)(66)(67)(68),<br>(0)(1)...(12)(13,39,26,52)(14,40,27,53)...(25,51,38,64)(65,68,66,67). | 156 = 13.3.4 |
| $D_4$ | $\alpha$,<br>(0)(1,3,9)(2,6,5)(4,12,10)(7,8,11)(13,19,24)(14,22,20)(15,25,16)(17,18,21)(23)(26,32,37)<br>(27,35,33)(28,38,29)(30,31,34)(36)(39,47,45)(40,50,41)(42,43,46)(44,49,51)(48)(52,60,58)<br>(53,63,54)(55,56,59)(57,62,64)(61)(65)(66)(67)(68),<br>(0,2)(1)(3,12)(4,11)(5,10)(6,9)(7,8)(13,47,26,60)(14,46,27,59)...(21,39,34,52)<br>(22,51,35,64)(23,50,36,63)...(25,48,38,61)(65,68,66,67). | 156 = 13.3.4 |

зайна водят до 16 неизоморфни 2-(52,13,4) дизайна. За всеки от тях в Таблица 2.4 е представен начина на конструиране, т.е. от кой симетричен дизайн и с отрязване на кой блок. Дадени са и редовете на пълните им групи от автоморфизми.

**Таблица 2.3: Дължини на орбитите спрямо пълната група от автоморфизми**

| Дизайн | Дължини на точковите орбити | Дължини на блоковите орбити |
|---|---|---|
| $D_1$ | 13,26,26,2,2 | 13,26,26,2,2 |
| $D_2$ | 13,26,26,2,2 | 13,26,26,2,2 |
| $D_3$ | 13,52,4 | 13,52,4 |
| $D_4$ | 13,52,4 | 13,52,4 |

**Таблица 2.4: Остатъчни 2-(52,13,4) дизайни**

| Дизайн | $d_1$ | $d_2$ | $d_3$ | $d_4$ | $d_5$ | $d_6$ | $d_7$ | $d_8$ | $d_9$ | $d_{10}$ | $d_{11}$ | $d_{12}$ | $d_{13}$ | $d_{14}$ | $d_{15}$ | $d_{16}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Получен от | $D_1$ | $D_1$ | $D_1$ | $D_1$ | $D_1$ | $D_2$ | $D_2$ | $D_2$ | $D_2$ | $D_2$ | $D_3$ | $D_3$ | $D_3$ | $D_4$ | $D_4$ | $D_4$ |
| Отрязан блок | 0 | 13 | 39 | 65 | 67 | 0 | 13 | 39 | 65 | 67 | 0 | 13 | 65 | 0 | 13 | 65 |
| $\|Aut(d_i)\|$ | 2 | 1 | 1 | 13 | 13 | 2 | 1 | 1 | 13 | 13 | 12 | 3 | 39 | 12 | 3 | 39 |

Както симетричните, така и остатъчните дизайни, се отделят напълно от инвариантите $(I_1)$ от раздел 1.2.7.

Изследванията от настоящата част са публикувани в [T1].

## 2.4 Алгоритъм за установяване на Адамарова еквивалентност и за намиране на групата от автоморфизми на Адамарова $n \times n$ матрица

Такива алгоритми са разработвани от доста автори (например [72], [98], [118], [165], [186]). Използваните в настоящата глава представляват интерес преди всичко заради това, че са ясни, прости и лесно приложими.

### 2.4.1 Използване на разширение ($2n \times 2n$ матрица)

Нека $H$ и $\bar{H}$ са две Адамарови матрици от ред $n$, такива, че всеки елемент на едната от тях е равен на съответния елемент на другата, умножен с $-1$, а $\mathcal{H}$ и $\bar{\mathcal{H}}$ са матриците получени от $H$ и $\bar{H}$ със замяна на $-1$ с 0. Матрицата $H^* = \begin{pmatrix} \mathcal{H} & \bar{\mathcal{H}} \\ \bar{\mathcal{H}} & \mathcal{H} \end{pmatrix}$ ще наричаме разширение на Адамаровата матрица $H$. Редът на пълната група от автоморфизми на $H$ е равен на реда на пълната група от автоморфизми на матрицата $H^*$, а две Адамарови матрици $H_1$ и $H_2$ са еквивалентни тогава и само тогава, когато матриците $H_1^*$ и $H_2^*$ са еквивалентни [118], [186].

Използването на матрица с 4 пъти повече елементи, обаче, забавя изчисленията с алгоритмите от раздели 1.2.7 и 1.2.8, а в някои случаи ги прави невъзможни. Затова с тези алгоритми, заедно с инварвариантите ($I_1$) от раздел 1.2.7 използваме и инвариантите ($I_2$) на редовете (стълбовете) на $H^*$.

($I_2$): За всеки ред (стълб) $i$ на $H^*$ пресмятаме вектора $(m_0, m_1, \ldots, m_{n/4-1})$, където $m_s$ е броят на тройките от редове (стълбове) $j, k, l$, различни от $i$ и такива, че точно $s$ стълба (реда) съдържат 1-ци във всеки от редовете (стълбовете) $i, j, k, l$.

Когато и това не помага, отначало пресмятаме инварианти на целите Адамарови матрици. Те ги разбиват на класове и проверка за еквивалентност правим само между матриците от един и същи клас. Тук представяме използваните инварианти.

### 2.4.2 Инварианти на Адамарови матрици

($Im_1$): Ако подредим векторите от ($I_2$) в лексикографски ред, получаваме инварианта на Адамаровата матрица.

($Im_2$): Съвкупността от 2-дизайните, които се получават от една Адамарова матрица (след нормализиране по всевъзможните начини), са нейна инварианта.

($Im_3$): Съвкупността от 3-дизайните, които се получават от една Адамарова матрица, са нейна инварианта.

## 2.5 Адамарови матрици от ред 44

### 2.5.1 Въведение

Адамаровите матрици от ред 44 не са класифицирани до момента. Преди резултатите от настоящата глава бяха известни два Адамарови 2-(43,21,10) дизайна [182], конструирани с автоморфизми от ред 43 [105]. Един от тях се получава и от теоремата на Paley [198]. Тук представяме класификацията на всички нееквивалентни Адамарови матрици от ред 44, които имат автоморфизми от ред 7 (384 на брой) и съответните им 3-(44,22,10) и 2-(43,21,10) дизайни (броят им е съответно 1683 и 57932).

Първият известен екстремален двоичен двойно-четен самодуален код с дължина 88 [172] може да бъде намерен като използваме един от 2-дизайните с автоморфизъм от ред 43 (този, който следва и от теоремата на Paley [198] и има пълна група от автоморфизми от ред 903=3.7.43). Освен него преди настоящата работа беше конструиран и още един код с тези параметри [109], т.е. бяха известни два нееквивалентни екстремални двоични двойно-четни самодуални [88,44,16] кода.

Използваните тук методи за конструиране на самодуални кодове от симетрични дизайни са описани в [29] и [233]. Ние намираме всички двоични двойно-четни самодуални кодове с дължина 88, които се получават по този начин от 2-(43,22,11) дизайните. Два от тях са екстремални, единият от които беше нов към момента на публикуване на резултата. От конструираните дизайни намираме и много самодуални кодове с дължина 86, но сред тях няма екстремални.

Изследванията извършваме в следната последователност:

- Намираме всички (886 броя) 2-(43,21,10) дизайни с автоморфизми от ред 7;
- От всеки 2-(43,21,10) дизайн конструираме Адамарова матрица от ред 44;
- Всяка Адамарова матрица нормализираме по всевъзможните начини и така получаваме съответстващите на нея неизоморфни 3- и 2-дизайни;
- Определяме класовете на еквивалентност на Адамаровите матрици с автоморфизми от ред 7;
- Конструираме всички двоични двойно-четни самодуални кодове с дължини 86 и 88, които се получават от 2-(43,21,10) дизайните. Понеже от дизайни, които са разширими до една и съща Адамарова матрица, се получава един и същ код, то конструираме само кодовете, съответстващи на един от дизайните от всяка Адамарова матрица.

### 2.5.2 Свойства и начин на конструиране на 2-(43,21,10) дизайните

**Възможни автоморфизми на 2-(43,21,10) дизайни**

**Теорема 2.5.1.** *Всички възможни прости делители на реда на групата от автоморфизми на 2-(43,21,10) дизайн са* 2, 3, 5, 7 *и* 43.

**Доказателство.** Ако 2-($v$,$k$,$\lambda$) дизайн притежава автоморфизъм от прост ред $p$, то $p \leq k$ или $p|v$ (иначе всички точки и блокове ще бъдат фиксирани). Поради това $p = 2, 3, 5, 7, 11, 13, 17, 19, 43$ са единствените възможни прости редове на автоморфизми на такъв дизайн. Разглеждаме $p = 11, 13, 17, 19$.

Допускаме, че съществува автоморфизъм от ред $p$, фиксиращ $f > 1$ точки. В такъв случай фиксиран блок съдържа или $k$, или $k - p$ фиксирани точки. Да означим с $h_1$ и $h_2$ броя на фиксираните блокове, съдържащи съответно $k$ и $k - p$ фиксирани точки. Тогава са в сила:

$$\begin{aligned} \binom{21}{2}h_1 + \binom{21-p}{2}h_2 &= \binom{f}{2}\lambda \\ h_2 &\leq \frac{43-f}{p} \\ h_1 + h_2 = f &\geq 1 \\ h_1 + h_2 &\leq 43 - p \end{aligned}$$

тази система няма целочислено решение.

Случаите $f = 0, 1$ са възможни само, ако съответно $p|v$ или $v = 1 (\mathrm{mod}\ p)$. Поради това един 2-(43,21,10) дизайн не може да притежава автоморфизми от редове $11, 13, 17, 19$. □

Дизайните с най-богатите възможни групи от автоморфизми притежават много симетрия, затова са сравнително малко и обикновено могат да бъдат конструирани по-лесно. Преди описаното тук изследване 7 беше най-големият възможен все още неразглеждан прост ред на автоморфизъм на 2-(43,21,10) дизайн.

**Теорема 2.5.2.** *Автоморфизъм от ред* 7 *на 2-(43,21,10) дизайн фиксира една точка и един блок.*

**Доказателство.** Допускаме, че 2-(43,21,10) дизайн притежава автоморфизъм от ред 7 фиксиращ $f > 1$ точки и повече от един блок. Автоморфизъм на симетричен дизайн фиксира най-много половината точки [50], [87], [161]. Затова $f = 8$ или $f = 15$. Един фиксиран блок може да съдържа точките на $m$ нетривиални точкови орбити и поради това $21 - 7m$ фиксирани точки ($m = 1, 2, 3$). Ако фиксиран блок няма фиксирани точки, то общите му точки с друг фиксиран блок няма да са 10 (а трябва да са 10, защото дизайнът е симетричен). Затова $m = 1, 2$ и фиксиран блок има или 7, или 14 фиксирани точки. Означаваме с $h_7$ и $h_{14}$ фиксираните блокове, съдържащи 7 и, съответно 14 фиксирани точки.

Допускаме, че $f = 15$. В такъв случай има 4 нетривиални точкови орбити. Два фиксирани блока не могат да съдържат точките на две едни и същи нетривиални точкови орбити, поради това $h_7 \leq \binom{4}{2} = 6$. Но $h_{14} \leq 1$ (иначе два блока ще имат повече от 10 общи точки), и следователно $f \leq 7$ - противоречие.

Допускаме, че $f = 8$. Има 5 нетривиални точкови орбити и, поради това, най-малко два блока, съдържащи точки от една и съща нетривиална точкова орбита. Тези блокове трябва да имат точно 3 общи фиксирани точки, което е невъзможно. □

**Построяване на 2-(43,21,10) дизайни с автоморфизми от ред 7**

Дизайните конструираме с използване на алгоритъма от раздел 2.2. Нека $D$ е 2-(43,21,10) дизайн с автоморфизъм $\alpha$ от ред 7 с една фиксирана точка и един фиксиран блок. Без загуба на общност приемаме, че $\alpha$ размества точките (блоковете) по следния начин:

$\alpha$ = (1,2,3,4,5,6,7)(8,9,10,11,12,13,14) $\cdots$ (36,37,38,39,40,41,42)(43).

Автоморфизмът $\alpha$ поражда циркулантна структура на матрицата на инцидентност $A$ на дизайна $D$:

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} & a_{1,6} & u^t \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} & a_{2,6} & u^t \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} & a_{3,6} & u^t \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} & a_{4,5} & a_{4,6} & z^t \\ a_{5,1} & a_{5,2} & a_{5,3} & a_{5,4} & a_{5,5} & a_{5,6} & z^t \\ a_{6,1} & a_{6,2} & a_{6,3} & a_{6,4} & a_{6,5} & a_{6,6} & z^t \\ u & u & u & z & z & z & 0 \end{pmatrix}$$

където $a_{i,j}, \quad i,j = 1,2,\ldots 6$ са циркуланти от ред 7, $u = (1,1,1,1,1,1,1)$, $z = (0,0,0,0,0,0,0)$.

Нека $m_{i,j}, \quad i,j = 1,2,\ldots 6$ е броя на 1-ците в ред на $a_{i,j}$. За матрицата $M = (m_{i,j})_{6\times 6}$ са в сила следните равенства:

$$\sum_{j=1}^{6} m_{i,j} = 20, \qquad \sum_{j=1}^{6} m_{i,j}^2 = 74, \quad i = 1,2,3$$

$$\sum_{j=1}^{6} m_{i,j} = 21, \qquad \sum_{j=1}^{6} m_{i,j}^2 = 81, \quad i = 4,5,6$$

$$\sum_{j=1}^{6} m_{i_1,j} m_{i_2,j} = 63, \qquad 1 \leq i_1 < i_2 \leq 3$$

$$\sum_{j=1}^{6} m_{i_1,j} m_{i_2,j} = 70, \qquad 1 \leq i_1 < i_2, i_2 > 3.$$

Първите четири равенства са получени с пресмятане по два начина на броя на точките и двойките точки в $i$-я циркулантен ред на $A$, а последните две с прес-

мятане на броя на двойките точки, такива, че едната е от $i_1$-я, а другата от $i_2$-я циркулантен ред на $A$.

Има две неизоморфни матрици $M_1$ и $M_2$, за които горните равенства са в сила.

$M_1$

| | | | | | |
|---|---|---|---|---|---|
| 5 | 2 | 3 | 4 | 4 | 2 |
| 3 | 5 | 2 | 2 | 4 | 4 |
| 2 | 3 | 5 | 4 | 2 | 4 |
| 4 | 4 | 2 | 5 | 2 | 4 |
| 2 | 4 | 4 | 4 | 5 | 2 |
| 4 | 2 | 4 | 2 | 4 | 5 |

$M_2$

| | | | | | |
|---|---|---|---|---|---|
| 4 | 4 | 2 | 5 | 3 | 2 |
| 2 | 4 | 4 | 2 | 5 | 3 |
| 4 | 2 | 4 | 3 | 2 | 5 |
| 5 | 2 | 3 | 3 | 5 | 3 |
| 3 | 5 | 2 | 3 | 3 | 5 |
| 2 | 3 | 5 | 5 | 3 | 3 |

Да разгледаме матрицата $A'$.

$$A' = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} & a_{1,6} & u^t \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} & a_{2,6} & u^t \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} & a_{3,6} & u^t \\ \bar{a}_{4,1} & \bar{a}_{4,2} & \bar{a}_{4,3} & \bar{a}_{4,4} & \bar{a}_{4,5} & \bar{a}_{4,6} & z^t \\ \bar{a}_{5,1} & \bar{a}_{5,2} & \bar{a}_{5,3} & \bar{a}_{5,4} & \bar{a}_{5,5} & \bar{a}_{5,6} & z^t \\ \bar{a}_{6,1} & \bar{a}_{6,2} & \bar{a}_{6,3} & \bar{a}_{6,4} & \bar{a}_{6,5} & \bar{a}_{6,6} & z^t \\ z & z & z & u & u & u & 0 \end{pmatrix} \tag{2.3}$$

където $\bar{a}_{i,j}$ са получени от $a_{i,j}$ чрез заместване на 0 с 1 и 1 с 0. Лесно се проверява, че $A'$ също е матрица на инцидентност на 2-(43,21,10) дизайн (да го означим с $D'$) с автоморфизми от ред 7 и, че ако $A$ съответства на $M_1$, то $A'$ съответства на $M_2$ и обратното. Двата дизайна $D$ и $D'$ се разширяват до еквивалентни Адамарови матрици. Затова всички 2-(43,21,10) дизайни с автоморфизми от ред 7 можем да получим като разширим само едната от матриците $M_1$ и $M_2$.

### 2.5.3 Класификация на Адамаровите дизайни

Замяната на елементите на $M_1$ с циркуланти от ред 7 води до 443 неизоморфни 2-(43, 21,10) дизайна. След това от тях получаваме 443-те дизайна, съответстващи на $M_2$. С алгоритъма от раздел 1.2.8 определяме групите от автоморфизми на всичките 886 неизоморфни дизайни. Два от тях са изоморфни на дизайните, намерени в [105] - с пълни групи от автоморфизми от редове 903 и 301 съответно. Има 81 дизайна с пълна група от автоморфизми от ред 21 – пет самодуални и 38 двойки дуални дизайни. Останалите 804 дизайна са с пълна група от автоморфизми от ред 7.

### 2.5.4 Класификация на Адамаровите матрици и дизайните, които пораждат

Класовете на еквивалентност на конструираните Адамарови матрици са 384 на брой и ги определяме като използваме техните разширения, а след това пресмятаме и инвариантите $Im_1$, $Im_2$ и $Im_3$, които са допълнителна проверка за нееквивалентността им (виж раздел 2.4).

От построените Адамарови матрици се получават 57932 неизоморфни 2-(43,21,10) и 1683 неизоморфни 3-(44,22,10) дизайна. Всички намерени по този начин 2-дизайни, които нямат автоморфизми от ред 7, нямат никакви нетривиални автоморфизми.

Реда на пълната им група от автоморфизми също намираме като ред на групата от автоморфизми на разширенията. Групите от автоморфизми са представени в Таблица 2.5, където е даден и броят на породените от тях 2- и 3-дизайни. В последната колона са работите, в които някои от тези дизайни са конструирани преди настоящото изследване.

**Таблица 2.5: Ред на пълната група от автоморфизми на Адамаровите матрици**

| $\|Aut(H)\|$ | | | Брой на матриците | 2-дизайни | 3-дизайни | коментар |
|---|---|---|---|---|---|---|
| 79464 | = | 2.44.43.21 | 1 | 2 | 1 | [198] |
| 602 | = | 2.43.7 | 1 | 10 | 2 | [105] |
| 84 | = | 4.21 | 40 | 48 | 2 | |
| 28 | = | 4.7 | 284 | 140 | 4 | |
| 14 | = | 2.7 | 58 | 280 | 8 | |

Само една матрица има транзитивна група от автоморфизми. Тя е от ред 79464. Адамаровата матрица, получена от втория дизайн, намерен в [105] е с група от ред 602. Четиридесет Адамарови матрици от ред 44 (две самодуални и 19 двойки дуални матрици) са с ред на пълната група от автоморфизми 84.

### 2.5.5 Двоични екстремални самодуални кодове с дължини 86 и 88

Нека $A_{21}$ е матрицата на инцидентност на 2-(43,21,10) дизайн, $A_{22}$ матрицата на инцидентност на съответния му 2-(43,22,11) дизайн и $A_{22}^{+} = \begin{pmatrix} A_{22} & U^t \\ U & 0 \end{pmatrix}$, където U е вектор с размерност 43, всички елементи на който са единици.

Тогава $(A_{21} \;\; I_{43})$ е генерираща матрица на самодуален код с дължина 86 [233]. Самодуалните кодове, които получаваме по този начин са с минимално тегло най-много 14, т.е. сред тях няма екстремални.

Генерираща матрица на двойно-четен самодуален код с дължина 88 може да бъде получена [233] като $(A_{22}^{+} \;\; I_{44})$. Така конструираме два нееквивалентни [88,44,16] кода. Първият се генерира от матрицата на Paley [198] и е еквивалентен на единия от известните до тогава кодове. Другият е нов към момента на изследването и се получава от дизайн със следните базови блокове:

$B_1$ = 2 3 9 10 12 16 17 18 20 21 23 25 27 30 31 32 34 35 37 38 39 43
$B_8$ = 2 3 4 5 7 8 11 16 20 21 24 27 28 31 32 33 36 37 39 41 42 43
$B_{15}$ = 2 3 4 5 6 8 10 12 15 18 19 22 24 27 28 29 30 31 34 35 37 42
$B_{22}$ = 2 3 6 8 13 14 15 17 19 20 21 22 23 25 27 28 29 33 37 38 39 41
$B_{29}$ = 2 3 5 8 11 12 13 14 15 17 22 23 24 25 26 30 32 35 36 39 42 43
$B_{36}$ = 2 4 6 9 10 11 13 14 19 20 21 24 25 30 33 34 35 36 37 39 40 42

Адамаровата матрица, която поражда кода, е с група от автоморфизми от ред 14. За да покажем, че този код не е еквивалентен на някой от двата известни тогава екстремални двоични самодуални кода с дължина 88 [79], пресмятаме следните инварианти на трите кода: Във всеки от тях има 32164 думи с тегло 16. Разглеждаме ги като редове на (0,1) матрица и за всеки нейн стълб намираме вектора с елементи скаларните му произведения с останалите 87 колони. Подреждаме в лексикографски ред елементите на всеки вектор и самите вектори. Получената инварианта е различна за трите кода.

Изследванията от тази част са публикувани в [T2].

## 2.6 Адамарови матрици от ред 64

### 2.6.1 Въведение

*Афинен* (или *афинно-разрешим*) дизайн е разрешим дизайн, такъв, че всеки два блока от различни паралелни класове имат еднакъв брой общи точки.

*Права* в един дизайн през точките $x$ и $y$ наричаме сечението на всички блокове, съдържащи $x$ и $y$. Максималният размер на права в Адамаров 2-дизайн е 3. Под *спред* от прави разбираме разбиване на множеството от точките на дизайна на непресичащи се прави.

Афинни 2-(64,16,5) дизайни могат да бъдат получени от спредове от прави на Адамарови 2-(63,31,15) дизайни чрез конструкция на Rahilly [205].

Долни граници за броя на Адамаровите дизайни и матрици, съдържащи Адамарови дизайни от по-малък ред, са изведени в [157] и [158]. Според тях има повече от 26! неизоморфни 2-(63,31,15) дизайна. Голям брой Адамарови матрици от редове, които се делят на 8, са получени чрез удвояваща конструкция в [100].

Hamada формулира хипотеза [107], според която дизайните, получени от точките и подпространствата с дадена размерност в $AG(n, p^m)$ ($p$ е просто) имат минимален $p$-ранг, а всички други дизайни със същите параметри имат по-голям $p$-ранг. Контрапримери на тази хипотеза са пет 3-(32,8,7) дизайна [230] и три 2-(64,16,5) дизайна [110], [185]. В този смисъл по-нататъшни изследвания на негеометрични 2-(64,16,5) дизайни с минимален ранг и съответстващите им посредством конструкцията на Rahilly [205] Адамарови дизайни са от особен интерес [235]. Трите Адамарови 2-(63,31,15) дизайна, които съответстват на трите известни 2-(64,16,5) дизайна с ранг 16, са инвариантни относно диедралната група от ред 10. Това ни мотивира да класифицираме всички Адамарови 2-(63,31,15) дизайни, инвариантни относно диедралната група от ред 10.

В [155] са конструирани 394 Адамарови матрици от ред 64 с две циркулантни ядра. Всички те имат 2-ранг по-голям от 16. В [76] са конструирани 38 на брой 2-(63,31,15) дизайна с група от автоморфизми от ред 155. Сред тях има 11 дизайна с 2-ранг не по-голям от 16, т.е. дизайнът от точките и хиперравнините в $PG(5, 2)$

(с ранг 16) и 10 дизайна с ранг 12, които не водят до 2-(64,16,5) дизайни с минимален ранг. Не са ни известни други конструкции на 2-(63,31,15) със зададени автоморфизми.

Ние построяваме 8330 неизоморфни 2-(63,31,15) дизайна, инвариантни относно диедралната група от ред 10 и съответните 1691 нееквивалентни Адамарови матрици от ред 64. За 2-(63,31,15) дизайните с 2-ранг не по-голям от 16, построяваме всички спредове от прави и съответните им 2-(64,16,5) дизайни с 2-ранг не по-голям от 16. Така получаваме точно трите 2-(64,16,5) дизайна с 2-ранг 16, които са известни до момента [110], [185].

### 2.6.2 Особености на конструктивния автоморфизъм и алгоритъма

**Теорема 2.6.1.** *Всички възможни прости делители на реда на групата от автоморфизми на 2-(63,31,15) дизайн са* 2, 3, 5, 7 *и* 31.

**Доказателство.** Ако 2-($v$,$k$,$\lambda$) дизайн притежава автоморфизъм от прост ред $p$, тогава $p \leq k$ или $p|v$ (иначе всички точки и блокове ще са фиксирани). Поради това $p = 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31$ са възможните прости редове на автоморфизми на 2-(63,31,15) дизайн.

а) Разглеждаме $p = 11, 13$. Допускаме, че съществува автоморфизъм от ред $p$, фиксиращ $f \geq 63 \; mod \; p$ точки. В този случай фиксиран блок може да съдържа точки от 0,1 или 2 нефиксирани точкови орбити и, съответно $k$, $k - p$ или $k - 2p$ фиксирани точки, т.е. $f \geq k - 2p$. Следва, че $f \geq 19$ за $p = 11$ и $f \geq 11$ за $p = 13$. Всеки два блока имат $\lambda = 15$ общи точки, следователно точките на две нетривиални точкови орбити могат да са в най-много един фиксиран блок. В такъв случай има фиксирани блокове с $k = 31$ фиксирани точки, т.е $f \geq 31$. Но за $f \geq 31$ има повече от 16 фиксирани блока с $k = 31$ фиксирани точки, което е невъзможно, защото за всяка двойка точки на дизайна има 16 блока, които не са инцидентни с никоя от двете точки.

b) Разглеждаме $p = 17, 19, 23, 29$. Допускаме, че съществува автоморфизъм от ред $p$, фиксиращ $f \geq 63 \; mod \; p$ точки. В такъв случай фиксиран блок може да съдържа точките на 0 или 1 нефиксирана точкова орбита и, съответно, $k$ или $k-p$ фиксирани точки. Но $\lambda = 15$ и поради това точките на всяка нетривиална точкова орбита трябва да се съдържат в най-много един фиксиран блок. По подобен на а) начин следва, че има повече от 16 фиксирани блока с $k = 31$ фиксирани точки, което е невъзможно. □

**Теорема 2.6.2.** *Автоморфизъм от ред* 5 *на 2-(63,31,15) дизайн фиксира* 3 *точки и* 3 *блока.*

**Доказателство.** Допускаме, че 2-(63,31,15) дизайн притежава автоморфизъм от ред 5, фиксиращ повече от 3 точки и блока. Автоморфизъм на симетричен дизайн може да фиксира не повече от половината точки [50], [87], [161]. Затова

$f = 8, 13, 18, 23, 28$. Нека $w = (63 - f)/5$ е броя на нетривиалните точкови/блокови орбити. Съкратената матрица на инцидентност на $D$ е:

$$\begin{pmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,w} \\ A_{2,1} & A_{2,2} & \dots & A_{2,w} \\ \dots & \dots & \dots & \dots \\ A_{w,1} & A_{w,2} & \dots & A_{w,w} \end{pmatrix}$$

където $A_{i,j}, \quad i, j = 1, 2, ..., w$ са циркуланти от ред 5. Нека $n_{i,j}, \ i, j = 1, 2, \dots w, \ $ е броят на 1-ците в ред на $A_{i,j}$, $s_i$ - броят на фиксираните блокове, които са инцидентни с точките на $i$-тата нетривиална точкова орбита, а $s_{i_1,i_2}$ - броят на фиксираните блокове, съдържащи двойката точкови орбити $(i_1, i_2)$. Следват равенства, които са в сила за матрицата $N = (n_{i,j})_{w \times w}$

$$\sum_{j=1}^{w} n_{i,j} = 31 - s_i, \qquad i = 1, 2, ..., w \tag{2.4}$$

$$\sum_{j=1}^{w} n_{i,j}^2 = 91 - 5s_i, \qquad i = 1, 2, ..., w \tag{2.5}$$

$$\sum_{i=1}^{w} n_{i_1,j} n_{i_2,j} = 75 - 5s_{i_1,i_2}, \qquad i_1, i_2 = 1, 2, ..., w, i_1 \neq i_2 \tag{2.6}$$

Системата няма целочислени решения. □

Без загуба на общност избираме дизайните да притежават автоморфизъм $\alpha$ от ред 5, който действа на точките и блоковете с пермутацията:

$(1, 2, 3, 4, 5)(6, 7, 8, 9, 10)...(56, 57, 58, 59, 60)(61)(62)(63)$.

Дизайнът има матрица на инцидентност, която можем да представим чрез четири подматрици:

$$\begin{pmatrix} A & H \\ F & C \end{pmatrix},$$

където $A = (a_{ij})_{12 \times 12}$ е съкратената матрица на инцидентност. Нейните елементи са циркуланти $a_{ij}$ от ред 5. Матриците $H$, $F$, и $C$ съответстват на фиксираните от $\alpha$ елементи. Пример за такъв дизайн е представен на фигура 2.1.

Без загуба на общност задаваме

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

Всеки елемент $f_{ij}$ от $F$ е $p = (1, 1, 1, 1, 1)$ ако $i$-тата фиксирана точка е в блоковете на $j$-тата нефиксирана блокова орбита и $o = (0, 0, 0, 0, 0)$ в противен случай. За $F$ има 4 възможности:

Фигура 2.1: Циркулантна структура на 2-(63,31,15) дизайн с автоморфизми от ред 5

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10000 | 11001 | 10110 | 10000 | 11001 | 10110 | 10000 | 10110 | 11001 | 10000 | 10110 | 11001 | 111 |
| 01000 | 11100 | 01011 | 01000 | 11100 | 01011 | 01000 | 01011 | 11100 | 01000 | 01011 | 11100 | 111 |
| 00100 | 01110 | 10101 | 00100 | 01110 | 10101 | 00100 | 10101 | 01110 | 00100 | 10101 | 01110 | 111 |
| 00010 | 00111 | 11010 | 00010 | 00111 | 11010 | 00010 | 11010 | 00111 | 00010 | 11010 | 00111 | 111 |
| 0 0001 | 10011 | 01101 | 00001 | 10011 | 01101 | 00001 | 01101 | 10011 | 00001 | 01101 | 10011 | 111 |
| 10110 | 10000 | 11001 | 10110 | 10000 | 11001 | 11001 | 10000 | 10110 | 11001 | 10000 | 10110 | 111 |
| 01011 | 01000 | 11100 | 01011 | 01000 | 11100 | 11100 | 01000 | 01011 | 11100 | 01000 | 01011 | 111 |
| 10101 | 00100 | 01110 | 10101 | 00100 | 01110 | 01110 | 00100 | 10101 | 01110 | 00100 | 10101 | 111 |
| 11010 | 00010 | 00111 | 11010 | 00010 | 00111 | 00111 | 00010 | 11010 | 00111 | 00010 | 11010 | 111 |
| 0 1101 | 00001 | 10011 | 01101 | 00001 | 10011 | 10011 | 00001 | 01101 | 10011 | 00001 | 01101 | 111 |
| 11001 | 10110 | 10000 | 11001 | 10110 | 10000 | 10110 | 11001 | 10000 | 10110 | 11001 | 10000 | 111 |
| 11100 | 01011 | 01000 | 11100 | 01011 | 01000 | 01011 | 11100 | 01000 | 01011 | 11100 | 01000 | 111 |
| 01110 | 10101 | 00100 | 01110 | 10101 | 00100 | 10101 | 01110 | 00100 | 10101 | 01110 | 00100 | 111 |
| 00111 | 11010 | 00010 | 00111 | 11010 | 00010 | 11010 | 00111 | 00010 | 11010 | 00111 | 00010 | 111 |
| 1 0011 | 01101 | 00001 | 10011 | 01101 | 00001 | 01101 | 10011 | 00001 | 01101 | 10011 | 00001 | 111 |
| 10000 | 11001 | 10110 | 01111 | 00110 | 01001 | 01111 | 00110 | 01001 | 10000 | 11001 | 10110 | 100 |
| 01000 | 11100 | 01011 | 10111 | 00011 | 10100 | 10111 | 00011 | 10100 | 01000 | 11100 | 01011 | 100 |
| 00100 | 01110 | 10101 | 11011 | 10001 | 01010 | 11011 | 10001 | 01010 | 00100 | 01110 | 10101 | 100 |
| 00010 | 00111 | 11010 | 11101 | 11000 | 00101 | 11101 | 11000 | 00101 | 00010 | 00111 | 11010 | 100 |
| 0 0001 | 10011 | 01101 | 11110 | 01100 | 10010 | 11110 | 01100 | 10010 | 00001 | 10011 | 01101 | 100 |
| 10110 | 10000 | 11001 | 01001 | 01111 | 00110 | 01001 | 01111 | 00110 | 10110 | 10000 | 11001 | 100 |
| 01011 | 01000 | 11100 | 10100 | 10111 | 00011 | 10100 | 10111 | 00011 | 01011 | 01000 | 11100 | 100 |
| 10101 | 00100 | 01110 | 01010 | 11011 | 10001 | 01010 | 11011 | 10001 | 10101 | 00100 | 01110 | 100 |
| 11010 | 00010 | 00111 | 00101 | 11101 | 11000 | 00101 | 11101 | 11000 | 11010 | 00010 | 00111 | 100 |
| 0 1101 | 00001 | 10011 | 10010 | 11110 | 01100 | 10010 | 11110 | 01100 | 01101 | 00001 | 10011 | 100 |
| 11001 | 10110 | 10000 | 00110 | 01001 | 01111 | 00110 | 01001 | 01111 | 11001 | 10110 | 10000 | 100 |
| 11100 | 01011 | 01000 | 00011 | 10100 | 10111 | 00011 | 10100 | 10111 | 11100 | 01011 | 01000 | 100 |
| 01110 | 10101 | 00100 | 10001 | 01010 | 11011 | 10001 | 01010 | 11011 | 01110 | 10101 | 00100 | 100 |
| 00111 | 11010 | 00010 | 11000 | 00101 | 11101 | 11000 | 00101 | 11101 | 00111 | 11010 | 00010 | 100 |
| 1 0011 | 01101 | 00001 | 01100 | 10010 | 11110 | 01100 | 10010 | 11110 | 10011 | 01101 | 00001 | 100 |
| 01111 | 00110 | 01001 | 10000 | 11001 | 10110 | 01111 | 00110 | 01001 | 10000 | 11001 | 10110 | 010 |
| 10111 | 00011 | 10100 | 01000 | 11100 | 01011 | 10111 | 00011 | 10100 | 01000 | 11100 | 01011 | 010 |
| 11011 | 10001 | 01010 | 00100 | 01110 | 10101 | 11011 | 10001 | 01010 | 00100 | 01110 | 10101 | 010 |
| 11101 | 11000 | 00101 | 00010 | 00111 | 11010 | 11101 | 11000 | 00101 | 00010 | 00111 | 11010 | 010 |
| 1 1110 | 01100 | 10010 | 00001 | 10011 | 01101 | 11110 | 01100 | 10010 | 00001 | 10011 | 01101 | 010 |
| 01001 | 01111 | 00110 | 10110 | 10000 | 11001 | 01001 | 01111 | 00110 | 10110 | 10000 | 11001 | 010 |
| 10100 | 10111 | 00011 | 01011 | 01000 | 11100 | 10100 | 10111 | 00011 | 01011 | 01000 | 11100 | 010 |
| 01010 | 11011 | 10001 | 10101 | 00100 | 01110 | 01010 | 11011 | 10001 | 10101 | 00100 | 01110 | 010 |
| 00101 | 11101 | 11000 | 11010 | 00010 | 00111 | 00101 | 11101 | 11000 | 11010 | 00010 | 00111 | 010 |
| 1 0010 | 11110 | 01100 | 01101 | 00001 | 10011 | 10010 | 11110 | 01100 | 01101 | 00001 | 10011 | 010 |
| 00110 | 01001 | 01111 | 11001 | 10110 | 10000 | 00110 | 01001 | 01111 | 11001 | 10110 | 10000 | 010 |
| 00011 | 10100 | 10111 | 11100 | 01011 | 01000 | 00011 | 10100 | 10111 | 11100 | 01011 | 01000 | 010 |
| 10001 | 01010 | 11011 | 01110 | 10101 | 00100 | 10001 | 01010 | 11011 | 01110 | 10101 | 00100 | 010 |
| 11000 | 00101 | 11101 | 00111 | 11010 | 00010 | 11000 | 00101 | 11101 | 00111 | 11010 | 00010 | 010 |
| 0 1100 | 10010 | 11110 | 10011 | 01101 | 00001 | 01100 | 10010 | 11110 | 10011 | 01101 | 00001 | 010 |
| 01111 | 00110 | 01001 | 01111 | 00110 | 01001 | 10000 | 10110 | 11001 | 10000 | 10110 | 11001 | 001 |
| 10111 | 00011 | 10100 | 10111 | 00011 | 10100 | 01000 | 01011 | 11100 | 01000 | 01011 | 11100 | 001 |
| 11011 | 10001 | 01010 | 11011 | 10001 | 01010 | 00100 | 10101 | 01110 | 00100 | 10101 | 01110 | 001 |
| 11101 | 11000 | 00101 | 11101 | 11000 | 00101 | 00010 | 11010 | 00111 | 00010 | 11010 | 00111 | 001 |
| 1 1110 | 01100 | 10010 | 11110 | 01100 | 10010 | 00001 | 01101 | 10011 | 00001 | 01101 | 10011 | 001 |
| 01001 | 01111 | 00110 | 01001 | 01111 | 00110 | 11001 | 10000 | 10110 | 11001 | 10000 | 10110 | 001 |
| 10100 | 10111 | 00011 | 10100 | 10111 | 00011 | 11100 | 01000 | 01011 | 11100 | 01000 | 01011 | 001 |
| 01010 | 11011 | 10001 | 01010 | 11011 | 10001 | 01110 | 00100 | 10101 | 01110 | 00100 | 10101 | 001 |
| 00101 | 11101 | 11000 | 00101 | 11101 | 11000 | 00111 | 00010 | 11010 | 00111 | 00010 | 11010 | 001 |
| 1 0010 | 11110 | 01100 | 10010 | 11110 | 01100 | 10011 | 00001 | 01101 | 10011 | 00001 | 01101 | 001 |
| 00110 | 01001 | 01111 | 00110 | 01001 | 01111 | 10110 | 11001 | 10000 | 10110 | 11001 | 10000 | 001 |
| 00011 | 10100 | 10111 | 00011 | 10100 | 10111 | 01011 | 11100 | 01000 | 01011 | 11100 | 01000 | 001 |
| 10001 | 01010 | 11011 | 10001 | 01010 | 11011 | 10101 | 01110 | 00100 | 10101 | 01110 | 00100 | 001 |
| 11000 | 00101 | 11101 | 11000 | 00101 | 11101 | 11010 | 00111 | 00010 | 11010 | 00111 | 00010 | 001 |
| 0 1100 | 10010 | 11110 | 01100 | 10010 | 11110 | 01101 | 10011 | 00001 | 01101 | 10011 | 00001 | 001 |
| 11111 | 11111 | 11111 | 00000 | 00000 | 00000 | 00000 | 00000 | 00000 | 11111 | 11111 | 11111 | 100 |
| 00000 | 00000 | 00000 | 11111 | 11111 | 11111 | 00000 | 00000 | 00000 | 11111 | 11111 | 11111 | 010 |
| 0 0000 | 00000 | 00000 | 00000 | 00000 | 00000 | 11111 | 11111 | 11111 | 11111 | 11111 | 11111 | 001 |

$$a = \begin{pmatrix} p & p & p & p & p & p & o & o & o & o & o & o \\ p & p & p & o & o & o & p & p & p & o & o & o \\ p & p & p & o & o & o & o & o & o & p & p & p \end{pmatrix} b = \begin{pmatrix} p & p & p & p & p & p & o & o & o & o & o & o \\ p & p & p & o & o & o & p & p & p & o & o & o \\ o & o & o & p & p & p & p & p & p & o & o & o \end{pmatrix}$$

$$c = \begin{pmatrix} p & p & p & p & p & p & o & o & o & o & o & o \\ p & p & p & o & o & o & p & p & p & o & o & o \\ p & o & o & p & p & o & p & p & o & p & o & o \end{pmatrix} d = \begin{pmatrix} p & p & p & p & p & p & o & o & o & o & o & o \\ p & p & p & o & o & o & p & p & p & o & o & o \\ p & p & o & p & o & o & p & o & o & p & p & o \end{pmatrix}$$

Четирите възможности за $H$ се задават с транспонираните на горните матрици. Има 16 начина за комбиниране на матриците $H$ и $F$, но дизайнът е симетричен и затова остават само 10 от тях: $(a, a^T)$, $(a, b^T)$, $(a, c^T)$, $(a, d^T)$, $(b, b^T)$, $(b, c^T)$,$(b, d^T)$, $(c, c^T)$, $(c, d^T)$, $(d, d^T)$. При различните възможности за фиксираната част условията, на които трябва да отговарят циркулантите от $A$ са различни. Нека $m_{i,j}$, $i, j = 1, 2, \ldots 12$, е броя на 1-ците в ред на $a_{i,j}$, $s_i$ - броя на фиксираните блокове, които са инцидентни с точките на $i$-тата нетривиална точкова орбита, а $s_{i_1,i_2}$ - броя на фиксираните блокове, съдържащи двойката точкови орбити $(i_1, i_2)$. Долните равенства са в сила за съкратената орбитна матрица $M = (m_{i,j})_{12\times 12}$.

$$\sum_{j=1}^{12} m_{i,j} = 31 - s_i, \qquad i = 1, 2, ..., 12 \tag{2.7}$$

$$\sum_{j=1}^{12} m_{i,j}^2 = 91 - 5s_i, \qquad i = 1, 2, ..., 12 \tag{2.8}$$

$$\sum_{i=1}^{12} m_{i_1,j} m_{i_2,j} = 75 - 5s_{i_1,i_2}, \qquad i_1, i_2 = 1, 2, ..., 12, i_1 \neq i_2 \tag{2.9}$$

Първо построяваме всички нееквивалентни решения за $M$ при всичките десет случая за фиксираната част. Понеже искаме дизайните да имат не само автоморфизъм от ред 5, а и от ред 2, се оказва, че конструирането на нееквивалентни решения за $M$ е много по-бавно от разширяването им до 2-(63,31,15) дизайни. В три от случаите няма орбитни матрици - $(b, b^T), (b, c^T), (b, d^T)$. В случаи $(c, d^T)$ и $(d, d^T)$ не отсяваме всички еквивалентни матрици, защото са твърде много, а голямата част от тях не водят до дизайни и е по-бързо да ги отхвърляме по тази причина. Броят на построените нееквивалентни съкратени орбитни матрици за различните случаи е: $(a, a^T)$ - 2616, $(a, b^T)$ - 4801, $(a, c^T)$ - 69581, $(a, d^T)$ - 69581, $(b, b^T)$ - 0, $(b, c^T)$ - 0, $(b, d^T)$ - 0, $(c, c^T)$ - 87216, $(c, d^T)$ - най-малко 90000 и $(d, d^T)$ - най-малко 90000.

Диедралната група от ред 10 ($D_{10}$) е групата от симетриите на правилен петоъгълник. Тя включва 4 пермутации от ред 5: $\alpha_1 = (1, 2, 3, 4, 5)$, $\alpha_2 = (1, 3, 5, 2, 4)$, $\alpha_3 = (1, 4, 2, 5, 3)$, $\alpha_4 = (1, 5, 4, 3, 2)$, 5 пермутации от ред 2: $\beta_1 = (1)(2, 5)(3, 4)$, $\beta_2 = (1, 3)(2)(4, 5)$, $\beta_3 = (1, 5)(2, 4)(3)$, $\beta_4 = (1, 2)(3, 5)(4)$ и $\beta_5 = (1, 4)(2, 3)(5)$ и

идентитета (1)(2)(3)(4)(5). Групата се генерира от всяка двойка от една пермутация от ред 5 и една от ред 2, например от $\alpha_1$ и $\beta_3$. Има 31 циркуланти от ред 5, но спрямо пермутация $\beta_n$ $(n = 1, \ldots 5)$ както на редовете, така и на стълбовете, са инвариантни само 7 циркуланти, които са симетрични относно главния диагонал. За всяка друга циркуланта от ред 5 съществуват пермутации на нейните редове и стълбове, които я изобразяват в някоя от седемте симетрични циркуланти. Симетричните циркуланти от ред 5 са:

$$\begin{array}{ccccc}
1&0&0&0&0\\
0&1&0&0&0\\
0&0&1&0&0\\
0&0&0&1&0\\
0&0&0&0&1
\end{array}
\quad
\begin{array}{ccccc}
0&0&1&1&0\\
0&0&0&1&1\\
1&0&0&0&1\\
1&1&0&0&0\\
0&1&1&0&0
\end{array}
\quad
\begin{array}{ccccc}
0&1&0&0&1\\
1&0&1&0&0\\
0&1&0&1&0\\
0&0&1&0&1\\
1&0&0&1&0
\end{array}
\quad
\begin{array}{ccccc}
1&0&1&1&0\\
0&1&0&1&1\\
1&0&1&0&1\\
1&1&0&1&0\\
0&1&1&0&1
\end{array}
\quad
\begin{array}{ccccc}
1&1&0&0&1\\
1&1&1&0&0\\
0&1&1&1&0\\
0&0&1&1&1\\
1&0&0&1&1
\end{array}
\quad
\begin{array}{ccccc}
0&1&1&1&1\\
1&0&1&1&1\\
1&1&0&1&1\\
1&1&1&0&1\\
1&1&1&1&0
\end{array}
\quad
\begin{array}{ccccc}
1&1&1&1&1\\
1&1&1&1&1\\
1&1&1&1&1\\
1&1&1&1&1\\
1&1&1&1&1
\end{array}$$

Ако приложим $\beta_n$ $(n = 1, \ldots 5)$ едновременно на редовете и стълбовете на някоя от тези симетрични циркуланти, то тя остава непроменена. Ако приложим $\beta_n$ на редовете на несиметрична циркуланта, то съществува $\beta_m \in D_{10}$, което можем да приложим на стълбовете, за да остане тя непроменена. Например, да разгледаме циркулантата

$$\begin{array}{ccccc}
0&1&0&1&0\\
0&0&1&0&1\\
1&0&0&1&0\\
0&1&0&0&1\\
1&0&1&0&0
\end{array}$$

Тя се запазва от следните 5 двойки пермутации на редове$(r)$ и стълбове$(c)$:

$r\ (1)(2,5)(3,4),\ c\ (1,5)(2,4)(3)$; $r\ (1,3)(2)(4,5),\ c\ (1,2)(3,5)(4)$; $r\ (1,5)(2,4)(3),\ c\ (1,4)(2,3)(5)$;
$r\ (1,2)(3,5)(4),\ c\ (1)(2,5)(3,4)$; $r\ (1,4)(2,3)(5),\ c\ (1,3)(2)(4,5)$.

Разглеждаме циркулантната структура на 2-(63,31,15) дизайн с автоморфизми от ред 5 от фигура 2.1. С пермутации на редове и стълбове на матрицата на инцидентност на дизайна можем да направим симетрични всички циркуланти в един циркулантен ред и един циркулантен стълб. Затова без загуба на общност ще разглеждаме матрица на инцидентност, за която $a_{1,j}$ и $a_{i,1}$ са симетрични циркуланти, $i, j = 1, 2, ..., 12$.

Да приложим пермутация $\varphi$ на редовете на такава матрица на инцидентност и нека $\varphi$ размества първите пет точки (първия ред от симетрични циркуланти $a_{1,j}$ ) според $\beta_n$. Да разгледаме като пример $n = 3$, $\beta_n = \beta_3 = (1,5)(2,4)(3)$. Дванадесетте циркуланти $a_{1,j}$ могат да останат същите ако приложим $\beta_n$ на стълбовете им, т.е. ако на стълбовете на матрицата на инцидентност приложим

$\phi = (1,5)(2,4)(3)(6,10)(7,9)(8)...(56,60)(57,59)(58)(61)(62)(63)$.

По този начин на симетричните циркуланти $a_{i,1}$ от първия циркулантен стълб прилагаме $\beta_n$ и те могат да останат същите, ако приложим $\beta_n$ и на редовете им, т.е. на редовете на матрицата на инцидентност прилагаме

$\varphi = (1,5)(2,4)(3)(6,10)(7,9)(8)\ ...\ (56,60)(57,59)(58)(61)(62)(63)$.

С $\varphi$ на редовете и $\phi$ на стълбовете, всъщност прилагаме $\beta_n$ на редовете и стълбовете на всички циркуланти $a_{i,j}$, $i, j = 1, 2, ..., 12$, и те няма да се проме-

нят, само ако всичките са симетрични. Ако това е така, дизайнът е инвариантен относно $D_{10}$, генерирана от автоморфизма $\alpha$ от ред 5 и автоморфизма $\varphi$ от ред 2.

Затова, за да конструираме дизайни, инвариантни относно $D_{10}$, заместваме елементите на съкратените орбитни матрици спрямо автоморфизъм от ред 5 със симетрични циркуланти. Тяхната симетрия обуславя наличието на автоморфизъм $\varphi$ от ред 2.

### 2.6.3 Класификация на Адамаровите дизайни - отсяване на неразширимите орбитни матрици

Повечето съкратени орбитни матрици не са разширими до инвариантни относно $D_{10}$ дизайни. В този конкретен случай при използване на алгоритъма, описан в раздел 1.2.5, намирането на орбитните матрици е изчислително по-тежката задача, а разширяването им до дизайни - по-леката. Затова още при конструирането на орбитните матрици прилагаме тест за разширимост до дизайни и така отсяваме голямата част от тях. Броят на орбитните матрици, които са разширими до дизайни, е: $(a, a^T)$ - 139, $(a, b^T)$ - 114, $(a, c^T)$ - 544, $(a, d^T)$ - 542, $(c, c^T)$ - 367, $(c, d^T)$ - 345, $(d, d^T)$ - 57. Групите от автоморфизми на получените неизоморфни дизайни са представени в Таблица 2.6.

**Таблица 2.6: Ред на групата от автоморфизми на Адамаровите 2-(63,31,15) дизайни, инвариантни относно диедралната група от ред 10**

| ред на групата | от | автоморфизми | самодуални | несамодуални | всички |
|---|---|---|---|---|---|
| 10 | = | 10 | 48 | 5576 | 5624 |
| 20 | = | 10.2 | 17 | 1316 | 1333 |
| 30 | = | 10.3 | 3 | 144 | 147 |
| 40 | = | $10.2^2$ | - | 74 | 74 |
| 60 | = | 10.3.2 | 15 | 254 | 269 |
| 120 | = | $10.3.2^2$ | 10 | 52 | 62 |
| 160 | = | $10.2^4$ | - | 6 | 6 |
| 180 | = | $10.3^2.2$ | - | 4 | 4 |
| 320 | = | $10.2^5$ | 3 | 658 | 661 |
| 360 | = | $10.3^2.2^2$ | 1 | 2 | 3 |
| 640 | = | $10.2^6$ | - | 60 | 60 |
| 960 | = | $10.3.2^5$ | - | 20 | 20 |
| 1920 | = | $10.3.2^6$ | 3 | 42 | 45 |
| 5760 | = | $10.3^2.2^6$ | - | 2 | 2 |
| 10240 | = | $10.2^{10}$ | - | 2 | 2 |
| 15360 | = | $10.3.2^9$ | 1 | - | 1 |
| 30720 | = | $10.3.2^{10}$ | 1 | 4 | 5 |
| 92160 | = | $10.3^2.2^{10}$ | 1 | 4 | 5 |
| 1105920 | = | $10.3^3.2^{12}$ | - | 4 | 4 |
| 30965760 | = | $10.7.3^3.2^{14}$ | - | 2 | 2 |
| 20158709760 | = | $10.31.7^2.3^4.2^{14}$ | 1 | - | 1 |
| Всички | | | 104 | 8226 | 8330 |

Общият брой на конструираните 2-(63,31,15) дизайни е 8330 (104 самодуални), като от $(a, a^T)$ - 398 (27 самодуални), от $(a, b^T)$ - 798, от $(a, c^T)$ - 2680, от $(a, d^T)$ - 2680, от $(c, c^T)$ - 841(61 самодуални), от $(c, d^T)$ - 852 и от $(d, d^T)$ - 80(16 самодуални).

### 2.6.4 Класификация на Адамаровите матрици - намиране на класовете на еквивалентност

Към матрицата на инцидентност на всеки 2-(63,31,15) дизайн добавяме ред и стълб, състоящи се само от единици (означаваме ги ред 0 и стълб 0). Като заменим 0-те с $-1$ получаваме съответната на този дизайн Адамарова матрица от ред 64.

Чрез замяна на редове и стълбове с техните обратни правим така, че ред $i$ ($i = 0, 61, 62, 63$) и стълб $j$ ($j = 0, 61, 62, 63$) да съдържат само единици. Като махнем единичните ред и стълб и заменим $-1$ с 0 получаваме Адамаров 2-(63,31,15) дизайн. Така намираме шестнадесет 2-(63,31,15) дизайна с конструктивните автоморфизми, които са разширими до една и съща Адамарова матрица. Сравнявайки тези съвкупности от по 16 дизайна, отделяме еквивалентните Адамарови матрици. Остават 1691 матрици. След това пресмятаме инвариантите $Im_1$ (от раздел 2.4), които ги разбиват на 1679 класа. За Адамаровите матрици от класове с повече от една матрица, намираме и инвариантите $Im_2$ (раздел 2.4), които са различни за всички тях. Това показва, че тези 1691 Адамарови матрици (39 самодуални и 826 двойки дуални матрици) са нееквивалентни. В Таблица 2.7 е представена тяхната класификация според реда на пълните им групи от автоморфизми, които са пресметнати с програма на Bouyukliev [36]. Същата програма използвахме и за друга проверка на нееквивалентността на тези Адамарови матрици.

### 2.6.5 Съответните афинни 2-(64,16,5) дизайни и хипотезата на Хамада

Конструкцията на Rahilly [205] свързва афинен $2-(16m, 4m, (4m-1)/3)$ дизайн $d$ с Адамаров $2-(16m-1, 8m-1, 4m-1)$ дизайн $D$, чийто дуален дизайн $D^*$ има спред от прави и правите са с размер 3.

Конструкция 1. Избираме точка $w$ от $d$ и разглеждаме като точки на $D$ всички точки от $d$ без $w$. Всеки паралелен клас $C$ на $d$ задава три блока от $D$ както следва: Нека $B_0$ е блок от $d$ от паралелния клас $C$, съдържащ $w$. За всеки блок $B$ от $d$ такъв, че $B \in C$ и $w \notin B$, дефинираме блок от $D$ като $B \cup B_0 - \{w\}$.

Обратното, ако $D$ е симетричен $2-(16m-1, 8m-1, 4m-1)$ дизайн, чийто дуален дизайн $D^*$ има спред $S$ от прави, дефинираме афинен $2-(16m, 4m, (4m-1)/3)$ дизайн $d$ както следва.

Конструкция 2. Точковото множество на $d$ съдържа точките от $D$ плюс една нова точка $w$. Нека $B_1, B_2, B_3$ са три блока от $D$, които съответстват на права в $D^*$ от спреда $S$. Нека $M = B_1 \cap B_2 \cap B_3$. Тогава дефинираме паралелен клас $C$ на $d$, който се състои от четирите блока $B_1' = B_1 - M$, $B_2' = B_2 - M$, $B_3' = B_3 - M$, $B_4' = M \cup \{w\}$.

Да разгледаме 2-ранговете на $D$ и $d$. Нека $a$ е матрицата на инцидентност на $d$, а $A$ – матрицата на инцидентност на $D$ с един допълнителен ред само от единици. Понеже стълбовете на $A$ са суми (над GF(2)) от стълбове на $a$, 2-ранга на $D$ е не по-голям от 2-ранга на $d$ [235].

Оттук нататък с $D$ и $D^*$ означаваме 2-(63,31,15) дизайн и неговия дуален, а с $d$ - съответния 2-(64,16,5) дизайн. Има два геометрични дизайна с тези параметри, т.е. един 2-(63,31,15), получен от PG(5,2) и един 2-(64,16,5), получен от AG(3,4). Известни са три 2-(64,16,5) дизайна с минимален ранг 16. Искаме да проверим дали още такива дизайни не могат да бъдат получени от класифицираните от нас

**Таблица 2.7: Ред на пълната група от автоморфизми на Адамаровите матрици от ред 64, инвариантни относно диедралната група от ред 10**

| ред на групата | от | автоморфизми | самодуални | несамодуални | всички |
|---:|---|---|---:|---:|---:|
| 20 | = | $10.2$ | 1 | 32 | 33 |
| 40 | = | $10.2^2$ | 3 | 210 | 213 |
| 60 | = | $10.3.2$ | 0 | 32 | 32 |
| 80 | = | $10.2^3$ | 2 | 338 | 340 |
| 120 | = | $10.3.2^2$ | 6 | 14 | 20 |
| 160 | = | $10.2^4$ | 2 | 238 | 240 |
| 240 | = | $10.3.2^3$ | 0 | 48 | 48 |
| 320 | = | $10.2^5$ | 0 | 84 | 84 |
| 360 | = | $10.3^2.2^2$ | 0 | 8 | 8 |
| 480 | = | $10.3.2^4$ | 4 | 74 | 78 |
| 640 | = | $10.2^6$ | 0 | 60 | 60 |
| 960 | = | $10.3.2^5$ | 1 | 44 | 45 |
| 1280 | = | $10.2^7$ | 1 | 116 | 117 |
| 1920 | = | $10.3.2^6$ | 1 | 30 | 31 |
| 2560 | = | $10.2^8$ | 1 | 92 | 93 |
| 2880 | = | $10.3^2.2^5$ | 0 | 2 | 2 |
| 3840 | = | $10.3.2^7$ | 1 | 26 | 27 |
| 5120 | = | $10.2^9$ | 0 | 34 | 34 |
| 5760 | = | $10.3^2.2^6$ | 2 | 10 | 12 |
| 7680 | = | $10.3.2^8$ | 1 | 24 | 25 |
| 10240 | = | $10.2^{10}$ | 0 | 12 | 12 |
| 15360 | = | $10.3.2^9$ | 2 | 14 | 16 |
| 20480 | = | $10.2^{11}$ | 0 | 2 | 2 |
| 23040 | = | $10.3^2.2^8$ | 1 | 0 | 1 |
| 30720 | = | $10.3.2^{10}$ | 0 | 4 | 4 |
| 40960 | = | $10.2^{12}$ | 0 | 8 | 8 |
| 61440 | = | $10.3.2^{11}$ | 2 | 2 | 4 |
| 81920 | = | $10.2^{13}$ | 0 | 12 | 12 |
| 122880 | = | $10.3.2^{12}$ | 2 | 6 | 8 |
| 163840 | = | $10.2^{14}$ | 0 | 16 | 16 |
| 184320 | = | $10.3^2.2^{11}$ | 1 | 0 | 1 |
| 245760 | = | $10.3.2^{13}$ | 2 | 12 | 14 |
| 327680 | = | $10.2^{15}$ | 0 | 4 | 4 |
| 491520 | = | $10.3.2^{14}$ | 1 | 12 | 13 |
| 737280 | = | $10.3^2.2^{13}$ | 0 | 4 | 4 |
| 983040 | = | $10.3.2^{15}$ | 0 | 12 | 12 |
| 2621440 | = | $10.2^{18}$ | 0 | 2 | 2 |
| 2949120 | = | $10.3^2.2^{15}$ | 0 | 2 | 2 |
| 3932160 | = | $10.3.2^{17}$ | 0 | 2 | 2 |
| 7864320 | = | $10.3.2^{18}$ | 0 | 4 | 4 |
| 566231040 | = | $10.3^3.2^{21}$ | 0 | 2 | 2 |
| 754974720 | = | $10.3^2.2^{23}$ | 1 | 0 | 1 |
| 9059696640 | = | $10.3^3.2^{25}$ | 0 | 2 | 2 |
| 15854469120 | = | $10.7.3^3.2^{23}$ | 0 | 2 | 2 |
| 165140150353920 | = | $10.31.7^2.3^4.2^{27}$ | 1 | 0 | 1 |
| Всички | | | 39 | 1652 | 1691 |

Адамарови дизайни. Понеже ни интересуват само афинни 2-(64,16,5) дизайни с минимален ранг (16), първо намираме 2-ранга на получените 2-(63,31,15) дизайни. Има 920 дизайна с 2-ранг най-много 16, сред тях: 1 дизайн с ранг 7($PG(5,2)$), 6 с ранг 8, 13 с ранг 11, 151 с ранг 12, 111 с ранг 13, 16 с ранг 14, 36 от ранг 15 и 586 с ранг 16. С конструкцията на Rahilly проверяваме дали от тези дизайни могат да бъдат получени 2-(64,16,5) дизайни с ранг не повече от 16. Намираме два такива 2-(64,16,5) дизайна, съответстващи на Адамарови 2-дизайни с рангове 7($PG(5,2)$) и 8 (с група от автоморфизми от ред 1105920), т.е. намираме само известните примери. По-долу описваме накратко алгоритъма, с който правим това.

### 2.6.6 Алгоритъм за намиране на 2-(64,16,5) дизайни с минимален ранг от Адамарови 2-(63,31,15) дизайни

Задачата свеждаме до построяването на спредове от прави на $D^*$ с минимален ранг. **Входните данни включват:**

- ранга $R$ на Адамаровия дизайн;

- система $S$ от $R$ линейно независими вектора (стълбове на $A$) с дължина 64;

- всички, съответстващи на правите от $D^*$, вектори с дължина 64 и тегло 16, получени като сечение на три стълба на $A$. Максималният брой прави е 651 (достигнат при геометричния дизайн с ранг 7).

- за всяка точка на $D^*$ (блок на $D$): броя на правите, които я съдържат (най-много 31 за тези параметри), като подреждаме точките според този брой.

**Описание на алгоритъма:**

1. Подреждаме точките на $D^*$ според броя на правите, които ги съдържат.
2. Ако първата точка на $D^*$ не е в нито една права, спред е невъзможен и алгоритъмът приключва.
3. Спреда построяваме с търсене с връщане. На поредната стъпка прибавяме права на $D^*$, която не се пресича с никоя от правите на частичното решение и съдържа най-малката неприбавена точка на $D^*$.
4. След прибавяне на права към спреда, прилагаме тест за минималност на частичното решение.
5. След успешен тест за минималност, проверяваме дали съответстващият на добавената права вектор с тегло 16 е линейно независим от векторите в $S$ и, ако е, го добавяме към $S$.
6. Ако рангът на $S$ стане по-голям от 16, както и след неуспешен тест за минималност, махаме от частичното решение последната права (и съответния вектор от $S$, ако сме добавили такъв) и опитваме следващата възможност.
7. Ако намерим спред, го записваме и продължаваме търсенето с връщане.

Резултатите в тази част са съвместни със Златка Матева. Те са публикувани в [T6] и са отразени в дисертацията на Матева за присъждане на научната и образователна степен доктор [9].

# Глава 3

# Дизайни с автоморфизми от малки редове при изброяване без конструиране

Има случаи, в които е известен броят на всички дизайни (заедно с изоморфните). Тогава можем да построим и изследваме само дизайните с нетривиални автоморфизми (те са сравнително малко) и по получените данни за тях да намерим броя на неизоморфните дизайни с тривиален автоморфизъм. Сред дизайните с нетривиални автоморфизми най-много и, респективно, най-трудни за конструиране са тези с автоморфизми от ред 2 или 3. При тях прилагането на локалния подход е затруднено от твърде големия брой орбитни матрици и често пъти в зависимост от спецификата на задачата може да е по-добре да конструираме директно блокове на дизайна, или части от тях, без да минаваме през орбитни матрици. Няколко такива случая са описани в раздели 3.2 и 3.3.

## 3.1 Изброяване без конструиране

Нека означим с $aut$ реда на групата от автоморфизмите на дизайн с дадени параметри, а с $l_{aut}$ броя на изоморфните му дизайни. Дизайните с тривиална група от автоморфизми имат най-голям брой изоморфни $l_1$. За останалите е в сила:

$$l_{aut} = \frac{l_1}{aut} \tag{3.1}$$

Нека за дадени параметри означим с $N_{all}$ броя на всички дизайни, с $N$ - броя на всички неизоморфни дизайни, а с $N_{aut}$ броя на всички неизоморфни дизайни с пълна група от автоморфизми от ред $aut$. Тогава

$$N_{all} = l_1 \sum_{aut} \frac{N_{aut}}{aut} \tag{3.2}$$

$$N = \sum_{aut} N_{aut} \tag{3.3}$$

Ако конструираме всички дизайни с нетривиални автоморфизми и намерим начин да определим $N_{all}$ , от 3.2 можем да пресметнем броя $N_1$ на дизайните с тривиални автоморфизми, а после от 3.3 и броя $N$ на всички неизоморфни дизайни. Такъв подход е приложен при двата случая, разгледани в следващите две части на настоящата глава.

## 3.2 Уилсонови Щайнерови системи от тройки от редове 19 и 21

### 3.2.1 Въведение

Щайнеровите системи от тройки от редове 19 и 21, които съдържат три Щайнерови подсистеми от тройки от ред 7 наричаме Уилсонови, защото са разглеждани за пръв път от Wilson [249], който с помощта на известни резултати за латински квадрати определя общия им брой и извежда долни граници за броя на техните класове на изоморфизъм.

Щайнеровите системи от тройки от ред n ($STS(n)$) са сред най-изучаваните и най-широко използвани дизайни. През миналия век бяха класифицирани всички $STS(n)$ за $n < 19$, а през 2004 Kaski и Östergård публикуваха пълната класификация на $STS(19)$ [150]. В момента $STS(21)$ са системите от най-малък ред, които все още не са класифицирани. Те представляват особен интерес и защото още не е известно дали сред тях има двойно-разрешими. Класификации на различни класове STS(21) са правени от редица автори - Mathon, Phelps и Rosa [179], [180], Tonchev [231], Kapralov и Topalova[143].

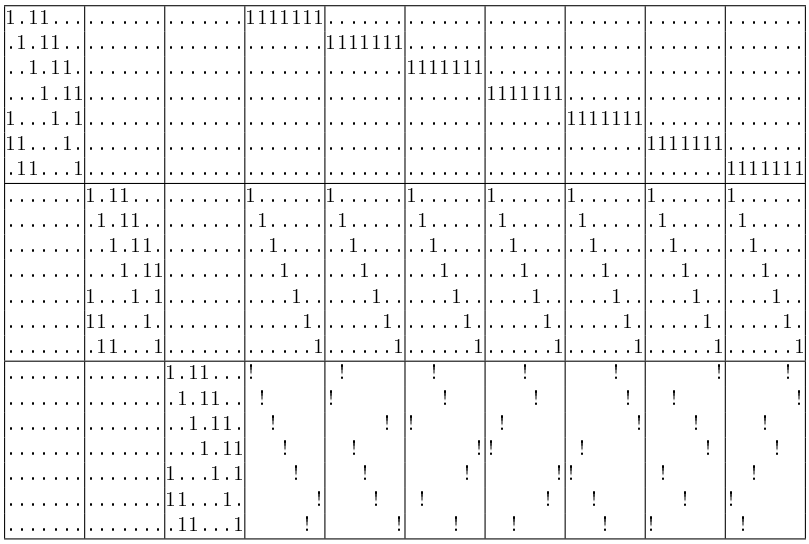
В [T5] е направена класификация на $STS(19)$ с подсистеми от ред 7 и на Уилсонови $STS(21)$. Резултатите са получени по няколко начина, което от една страна цели проверка на резултатите, а от друга дава възможност за изследване на различни страни на получените дизайни. В частност Kaski и Östergård използват модификация на алгоритъма, с който класифицират всички $STS(19)$ в [150], а ние със Златарски построяваме Уилсоновите $STS(19)$ и $STS(21)$ с всички възможни автоморфизми от прост ред и след това прилагаме (3.2) и (3.3). Почти всички резултати сме получавали паралелно и двамата по различни начини или поне с различни програмни реализации. Тук става въпрос за автоморфизми от ред 2 или 3, при каквито локалният подход не винаги е достатъчно ефективен. Затова в повечето случаи сме използвали особеностите на конкретната задача, за да реализираме изчерпващо търсене за конкретния случай. Представа за подхода ни може да бъде получена от следващите раздели, където са дадени подробности за начина, по който авторът е конструирал дизайните в различните случаи. Освен това със софтуер на автора (раздел 1.2.9) са класифицирани резолюциите на всички $STS(21)$ с тривиален автоморфизъм, построени от Kaski и Östergård и е установено, че сред тях няма двойно-разрешими. Всички Киркманови системи

от тройки с нетривиални автоморфизми са известни [62], но тези с тривиални са нови и преди тази работа не бяха изследвани за двойна разрешимост.

### 3.2.2 Уилсонови $STS(19)$ и $STS(21)$

Wilson разглежда STS(21) с три подсистеми STS(7) върху непресичащи се множества от точки [249] и показва, че броят на неизоморфните STS(21) от този тип е поне 2160980. Да се спрем на специфичните свойства на такава STS(21). Без загуба на общност ще разглеждаме *подредени* системи, т.е. ще считаме, че точковите множества на трите STS(7) са $P_i = \{1+7i, 2+7i, \ldots, 7+7i\}$, $i \in \{0,1,2\}$. С размяна на редове и стълбове матрицата на инцидентност може да бъде представена във вида, показан на фигура 3.1. Правоъгълниците с удивителни съдържат квадратни матрици $7 \times 7$ с по една единица в ред и стълб и са тази част от матрицата на инцидентност, която трябва да бъде попълнена по всички възможни начини. С удивителните (вместо единици) е дадено едно такова попълване.

Фигура 3.1: Wilson type STS(21)



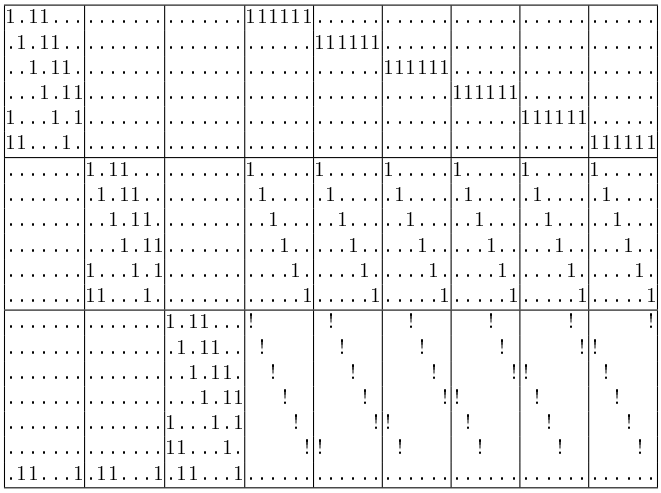
Ако вместо всяка удивителна запишем в кое стълбче на $7 \times 7$ подматрицата се намира тя, получаваме следния латински квадрат от ред 7:

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 7 | 6 |
| 2 | 1 | 4 | 5 | 6 | 3 | 7 |
| 3 | 6 | 1 | 2 | 7 | 5 | 4 |
| 4 | 3 | 7 | 1 | 2 | 6 | 5 |
| 5 | 4 | 6 | 7 | 1 | 2 | 3 |
| 7 | 5 | 2 | 6 | 3 | 4 | 1 |
| 6 | 7 | 5 | 3 | 4 | 1 | 2 |

Уилсонова STS(19) съдържа три подсистеми STS(7), които имат една обща точка. И в този случай ще разглеждаме *подредени* системи, т.е. ще считаме, че

точковите множества на трите STS(7) са $P_i = \{1+6i, 2+6i, \ldots, 6+6i, 19\}$, $i \in \{0,1,2\}$ (фигура 3.2).

Фигура 3.2: Wilson type STS(19)



Ако вместо всяка удивителна запишем в кое стълбче на $6 \times 6$ подматрицата се намира тя, получаваме следния латински квадрат от ред 6:

| | | | | | |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 |
| 2 | 3 | 4 | 5 | 6 | 1 |
| 3 | 4 | 5 | 6 | 1 | 2 |
| 4 | 5 | 6 | 1 | 2 | 3 |
| 5 | 6 | 1 | 2 | 3 | 4 |
| 6 | 1 | 2 | 3 | 4 | 5 |

Заради тази връзка оттук нататък неизвестната подматрица на матрицата на инцидентност на фигури 3.1 и 3.2 ще наричаме *латинска част* на съответната Уилсонова Щайнерова система от тройки.

Един латински квадрат се нарича редуциран, ако числата в първия му ред и стълб са подредени във възходящ ред. Броят на редуцираните латински квадрати от ред 7 е 16942080, а от ред 6 - 9408. Следователно броят на всички латински квадрати от ред 7 е 7!6!16942080, а от ред 6 - 6!5!9408. Толкова са и различните начини за попълване на съответните латински части.

Щайнеровата система от тройки от ред 7 е единствена с точност до изоморфизъм и има пълна група от автоморфизми от ред 168. От (3.1) следва, че броят на различните STS(7) е $7!/168 = 30$. Тогава броят на всички подредени Уилсонови STS(21) и STS(19) е $30^3$7!6!16942080 и съответно $30^3$6!5!9408. За по-нататъшните разглеждания е важно да знаем дали Уилсоновите STS(21) и STS(19) могат да имат повече от три подсистеми STS(7).

**Лема 3.2.1.** *Уилсонова STS(21) има точно три подсистеми STS(7).*

**Доказателство.** Нека $X_1, X_2, X_3$ са подсистеми STS(7) на Уилсонова STS(21). Ако две подсистеми на STS($v$) имат общи точки, то сечението им също е подсистема [71, Lemma 6.45]. Затова една подсистемата $X$ от ред 7 трябва или (а) да бъде измежду подсистемите $X_1, X_2, X_3$; или (б) да пресича точковите множества на $X_1, X_2, X_3$ в 1, 3, 3 точки (в някакъв ред). Подсистема STS(3) се състои от един блок, съдържащ всичките 3 точки. В случай (б) $X$ трябва да има две непресичащи се подсистеми STS(3), тоест два непресичащи се блока. Понеже всеки два блока на една STS(7) имат точно една обща точка, случай (б) е невъзможен. □

От Лема 3.2.1 следва, че една подредена система може да бъде трансформирана в друга само чрез пермутации на трите подсистеми и пермутации на точките на всяка от тях. Затова Уилсонова STS(21) без автоморфизми има $3!(7!)^3$ изоморфни подредени системи, от което Wilson заключава, че броят на неизоморфните STS(21) е поне

$$\frac{30^3 7!6!16942080}{3!(7!)^3} = 2160980.$$

При точното разглеждане от (3.2) получаваме

$$\sum_{aut} \frac{N_{aut} 3!(7!)^3}{aut} = 30^3 7!6!16942080. \tag{3.4}$$

Оттук можем да определим броя на неизоморфните дизайни без автоморфизми, ако построим всички дизайни с нетривиални автоморфизми. Затова първо ще се спрем на това какви автоморфизми са възможни.

**Теорема 3.2.2.** *Възможните автоморфизми на Уилсонова STS(21) са:*

*1. Автоморфизъм от ред 2 с 9 фиксирани точки и 18 фиксирани блока.*

*2. Автоморфизъм от ред 2 със 7 фиксирани точки и 14 фиксирани блока.*

*3. Автоморфизъм от ред 2 с 3 фиксирани точки и 10 фиксирани блока.*

*4. Автоморфизъм от ред 3 с 3 фиксирани точки и 4 фиксирани блока.*

*5. Автоморфизъм от ред 3 без фиксирани точки и 1, 4 или 7 фиксирани блока.*

*6. Автоморфизъм от ред 7 със 7 фиксирани точки и 7 фиксирани блока.*

*7. Автоморфизъм от ред 7 без фиксирани точки и блокове.*

**Доказателство.** Ако Щайнерова система от тройки от ред $v$ има автоморфизъм от някакъв прост ред и с $f$ фиксирани точки, то на тези точки има подсистема STS($f$) и $f < v/2$ [10], [67], [80]. Затова броят на фиксираните точки винаги е ред на Щайнерова система от тройки. Следователно $STS(21)$ допуска автоморфизми с 0, 1, 3, 7 или 9 фиксирани точки.

Понеже освен трите STS(7) (от дефиницията) няма други подсистеми STS(7) ( Лема 3.2.1), автоморфизъм на Уилсонова STS(21) трябва да изобразява всяка от

трите подсистеми в някоя от другите (автоморфизъм от ред 2 или 3) или в себе си ( тогава част от точковите орбити са орбити спрямо автоморфизъм на STS(7), тоест от ред 2, 3 или 7). Затова 2, 3 и 7 са възможните редове на автоморфизми на Уилсонова STS(21).

Да разгледаме автоморфизмите, които трансформират $X_1, X_2, X_3$ една в друга. Ако трансформацията е циклична $(X_1\ X_2\ X_3)$, получаваме номер 5 от горния списък. Ако $(X_1\ X_2)(X_3)$, има две възможности: Ако точките на третата STS(7) са фиксирани следва номер 2, а ако отиват една в друга при автоморфизма от ред 2, получаваме номер 3.

Да разгледаме автоморфизмите, които фиксират всяка от трите STS(7). Ако автоморфизмът е от ред 2 (следователно с 3 фиксирани точки за всяка STS(7)), следва номер 1. Ако автоморфизмът е от ред 7 следват номера 6 и 7 в зависимост от това дали точките на една от подсистемите STS(7) са фиксирани или не. Ако автоморфизмът е от ред 3 (1 фиксирана точка за всяка STS(7)) получаваме номер 4 с 3 фиксирани точки. В последния случай, въпреки, че $9 < 21/2$, не е възможно точките на едната подсистема да са фиксирани, защото подсистемата STS(9) на фиксираните точки ще съдържа подсистема STS(7), което е невъзможно $(7 > 9/2)$.

Във всички разгледани случаи броят на фиксираните блокове се получава от условието всяка двойка точки да се съдържа точно в един блок. □

Всички STS(21) с автоморфизми от ред 7 са класифицирани в [231]. Ние установихме, че сред тях има 8 Уилсонови с автоморфизми от ред 7 със 7 фиксирани точки и 6 с автоморфизми от ред 7 без фиксирани точки. Всички те притежават и автоморфизми от други прости редове. Ние конструираме всички неизоморфни Уилсонови STS(21) с всички възможни автоморфизми от редове 2 и 3. В следващия раздел е представен подходът, който е използван във всеки от тези случаи.

### 3.2.3 Алгоритми и особености на програмната реализация

#### Тест за минималност

Нека $G_X$ е групата от пермутации, която фиксира всяка от трите подсистеми от ред 7, като всяка пермутация от тази група размества точките на трите подсистеми по един и същ начин. Две Уилсонови STS(21) са изоморфни, ако едната се изобразява в другата при прилагане на пермутация от $G_X$. Затова тестът за минималност проверява дали пермутация от $G_X$ трансформира даденото частично решение в лексикографски по-малко. Понеже $|G_X| = 168$ (редът на групата от автоморфизми на STS(7)), този тест е много бърз, но сред получените дизайни има и изоморфни (те се изобразяват един в друг от пермутации, при които $X_3$ отива в $X_1$ или $X_2$), т.е. тестът за минималност е частичен. Това налага прилагането на тест за изоморфизъм на пълните решения. Теста за минималност включваме на различни места при различните случаи, чието описание следва.

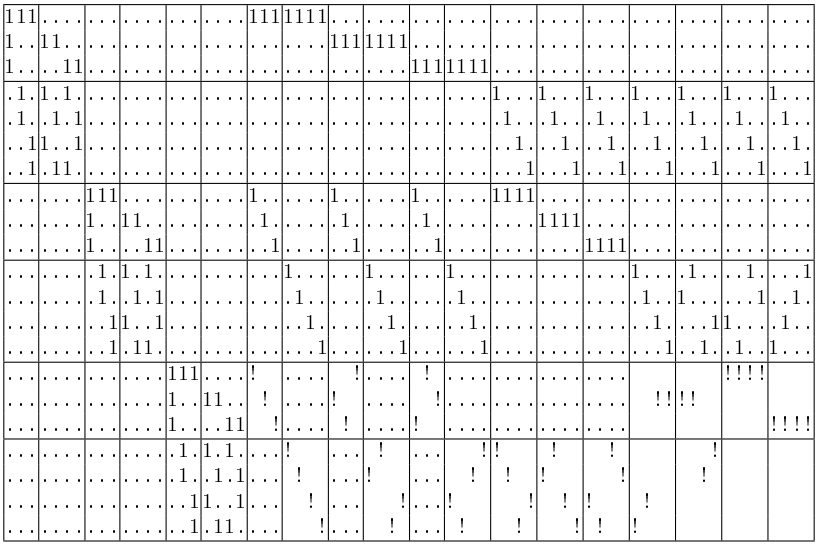
**Автоморфизъм от ред 2 с 9 фиксирани точки**

Разглеждаме автоморфизъм $\alpha$ от ред 2, зададен с
$\alpha_p = (1)(2)(3)(4,5)(6,7)(8)(9)(10)(11,12)(13,14)(15)(16)(17)(18,19)(20,21)$
$\alpha_b = (1)(2)(3)(4,5)(6,7)(8)(9)(10)(11,12)(13,14)(15)(16)(17)(18,19)(20,21)(22)(23)$
$(24)(25,26)(27,28)(29)(30)(31)(32,33)(34,35)(36)(37)(38)(39,40)(41,42)...(69,70)$.

Представяме матрицата на инцидентност във вида, показан на фигура 3.3.

Неизвестната част от нея конструираме блок по блок. Тест за минималност прилагаме след попълване на неизвестната част на блокове 22 - 28, т.е първите два правоъгълника с удивителни (за които остават 4 вместо 48 решения) и на пълните решения. Получаваме 984 неизоморфни дизайна.

Фигура 3.3: STS(21) - Автоморфизъм от ред 2 с 9 фиксирани точки



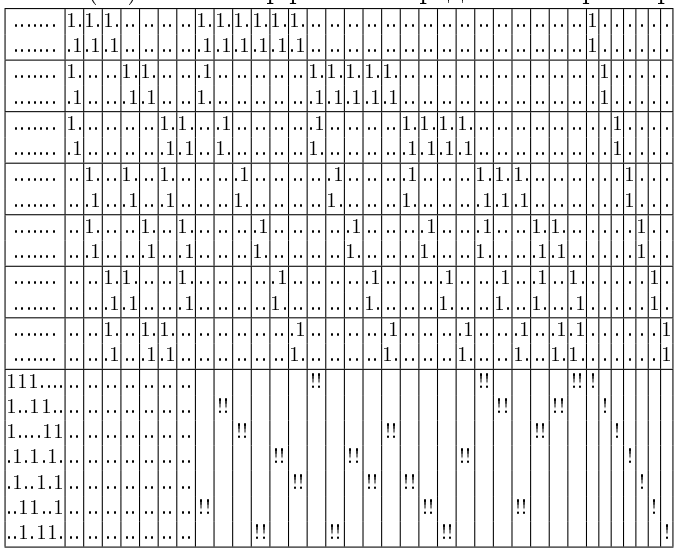
**Автоморфизъм от ред 2 със 7 фиксирани точки**

Разглеждаме автоморфизъм $\alpha$, зададен с
$\alpha_p = (1,2)(3,4)...(13,14)(15)(16)(17)(18)(19)(20)(21)$
$\alpha_b = (1)(2)...(7)(8,9)(10,11)...(62,63)(64)(65)...(70)$.
Започваме от матрицата на инцидентност от фигура 3.4. Първо намираме възможните решения за всички фиксирани блокове и им прилагаме теста за минималност. След това изчерпваме всички възможности за фиксираната точка на всеки нефиксиран блок. Получаваме 1256 неизоморфни дизайна.

**Автоморфизъм от ред 2 с 3 фиксирани точки**

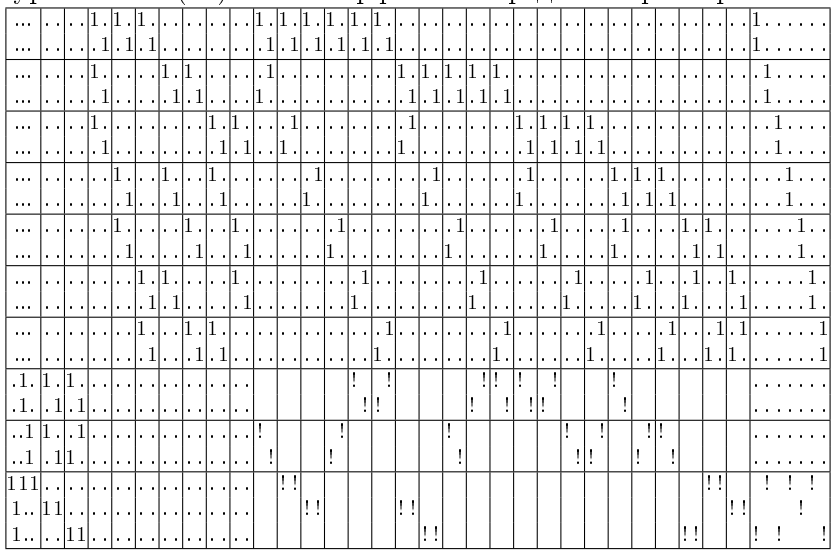
Разглеждаме автоморфизъм $\alpha$, зададен с
$\alpha_p = (1,2)(3,4)(5,6)...(17,18)(19)(20)(21)$,
$\alpha_b = (1)(2)(3)(4,5)(6,7)...(62,63)(64)(65)...(69)(70)$.

Фигура 3.4: STS(21) - Автоморфизъм от ред 2 със 7 фиксирани точки



Преобразуваме матрицата на инцидентност до тази на фигура 3.5. Да разгледаме свойствата на тази матрица. Частта, която съответства на нетривиалните точкови и блокови орбити ( редове 1 - 18 и колони 8 - 63 на фигура 3.5 ) се състои от циркуланти от ред 2. Ако заместим ненулевите циркуланти с 1, а останалите с 0, получаваме матрица от нули и единици с размери $9 \times 28$. Тя може да бъде разширена до матрица на инцидентност на 2-(10,3,2) дизайн чрез добавяне по единствен начин на един ред и две колони.

Фигура 3.5: STS(21) - Автоморфизъм от ред 2 с 3 фиксирани точки



Започваме от матриците на инцидентност на 2-(10,3,2) дизайните (те са 46), които съдържат подсистема STS(7), изтриваме по всевъзможните начини един ред и две колони и заместваме всяка 0 с матрица от 4 нули, а всяка единица - с

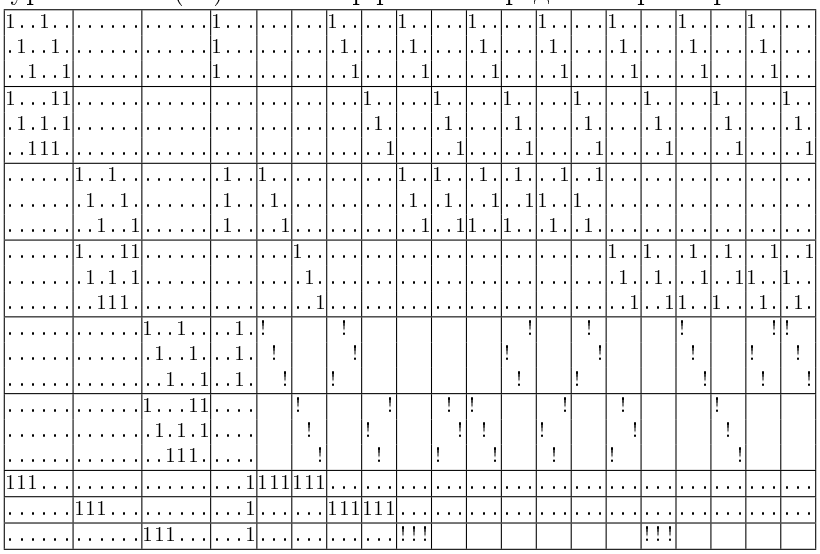
циркуланта от ред две. След пълен тест за еквивалентност намираме 71 неекви-валентни решения за първите 18 реда на матрицата на инцидентност на търсените дизайни. Накрая добавяме трите фиксирани реда по всички възможни начини и след теста за изоморфизъм остават 7617 неизоморфни дизайна.

**Автоморфизъм от ред 3 с 3 фиксирани точки**

Разглеждаме автоморфизъм $\alpha$ от ред 3, зададен с
$\alpha_p = (1,2,3)(4,5,6)...(16,17,18)(19)(20)(21)$,
$\alpha_b = (1,2,3)(4,5,6)...(19,20,21)(22)(23)(24)(25)(26,27,28)(29,30,31)...(68,69,70)$.

Работим с матрицата на инцидентност от фигура 3.6.

Фигура 3.6: STS(21) - Автоморфизъм от ред 3 с 3 фиксирани точки



С помощта на локалния подход построяваме 485 неизоморфни дизайна.

**Автоморфизъм от ред 3 без фиксирани точки**

Разглеждаме автоморфизъм $\alpha$, зададен с
$\alpha_p = (1,8,15)(2,9,16)(3,10,17)(4,11,18)(5,12,19)(6,13,20)(7,14,21)$.

Работим с матрицата на инцидентност от фигура 3.1. Да означим с $(l,m,n)$ блок с точки $l$, $7+m$ и $14+n$. При този автоморфизъм дизайнът съдържа и блоковете $(m,n,l)$ и $(n,l,m)$. Ако $m=n=l$ блокът е фиксиран. От фигура 3.1 се вижда, че можем да фиксираме блокове $22,30,38,46,54,62,70$. При фиксиране на всичките получаваме автоморфизъм със 7 фиксирани блока. Когато разглеждаме случая с един фиксиран блок, без загуба на общност можем да изберем блок 22. Има два нееквивалентни начина за избиране на 4 фиксирани блока, т.е. $22,30,38,46$ и $22,30,38,62$. Четирите случая разглеждаме с помощта на локалния подход и получаваме 469 неизоморфни дизайна.

### 3.2.4 Класификационни резултати

Резултатите за STS(21) са представени в Таблица 3.1, където в колона $a_b$ е даден броят на дизайните с автоморфизъм от ред $a$ с $b$ фиксирани точки. С използване на (3.4) получаваме, че броят на Уилсоновите STS(21) с тривиалния автоморфизъм е 2156186 и следователно всички STS(21) от Уилсонов тип са 2166351. Конструираните Уилсонови STS(21) са тествани за разрешимист и двойна разрешимост. В Таблица 3.1 $Nrd$ е броят на разрешимите дизайни, а $Nr$ - на неизоморфните резолюции. Сред тях няма двойно-разрешими.

**Таблица 3.1: Уилсонови STS(21)**

| | Конструктивен автоморфизъм | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| \|Aut\| | $2_9$ | $3_3$ | $2_3$ | $3_0$ | $2_7$ | $7_0$ | $7_7$ | All |
| 1 | | | | | | | | 2156186 |
| 2 | 636 | | 7239 | | 1039 | | | 8914 |
| 3 | | 317 | | 368 | | | | 685 |
| $4=2^2$ | 271 | | 208 | | 52 | | | 271 |
| $6=2.3$ | | 126 | 95 | 71 | 102 | | | 197 |
| $8=2^3$ | 41 | | 39 | | 27 | | | 41 |
| $9=3^2$ | | 3 | | 3 | | | | 3 |
| $12=2^2.3$ | 6 | 1 | 4 | 5 | | | | 6 |
| $16=2^4$ | 4 | | 4 | | 4 | | | 4 |
| $18=2.3^2$ | | 7 | 2 | 7 | 5 | | | 7 |
| $21=3.7$ | | 2 | | | | | 2 | 2 |
| $24=2^3.3$ | 17 | 11 | 17 | 6 | 10 | | | 17 |
| $42=2.3.7$ | | 6 | | | 6 | 2 | 4 | 6 |
| $48=2^4.3$ | 2 | 2 | 2 | | 2 | | | 2 |
| $72=2^3.3^2$ | 5 | 5 | 5 | 5 | 4 | | | 5 |
| $126=2.3^2.7$ | | 1 | | 1 | 1 | 1 | | 1 |
| $144=2^4.3^2$ | 1 | 1 | 1 | 1 | 1 | | | 1 |
| $294=2.3.7^2$ | | 1 | | | 1 | 1 | 1 | 1 |
| $882=2.3^2.7^2$ | | 1 | | 1 | 1 | 1 | 1 | 1 |
| $1008=2^4.3^2.7$ | 1 | 1 | 1 | 1 | 1 | 1 | | 1 |
| All | 984 | 485 | 7617 | 469 | 1256 | 6 | 8 | 2166351 |

| Разрешимост | | |
|---|---|---|
| \|Aut\| | Nrd | Nr |
| 1 | 10981 | 12030 |
| 2 | 352 | 677 |
| 3 | 67 | 127 |
| 4 | 23 | 51 |
| 6 | 19 | 53 |
| 8 | 5 | 19 |
| 9 | | |
| 12 | | |
| 16 | 1 | 1 |
| 18 | 2 | 10 |
| 21 | 1 | 1 |
| 24 | 6 | 21 |
| 42 | 4 | 7 |
| 48 | | |
| 72 | 1 | 11 |
| 126 | 1 | 4 |
| 144 | | |
| 294 | 1 | 2 |
| 882 | 1 | 4 |
| 1008 | 1 | 18 |
| Total | 11466 | 13036 |

С подход подобен на описания в предходния раздел конструираме и всички Уилсонови STS(19) с нетривиални автоморфизми. Резултатите са представени в Таблица 3.2, където $s$ в означението на конструктивния автоморфизъм означава, че той трансформира една в друга подсистемите STS(7), докато в останалите случаи трансформира всяка от тях в себе си.

Уилсонова STS(19) може да съдържа повече от 3 подсистеми STS(7) и поради това класификацията на Уилсоновите STS(19) с нетривиални автоморфизми не може да се използва за определяне на броя на неизоморфните дизайни с тривиалния автоморфизъм по начин, подобен на показания за STS(21). Тя може да се използва, обаче, за проверка на верността на получените резултати, ако дизайни-

**Таблица 3.2: Уилсонови STS(19) с нетривиални автоморфизми**

| \|Aut\| | $2_3$s | $2_7$ | $2_7$s | $3_1$s | $3_1$ | $3_7$ | Nd |
|---|---|---|---|---|---|---|---|
| 2 | 126 | 230 | 482 | | | | 838 |
| 3 | | | | 41 | | 64 | 105 |
| 4 | 98 | 137 | 74 | | | | 137 |
| 6 | 16 | 12 | 40 | 22 | 8 | 38 | 68 |
| 8 | 8 | 8 | 8 | | | | 8 |
| 12 | 18 | 21 | 18 | 11 | | 10 | 21 |
| 18 | 2 | | 8 | 2 | 10 | 8 | 10 |
| 54 | | | 2 | 2 | 2 | 2 | 2 |
| 108 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 144 | 1 | 1 | 1 | 1 | 1 | | 1 |
| 432 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Total | 271 | 411 | 635 | 81 | 23 | 124 | 1192 |

те с тривиална група са построени по някакъв начин. Изследването на дизайните (9297 на брой) с тривиален автоморфизъм (построени в [T5] от съавторите) показва, че никой от тях не съдържа повече от три подсистеми STS(7). Тогава за Уилсоновите STS(19) е в сила:

$$\sum_{aut} \frac{N_{autW} 3!(6!)^3}{autW} = 30^3 6!5!9408, \tag{3.5}$$

където $autW$ е редът на групата от автоморфизми на дизайна, която може да бъде генерирана от пермутации на трите подредени STS(7) и точките на всяка от тях. Тази група не съвпада с пълната група от автоморфизми само за последните три дизайна от Таблица 3.2. Те имат по 12 подсистеми STS(7) и за тях $autW = |Aut|/4$.

Изследванията от настоящата част са включени в съвместната ни статия [T5] с Petteri Kaski, Patric Östergård и Росен Златарски.

## 3.3 Двойни на проективната равнина от ред 4

### 3.3.1 Въведение

Двойните на даден дизайн са друг случай, в който е известен общият брой (заедно с изоморфните). Всеки 2-$(v, k, \lambda)$ дизайн обуславя съществуването на 2-$(v, k, m\lambda)$ дизайни (за цяло $m > 1$), които наричаме *квазикратни* на 2-$(v, k, \lambda)$ дизайна. Един квазикратен 2-$(v, k, m\lambda)$ е *разделим* на $m$ 2-$(v, k, \lambda)$ дизайна ако съществува разбиване на блоковете му на $m$ съвкупности, всяка от които формира 2-$(v, k, \lambda)$ дизайн. За $m = 2$ квазикратните дизайни се наричат *квазидвойни*, а разделимите квазидвойни дизайни – *двойни*.

Ще означаваме с $(D_1 \cup D_2)$ двоен дизайн, който може да бъде разделен на двата дизайна $D_1$ и $D_2$. *Разделянето* на двойния дизайн $D$ с параметри $t$-$(v, k, 2\lambda)$ можем да представим чрез две множества от половината блокове, всяко от които формира 2-$(v, k, \lambda)$ дизайн. Едно очевидно разделяне на двойния дизайн $(D_1 \cup D_2)$ е $\{D_1, D_2\}$; редът на изброяване на съставните дизайни не е от значение. Ще използваме означението $D_2 = \mu D_1$, в което $\mu$ е пермутация на точките на $D_1$, с която се получава $D_2$. Двойните дизайни могат да бъдат разделими по повече от един начин.

Две разделяния $\{D_1, D_2\}$ и $\{D_3, D_4\}$ на един и същ двоен дизайн са *еквивалентни* тогава и само тогава, когато съществува пермутация $\mu$ такава, че $D_3 = \mu D_1$ и $D_4 = \mu D_2$, или $D_4 = \mu D_1$ и $D_3 = \mu D_2$. Двоен дизайн, който има, с точност до еквивалентност , само едно разделяне е *уникално разделим*.

Разделимите 2-$(21, 5, 2)$ дизайни са обект на настоящия раздел. Ще покажем, че те са уникално разделими. С точност до еквивалентност, има единствен 2-$(21, 5, 1)$ дизайн (проективната равнина от ред 4 - $PG(2, 4)$) и разделимите 2-$(21, 5, 2)$ дизайни са негови двойни. Първата долна граница за броя на разделимите 2-$(21, 5, 2)$ дизайни е изведена в [181] и е 10. Долни граници за броя на двойните на проективни равнини са представени в [137] и [138]. Според тях двойните на проективната равнина от ред 4 са поне 24.

Тук ние изброяваме разделимите 2-$(21, 5, 2)$ дизайни, като конструираме тези от тях, които имат нетривиални автоморфизми и чрез тях изчисляваме броя на всички разделими 2-$(21, 5, 2)$. Това е възможно, защото те съдържат поддизайни 2-$(21, 5, 1)$ с известни свойства. Освен в раздел 3.2, примери за изброяване на дизайни с поддизайни са дадени в [157], [158], [249].

В [13] са конструирани всички 2-$(21, 5, 2)$ дизайни с автоморфизми от нечетен прост ред. Те са 22998 на брой, като сред тях има 4170 разделими. За нуждите на настоящата работа остава да бъдат построени разделимите 2-$(21, 5, 2)$ дизайни с автоморфизми от ред 2. Има два типа такива автоморфизми, т.е. които изобразяват всеки от съставните 2-$(21, 5, 1)$ дизайни в себе си и такива, които изобразяват единия 2-$(21, 5, 1)$ в другия (и обратното). Ние построяваме всичките 40485

дизайна от първия тип и всичките 991957 от втория и изследваме групите от автоморфизмите им. По тях пресмятаме, че броят на всички двойни на проективната равнина от ред 4 е 1746461307.

### 3.3.2 Двойни на уникално разделим дизайн

Оттук нататък разглеждаме двойни на единствени за дадените параметри (с точност до изоморфизъм) дизайни. Затова вместо $(D_1 \cup D_2)$ често използваме означението $(D \cup \varphi D)$, където съставният дизайн $\varphi D$ е получен от $D$ с пермутация $\varphi$ на точките.

В останалата част на този раздел с $D$ ще означаваме 2-$(v, k, \lambda)$ дизайн, а с $(D \cup \varphi D)$ уникално разделим двоен на $D$. С $G$ означаваме пълната група от автоморфизми на $D$, с $G_\varphi$ – общата част на пълните групи от автоморфизми на $D$ и $\varphi D$, а с $\widehat{G}_\varphi$ – пълната група от автоморфизми на двойния дизайн $(D \cup \varphi D)$.

Множеството от всички $v!$ пермутации $\varphi$ на точките на $D$ може да бъде разделено на класове $\mathcal{C}_G(\varphi)$, където $\mathcal{C}_G(\varphi)$ е множество от пермутации $\psi$, такова, че двойният $(D \cup \psi D)$ е изоморфен на $(D \cup \varphi D)$. Тогава

$$v! = \sum_{\mathcal{C}_G(\varphi)} |\mathcal{C}_G(\varphi)| \tag{3.6}$$

В следващото твърдение определяме размера на класа на изоморфизъм $\mathcal{C}_G(\varphi)$, съдържащ пермутацията $\varphi$.

**Твърдение 3.3.1.** *За множеството $\mathcal{C}_G(\varphi)$ от точкови пермутации $\psi$, за които $(D \cup \psi D)$ е изоморфен на $(D \cup \varphi D)$ е в сила:*

$$\mathcal{C}_G(\varphi) = G\varphi G \cup G\varphi^{-1}G.$$

*Броят на тези пермутации е*

$$|\mathcal{C}_G(\varphi)| = \begin{cases} |G|^2/|G_\varphi| & \text{ако } G\varphi G = G\varphi^{-1}G, \\ 2|G|^2/|G_\varphi| & \text{в противен случай, т.е. ако } G\varphi G \cap G\varphi^{-1}G = \emptyset. \end{cases} \tag{3.7}$$

**Доказателство.** Нека $\psi \in G\varphi G$. Тогава $\exists \alpha, \beta \in G : \psi = \beta\varphi\alpha$. Следователно $(D \cup \psi D)$ е изоморфен на $(D \cup \varphi D)$ понеже $\beta^{-1}(D \cup \beta\varphi\alpha D) = (D \cup \varphi D)$. Аналогично, ако $\psi \in G\varphi^{-1}G$, то $\exists \alpha, \beta \in G : \psi = \beta\varphi^{-1}\alpha$, и $(D \cup \psi D)$ е изоморфен на $(D \cup \varphi D)$ понеже $\varphi\beta^{-1}(D \cup \beta\varphi^{-1}\alpha D) = (\varphi D \cup D)$.

Обратно, ако $(D \cup \psi D)$ е изоморфен на $(D \cup \varphi D)$, то понеже $(D \cup \varphi D)$ е уникално разделим, са възможни само два случая. В първия съществува пермутация $\mu$, за която $\mu D = D$ и $\mu\varphi D = \psi D$, от където следва, че $\mu \in G$ и $\varphi^{-1}\mu^{-1}\psi \in G$ и затова $\psi \in G\varphi G$. Във втория случай съществува пермутация $\mu$ такава, че $\mu D = \psi D$ и $\mu\varphi D = D$, откъдето следва $\mu^{-1}\psi \in G$ и $\mu\varphi \in G$ и поради това $\psi \in \mu G$ и $\mu \in G\varphi^{-1}$ $\Rightarrow \psi \in G\varphi^{-1}G$.

От теорията за съседните класове [117, Theorem 2.19] следва, че $|G\varphi G| = |G\varphi^{-1}G| = |G|^2/|G_\varphi|$, а $G\varphi G \cap G\varphi^{-1}G = \emptyset$ или $G\varphi G = G\varphi^{-1}G$. Тогава $|G\varphi G \cup G\varphi^{-1}G| = 2|G|^2/|G_\varphi|$ ако $G\varphi G \cap G\varphi^{-1}G = \emptyset$ и $|G\varphi G \cup G\varphi^{-1}G| = |G|^2/|G_\varphi|$ в противен случай. □

**Твърдение 3.3.2.** *Ако $G\varphi G = G\varphi^{-1}G$, съществува $\omega \in \widehat{G}_\varphi$ такава, че $(D \cup \varphi D) = (D \cup \omega D)$. Пермутацията $\omega$ изобразява $D$ във $\varphi D$ и обратното. Ако $|G_\varphi| = 1$, то $\omega$ е от ред 2.*

**Доказателство.** Понеже $G\varphi G = G\varphi^{-1}G$, то $\varphi^{-1} \in G\varphi G$ и следователно $\exists \rho, \sigma \in G : \varphi^{-1} = \rho\varphi\sigma$. Това означава, че $\varphi\rho\varphi\sigma = 1$, т.е. $\varphi\rho\varphi \in G$.

Нека $\omega = \varphi\rho$. Тогава $\omega D = \varphi\rho D = \varphi D$. Понеже $\omega\varphi \in G$, $\omega\varphi D = D$, $(D \cup \varphi D) = (D \cup \omega D)$ и $\omega$ изобразява $D$ във $\varphi D$ и обратното.

От $\omega^2 D = D$ и $\omega^2 \varphi D = \varphi D$ следва, че $\omega^2 \in G_\varphi$. Ако $|G_\varphi| = 1$, това означава $\omega^2 = 1$. □

**Следствие 3.3.3.** *Ако $|\widehat{G}_\varphi| = 1$, то $|\mathcal{C}_G(\varphi)| = 2|G|^2$.*

Нека $N_i$ ($N_i'$) е броят на класовете на изоморфизъм $\mathcal{C}_G(\varphi)$, за които $|G_\varphi| = i$ и $G\varphi G \cap G\varphi^{-1}G = \emptyset$ ($G\varphi G = G\varphi^{-1}G$). Използвайки (3.7), можем да преобразуваме равенството (3.6) до

$$v! = 2|G|^2 N_1 + |G|^2 N_1' + \sum_{i>1} \frac{2|G|^2}{i} N_i + \sum_{i>1} \frac{|G|^2}{i} N_i' \,. \tag{3.8}$$

Нека $N$ е общият брой неизоморфни двойни на $D$. Тогава

$$N = N_1 + N_1' + \sum_{i>1} N_i + \sum_{i>1} N_i' \,. \tag{3.9}$$

Ако знаем броя на двойните на $D$ с нетривиални автоморфизми, т.е. знаем $N_1'$, както и $N_i$ и $N_i'$ за всяко $i > 1$, можем да използваме равенство (3.8), за да получим броя $N_1$ на двойните на $D$ с тривиален автоморфизъм, а след това от (3.9) да пресметнем пълния брой $N$ на двойните на $D$.

**Следствие 3.3.4.** *Един 2-($v$,$k$,$\lambda$) дизайн $D$, на който всички двойни са уникално разделими, има най-малко $v!/(2|G|^2)$ неизоморфни двойни ($G$ е пълната група от автоморфизми на $D$).*

Всички квазидвойни на проективните равнини от редове 2 и 3 бяха известни преди настоящото изследване. С точност до изоморфизъм има 4 двойни на проективната равнина от ред 2 и 184 на проективната равнина от ред 3. Ние установихме, че те са уникално разделими и удовлетворяват равенствата (3.8) и (3.9).

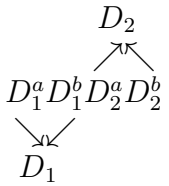
Също така проверихме дали (3.8) и (3.9) са в сила за двойните на афинните равнини от редове 2, 3 и 4. Най-тясно свързани с настоящия резултат са двойните

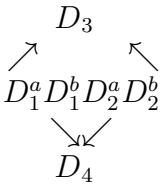
на афинната равнина от ред 4, чиято уникална разделимост следва от разглежданията в раздел 3.3.3 . Всички двойни на афинната равнина от ред 4 са между разрешимите 2-$(16, 4, 2)$ дизайни, конструирани в [149], за които дизайните с нетривиални автоморфизми са достъпни от страницата на авторите. Ние установихме, че 9102 от тях са разделими и с равенства (3.8) и (3.9) намерихме, че броят на двойните на афинната равнина от ред 4 е 320061. С наш софтуер конструирахме по друг начин всички тези двойни и получихме същия резултат.

Изброяването на двойните на проективната равнина от ред 4 е обект на раздел 3.3.6.

### 3.3.3 2-(21,5,2) дизайните са уникално-разделими

За да докажем, че 2-(21,5,2) дизайните са уникално-разделими разглеждаме всички различни разделяния и показваме, че са еквивалентни. Всички разделяния, различни от тривиалното разделяне $\{D_1, D_2\}$, можем да представим във вида $\{D_1^a \cup D_2^b, D_2^a \cup D_1^b\}$, където съвкупностите от блокове $D_1^a$ и $D_1^b$ формират $D_1$, $D_2^a$ и $D_2^b$ – $D_2$, $D_1^a$ и $D_2^b$ – $D_3$ и $D_1^b$ и $D_2^a$ – $D_4$, а $D_1$, $D_2$, $D_3$ и $D_4$ са изоморфни помежду си:




Без загуба на общност приемаме, че $a$ частите имат най-малко толкова блокове, колкото съответните $b$ части и, че $D_1^b$ и $D_2^b$ нямат общи блокове (ако има такива, можем да ги преместим в $a$ частите).

**Твърдение 3.3.5.** *Нека $n$ е броят на блоковете в $D_1^b$ ($D_2^b$), а $\beta_i$ – броят на тези блокове от $D_1^b$ ($D_2^b$), които съдържат точката $i$ ($i = 1, 2, \ldots, v$). Тогава:*

*(a) Всяка точка е в един и същ брой блокове на $D_1^a$ и на $D_2^a$ ($D_1^b$ и $D_2^b$).*

*(b) Всяка двойка точки е в един и същ брой блокове на $D_1^a$ и $D_2^a$ ($D_1^b$ и $D_2^b$).*

*(c) Ако $D_1$ е проективна равнина от ред $q$ (т.е. 2-$(q^2+q+1, q+1, 1)$ design), е в сила:*

$$\beta_i \neq 1,\ i = 1, 2, \ldots, v \tag{3.10}$$

$$\sum_{i=1}^{v} \beta_i = n(q+1) \tag{3.11}$$

$$\sum_{i=1}^{v} \beta_i^2 = n(n+q) \tag{3.12}$$

$$n \leq \frac{q^2+q}{2} \tag{3.13}$$

**Доказателство.** (a) $D_1^b$ и $D_1^a$ (или $D_2^a$) образуват 2-$(v,k,\lambda)$ дизайн и поради това точка $i$ е в $r-\beta_i$ блока на $D_1^a$ ($D_2^a$).

(b) Нека двойката точки $(i,j)$ е в $\lambda_{ij}$ блока на $D_1^b$. Понеже $D_1^b$ и $D_1^a$ (или $D_2^a$) образуват 2-$(v,k,\lambda)$ дизайн, двойката точки $(i,j)$ е в $\lambda-\lambda_{ij}$ блока на $D_1^a$ ($D_2^a$).

(c) В 2-$(q^2+q+1,q+1,1)$ дизайн всеки два блока имат точно една обща точка. Разглеждаме произволна точка $i$. Ако отделим подмножество $S_q$ от $q$ блока, инцидентни с точката $i$ (всички такива блокове са $q+1$), то невлизащият в $S_q$ блок с точка $i$ е инцидентен с всичките $q$ точки, които не са в блокове от $S_q$. Следователно този блок е фиксиран с избора на останалите $q$ блока. Ако $\beta_i=1$, то $D_1^a(D_2^a)$ има $q$ блока, съдържащи точка $i$. Тогава блокът от $D_1^b$, съдържащ точката $i$, ще е същия като блока от $D_2^b$, съдържащ точка $i$, но ние приемаме, че $D_1^b$ и $D_2^b$ нямат общи блокове, затова е в сила (3.10).

(3.11) се получава с преброяване по два начина на единиците в матрицата на инцидентност на $D_1^b$ ($D_2^b$).

Всеки 2-$(q^2+q+1,q+1,1)$ дизайн е симетричен и от условието $\lambda=1$, приложено върху блоковете на $D_1^b$ ($D_2^b$) частта получаваме $\binom{n}{2}=\sum_{i=1}^{v}\binom{\beta_i}{2}$. Като използваме и (3.11) следва (3.12).

$D_1^b$ ($D_2^b$) има не повече блокове от $D_1^a(D_2^a)$. Затова $n\leq\lfloor v/2\rfloor$ и тогава (3.13) е в сила.

□

**Твърдение 3.3.6.** *Всеки разделим 2-(21,5,2) дизайн е уникално-разделим.*

**Доказателство.** За $q=4$, системата от равенства (3.10), (3.11), (3.12) и (3.13) има решения само за $n=6,8,9,10$. За всеки от тези случаи правим изчерпващо търсене за структури, удовлетворяващи (a) и (b) на Твърдение 3.3.5 по следния начин:

- Построяваме множеството от всички неекивалентни $D_1^b$.
- За всяко построено $D_1^b$, намираме множеството от всички неекивалентни $D_2^b$, такива, че $D_1^b$ и $D_2^b$ нямат общи блокове.
- За всяка конструирана двойка $D_1^b$ и $D_2^b$, намираме всички неекивалентни $a$-части ($D_1^a$ или $D_2^a$), и установяваме, че всички получени разделяния са еквивалентни.

При $n=6$ има единствено решение за стойностите на $\beta_i$. То е $(2_{15},0_6)$, т.е. 15 двойки и 6 нули. С точност до еквивалентност съществува единствен начин да изберем 6-тте блока от $D_1^b$, но за него няма съответен $D_2^b$. Следователно $n=6$ е невъзможно.

При $n=8$ единствено решение за $\beta_i$ е $(4_2,2_{16},0_3)$. С точност до еквивалентност съществува единствен начин да изберем 8-те блока на $D_1^b$ и единствен начин да

Фигура 3.7: Разделяне при $n = 8$

| $D_1^a$ | | | | $D_1^b$ | | $D_2^a$ | | | | $D_2^b$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1111 | .... | .... | .... | .... | 1 | .... | 1111 | .... | .... | .... |
| 1 | .... | 1111 | .... | .... | .... | 1 | .... | .... | 1111 | .... | .... |
| 1 | .... | .... | 1111 | .... | .... | 1 | 1111 | .... | .... | .... | .... |
| 1 | .... | .... | .... | 1111 | .... | 1 | .... | .... | .... | .... | 1111 |
| 1 | .... | .... | .... | .... | 1111 | 1 | .... | .... | .... | 1111 | .... |
| . | 1... | 1... | 1... | 1... | ...1 | . | 1... | 1... | 1... | 1... | ...1 |
| . | .1.. | .1.. | .1.. | .1.. | ...1 | . | .1.. | .1.. | .1.. | .1.. | ...1 |
| . | ..1. | ..1. | ..1. | ..1. | ...1 | . | ..1. | ..1. | ..1. | ..1. | ...1 |
| . | ...1 | ...1 | ...1 | ...1 | ...1 | . | ...1 | ...1 | ...1 | ...1 | ...1 |
| . | .1.. | ..1. | ...1 | 1... | ..1. | . | .1.. | ..1. | ...1 | 1... | ..1. |
| . | 1... | ...1 | ..1. | .1.. | ..1. | . | 1... | ...1 | ..1. | .1.. | ..1. |
| . | ...1 | 1... | .1.. | ..1. | ..1. | . | ...1 | 1... | .1.. | ..1. | ..1. |
| . | ..1. | .1.. | 1... | ...1 | ..1. | . | ..1. | .1.. | 1... | ...1 | ..1. |
| . | ..1. | ...1 | .1.. | 1... | .1.. | . | ..1. | ...1 | .1.. | 1... | .1.. |
| . | ...1 | ..1. | 1... | .1.. | .1.. | . | ...1 | ..1. | 1... | .1.. | .1.. |
| . | 1... | .1.. | ...1 | ..1. | .1.. | . | 1... | .1.. | ...1 | ..1. | .1.. |
| . | .1.. | 1... | ..1. | ...1 | .1.. | . | .1.. | 1... | ..1. | ...1 | .1.. |
| . | ...1 | .1.. | ..1. | 1... | 1... | . | ...1 | .1.. | ..1. | 1... | 1... |
| . | ..1. | 1... | ...1 | .1.. | 1... | . | ..1. | 1... | ...1 | .1.. | 1... |
| . | .1.. | ...1 | 1... | ..1. | 1... | . | .1.. | ...1 | 1... | ..1. | 1... |
| . | 1... | ..1. | .1.. | ...1 | 1... | . | 1... | ..1. | .1.. | ...1 | 1... |

построим $D_2^b$. За единствената комбинация $D_1^b$ и $D_2^b$ има 12 неекивалентни $D_1^a$ ($D_2^a$). За всички получени разделяния (едно от тях е показано на фигура 3.7), съществуват пермутации $\varphi_a$ и $\varphi_b$, за които:

- $D_2 = \varphi_b \varphi_a D_1$
- $D_2^b = \varphi_b D_1^b$, $D_1^a = \varphi_b D_1^a$, $D_2^a = \varphi_b D_2^a$, $(\varphi_b)^2 = 1$.
- $D_2^a = \varphi_a D_1^a$, $D_1^b = \varphi_a D_1^b$, $D_2^b = \varphi_a D_2^b$.

Разделянето $\{D_1^a \cup D_2^b, D_2^a \cup D_1^b\}$ е еквивалентно на разделянето $\{D_1, D_2\}$, защото $\varphi_b(D_1^a \cup D_2^b) = (D_1^a \cup D_1^b) = D_1$ и $\varphi_b(D_2^a \cup D_1^b) = (D_2^a \cup D_2^b) = D_2$.

Има 3 решения за $\beta_i$ ако $n = 9$:
$(5_1, 3_4, 2_{14}, 0_2)$, $(4_3, 3_1, 2_{15}, 0_2)$ и $(3_9, 2_9, 0_3)$. Само за третото от тях могат да бъдат избрани девет блока от 2-$(21, 5, 1)$ дизайн и то по начин, единствен с точност до еквивалентност. За него не съществува $D_2^b$.

Има 3 решения за $\beta_i$ при $n = 10$: $(5_1, 4_2, 3_3, 2_{14}, 0_1)$, $(4_5, 2_{15}, 0_1)$ и $(4_2, 3_8, 2_9, 0_2)$. С втория и третия може да бъде намерено $D_1^b$, но за него няма $D_2^b$.

Поради това всеки 2-$(21, 5, 2)$ дизайн е уникално-разделим. □

### 3.3.4 Алгоритъм за конструиране на двойни дизайни

**Цел на задачата.**

Дадени са два $t$-$(v, k, \lambda)$ дизайна $D_1$ и $D_2$. Трябва да построим всички неизоморфни $t$-$(v, k, 2\lambda)$ дизайни, които са разделими на изоморфни на $D_1$ и $D_2$. Всички такива дизайни (заедно с изоморфните) са $v!$ на брой и могат да бъдат получени като към блоковете на $D_1$ добавим тези на $\varphi D_2$, където $\varphi$ е пермутация на точките на $D_2$. Тогава задачата се свежда до конструирането на всички пермутации $\varphi$, които водят до неизоморфни дизайни.

**Конструктивни условия.**

Възможно е конкретната задача да изисква конструирането само на двойни дизайни от определен вид. Тогава налагаме някакви допълнителни условия за вида на пермутациите $\varphi$ и строим само такива, които отговарят на тях.

**Множество на търсене.**

Това са числата от 1 до $v$, които с търсене с връщане нареждаме на $v$ места по всевъзможните начини, удовлетворяващи конструктивните условия.

Предварително намираме всички автоморфизми на $D_1$ и $D_2$ (пълните групи означаваме с $G_1$ и $G_2$). След това генерираме в лексикографски ред всички пермутации, които отговарят на конструктивните условия. Изоморфните решения отхвърляме с теста за минималност.

**Тест за минималност.**

При намиране на текущата пермутация $\varphi$, проверяваме дали съществуват $\alpha \in G_2$ и $\beta \in G_1$, такива, че $\beta\varphi\alpha$ е пермутация, която е лексикографски по-малка от $\varphi$ и удовлетворява конструктивните условия. Ако намерим такива $\alpha$ и $\beta$, то понеже $\beta^{-1}(D_1 \cup \beta\varphi\alpha D_2) = (D_1 \cup \varphi D_2)$, отхвърляме текущото решение, защото е еквивалентно на лексикографски по-малко от него, което вече е генерирано.

Ако $D_1$ и $D_2$ са изоморфни дизайни проверяваме и за $\alpha \in G_2$ и $\beta \in G_1$, такива, че $\beta\varphi^{-1}\alpha$ е лексикографски по-малка от $\varphi$ (виж Твърдение 3.3.1) и удовлетворява конструктивните условия. В този случай също отхвърляме текущото решение.

Ако конструираните двойни дизайни са уникално-разделими, с прилагането на настоящия тест за минималност построяваме само неизоморфни дизайни.

Както обикновено, трябва внимателно да преценим дали конструктивните условия позволяват прилагането на теста върху частични решения. Примери за това са дадени в следващия раздел. Там са показани и начини да намалим броя на разглежданите пермутации $\beta\varphi\alpha$ и $\beta\varphi^{-1}\alpha$.

**Предимства.**

Въпреки, че алгоритъмът е експоненциален, прилагането на теста за минималност не върху получените дизайни, а върху пермутациите, които ги генерират, позволява успешното му използване в задачите от тази част.

**Сравнение с други алгоритми.**

Няколко години след публикуването на настоящите изследвания Матева [176] разработи подобен алгоритъм, но с допълнително отчитане на орбитите на стабилизаторите в $G_1$ и $G_2$ на точките от частичното решение, което позволява генерирането на по-малък брой пермутации $\varphi$.

### 3.3.5 Разделими 2-$(21,5,2)$ с автоморфизми от ред 2

**Автоморфизми, за които $|G_\varphi| \neq 1$**

Разглеждаме $(D_1 \cup D_2)$ с пълна група от автоморфизми от ред $2^s$ $(s \geq 1)$ и $|G_\varphi| \neq 1$. Тогава $D_1$ и $D_2$ имат общи автоморфизми от ред 2. Един 2-$(21,5,1)$ дизайн има автоморфизми от ред 2 с 5 фиксирани точки и автоморфизми от ред 2 със 7 фиксирани точки (фигури 3.8 и 3.9).

Ние построяваме всички двойни дизайни с автоморфизми от ред 2, които са автоморфизми едновременно на $D_1$ и $D_2$. Разглеждаме двата възможни случая:

**Автоморфизъм от ред 2 с 5 фиксирани точки:** Използваме матрицата на инцидентност на $D_1$, представена на фигура 3.8 и автоморфизъм $\gamma$, който действа на точките на двойния дизайн като
$(1)(2)\cdots(5)\,(6,7)(8,9)\cdots(20,21)$, а на блоковете като
$(1)(2)\cdots(5)\,(6,7)(8,9)\cdots(20,21)\,(22)(23)\cdots(26)\,(27,28)(29,30)\cdots(41,42)$.

**Автоморфизъм от ред 2 със 7 фиксирани точки:** Разглеждаме матрицата на инцидентност на $D_1$, представена на фигура 3.9 и автоморфизъм $\delta$, който действа на точките на двойния дизайн като
$(1)(2)\cdots(7)\,(8,9)(10,11)\cdots(20,21)$, а на блоковете като
$(1)(2)\cdots(7)\,(8,9)\cdots(20,21)\,(22)(23)\cdots(28)\,(29,30)\cdots(41,42)$.

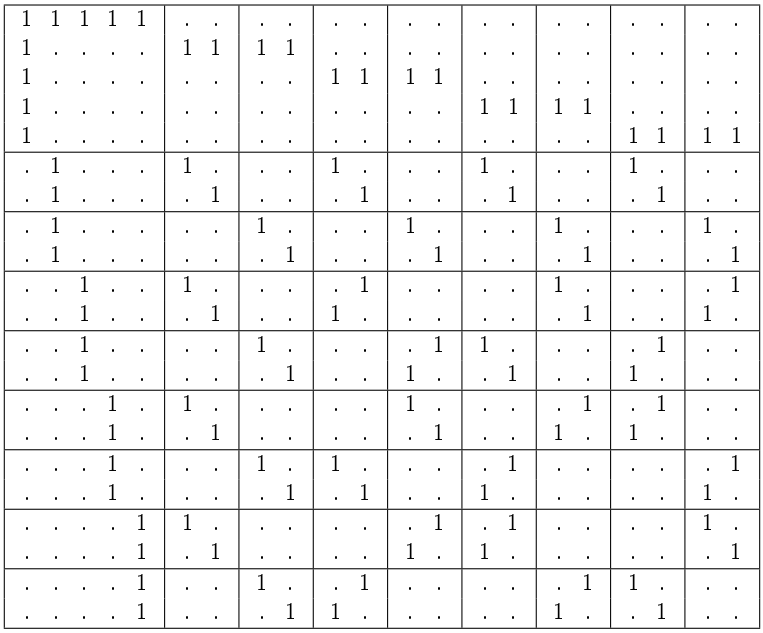
Нека $D_2 = \varphi D_1$. Тогава $\varphi$ е пермутация, която:

(a) изобразява всяка фиксирана (относно $\gamma$ или $\delta$) точка във фиксирана точка;

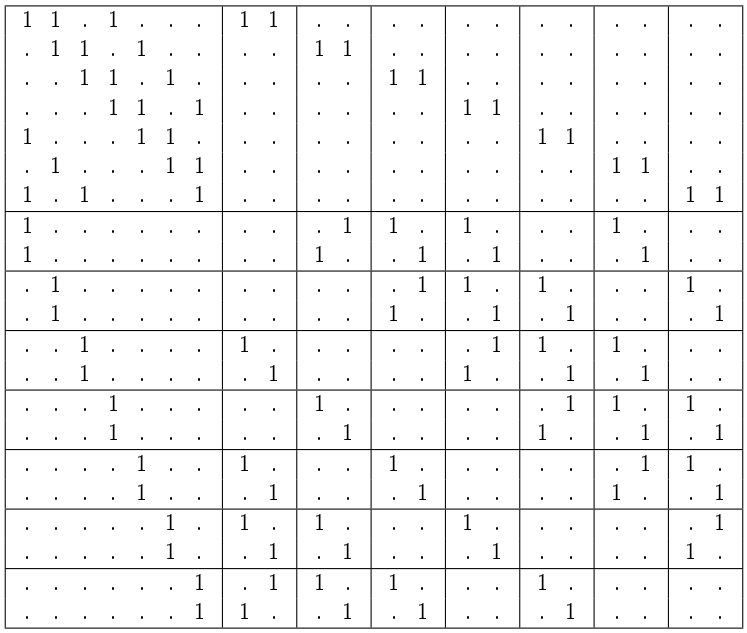
(b) изобразява две точки от една и съща орбита (относно $\gamma$ или $\delta$) в точки от една и съща орбита.

Дизайните конструираме с модификация на алгоритъма от раздел 3.3.4 като условията (a) и (b) са **конструктивните условия**. Те са такива, че позволяват прилагането на теста за минималност върху частични решения, т.е. ако текущо е намерена само част от пермутацията $\varphi$ (първите няколко точки), то непостроената още част не зависи от построената. Затова, ако $\beta\varphi\alpha$ или $\beta\varphi^{-1}\alpha$ изобразява

Фигура 3.8: Дизайнът $D_1$ (автоморфизъм от ред 2 с 5 фиксирани точки)



Фигура 3.9: Дизайнът $D_1$ (автоморфизъм от ред 2 със 7 фиксирани точки)



построената част в лексикографски по-малка, можем да отхвърлим частичното решение. Това значително ускорява пресмятанията.

**Тест за минималност.** Понеже редът на групата от автоморфизми $G$ на 2-$(21, 5, 1)$ дизайн е 120960, проверката на всички $120960^2$ комбинации е твърде бавна. Затова избираме $\alpha$ и $\beta$ измежду случайно подбрана извадка от елементи на $G$. Накрая отсяваме изоморфните решения (които са много малко) с пълен тест за изоморфизъм.

По този начин построяваме 9564 неизоморфни двойни с автоморфизъм от ред 2 с 5 фиксирани точки и 31094 с автоморфизъм от ред 2 със 7 фиксирани точки. Общият брой на дизайните е 40485, защото 173 имат едновременно автоморфизми от ред 2 с 5 и със 7 фиксирани точки. Между тях 305 дизайна притежават и автоморфизъм от нечетен прост ред.

Резултатите са паралелно получени от съавторите с алгоритъм, при който първо се получават всички нееквивалентни решения за нетривиалните орбити, а след това се добавят по всевъзможните начини фиксираните точки и блокове и отново се отсяват изоморфните решения.

**Автоморфизми от ред 2 с $|G_\varphi| = 1$**

Трябва да построим всички дизайни $(D \cup \varphi D)$, за които $\varphi$ е пермутация от ред 2 (Твърдение 3.3.2). Нека $\varphi = \{\varphi_1, \varphi_2, \ldots \varphi_v\}$. Групата от автоморфизми на проективната равнина от ред 4 е двойно транзитивна, затова без загуба на общност фиксираме една нетривиална орбита, т.е. за всички построени пермутации $\varphi_1 = 2$ и $\varphi_2 = 1$.

Използваме модификация на алгоритъма от предния раздел 3.3.4. **Конструктивните условия** изискват:

- $\varphi_1 = 2$, $\varphi_2 = 1$
- Ако $\varphi_a = b$, то $\varphi_b = a$ за всички $a, b = 1, 2, \ldots v$ (пермутацията е от ред 2).

**Тест за минималност.** Ако $\varphi$ е от ред 2, то $\varphi = \varphi^{-1}$ и $\beta\varphi\alpha$ също е от ред 2, т.е.

$$\beta\varphi\alpha\beta\varphi\alpha = 1 \Rightarrow \varphi\alpha\beta\varphi = \beta^{-1}\alpha^{-1} \Rightarrow \varphi\alpha\beta\varphi \in G.$$

Но $\varphi^{-1}\alpha\beta\varphi$ е също автоморфизъм на $\varphi D$. Следователно $\varphi\alpha\beta\varphi \in G_\varphi$. If $|G_\varphi| = 1$, то $\alpha\beta = 1 \Rightarrow \beta = \alpha^{-1}$. Понеже $|G_\varphi| = 1$ за повечето дизайни, конструирани по този начин, то в теста за минималност вместо $\beta\varphi\alpha$ търсим $\alpha^{-1}\varphi\alpha$, която е лексикографски по-малка от $\varphi$ и ако намерим такава, отхвърляме текущото решение. Задачата става решима благодарение на бързината на този тест за минималност. Приложеният накрая пълен тест за изоморфизъм отхвърля много малка част от намерените дизайни.

Конструктивните условия, обаче, са такива, че тест за минималност, на практика, не можем да прилагаме на частични решения, т.е. можем само, ако когато

сме построили $\{\varphi_1, \varphi_2, \ldots \varphi_w\}$ за някакво $w < v$, то $\varphi_a \leq w$ за всяко $a \leq w$ и намерим $\alpha$, за която $(\alpha^{-1}\varphi\alpha)_a \leq w$ за всяко $a \leq w$.

Построяваме 991957 неизоморфни дизайна с автоморфизъм от ред 2, който изобразява единия съставен дизайн в другия. За 984549 от тях редът на пълната група от автоморфизми е 2 и $|G_\varphi| = 1$. Резултатите са проверени с две различни програмни реализации (на автора и на съавтор, който използва софтуера *nauty* [187] на McKay за финалния пълен тест за изоморфизъм).

### 3.3.6 Класификация на двойните на проективната равнина от ред 4

Според следствие 3.3.4 броят на неизоморфните двойни на проективната равнина от ред 4 е най-малко 1745944200. За да определим точния им брой с равенства (3.8) и (3.9), трябва да построим всички дизайни с нетривиални автоморфизми.

По-точно, това означава, че трябва да построим двойните дизайни,
– за които $|G_\varphi| \neq 1$ и да определим $N_i$ и $N_i'$, $i > 1$, и
– за които $|G_\varphi| = 1$ и $G\varphi G = G\varphi^{-1}G$ и да намерим броя $N_1'$.

Всички 2-$(21, 5, 2)$ дизайни с автоморфизми от нечетен прост ред са конструирани в [13]. Сред тях има 4170 разделими, които използваме за настоящата класификация. Останалите дизайни имат автоморфизми от ред 2 и са построени в раздел 3.3.5.

Класификационните резултати са дадени в Таблица 3.3. Редът на пълната група от автоморфизми на двойните дизайни е даден в колона $|\widehat{G}_\varphi|$, редът на общата подгрупа на пълните групи от автоморфизми на $D$ и $\varphi D$ – в колона $|G_\varphi|$), а дали $G\varphi G = G\varphi^{-1}G$ показва третата колона от таблицата. В колона $N^{(\prime)}_{|G_\varphi|}$ е броят на неизоморфните двойни с дадените свойства. Броят[1] на дизайните в класа на изоморфизъм на двойните дизайни (определен според Твърдение 3.3.1) е представен в колона $|\mathcal{C}_G(\varphi)|/|G|$. Произведението на стойностите в колони $N^{(\prime)}_{|G_\varphi|}$ и $|\mathcal{C}_G(\varphi)|/|G|$ е записано в последната колона $N^{(\prime)}_{|G_\varphi|} \times |\mathcal{C}_G(\varphi)|/|G|$.

Разполагайки с класификацията на всичките 1028899 двойни с нетривиални автоморфизми, ние използваме (3.8) и (3.9) за да намерим броя на неизоморфните двойни дизайни с тривиалния автоморфизъм. Този брой е 1745432408. Затова първият (сив) ред на Таблица 3.3 е получен от данните в останалите редове. Броят на всички двойни 2-$(21, 5, 2)$ дизайни е 1746461307 и е доста близък до долната граница според Следствие 3.3.4.

Резултатите от настоящата част 3.3 са получени съвместно с Veerle Fack, Joost Winne и Росен Златарски и са публикувани в [T4].

[1] За да избегнем твърде големите числа, стойностите в последните две колони са разделени на $|G| = 120960$, т.е. на реда на пълната група от автоморфизми на проективната равнина от ред 4.

**Таблица 3.3: Класификация на двойните на проективната равнина от ред 4**

| $\lvert\widehat{G}_\varphi\rvert$ | $\lvert G_\varphi\rvert$ | $G\varphi G \stackrel{?}{=} G\varphi^{-1}G$ | $N^{(')}_{\lvert G_\varphi\rvert}$ | $\frac{\lvert\mathcal{C}_G(\varphi)\rvert}{\lvert G\rvert}$ | $N^{(')}_{\lvert G_\varphi\rvert} \times \frac{\lvert\mathcal{C}_G(\varphi)\rvert}{\lvert G\rvert}$ |
|---|---|---|---|---|---|
| 1 | 1 | не | 1 745 432 408 | 241 920 | 422 255 008 143 360 |
| 2 | 1 | да | 984 549 | 120 960 | 119 091 047 040 |
| 2 | 2 | не | 33 631 | 120 960 | 4 068 005 760 |
| 3 | 3 | не | 2 764 | 80 640 | 222 888 960 |
| 4 | 2 | да | 5 709 | 60 480 | 345 280 320 |
| 4 | 4 | не | 389 | 60 480 | 23 526 720 |
| 5 | 5 | не | 26 | 48 384 | 1 257 984 |
| 6 | 3 | да | 1 019 | 40 320 | 41 086 080 |
| 6 | 6 | не | 67 | 40 320 | 2 701 440 |
| 8 | 4 | да | 345 | 30 240 | 10 432 800 |
| 8 | 8 | не | 17 | 30 240 | 514 080 |
| 9 | 9 | не | 1 | 26 880 | 26 880 |
| 10 | 5 | да | 30 | 24 192 | 725 760 |
| 12 | 6 | да | 167 | 20 160 | 3 366 720 |
| 12 | 12 | не | 2 | 20 160 | 40 320 |
| 14 | 7 | да | 2 | 17 280 | 34 560 |
| 14 | 14 | не | 1 | 17 280 | 17 280 |
| 16 | 8 | да | 55 | 15 120 | 831 600 |
| 16 | 16 | не | 3 | 15 120 | 45 360 |
| 18 | 9 | да | 18 | 13 440 | 241 920 |
| 18 | 18 | не | 1 | 13 440 | 13 440 |
| 21 | 21 | не | 1 | 11 520 | 11 520 |
| 24 | 12 | да | 24 | 10 080 | 241 920 |
| 28 | 14 | да | 2 | 8 640 | 17 280 |
| 30 | 15 | да | 1 | 8 064 | 8 064 |
| 32 | 16 | да | 20 | 7 560 | 151 200 |
| 32 | 32 | не | 1 | 7 560 | 7 560 |
| 36 | 18 | да | 15 | 6 720 | 100 800 |
| 40 | 20 | да | 1 | 6 048 | 6 048 |
| 42 | 21 | да | 2 | 5 760 | 11 520 |
| 48 | 24 | да | 3 | 5 040 | 15 120 |
| 54 | 27 | да | 1 | 4 480 | 4 480 |
| 64 | 32 | да | 7 | 3 780 | 26 460 |
| 96 | 48 | да | 4 | 2 520 | 10 080 |
| 96 | 96 | не | 1 | 2 520 | 2 520 |
| 108 | 54 | да | 2 | 2 240 | 4 480 |
| 120 | 60 | да | 1 | 2 016 | 2 016 |
| 128 | 64 | да | 2 | 1890 | 3 780 |
| 192 | 96 | да | 4 | 1 260 | 5 040 |
| 252 | 126 | да | 1 | 960 | 960 |
| 256 | 128 | да | 1 | 945 | 945 |
| 384 | 96 | да | 1 | 1 260 | 1 260 |
| 384 | 192 | да | 1 | 630 | 630 |
| 480 | 240 | да | 1 | 504 | 504 |
| 576 | 288 | да | 1 | 420 | 420 |
| 1152 | 288 | да | 1 | 420 | 420 |
| 1152 | 576 | да | 1 | 210 | 210 |
| 1536 | 384 | да | 1 | 315 | 315 |
| 3840 | 1920 | да | 1 | 63 | 63 |
| 120960 | 120960 | да | 1 | 1 | 1 |
| $\lvert\widehat{G}_\varphi\rvert$ | $\lvert G_\varphi\rvert$ | $G\varphi G \stackrel{?}{=} G\varphi^{-1}G$ | $N^{(')}_{\lvert G_\varphi\rvert}$ | $\frac{\lvert\mathcal{C}_G(\varphi)\rvert}{\lvert G\rvert}$ | $N^{(')}_{\lvert G_\varphi\rvert} \times \frac{\lvert\mathcal{C}_G(\varphi)\rvert}{\lvert G\rvert}$ |
| Всички | | | 1 746 461 307 | | $21!/\lvert G\rvert =$ 422 378 820 864 000 |

# Глава 4

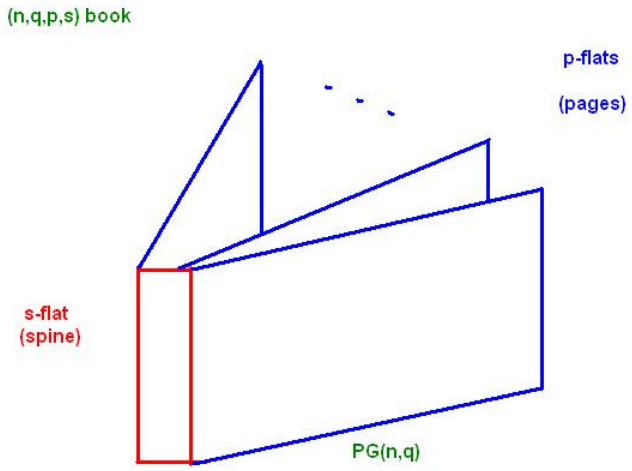
# Спредове на $PG(n,2)$

## 4.1 Дефиниции, означения и предварителни резултати

Нека отначало да припомним (дефиниции 1.1.22 - 1.1.24), че *t-спредът* в $PG(n,q)$ представлява разбиване на точките на проективното пространство на $t$-мерни подпространства и, че два спреда са еквивалентни, ако съществува автоморфизъм на проективното пространство, който изобразява единия в другия. Обикновено 1-спредовете наричаме спредове или спредове от прави.

**Дефиниция 4.1.1.** *Един спред $S$ в $PG(n,2)$ е* ***регулярен (геометричен)****, ако за всеки три прави $a,b,c \in S$: или $c \in \langle a,b\rangle$, или $c$ не съдържа точки от $\langle a,b\rangle$. Както обикновено, с $\langle a,b\rangle$ означаваме най-малкото подпространство, съдържащо правите $a$ и $b$.*

**Дефиниция 4.1.2.** *Под* $(\boldsymbol{n,q,r,s})$ ***книга*** *разбираме съвкупност от $r$-мерни подпространства в $PG(n,q)$, наречени* ***страници****, които покриват цялото проективно пространство и се пресичат в общо $s$-мерно подпространство, наречено* ***гръбнак*** *и такова, че всяка точка извън гръбнака е в точно една страница.*



**Дефиниция 4.1.3.** *Един $t$-спред в $PG(n,q)$ е* ***$(n,q,r,s)$ книжен $t$-спред****, ако точките от всяка страница на $(n,q,r,s)$ книга и точките от нейния гръбнак се разбиват на $t$-мерни подпространства от този $t$-спред.*

Възможно е да съществува $t$-спред, който е книжен спред за различни базисни $(n,q,r,s)$ книги.

**Дефиниция 4.1.4.** *Един $(n,q,r,s)$ книжен $t$-спред е* ***цикличен*** *върху страниците, ако има автоморфизъм, който пермутира страниците в един цикъл.*

Спредовете в проективни пространства представляват интерес и заради приложенията си в други области. В криптографията ги свързват с линейни шифри [202], а в Теория на кодирането с кодове с константна размерност и спред кодове [83], [84], [173].

Спредовете са обект на множество статии. Общи конструкции са представени в [17] и [131]. Максималната големина на нетривиални максимални частични спредове е обект на [25], [44], [45], [90], [111], [112], [113]. Известни са и немалко класификационни резултати за спредове. Van Dam класифицира спредове в $PG(3,4)\setminus PG(3,2)$ [243], а Gordon, Shaw и Soicher – частичните спредове в $PG(4,2)$ [103]. Jha и Johnson описват групите от автоморфизми на спредове в $PG(3,q)$ с линейни групи от ред $q(q+1)$ [123], [124], а Johnson на тези с няколко групи от ред $q+1$ [130].

Останалите класификации са компютърни. С помощта на пакета GRAPE от GAP [96] Soicher [217] прави конструктивна класификация с точност до еквивалентност на всички частични спредове в $PG(3,2)$, $PG(4,2)$ и $PG(3,3)$, на всички максимални частични спредове в $PG(3,4)$ и на максималните частични спредове в $PG(3,7)$ с 45 елемента, които са инвариантни относно група от ред 5. По-късно Blokhuis, Brouwer и Wilbrink класифицират всички максимални частични спредове в $PG(3,7)$ с 45 елемента [33].

В следващите раздели класифицираме всички спредове в $PG(5,2)$, всички книжни спредове в $PG(5,2)$ и някои видове книжни спредове в $PG(7,2)$ с модификации на алгоритъма, описан в раздел 4.2.

## 4.2 Алгоритъм за конструиране на всички (с точност до проективна еквивалентност) спредове на $PG(n,2)$

Нека $(l_1,l_2)$ и $(l_1',l_2')$ са две произволни двойки непресичащи се прави от $PG(n,2)$. Тогава съществува автоморфизъм $\alpha$ на проективното пространство, който изобразява едната двойка в другата. Това ни дава право да фиксираме две от правите на спредовете, които конструираме. Без загуба на общност ще считаме, че конструираните спредове съдържат правите $l_1$ и $l_2$.

### 4.2.1 Множество на търсене

То съдържа всички прави от $PG(n,2)$, които нямат общи точки с $l_1$ и $l_2$. Подреждаме ги в лексикографски ред и за всяка от тях намираме и запазваме по един автоморфизъм, който я изобразява в $l_1$ и един, който я изобразява в $l_2$, като фиксира $l_1$.

### 4.2.2 Конструиране на спреда

Строим го с добавяне на правите една по една с изчерпващо търсене с връщане (раздел 1.2.2). Ако към текущия частичен спред сме добавили $m$ прави, то $m+1$-вата избираме между правите от множеството на търсене, които:

- съдържат най-малката точка, която не е в никоя от вече избраните прави;
- не съдържат точки от вече избраните прави.

### 4.2.3 Ефективност

Алгоритъмът е експоненциален. Реалната му ефективност и приложимост при задачите от настоящата глава се дължи на сравнително бързия тест за минималност на частичните решения.

### 4.2.4 Тест за минималност на частичните решения

Да означим с $N$ броя на правите от спреда, а с $N_p$ – броя на правите от частичното решение. Проверяваме дали някой от автоморфизмите на проективното пространство не трансформира частичното решение в лексикографски по-малко. Понеже броят на всички автоморфизми е твърде голям, използваме наличието на фиксирани елементи.

Нека $S_1$ и $S_2$ са два еквивалентни спреда от конструираните с 2 фиксирани прави $l_1$ и $l_2$ и нека $\alpha \in G$ изобразява правите от $S_1$ в прави от $S_2$. Нека $\alpha l_i = l_1$ и $\alpha l_j = l_2$. Нека има и друг автоморфизъм $\beta \in G$, който изобразява тези прави по същия начин, т.е. $\beta l_i = l_1$ и $\beta l_j = l_2$. Съществува $\varphi \in G$, такъв, че $\alpha = \varphi\beta$. Тогава $\varphi = \alpha\beta^{-1}$, което означава, че $\varphi$ фиксира правите $l_1$ и $l_2$. Следва, че всеки автоморфизъм, който изобразява правите от $S_1$ в прави от $S_2$, може да се представи като произведение на $\alpha$ и някой от автоморфизмите, които фиксират правите $l_1$ и $l_2$.

Проверяваме само с част от автоморфизмите на проективното пространство, т.е с автоморфизмите $\varphi_l\beta_{l_j}\alpha_{l_i}$, където:

- $\alpha_{l_i}$ изобразява правата $l_i$ в $l_1$,
- $\beta_{l_j}$ фиксира $l_1$ и изобразява правата $l_j$ в $l_2$,
- $\varphi_l$ е автоморфизъм, фиксиращ правите $l_1$ и $l_2$.

Тук $l_i$ и $l_j$ са прави от спреда, чиито брой е съответно $N_p$ и $N_p - 1$. Нека $a$ е реда на групата от автоморфизми, фиксираща правите $l_1$ и $l_2$. Тогава $l = 1, 2, ..., a$.

Следователно в проверката за минималност използваме най-много $N(N-1)a$ от автоморфизмите на проективното пространство и това прави класификацията възможна.

Изчистен от подробностите, тестът за минималност изпълнява следното:

for( i=1; i<=$N_p$; i++) // $N_p$ прави в частичното решение
{
    construct $\pi_1 = \alpha_{l_i}$
    for( j=1; j<=$N_p$; j++) if($\pi_1 l_j > l_1$) // освен първата има още $N_p - 1$ прави в частичното решение
    {
        construct $\pi_2 = \beta_{\pi_1 l_j} \pi_1$
        for( l=1; l<=a; l++)
        {
            construct $\pi = \varphi_l \pi_2$
            if Smaller($\pi$) return false;
        }
    }
}
return true;

Функцията Smaller($\pi$) дава положителен отговор, ако автоморфизмът $\pi$ изобразява текущото решение в лексикографски по-малко. Когато прилагаме теста върху цели решения, променяме малко функцията Smaller, така, че да можем да преброим случаите, в които $\pi$ изобразява спреда в себе си и по този начин в процеса на конструиране да определим и реда на пълната група от автоморфизми.

### 4.2.5 Модификации

Ако размерността на проективното пространство е по-голяма от 3, или строим само спредове с някакви допълнителни свойства, могат да бъдат фиксирани повече от две прави от спреда, което води до конструирането на по-малък брой решения и до още по-бърз тест за минималност. Точно такива са случаите, разгледани в следващите три раздела - в раздел 4.3 фиксираните прави са 3, в 4.4 – 6, а в 4.5 – 7. Във всеки от тези случаи, обаче, е необходимо предварително да намерим и запазим автоморфизми, които изобразяват правите от множеството на търсене във фиксираната част, като вземем предвид подпространствата, дефинирани от фиксираните прави. Подробностите са дадени в съответните раздели.

## 4.3 Класификация на всички спредове на $PG(5,2)$

### 4.3.1 Въведение

С точност до еквивалентност в $PG(3,2)$ съществува един единствен спред. Преди настоящата работа $PG(5,2)$ беше най-малкият отворен случай за пълна класификация на спредове в $PG(n,2)$. Търсейки афинни 2-$(64,16,5)$ дизайни с минимален ранг, Mavron, McDonough и Tonchev конструират [185] повече от 30000 спреда на $PG(5,2)$ (без да отделят еквивалентните) и ги класифицират според ранга на 2-(64,16,5) дизайна, получен от всеки от тях чрез конструкцията на Rahilly [205].

Тук ние построяваме всички 131044 неекивалентни спреда и представяме резултати за групите от автоморфизми, които ги запазват, за броя на правите от спреда, съдържащи се в тримерните подпространства и за ранга на 2-(64,16,5) дизайна, получен по Rahilly [205]. Резултатите показват, че въпреки, че не са конструирали всички спредове, авторите на [185] са намерили представители със съответни 2-(64,16,5) дизайни от всички възможни рангове.

### 4.3.2 Особености на групата от автоморфизми на $PG(5,2)$ и използването им в класификационния алгоритъм

В $PG(5,2)$ има 63 точки и 651 прави. Всяко $g$-мерно подпространство ($g = 0,1,2,3,4$) има $2^{g+1}-1$ точки и $(2^{2g+1}+1)/3-2^g$ прави. Инцидентността на точките и правите съответства на $STS(63)$, а на точките и хиперравнините – на Адамаров 2-(63,31,15) дизайн. Всеки две непресичащи се прави $A$ и $B$ на $PG(5,2)$ задават 3-мерно подпространство $\langle A,B\rangle$. Пълната група от автоморфизми $G$ на $PG(5,2)$ е изоморфна на $GL(6,2)$, която е двойно-транзитивна върху точките. Всеки нейн автоморфизъм фиксира или 0 точки, или всички точки на някое подпространство. За всяко подпространство $P$ съществува подгрупа, която фиксира всичките му точки и е транзитивна върху $g$-мерните подпространства, които нямат общи точки с $P$. Понеже едно подпространство има $1,3,7,15$, или 31 точки, редът на пълната група от автоморфизми е:

$|PGL(6,2)| = 63(63-1)(63-3)(63-7)(63-15)(63-31) = 20158709760.$

Правата, която е инцидентна с точки $a,b$ и $c$, означаваме с $\{a,b,c\}$. Без загуба на общност номерираме точките на $PG(5,2)$ с числата $1,2,\ldots,63$ така, че $\{1,2,3\},\{4,5,6\},\ldots,\{61,62,63\}$ да е спред и точковото множество $\{1,2,3,\ldots,15\}$ да е 3-мерно подпространство $P_1$.

За всеки две прави от спреда съществува автоморфизъм, който ги изобразява в правите $\{1,2,3\}$ и $\{4,5,6\}$ и за всяко 3-мерно подпространство съществува автоморфизъм, който го изобразява в подпространството $P_1 = \langle\{1,2,3\},\{4,5,6\}\rangle$.

За всяка права $X$, която е извън $P_1$, има автоморфизъм, който фиксира точките на $P_1$ и изобразява $X$ в $\{16,17,18\}$. Следователно, за всяко 3-мерно подпрос-

транство $P$, съдържащо точки 1 и 2 и различно от $P_1$, съществува автоморфизъм, който фиксира точките на $P_1$ и изобразява $P$ в 3-мерното подпространство $P_2 = \langle\{1,2,3\},\{16,17,18\}\rangle$.

Всеки спред има прави извън $P_1$ и, следователно, съществува автоморфизъм, който го изобразява в спред, съдържащ правите $\{1,2,3\}$, $\{4,5,6\}$, и $\{16,17,18\}$. Затова ние построяваме само спредове, съдържащи тези три прави.

За конструиране на спредовете използваме модификация на алгоритъма от раздел 4.2, при която фиксираните прави са с една повече, така, че допълнително за всяка права извън $P_1$ намираме и запазваме и автоморфизъм, който фиксира $P_1$ и изобразява тази права в $\{16,17,18\}$. Теста за минималност прилагаме след намиране на пълно решение. Той проверява дали някой от автоморфизмите $\varphi_l\eta_{l_m}\beta_{l_j}\alpha_{l_i}$ изобразява текущото решение в лексикографски по-малко, където:

- $\alpha_{l_i}$ изобразява правата $l_i$ в $\{1,2,3\}$,
- $\beta_{l_j}$ фиксира $\{1,2,3\}$ и изобразява правата $l_j$ в $\{4,5,6\}$,
- $\eta_{l_m}$ фиксира $P_1$ и изобразява правата $l_m$ в $\{16,17,18\}$,
- $\varphi_l$ е автоморфизъм, фиксиращ правите $\{1,2,3\}$, $\{4,5,6\}$ и $\{16,17,18\}$.

Тук $l_i$, $l_j$ и $l_m$ са прави от спреда, чиито брой е съответно 21, 20 и 16. Групата от автоморфизми, която запазва трите прави $\{1,2,3\}$, $\{4,5,6\}$ и $\{16,17,18\}$ е от ред 216 и поради това $l = 1,2,...,216$. По този начин вместо всичките 20158709760 автоморфизми, в теста за минималност използваме $1451520 = 21.20.16.216$ от тях, което значително го ускорява и прави класификацията възможна.

Тестът за минималност от раздел 4.2.4 придобива вида:

```
for( i=1; i<=21; i++) // 21 прави в спреда
{
    construct π1 = α_{l_i}
    for( j=1; j<=21; j++) if(π1 l_j > l_1) // освен първата има още 20 прави от спреда
    {
        construct π2 = β_{π1 l_j} π1
        for( m=1; m<=21; m++) if(π2 l_m ∉ ⟨l_1, l_2⟩) // 16 прави от спреда са извън ⟨l_1, l_2⟩
        {
            construct π3 = η_{π2 l_m} π2
            for( l=1; l<=216; l++)
            {
                construct π = φ_l π3
                if Smaller(π) return false;
            }
        }
    }
}
return true;
```

### 4.3.3 Класификация - модели по подпространства

Всичките 131044 спреда и 2-(63,31,15) дизайна от точките и хиперравнините могат да бъдат изтеглени от http://www.moi.math.bas.bg/~svetlana. Редовете на групите от автоморфизми, които ги запазват, са представени в Таблица 4.1.

**Таблица 4.1: Ред на групите от автоморфизми на спредовете**

| ред | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 12 | 15 | 16 | 18 | 21 | 24 | 32 | 36 | 42 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| спредове | 128474 | 2108 | 173 | 84 | 17 | 96 | 3 | 17 | 4 | 1 | 19 | 1 | 3 | 7 | 1 | 7 | 3 | 5 | 1 |

| ред | 48 | 64 | 72 | 96 | 108 | 192 | 288 | 324 | 384 | 480 | 576 | 1152 | 1728 | 5760 | 362880 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| спредове | 3 | 1 | 1 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

За всеки спред намираме 2-ранга на съответстващия му чрез конструкцията на Rahilly [205] афинен 2-(64,16,5) дизайн. Само два спреда имат минималния ранг 16. Те са конструирани и в [185]. Резултатите за ранга са дадени в Таблица 4.2.

**Таблица 4.2: Ранг на съответните 2-(64,16,5) дизайни**

| ранг | 16 | 17 | 18 | 19 | 20 | 21 | 22 | all |
|---|---|---|---|---|---|---|---|---|
| дизайни | 2 | 3 | 14 | 29 | 192 | 6101 | 124703 | 131044 |

С точност до еквивалентност има точно един регулярен спред. Означаваме с $n_i$ броя на тримерните подпространства, съдържащи $i$ прави от спреда, $i = 0, 1, 2, 3, 4, 5$. Има 238 различни разпределения ($n_2$, $n_3$, $n_4$, $n_5$) на правите от спреда в тримерните подпространства. Ще ги наричаме *модели*. Стойностите на $n_1$ и $n_0$ не са включени, защото могат да бъдат пресметнати по формулите: $n_1 = 735 - 5n_5 - 4n_4 - 3n_3 - 2n_2$ и $n_0 = 651 - n_5 - n_4 - n_3 - n_2 - n_1$. Моделът на регулярния спред е $(0, 0, 0, 21)$, а моделът на 6-те нееквивалентни спреда с най-много две прави в едно и също 3-мерно подпространство е $(6, 210, 0, 0)$.

Всички модели са представени в Таблици 4.3 и 4.4, където $N$ е броят на нееквивалентните спредове с дадения модел, $auts$ са редовете на групите от автоморфизми, а $rs$ са 2-ранговете на 2-(64,16,5) дизайните.

Изследванията в тази част са съвместни със Златка Матева и са публикувани в [T7].

**Таблица 4.3: Модели по подпространства, $N \geq 12$**

| $N$ | $n_2$ | $n_3$ | $n_4$ | $n_5$ | $rs$ | $auts$ |
|---|---|---|---|---|---|---|
| 12194 | 168 | 14 | 0 | 0 | 20 21 22 | 1 2 7 |
| 11707 | 171 | 13 | 0 | 0 | 20 21 22 | 1 2 |
| 11342 | 165 | 15 | 0 | 0 | 20 21 22 | 1 2 3 5 6 |
| 10563 | 174 | 12 | 0 | 0 | 20 21 22 | 1 2 3 6 |
| 9663 | 162 | 16 | 0 | 0 | 19 20 21 22 | 1 2 6 |
| 8613 | 177 | 11 | 0 | 0 | 21 22 | 1 2 |
| 7633 | 159 | 17 | 0 | 0 | 20 21 22 | 1 2 |
| 6834 | 180 | 10 | 0 | 0 | 20 21 22 | 1 2 5 |
| 5427 | 156 | 18 | 0 | 0 | 20 21 22 | 1 2 3 6 |
| 4768 | 183 | 9 | 0 | 0 | 20 21 22 | 1 2 3 |
| 3598 | 153 | 19 | 0 | 0 | 20 21 22 | 1 2 |
| 3135 | 186 | 8 | 0 | 0 | 20 21 22 | 1 2 |
| 2920 | 159 | 15 | 1 | 0 | 20 21 22 | 1 2 3 6 |
| 2848 | 156 | 16 | 1 | 0 | 20 21 22 | 1 2 4 |
| 2645 | 162 | 14 | 1 | 0 | 20 21 22 | 1 2 3 4 6 |
| 2465 | 153 | 17 | 1 | 0 | 20 21 22 | 1 2 3 |
| 2182 | 165 | 13 | 1 | 0 | 20 21 22 | 1 2 |
| 2034 | 150 | 20 | 0 | 0 | 20 21 22 | 1 2 5 |
| 1884 | 150 | 18 | 1 | 0 | 20 21 22 | 1 2 3 4 6 |
| 1861 | 189 | 7 | 0 | 0 | 20 21 22 | 1 2 7 |
| 1792 | 168 | 12 | 1 | 0 | 20 21 22 | 1 2 3 4 6 |
| 1185 | 147 | 19 | 1 | 0 | 20 21 22 | 1 2 |
| 1109 | 171 | 11 | 1 | 0 | 20 21 22 | 1 2 3 |
| 1040 | 192 | 6 | 0 | 0 | 20 21 22 | 1 2 3 6 9 |
| 1016 | 147 | 21 | 0 | 0 | 20 21 22 | 1 2 3 7 |
| 779 | 174 | 10 | 1 | 0 | 20 21 22 | 1 2 |
| 733 | 144 | 20 | 1 | 0 | 20 21 22 | 1 2 3 4 6 |
| 603 | 150 | 16 | 2 | 0 | 20 21 22 | 1 2 3 4 |
| 558 | 147 | 17 | 2 | 0 | 20 21 22 | 1 2 3 |
| 515 | 144 | 22 | 0 | 0 | 20 21 22 | 1 2 |
| 492 | 195 | 5 | 0 | 0 | 21 22 | 1 2 5 15 |
| 489 | 153 | 15 | 2 | 0 | 20 21 22 | 1 2 3 |
| 468 | 144 | 18 | 2 | 0 | 21 22 | 1 2 3 6 |
| 427 | 177 | 9 | 1 | 0 | 21 22 | 1 2 3 |
| 406 | 156 | 14 | 2 | 0 | 21 22 | 1 2 3 4 |
| 394 | 141 | 21 | 1 | 0 | 20 21 22 | 1 2 3 |
| 308 | 141 | 19 | 2 | 0 | 20 21 22 | 1 2 3 |
| 277 | 180 | 8 | 1 | 0 | 20 21 22 | 1 2 3 |
| 269 | 159 | 13 | 2 | 0 | 20 21 22 | 1 2 3 |
| 215 | 138 | 20 | 2 | 0 | 20 21 22 | 1 2 3 4 |
| 207 | 198 | 4 | 0 | 0 | 21 22 | 1 2 |
| 207 | 138 | 22 | 1 | 0 | 20 21 22 | 1 2 4 |
| 187 | 141 | 23 | 0 | 0 | 21 22 | 1 2 |
| 159 | 162 | 12 | 2 | 0 | 21 22 | 1 2 3 |
| 143 | 144 | 16 | 3 | 0 | 20 21 22 | 1 2 3 4 6 |
| 137 | 183 | 7 | 1 | 0 | 20 21 22 | 1 2 |
| 124 | 138 | 18 | 3 | 0 | 19 21 22 | 1 2 4 6 |

| $N$ | $n_2$ | $n_3$ | $n_4$ | $n_5$ | $rs$ | $auts$ |
|---|---|---|---|---|---|---|
| 120 | 141 | 17 | 3 | 0 | 21 22 | 1 2 3 |
| 111 | 147 | 15 | 3 | 0 | 20 21 22 | 1 2 3 |
| 108 | 165 | 11 | 2 | 0 | 21 22 | 1 2 |
| 104 | 135 | 21 | 2 | 0 | 21 22 | 1 2 3 |
| 100 | 150 | 14 | 3 | 0 | 20 21 22 | 1 2 3 4 |
| 94 | 138 | 24 | 0 | 0 | 20 21 22 | 1 2 9 |
| 73 | 135 | 23 | 1 | 0 | 20 21 22 | 1 2 3 |
| 68 | 140 | 20 | 0 | 1 | 19 20 21 22 | 1 2 4 5 |
| 66 | 146 | 18 | 0 | 1 | 21 22 | 1 2 3 6 |
| 66 | 132 | 22 | 2 | 0 | 21 22 | 1 2 4 |
| 63 | 186 | 6 | 1 | 0 | 20 21 22 | 1 2 3 6 |
| 60 | 152 | 16 | 0 | 1 | 18 19 20 21 22 | 1 2 4 8 24 32 |
| 58 | 132 | 24 | 1 | 0 | 20 21 22 | 1 2 |
| 57 | 201 | 3 | 0 | 0 | 22 | 1 3 |
| 57 | 135 | 19 | 3 | 0 | 21 22 | 1 2 3 |
| 56 | 168 | 10 | 2 | 0 | 20 21 22 | 1 2 4 |
| 55 | 132 | 20 | 3 | 0 | 20 21 22 | 1 2 3 4 |
| 47 | 134 | 20 | 1 | 1 | 20 21 22 | 1 2 3 4 |
| 46 | 132 | 18 | 4 | 0 | 20 21 22 | 1 2 4 6 |
| 41 | 138 | 16 | 4 | 0 | 21 22 | 1 2 4 |
| 36 | 153 | 13 | 3 | 0 | 21 22 | 1 |
| 33 | 128 | 22 | 1 | 1 | 21 22 | 1 2 |
| 32 | 171 | 9 | 2 | 0 | 21 22 | 1 2 3 |
| 32 | 156 | 12 | 3 | 0 | 20 21 22 | 1 2 3 4 6 18 |
| 31 | 134 | 22 | 0 | 1 | 21 22 | 1 2 |
| 29 | 135 | 17 | 4 | 0 | 20 21 22 | 1 2 3 |
| 28 | 128 | 24 | 0 | 1 | 18 19 20 21 22 | 1 2 4 6 8 16 48 96 |
| 27 | 164 | 12 | 0 | 1 | 19 20 22 | 1 2 3 4 6 8 18 |
| 26 | 135 | 25 | 0 | 0 | 21 22 | 1 2 |
| 25 | 128 | 20 | 2 | 1 | 20 21 22 | 1 2 3 6 |
| 24 | 141 | 15 | 4 | 0 | 21 22 | 1 2 3 |
| 23 | 174 | 8 | 2 | 0 | 21 22 | 1 2 3 |
| 22 | 189 | 5 | 1 | 0 | 21 22 | 1 2 |
| 22 | 140 | 18 | 1 | 1 | 21 22 | 1 2 3 6 |
| 20 | 204 | 2 | 0 | 0 | 21 22 | 1 2 |
| 19 | 122 | 22 | 2 | 1 | 21 22 | 1 2 6 |
| 18 | 144 | 14 | 4 | 0 | 21 22 | 1 2 4 |
| 18 | 129 | 21 | 3 | 0 | 20 21 22 | 1 2 3 |
| 17 | 134 | 18 | 2 | 1 | 21 22 | 1 2 3 |
| 17 | 129 | 23 | 2 | 0 | 21 22 | 1 2 |
| 16 | 129 | 25 | 1 | 0 | 21 22 | 1 2 |
| 14 | 132 | 16 | 5 | 0 | 21 22 | 1 2 6 |
| 13 | 134 | 16 | 3 | 1 | 21 22 | 1 2 4 6 |
| 13 | 122 | 18 | 4 | 1 | 20 21 22 | 1 2 3 6 12 |
| 12 | 159 | 11 | 3 | 0 | 21 22 | 1 2 |
| 12 | 158 | 14 | 0 | 1 | 21 22 | 1 2 |
| 12 | 132 | 26 | 0 | 0 | 21 22 | 1 2 |

**Таблица 4.4: Модели по подпространства, $N \leq 12$**

| $N$ | $n_2$ | $n_3$ | $n_4$ | $n_5$ | $rs$ | $auts$ |
|---|---|---|---|---|---|---|
| 12 | 126 | 24 | 2 | 0 | 21 22 | 1 2 3 |
| 12 | 122 | 24 | 1 | 1 | 20 21 22 | 1 2 4 |
| 11 | 150 | 12 | 4 | 0 | 21 22 | 1 2 6 |
| 11 | 146 | 16 | 1 | 1 | 20 21 22 | 1 2 4 6 |
| 11 | 116 | 24 | 2 | 1 | 19 20 21 22 | 2 4 |
| 10 | 162 | 10 | 3 | 0 | 21 22 | 1 2 |
| 10 | 126 | 22 | 3 | 0 | 21 22 | 1 2 |
| 10 | 122 | 20 | 3 | 1 | 21 22 | 1 2 |
| 9 | 158 | 12 | 1 | 1 | 20 21 22 | 2 4 12 |
| 9 | 138 | 14 | 5 | 0 | 21 22 | 1 2 3 |
| 9 | 126 | 26 | 1 | 0 | 21 22 | 1 2 3 |
| 8 | 192 | 4 | 1 | 0 | 20 21 22 | 1 2 4 |
| 8 | 140 | 16 | 2 | 1 | 20 22 | 1 2 4 |
| 8 | 135 | 15 | 5 | 0 | 21 22 | 1 3 |
| 8 | 129 | 19 | 4 | 0 | 20 21 22 | 1 2 |
| 7 | 177 | 7 | 2 | 0 | 21 22 | 1 2 |
| 7 | 176 | 8 | 0 | 1 | 18 19 20 21 22 | 2 4 8 32 |
| 7 | 165 | 9 | 3 | 0 | 20 22 | 1 3 6 |
| 7 | 147 | 13 | 4 | 0 | 21 22 | 1 3 |
| 7 | 129 | 17 | 5 | 0 | 21 22 | 1 2 |
| 7 | 116 | 22 | 3 | 1 | 21 22 | 1 2 |
| 7 | 104 | 24 | 4 | 1 | 18 19 21 22 | 6 8 16 24 |
| 6 | 210 | 0 | 0 | 0 | 20 22 | 3 5 21 |
| 6 | 152 | 14 | 1 | 1 | 21 22 | 1 2 3 4 |
| 6 | 144 | 12 | 5 | 0 | 21 22 | 1 2 3 4 6 |
| 6 | 126 | 20 | 4 | 0 | 21 22 | 1 2 |
| 6 | 120 | 24 | 3 | 0 | 20 21 22 | 1 2 12 |
| 5 | 207 | 1 | 0 | 0 | 22 | 1 |
| 5 | 180 | 6 | 2 | 0 | 21 22 | 1 2 6 |
| 5 | 170 | 10 | 0 | 1 | 21 22 | 2 10 |
| 5 | 132 | 14 | 6 | 0 | 21 22 | 1 2 4 |
| 5 | 126 | 18 | 5 | 0 | 20 21 22 | 1 2 4 6 |
| 5 | 120 | 26 | 2 | 0 | 20 21 22 | 1 2 6 |
| 5 | 116 | 26 | 1 | 1 | 22 | 1 2 |
| 5 | 116 | 18 | 5 | 1 | 19 20 21 22 | 2 6 |
| 4 | 186 | 4 | 2 | 0 | 21 22 | 2 |
| 4 | 174 | 6 | 3 | 0 | 19 20 22 | 1 2 6 12 |
| 4 | 128 | 18 | 3 | 1 | 22 | 1 2 3 |
| 4 | 128 | 16 | 4 | 1 | 18 19 21 | 8 24 64 96 |
| 4 | 126 | 16 | 6 | 0 | 20 22 | 1 2 6 |
| 4 | 123 | 23 | 3 | 0 | 21 22 | 1 2 3 |
| 4 | 122 | 16 | 5 | 1 | 21 22 | 1 2 4 |
| 4 | 110 | 24 | 3 | 1 | 21 22 | 2 3 |
| 4 | 110 | 18 | 6 | 1 | 20 21 | 2 4 6 |
| 4 | 100 | 18 | 6 | 2 | 20 21 22 | 4 6 12 |
| 3 | 198 | 2 | 1 | 0 | 21 22 | 1 2 |
| 3 | 171 | 7 | 3 | 0 | 22 | 1 2 |
| 3 | 156 | 10 | 4 | 0 | 21 22 | 1 2 |
| 3 | 152 | 6 | 5 | 1 | 22 | 4 12 36 |
| 3 | 146 | 12 | 3 | 1 | 22 | 4 12 |
| 3 | 129 | 15 | 6 | 0 | 21 22 | 1 3 |
| 3 | 123 | 21 | 4 | 0 | 21 | 2 6 |
| 3 | 114 | 24 | 4 | 0 | 21 22 | 2 6 |
| 3 | 112 | 26 | 0 | 2 | 20 21 | 2 4 |
| 2 | 200 | 0 | 0 | 1 | 18 | 96 480 |
| 2 | 176 | 0 | 4 | 1 | 18 19 | 24 192 |
| 2 | 170 | 8 | 1 | 1 | 20 | 4 |
| 2 | 168 | 8 | 3 | 0 | 22 | 1 2 |
| 2 | 152 | 12 | 2 | 1 | 22 | 2 6 |
| 2 | 150 | 10 | 5 | 0 | 21 22 | 2 12 |
| 2 | 138 | 12 | 6 | 0 | 22 | 1 12 |
| 2 | 128 | 12 | 6 | 1 | 22 | 6 18 |
| 2 | 126 | 28 | 0 | 0 | 19 22 | 1 42 |
| 2 | 123 | 25 | 2 | 0 | 22 | 1 |
| 2 | 120 | 14 | 8 | 0 | 22 | 4 6 |
| 2 | 116 | 28 | 0 | 1 | 19 20 | 2 8 |
| 2 | 116 | 20 | 4 | 1 | 22 | 1 2 |
| 2 | 112 | 24 | 1 | 2 | 22 | 2 6 |
| 2 | 112 | 18 | 4 | 2 | 21 22 | 6 12 |
| 2 | 110 | 28 | 1 | 1 | 21 | 2 |
| 2 | 110 | 26 | 2 | 1 | 22 | 2 3 |
| 2 | 110 | 16 | 7 | 1 | 20 21 | 4 |
| 2 | 106 | 24 | 2 | 2 | 22 | 2 6 |
| 2 | 104 | 32 | 0 | 1 | 18 19 | 8 32 |
| 2 | 104 | 28 | 2 | 1 | 21 | 2 6 |
| 2 | 104 | 26 | 3 | 1 | 22 | 2 |
| 2 | 100 | 24 | 3 | 2 | 22 | 6 |
| 2 | 100 | 22 | 4 | 2 | 21 | 6 |
| 2 | 98 | 18 | 8 | 1 | 20 | 12 18 |
| 2 | 94 | 12 | 10 | 2 | 19 22 | 12 18 |
| 1 | 183 | 5 | 2 | 0 | 22 | 1 |
| 1 | 182 | 6 | 0 | 1 | 22 | 6 |
| 1 | 180 | 4 | 3 | 0 | 21 | 2 |
| 1 | 177 | 5 | 3 | 0 | 22 | 3 |
| 1 | 176 | 6 | 1 | 1 | 22 | 12 |
| 1 | 164 | 10 | 1 | 1 | 21 | 2 |
| 1 | 164 | 8 | 2 | 1 | 19 | 4 |
| 1 | 160 | 0 | 0 | 5 | 16 | 5760 |
| 1 | 156 | 8 | 5 | 0 | 22 | 2 |
| 1 | 153 | 11 | 4 | 0 | 22 | 1 |
| 1 | 152 | 8 | 4 | 1 | 17 | 96 |
| 1 | 147 | 11 | 5 | 0 | 22 | 3 |
| 1 | 146 | 14 | 2 | 1 | 20 | 2 |
| 1 | 141 | 13 | 5 | 0 | 22 | 1 |
| 1 | 140 | 12 | 4 | 1 | 22 | 6 |
| 1 | 138 | 10 | 7 | 0 | 21 | 2 |
| 1 | 136 | 18 | 0 | 2 | 20 | 12 |
| 1 | 134 | 12 | 5 | 1 | 22 | 12 |
| 1 | 132 | 12 | 7 | 0 | 22 | 2 |
| 1 | 129 | 27 | 0 | 0 | 22 | 1 |
| 1 | 129 | 9 | 9 | 0 | 22 | 9 |
| 1 | 126 | 12 | 8 | 0 | 22 | 2 |
| 1 | 126 | 0 | 9 | 3 | 22 | 324 |
| 1 | 124 | 16 | 3 | 2 | 20 | 12 |
| 1 | 123 | 19 | 5 | 0 | 21 | 2 |
| 1 | 123 | 17 | 6 | 0 | 22 | 3 |
| 1 | 122 | 26 | 0 | 1 | 22 | 2 |
| 1 | 120 | 22 | 4 | 0 | 22 | 2 |
| 1 | 120 | 20 | 5 | 0 | 22 | 2 |
| 1 | 120 | 18 | 6 | 0 | 22 | 2 |
| 1 | 120 | 12 | 9 | 0 | 20 | 36 |
| 1 | 118 | 24 | 0 | 2 | 22 | 6 |
| 1 | 117 | 27 | 2 | 0 | 22 | 2 |
| 1 | 114 | 26 | 3 | 0 | 22 | 2 |
| 1 | 114 | 22 | 5 | 0 | 22 | 2 |
| 1 | 114 | 20 | 6 | 0 | 21 | 2 |
| 1 | 114 | 16 | 8 | 0 | 22 | 4 |
| 1 | 114 | 12 | 10 | 0 | 21 | 6 |
| 1 | 112 | 12 | 7 | 2 | 21 | 18 |
| 1 | 110 | 22 | 4 | 1 | 21 | 2 |
| 1 | 110 | 20 | 5 | 1 | 20 | 4 |
| 1 | 110 | 12 | 9 | 1 | 20 | 12 |
| 1 | 108 | 26 | 4 | 0 | 21 | 6 |
| 1 | 108 | 18 | 3 | 3 | 22 | 36 |
| 1 | 106 | 22 | 3 | 2 | 19 | 12 |
| 1 | 106 | 18 | 5 | 2 | 22 | 6 |
| 1 | 104 | 18 | 7 | 1 | 21 | 6 |
| 1 | 104 | 16 | 8 | 1 | 18 | 24 |
| 1 | 104 | 8 | 12 | 1 | 19 | 48 |
| 1 | 100 | 26 | 2 | 2 | 21 | 4 |
| 1 | 100 | 20 | 5 | 2 | 22 | 2 |
| 1 | 94 | 20 | 6 | 2 | 21 | 6 |
| 1 | 90 | 18 | 6 | 3 | 22 | 18 |
| 1 | 90 | 6 | 12 | 3 | 20 | 36 |
| 1 | 88 | 24 | 0 | 5 | 18 | 288 |
| 1 | 80 | 24 | 8 | 1 | 21 | 48 |
| 1 | 80 | 0 | 20 | 1 | 18 | 576 |
| 1 | 70 | 24 | 3 | 5 | 20 | 36 |
| 1 | 64 | 32 | 0 | 5 | 17 | 384 |
| 1 | 64 | 24 | 4 | 5 | 19 | 72 |
| 1 | 64 | 0 | 16 | 5 | 17 | 1152 |
| 1 | 60 | 18 | 6 | 6 | 20 | 108 |
| 1 | 48 | 0 | 12 | 9 | 18 | 1728 |
| 1 | 0 | 0 | 0 | 21 | 16 | 362880 |

## 4.4 Класификация на всички книжни спредове на $PG(5,2)$

### 4.4.1 Въведение

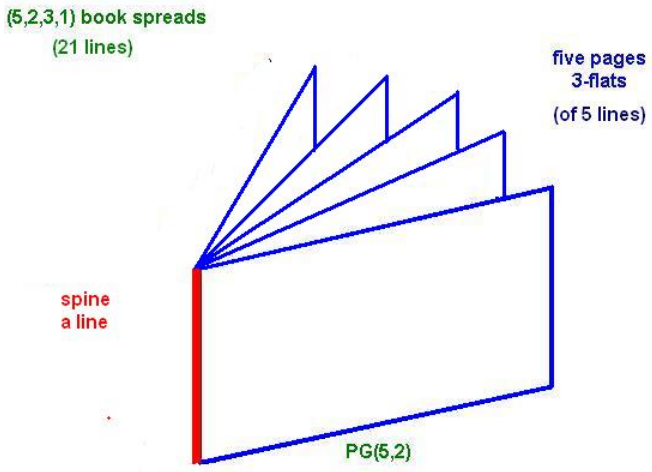
Когато класификацията на всички комбинаторни структури с дадени параметри е невъзможна, обикновено разглеждаме само тези от тях, които притежават повече симетрия, като най-често това са структури с големи групи от автоморфизми. При книжните спредове става въпрос за повече симетрия, но в геометричните свойства.

Дефинирането и разглеждането на книжни спредове е идея на Ronald Shaw. Няколко години преди ние с Матева да публикуваме пълната класификация на спредовете на $PG(5,2)$ (раздел 4.3), Shaw класифицира изцяло теоретично 9-те книжни спреда в $PG(5,2)$, а McDonough прави частична проверка на резултата му с компютър. След време по данните от пълната класификация на спредовете на $PG(5,2)$ Shaw установява, че точно 9 спреда съдържат всичките пет прави на поне пет тримерни подпространства и следователно това могат да бъдат неговите книжни спредове. Това се потвърди от компютърната класификация на всички книжни спредове на $PG(5,2)$, която е обект на настоящия раздел.

### 4.4.2 Свойства на книжните спредове на $PG(5,2)$ и алгоритмите за конструирането им

Задачата може да се разглежда като подслучай на описаната в раздел 4.3.2, но при наложени допълнителни условия, които разглеждаме тук.

Спред в $PG(5,2)$ има 21 прави. Гръбнакът на $(5,2,3,1)$ книжен спред е права. Книжният спред има 5 страници $p_1, p_2, \ldots, p_5$, всяка от които е 3-мерно подпространство от 5 прави (гръбнакът и 4 други прави от спреда).




Без загуба на общност строим $(5,2,3,1)$ книжен спред с прави $l_1, l_2, \ldots l_{21}$ (номерацията на правите отговаря на реда на добавянето им към частичното решение), който да отговаря на следните условия: $l_1$ е гръбнакът, а страниците съдържат правите: $p_1 : l_1, l_2, l_3, l_4, l_5$; $p_2 : l_1, l_6, l_7, l_8, l_9$; $p_3 : l_1, l_{10}, l_{11}, l_{12}, l_{13}$;

$p_4 : l_1, l_{14}, l_{15}, l_{16}, l_{17}$;  $p_5 : l_1, l_{18}, l_{19}, l_{20}, l_{21}$.

Понеже спредът в $PG(3,2)$ е единствен с точност до еквивалентност, по начин, подобен на този в раздел 4.3, без загуба на общност можем да фиксираме първите 6 прави $l_1, l_2, \ldots l_6$. За построяване на спредовете използваме модификация на алгоритъма от раздел 4.3.2, която съблюдава горните изисквания за страниците. Например, ако текущото решение съдържа 15 прави, то следващата права добавяме, само ако е от подпространството $\langle l_1, l_{14} \rangle$. Тези изисквания многократно намаляват броя на получените решения. Конструираните спредове са само 2048 (от тях 9 неизоморфни), докато без условията за книжен спред, при същото фиксиране на първите 6 точки, като първите 5 са от едно подпространство, се получават 15296512 (696 неизоморфни) спреда. Това показва колко силно рестриктивни са условията за книжност на спреда.

Теста за минималност от раздел 4.3.2 допълваме до:

```
for( i=1; i<=21; i++) // 21 прави в спреда
{
    construct π1 = α_{l_i}
    for( j=1; j<=21; j++) if(π1 l_j > l_1) // освен първата има още 20 прави от спреда
    {
        construct π2 = β_{π1 l_j} π1
        for( h=1; h<=21; h++) if(π2 l_h ∈ ⟨l_1, l_2⟩) if(π2 l_h > l_2) // още 3 прави в ⟨l_1, l_2⟩
        {
            construct π3 = γ_{π2 l_h} π2
            for( k=1; k<=21; k++) if(π3 l_k ∈ ⟨l_1, l_2⟩) if(π3 l_k > l_3)// още 2 прави в ⟨l_1, l_2⟩
            {
                construct π4 = δ_{π3 l_k} π3
                for( m=1; m<=21; m++) if(π4 l_m ∉ ⟨l_1, l_2⟩) // 16 прави извън ⟨l_1, l_2⟩
                {
                    construct π5 = η_{π4 l_m} π4
                    for( l=1; l<=18; l++)
                    {
                        construct π = φ_l π5
                        if Smaller(π) return false;
                    }
                }
            }
        }
    }
}
return true;
```

Тестът проверява дали някой от автоморфизмите $\varphi_l \eta_{l_m} \delta_{l_k} \gamma_{l_h} \beta_{l_j} \alpha_{l_i}$ (на брой

21.20.3.2.16.18 = 725760) изобразява текущото решение в лексикографски по-малко. Теста за минималност прилагаме след намиране на пълно решение.

Тъй като фиксираме първите 6 прави, освен описаните в 4.3 автоморфизми, за правите от множеството на търсене, които са от тримерното подпространство $\langle l_1, l_2 \rangle$ запазваме още два автоморфизма:

– $\gamma_{l_h}$, изобразяващ $l_h$ в $l_3$ и фиксиращ $l_1$ и $l_2$,

– $\delta_{l_k}$, изобразяващ $l_k$ в $l_4$ и фиксиращ $l_1$, $l_2$ и $l_3$ (Тук трябва да поясним, че автоморфизъм, който фиксира една права, може да не фиксира точките, с които е инцидентна, а да ги изобразява една в друга).

### 4.4.3 Класификация - инварианти и свойства

Резултатите са представени в Таблица 4.5. Както в раздел 4.3.2 с $n_i$ означаваме броя на тримерните подпространства, съдържащи $i$ прави от спреда, $i = 2, 3, 4, 5$, с $N$ – броя на неекивалентните спредове, с $auts$ – редовете на групите от автоморфизми и с $rs$ – 2-ранговете на съответните 2-(64,16,5) дизайни. Прави впечатление, че всички книжни спредове са с богати групи от автоморфизми и, че и двата спреда, които съответстват на 2-(64,16,5) дизайни с минимален ранг, са книжни спредове.

**Таблица 4.5: Модели по подпространства**

| $N$ | $n_2$ | $n_3$ | $n_4$ | $n_5$ | $rs$ | $auts$ |
|---|---|---|---|---|---|---|
| 1 | 160 | 0 | 0 | 5 | 16 | 5760 |
| 1 | 88 | 24 | 0 | 5 | 18 | 288 |
| 1 | 70 | 24 | 3 | 5 | 20 | 36 |
| 1 | 64 | 32 | 0 | 5 | 17 | 384 |
| 1 | 64 | 24 | 4 | 5 | 19 | 72 |
| 1 | 64 | 0 | 16 | 5 | 17 | 1152 |
| 1 | 60 | 18 | 6 | 6 | 20 | 108 |
| 1 | 48 | 0 | 12 | 9 | 18 | 1728 |
| 1 | 0 | 0 | 0 | 21 | 16 | 362880 |

Резултатите от настоящата част са публикувани в [T10] заедно с теоретичното доказателство на Shaw и частичната проверка на McDonough.

## 4.5 Класификация на някои видове книжни спредове на $PG(7,2)$

### 4.5.1 Въведение

Само 9 от всичките 131044 спреда в $PG(5,2)$ са книжни, но се оказва, че са сред най-интересните, защото:

- Те са малка част от всички спредове и поради това са по-лесни за класификация.
- Допълнителните геометрични свойства са причина да очакваме:
  - много книжни спредове с богати групи от автоморфизми;
  - свойства, които са теоретично предсказуеми.
- Класификацията на книжните спредове в $PG(5,2)$ (раздел 4.5) е показателна:
  - всичките 9 спреда имат богати групи от автоморфизми;
  - те са единствените спредове, при които най-малко пет 3-мерни подпространства се покриват напълно от прави от спреда;
  - именно предсказуемостта на някои свойства направи възможна и теоретичната им класификация.

Всичко това ни мотивира за по-нататъшни изследвания на книжни спредове. В [T11] са изведени следните условия за съществуването на книжни спредове, както и на книжни спредове с някои допълнителни свойства:

**Теорема 4.5.1.1.** *$(n,q,r,s)$ книга съществува, тогава и само тогава, когато $r-s$ дели $n-s$.*

**Теорема 4.5.1.2.** *Броят на страниците на $(n,q,r,s)$ книга е*

$$N = 1 + q^{r-s} + q^{2(r-s)} + \ \ldots \ + q^{(a-1)(r-s)}, \tag{4.1}$$

*където $a = (n-s)/(r-s)$.*

**Теорема 4.5.1.3.** *Ако съществува $(n,q,r,s)$ книжен $t$-спред, то неговите параметри $n,r,s,t$ удовлетворяват условията:*

$$\text{(i) } r-s \mid n-s, \quad \text{(ii) } t+1 \mid s+1, \quad \text{(iii) } t+1 \mid r+1\ . \tag{4.2}$$

*Ако тези условия са в сила, всеки $t$-спред $\mathcal{S}_L$ на гръбнака $L$ на $(n,q,r,s)$ книга може да бъде разширен до $(n,q,r,s)$ книжен $t$-спред.*

**Дефиниция 4.5.1.** *Един $(n,q,r,s)$ книжен $t$-спред е* ***цикличен****, ако има автоморфизъм, който пермутира страниците му в един цикъл $(P_1,P_2,\ldots,P_N)$.*

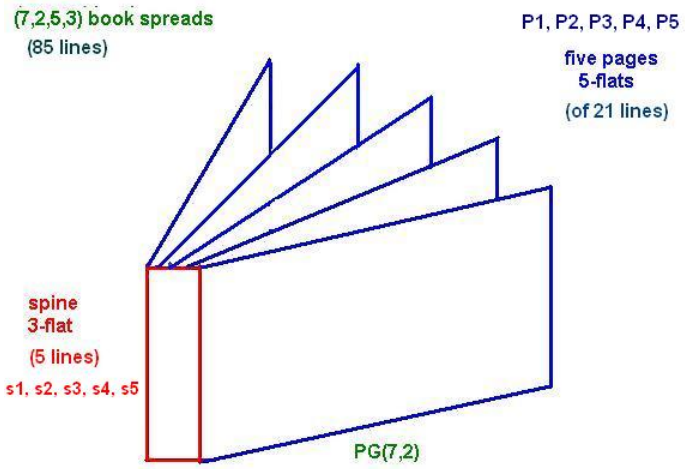
**Теорема 4.5.1.4.** *Нека $\mathcal{S}_L$ е $t$-спред в някое $s$-мерно подпространство $L$ на $PG(n,q)$. Ако условията (4.2) са в сила, съществува цикличен $(n,q,r,s)$ книжен $t$-спред с гръбнак $L$.*

**Следствие 4.5.1.** *Цикличен $(n,2,n-2,n-4)$ книжен спред с 5 страници съществува за всяко нечетно $n \geq 5$.*

**Пример 4.5.1.** *С точност до еквивалентност има два циклични $(5,2,3,1)$ книжни спреда – регулярния спред ($t_1$ в Таблица 4.6), и този, който е под номер $t_9$ в Таблица 4.6. При цикличен $(7,2,5,3)$ книжен спред автоморфизмът от ред 5 или фиксира всичките пет прави на гръбнака, или ги пермутира в цикъл от ред 5.*

### 4.5.2 Видове книжни спредове в $PG(7,2)$ и алгоритмите, с които ги конструираме

В настоящия раздел представяме класификация на $(7,2,5,3)$ книжни спредове, отговарящи на някои допълнителни условия. В $PG(7,2)$ има 255 точки и 10795 прави. Всеки спред има 85 прави. Гръбнакът на $(7,2,5,3)$ книжен спред е 3-мерно подпространство, съдържащо 5 прави от спреда, които ще означаваме с $l_1, l_2, l_3, l_4$ и $l_5$. Спредът има 5 страници. Означаваме ги с $P_1, P_2, P_3, P_4$ и $P_5$. Всяка страница е 5-мерно подпространство с 21 прави от спреда (5-те прави на гръбнака и 16 други).




Невъзможността да класифицираме всички книжни спредове в $PG(7,2)$ ни кара да търсим допълнителни свойства. Типът на спредовете в петте страници използваме, за да дефинираме следните три типа $(7,2,5,3)$ книжни спредове, т.е.

- $(7,2,5,3)_1$:
  правите на всяка страница образуват $(5,2,3,1)$ книжен спред с гръбнак $l_1$
- $(7,2,5,3)_2$:
  правите на страница $P_i$ образуват $(5,2,3,1)$ книжен спред с гръбнак $l_i$ за $i = 1,2,3,4,5$.

- $(7,2,5,3)_3$:
  всички останали $(7,2,5,3)$ книжни спредове.

С $(7,2,5,3)_{1(2)}$ означаваме спредове, които са или $(7,2,5,3)_1$ или $(7,2,5,\ 3)_2$, а с $(7,2,5,3)_{1+2}$ – спредове, които са едновременно $(7,2,5,3)_1$ и $(7,2,5,3)_2$.

По-нататък разглеждаме само $(7,2,5,3)_{1(2)}$ книжни спредове (без да го обясняваме допълнително). Понеже има девет $(5,2,3,1)$ книжни спреда, разглеждаме девет типа страници според книжните спредове, които съдържат, т.е. $t_1, t_2, ...t_9$. Инвариантите на съответните $(5,2,3,1)$ книжни спредове са представени в Таблица 4.6, където $|Aut|$ е реда на групата от автоморфизми, а $n_i$ – броят на 3-мерните подпространства, съдържащи $i$ прави от спреда. Страници от тип $t_1$ съдържат регулярния спред в $PG(5,2)$.

**Таблица 4.6: Деветте** $(5,2,3,1)$ **книжни спреда дефинират девет типа страници на** $(7,2,5,3)_{1(2)}$ **книжни спредове**

| тип страница | $\|Aut\|$ | $n_3$ | $n_4$ | $n_5$ |
|---|---|---|---|---|
| $t_1$ | 362880 | 0 | 0 | 21 |
| $t_2$ | 1728 | 0 | 12 | 9 |
| $t_3$ | 1152 | 0 | 16 | 5 |
| $t_4$ | 108 | 18 | 6 | 6 |
| $t_5$ | 72 | 24 | 4 | 5 |

| тип страница | $\|Aut\|$ | $n_3$ | $n_4$ | $n_5$ |
|---|---|---|---|---|
| $t_6$ | 384 | 32 | 0 | 5 |
| $t_7$ | 36 | 24 | 3 | 5 |
| $t_8$ | 288 | 24 | 0 | 5 |
| $t_9$ | 5760 | 0 | 0 | 5 |
| | | | | |

Типът на $(7,2,5,3)_{1(2)}$ книжен спред е $a_1a_2a_3a_4a_5$, където $t_{a_i}$ е типът на $i$-та страница. Книжните спредове $(7,2,5,3)_{1(2)}$ са малка част от всички $(7,2,5,3)$ книжни спредове, но и тяхната пълна класификация е невъзможна по нашия начин. Затова класифицираме само

1. $(7,2,5,3)_1$ и $(7,2,5,3)_2$ книжни спредове с изоморфни страници и от типове 11111, 22222, 33333 и 99999;
2. $(7,2,5,3)_{1+2}$ книжни спредове от всички възможни типове.

Точките на $PG(7,2)$ съответстват на ненулевите 8-мерни вектори над $GF(2)$. От тях намираме подпространствата и построяваме свързаните с тях дизайни. На 255-те вектора (точки) даваме номера $p_x$ такива, че $\psi(p_x) = x$, където $x$ ($x = 1,2,\ldots 255$) е цялото число, чието двоично представяне отговаря на съответния вектор, а функцията $\psi$ е дадена с Таблица 4.7. Това специфично подреждане на точките избираме, за да са в сила следните условия:

- първите 15 точки са от едно и също 3-мерно подпространство,
- първите 63 точки са от едно и също 5-мерно подпространство,
- точките $p_x$, $p_{x+1}$ и $p_{x+2}$ са от една права за $x \equiv 1 (mod\ 3)$,
- правите $(p_1,p_2,p_3)$, $(p_4,p_5,p_6)$, $\ldots$, $(p_{253},p_{254},p_{255})$ образуват регулярен книжен спред.

**Таблица 4.7:** $\psi$

| $p_x$ | $x$ | $p_x$ | $x$ | $p_x$ | $x$ | $p_x$ | $x$ | $p_x$ | $x$ | $p_x$ | $x$ | $p_x$ | $x$ | $p_x$ | $x$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 33 | 63 | 65 | 128 | 97 | 75 | 129 | 255 | 161 | 176 | 193 | 107 | 225 | 235 |
| 2 | 2 | 34 | 22 | 66 | 192 | 98 | 141 | 130 | 86 | 162 | 208 | 194 | 189 | 226 | 118 |
| 3 | 3 | 35 | 43 | 67 | 65 | 99 | 198 | 131 | 171 | 163 | 97 | 195 | 214 | 227 | 159 |
| 4 | 4 | 36 | 61 | 68 | 130 | 100 | 76 | 132 | 253 | 164 | 178 | 196 | 108 | 228 | 233 |
| 5 | 8 | 37 | 23 | 69 | 195 | 101 | 132 | 133 | 87 | 165 | 211 | 197 | 180 | 229 | 119 |
| 6 | 12 | 38 | 41 | 70 | 66 | 102 | 200 | 134 | 169 | 166 | 98 | 198 | 216 | 230 | 157 |
| 7 | 5 | 39 | 62 | 71 | 131 | 103 | 77 | 135 | 254 | 167 | 179 | 199 | 109 | 231 | 234 |
| 8 | 10 | 40 | 24 | 72 | 193 | 104 | 134 | 136 | 88 | 168 | 209 | 200 | 182 | 232 | 120 |
| 9 | 15 | 41 | 44 | 73 | 67 | 105 | 203 | 137 | 172 | 169 | 99 | 201 | 219 | 233 | 148 |
| 10 | 6 | 42 | 52 | 74 | 129 | 106 | 78 | 138 | 244 | 170 | 177 | 202 | 110 | 234 | 236 |
| 11 | 11 | 43 | 25 | 75 | 194 | 107 | 135 | 139 | 89 | 171 | 210 | 203 | 183 | 235 | 121 |
| 12 | 13 | 44 | 46 | 76 | 68 | 108 | 201 | 140 | 174 | 172 | 100 | 204 | 217 | 236 | 150 |
| 13 | 7 | 45 | 55 | 77 | 136 | 109 | 79 | 141 | 247 | 173 | 184 | 205 | 111 | 237 | 239 |
| 14 | 9 | 46 | 26 | 78 | 204 | 110 | 133 | 142 | 90 | 174 | 220 | 206 | 181 | 238 | 122 |
| 15 | 14 | 47 | 47 | 79 | 69 | 111 | 202 | 143 | 175 | 175 | 101 | 207 | 218 | 239 | 151 |
| 16 | 16 | 48 | 53 | 80 | 138 | 112 | 80 | 144 | 245 | 176 | 186 | 208 | 112 | 240 | 237 |
| 17 | 32 | 49 | 27 | 81 | 207 | 113 | 160 | 145 | 91 | 177 | 223 | 209 | 144 | 241 | 123 |
| 18 | 48 | 50 | 45 | 82 | 70 | 114 | 240 | 146 | 173 | 178 | 102 | 210 | 224 | 242 | 149 |
| 19 | 17 | 51 | 54 | 83 | 139 | 115 | 81 | 147 | 246 | 179 | 187 | 211 | 113 | 243 | 238 |
| 20 | 34 | 52 | 28 | 84 | 205 | 116 | 162 | 148 | 92 | 180 | 221 | 212 | 146 | 244 | 124 |
| 21 | 51 | 53 | 36 | 85 | 71 | 117 | 243 | 149 | 164 | 181 | 103 | 213 | 227 | 245 | 152 |
| 22 | 18 | 54 | 56 | 86 | 137 | 118 | 82 | 150 | 248 | 182 | 185 | 214 | 114 | 246 | 228 |
| 23 | 35 | 55 | 29 | 87 | 206 | 119 | 163 | 151 | 93 | 183 | 222 | 215 | 147 | 247 | 125 |
| 24 | 49 | 56 | 38 | 88 | 72 | 120 | 241 | 152 | 166 | 184 | 104 | 216 | 225 | 248 | 154 |
| 25 | 19 | 57 | 59 | 89 | 140 | 121 | 83 | 153 | 251 | 185 | 188 | 217 | 115 | 249 | 231 |
| 26 | 33 | 58 | 30 | 90 | 196 | 122 | 161 | 154 | 94 | 186 | 212 | 218 | 145 | 250 | 126 |
| 27 | 50 | 59 | 39 | 91 | 73 | 123 | 242 | 155 | 167 | 187 | 105 | 219 | 226 | 251 | 155 |
| 28 | 20 | 60 | 57 | 92 | 142 | 124 | 84 | 156 | 249 | 188 | 190 | 220 | 116 | 252 | 229 |
| 29 | 40 | 61 | 31 | 93 | 199 | 125 | 168 | 157 | 95 | 189 | 215 | 221 | 156 | 253 | 127 |
| 30 | 60 | 62 | 37 | 94 | 74 | 126 | 252 | 158 | 165 | 190 | 106 | 222 | 232 | 254 | 153 |
| 31 | 21 | 63 | 58 | 95 | 143 | 127 | 85 | 159 | 250 | 191 | 191 | 223 | 117 | 255 | 230 |
| 32 | 42 | 64 | 64 | 96 | 197 | 128 | 170 | 160 | 96 | 192 | 213 | 224 | 158 | | |

За конструиране на спредовете използваме модификация на алгоритъма от раздел 4.2. Правите на спреда означаваме с $l_1, l_2, \ldots, l_{85}$ и построяваме само спредове със следните свойства:

- Правите $l_1, l_2, l_3, l_4$, и $l_5$ образуват гръбнака
- Освен правите на гръбнака, $P_i$ съдържа и правите $l_{16(i-1)+a}$, където $i = 1, 2, ..., 5$ и $a = 6, 7, ...21$.
- Правите $l_j$, $l_{j+1}$, $l_{j+2}$ и $l_{j+3}$ са от една и съща страница на $(5, 2, 3, 1)$ книжния спред в $P_i$ $(i = 1, 2, ..., 5)$ ако $j = 16(i-1) + b$, където $b = 6, 10, 14, 18$.

Без загуба на общност фиксираме седем прави от спреда, т.е. $l_1 = (p_1, p_2, p_3)$, $l_2 = (p_4, p_5, p_6)$, $\ldots$, $l_6 = (p_{16}, p_{17}, p_{18})$ и $l_{22} = (p_{64}, p_{65}, p_{66})$. Това фиксиране е възможно, защото спредът в $PG(3, 2)$ (от правите $l_1, l_2, l_3, l_4$, и $l_5$) е единствен с точност до еквивалентност, а групата от автоморфизми на $PG(7, 2)$, която фиксира 3(5)-мерно подпространство, е транзитивна върху правите извън него.

Останалите прави избираме от множеството на търсене $D$, състоящо се от

всички прави, непресичащи никоя от седемте фиксирани прави. Броят им е 1747 (докато всички прави на проективното пространство са 10795). Тест за минималност на частичните решения прилагаме след добавяне на цяла страница от книжния $(7, 2, 5, 3)$ спред, т.е. след добавяне на правите $l_{21}, l_{37}, l_{53}, l_{69}$ или $l_{85}$. Той придобива вида:

```
for( i=1; i<=85; i++) // 85 прави в спреда
{
    construct π1 = α_{l_i}
    for( j=1; j<=85; j++) if(π1 l_j > l_1) // освен първата има още 84 прави от спреда
    {
        construct π2 = β_{π1 l_j} π1
        for( h=1; h<=85; h++) if(π2 l_h ∈ ⟨l_1, l_2⟩) if(π2 l_h > l_2) // още 3 прави в ⟨l_1, l_2⟩
        {
            construct π3 = γ_{π2 l_h} π2
            for( k=1; k<=85; k++) if(π3 l_k ∈ ⟨l_1, l_2⟩) if(π3 l_k > l_3)// още 2 прави в ⟨l_1, l_2⟩
            {
                construct π4 = δ_{π3 l_k} π3
                for( m=1; m<=85; m++) if(π4 l_m ∉ ⟨l_1, l_2⟩) // 80 прави извън ⟨l_1, l_2⟩
                {
                    construct π5 = η_{π4 l_m} π4
                    if Smaller(π5) return false;
                    for( s=1; s<=85; s++) if(π5 l_s ∉ ⟨l_1, l_2, l_6⟩) // 64 извън ⟨l_1, l_2, l_3⟩
                    {
                        construct π6 = ρ_{π5 l_s} π5
                        for( l=1; l<=108; l++)
                        {
                            construct π = φ_l π6
                            if Smaller(π) return false;
                        }
                    }
                }
            }
        }
    }
}
return true;
```

В този тест използваме предварително запазените автоморфизми, от които:
– $\alpha_{l_i}$ изобразява правата $l_i$ в $l_1$,
– $\beta_{l_j}$ фиксира $l_1$ и изобразява правата $l_j$ в $l_2$,
– $\gamma_{l_h}$ изобразява $l_h$ в $l_3$, като фиксира $l_1$ и $l_2$,

– $\delta_{l_k}$ изобразява $l_k$ в $l_4$, като фиксира $l_1$, $l_2$ и $l_3$,
– $\eta_{l_m}$ фиксира $\langle l_1, l_2 \rangle$ и изобразява правата $l_m$ в $l_6$,
– $\rho_{l_s}$ фиксира $\langle l_1, l_2, l_6 \rangle$ и изобразява правата $l_s$ в $l_{22}$,
– $\varphi_l$ е автоморфизъм, фиксиращ правите $l_1, l_2, ...l_6$ и $l_{22}$.
С горните автоморфизми изобразяваме правите от спреда в първите шест фиксирани прави $l_1, l_2, ...l_6$ по всичките 85.84.3.2.80 начина (т.е. 85 начина за $l_1$, 84 за $l_2$, 3 за $l_3$, 2 за $l_4$ и 80 за $l_6$). Ако по този начин получим лексикографски по-малко решение за първата страница (първите 21 прави), отхвърляме текущото частично решение. В противен случай избираме права, която да изобразим в $l_{22}$ (по всички 64 начина) и прилагаме 108-те автоморфизми, които запазват всички фиксирани прави. Ако някой от тях изобразява текущото решение в лексикографски по-малко, отхвърляме текущото частично решение.

### 4.5.3 Класификация - книжни спредове с изоморфни страници, инварианти

Класификационните резултати са обобщени в следващите таблици. В тях са използвани следните означения:

- *cov5* - брой 5-мерни подпространства, които са изцяло покрити от прави от спреда. Този брой е най-малко 5, защото книжните спредове, които разглеждаме, имат 5 страници.
- *cov3* - брой 3-мерни подпространства, които са изцяло покрити от прави от спреда. Този брой е най-малко 21, защото всяка страница на $(7,2,5,3)_{1(2)}$ книжен спред съдържа гръбнака и четири други 3-мерни подпространства.
- *Sp* - брой 3-мерни подпространства, които са изцяло покрити от прави от спреда и са гръбнаци. Възможно е да сме конструирали $(7,2,5,3)_t$ спред, докато някои от тези *Sp* гръбнаци да са гръбнаци на $(7,2,5,3)_{t'}$ книжен спред, където $t \neq t'$. Ако $Sp = 1$, спреда наричаме прост.
- *lSp* - брой на правите от спреда, които заедно с коя да е права от спреда задават изцяло покрито 3-мерно подпространство. Такава права е гръбнак на $(7,2,3,1)$ книжен спред и би могла да е гръбнак на някоя от страниците на $(7,2,5,3)_1$ книжен спред. За $(7,2,5,3)_1$ *lSp* е най-малко 1.
- $|Aut|$ - ред на пълната група от автоморфизми на спреда.
- *spr* - брой на спредовете с такива инварианти

#### $(7,2,5,3)_1$ и $(7,2,5,3)_2$ книжни спредове с изоморфни страници

Установяваме, че всичките $(7,2,5,3)_{1(2)}$ книжни спредове от тип 11111 са $(7,2,5,3)_{1+2}$ книжни спредове. Техният брой е 13, като единият е регулярният

спред с екстремални стойности на инвариантите. Класификационните резултати са представени в Таблица 4.8.

**Таблица 4.8:** $(7,2,5,3)_1/(7,2,5,3)_2$ **книжни спредове от тип** 11111

| $\|Aut\|$ | spr |
|---|---|
| $432 = 3^3.2^4$ | 1 |
| $1152 = 3^2.2^7$ | 1 |
| $3456 = 3^3.2^7$ | 1 |
| $6480 = 5.3^4.2^4$ | 1 |
| $9216 = 3^2.2^{10}$ | 1 |
| $13824 = 3^3.2^9$ | 1 |
| $15360 = 5.3.2^{10}$ | 1 |
| $18432 = 3^2.2^{11}$ | 1 |
| $73728 = 3^2.2^{13}$ | 1 |
| $207360 = 5.3^4.2^9$ | 1 |
| $221184 = 3^3.2^{13}$ | 1 |
| $1105920 = 5.3^3.2^{13}$ | 1 |
| $5922201600 = 17.7.5^2.3^5.2^{13}$ | 1 |
| All | 13 |

| cov5 | Sp | lSp | cov3 | $\|Aut\|$ | spr |
|---|---|---|---|---|---|
| 5 | 1 | 5 | 101 | 1152 - 18432 | 4 |
| 6 | 1 | 5 | 101 | 432 | 1 |
| 9 | 2 | 5 | 101 | 3456 - 13824 | 2 |
| 10 | 2 | 5 | 102 | 6480 | 1 |
| 21 | 21 | 5 | 101 | 73728 - 1105920 | 3 |
| 25 | 22 | 6 | 117 | 207360 | 1 |
| 85 | 357 | 85 | 357 | 5922201600 | 1 |
| All | | | | | 13 |

В Таблици 4.9 до 4.14 са представени класификационни резултати за $(7,2,5,3)_{1(2)}$ книжни спредове със страници от типове $t_2$, $t_3$ и $t_9$. Броят на спредовете със страници от типове $t_4 \dots t_8$ е значително по-голям и класификация с нашите средства не е възможна. Сред спредовете от тип 22222 има 188 книжни $(7,2,5,3)_{1+2}$ спреда.

**Всички** $(7,2,5,3)_{1+2}$ **книжни спредове**

Успяхме да класифицираме всички $(7,2,5,3)_{1+2}$ книжни спредове (Таблица 4.15), защото броят им е сравнително малък поради многото допълнителни свойства. Книжните $(5,2,3,1)$ спредове в четири от техните страници трябва да имат най-малко две прави, които са гръбнаци на съответния $(5,2,3,1)$ спред и най-малко девет 3-мерни подпространства трябва да са изцяло покрити от прави от спреда. Тези условия са в сила само за страници от типове $t_1$ и $t_2$ (Таблица 4.6). Затова само една от страниците на $(7,2,5,3)_{1+2}$ може да е от произволен тип, останалите четири трябва да са от типове $t_1$ или $t_2$.

Конструираните спредове и инвариантите им са достъпни онлайн. Те могат да бъдат изтеглени от http://www.moi.math.bas.bg/∼svetlana и да бъдат използвани от всеки, който проявява интерес. Изследванията от тази част са съвместни с Ronald Shaw и са публикувани в [T11].

**Таблица 4.9:** $(7,2,5,3)_1$ **книжни спредове от тип** 22222

| $\|Aut\|$ | spr |
|---|---|
| 1 | 57823 |
| 2 | 8624 |
| 3 | 1344 |
| $4 = 2^2$ | 655 |
| $6 = 3.2$ | 2027 |
| $8 = 2^3$ | 264 |
| $9 = 3^2$ | 31 |
| $12 = 3.2^2$ | 305 |

| $\|Aut\|$ | spr |
|---|---|
| $16 = 2^4$ | 14 |
| $18 = 3^2.2$ | 445 |
| $24 = 3.2^3$ | 236 |
| $32 = 2^5$ | 10 |
| $36 = 3^2.2^2$ | 87 |
| $48 = 3.2^4$ | 35 |
| $54 = 3^3.2$ | 74 |
| $72 = 3^2.2^3$ | 95 |

| $\|Aut\|$ | spr |
|---|---|
| $96 = 3.2^5$ | 60 |
| $108 = 3^3.2^2$ | 23 |
| $120 = 5.3.2^3$ | 1 |
| $144 = 3^2.2^4$ | 6 |
| $162 = 3^4.2$ | 5 |
| $192 = 3.2^6$ | 2 |
| $216 = 3^3.2^3$ | 22 |
| $288 = 3^2.2^5$ | 37 |

| $\|Aut\|$ | spr |
|---|---|
| $324 = 3^4.2^2$ | 4 |
| $360 = 5.3^2.2^3$ | 3 |
| $384 = 3.2^7$ | 10 |
| $432 = 3^3.2^4$ | 2 |
| $576 = 3^2.2^6$ | 1 |
| $648 = 3^4.2^3$ | 3 |
| $864 = 3^3.2^5$ | 13 |
| $1080 = 5.3^3.2^3$ | 2 |

| $\|Aut\|$ | spr |
|---|---|
| $1152 = 3^2.2^7$ | 12 |
| $1296 = 3^4.2^4$ | 1 |
| $1440 = 5.3^2.2^5$ | 1 |
| $3456 = 3^3.2^7$ | 7 |
| $4320 = 5.3^3.2^5$ | 1 |
| $4608 = 3^2.2^9$ | 1 |
| $5184 = 3^4.2^6$ | 1 |
| $13824 = 3^3.2^9$ | 2 |
| $20736 = 3^4.2^8$ | 1 |
| $41472 = 3^4.2^9$ | 1 |
| $69120 = 5.3^3.2^9$ | 1 |
| All | 72292 |

| cov5 | Sp | lSp | cov3 | $\|Aut\|$ | spr |
|---|---|---|---|---|---|
| 5 | 1 | 1 | 41 | 1 - 1152 | 57333 |
| 6 | 1 | 1 | 41 - 57 | 1 - 1080 | 12360 |
| 7 | 1 | 1 | 42 - 57 | 3 - 324 | 102 |
| 9 | 2 | 1 | 41 - 93 | 1 - 3456 | 2174 |
| 10 | 2 | 1 | 42 - 93 | 6 - 1296 | 36 |
| 13 | 3 | 1 | 105 | 3456 | 1 |
| 13 | 3 | 2 | 57 - 93 | 5184 - 20736 | 2 |
| 21 | 21 | 1 | 41 - 147 | 1 - 69120 | 274 |
| 22 | 21 | 1 | 66 - 147 | 108 - 1080 | 9 |
| 25 | 22 | 2 | 153 | 41472 | 1 |
| All | | | | | 72292 |

**Таблица 4.10:** $(7,2,5,3)_2$ **книжни спредове от тип** 22222

| $\|Aut\|$ | spr |
|---|---|
| 1 | 1469324 |
| 2 | 38311 |
| 3 | 18944 |
| $4 = 2^2$ | 494 |

| $\|Aut\|$ | spr |
|---|---|
| $6 = 3.2$ | 4568 |
| $8 = 2^3$ | 143 |
| $9 = 3^2$ | 426 |
| $12 = 3.2^2$ | 163 |

| $\|Aut\|$ | spr |
|---|---|
| $15 = 5.3$ | 2 |
| $18 = 3^2.2$ | 463 |
| $24 = 3.2^3$ | 111 |
| $27 = 3^3$ | 5 |

| $\|Aut\|$ | spr |
|---|---|
| $32 = 2^5$ | 4 |
| $36 = 3^2.2^2$ | 29 |
| $54 = 3^2.2$ | 27 |
| $60 = 5.3.2^2$ | 2 |

| $\|Aut\|$ | spr |
|---|---|
| $72 = 3^2.2^3$ | 48 |
| $96 = 3.2^5$ | 12 |
| $108 = 3^2.2^2$ | 4 |
| $216 = 3^2.2^3$ | 5 |
| $288 = 3^2.2^5$ | 11 |
| $432 = 3^2.2^4$ | 1 |
| $576 = 3^2.2^6$ | 1 |
| $864 = 3^2.2^5$ | 4 |
| $2160 = 5.3^3.2^4$ | 2 |
| All | 1533104 |

| cov5 | Sp | lSp | cov3 | $\|Aut\|$ | spr |
|---|---|---|---|---|---|
| 5 | 1 | 0 | 41 - 136 | 1 - 72 | 1522305 |
| 5 | 1 | 1 | 41 | 1 - 288 | 130 |
| 6 | 1 | 0 | 41 - 137 | 1 - 72 | 10471 |
| 6 | 1 | 1 | 41 - 45 | 2 - 54 | 32 |
| 7 | 1 | 0 | 42 - 135 | 2 - 108 | 72 |
| 9 | 2 | 0 | 41 - 57 | 3 - 432 | 62 |
| 9 | 2 | 1 | 41 - 57 | 6 - 864 | 20 |
| 10 | 2 | 0 | 48 - 129 | 108 - 2160 | 6 |
| 21 | 21 | 1 | 41 - 57 | 72 - 864 | 6 |
| All | | | | | 1533104 |

**Таблица 4.11:** $(7,2,5,3)_1$ **книжни спредове от тип** 33333

| $\|Aut\|$ | spr |
|---|---|
| 1 | 6769 |
| 2 | 395 |
| 3 | 153 |
| $4 = 2^2$ | 25771 |
| $6 = 3.2$ | 34 |
| $8 = 2^3$ | 1387 |
| $12 = 3.2^2$ | 1267 |
| $16 = 2^4$ | 17811 |
| $24 = 3.2^3$ | 249 |
| $32 = 2^5$ | 1259 |
| $36 = 3^2.2^2$ | 16 |
| $48 = 3.2^4$ | 3115 |
| $64 = 2^6$ | 176 |
| $72 = 3^2.2^3$ | 14 |

| $\|Aut\|$ | spr |
|---|---|
| $96 = 3.2^5$ | 297 |
| $128 = 2^8$ | 61 |
| $144 = 3^2.2^4$ | 72 |
| $192 = 3.2^6$ | 2531 |
| $256 = 2^8$ | 7 |
| $288 = 3^2.2^5$ | 14 |
| $320 = 5.2^6$ | 1 |
| $384 = 3.2^8$ | 365 |
| $512 = 2^9$ | 3 |
| $576 = 3^2.2^6$ | 51 |
| $768 = 3.2^8$ | 66 |
| $960 = 5.3.2^6$ | 1 |
| $1152 = 3^2.2^7$ | 39 |
| $1536 = 3.2^8$ | 24 |

| $\|Aut\|$ | spr |
|---|---|
| $1920 = 5.3.2^7$ | 1 |
| $2304 = 3^2.2^8$ | 13 |
| $3072 = 3.2^9$ | 3 |
| $3840 = 5.3.2^8$ | 2 |
| $4608 = 3^2.2^9$ | 6 |
| $5760 = 5.3^2.2^7$ | 1 |
| $6144 = 3.2^{10}$ | 2 |
| $9216 = 3^2.2^{10}$ | 1 |
| $11520 = 5.3^2.2^8$ | 1 |
| $18432 = 3^2.2^{11}$ | 2 |
| $27648 = 3^3.2^{10}$ | 1 |
| $55296 = 3^3.2^{11}$ | 1 |
| $73728 = 3^2.2^{13}$ | 1 |
| $184320 = 5.3^2.2^{12}$ | 1 |
| All | 61984 |

| cov5 | Sp | lSp | cov3 | $\|Aut\|$ | spr |
|---|---|---|---|---|---|
| 5 | 1 | 1 | 21 | 1 - 1536 | 41293 |
| 6 | 1 | 1 | 21 - 22 | 4 - 144 | 10656 |
| 9 | 2 | 1 | 21 - 37 | 16 - 4608 | 7010 |
| 21 | 21 | 1 | 21 - 149 | 192 - 184320 | 3025 |
| All | | | | | 61984 |

**Таблица 4.12:** $(7,2,5,3)_2$ **книжни спредове от тип** 33333

| $\|Aut\|$ | spr |
|---|---|
| 1 | 344357 |
| 2 | 5998 |
| 3 | 3446 |
| $4 = 2^2$ | 43808 |
| $6 = 3.2$ | 294 |
| $8 = 2^3$ | 994 |
| $9 = 3^2$ | 25 |

| $\|Aut\|$ | spr |
|---|---|
| $12 = 3.2^2$ | 2862 |
| $16 = 2^4$ | 65 |
| $18 = 3^2.2$ | 10 |
| $24 = 3.2^3$ | 186 |
| $32 = 2^5$ | 6 |
| $36 = 3^2.2^2$ | 6 |
| $48 = 3.2^4$ | 31 |

| $\|Aut\|$ | spr |
|---|---|
| $60 = 5.3.2^2$ | 3 |
| $72 = 3^2.2^3$ | 1 |
| $96 = 3.2^5$ | 14 |
| $120 = 5.3.2^3$ | 2 |
| $240 = 5.3.2^4$ | 2 |
| $360 = 5.3^2.2^3$ | 1 |
| All | 402111 |

| cov5 | Sp | lSp | cov3 | $\|Aut\|$ | spr |
|---|---|---|---|---|---|
| 5 | 1 | 0 | 21 - 93 | 1 - 360 | 402111 |
| All | | | | | 402111 |

**Таблица 4.13:** $(7,2,5,3)_1$ **книжни спредове от тип** 99999

| $\|Aut\|$ | spr |
|---|---|
| 1 | 1 |
| 2 | 2 |
| $4 = 2^2$ | 5 |
| $8 = 2^3$ | 4 |
| $12 = 3.2^2$ | 2 |
| $16 = 2^4$ | 2 |

| $\|Aut\|$ | spr |
|---|---|
| $24 = 3.2^3$ | 2 |
| $32 = 2^5$ | 4 |
| $48 = 3.2^4$ | 13 |
| $64 = 2^6$ | 6 |
| $96 = 3.2^5$ | 8 |
| $128 = 2^7$ | 4 |

| $\|Aut\|$ | spr |
|---|---|
| $144 = 3^2.2^4$ | 1 |
| $192 = 3.2^6$ | 5 |
| $256 = 2^8$ | 2 |
| $384 = 3.2^7$ | 6 |
| $512 = 2^8$ | 2 |
| $768 = 3.2^8$ | 2 |

| $\|Aut\|$ | spr |
|---|---|
| $960 = 5.3.2^6$ | 1 |
| $1024 = 2^9$ | 2 |
| $1152 = 3^2.2^7$ | 1 |
| $1536 = 3.2^9$ | 3 |
| $2048 = 2^{11}$ | 1 |
| $3072 = 3.2^{10}$ | 2 |

| $\|Aut\|$ | spr |
|---|---|
| $3840 = 5.3.2^8$ | 1 |
| $4608 = 3^2.2^9$ | 1 |
| $6144 = 3.2^{11}$ | 1 |
| $11520 = 5.3^2.2^8$ | 1 |
| $12288 = 3.2^{12}$ | 1 |
| $18432 = 3^2.2^{11}$ | 1 |
| All | 87 |

| cov5 | Sp | lSp | cov3 | $\|Aut\|$ | spr |
|---|---|---|---|---|---|
| 5 | 1 | 1 | 21 | 1 - 6144 | 62 |
| 6 | 1 | 1 | 21 - 22 | 4 - 48 | 8 |
| 9 | 2 | 1 | 21 - 37 | 32 - 4608 | 9 |
| 21 | 21 | 1 | 21 - 85 | 384 - 18432 | 8 |
| All | | | | | 87 |

**Таблица 4.14:** $(7,2,5,3)_2$ **книжни спредове от тип** 99999

| $\|Aut\|$ | spr |
|---|---|
| 1 | 86 |
| 2 | 26 |
| 3 | 17 |
| $4 = 2^2$ | 19 |
| $6 = 3.2^2$ | 14 |

| $\|Aut\|$ | spr |
|---|---|
| $8 = 2^3$ | 15 |
| $9 = 3^2$ | 1 |
| $12 = 3.2^2$ | 7 |
| $16 = 2^4$ | 4 |
| $18 = 3^2.2$ | 2 |

| $\|Aut\|$ | spr |
|---|---|
| $24 = 3.2^3$ | 5 |
| $32 = 2^5$ | 6 |
| $36 = 3^2.2^2$ | 2 |
| $72 = 3^2.2^3$ | 2 |
| $80 = 5.2^4$ | 1 |

| $\|Aut\|$ | spr |
|---|---|
| $96 = 3.2^5$ | 2 |
| $240 = 5.3.2^4$ | 1 |
| $360 = 5.3^2.2^3$ | 1 |
| All | 211 |

| cov5 | Sp | lSp | cov3 | $\|Aut\|$ | spr |
|---|---|---|---|---|---|
| 5 | 1 | 0 | 21 - 25 | 1 - 360 | 211 |
| All | | | | | 211 |

**Таблица 4.15:** $(7,2,5,3)_{1+2}$ **книжни спредове**

а) различни типове, $x = 1, 2, \ldots 9$

| $\lvert Aut\rvert$ | $1111x$ | $1112x$ | $1122x$ | 12222 | 22222 | spr |
|---|---|---|---|---|---|---|
| 1 | 7 | 512 | 1954 | 40 | 8 | 2521 |
| 2 | 67 | 254 | 418 | 37 | 36 | 812 |
| 3 | 16 | 228 | 573 | 40 | | 857 |
| $4 = 2^2$ | 58 | 653 | 1203 | 27 | 6 | 1947 |
| $6 = 3.2$ | 86 | 197 | 123 | 51 | 32 | 489 |
| $8 = 2^3$ | 6 | 468 | 398 | 40 | 20 | 932 |
| $9 = 3^2$ | 2 | 4 | 4 | | | 10 |
| $12 = 3.2^2$ | 61 | 753 | 380 | 6 | 4 | 1204 |
| $16 = 2^4$ | 67 | 118 | 216 | | | 401 |
| $18 = 3^2.2$ | 53 | 38 | 9 | 14 | 16 | 130 |
| $24 = 3.2^3$ | 46 | 468 | 115 | 34 | 20 | 683 |
| $32 = 2^5$ | 109 | 73 | 71 | 3 | 4 | 260 |
| $36 = 3^2.2^2$ | 23 | 26 | 2 | 2 | 2 | 55 |
| $48 = 3.2^4$ | 102 | 376 | 84 | | | 562 |
| $54 = 3^3.2$ | 17 | 4 | | 3 | 4 | 28 |
| $64 = 2^6$ | 43 | 25 | 18 | | | 86 |
| $72 = 3^2.2^3$ | 41 | 147 | 9 | 20 | 8 | 225 |
| $96 = 3.2^5$ | 135 | 171 | 24 | 9 | 12 | 351 |
| $128 = 2^7$ | 56 | | | 1 | | 57 |
| $144 = 3^2.2^4$ | 12 | 25 | 2 | | | 39 |
| $192 = 3.2^6$ | 90 | 13 | 6 | | | 109 |
| $216 = 3^3.2^3$ | 19 | 27 | 2 | 5 | | 53 |
| $288 = 3^2.2^5$ | 51 | 51 | 3 | 7 | 11 | 123 |
| $384 = 3.2^7$ | 141 | | | 5 | | 146 |
| $576 = 3^2.2^6$ | 11 | 6 | | | 1 | 18 |
| $648 = 3^4.2^3$ | 3 | 4 | | | | 7 |
| $864 = 3^3.2^5$ | 10 | 13 | 1 | 1 | 4 | 29 |
| $1152 = 3^2.2^7$ | 79 | | | 7 | | 86 |
| $1728 = 3^3.2^6$ | | 2 | | | | 2 |
| $3456 = 3^3.2^7$ | 20 | | | 2 | | 22 |
| $6912 = 3^3.2^8$ | 2 | | | 1 | | 3 |
| All | | | | | | 12247 |

b) само от тип $1111x$

| $\lvert Aut\rvert$ | spr |
|---|---|
| $108 = 3^3.2^2$ | 3 |
| $162 = 3^4.2$ | 2 |
| $256 = 2^8$ | 15 |
| $432 = 3^3.2^4$ | 3 |
| $512 = 2^9$ | 9 |
| $768 = 3.2^8$ | 24 |
| $1024 = 2^{10}$ | 4 |
| $1536 = 3.2^9$ | 41 |
| $2048 = 2^{11}$ | 1 |
| $2304 = 3^2.2^8$ | 11 |
| $3072 = 3.2^{10}$ | 4 |
| $4608 = 3^2.2^9$ | 40 |
| $6144 = 3.2^{11}$ | 5 |
| $6480 = 5.3^4.2^4$ | 1 |
| $9216 = 3^2.2^{10}$ | 3 |
| $10368 = 3^4.2^7$ | 1 |
| $12288 = 3.2^{12}$ | 2 |
| $13824 = 3^3.2^9$ | 18 |
| $15360 = 5.3.2^{10}$ | 1 |
| $18432 = 3^2.2^{11}$ | 9 |
| $20736 = 3^4.2^8$ | 2 |
| $27648 = 3^3.2^{10}$ | 1 |
| $36864 = 3^2.2^{12}$ | 2 |
| $41472 = 3^4.2^9$ | 1 |
| $55296 = 3^3.2^{11}$ | 1 |
| $73728 = 3^2.2^{13}$ | 1 |
| $110592 = 3^3.2^{12}$ | 1 |
| $207360 = 5.3^4.2^9$ | 1 |
| $221184 = 3^3.2^{13}$ | 1 |
| $1105920 = 5.3^3.2^{13}$ | 2 |
| $1658880 = 5.3^4.2^{12}$ | 1 |
| $17.7.5^2.3^5.2^{13}$ | 1 |
| All | 212 |

| cov5 | Sp | lSp | cov3 | $\lvert Aut\rvert$ | spr |
|---|---|---|---|---|---|
| 5 | 1 | 1 | 41 - 89 | 1 - 18432 | 8921 |
| 5 | 1 | 2 | 77 - 89 | 24 - 4608 | 19 |
| 5 | 1 | 5 | 101 | 1152 - 18432 | 4 |
| 6 | 1 | 1 | 41 - 93 | 1 - 1728 | 2543 |
| 6 | 1 | 2 | 89 | 36 - 216 | 4 |
| 6 | 1 | 5 | 101 | 432 | 1 |
| 7 | 1 | 1 | 78 - 90 | 18 - 162 | 14 |
| 7 | 1 | 2 | 90 | 216 | 1 |
| 9 | 2 | 1 | 41 - 149 | 1 - 13824 | 741 |
| 9 | 2 | 2 | 89 | 216 - 13824 | 6 |
| 9 | 2 | 5 | 101 | 3456 - 13824 | 2 |
| 10 | 2 | 1 | 81 - 96 | 54 - 648 | 7 |

| cov5 | Sp | lSp | cov3 | $\lvert Aut\rvert$ | spr |
|---|---|---|---|---|---|
| 10 | 2 | 2 | 93 | 648 | 1 |
| 10 | 2 | 5 | 102 | 6480 | 1 |
| 13 | 3 | 1 | 102 - 150 | 648 - 10368 | 6 |
| 21 | 21 | 1 | 41 - 277 | 2 - 1105920 | 176 |
| 21 | 21 | 2 | 89 | 3456 - 13824 | 2 |
| 21 | 21 | 5 | 101 | 73728 - 1105920 | 3 |
| 22 | 21 | 1 | 129 | 648 | 1 |
| 25 | 22 | 1 | 198 - 246 | 20736 | 2 |
| 25 | 22 | 2 | 153 | 41472 | 1 |
| 25 | 22 | 6 | 117 | 207360 | 1 |
| 37 | 41 | 2 | 297 | 1658880 | 1 |
| 85 | 357 | 85 | 357 | 5922201600 | 1 |
| All | | | | | 12459 |

# Глава 5

# Разрешими дизайни и $t$-паралелизми на $PG(n,q)$

В тази глава са събрани задачи, които включват класификация (с точност до изоморфизъм) на резолюциите на даден дизайн или само на тези от тях, които са инвариантни относно избрана група от автоморфизми от прост ред. Част от резултатите касаят $t$-паралелизми на $PG(n,q)$. Последните могат да бъдат разглеждани като резолюции на дизайна от точките и $t$-мерните подпространства на $PG(n,q)$.

## 5.1 Дефиниции, означения и предварителни резултати

Резолюциите на комбинаторните дизайни имат разнообразни важни приложения (виж, например, [66], [83], [114], [135], [211], [222], [236]). Специален интерес представляват резолюциите на циклични дизайни, каквито разглеждаме в раздел 5.5, ортогоналните резолюции, за каквито става въпрос в раздел 5.6 и $t$-паралелизмите (резолюциите на дизайните от точките и $t$-мерните подпространства на $PG(d,q)$), каквито разглеждаме в раздел 5.7.

Интересът към $t$-спредовете и $t$-паралелизмите в $PG(n,q)$ възниква заради съответствието на транслационни равнини [81], но се оказва, че те имат още много връзки с други математически обекти, както и разнообразни приложения. Множествата от взаимно ортогонални паралелизми на $PG(3,q)$ са свързани с проективни равнини от ред $q(q+1)$ [42], [129], съответствието им на резолюции на Щайнерови системи води до криптографски приложения при анонимни $(2,q+1)$-прагови схеми [222], а в Теорията на кодирането $t$-паралелизмите се използват в конструкции на шумозащитни кодове с константна размерност, съдържащи лифтингови MRD кодове [83], [216].

Конструкция на паралелизми в $PG(d,2)$ е предложена от Zaicev, Zinoviev и Semakov [253] и независимо от тях от Baker [24], а на паралелизми в $PG(2^n-1,q)$ от Beutelspacher [28]. Множество конструкции на паралелизми в $PG(3,q)$ са дадени

от Denniston [77], [78], Johnson [126], [127], Johnson и Pomareda [134], Penttila и Williams [201].

Класификации с помощта на компютър и при предварително зададени автоморфизми са правени от Prince [203], [204], Stinson и Vanstone [223], Sarmiento [208] [209], Topalova и Zhelezova [238], [241], [240], Zhelezova [259]. Неотдавна Betten [27] класифицира всички паралелизми на $PG(3,3)$.

За отделяне на изоморфните решения при компютърни класификации със зададени автоморфизми може да се използва нормализатора на зададената група от автоморфизми [204], [208], [238], [241], [240], [259]. Идеята е следната:

Нека $G_D$ е групата от автоморфизми на дизайна, $G_c$ е зададената група от автоморфизми на изоморфните резолюции $R$ и $R'$. Нека

$$R' = \varphi R.$$

Избираме $\alpha \in G_c$. Тогава $\varphi R = \alpha\varphi R$ и съответно $R = \varphi^{-1}\alpha\varphi R$, т.е. $\varphi^{-1}\alpha\varphi$ също е автоморфизъм на $R$. Тогава $R$ е инвариантна, както относно $G_c$, така и относно $\varphi^{-1}G_c\varphi$. Тези две групи или съвпадат, или са спрегнати подгрупи на $G_D$. Ако

$$\varphi^{-1}G_c\varphi = G_c,$$

то $\varphi$ е от нормализатора $N(G_c)$ на $G_c$ в $G_D$, който се дефинира като

$$N(G_c) = \{g \in G_D \mid gG_cg^{-1} = G_c\}.$$

Нека $G$ е пълната група от автоморфизми на $R$. Ако $\varphi$ не е от нормализатора $N(G_c)$, но $\varphi^{-1}G_c\varphi$ и $G_c$ са спрегнати подгрупи на $G$ (това винаги е изпълнено ако $G_c$ е Силова подгрупа на $G_D$), то понеже $\varphi$ не е автоморфизъм на $R$, трябва да съществува автоморфизъм $\psi$ на $R$, за който

$$\varphi^{-1}G_c\varphi = \psi G_c\psi^{-1}.$$

Тогава $G_c = \varphi\psi G_c\psi^{-1}\varphi^{-1}$ и следователно $\varphi\psi \in N(G_c)$. Понеже $\varphi\psi R = \varphi R$, то за да установим дали са изоморфни резолюциите $R$ и $R'$, е достатъчно да проверим дали съществува пермутация от нормализатора $N(G_c)$, която изобразява $R$ в $R'$.

Тук трябва да отбележим, че по този начин пълната група от автоморфизми на резолюциите не може да бъде намерена в процеса на тяхното конструиране и тя не е определена в класификациите на паралелизми със зададени автоморфизми на някои автори, например [203], [204], [208] и [209]. В раздел 5.7 от настоящата дисертация за целта са използвани алгоритмите, описани в раздели 5.3 и 5.4.

Групата от автоморфизми е една и съща за проективното пространство и всички свързани с него дизайни, затова за намиране на пораждащо множество от пермутации за нея и някои подгрупи, нужни за направените класификации, използваме симетричния дизайн от точките и хиперравнините, който има най-малко блокове от всички свързани с $PG(n,q)$ дизайни и следователно тези пресмятания стават най-бързо на него. За намиране на нормализатора $N(G_c)$ и някои от необходимите подгрупи използваме GAP [96].

## 5.2 Алгоритъм за класификация на резолюции със зададени автоморфизми от прост ред

### 5.2.1 Цели

Класификация с точност до изоморфизъм на всички резолюции на даден дизайн, които притежават зададен автоморфизъм $\alpha$ от прост ред $p$

### 5.2.2 Множество на търсене

То се състои от всички блокове на дизайна, които предварително сортираме в лексикографски ред. Има две възможности за дължината на орбитата на един паралелен клас спрямо $\alpha$: дължина $p$ или дължина 1 (фиксиран паралелен клас).

### 5.2.3 Конструиране на паралелните класове

Класовете строим блок по блок с изчерпващо търсене с връщане (раздел 1.2.2). Ако към текущия паралелен клас сме добавили $n$ блока, то $n+1$-вия избираме между блоковете, които:

- съдържат най-малката точка, която не е в никой от вече избраните блокове на текущия клас;
- не съдържат точки от вече избраните блокове на класа.

Дотук подходът е като при алгоритъма от раздел 1.2.9. Наличието на автоморфизма $\alpha$ налага и **допълнителни конструктивни условия** за орбитите на блоковете от един и същ паралелен клас:

- Всеки два блока от нефиксиран паралелен клас трябва да са от различни блокови орбити на $\alpha$.
- Ако един блок е във фиксиран паралелен клас, то и всички блокове от неговата орбита спрямо $\alpha$ трябва да бъдат в този паралелен клас.

Като имаме предвид дали текущо строеният паралелен клас е фиксиран или не, преди да добавим нов блок към него, ние проверяваме дали едното от горните две условия е в сила за него и всеки от вече добавените към паралелния клас блокове.

### 5.2.4 Тест за минималност на частичните решения

Този тест прилагаме след построяване изцяло на всеки пореден паралелен клас. С него проверяваме дали някой от автоморфизмите на дизайна не изобразява частичното решение в лексикографски по-малко. Ако го изобразява в себе си, то той е автоморфизъм на резолюцията.

Ако броят на автоморфизмите на дизайна не е прекалено голям и можем да направим тази проверка за всеки от тях, то по този начин в процеса на конструиране определяме и пълната група от автоморфизми на резолюцията. При много голяма група от автоморфизми на дизайна, обаче, тестът за минималност проверява дали конструираните резолюции с даден автоморфизъм се изобразяват в лексикографски по-малки само от автоморфизми от нормализатора $N$ на предварително зададената група. В такъв случай се налага допълнително определяне на пълната група от автоморфизми на резолюцията. Използваните алгоритми са представени в следващите две части.

## 5.3 Алгоритъм за намиране на пълната група от автоморфизми на резолюция

Този алгоритъм представлява модификация на алгоритъма за намиране на автоморфизмите на дизайн, описан в раздел 1.2.8. Разликите касаят основно функцията SameDes, която заменяме със SameParallel. Функцията SameParallel проверява дали дадената пермутация на точките изобразява първите $n$ реда от матрицата на инцидентност на всеки паралелен клас в тази на друг паралелен клас.

```
bool SameParallel(int n)
{
	int j, k;
	bool done[R+1];
	for(j=1; j<=R; j++) done[j] = false;
	for(j=1; j<=R; j++)
	{
		for(k=1; k<=R; k++) if(!done[k]) if(SameClasses(j, k, n)) break;
		if(k>R) return false;
		else done[k] = true;
	}
	 return true;
}
```

Тук $R$ е броят на паралелните класове, а функцията SameClasses(j,k,n) проверява дали първите $n$ реда от матрицата на инцидентност на $j$-я паралелен клас се изобразяват в първите $n$ реда от тази на $k$-я паралелен клас.

В няколко от разглежданите случаи този алгоритъм се оказва твърде бавен за разглежданите параметри. Тогава или използваме алгоритъма от следващата част, или настоящия, но на паралелния компютър BlueGene, където е възможно компютърът да работи едновременно по намирането на групите от автоморфизми на 512 резолюции. Подходът ни е следният: Всеки от 512-тте процесора има свой номер $0 \leq a < 512$ и всяка резолюция е с пореден номер $0 \leq b < m$. Всеки

процесор прочита всяка резолюция, но намира нейната група от автоморфизми само ако $b \equiv a \ (mod\ 512)$.

## 5.4 Алгоритъм за намиране на пълната група от автоморфизми на резолюция с даден автоморфизъм

При много голяма група от автоморфизми на дизайна тестът за минималност проверява дали конструираните резолюции с даден автоморфизъм се изобразяват в лексикографски по-малки само от автоморфизми от нормализатора $N$ на предварително зададената група. Затова пълната група от автоморфизми на намерените резолюции не може да бъде определена в процеса на конструирането им. В настоящия раздел са описани разработените алгоритми за нейното намиране, които се основават на генерирането на спрегнати на зададения автоморфизъм.

Как да изброим класовете от спрегнати подгрупи на голяма група и да намерим по един представител от всеки клас е въпрос, разглеждан в редица статии, например [49], [51], [52], [88], [116], [188], [246]. Но за нашите цели е необходим не по един представител от всеки клас, а всички елементи на даден клас. В софтуерната система за компютърна алгебра GAP [96] има функция, която ги намира, но съхранението на всички спрегнати елементи изисква много памет. Това прави тази функция неизползваема, ако групата от автоморфизми е твърде голяма. По-нататъшните ни разглеждания показват, че в нашия случай можем да използваме само част от спрегнатите елементи (необходимо множество). Тук са описани алгоритмите за тяхното генериране и използване.

### 5.4.1 Спрегнатите елементи в групата от автоморфизми

Разглеждаме резолюция $R$, която е инвариантна относно подгрупата $G_c$ на групата от автоморфизми $G_D$ на дизайна. Нека $R$ има автоморфизъм

$$\alpha \in G, \qquad \alpha \notin N(G_c),$$

където $G$ е пълната група от автоморфизми на $R$, а $N(G_c)$ е нормализаторът на $G_c$ в $G_D$. Резолюцията $R$ е инвариантна и относно $\alpha^{-1}G_c\alpha$.

Нека $G_{N+C}$ е групата от автоморфизми на $R$, съдържаща всички елементи от $N(G_c)$, които са автоморфизми на $R$ и всички подгрупи на $G_D$, които са спрегнати на $G_c$ и се състоят от автоморфизми на $R$. Нека

$$\alpha \notin G_{N+C}.$$

От дефиницията на $G_{N+C}$ следва, че $G_c$ и $\alpha^{-1}G_c\alpha$ са подгрупи на $G_{N+C}$. Ако $G_c$ и $\alpha^{-1}G_c\alpha$ са спрегнати подгрупи на $G_{N+C}$ (това винаги е така ако $G_c$ е Силова подгрупа на $G_D$), то съществува $\beta \in G_{N+C}$ такава, че $\alpha^{-1}G_c\alpha = \beta^{-1}G_c\beta$ и $G_c = \alpha\beta^{-1}G_c\beta\alpha^{-1}$, т.е. $\alpha\beta^{-1} \in N(G_c)$. Нека $\alpha\beta^{-1} = \gamma \in N(G_c)$. Тогава

$$\alpha = \gamma\beta, \text{ където } \gamma \in N(G_c) \text{ и } \beta \in G_{N+C}.$$

Затова можем да използваме следния алгоритъм:

**Основен алгоритъм** за намиране на групата от автоморфизми $G$ на резолюция, ако знаем Силова подгрупа $G_c$ на $G$:

1. Намираме $G_N$ - подгрупата на $N(G_c)$, относно която $R$ е инвариантна.
2. Намираме подгрупите на $G_D$, които са спрегнати на $G_c$ и запазват $R$.
3. Ако никоя спрегната на $G_c$ не запазва $R$, то пълната група от автоморфизми на $R$ е $G_N$ и в такъв случай пропускаме следващите три стъпки.
4. Намираме групата $G_{N+C}$, породена от генераторите на $G_N$ и подгрупите на $G_D$, които са спрегнати на $G_c$ и запазват $R$.
5. За всички $\beta \in G_{N+C}$ и $\gamma \in N(G_c)$ проверяваме дали $\beta\gamma$ е автоморфизъм на $R$ и ако е така, го съхраняваме.
6. Намираме пълната група от автоморфизми $G$, която се поражда от $G_{N+C}$ и автоморфизмите, намерени на стъпка 5.

Ако $G_c$ е Силова подгрупа на $G_D$, то тя е Силова подгрупа и на $G_{N+C}$ и тогава пълната група от автоморфизми $G$ може да бъде намерена с този алгоритъм. С малко повече внимание, обаче, той може да бъде използван и в случаи, когато $G_c$ не е Силова подгрупа на $G_D$. Най-голям е винаги броят на резолюциите, които са инвариантни само относно $G_c$. За тях алгоритъмът приключва на стъпка 3, а дотам няма изисквания за $G_c$. Условието $G_c$ да е Силова подгрупа на $G_{N+C}$ е лесно проверимо и обикновено не е в сила само за малък брой резолюции с доста богати групи от автоморфизми. В такъв случай е възможно алгоритъмът да намери не $G$, а нейна подгрупа $G'$. Тогава може да се опита същия алгоритъм, но като вместо $G_c$ се използва друга подгрупа на $G'$.

Ако $G_c$ е циклична група от прост ред, то всяка спрегната на $G_c$ подгрупа на $G_D$ се генерира от една от спрегнатите пермутации на дадена пермутация $c \in G_c$. Поради това стъпки 2, 3 и 4 на основния алгоритъм могат да бъдат заменени със следните:

2. Вземаме произволна пермутация $c \in G_c$ и проверяваме дали $R$ е инвариантна относно някоя от спрегнатите ѝ пермутации.
3. Ако никоя спрегната пермутация на $c$ не запазва $R$, то пълната група от автоморфизми на $R$ е $G_N$ и в такъв случай пропускаме следващите три стъпки.
4. Намираме групата $G_{N+C}$, генерирана от пораждащите на $G_N$ и спрегнатите пермутации на $c$, които запазват $R$

Ако $G_c$ не е циклична група от прост ред, можем да използваме някоя от нейните подгрупи от прост ред. За стъпка 2 са необходими всички спрегнати пермутации на пермутация от прост ред $c \in G_D$.

### 5.4.2 Конструиране на спрегнати пермутации

Нека $S_g = \{g_1, g_2, \dots g_m\}$ е генериращо множество на $G_D$. Означаваме с $C_c$ централизатора на $c$ в $G_D$, а с $n$ броя на спрегнатите пермутации на $c \in G_D$, където $n = |G_D|/|C_c|$. Построяваме спрегнатите пермутации на няколко рунда. Преди рунд 1 множеството от конструираните спрегнати пермутации съдържа един елемент – пермутацията $c$. На рунд $r$ към спрегнатите пермутации, конструирани на $r-1$-я рунд, се добавят нови, като се използва $S_g$. Алгоритъмът спира след рунд $r$, ако нито една нова спрегната пермутация не е добавена на този рунд.

**Алгоритъм 1: Конструиране на всички спрегнати пермутации**

Този алгоритъм намира множеството $S_{all}$ от всички $n$ спрегнати пермутации. На рунд $r$ се изпълнява следното:

for each $\delta$ which was added to $S_{all}$ at round $r-1$
{
  for $j = 1, 2, \dots m$
  {
    construct $\delta_j = g_j^{-1} \delta g_j$
    if $\delta_j \notin S_{all}$ then add $\delta_j$ to $S_{all}$.
  }
}

**Лема 5.4.1.** *Алгоритъм 1 генерира всички спрегнати пермутации на $c \in G_D$, ако $c$ е от прост ред.*

**Доказателство.** Нека $\alpha^{-1} c \alpha$ е спрегната пермутация на $c$ и нека $\alpha = g_{i_1} g_{i_2} \dots g_{i_s}$. Тогава $g_{i_1}^{-1} c g_{i_1}$ ще бъде добавена към $S_{all}$ на рунд 1, $g_{i_2}^{-1} g_{i_1}^{-1} c g_{i_1} g_{i_2}$ на рунд 2 и $g_{i_s}^{-1} \dots g_{i_2}^{-1} g_{i_1}^{-1} c g_{i_1} g_{i_2} \dots g_{i_s}$ на рунд $s$. □

### 5.4.3 Спрегнати пермутации, които можем да не конструираме

Нашата цел е да намерим пълната група от автоморфизми $G$ на резолюция $R$ с автоморфизъм $c$ от прост ред $p$ по горния метод. Всъщност ни трябва $G_{N+C}$. Затова трябва да построим групата, генерирана от всички спрегнати пермутации на $c$, които са автоморфизми на $R$. Нека $\delta$ е спрегната пермутация на $c$. Понеже $\delta^k$ запазва $R$ тогава и само тогава, когато $\delta$ запазва $R$, множеството от спрегнатите пермутации, които ни трябват, може да съдържа само по една от степените на всяка спрегната пермутация. Освен това $c^{-i} \delta c^i$ е автоморфизъм на $R$ тогава и само тогава, когато $\delta$ е автоморфизъм на $R$.

***Необходимо множество от спрегнати пермутации*** на $c$ ще наричаме множеството от спрегнати пермутации, за които:

– Ако $\delta$ е спрегната пермутация на $c$ и $\delta \notin S$, то $\exists\ k, i$ такива, че $c^{-i} \delta^k c^i \in S$.

– Ако $\delta \in S$, то $c^{-i} \delta^k c^i \notin S$ за $k = 2, \dots p-1, i = 1, \dots p-1$.

**Лема 5.4.2.** *Разглеждаме резолюция $R$ с автоморфизъм $c$ от прост ред $p$. Групата, генерирана от всички спрегнати пермутации на $c$, които са автоморфизми на $R$, може да бъде намерена ако разполагаме с необходимото множество $S$ от спрегнати пермутации на $c$.*

**Доказателство.** Ако $\delta$ запазва $R$ и $\delta \notin S$, то по дефиницията за необходимо множество $\exists\, k, i$ такива, че $c^{-i}\delta^k c^i \in S$. Ако $\delta$ запазва $R$, то $c^{-i}\delta^k c^i$ също я запазва. Понеже $c$ е автоморфизъм на $R$, $\delta^k$ също е автоморфизъм и $\delta \in \langle c^{-i}\delta^k c^i \rangle$. □

Проверката дали една спрегната пермутация запазва $R$ или не, е сравнително сложна и затова е от голямо значение да използваме необходимото множество вместо много по-голямото множество от всички спрегнати пермутации. Ако вече сме конструирали множеството $S_{all}$ от всички спрегнати пермутации (например с Алгоритъм 1), тогава необходимото множество $S$ можем да намерим с обикновен последователен тест:

$S = \emptyset$
for each $\delta \in S_{all}$
{
    if $New(\delta)$ add $\delta$ to $S$.
}

Функцията $New$ връща true ако $c^{-i}\delta^k c^i \notin S$ за $i = 0, 1, \ldots p-1$ и $k = 1, \ldots p-1$.

Алгоритъм 1 е полиномиален ($O(n^2)$), но има нужда от много памет. Ако няма достатъчно памет за Алгоритъм 1, можем да използваме следния Алгоритъм 2:

**Алгоритъм 2: Конструиране на необходимо множество от спрегнати пермутации**

Това е модификация на Алгоритъм 1, която намира необходимо множество от спрегнати пермутации $S \subseteq S_{all}$. На рунд $r$ се извършва следното:

for each $\delta$ which was added to $S$ at round $r-1$
{
    for $j = 1, 2, \ldots m$
    {
        for $i = 0, 1, \ldots p-1$
        {
            construct $\delta_{i,j} = g_j^{-1} c^{-i} \delta c^i g_j$ (1)
            if New($\delta_{i,j}$) then add $\delta_{i,j}$ to $S$.
        }
    }
}

Функцията $New$ връща положителен отговор, ако $c^{-i}\delta_{i,j}^k c^i \notin S$ за всяко $i = 0, 1, \ldots p-1$ и всяко $k = 1, \ldots p-1$.

**Лема 5.4.3.** *Алгоритъм 2 генерира необходимо множество $S$ от спрегнати пермутации на $c \in G_D$, където $c$ е от прост ред $p$.*

**Доказателство.** Алгоритъм 2 не запазва всички спрегнати пермутации (запазва само необходимо множество), но всъщност конструира всички. За да може всички спрегнати пермутации да бъдат конструирани на рунд $r$, намираме спрегнатите на всички (не само на запазените) спрегнати пермутации, които сме конструирали на рунд $r-1$.

Нека $\delta = \alpha^{-1}c\alpha$ е спрегната пермутация на $c$. Тогава

$$\delta^2 = \alpha^{-1}c\alpha\alpha^{-1}c\alpha = \alpha^{-1}c^2\alpha.$$

Следователно

$$\delta^k = \alpha^{-1}c^k\alpha.$$

Това означава, че $\delta$ и $\delta^k$ се *размножават* по един и същ начин на всеки рунд. Поради това наличието на само една от степените на всяка спрегната пермутация в необходимото множество не води до загуба на спрегнати пермутации.

Нека спрегнатата пермутация $\delta = \alpha^{-1}c\alpha$ да е запазена на рунд $r-1$, а спрегнатата пермутация

$$\rho = c^{-i}\alpha^{-1}c^k\alpha c^i$$

да е конструирана, но да не е запазена. От ред (1) на алгоритъма следва, че

$$c^{-i}\alpha^{-1}c\alpha c^i = \rho^{1-k}$$

ще се размножи на рунд $r$. Понеже $\rho$ и $\rho^{1-k}$ се размножават по-нататък по същия начин, няма загуба на спрегнати пермутации. □

### 5.4.4 Заключителни бележки

Нека намерим броя $n_n$ на спрегнатите пермутации в необходимото множество $S$.

$$n_n = n_c + n_r,$$

където $n_c$ е броя на спрегнатите пермутации от $S$, за които $G_c$ е от техния централизатор, а $n_r$ – на останалите. Нека построим от всяка спрегната пермутация $\delta \in S$ спрегнатите пермутации

$$c^{-i}\delta c^i$$

за $i = 0, 1, \ldots p-1$. Ще получим множество $S_a$ от

$$n_c + pn_r$$

спрегнати пермутации. Разглеждаме спрегната пермутация

$$\delta = \psi^{-1}c\psi$$

от $S_a$. От нея можем да получим спрегнатите пермутации

$$\psi^{-1}c^k\psi$$

за $k = k_1, k_2, \ldots k_h$, където $h$ е броят на степените на $c$, които са в един спрегнат клас с $c$. Така получаваме всички $n$ спрегнати пермутации на $c$ и

$$n = h(n_c + pn_r).$$

Понеже $n_c$ е малко в сравнение с $n_r$, ***броят на спрегнатите пермутации в необходимото множество е приблизително*** $hp$ ***пъти по-малък от броя на всички спрегнати пермутации в класа***.

Алгоритъм 2 е лесен за реализация и пести памет. Той може да се използва и в задачи, за които са необходими всички спрегнати пермутации, защото от всяка спрегната пермутация от необходимото множество $S$ можем директно да получим $ph$ или $h$ спрегнати пермутации, които не са от $S$ и това са всички спрегнати пермутации.

Алгоритмите от тази част са публикувани в [T12]. Освен в задачи от настоящата глава, те са използвани успешно в [238], [239], [240] и [241].

## 5.5 Резолюции на циклични дизайни с малки параметри

### 5.5.1 Въведение

Една резолюция е *точково циклична* (point-cyclic), ако притежава автоморфизъм, който пермутира точките в един цикъл. Само циклични дизайни могат да имат точково циклични резолюции. Един дизайн е *точково циклично разрешим* (point-cyclically resolvable), ако има най-малко една точково циклична резолюция.

Резолюции с богати групи от автоморфизми представляват особен интерес с оглед на някои приложения (Например, точково циклични резолюции на Щайнерови системи се използват в конструкции на някои видове шумозащитни кодове [104], [135]). Затова резолюциите на циклични дизайни са обект на множество статии.

Общи конструкции на резолюции на циклични дизайни са представени в [97], [159], [190], [191]. Цикличните $STS(v)$ са разрешими тогава и само тогава, когато $v \equiv 3 \pmod 6$ и $v \geq 15$. Известни са няколко компютърни класификации на $KTS(v)$ от циклични дизайни. Всички резолюции на $STS(15)$ [139] и на цикличните $STS(21)$ [179] са намерени. Точково циклични $KTS(21)$ и $KTS(39)$ са разгледани в [160], $KTS(33)$ от циклични дизайни и с автоморфизми от ред 11 са конструирани в [237], а точково цикличните $KTS(27)$ са част от конструираните в [68] транзитивни $KTS(27)$. Съществуват още много изследвания на $KTS(v)$ с $v \leq 39$, но на автора не са известни други класификационни резултати за останалите случаи на циклични $STS(v)$, които разглеждаме в този раздел.

Целта на настоящата работа е от една страна конструирането на нови $KTS(v)$, а от друга изследване на свойствата на всички известни резолюции на циклични $STS(v)$ с $v \leq 39$, както и събирането им и предоставянето на web-достъп до тях с оглед на възможни бъдещи приложения и теоретични изследвания.

### 5.5.2 Особености, които касаят конструктивния алгоритъм

Циклични 2-($v$,$k$,1) дизайни с малки параметри са класифицирани за първи път в [70]. Тази класификация е разширена за някои следващи параметри в [19]. Понеже всички необходими дизайни са налични, в настоящия раздел се интересуваме само от конструирането и изследването на техните резолюции.

За конструирането на всички неизоморфни резолюции използваме алгоритъма от раздел 1.2.9, а на неизоморфните резолюции със зададен автоморфизъм – модификация на алгоритъма от раздел 5.2. Модификацията е свързана с използването на следните ***допълнителни конструктивни условия***, наложени от зададения автоморфизъм $\alpha$ от прост ред $p$.

Понеже броят на паралелните класове е $(v-1)/2$, трябва да има най-малко

$$c_f = \frac{v-1}{2} \; mod \; p$$

паралелни класа, които са фиксирани от $\alpha$. Броят на блоковете в един паралелен клас е $v/3$ и следователно фиксиран паралелен клас трябва да съдържа най-малко

$$b_f = \frac{v}{3} \; mod \; p$$

фиксирани блока. Затова относно $\alpha$ трябва да има:

- най-малко $b_f c_f$ фиксирани блока и
- най-малко $c_f(\left\lfloor \frac{v}{3p} \right\rfloor + b_f)$ блокови орбити такива, че всяка точка се съдържа в не повече от един блок на орбитата.

Алгоритъмът започва с проверка на тези две условия и спира, ако не са в сила.

### 5.5.3 Класификация - типове според късата орбита на дизайна, точково циклични резолюции

#### Инварианти на резолюциите

Блоковете на циклична $STS(v)$ с $v \equiv 3 \pmod 6$ се разбиват от автоморфизма от ред $v$ на $(v-3)/6$ ***пълни*** орбити с дължина $v$ и една ***къса*** орбита с дължина $v/3$. В [190] са дефинирани три типа точково циклични резолюции на Щайнерови системи от тройки в зависимост от ролята на късата блокова орбита в тези резолюции. По подобен начин можем да дефинираме три типа резолюции (които могат и да не са точково циклични) по следния начин:

Една резолюция на циклична Щайнерова система от тройки е от тип

- $T_1$, ако късата орбита е паралелен клас във всички резолюции от класа на изоморфизъм.
- $T_2$, ако късата орбита не е паралелен клас в никоя резолюция от класа на изоморфизъм.
- $T_3$, ако късата орбита е паралелен клас в някои резолюции от класа на изоморфизъм и не е в останалите.

В настоящото изследване за всяка конструирана резолюция са определени типа, реда на пълната група от автоморфизми и орбитите на паралелните класове спрямо тази група. Изброените свойства са представени в таблици с по 8 колони. Първите четири съдържат инвариантите на построените резолюции:

- $autD$ – ред на пълната група от автоморфизми на дизайна
- $autR$ – ред на пълната група от автоморфизми на резолюцията
- $T$ – тип на резолюцията
- $orbR$ – брой на орбитите, на които пълната група от автоморфизми на резолюцията разбива паралелните класове

Броят на решенията със съответните инварианти е представен в последните четири колони:

- $D$ – брой на дизайните, от които са получени резолюциите
- $Dcyc$ – брой на тези измежду всичките $D$ дизайна, от които са получени точково циклични резолюции
- $R$ – брой на резолюциите с тези инварианти
- $Rcyc$ – брой на точково цикличните измежду всичките $R$ резолюции

Следват някои коментари върху свойствата на конструираните резолюции.

**KTS(15)**

Всичките седем $KTS(15)$ са добре познати [139]. Има две циклични $STS(15)$ с пълни групи от автоморфизми от редове 60 и 20160. Едната от тях е разрешима (Таблица 6.3.3.1) и това е тази $STS(15)$, която съответства на инцидентността на точките и правите на $PG(3,2)$. Тя има две неизоморфни резолюции, съответстващи на добре известните двойно-транзитивни паралелизми на $PG(3,2)$ [128]. Транзитивността, обаче, е върху паралелните класове ($orbR = 1$), докато тези резолюции не са точково циклични.

**Таблица 5.1: Резолюции на цикличните $STS(15)$**

| autD | autR | T | orbR | D | Dcyc | R | Rcyc |
|---|---|---|---|---|---|---|---|
| 20160 | 168 | 3 | 1 | 1 | 0 | 2 | 0 |

**KTS(21)**

Има 7 циклични $STS(21)$ с пълни групи от автоморфизми от редове 21, 42, 126(2), 504, 882 и 1008. Три от тях са разрешими с 26 неизоморфни резолюции, конструирани в [179] (Таблица 5.2). Две от тях са точково циклични и са намерени и в [160]. Още две резолюции имат група от автоморфизми от ред 21, но не са точково циклични.

**Таблица 5.2: Резолюции на цикличните STS(21)**

| autD | autR | T | orbR | D | Dcyc | R | Rcyc |
|---|---|---|---|---|---|---|---|
| 126 | 9 | 2 | 4 | 1 | 0 | 3 | 0 |
| 126 | 63 | 2 | 2 | 1 | 1 | 1 | 1 |
| 882 | 3 | 2 | 4 | 1 | 0 | 1 | 0 |
| 882 | 9 | 3 | 4 | 1 | 0 | 1 | 0 |
| 882 | 21 | 2 | 2 | 1 | 0 | 1 | 0 |
| 882 | 63 | 3 | 2 | 1 | 1 | 1 | 1 |
| 1008 | 1 | 1 | 10 | 1 | 0 | 10 | 0 |
| 1008 | 3 | 1 | 6 | 1 | 0 | 7 | 0 |
| 1008 | 21 | 1 | 4 | 1 | 0 | 1 | 0 |
| – | 1 | – | – | – | – | 10 | 0 |
| – | 3 | – | – | – | – | 8 | 0 |
| – | 9 | – | – | – | – | 4 | 0 |
| – | 21 | – | – | – | – | 2 | 0 |
| – | 63 | – | – | – | – | 2 | 2 |
| 126 | – | – | – | 1 | 1 | 4 | 1 |
| 882 | – | – | – | 1 | 1 | 4 | 1 |
| 1008 | – | – | – | 1 | 0 | 18 | 0 |
| – | – | – | – | 3 | 2 | 26 | 2 |

**KTS(27)**

Осемте циклични $STS(27)$ са с пълни групи от автоморфизми от ред 27. Четири от тях са разрешими и водят до 19336 неизоморфни резолюции (Таблица 5.3). Никоя от тях няма тривиална група от автоморфизми. Транзитивните $KTS(27)$ са конструирани в [68]. Сред тях са конструираните тук 4 точково циклични $KTS(27)$.

**Таблица 5.3: Резолюции на цикличните STS(27)**

| autD | autR | T | orbR | D | Dcyc | R | Rcyc |
|---|---|---|---|---|---|---|---|
| 27 | 3 | 1 | 9 | 2 | 0 | 8 | 0 |
| 27 | 3 | 1 | 11 | 2 | 0 | 64 | 0 |
| 27 | 3 | 1 | 13 | 2 | 0 | 112 | 0 |
| 27 | 3 | 2 | 9 | 3 | 0 | 390 | 0 |
| 27 | 3 | 2 | 11 | 2 | 0 | 5088 | 0 |
| 27 | 3 | 2 | 13 | 2 | 0 | 13648 | 0 |
| 27 | 9 | 1 | 7 | 2 | 0 | 4 | 0 |
| 27 | 9 | 2 | 3 | 1 | 0 | 2 | 0 |
| 27 | 9 | 2 | 5 | 2 | 0 | 16 | 0 |
| 27 | 27 | 1 | 3 | 2 | 2 | 4 | 4 |
| 27 | 3 | – | – | – | – | 19310 | 0 |
| 27 | 9 | – | – | – | – | 22 | 0 |
| 27 | 27 | – | – | – | – | 4 | 4 |
| 27 | – | – | – | 4 | 2 | 19336 | 4 |

**KTS(33)**

Има 84 циклични $STS(33)$ с пълни групи от автоморфизми от редове 33(78), 66(3), 165(2) и 330(1). Седемдесет и девет от тях притежават общо 28 резолюции с автоморфизми от ред 11 (които са конструирани в [237]), 141022 резолюции с автоморфизми от ред 3 и 703296 резолюции с автоморфизми от ред 2. Всички дизайни, които притежават резолюции с автоморфизми от ред 11, имат и резолюции с автоморфизми от ред 3, но нямат точково циклични резолюции (Таблица 5.4). Няма резолюции с автоморфизми от ред 5.

**Таблица 5.4: Резолюции на циклични STS(33) с нетривиални автоморфизми**

| autD | autR | T | orbR | D | Dcyc | R | Rcyc |
|---|---|---|---|---|---|---|---|
| 33 | 11 | 1 | 6 | 13 | 0 | 18 | 0 |
| 33 | 11 | 2 | 6 | 8 | 0 | 10 | 0 |
| 33 | 3 | 2 | 8 | 76 | 0 | 127621 | 0 |
| 33 | 3 | 1 | 6 | 11 | 0 | 13 | 0 |
| 66 | 3 | 2 | 8 | 3 | 0 | 13388 | 0 |
| 66 | 2 | 1 | 11 | 3 | 0 | 7646 | 0 |
| 66 | 2 | 2 | 11 | 3 | 0 | 695650 | 0 |
| – | 11 | – | – | – | – | 28 | 0 |
| – | 3 | – | – | – | – | 141022 | 0 |
| – | 2 | – | – | – | – | 703296 | 0 |
| 33 | – | – | – | 76 | 0 | 127662 | 0 |
| 66 | – | – | – | 3 | 0 | 716684 | 0 |
| – | – | – | – | 79 | 0 | 844346 | 0 |

**KTS(39)**

Цикличните $STS(39)$ са 798 и притежават пълни групи от автоморфизми от редове 39(730), 78(4), 117(55), 156(2), 234(4), 468(2) и 3042(1). Сред тях има 375 с резолюции с автоморфизми от ред 13 без фиксирани точки (Таблица 5.5). Тук трябва да обърнем внимание на това, че автоморфизъм от ред 13 с фиксирани точки (един от дизайните има такъв) не може да фиксира паралелен клас и поради това не може да бъде автоморфизъм на резолюция. Броят на резолюциите с автоморфизми от ред 13 е 2827. Сред тях има 528 точково циклични (конструирани и в [160]). Някои от останалите резолюции имат пълна група от автоморфизми от ред 39, но не са точково циклични. Получават се от дизайни с групи от автоморфизми от редове 117, 234 и 3042. Само резолюциите на дизайна с група от автоморфизми от ред 3042 са от тип $T_3$.

**Таблица 5.5: Резолюции на циклични STS(39) с автоморфизми от ред 13**

| autD | autR | T | orbR | D | Dcyc | R | Rcyc |
|---|---|---|---|---|---|---|---|
| 39 | 13 | 1 | 7 | 187 | 0 | 769 | 0 |
| 39 | 13 | 2 | 7 | 44 | 0 | 601 | 0 |
| 39 | 39 | 2 | 3 | 252 | 252 | 462 | 462 |
| 117 | 13 | 1 | 7 | 21 | 0 | 327 | 0 |
| 117 | 13 | 2 | 7 | 13 | 0 | 354 | 0 |
| 117 | 39 | 1 | 7 | 20 | 0 | 33 | 0 |
| 117 | 39 | 1 | 5 | 28 | 0 | 50 | 0 |
| 117 | 39 | 2 | 7 | 9 | 0 | 14 | 0 |
| 117 | 39 | 2 | 5 | 19 | 0 | 23 | 0 |
| 117 | 39 | 2 | 3 | 22 | 8 | 42 | 20 |
| 117 | 117 | 2 | 3 | 27 | 27 | 41 | 41 |
| 234 | 13 | 1 | 7 | 4 | 0 | 39 | 0 |
| 234 | 13 | 2 | 7 | 4 | 0 | 55 | 0 |
| 234 | 39 | 1 | 7 | 3 | 0 | 3 | 0 |
| 234 | 39 | 1 | 5 | 2 | 0 | 2 | 0 |
| 234 | 39 | 2 | 5 | 2 | 0 | 3 | 0 |
| 234 | 39 | 2 | 3 | 2 | 2 | 2 | 2 |
| 234 | 117 | 2 | 3 | 1 | 1 | 2 | 2 |
| 3042 | 13 | 3 | 7 | 1 | 0 | 3 | 0 |
| 3042 | 39 | 3 | 7 | 1 | 0 | 1 | 0 |
| 3042 | 117 | 3 | 3 | 1 | 1 | 1 | 1 |
| – | 13 | – | – | – | – | 2148 | 0 |
| – | 39 | – | – | – | – | 635 | 484 |
| – | 117 | – | – | – | – | 44 | 44 |
| 39 | – | – | – | 322 | 252 | 1832 | 462 |
| 117 | – | – | – | 48 | 36 | 884 | 61 |
| 234 | – | – | – | 4 | 3 | 106 | 4 |
| 3042 | – | – | – | 1 | 1 | 5 | 1 |
| – | – | – | – | 375 | 292 | 2827 | 528 |

Резултатите, получени в тази част са в съответствие с всички предварително известни изследвания и конструкции на други автори. Файлове с конструираните резолюции могат да бъдат изтеглени от http://www.moi.math.bas.bg/∼ svetlana. Изследванията са публикувани в [T13].

## 5.6 Резолюции на $STS(21)$ с автоморфизми от ред 3 с 3 фиксирани точки и 7 фиксирани блока

### 5.6.1 Въведение

Всички Щайнерови системи от тройки от редове $v \leq 19$ са класифицирани [69], [150]. Поради това STS(21) са понастоящем Щайнеровите системи от тройки с най-малките параметри, които все още не са класифицирани. Освен това те са и Щайнеровите системи от тройки с най-малките параметри, за които все още не се знае дали сред тях съществуват двойно-разрешими.

Wilson доказва, че неизоморфните STS(21) с три подсистеми STS(7) са най-малко 2160980 [249]. В част 3.2 установяваме, че точният брой на тези дизайни е 2166351. От тях се получават 13036 неизоморфни Киркманови системи от тройки.

Класификации на други по-малки класове STS(21) са правени от няколко автори – Mathon, Phelps и Rosa [179], [180], Tonchev [231], Kapralov и Topalova [143]. В тези статии са конструирани общо 2049 неизоморфни STS(21) и 215 KTS(21). Класификацията на STS(21) с някои автоморфизми от ред 3, с автоморфизми от ред 2 и без автоморфизми е отворен проблем.

Резолюциите на STS($v$) се наричат Киркманови системи от тройки от ред $v$ ( KTS($v$) ). Всички Киркманови системи от тройки от ред 21 с нетривиални автоморфизми са конструирани в [62].

В настоящия раздел е представена класификацията на всички STS(21) с автоморфизми от ред 3 с 3 фиксирани точки и 7 фиксирани блока. Получените 2963 неизоморфни дизайна са класифицирани според тяхната разрешимост и реда на групата от автоморфизми. Сред тях има 28 разрешими дизайна с 40 Киркманови системи от тройки от ред 21. Нито един от разрешимите дизайни не е двойно-разрешим.

### 5.6.2 Построяване на дизайните и резолюциите им

Нека $D$ е 2-$(21, 3, 1)$ дизайн с автоморфизъм $\varphi$ от ред 3, фиксиращ 3 точки и 7 блока. Без загуба на общност задаваме $\varphi$ с пермутацията

$$(1, 2, 3)(4, 5, 6)(7, 8, 9)(10, 11, 12)(13, 14, 15)(16, 17, 18)(19)(20)(21)$$

на точките и пермутацията

$$(1, 2, 3)(4, 5, 6)...(61, 62, 63)(64)(65)(66)(67)(68)(69)(70)$$

на блоковете. Тогава матрицата на инцидентност на $D$ е от вида:

$$
\begin{array}{l}
a_{1,1}\ a_{1,2}\ a_{1,3}\ a_{1,4}\ a_{1,5}\ a_{1,6}\ a_{1,7}\ a_{1,8}\ a_{1,9}\ a_{1,10}\ \ldots\ a_{1,21}\ i\ o\ o\ o\ o\ o\ o \\
a_{2,1}\ a_{2,2}\ a_{2,3}\ a_{2,4}\ a_{2,5}\ a_{2,6}\ a_{2,7}\ a_{2,8}\ a_{2,9}\ a_{2,10}\ \ldots\ a_{2,21}\ o\ i\ o\ o\ o\ o\ o \\
a_{3,1}\ a_{3,2}\ a_{3,3}\ a_{3,4}\ a_{3,5}\ a_{3,6}\ a_{3,7}\ a_{3,8}\ a_{3,9}\ a_{3,10}\ \ldots\ a_{3,21}\ o\ o\ i\ o\ o\ o\ o \\
a_{4,1}\ a_{4,2}\ a_{4,3}\ a_{4,4}\ a_{4,5}\ a_{4,6}\ a_{4,7}\ a_{4,8}\ a_{4,9}\ a_{4,10}\ \ldots\ a_{4,21}\ o\ o\ o\ i\ o\ o\ o \\
a_{5,1}\ a_{5,2}\ a_{5,3}\ a_{5,4}\ a_{5,5}\ a_{5,6}\ a_{5,7}\ a_{5,8}\ a_{5,9}\ a_{5,10}\ \ldots\ a_{5,21}\ o\ o\ o\ o\ i\ o\ o \\
a_{6,1}\ a_{6,2}\ a_{6,3}\ a_{6,4}\ a_{6,5}\ a_{6,6}\ a_{6,7}\ a_{6,8}\ a_{6,9}\ a_{6,10}\ \ldots\ a_{6,21}\ o\ o\ o\ o\ o\ i\ o \\
111\ 111\ 111\ 000\ 000\ 000\ 000\ 000\ 000\ \ 000\ \ldots\ 000\ 0\ 0\ 0\ 0\ 0\ 0\ 1 \\
000\ 000\ 000\ 111\ 111\ 111\ 000\ 000\ 000\ \ 000\ \ldots\ 000\ 0\ 0\ 0\ 0\ 0\ 0\ 1 \\
000\ 000\ 000\ 000\ 000\ 000\ 111\ 111\ 111\ \ 000\ \ldots\ 000\ 0\ 0\ 0\ 0\ 0\ 0\ 1
\end{array}
$$

където $a_{i,j}, \quad i = 1, 2, \ldots, 6, \quad j = 1, 2, \ldots, 21$ са циркуланти от ред 3, $i = (1, 1, 1)^t$, $o = (0, 0, 0)^t$.

Нека с $m_{i,j}, \ i = 1, 2, 3, 4, 5, 6, \ j = 1, 2, \ldots, 21$ означим броя на единиците в ред/стълб на $a_{i,j}$. За матрицата $M = (m_{i,j})_{6\times 21}$ са в сила следните равенства:

$$\sum_{j=1}^{21} m_{i,j} = 9, \qquad i = 1, 2, \ldots, 6 \tag{5.1}$$

$$\sum_{j=1}^{3} m_{i,j} = 1, \quad \sum_{j=4}^{6} m_{i,j} = 1, \quad \sum_{j=7}^{9} m_{i,j} = 1, \qquad i = 1, 2, \ldots, 6 \tag{5.2}$$

$$\sum_{j=1}^{21} m_{i,j}^2 = 9, \qquad i = 1, 2, \ldots, 6 \tag{5.3}$$

$$\sum_{j=1}^{21} m_{i_1,j} m_{i_2,j} = 3, \qquad 1 \le i_1 < i_2 \le 6. \tag{5.4}$$

От (5.1) и (5.3) следва, че $m_{i,j}$ може да е или 0, или 1. Като отчетем и (5.2) и (5.4) виждаме, че $M$ може да бъде разширена по единствен начин до матрица на инцидентност на 2-(7,3,3) дизайн чрез добавяне на ред, който започва с девет 1-ци, следвани от дванадесет 0-ли.

Има 10 неизоморфни 2-(7,3,3) дизайна [182]. Чрез отрязване по всички възможни начини на един ред от всяка от 10-те матрици на инцидентност, се получават 21 нееквивалентни решения за $M$. В 20 от тези решения колоните, които имат две 1-ци могат да бъдат разпределени в три групи от по три колони (5.2) по единствен начин. Само за една от матриците има две възможности. По този начин получаваме 22 матрици, които после разширяваме до дизайни, като заменяме 1-те в тях с циркуланти от ред 3 (с една 1-ца във всеки ред/стълб) по всички възможни начини (виж раздел 1.2.5). Тогава добавяме фиксираната част от матрицата на инцидентност и така получаваме 2963 неизоморфни 2-(21,3,1) дизайна. Резолюциите на дизайните намираме с алгоритъма, описан в раздел 1.2.9.

### 5.6.3 Класификация

Построяваме 2963 неизоморфни дизайна, от които 28 са разрешими и водят до 40 резолюции. Три от дизайните (с пълна група от автоморфизми от ред 9) притежават и автоморфизъм от ред 3 без фиксирани точки и със 7 фиксирани блока и са конструирани и в [143]. Останалите са нови.

В Таблица 5.6 е представена класификация на конструираните дизайни според реда на пълната им група от автоморфизми и броя на резолюциите им, а в Таблица 5.7 са дадени групите от автоморфизми на конструираните резолюции. С $AutD$ означаваме реда на групата от автоморфизми, с $Des$ броя на дизайните, с $RDes$ броя на разрешимите между тях и с $Res$ броя на неизоморфните резолюции. Само една от всичките 40 резолюции е с тривиална група от автоморфизми, останалите са конструирани и в [62].

**Таблица 5.6: Ред на пълната група от автоморфизми на $STS(21)$ с автоморфизми от ред 3 с 3 фиксирани точки и 7 фиксирани блока**

| $AutD$ | 3 | 6 | 9 | 12 | 18 | 27 | 36 | 54 | 108 | Общо |
|---|---|---|---|---|---|---|---|---|---|---|
| $Des$ | 2868 | 46 | 25 | 4 | 14 | 3 | 1 | 1 | 1 | 2963 |
| $RDes$ | 17 | | 8 | | 2 | 1 | | | | 28 |
| $Res$ | 17 | | 13 | | 2 | 8 | | | | 40 |

**Таблица 5.7: Автоморфизми на $KTS(21)$**

| $AutD$ | $AutR$ | $Des$ | $Res$ |
|---|---|---|---|
| 3 | 1 | 1 | 1 |
| 3 | 3 | 16 | 16 |
| 9 | 3 | 2 | 2 |
| 9 | 9 | 8 | 11 |
| 18 | 9 | 2 | 2 |
| 27 | 3 | 1 | 1 |
| 27 | 9 | 1 | 3 |
| 27 | 27 | 1 | 4 |

Базисните блокове на дизайните с ред на групата от автоморфизми по-голям от 12 са представени в Таблица 5.8, където фиксираните блокове са пропуснати.

Изследванията от тази част са публикувани в [T3].

**Таблица 5.8: Базисни блокове на някои дизайни**

| Des | $B_1$ | $B_4$ | $B_7$ | $B_{10}$ | $B_{13}$ | $B_{16}$ | $B_{19}$ | $B_{22}$ | $B_{25}$ | $B_{28}$ | $B_{31}$ | $B_{34}$ | $B_{37}$ | $B_{40}$ | $B_{43}$ | $B_{46}$ | $B_{49}$ | $B_{52}$ | $B_{55}$ | $B_{58}$ | $B_{61}$ | $AutD$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 7 | 13 | 1 | 7 | 13 | 1 | 7 | 13 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 4 | 4 | |
| 1 | 4 | 10 | 16 | 6 | 12 | 18 | 5 | 11 | 17 | 7 | 9 | 8 | 10 | 12 | 11 | 7 | 9 | 8 | 10 | 12 | 11 | 108 |
| | 19 | 19 | 19 | 20 | 20 | 20 | 21 | 21 | 21 | 13 | 14 | 15 | 16 | 17 | 18 | 16 | 17 | 18 | 13 | 14 | 15 | |
| | 1 | 7 | 13 | 1 | 7 | 13 | 1 | 7 | 13 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 4 | 4 | |
| 2 | 4 | 10 | 16 | 6 | 12 | 18 | 5 | 11 | 17 | 7 | 9 | 8 | 10 | 12 | 11 | 7 | 9 | 8 | 10 | 12 | 11 | 18 |
| | 19 | 19 | 19 | 20 | 20 | 20 | 21 | 21 | 21 | 13 | 14 | 15 | 16 | 17 | 18 | 18 | 16 | 17 | 13 | 14 | 15 | |
| | 1 | 7 | 13 | 1 | 7 | 13 | 1 | 7 | 13 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 4 | 4 | |
| 3 | 4 | 10 | 16 | 6 | 12 | 18 | 5 | 11 | 17 | 7 | 9 | 8 | 10 | 12 | 11 | 7 | 9 | 8 | 10 | 12 | 11 | 54 |
| | 19 | 19 | 19 | 20 | 20 | 20 | 21 | 21 | 21 | 13 | 14 | 15 | 18 | 16 | 17 | 17 | 18 | 16 | 13 | 14 | 15 | |
| | 1 | 7 | 13 | 1 | 7 | 13 | 1 | 7 | 13 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 4 | 4 | |
| 4 | 4 | 10 | 16 | 6 | 12 | 18 | 5 | 11 | 17 | 7 | 9 | 8 | 10 | 12 | 11 | 7 | 9 | 8 | 10 | 11 | 12 | 18 |
| | 19 | 19 | 19 | 20 | 20 | 20 | 21 | 21 | 21 | 13 | 14 | 18 | 15 | 16 | 17 | 14 | 18 | 16 | 13 | 15 | 17 | |
| | 1 | 7 | 13 | 1 | 7 | 13 | 1 | 7 | 13 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 4 | 4 | |
| 5 | 4 | 10 | 16 | 6 | 12 | 18 | 5 | 11 | 17 | 7 | 9 | 8 | 12 | 10 | 11 | 7 | 9 | 8 | 10 | 12 | 11 | 36 |
| | 19 | 19 | 19 | 20 | 20 | 20 | 21 | 21 | 21 | 13 | 14 | 16 | 15 | 18 | 17 | 14 | 16 | 17 | 15 | 13 | 18 | |
| | 1 | 7 | 13 | 1 | 7 | 10 | 1 | 7 | 10 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 4 | 4 | |
| 6 | 4 | 10 | 16 | 6 | 13 | 16 | 5 | 16 | 13 | 7 | 9 | 8 | 10 | 11 | 15 | 7 | 8 | 9 | 10 | 12 | 14 | 18 |
| | 19 | 19 | 19 | 20 | 20 | 20 | 21 | 21 | 21 | 12 | 13 | 18 | 14 | 16 | 17 | 11 | 13 | 17 | 15 | 16 | 18 | |
| | 1 | 7 | 13 | 1 | 7 | 10 | 1 | 7 | 10 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 4 | 4 | |
| 7 | 4 | 10 | 16 | 6 | 13 | 16 | 5 | 16 | 13 | 7 | 9 | 8 | 10 | 11 | 13 | 7 | 8 | 9 | 10 | 12 | 13 | 18 |
| | 19 | 19 | 19 | 20 | 20 | 20 | 21 | 21 | 21 | 12 | 14 | 18 | 15 | 16 | 17 | 11 | 15 | 17 | 14 | 16 | 18 | |
| | 1 | 7 | 13 | 1 | 7 | 10 | 1 | 7 | 10 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 4 | 4 | |
| 8 | 4 | 10 | 16 | 6 | 13 | 16 | 5 | 16 | 13 | 7 | 9 | 8 | 11 | 10 | 14 | 7 | 8 | 9 | 12 | 10 | 15 | 18 |
| | 19 | 19 | 19 | 20 | 20 | 20 | 21 | 21 | 21 | 12 | 13 | 16 | 15 | 17 | 18 | 11 | 13 | 16 | 14 | 18 | 17 | |
| | 1 | 4 | 13 | 1 | 7 | 10 | 7 | 10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 4 | 7 | |
| 9 | 7 | 10 | 16 | 4 | 13 | 16 | 14 | 18 | 6 | 5 | 9 | 8 | 12 | 13 | 15 | 7 | 9 | 8 | 12 | 11 | 12 | 18 |
| | 19 | 19 | 19 | 20 | 20 | 20 | 21 | 21 | 21 | 17 | 10 | 11 | 14 | 18 | 16 | 15 | 17 | 18 | 13 | 14 | 16 | |
| | 1 | 4 | 13 | 1 | 7 | 10 | 7 | 10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 4 | 7 | |
| 10 | 7 | 10 | 16 | 4 | 13 | 16 | 14 | 18 | 6 | 5 | 9 | 8 | 11 | 13 | 14 | 9 | 7 | 8 | 12 | 11 | 11 | 18 |
| | 19 | 19 | 19 | 20 | 20 | 20 | 21 | 21 | 21 | 18 | 12 | 10 | 15 | 17 | 16 | 14 | 16 | 18 | 15 | 13 | 18 | |
| | 1 | 4 | 13 | 1 | 4 | 10 | 7 | 1 | 10 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 7 | 7 | |
| 11 | 10 | 7 | 16 | 7 | 13 | 16 | 16 | 4 | 13 | 6 | 5 | 8 | 9 | 11 | 13 | 8 | 9 | 12 | 11 | 10 | 15 | 18 |
| | 19 | 19 | 19 | 20 | 20 | 20 | 21 | 21 | 21 | 14 | 16 | 12 | 15 | 18 | 17 | 10 | 17 | 14 | 16 | 14 | 17 | |
| | 1 | 4 | 13 | 1 | 4 | 10 | 7 | 1 | 10 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 7 | 7 | |
| 12 | 10 | 7 | 16 | 7 | 13 | 16 | 16 | 4 | 13 | 6 | 5 | 8 | 9 | 11 | 15 | 8 | 9 | 12 | 10 | 12 | 15 | 18 |
| | 19 | 19 | 19 | 20 | 20 | 20 | 21 | 21 | 21 | 14 | 17 | 12 | 13 | 18 | 16 | 11 | 17 | 14 | 18 | 13 | 17 | |
| | 1 | 4 | 13 | 1 | 4 | 10 | 7 | 1 | 10 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 7 | 7 | |
| 13 | 10 | 7 | 16 | 8 | 13 | 17 | 16 | 4 | 13 | 6 | 5 | 9 | 7 | 12 | 14 | 8 | 9 | 10 | 11 | 10 | 13 | 18 |
| | 19 | 19 | 19 | 20 | 20 | 20 | 21 | 21 | 21 | 13 | 16 | 11 | 15 | 17 | 18 | 12 | 16 | 15 | 17 | 14 | 18 | |
| | 1 | 4 | 13 | 1 | 4 | 10 | 7 | 1 | 10 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 7 | 7 | |
| 14 | 10 | 7 | 16 | 8 | 13 | 17 | 16 | 4 | 13 | 6 | 5 | 9 | 7 | 12 | 15 | 8 | 9 | 10 | 12 | 11 | 13 | 18 |
| | 19 | 19 | 19 | 20 | 20 | 20 | 21 | 21 | 21 | 13 | 18 | 11 | 14 | 17 | 16 | 11 | 16 | 15 | 18 | 15 | 18 | |
| | 1 | 7 | 10 | 1 | 7 | 10 | 7 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 4 | 7 | |
| 15 | 4 | 16 | 13 | 6 | 17 | 14 | 14 | 18 | 10 | 5 | 9 | 8 | 12 | 15 | 14 | 8 | 7 | 10 | 12 | 14 | 11 | 18 |
| | 19 | 19 | 19 | 20 | 20 | 20 | 21 | 21 | 21 | 7 | 11 | 13 | 18 | 16 | 17 | 11 | 13 | 15 | 17 | 16 | 18 | |
| | 1 | 4 | 7 | 1 | 4 | 7 | 4 | 1 | 7 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 7 | 7 | 7 | |
| 16 | 10 | 16 | 13 | 12 | 18 | 15 | 10 | 13 | 16 | 4 | 6 | 5 | 11 | 15 | 14 | 12 | 11 | 15 | 12 | 10 | 11 | |
| | 19 | 19 | 19 | 20 | 20 | 20 | 21 | 21 | 21 | 7 | 8 | 9 | 18 | 16 | 17 | 13 | 14 | 17 | 14 | 18 | 17 | 18 |
| | 1 | 4 | 7 | 1 | 4 | 7 | 1 | 4 | 7 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 7 | 7 | 7 | |
| 17 | 10 | 13 | 16 | 15 | 17 | 10 | 18 | 10 | 14 | 4 | 6 | 5 | 12 | 11 | 14 | 12 | 11 | 15 | 12 | 11 | 13 | 27 |
| | 19 | 19 | 19 | 20 | 20 | 20 | 21 | 21 | 21 | 7 | 8 | 9 | 13 | 16 | 17 | 14 | 18 | 16 | 15 | 17 | 18 | |
| | 1 | 4 | 7 | 1 | 4 | 7 | 1 | 4 | 7 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 7 | 7 | 7 | |
| 18 | 10 | 13 | 16 | 13 | 16 | 10 | 18 | 12 | 15 | 4 | 6 | 5 | 12 | 11 | 15 | 11 | 10 | 15 | 12 | 11 | 14 | 27 |
| | 19 | 19 | 19 | 20 | 20 | 20 | 21 | 21 | 21 | 7 | 8 | 9 | 14 | 17 | 16 | 14 | 18 | 17 | 13 | 18 | 17 | |
| | 1 | 4 | 7 | 1 | 4 | 7 | 1 | 4 | 7 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 7 | 7 | 7 | |
| 19 | 10 | 13 | 16 | 15 | 17 | 10 | 17 | 12 | 13 | 4 | 6 | 5 | 12 | 11 | 14 | 11 | 10 | 15 | 12 | 11 | 15 | 27 |
| | 19 | 19 | 19 | 20 | 20 | 20 | 21 | 21 | 21 | 7 | 8 | 9 | 13 | 16 | 18 | 14 | 16 | 18 | 14 | 18 | 17 | |
| | 1 | 4 | 7 | 7 | 4 | 1 | 1 | 4 | 7 | 1 | 10 | 1 | 1 | 1 | 1 | 1 | 7 | 4 | 4 | 7 | 4 | |
| 20 | 13 | 16 | 10 | 18 | 15 | 12 | 17 | 11 | 14 | 15 | 13 | 4 | 6 | 5 | 9 | 11 | 12 | 8 | 10 | 11 | 13 | 18 |
| | 19 | 20 | 21 | 19 | 21 | 20 | 21 | 19 | 20 | 16 | 16 | 7 | 8 | 10 | 14 | 18 | 17 | 17 | 14 | 13 | 18 | |

## 5.7 2-паралелизми на $PG(5,2)$ с автоморфизъм от ред 31

### 5.7.1 Въведение

Транзитивност и двойна транзитивност при $t$-паралелизмите е разглеждана от Johnson [128] и от Johnson и Montinaro [132], които доказват, че съществуват само два двойно транзитивни 1-паралелизми. В [132] те показват, че въпросът за съществуването на транзитивни $t$-паралелизми в $PG(n,q)$ за $t>1$ е отворен. Sarmiento класифицира 2-паралелизмите на $PG(5,2)$, които са инвариантни относно циклична група от ред 63 [209], но никой от тях не е транзитивен. След публикуването в [T8] на резултатите от изследванията в настоящия раздел, Johnson и Montinaro определят структурата на групата от автоморфизми на транзитивен паралелизъм на $PG(n,q)$ [133] и доказват, че транзитивни $t$-паралелизми в $PG(n,q)$ могат да съществуват само за $t=1$, или за $t=2$ и $(n,q)=(5,2)$ или $(n,q)=(5,3)$.

Част от конструираните тук 2-паралелизми на $PG(5,2)$ имат пълна група от автоморфизми от ред 155 и са първият известен пример за транзитивни $t$-паралелизми при $t>1$.

Ортогоналност при $t$-паралелизмите е изследвана от Fuji-Hara [94] и Stinson и Vanstone [223], като последните намират множество от 6 взаимно ортогонални 1-паралелизми на $PG(5,2)$. Класифицираните от Sarmiento 2-паралелизми [209] са с общ спред, затова няма взаимно ортогонални между тях.

В настоящата глава са изследвани за взаимна ортогоналност 92-та транзитивни 2-паралелизми на $PG(5,2)$. Представено е множество от 10 взаимно ортогонални 2-паралелизми. Това е първият пример за ортогонални $t$-паралелизми при $t>1$.

### 5.7.2 Конструиране на паралелизмите

Разглеждаме $PG(5,2)$, т.е. проективно пространство с размерност $n=5$ и ред $q=2$. То е с 63 точки и 651 прави и има пълна групата от автоморфизми $G$, която е изоморфна на $GL(6,2)$ и е от ред $|G|=20158709760=31.7^2.5.3^4.2^{15}$.

Точките на $PG(5,2)$ определяме чрез съответствието им на 6-мерните ненулеви вектори $(v_1,v_2,v_3,v_4,v_5,v_6)$ над $GF(2)$. Подреждаме тези 85 вектора в нарастващ лексикографски ред и им даваме номера така, че $(1,0,0,0,0,0)$ съответства на точка номер 1, а $(1,1,1,1,1,1)$ – на номер 63. Тогава построяваме свързаните с проективното пространство дизайни (зададени от инцидентността на точките и $t$-мерните подпространства) и намираме генераторите на пълната им група от автоморфизми $G$.

Всеки 2-паралелизъм на $PG(5,2)$ се състои от 155 спреда с по 9 елемента. Дизайнът от точките и двумерните подпространства на $PG(5,2)$ е с параметри 2-$(63,7,15)$ и има $b=1395$ блока (съответстващи на двумерни подпространства). На резолюциите му (всяка със 155 паралелни класа с по 9 блока) съответстват

2-паралелизми.

Понеже 31 дели реда на $G$, но $31^2$ не го дели, всяка подгрупа от ред 31 е Силова. Всички Силови 31-подгрупи на $G$ са спрегнати и за да построим всички нееквивалентни $t$-паралелизми с автоморфизми от ред 31, без ограничение на общността можем да конструираме само тези, които са инвариантни спрямо една от подгрупите от ред 31 - ще я означим с $G_{31}$. Тя се поражда от елемента $\gamma$ (Фигура 5.2).

Когато резолюцията е инвариантна спрямо $G_{31}$, то тази група разбива паралелните ѝ класове на 5 орбити с дължина 31. За да построим такава резолюция е достатъчно да знаем по един паралелен клас (лидер) от всяка от петте орбити (фигура 5.1)

За отсяването на изоморфните 2-паралелизми с теста за минималност на частичните решения ни е необходим нормализаторът $N(G_{31})$ на $G_{31}$ в $G$, където $N(G_{31}) = \{g \in G \mid gG_{31}g^{-1} = G_{31}\}$. Той е подгрупа от ред 155 и се поражда от елементите $\gamma$ и $\varphi$ (Фигура 5.2).

С алгоритъма, описан в раздел 5.2, получаваме всички (12312) неизоморфни 2-паралелизми с автоморфизми от ред 31, а с алгоритъма от раздел 5.3 намираме реда на пълните групи от автоморфизмите им.

### 5.7.3 Намиране на множества от взаимно ортогонални паралелизми

Изследваме за взаимна ортогоналност транзитивните резолюции с автоморфизми от ред 155 на 2-$(63, 7, 15)$ дизайна от точките и 2-мерните подпространства на $PG(5, 2)$. Разглеждаме резолюциите в реда, в който сме ги построили. Преди търсенето за всяка от тях отбелязваме двойките блокове, участващи в един и същ паралелен клас.

Всяка резолюция до 91-вата избираме като начало на множество от ортогонални резолюции. За всяка резолюция след избраната за начало проверяваме дали притежава двойки блокове от един и същ паралелен клас, които да съвпадат с някои от записаните досега в множеството. Ако е така, тя не може да е ортогонална на текущото множество от ортогонални резолюции. Тази проверка продължава до 92-та резолюция, защото търсим максималните множества. За всяко получено множество проверяваме дали не се съдържа в някое, построено преди това с друга начална резолюция, защото тогава няма да е максимално.

Фигура 5.1: Конструиране на резолюции с $G_{31}$ на 2-$(63, 7, 15)$ дизайна от точките и 2-мерните подпространства на $PG(5, 2)$

а)паралелен клас, лидер на орбита

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 2 | 16 | 19 | 21 | 22 | 23 | 20 | 18 | 17 |
| 3 | 24 | 26 | 31 | 29 | 27 | 25 | 28 | 30 |
| 4 | 33 | 37 | 34 | 38 | 35 | 39 | 32 | 36 |
| 5 | 41 | 44 | 40 | 45 | 47 | 42 | 46 | 43 |
| 6 | 49 | 54 | 55 | 48 | 52 | 51 | 50 | 53 |
| 7 | 57 | 63 | 60 | 59 | 56 | 62 | 60 | 58 |

б) резолюция, представена с лидерите на орбити

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 2 | 16 | 19 | 21 | 22 | 23 | 20 | 18 | 17 |
| 3 | 24 | 26 | 31 | 29 | 27 | 25 | 28 | 30 |
| 4 | 33 | 37 | 34 | 38 | 35 | 39 | 32 | 36 |
| 5 | 41 | 44 | 40 | 45 | 47 | 42 | 46 | 43 |
| 6 | 49 | 54 | 55 | 48 | 52 | 51 | 50 | 53 |
| 7 | 57 | 63 | 60 | 59 | 56 | 62 | 60 | 58 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 4 | 5 | 6 | 7 | 8 | 11 | 13 | 15 |
| 2 | 27 | 24 | 10 | 9 | 20 | 21 | 23 | 22 |
| 3 | 31 | 29 | 12 | 14 | 28 | 30 | 26 | 25 |
| 16 | 33 | 41 | 34 | 50 | 35 | 38 | 39 | 32 |
| 17 | 37 | 44 | 36 | 53 | 43 | 45 | 42 | 47 |
| 18 | 58 | 49 | 40 | 59 | 55 | 51 | 48 | 54 |
| 19 | 62 | 52 | 46 | 60 | 63 | 56 | 61 | 57 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 4 | 5 | 6 | 7 | 8 | 9 | 13 | 15 |
| 2 | 25 | 11 | 10 | 17 | 20 | 19 | 18 | 23 |
| 3 | 29 | 14 | 12 | 22 | 28 | 26 | 31 | 24 |
| 40 | 35 | 16 | 49 | 34 | 36 | 38 | 32 | 33 |
| 41 | 39 | 21 | 55 | 37 | 44 | 47 | 45 | 46 |
| 42 | 58 | 27 | 59 | 51 | 48 | 53 | 50 | 54 |
| 43 | 62 | 30 | 61 | 52 | 56 | 60 | 63 | 57 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 4 | 5 | 6 | 7 | 8 | 11 | 12 | 13 |
| 2 | 10 | 17 | 9 | 18 | 23 | 22 | 16 | 19 |
| 3 | 14 | 20 | 15 | 21 | 31 | 29 | 28 | 30 |
| 24 | 32 | 40 | 50 | 34 | 33 | 39 | 35 | 38 |
| 25 | 36 | 45 | 52 | 37 | 41 | 44 | 47 | 43 |
| 26 | 42 | 57 | 59 | 48 | 54 | 49 | 51 | 53 |
| 27 | 46 | 60 | 61 | 55 | 62 | 58 | 63 | 56 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 4 | 5 | 6 | 7 | 10 | 11 | 13 | 15 |
| 2 | 8 | 24 | 17 | 9 | 16 | 20 | 19 | 22 |
| 3 | 12 | 29 | 23 | 14 | 26 | 31 | 30 | 25 |
| 36 | 48 | 42 | 40 | 18 | 33 | 34 | 32 | 35 |
| 37 | 52 | 47 | 46 | 21 | 43 | 41 | 45 | 44 |
| 38 | 56 | 50 | 57 | 27 | 49 | 54 | 51 | 53 |
| 39 | 60 | 55 | 63 | 28 | 59 | 61 | 62 | 58 |

### 5.7.4 Класификация - първите транзитивни $t$-паралелизми за $t > 1$.

Конструираните неизоморфни 2-паралелизми са 12312 на брой. Деветдесет и два от тях са инвариантни спрямо $N(G_{31})$. Те имат пълна група от автоморфизми от ред 155, която е транзитивна върху спредовете. Така получаваме първите и засега единствени примери за транзитивни $t$-паралелизми с $t > 1$. Останалите 2-паралелизми са с пълна група от автоморфизми от ред 31.

Резултатите от изследването на транзитивните паралелизми за ортогоналност са показани в таблица 5.9, а едно от множествата с 10 взаимно ортогонални 2-паралелизми, представени с по един спред всеки (понеже пълната им група е транзитивна върху спредовете), е дадено на фигура 5.2.

**Таблица 5.9: Множества от взаимно ортогонални 2-паралелизми с една и съща пълна група от автоморфизми от ред** 155

| | максимални | всички |
|---|---|---|
| 2-MOR | 0 | 2308 |
| 3-MOR | 0 | 20602 |
| 4-MOR | 27 | 72504 |
| 5-MOR | 2234 | 106322 |
| 6-MOR | 9263 | 67549 |
| 7-MOR | 6930 | 19242 |
| 8-MOR | 1413 | 2563 |
| 9-MOR | 98 | 192 |
| 10-MOR | 10 | 10 |

След горните резултати и теоретичните разглеждания на Johnson и Montinaro [133] въпросът за съществуването на транзитивни $t$-паралелизми в $PG(n, q)$ за $t > 1$ остава отворен само за $PG(5, 3)$.

Изследванията в тази част са съвместни със Стела Железова. Те са публикувани в [T8] и са отразени в дисертацията на Железова за присъждане на научната и образователна степен доктор [3].

Фигура 5.2: Десет взаимно ортогонални 2-паралелизми с група от автоморфизми $G_{155}$
Генератори на $G_{155}$:

$\gamma$=(1,34,9,50,13,49,45,58,29,53,46,26,22,5,33,41,57,61,62,30,21,37,42,25,54,14,17,38,10,18,6)
(2)(3,32,11,48,15,51,47,56,31,55,44,24,20,7,35,43,59,63,60,28,23,39,40,27,52,12,19,36,8,16,4)

$\varphi$=(1)(2)(3)(4,16,36,27,43)(5,17,37,26,42)(6,18,38,25,41)(7,19,39,24,40)(8,12,28,56,35)(9,13,29,57,34)
(10,14,30,58,33)(11,15,31,59,32)(20,52,63,48,47)(21,53,62,49,46)(22,54,61,50,45)(23,55,60,51,44)

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 1 | | | | | | | | | 2 | | | | | |
| 1 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 1 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 2 | 16 | 19 | 23 | 20 | 18 | 17 | 21 | 22 | 2 | 16 | 23 | 22 | 17 | 19 | 20 | 21 | 18 |
| 3 | 24 | 26 | 29 | 31 | 30 | 28 | 27 | 25 | 3 | 24 | 30 | 28 | 26 | 31 | 25 | 27 | 29 |
| 4 | 34 | 36 | 33 | 39 | 35 | 37 | 32 | 38 | 4 | 39 | 36 | 32 | 35 | 34 | 33 | 37 | 38 |
| 5 | 42 | 45 | 43 | 44 | 47 | 40 | 46 | 41 | 5 | 47 | 45 | 42 | 40 | 46 | 44 | 43 | 41 |
| 6 | 50 | 55 | 54 | 51 | 49 | 52 | 53 | 48 | 6 | 55 | 51 | 54 | 50 | 49 | 53 | 48 | 52 |
| 7 | 58 | 62 | 60 | 56 | 61 | 57 | 59 | 63 | 7 | 63 | 58 | 60 | 57 | 61 | 56 | 62 | 59 |
| | | | | 3 | | | | | | | | | 4 | | | | | |
| 1 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 1 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 2 | 17 | 19 | 20 | 22 | 23 | 21 | 18 | 16 | 2 | 17 | 21 | 18 | 22 | 19 | 23 | 16 | 20 |
| 3 | 25 | 26 | 30 | 29 | 27 | 24 | 28 | 31 | 3 | 25 | 28 | 24 | 29 | 31 | 26 | 30 | 27 |
| 4 | 38 | 33 | 39 | 32 | 35 | 36 | 34 | 37 | 4 | 39 | 33 | 35 | 37 | 38 | 32 | 34 | 36 |
| 5 | 46 | 40 | 45 | 43 | 47 | 41 | 44 | 42 | 5 | 47 | 40 | 41 | 46 | 42 | 45 | 44 | 43 |
| 6 | 55 | 50 | 51 | 54 | 52 | 49 | 48 | 53 | 6 | 54 | 52 | 49 | 51 | 53 | 55 | 50 | 48 |
| 7 | 63 | 59 | 57 | 61 | 56 | 60 | 62 | 58 | 7 | 62 | 61 | 59 | 56 | 57 | 58 | 60 | 63 |
| | | | | 5 | | | | | | | | | 6 | | | | | |
| 1 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 1 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 2 | 18 | 20 | 19 | 21 | 16 | 22 | 17 | 23 | 2 | 20 | 18 | 16 | 22 | 21 | 19 | 17 | 23 |
| 3 | 26 | 29 | 25 | 30 | 28 | 27 | 31 | 24 | 3 | 28 | 27 | 26 | 29 | 25 | 30 | 31 | 24 |
| 4 | 33 | 37 | 36 | 32 | 38 | 34 | 35 | 39 | 4 | 35 | 39 | 32 | 36 | 33 | 37 | 34 | 38 |
| 5 | 41 | 44 | 46 | 43 | 42 | 47 | 45 | 40 | 5 | 43 | 46 | 42 | 47 | 45 | 40 | 44 | 41 |
| 6 | 51 | 49 | 55 | 53 | 54 | 52 | 50 | 48 | 6 | 55 | 53 | 48 | 50 | 52 | 54 | 51 | 49 |
| 7 | 59 | 56 | 61 | 62 | 58 | 57 | 60 | 63 | 7 | 63 | 60 | 58 | 57 | 56 | 59 | 61 | 62 |
| | | | | 7 | | | | | | | | | 8 | | | | | |
| 1 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 1 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 2 | 22 | 20 | 17 | 19 | 21 | 23 | 18 | 16 | 2 | 23 | 21 | 18 | 16 | 22 | 20 | 19 | 17 |
| 3 | 30 | 29 | 27 | 24 | 25 | 26 | 28 | 31 | 3 | 31 | 28 | 24 | 27 | 26 | 25 | 29 | 30 |
| 4 | 39 | 35 | 36 | 32 | 37 | 33 | 38 | 34 | 4 | 33 | 38 | 39 | 32 | 34 | 37 | 36 | 35 |
| 5 | 47 | 42 | 46 | 43 | 41 | 44 | 40 | 45 | 5 | 41 | 47 | 45 | 43 | 46 | 40 | 42 | 44 |
| 6 | 49 | 55 | 53 | 51 | 48 | 54 | 52 | 50 | 6 | 54 | 51 | 53 | 48 | 52 | 49 | 55 | 50 |
| 7 | 57 | 62 | 63 | 56 | 60 | 59 | 58 | 61 | 7 | 62 | 58 | 63 | 59 | 56 | 60 | 57 | 61 |
| | | | | 9 | | | | | | | | | 10 | | | | | |
| 1 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 1 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 2 | 23 | 17 | 19 | 21 | 16 | 22 | 20 | 18 | 2 | 23 | 20 | 16 | 19 | 21 | 22 | 18 | 17 |
| 3 | 31 | 24 | 25 | 30 | 28 | 27 | 26 | 29 | 3 | 31 | 29 | 26 | 24 | 25 | 27 | 28 | 30 |
| 4 | 34 | 33 | 37 | 38 | 39 | 36 | 32 | 35 | 4 | 37 | 33 | 35 | 39 | 34 | 38 | 36 | 32 |
| 5 | 42 | 40 | 47 | 45 | 43 | 41 | 46 | 44 | 5 | 45 | 40 | 41 | 44 | 46 | 43 | 42 | 47 |
| 6 | 53 | 48 | 54 | 51 | 55 | 50 | 52 | 49 | 6 | 50 | 53 | 51 | 52 | 55 | 48 | 54 | 49 |
| 7 | 61 | 57 | 60 | 56 | 59 | 63 | 58 | 62 | 7 | 58 | 60 | 57 | 63 | 59 | 61 | 56 | 62 |

# Глава 6

# Циклично-пермутационни константно-тегловни кодове, циклични дизайни, антиконфликтни и силни антиконфликтни кодове

Настоящата глава обединява класификационни резултати за обекти, които са различни, но имат сходна циклична структура и множество връзки помежду си, което определя и сходния подход към класификационната задача в тези случаи. За краткост кодовете, разглеждани в тази глава, ще наричаме и с общото име ***кодове с циклична структура***.

## 6.1 Дефиниции и означения

**Дефиниция 6.1.1.** *Двоичният* $(v, k, \lambda)$ ***циклично-пермутационен константно-тегловен (CPCW) код*** $\mathcal{C}$ *е колекция от* $\{0,1\}$ *последователности с дължина* $v$ *и тегло по Хеминг* $k$*, такава, че:*

$$\sum_{i=0}^{v-1} x(i)x(i \oplus j) \leq \lambda,\ 1 \leq j \leq v-1 \tag{6.1}$$

$$\sum_{i=0}^{v-1} x(i)y(i \oplus j) \leq \lambda,\ 0 \leq j \leq v-1 \tag{6.2}$$

*за всички двойки различни последователности* $x, y \in \mathcal{C}$*.*

Всеки $(v, k, \lambda)$ CPCW код може да бъде дефиниран и по следния начин, който е по-удобен за нашите разглеждания:

**Дефиниция 6.1.2.** *Всеки* $(v, k, \lambda)$ *циклично-пермутационен константно-тегловен код е колекция* $\mathcal{C} = \{C_1, \ldots, C_s\}$ *от* $k$*-елементни подмножества (кодови ду-*

*ми) на $Z_v$, такива че всеки две различни транслации на кодова дума имат най-много $\lambda$ общи елемента и всеки две транслации на две различни кодови думи също имат най-много $\lambda$ общи елемента:*

$$|C_i \cap (C_i \oplus t)| \leq \lambda, \quad 1 \leq i \leq s, \quad 1 \leq t \leq v-1 \tag{6.3}$$
$$|C_i \cap (C_j \oplus t)| \leq \lambda, \quad 1 \leq i < j \leq s, \quad 0 \leq t \leq v-1 \tag{6.4}$$

Всеки CPCW код може да бъде разглеждан като $(v, k, \lambda, \lambda)$ оптичен ортогонален код (OOC) и затова следващите няколко понятия и дефиниции са аналогични на тези за OOC.

За всяка кодова дума $C = \{c_1, c_2, \ldots, c_k\}$ на CPCW код означаваме с $\triangle' C$ мултимножеството от стойности на разликите $c_i - c_j,\ i \neq j,\ i, j = 1,\ 2,\ \ldots,\ k$. Най-много $\lambda$ разлики са еднакви. Нека да означим с $\triangle C$ съответното на мултимножеството $\triangle' C$ множество. Ако $\lambda = 1$, то $\Delta C_1 \bigcap \Delta C_2 = \emptyset$ за две кодови думи $C_1$ и $C_2$ от $(v, k, 1)$ CPCW код.

**Дефиниция 6.1.3.** *$(v, k, 1)$ CPCW код е* ***съвършен****, ако*

$$|\bigcup_{i=1}^{s} \Delta C_i| = v - 1,$$

*което означава, че всички възможни ненулеви разлики са покрити.*

**Дефиниция 6.1.4.** *$(v, k, 1)$ Един CPCW код е $g$-регулярен ако непокритите разлики и нулата образуват адитивна подгрупа на $Z_v$ от ред $g$ и с генериращ елемент $v/g$. Съвършените кодове са 1-регулярни.*

**Дефиниция 6.1.5.** *Един CPCW код е максимален, ако има максималния възможен (за дадените параметри) брой кодови думи.*

**Дефиниция 6.1.6.** ***Разностна*** *$(v, k, \lambda)* ***фамилия*** *наричаме множество $\mathcal{C} = \{C_1, \ldots, C_s\}$, където $C_i = \{c_{i_1}, c_{i_2}, \ldots, c_{i_k}\}$ са $k$-елементни подмножества на $Z_v$, такива, че всеки ненулев елемент на $Z_v$ се среща точно $\lambda$ пъти сред разликите $c_{i_j} - c_{i_l}$ за $1 \leq i \leq s$, $1 \leq j \neq l \leq k$.*

**Дефиниция 6.1.7.** *Нека $V = \{P_i\}_{i=1}^{v}$ да е крайно множество от точки и $\mathcal{B} = \{B_j\}_{j=1}^{b}$ да е крайна колекция от $k$-елементни подмножества на $V$, наречени блокове. $D = (V, \mathcal{B})$ е* ***частичен дизайн*** *с параметри $t$-($v$,$k$,$\lambda$), ако всяко $t$-елементно подмножество на $V$ се съдържа в най-много $\lambda$ блока на $\mathcal{B}$.*

Частичните дизайни се наричат също *опаковки* [224] или *пакетиращи дизайни* [95]. Ние ще ги наричаме частични дизайни следвайки [60].

**Дефиниция 6.1.8.** *Един $t$-($v$,$k$,$\lambda$) дизайн (частичен дизайн) е* ***циклични****, ако съществува автоморфизъм $\alpha$, който пермутира точките му в един цикъл и строго циклични, ако всяка орбита относно този автоморфизъм е с дължина $v$ (няма къси орбити).*

Всеки $(v,k,\lambda)$ CPCW код съответства на строго цикличен $(\lambda+1)$-$(v,k,1)$ частичен дизайн и всеки строго цикличен $t$-$(v,k,1)$ частичен дизайн съответства на $(v,k,t-1)$-CPCW код за $\lambda \geq 1$ и $t \geq 2$ [60]. Матрицата на инцидентност на строго цикличен $t$-$(v,k,1)$ частичен дизайн съдържа подматрици, които са циркуланти от ред $v$. Всяка циркуланта съответства на съвкупността от една кодова дума на $(v,k,t-1)$-CPCW код и нейните транслации (фиг. 6.1).

**Дефиниция 6.1.9.** *Двоичен* ***константно-тегловен код (CWC)*** *с дължина $n$, тегло $w$ и минимално разстояние $d$ е съвкупност от двоични вектори с дължина $n$ (кодови думи), които имат точно $w$ ненулеви позиции и Хеминговото разстояние между всеки две от тях е поне $d$.*

CWC е цикличен, ако при цикличното завъртане на всяка кодова дума се получава друга кодова дума. Цикличен CWC съответства на $(n,w,w-d/2)$ CPCW код (фиг. 6.1).

Връзката с частичните дизайни показва, че оптичните ортогонални кодове и CPCW кодовете могат да се разглеждат също и като циклични комбинаторни структури. Автоморфизмите на цикличната група от ред $v$ съпоставят на всяка циркулантна матрица от ред $v$ циркулантна матрица от ред $v$. Ето защо за циклични комбинаторни структури и за разглежданите кодове с циклична структура се дефинира *мултипликативна еквивалентност* [70], [91].

**Дефиниция 6.1.10.** *Два $(v,k,\lambda)$ CPCW кода $\mathcal{C}$ и $\mathcal{C}'$ са* ***мултипликативно еквивалентни****, ако могат да бъдат получени един от друг чрез прилагане на автоморфизъм на $Z_v$ и замяна на кодова дума с някоя от нейните транслации.*

**Дефиниция 6.1.11.** ***Антиконфликтен код*** *с дължина $v$ за $k$ активни потребители наричаме множество $C \subseteq \{0,1\}^v$ от двоични вектори с Хеминговo тегло $k$, такова, че цикличните завъртания $x'$, $y'$ на две различни думи $x,y \in C$ имат най-много една еднаква координата, т.е. $dist(x',y') \geq 2k-2$, където $dist(x',y')$ е Хеминговото разстояние между $x'$ и $y'$, а под циклично завъртане на думата разбираме $m$-кратно циклично завъртане за произволно $m$.*

Отново използваме следната по-удобна за нас дефиниция.

**Дефиниция 6.1.12.** *Всеки $(v,k)$ антиконфликтен код $C$ можем да разглеждаме като съвкупност от $k$-елементни подмножества на $Z_v$ (кодови думи), такива, че*

$$\Delta(X) \cap \Delta(Y) = \emptyset \qquad X,Y \in C,$$

*където както и при CPCW кодовете с $\Delta(X)$ и $\Delta(Y)$ означаваме множествата от ненулевите разлики между елементите на кодовите думи.*

Всеки $(v,k)$ антиконфликтен код е еквивалентен на $(v,k,k,1)$ OOC.

Фигура 6.1: Съответствия

a) - Оптимален (41,5,1) CPCW код

- Оптимален (41,5,1) OOC

| $\mathcal{C}$ | $\triangle C$ |
|---|---|
| {0, 1, 4, 11, 29} | {1, 3, 4, 7, 10, 11, 12, 13, 16, 18, 23, 25, 28, 29, 30, 31, 34, 37, 38, 40} |
| {0, 2, 8, 17, 22} | {2, 5, 6, 8, 9, 14, 15, 17, 19, 20, 21, 22, 24, 26, 27, 32, 33, 35, 36, 39} |

b) - Съответният цикличен 2-(41,5,1) частичен дизайн

- Съответният цикличен константно-тегловен код с дължина 41, тегло 5 и минимално разстояние 8

| | | |
|---|---|---|
| 0 | 10000000000010000000000000000010000001001 | 10000000000000000001000010000000010000010 |
| 1 | 11000000000001000000000000000001000000100 | 01000000000000000000100001000000001000001 |
| 2 | 01100000000000100000000000000000100000010 | 10100000000000000000100001000000000100000 |
| 3 | 00110000000000010000000000000000010000001 | 01010000000000000000010000100000000010000 |
| 4 | 10011000000000001000000000000000001000000 | 00101000000000000000001000010000000001000 |
| 5 | 01001100000000000100000000000000000100000 | 00010100000000000000000100001000000000100 |
| 6 | 00100110000000000010000000000000000010000 | 00001010000000000000000010000100000000010 |
| 7 | 00010011000000000001000000000000000001000 | 00000101000000000000000001000010000000001 |
| 8 | 00001001100000000000100000000000000000100 | 10000010100000000000000000010000100000000 |
| 9 | 00000100110000000000010000000000000000010 | 01000001010000000000000000001000010000000 |
| 10 | 00000010011000000000001000000000000000001 | 00100000101000000000000000000100001000000 |
| 11 | 10000001001100000000000100000000000000000 | 00010000010100000000000000000010000100000 |
| 12 | 01000000100110000000000010000000000000000 | 00001000001010000000000000000001000010000 |
| 13 | 00100000010011000000000001000000000000000 | 00000100000101000000000000000000100001000 |
| 14 | 00010000001001100000000000010000000000000 | 00000010000010100000000000000000010000100 |
| 15 | 00001000000100110000000000001000000000000 | 00000001000001010000000000000000001000010 |
| 16 | 00000100000010011000000000000100000000000 | 00000000100000101000000000000000000100001 |
| 17 | 00000010000001001100000000000010000000000 | 10000000010000010100000000000000000010000 |
| 18 | 00000001000000100110000000000001000000000 | 01000000001000001010000000000000000001000 |
| 19 | 00000000100000010011000000000000100000000 | 00100000000100000101000000000000000000100 |
| 20 | 00000000010000001001100000000000010000000 | 00010000000010000010100000000000000000010 |
| 21 | 00000000001000000100110000000000001000000 | 00001000000001000001010000000000000000001 |
| 22 | 00000000000100000010011000000000000100000 | 10000100000000100000101000000000000000000 |
| 23 | 00000000000010000001001100000000000010000 | 01000010000000010000010100000000000000000 |
| 24 | 00000000000001000000100110000000000001000 | 00100001000000001000001010000000000000000 |
| 25 | 00000000000000100000010011000000000000100 | 00010000100000000100000101000000000000000 |
| 26 | 00000000000000010000001001100000000000010 | 00001000010000000010000010100000000000000 |
| 27 | 00000000000000001000000100110000000000001 | 00000100001000000001000001010000000000000 |
| 28 | 00000000000000000100000010011000000000001 | 00000010000100000000100000101000000000000 |
| 29 | 10000000000000000010000001001100000000000 | 00000001000010000000010000010100000000000 |
| 30 | 01000000000000000001000000100110000000000 | 00000000100001000000001000001010000000000 |
| 31 | 00100000000000000000100000010011000000000 | 00000000010000100000000100000101000000000 |
| 32 | 00010000000000000000010000001001100000000 | 00000000001000010000000010000010100000000 |
| 33 | 00001000000000000000001000000100110000000 | 00000000000100001000000010000010100000000 |
| 34 | 00000100000000000000000100000010011000000 | 00000000000010000100000001000001010000000 |
| 35 | 00000010000000000000000010000001001100000 | 00000000000001000010000000100000101000000 |
| 36 | 00000001000000000000000001000000100110000 | 00000000000000100001000000010000010100000 |
| 37 | 00000000100000000000000000100000010011000 | 00000000000000010000100000001000001010000 |
| 38 | 00000000010000000000000000010000001001100 | 00000000000000001000010000000010000010100 |
| 39 | 00000000001000000000000000001000000100110 | 00000000000000000100001000000001000001010 |
| 40 | 00000000000100000000000000000100000010011 | 00000000000000000010000100000000100000101 |

В част 6.5 конструираме и ***силни антиконфликтни кодове*** (SCAC), които са антиконфликтни кодове, отговарящи на някои допълнителни условия.

Оттук нататък ще разглеждаме кодовите думи на всички кодове с циклична структура като подмножества на $Z_v$. Както при OOC, върху кодовите думи на CPCW, CAC и SCAC дефинираме **лексикографска наредба** по следния начин: Приемаме, че $c_1 < c_2 < \ldots < c_k$ за всяка кодова дума $C = \{c_1, c_2, \ldots, c_k\}$. Тогава $C' = \{c'_1, c'_2, \ldots, c'_k\}$ е лексикографски по-малка от $C'' = \{c''_1, c''_2, \ldots, c''_k\}$, ако $|\Delta(C')| < |\Delta(C'')|$, или ако $|\Delta(C')| = |\Delta(C'')|$, но $c'_i = c''_i$ за $i < a$, а $c'_a < c''_a$.

Ако заменим кодова дума $C$ на код с циклична структура с някоя нейна транслация $C+t$, то получаваме еквивалентен код. Затова, без загуба на общност, приемаме, че всяка кодова дума е лексикографски по-малка от всички свои транслации. Това означава, че $c_1 = 0$ и когато казваме, че автоморфизмът $\varphi$ изобразява $C_1$ в $C_2$, ние имаме предвид, че $C_2$ е най-малката транслация на $\varphi(C_1)$.

## 6.2 Алгоритъм за класификация на кодове с циклична структура при сравнително малък брой решения

Цикличната структура на всички обекти, разглеждани в тази глава ни позволява да използваме един и същ подход към задачата за тяхната класификация. Всички алгоритми, използвани за класификация на обектите от настоящата глава, могат да бъдат разглеждани като модификации на алгоритъма за класификация на оптични ортогонални кодове, описан в раздел 1.2.6, защото използват същия начин на предварително подреждане и записване в масив на множеството от възможните кодови думи и множествата от разликите им. По-точно, орбитите на тези думи относно групата от автоморфизми на $Z_n$ са подредени в масива една след друга съобразно лексикографската наредба на техните лидери, което осигурява бързината на теста за минималност на частичните решения.

Всичко останало в тези алгоритми отразява специфичните особености на конструираните обекти. На тях е обърнато внимание в съответните следващи раздели, докато тук ще опишем модификация на основния алгоритъм, която е приложима за всички обекти от тази глава. Тя използва прост допълнителен тест за разширимост на частичните решения, който, обаче, се оказва много ефективен в случаите, когато за конкретните параметри не съществува код с дадения брой кодови думи, или съществуват малък брой кодове.

С нарастване на параметъра $k$ масивът с възможните кодови думи става много голям, а търсенето с връщане бавно. Освен това, ако вече сме намерили първите $s-1$ от всичките $s$ кодови думи, последната кодова дума трябва да отговаря на твърде много условия и случаите, когато такава не съществува са чести. От друга страна, обикновено броят на думите, за които трябва да проверим дали са подходящи, е много голям. Затова преди да започнем търсенето на последната дума

правим този тест.

### 6.2.1 Допълнителен тест за разширимост: съществуване на следваща кодова дума

Целта му е по бърз начин да провери подходящо необходимо условие за съществуването на следваща кодова дума.

Нека $U$ е множеството от разликите, които не са покрити от първите $s-1$ кодови думи. Тестът търси цели числа $a_1 < a_2 < \ldots < a_{k-1}$ такива, че $a_i, a_j, a_i - a_j \in U$, където $i > j$ и $i, j = 1, \ldots, k-1$. Ако намери такава съвкупност от числа, тестът е положителен. Ако такива числа няма, тестът е отрицателен и алгоритъмът се връща една стъпка назад, т.е. търси следваща възможност за $s-1$-та кодова дума.

### 6.2.2 Приложимост

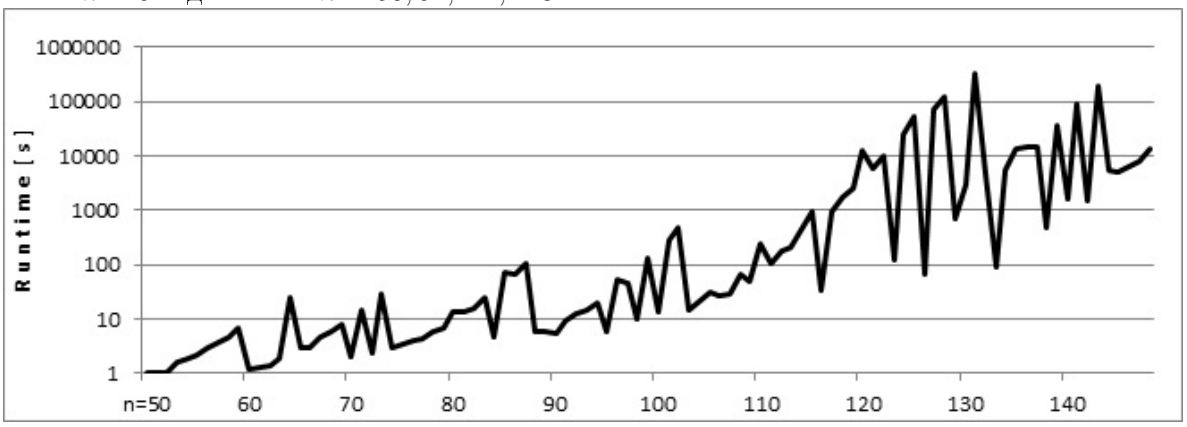
Алгоритъмът е обикновено няколко пъти по-бърз от описания в раздел 1.2.6, ако броят на непокритите разлики е сравнително малък и малко по-бавен от него в останалите случаи. Ако максималният брой кодови думи за код с дадените параметри не е предварително известен, трябва не само да направим класификация на кодовете с възможно най-много думи, но и на тези с една дума повече, защото за тях трябва да покажем, че не съществуват. Именно при проверката за несъществуване е удачно да бъде включен допълнителният тест, но има и други специфични случаи когато се оказва много подходящ. Такова приложение, например, е описано в част 6.3 за циклично - пермутационни константно - тегловни кодове с дължини близки до тези на съвършените кодове и именно то направи възможно доказателството за несъществуване на разностна фамилия с параметри $(127, 7, 1)$.

Иначе поведението на двата алгоритъма е сходно. За случаите, разгледани в настоящата глава, времето за класификация за дадени фиксирани параметри е между няколко секунди и няколко дни на 3.5 GHz PC. То, разбира се, нараства с дължината на кодовете, но силно зависи и от размера на оптималния код и големината на множеството на търсене (масива с възможните кодови думи). Когато броят на кодовите думи е сравнително голям за даден диапазон от кодови дължини, броят на конструираните кодове е обикновено малък и класификацията може да бъде многократно по-бърза от тази за съседни дължини. Така най-интересните оптимални кодове са най-лесни за получаване.

**Пример 6.2.1.** *Да разгледаме силни антиконфликтни кодове с $k = 5$ (Таблица 6.18). Оптималните кодове с дължини $n = 130$ и 132 имат 7 кодови думи и класификациите им отнемат по около час, докато за $n = 131$ броят на думите е 6, има много кодове и класификацията отнема почти 4 дена. За $n = 133$ кодовите*

*думи са* 8, *има само* 6 *кода и търсенето приключва за по-малко от* 2 *минути. Подобна картина се наблюдава и в останалите диапазони от кодови дължини (виж Фигура 6.2), както и при други обекти и параметри.*

Фигура 6.2: Компютърно време за класификацията на силни антиконфликтни кодове с тегло $k = 5$ и дължини $n = 50, 51, \ldots, 148$.




### 6.2.3 Други алгоритми за кодове с циклична структура

Преди изследванията от настоящата глава CPCW кодове са конструирани в [60], където е представен по един максимален $(v, 4, 2)$ CPCW код за всяко $v \leq 44$ (с 3 възможни изключения), като е използван алгоритъм за намиране на максимална клика в граф. В [256] е описан евристичен алгоритъм за генериране на $(v, k, \lambda, 1)$ OOC с голям брой кодови думи, но кодовете, получени с него, рядко са оптимални, а за класификация изобщо не става въпрос. Класификация с алгоритъма, описан в раздел 1.2.6 е правена в [18], [20] и [22].

## 6.3 Оптимални $(v, k, 1)$ двоични циклично - пермутационни константно - тегловни кодове и циклични дизайни

### 6.3.1 Въведение

*Циклично - пермутационните кодове* (cyclically permutable codes, CPC) са въведени от Gilbert [102] през 1963 година. Те са двоични блокови кодове с дължина $v$, такива че всяка кодова дума има $v$ различни циклични завъртания и нито една кодова дума не може да бъде получена като циклично завъртане на някоя друга кодова дума. *Циклично-пермутационните константно-тегловни* кодове са едновременно константно-тегловни и циклично-пермутационни. Тези кодове са изследвани в [32, 194, 196] и са приложени при конструирането на последователности за комуникационни канали без обратна връзка, използвани едновременно от много потребители.

CPCW кодовете имат важни приложения [31], [61], [66] и са обект на много изследвания (например [32], [194], [196]).

Нека $M(v,k,1)$ да е най-големият възможен размер на $(v,k,1)$ цикличнопермутационен константно-тегловен код. В сила е следната горна граница [61]:

$$M(v,k,1) \leq \left\lfloor \frac{(v-1)}{k(k-1)} \right\rfloor .$$

CPCW кодове с $\lambda = 1$, които достигат тази граница, се наричат ***оптимални***. Ако $M(v,k,1)$ е точно $(v-1)/k(k-1)$, то $(v,k,1)$ циклично-пермутационният константно-тегловен код е съвършен и съответства на цикличен 2-$(v,k,1)$ дизайн и на циклична $(v,k,1)$ разностна фамилия.

В [41] е доказано, че оптимален $(v,3,1)$ CPCW код съществува за всички $v$ с изключение на $v = 6t+2$ и $t \equiv 2$ или 3 (mod 4). За $k \geq 4$, проблемът за съществуването на оптимален $(v,k,1)$ CPCW код остава нерешен, въпреки наличието на много резултати. CPCW кодове с $k = 3$ и малки параметри са класифицирани в [22]. В настоящата работа разглеждаме следващите най-малки отворени случаи, т.е. $k = 4, 5, 6$ и 7.

Ще представим накратко известните резултати за тези тегла. Комбинаторни конструкции на оптимални $(v,k,1)$ OOC от $g$-регуларни кодове с по-малки параметри са представени в [95] и [251]. Съществуването на оптимални $(v,4,1)$ CPCW кодове е тема на редица статии, сред които [15], [41], [46], [48], [53], [57] и [244]. В тях са предложени конструкции на безкрайни класове $(v,4,1)$ CPCW кодове за някои стойности на $v$. Известно е, че оптимален $(v,4,1)$ CPCW код съществува [244] за всяко $v \leq 1212, v \neq 25$. Конструкции на оптимални $(v,5,1)$ CPCW кодове за дължини, отговарящи на някои условия са представени в [54], [108], [169], [170], [226], а за $(v,6,1)$ - в [245].

Някои резултати за CPCW кодове следват от резултати за циклични разностни фамилии с параметри $(v,5,1)$ [48], [57], [136], $(v,6,1)$ [56] и $(v,7,1)$ [55]. В [55] е доказано, че съществува $(q,7,1)$ циклична разностна фамилия за всяко $q$, което е степен на просто число и $q \equiv 1$ (mod 42) с изключение на $q = 43$ и, може би, на $q = 127, 211, 316$ и някои $q > 261239791$. Нашите резултати касаят първия неизяснен случай, като показват, че $(127,7,1)$ циклична разностна фамилия не съществува.

Известни са рекурсивни конструкции на CPCW кодове с големи дължини от CPCW кодове с малки параметри и някои допълнителни свойства [59], [95], [251]. Настоящите класификационни резултати могат да бъдат използвани в такива конструкции.

Преди настоящото изследване не бяха известни класификационни резултати за $(v,k,1)$ CPCW кодове за $k = 4, 5, 6$ и 7, но имаше класификации на циклични 2-$(v,k,1)$ дизайни (циклични Щайнерови системи $S(2,k,v)$). Циклични дизайни с $k = 4$ бяха класифицирани преди [T9] за $v \leq 64$ [70], а преди [T18] тези с $k = 5$

и $v \leq 85$ [19], [70], с $k = 6$ и $v \leq 96$ [19], [63], [120] и с $k = 7$ и $v \leq 91$ [19], [55]. Сред тях дизайните с $v \equiv 1 \pmod{k(k-1)}$ са строго циклични и съответстващи на $(v, k, 1)$ CPCW кодове, докато тези с $v \equiv k \pmod{k(k-1)}$ притежават и една къса орбита. Щайнеровите системи $S(2, 4, v)$ представляват особен интерес и са изследвани от много автори, което може да се види от обзора на Reid и Rosa [207], според който класификация на циклични $S(2, 4, v)$ за $v > 64$ не беше известна до настоящата работа. Както вече споменахме, не бяха известни и примери за циклични 2-$(127, 7, 1)$ дизайни.

Ние класифицираме с точност до мултипликативна еквивалентност оптималните CPCW кодове с $k = 4$ и $v \leq 76$, $k = 5$ и $v \leq 89$, $k = 6$ и $v \leq 105$, и $k = 7$ и $v \leq 127$. От резултатите следва несъществуването на $(127, 7, 1)$ циклична разностна фамилия. След това класифицираме с точност до изоморфизъм цикличните 2-$(v, 4, 1)$ дизайни с $v = 73$ (еквивалентни на $(73, 4, 1)$ циклични разностни фамилии) и $v = 76$. Също така повтаряме с нашия софтуер известните класификации на циклични 2-$(v, k, 1)$ дизайни и получаваме същото. Поради съответствието между оптималните $(v, k, 1)$ CPCW, оптималните циклични двоични константно-тегловни кодове с тегло $k$ и минимално разстояние $2(k-1)$ и оптималните $(v, k, 1)$ оптични ортогонални кодове, класификацията на CPCW кодовете важи и за тях.

### 6.3.2 Особености на използваните алгоритми

Отначало класифицираме оптималните $(v, k, 1)$ CPCW кодове и цикличните 2-$(v, k, 1)$ дизайни с точност до мултипликативна еквивалентност с модификации на алгоритмите, описани, съответно, в раздел 1.2.6 и част 6.2 , като и от двата изключваме теста за разширимост на частичните решения, който проверява типа на оставащите кодови думи, защото при разглежданите CPCW кодове $\lambda_a = \lambda_b = 1$ и всички възможни думи са от един и същи тип. По тази причина знаем и точния брой на разликите, които оптималният за дадена дължина код не покрива. Когато той е малък (т.е за дължини, близки до тези на съвършените кодове) използваме алгоритъма от част 6.2.

За конструирането на циклични дизайни с къса орбита използваме модификация на алгоритмите, при която отначало фиксираме елементите на късата орбита, чиито лидер е $\{0, v/k, 2v/k, \ldots, (k-1)v/k\}$.

### 6.3.3 Циклични 2-$(73, 4, 1)$, 2-$(76, 4, 1)$ и 2-$(127, 7, 1)$ дизайни - класификация и изоморфизъм

Тук разглеждаме отворените преди настоящата работа случаи за класификация на циклични дизайни. Съвършените (73,4,1) CPCW кодове са циклични 2-(73,4,1) дизайни, чиито брой с точност до мултипликативна еквивалентност е 1426986, а съвършените $(127, 7, 1)$ CPCW, каквито установяваме, че не съществуват, отговарят на циклични 2-$(127, 7, 1)$ дизайни. С модификация на основния

алгоритъм конструираме цикличните 2-$(76, 4, 1)$ дизайни. Те имат една къса орбита с дължина $v/4$ и базисен блок $\{0, v/4, v/2, 3v/4\}$. Получаваме 1113024 циклични 2-$(76, 4, 1)$ дизайна.

Понеже намираме всички 2-$(73, 4, 1)$, 2-$(76, 4, 1)$ циклични дизайни с точност до мултипликативна еквивалентност, трябва да проверим дали измежду тях няма изоморфни. Тук описваме начина, по който правим това.

Отначало прилагаме тест за изоморфизъм на циркулантите, от които са съставени дизайните.

**Теорема 6.3.3.1.** *Всички циркуланти от ред* $73(76)$*, от които са конструирани мултипликативно нееквивалентните* 2-$(73, 4, 1)$ (2-$(76, 4, 1)$) *дизайни, са изоморфни помежду си, тогава и само тогава, когато са мултипликативно еквивалентни.*

**Доказателство.** Ние проверяваме това с компютър. За повечето стойности на $v$ използваме нашия софтуер за установяване на изоморфизъм на дизайни. За някои стойности на $v$, обаче, тестът за изоморфизъм на циркуланти е трудна задача за софтуер за общия случай на дизайни. Ето защо за някои $v$ конструираме множеството от обобщени множители, дефинирани от Muzychuk [195]. Те водят до набор от пермутации на $Z_v$, наречен *оперативно множество* [195]. Обобщените множители и съответното оперативно множество са дефинирани за всяка стойност на $v$. Muzychuk доказва, че две циркуланти от ред $v$ са изоморфни, ако могат да бъдат изобразени една в друга от някоя пермутация на оперативното множество. За да получим оперативно множество, първо намираме делителите на $Z_v$, такива, че $Z_v = a_1 a_2 \dots a_r$, където $a_i = p_i^{\alpha_i}$ и $p_i \neq p_j$ за $i, j = 1, 2, \dots, r, i \neq j$. Всеки елемент $x \in Z_v$ може да бъде представен с $r$-орката $(x_1, x_2, \dots, x_r)$, където $x_i \in Z_{a_i}$ и $x = \sum_{i=1}^{r} x_i \prod_{l=1}^{i-1} a_l$. Цялото число $x_i$ по-нататък се представя като $(x_{i_0}, x_{i_1}, \dots, x_{i_{\alpha_i - 1}})$, където $x_{i_j} \in Z_{p_i}$ и $x_i = \sum_{j=0}^{\alpha_i - 1} x_{i_j} p_i^j$.. Обобщеният множител е $r$-орка $(\vec{m_1}, \vec{m_2}, \dots, \vec{m_r})$, където $\vec{m_i} = (m_{i_1}, m_{i_2}, \dots m_{i_{\alpha_i}})$, $m_{i_j}$ е взаимопросто с $p_i$ и $m_{i_j} < p_i^j \prod_{l=1}^{i-1} a_l$. Пермутацията $\varphi$ , съответстваща на обобщения множител, се задава като: $\varphi x_i = \sum_{j=0}^{\alpha_i - 1} m_{i_{\alpha_i - j}} x_{i_j} p_i^j$. □

От Теорема 6.3.3.1 не следва, че мултипликативно нееквивалентните циклични дизайни с $v = 37, 40, 49, 52, 61, 64, 73$ и 76 са неизоморфни, тъй като е възможно два дизайна $D$ и $D'$ да бъдат трансформирани един в друг чрез пермутация, която изобразява блокове от една и съща циркуланта (блокова орбита спрямо конструктивния автоморфизъм от ред $v$) на $D$ в блокове от различни циркуланти на $D'$.

**Теорема 6.3.3.2.** *Мултипликативно нееквивалентните циклични* 2-$(73, 4, 1)$ *и* 2-$(76, 4, 1)$ *дизайни са неизоморфни.*

**Доказателство.** Неизоморфността на 2-$(73, 4, 1)$ дизайните следва от резултат на Pálfy, който доказва в [199], че две циклични комбинаторни структури с брой

на точките просто число са изоморфни тогава и само тогава, когато са мултипликативно еквивалентни.

За всеки 2 - $(76, 4, 1)$ дизайн изчисляваме следните инварианти на блоковете, а именно: за всеки блок $B$ намираме вектора $(N_0, n_1, ..., n_{vs-3})$, където $n_i$ $(i = 0, 1, ..., vs-3)$ е броят на двойките блокове $(V, W)$, различни от $B$ и такива, че съществуват точно $i$ други блока, които имат обща точка с всеки от блоковете $B$, $V$, $W$. Блокове от една и съща циркуланта имат едни и същи инварианти. За 1112992 от дизайните, всеки два блока от различни циркуланти имат различни инварианти. Ако два от тези дизайни са изоморфни, циркулантите се изобразяват при този изоморфизъм в циркуланти. От Теорема 6.3.3.1 следва, че изоморфизъм е възможен само сред оставащите 32 дизайна.

Да предположим, че $D_1$ и $D_2$ са два от тези 32 дизайна и има пермутация $\varphi$ на точките, която не е автоморфизъм на $Z_v$ и изобразява $D_1$ в $D_2$, т.е. $\varphi D_1 = D_2$ От начина на конструиране двата дизайна имат група от автоморфизми $G_{76}$ с генератор $(1, 2, ..., 76)$. Нека $\alpha \in G_{76}$. Тогава $\varphi D_1 = \alpha\varphi D_1$ и следователно

$$D_1 = \varphi^{-1}\alpha\varphi D_1,$$

т.е. $\varphi^{-1}\alpha\varphi$ също е автоморфизъм на $D_1$. Затова $D_1$ е инвариантен както спрямо $G_{76}$, така и спрямо $\varphi^{-1}G_{76}\varphi$. Ако $\varphi^{-1}G_{76}\varphi = G_{76}$, то $\varphi$ е автоморфизъм на $Z_v$. В противен случай $D_1$ притежава автоморфизъм от ред 76, който не е от $G_{76}$. Ние намираме групите от автоморфизми на тези 32 дизайна. Всички автоморфизми от ред 76 се генерират от конструктивния автоморфизъм $(1, 2, ..., 76)$, т.е. $\varphi G_{76}\varphi^{-1} = G_{76}$. Следователно дизайните са неизоморфни. □

Така получаваме 1426986 неизоморфни циклични 2-(73,4,1) дизайна и 1113024 циклични 2-$(76, 4, 1)$ дизайна. Повтаряме и класификацията на циклични 2-$(v, 4, 1)$ дизайни с една къса орбита за $v = 40, 52$ и 64 и получаваме същия брой като в [70].

### 6.3.4 Класификация на оптимални $(v, k, 1)$ двоични циклично - пермутационни константно - тегловни кодове с $k = 4, 5, 6$ и 7 и малки дължини.

В Таблици 6.1 – 6.4 са представени резултатите от класификацията с точност до мултипликативна еквивалентност за оптимални $(v, k, 1)$ CPCW кодове . Буквата $p$ след стойността на $v$ означава, че кодовете са съвършени (perfect). Броят на кодовите думи е означен с $s$, а мощността на оптималния код с $s_b$. Само за $k = 4$ за всички разглеждани дължини има оптимални кодове, т.е. $s = s_b$. В останалите случаи за дължини, за които няма оптимален код, класифицираме и кодовете с максимален брой думи (максимални кодове). Таблиците започват с най-малката дължина, за която $s_b = 2$. Файловете с всички конструирани CPCW кодове могат да бъдат изтеглени от http://www.moi.math.bas.bg/∼ tsonka.

Използването на конструираните кодове за получаване на кодове с по-големи параметри чрез известни рекурсивни конструкции не винаги е проста процедура. Например в [59, Corollary 3] е доказано, че ако съществува $(v, k, 1)$ CPCW код с $s$ кодови думи, то съществува $(vm, k, 1)$ CPCW код с $sm$ кодови думи ако простите делители на $m$ са не по-малки от $k$. Ако вземем някой от оптималните $(81, 5, 1)$ CPCW кодове (с 4 кодови думи) и изберем $m = 7$, получаваме оптимален $(567, 5, 1)$ CPCW код с 28 кодови думи. В този случай съществуването на такъв код следва директно, но ако трябва да построим самия код, то първо трябва да намерим $k \times m$ 2-проста матрица над $Z_m$ чрез специална конструкция за $r$-прости матрици и тогава от елементите на тази матрица и кодовите думи на кода с дължина 81 по определен начин да получим кодовите думи на кода с дължина 567.

Доста рекурсивни конструкции използват $g$-регулярни CPCW кодове [95], [251]. Затова за $k = 5, 6$ и 7 в Таблица 6.5 представяме отделна класификация на тези от конструираните кодове, които са $g$–регулярни. Файлове с тях могат да бъдат изтеглени от същия сайт.

Резултатите в тази част са съвместни с Цонка Байчева и са публикувани в [T9] и [T18].

**Таблица 6.1: Оптимални (v,4,1) CPCW кодове**

| v | s | кодове | v | s | кодове | v | s | кодове |
|---|---|---|---|---|---|---|---|---|
| 26 | 2 | 1 | 43 | 3 | 1772 | 60 | 4 | 7585950 |
| 27 | 2 | 4 | 44 | 3 | 3208 | 61p | 5 | 18132 |
| 28 | 2 | 4 | 45 | 3 | 12428 | 62 | 5 | 20736 |
| 29 | 2 | 11 | 46 | 3 | 9999 | 63 | 5 | 529996 |
| 30 | 2 | 41 | 47 | 3 | 20692 | 64 | 5 | 409632 |
| 31 | 2 | 42 | 48 | 3 | 51510 | 65 | 5 | 3774498 |
| 32 | 2 | 64 | 49p | 4 | 224 | 66 | 5 | 6512840 |
| 33 | 2 | 196 | 50 | 4 | 336 | 67 | 5 | 18814608 |
| 34 | 2 | 181 | 51 | 4 | 5530 | 68 | 5 | 27675160 |
| 35 | 2 | 378 | 52 | 4 | 6382 | 69 | 5 | 153524880 |
| 36 | 2 | 731 | 53 | 4 | 28672 | 70 | 5 | 204850952 |
| 37p | 3 | 2 | 54 | 4 | 56064 | 71 | 5 | 425759570 |
| 38 | 3 | 12 | 55 | 4 | 213662 | 72 | 5 | 979134632 |
| 39 | 3 | 96 | 56 | 4 | 263102 | 73p | 6 | 1426986 |
| 40 | 3 | 86 | 57 | 4 | 1105056 | 74 | 6 | 2140556 |
| 41 | 3 | 338 | 58 | 4 | 1011104 | 75 | 6 | 59992260 |
| 42 | 3 | 998 | 59 | 4 | 2575944 | 76 | 6 | 42145856 |

**Таблица 6.2: Максимални (v,5,1) CPCW кодове**

| v | s | $s_b$ | кодове | v | s | $s_b$ | кодове | v | s | $s_b$ | кодове |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 41p | 2 | 2 | 1 | 57 | 2 | 2 | 14765 | 73 | 3 | 3 | 354208 |
| 42 | 2 | 2 | 0 | 58 | 2 | 2 | 14671 | 74 | 3 | 3 | 531778 |
| 42 | 1 | 2 | 289 | 59 | 2 | 2 | 19474 | 75 | 3 | 3 | 2296603 |
| 43 | 2 | 2 | 2 | 60 | 2 | 2 | 61566 | 76 | 3 | 3 | 1987306 |
| 44 | 2 | 2 | 2 | 61p | 3 | 3 | 10 | 77 | 3 | 3 | 4609494 |
| 45 | 2 | 2 | 25 | 62 | 3 | 3 | 0 | 78 | 3 | 3 | 9070318 |
| 46 | 2 | 2 | 16 | 62 | 2 | 3 | 62088 | 79 | 3 | 3 | 10004862 |
| 47 | 2 | 2 | 36 | 63 | 3 | 3 | 128 | 80 | 3 | 3 | 19686568 |
| 48 | 2 | 2 | 133 | 64 | 3 | 3 | 96 | 81p | 4 | 4 | 528 |
| 49 | 2 | 2 | 172 | 65 | 3 | 3 | 640 | 82 | 4 | 4 | 1108 |
| 50 | 2 | 2 | 328 | 66 | 3 | 3 | 1308 | 83 | 4 | 4 | 9472 |
| 51 | 2 | 2 | 880 | 67 | 3 | 3 | 3936 | 84 | 4 | 4 | 23448 |
| 52 | 2 | 2 | 1022 | 68 | 3 | 3 | 6424 | 85 | 4 | 4 | 164390 |
| 53 | 2 | 2 | 1590 | 69 | 3 | 3 | 34264 | 86 | 4 | 4 | 168400 |
| 54 | 2 | 2 | 3838 | 70 | 3 | 3 | 42312 | 87 | 4 | 4 | 1666288 |
| 55 | 2 | 2 | 5381 | 71 | 3 | 3 | 93112 | 88 | 4 | 4 | 1654648 |
| 56 | 2 | 2 | 7733 | 72 | 3 | 3 | 232842 | 89 | 4 | 4 | 7108250 |

**Таблица 6.3: Максимални (v, 6, 1) CPCW кодове**

| v | s | $s_b$ | кодове | v | s | $s_b$ | кодове | v | s | $s_b$ | кодове |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 61p | 2 | 2 | 0 | 74 | 2 | 2 | 1065 | 91p | 3 | 3 | 4 |
| 61 | 1 | 2 | 1377 | 75 | 2 | 2 | 4385 | 92 | 3 | 3 | 0 |
| 62 | 2 | 2 | 0 | 76 | 2 | 2 | 4152 | 92 | 2 | 3 | 3834550 |
| 62 | 1 | 2 | 2185 | 77 | 2 | 2 | 9046 | 93 | 3 | 3 | 16 |
| 63 | 2 | 2 | 2 | 78 | 2 | 2 | 18548 | 94 | 3 | 3 | 20 |
| 64 | 2 | 2 | 0 | 79 | 2 | 2 | 20646 | 95 | 3 | 3 | 152 |
| 64 | 1 | 2 | 2828 | 80 | 2 | 2 | 39532 | 96 | 3 | 3 | 300 |
| 65 | 2 | 2 | 0 | 81 | 2 | 2 | 80680 | 97 | 3 | 3 | 1036 |
| 65 | 1 | 2 | 2929 | 82 | 2 | 2 | 80806 | 98 | 3 | 3 | 1878 |
| 66 | 2 | 2 | 4 | 83 | 2 | 2 | 124738 | 99 | 3 | 3 | 12224 |
| 67 | 2 | 2 | 8 | 84 | 2 | 2 | 355448 | 100 | 3 | 3 | 12808 |
| 68 | 2 | 2 | 20 | 85 | 2 | 2 | 358766 | 101 | 3 | 3 | 43008 |
| 69 | 2 | 2 | 51 | 86 | 2 | 2 | 431172 | 102 | 3 | 3 | 110654 |
| 70 | 2 | 2 | 90 | 87 | 2 | 2 | 930464 | 103 | 3 | 3 | 215110 |
| 71 | 2 | 2 | 162 | 88 | 2 | 2 | 1028267 | 104 | 3 | 3 | 347220 |
| 72 | 2 | 2 | 490 | 89 | 2 | 2 | 1187303 | 105 | 3 | 3 | 2136655 |
| 73 | 2 | 2 | 684 | 90 | 2 | 2 | 3446742 | | | | |

**Таблица 6.4: Максимални (v, 7, 1) CPCW кодове**

| v | s | $s_b$ | кодове |
|---|---|---|---|
| 85p | 2 | 2 | 0 |
| 85 | 1 | 2 | 27159 |
| 86 | 2 | 2 | 0 |
| 86 | 1 | 2 | 32627 |
| 87 | 2 | 2 | 0 |
| 87 | 1 | 2 | 41221 |
| 88 | 2 | 2 | 0 |
| 88 | 1 | 2 | 46520 |
| 89 | 2 | 2 | 0 |
| 89 | 1 | 2 | 33110 |
| 90 | 2 | 2 | 0 |
| 90 | 1 | 2 | 97826 |
| 91 | 2 | 2 | 4 |
| 92 | 2 | 2 | 0 |
| 92 | 1 | 2 | 70056 |
| 93 | 2 | 2 | 0 |
| 93 | 1 | 2 | 79117 |
| 94 | 2 | 2 | 2 |
| 95 | 2 | 2 | 10 |
| 96 | 2 | 2 | 21 |
| 97 | 2 | 2 | 58 |
| 98 | 2 | 2 | 78 |
| 99 | 2 | 2 | 320 |
| 100 | 2 | 2 | 440 |
| 101 | 2 | 2 | 882 |
| 102 | 2 | 2 | 1930 |
| 103 | 2 | 2 | 2982 |
| 104 | 2 | 2 | 5276 |
| 105 | 2 | 2 | 20219 |
| 106 | 2 | 2 | 13954 |
| 107 | 2 | 2 | 29324 |
| 108 | 2 | 2 | 66854 |
| 109 | 2 | 2 | 82224 |
| 110 | 2 | 2 | 160614 |
| 111 | 2 | 2 | 338142 |
| 112 | 2 | 2 | 397005 |
| 113 | 2 | 2 | 542309 |
| 114 | 2 | 2 | 1268036 |
| 115 | 2 | 2 | 1645949 |
| 116 | 2 | 2 | 2071728 |
| 117 | 2 | 2 | 4804470 |
| 118 | 2 | 2 | 4369688 |
| 119 | 2 | 2 | 7757641 |
| 120 | 2 | 2 | 18598700 |
| 121 | 2 | 2 | 14406358 |
| 122 | 2 | 2 | 19993519 |
| 123 | 2 | 2 | 41435642 |
| 124 | 2 | 2 | 43048806 |
| 125 | 2 | 2 | 64040546 |
| 126 | 2 | 2 | 139242441 |
| 127p | 3 | 3 | 0 |
| 127 | 2 | 3 | 98528146 |

**Таблица 6.5: Неизоморфни $g$-регулярни максимални (v,k,1) CPCW кодове**

| v | k | s | g | codes |
|---|---|---|---|---|
| 41p | 5 | 2 | 1 | 1 |
| 61p | 5 | 3 | 1 | 10 |
| 63 | 5 | 3 | 3 | 10 |
| 64 | 5 | 3 | 4 | 4 |
| 65 | 5 | 3 | 5 | 2 |
| 81p | 5 | 4 | 1 | 528 |
| 82 | 5 | 4 | 2 | 1108 |
| 84 | 5 | 4 | 4 | 1072 |
| 85 | 5 | 4 | 5 | 170 |
| 88 | 5 | 4 | 8 | 168 |
| 91p | 6 | 3 | 1 | 4 |
| 91 | 7 | 2 | 7 | 2 |

## 6.4 Антиконфликтни кодове

### 6.4.1 Въведение

В комуникационните системи с множествен достъп без обратна връзка се налага използването на кодове, които да гарантират, че всеки измежду $s$ потенциални участници в комуникацията може да изпрати успешно поне един пакет с данни в рамките на определен времеви период, като се има предвид, че най-много $k$ участници са активни едновременно. Тези кодове се наричат антиконфликтни кодове (CAC). Времевият период е разделен на равни интервали, наречени слотове. На всеки участник в комуникацията се дава по една кодова дума, от чието непрекъснато повтаряне се получава безкрайна двоична протоколна последователност и участникът може да обменя данни през текущия слот, ако е в активно състояние и поредното число в последователността е 1. Трансмисията е успешна, ако няма друг участник, който да обменя данни през същия слот.

**Пример 6.4.1.** *Нека всеки времеви период се състои от* $n = 17$ *слота, всички участници са* $M = 4$*, от които* $k = 3$ *могат да бъдат активни едновременно и се използва CAC с дължина* $n = 17$ *и кодови думи {1,2,3}, {0,3,6}, {2,6,10}, {0,5,10}. На участниците са дадени протоколни последователности* $01110000000000000011100\ldots$, $10010010000000000010010001000\ldots$, *и т.н. Нека първите* 3*-ма участници са активни през някои* 17 *последователни слота. Тогава*

| *слот* | *0* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* | *11* | *12* | *13* | *14* | *15* | *16* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *участник 1* | | ═══ | ═══ | ═══ | | | | | | | | | | | | | |
| *участник 2* | ═══ | | | ═══ | | | ═══ | | | | | | | | | | |
| *участник 3* | | | ═══ | | | | ═══ | | | | ═══ | | | | | | |

*Първият участник комуникира успешно през слот* 1*, вторият през слот* 0*, а третият през слот* 10*. Ако започнат от друго място в протоколните последователности, вместо самите кодови думи на картината ще се появят техни циклични завъртания и отново всеки ще комуникира успешно в поне един слот.*

Броят на кодовите думи в един такъв код определя броя на потенциалните участници в комуникацията през канала. По тази причина, особен интерес представляват кодовете с максимален обем (оптимални кодове) при фиксирани параметри. В тази част класифицираме с точност до мултипликативна еквивалентност оптималните антиконфликтни кодове за $k = 3, 4, 5, 6$ и 7 активни потребители и малки дължини.

Припомняме, че всеки $(n, k)$ антиконфликтен код $C$ можем да разглеждаме като $(n, k, k, 1)$ OOC и като съвкупност от $k$-елементни подмножества на $Z_n$ (кодови думи), такива, че

$$\Delta(X) \cap \Delta(Y) = \emptyset \qquad X, Y \in C.$$

**Дефиниция 6.4.1.** *Казваме, че един $(n,k)$ CAC със $s$ кодови думи е* ***плътен****, ако $|\bigcup_{i=1}^{s}\Delta X_i| = n-1$, т.е. всички ненулеви разлики са покрити.*

Всеки плътен $(n,k)$ CAC съответства на съвършен $(n,k,k,1)$ OOC.

**Пример 6.4.2.** *За всяка от четирите думи на плътен $(15,3)$ CAC е дадено множеството от разликите и типа ѝ (броя на елементите в множеството от разликите).*

$X_1 = \{0,5,10\}$ $\Delta(X_1) = \{5,10\}$ $T(X_1) = 2$
$X_2 = \{0,1,2\}$ $\Delta(X_2) = \{1,2,13,14\}$ $T(X_1) = 4$
$X_3 = \{0,7,11\}$ $\Delta(X_3) = \{4,7,8,11\}$ $T(X_1) = 4$
$X_4 = \{0,6,12\}$ $\Delta(X_4) = \{3,6,9,12\}$ $T(X_1) = 4$

**Дефиниция 6.4.2.** *Ако всяка кодова дума или някоя от транслациите ѝ има вида $\{0,i,\ldots,(k-1)i\}$, то този CAC наричаме еквидистантен.*

**Дефиниция 6.4.3.** *Нека $M(n,k)$ е максималната мощност на CAC с дължина $n$ за $k$ едновременно активни потребители. Един антиконфликтен код е оптимален ако $|C| = M(n,k)$.*

**Дефиниция 6.4.4.** *Два $(n,k)$ CAC $C$ и $C'$ са мултипликативно еквивалентни, ако могат да бъдат получени един от друг чрез прилагане на автоморфизъм на $Z_v$ и замяна на кодова дума с някоя от нейните транслации.*

Съществуват множество връзки между CAC и OOC.

**Твърдение 6.4.1.** *Всеки $(n,k)$ CAC съответства на $(n,k,k,1)$ OOC.*

**Доказателство.** Понеже са $k$-елементни подмножества на $Z_n$, кодовата дума и нейна транслация могат да имат най-много $k$ еднакви елемента. Затова OOC с авто-корелация $k$ е всъщност OOC без никакви изисквания към автокорелацията. Ако крос-корелацията е 1, получаваме $(n,k)$ CAC. □

Един $(n,k)$ CAC може да има кодови думи, които съвпадат с някои от транслациите си. Ще ги наричаме *CAC-специфични кодови думи.*

**Пример 6.4.3.** *Възможни CAC-специфични кодови думи на $(12,4)$ CAC.*

$X_1 = \{0,1,6,7\}$, $X_2 = \{0,2,6,8\}$, $X_3 = \{0,3,6,9\}$, $X_4 = \{0,4,6,10\}$, $X_5 = \{0,5,6,11\}$

*всяка от тях съвпада с някои от транслациите си:*

$X_1 = \{0,1,6,7\} = \{0+t,1+t,6+t,7+t\}$ *за* $t = 6$,
$X_2 = \{0,2,6,8\} = \{0+t,2+t,6+t,8+t\}$ *за* $t = 6$,
$X_3 = \{0,3,6,9\} = \{0+t,3+t,6+t,9+t\}$ *за* $t = 3,6,9$,
$X_4 = \{0,4,6,10\} = \{0+t,4+t,6+t,10+t\}$ *за* $t = 6$,
$X_5 = \{0,5,6,11\} = \{0+t,5+t,6+t,11+t\}$ *за* $t = 6$.

**Пример 6.4.4.** *Възможни CAC-специфични думи на $(n,k)$ CAC с $gcd(n,k)>1$.*

*Нека $n=t\cdot m$ и $k=b\cdot m$.*

*Ако $b=1$, то $\{0,t,\dots(m-1)t\}$ е CAC-специфична кодова дума и ако $k$ е просто число, тя е единствената възможна. Ако $b=2$, то за всяко $c<t$ думата $\{0,c,t,c+t,\dots(m-1)t,c+(m-1)t\}$ е CAC-специфична кодова дума. Ако $b>2$, то $\{0,c_1,\dots c_{b-1},t,c_1+t,\dots c_{b-1}+t,\dots(m-1)t,c_1+(m-1)t\dots c_{b-1}+(m-1)t\}$ е CAC-специфична кодова дума за $0<c_1<\ldots<c_b<t$.*

Съществуват и връзки между $(n,k)$ CAC и $(n,k,k-1,1)$ OOC. В общия случай, липсата на автокорелационно условие води или до повече кодови думи, или до повече кодове. Има, обаче параметри, за които всички $(n,k)$ CAC са $(n,k,k-1,1)$ OOC.

**Твърдение 6.4.2.** *Всеки $(n,k)$ CAC съответства на $(n,k,k-1,1)$ OOC ако $gcd(n,k)=1$.*

**Доказателство.** Да разгледаме една произволна CAC-специфична кодова дума $X=\{x_1,x_2,\dots x_k\}=\{x_1+t,x_2+t,\dots x_k+t\}$ за $t<n$. Тогава $t|n$. Нека $n=m\cdot t$. Дефинираме орбити на $x_i$, такива, че $x_i$ и $x_i+r\cdot t$ са в една и съща орбита за всяко цяло $r<n$. Дължината на всяка орбита е $m$. Нека $b$ е броят на тези орбити. Тогава $k=b\cdot m$ и затова $gcd(n,k)=m$. Следователно CAC-специфична кодова дума не съществува ако $gcd(n,k)=1$. Затова за такива кодове е в сила крос - корелация $k-1$. □

**Следствие 6.4.3.** *Всеки оптимален CAC с дължина $n$ и тегло 3 може да бъде разглеждан като максимален $(n,3,2,1)$ OOC ако $3\nmid n$, а всеки оптимален CAC с дължина $n$ и тегло 4 може да бъде разглеждан като максимален $(n,4,3,1)$ OOC ако $2\nmid n$.*

CAC са интензивно изследвани. Случаят $k=3$ е напълно решен: Levenstein и Tonchev [167] и Levenshtein [166] представят конструкция на оптимални CAC на дължина n за всяко $n\equiv 2\ (mod\ 4)$. Jimbo et al. [125] подобряват границата на Levenstein и дават конструкция на оптимални CAC в случаите, когато $n\equiv 0\ (mod\ 4)$. По-късно Mishima, Fu и Uruno представят директни конструкции на оптимални CAC с дължина $n\equiv 0\ (mod\ 16)$ [189], а Fu, Lin и Mishima [92] - с дължината $n\equiv 4\ (mod\ 8)$ и с това затварят случая на четни дължини. За CAC с нечетна дължина Levenstein и Tonchev [167] представят някои оптимални CAC за конкретни прости дължини, а Levenstein [166] разширява резултата за още нечетни дължини. Wu и Fu дават необходими и достатъчни условия за съществуването на оптимални плътни еквидистантни CAC с дължина $n=2^k\pm 1$ и тегло 3 [250], а Momihara [192] - за плътни еквидистантни CAC с прости нечетни дължини. По-късно Momihara, Müler, Saton и Jimbo [193] предлагат директни и рекурсивни конструкции за оптимални еквидистантни CAC. Накрая, през 2014 г. Fu, Lo и Shum [93] конструират оптимални CAC за нечетни $n$ и $k=3$.

Конструкции на оптимални CAC с тегла 4 and 5 са дадени в [193]. Shum and Wong [214] и Shum, Wong и Chen [215] извеждат горни граници за размера на CAC, които са приложими за всякакви дължини и тегла. Примери за CAC с тегла 3, 4 и 5 и малки дължини има в [234].

В настоящата работа определяме много неизвестни преди стойности на максималния брой кодови думи $M(n,k)$ на $(n,k)$ CAC с $k > 3$ и малки дължини. За повечето разглеждани дължини представяме и класификационни резултати.

### 6.4.2 Особености на антиконфликтните кодове, които са отразени в алгоритмите

#### Общи особености

Поради връзката на CAC с $(n,k,k,1)$ OOC, за конструиране на всички кодове с $M(n,k)$ кодови думи и за установяване на несъществуването на $(n,k)$ CAC с $M(n,k)+1$ кодови думи използваме модификации на алгоритмите, описани в раздел 1.2.6 и част 6.2. Тук ще опишем различното в тези модификации.

Базовият алгоритъмът избира поредната кодова дума сред всички *възможни кодови думи*, които генерираме и подреждаме преди да започне търсенето. Понеже CAC са OOC с най-голямата възможна автокорелация, множеството на търсене е много голямо. Затова търсим възможности то да бъде намалено с предварителни разглеждания и по този начин конструирането на кодовете да стане значително по-бързо.

Нека $T_{min}$ е най-малкият тип, за който съществува възможна кодова дума. Ние конструираме код с фиксиран брой $s$ на кодовите думи. Този код не може да има думи от тип по-голям от $T_{max} = n - 1 - (s-1)T_{min}$ и ние ги отстраняваме от множеството на търсене. Затова в някои случаи възможните кодови думи за код с размер $s+1$ могат да са много по-малко от тези за размер $s$ и, съответно, търсенето може да е много по-бързо. Това е една от причините в някои случаи да определяме $M(n,k)$, но да не представяме класификация на оптималните кодове.

При по-големи $k$ класификационната задача става изчислително по-тежка. Затова за $k = 6$ и $k = 7$ отсяваме част от кодовете, които са различни като комбинаторни конфигурации, но еднакви от гледна точка на приложението им в каналите с множествен достъп. Това правим като следвайки [250] считаме, че:

**Дефиниция 6.4.5.** *Две кодови думи са* ***еквивалентни*** *ако* $\Delta(X) = \Delta(Y)$.

С дефиниция 6.4.5 намаляваме още повече броя на възможните кодови думи, което ускорява търсенето. Множеството на търсене се състои от последователни орбити от думи, като думите от една и съща орбита отиват една в друга при прилагането на автоморфизмите на $Z_n$ (раздел 1.2.6). При построяване на CAC с $k = 6$ и $k = 7$ ние отстраняваме една орбита от множеството от възможните кодови думи, ако множеството от разликите на нейния лидер $X$ съвпада с множеството

от разликите на лидера $Y$ ($\Delta(X) = \Delta(Y)$).

**Особености при плътните CAC**

За някои параметри, за които не сме успели да класифицираме всички кодове, правим класификация само на плътните CAC със следната модификация на конструктивния алгоритъм:

Според дефиниция 6.4.5 отстраняваме една орбита от множеството от възможните кодови думи, ако множеството от разликите на нейния лидер $X$ съвпада с множеството от разликите на произволна дума $Y$ от друга орбита ($\Delta(X) = \Delta(Y)$).

***Допълнителен тест за разширимост на частичните решения***. Нека сме намерили $r$ кодови думи и $T$ е типът на $r$-тата избрана дума, а $d$ броят на покритите разлики. Знаейки типовете на оставащите възможни кодови думи, се опитваме да намерим сред тях типове $T_1 \dots T_{(s-r)}$, за които е в сила

$$d + T_1 + \dots T_{(s-r)} = n - 1.$$

Ако няма такива, тестът връща отрицателен резултат и разглеждаме следващата възможност за $(r-1)$-вата кодова дума на плътния CAC.

### 6.4.3 Класификация - обща и за плътни кодове

За всяка стойност на параметрите изчисляваме най-добрата горна граница за размера на кода според следните три граници, изведени в [214]:

*Граница 1.* ([214], Теорема 1) За всяко $k \geq 2$, е в сила

$$\limsup_{n\to\infty} \frac{M(n,k)}{n} = \frac{1}{2k-2}.$$

*Граница 2.* ([214], Теорема 5) Ако дължината на CAC с тегло $k$ удовлетворява едно от условията:
(i) $n$ е просто число,
(ii) всички прости делители на $n$ са по-големи или равни на $2k-1$.

Тогава

$$M(n,k) \leq \left\lfloor \frac{n-1}{2k-2} \right\rfloor.$$

*Граница 3.* ([214], Теорема 7) Нека $p$ и $k-1$ са различни нечетни прости числа и $p > 2k-2$. Тогава броят на кодовите думи на $(k-1)p, k)$ CAC е най-много $(p-1)/2$.

В таблиците са данните за кодове с поне 2 кодови думи. За всяка дължина е дадена стойността на $M(n,k)$ и, ако тя е била известна преди настоящото изследване, цитираме съответната статия, в която е определена. Освен това в колона $bb$ представяме и най-добрата горна граница за размера на кода. В колоната $CAC$ е записан броят на нееквивалентните оптимални кодове.

Кодовете могат да бъдат изтеглени от http://www.moi.math.bas.bg/∼svetlana. Там е дадена и допълнителна информация за вида на кода според типа на кодовите думи, както е показано в следващия пример.

**Пример 6.4.5.** *Има* 12 *неекивалентни* $(34, 4, 4, 1)$ *CAC от* 4 *вида, които са представени като:*

*0)3: 5-1 6-4*
*1)6: 5-1 6-3 8-1*
*2)2: 5-1 6-2 8-2*
*3)1: 6-4 7-1*

*Това означава, че трите кода от вид* 0 *имат* 1 *кодова дума от тип* 5 *и* 4 *кодови думи от тип* 6*, шестте кода от вид* 1 *имат една кодова дума от тип* 5*, . . . и т.н. Двата кода от вид* 2 *са плътни, защото в този случай всички разлики са покрити* $(5 + 6.2 + 8.2 = 33)$.

Плътните CAC притежават допълнителни свойства, които ги правят интересни като структури на инцидентност и поради това могат да се окажат подходящи в конструкции на други кодове или комбинаторни структури. Такива са, например, някои рекурсивни конструкции на CAC от плътни CAC с по-малки дължини [171].

Условия за съществуване и конструкции на плътни $(n, 3)$ CAC за дадени $n$ са представени в [93], [168], [171], [192] и [250], но понякога проверката на условията и прилагането на конструкциите е доста сложно за тези, които се интересуват от кодовете с оглед на някакво приложение.

Ние класифицираме оптималните плътни CAC с тегла и дължини, съответно, $k = 3$ и $n \leq 111$, $k = 4$ и $n \leq 120$, $k = 5$ и $n \leq 118$, $k = 6$ и $n \leq 119$, и $k = 7$ и $n \leq 95$. Използваният алгоритъм позволява да обхванем и плътни кодове с 21 дължини, за които не е известен броят на всички оптимални CAC.

Резултатите са представени в Таблици 6.11 до 6.15, като дължините, за които не съществуват плътни оптимални CAC са пропуснати, а броят на неекивалентните плътни оптимални CAC е даден в колона TCAC.

**Таблица 6.6: оптимални CAC с $k = 3$ и $n \leq 119$**

| n | M(n,3) | Reff | CAC |
|---|---|---|---|
| 9 | 2 | [166] | 1 |
| 10 | 2 | [167] | 1 |
| 11 | 2 | [166] | 1 |
| 12 | 3 | [167] | 1 |
| 13 | 3 | [166] | 1 |
| 14 | 3 | [167] | 1 |
| 15 | 4 | [166] | 2 |
| 16 | 3 | [167] | 6 |
| 17 | 4 | [166] | 1 |
| 18 | 4 | [167] | 6 |
| 19 | 4 | [166] | 3 |
| 20 | 4 | [167] | 15 |
| 21 | 5 | [166] | 4 |
| 22 | 5 | [167] | 1 |
| 23 | 5 | [166] | 2 |
| 24 | 5 | [167] | 57 |
| 25 | 6 | [166] | 2 |
| 26 | 6 | [167] | 1 |
| 27 | 6 | [166] | 14 |
| 28 | 6 | [167] | 14 |
| 29 | 7 | [166] | 1 |
| 30 | 7 | [167] | 5 |
| 31 | 7 | [167] | 2 |
| 32 | 7 | [167] | 4 |
| 33 | 8 | [167] | 3 |
| 34 | 8 | [167] | 1 |
| 35 | 8 | [166] | 4 |
| 36 | 8 | [167] | 184 |
| 37 | 9 | [166] | 1 |
| 38 | 9 | [167] | 1 |
| 39 | 10 | [166] | 2 |
| 40 | 8 | [167] | 3618 |
| 41 | 10 | [166] | 1 |
| 42 | 10 | [167] | 3 |
| 43 | 10 | [167] | 2 |
| 44 | 9 | [167] | 5452 |
| 45 | 11 | [166] | 4 |
| 46 | 11 | [167] | 1 |
| 47 | 11 | [166] | 2 |
| 48 | 10 | [167] | 63904 |
| 49 | 11 | [93] | 70 |
| 50 | 12 | [167] | 1 |
| 51 | 13 | [166] | 4 |
| 52 | 11 | [92] | 7452 |
| 53 | 13 | [166] | 1 |
| 54 | 13 | [167] | 6 |
| 55 | 13 | [166] | 2 |
| 56 | 12 | [125] | 1344 |
| 57 | 14 | [93] | 5 |
| 58 | 14 | [167] | 1 |
| 59 | 14 | [166] | 3 |
| 60 | 13 | [92] | 294474 |
| 61 | 15 | [166] | 1 |
| 62 | 15 | [167] | 1 |
| 63 | 15 | [93] | 164 |
| 64 | 13 | [189] | 6666368 |
| 65 | 16 | [166] | 10 |
| 66 | 16 | [167] | 3 |
| 67 | 16 | [166] | 5 |
| 68 | 15 | [92] | 1600 |
| 69 | 17 | [166] | 8 |
| 70 | 17 | [167] | 1 |
| 71 | 17 | [166] | 4 |
| 72 | 16 | [125] | 21248 |
| 73 | 17 | [93] | 205 |
| 74 | 18 | [167] | 1 |
| 75 | 19 | [166] | 8 |
| 76 | 16 | [92] | 18781060 |
| 77 | 18 | [93] | 192 |
| 78 | 19 | [167] | 3 |
| 79 | 19 | [166] | 6 |
| 80 | 17 | [189] | 10985704 |
| 81 | 19 | [93] | 12240 |
| 82 | 20 | [167] | 1 |
| 83 | 20 | [166] | 5 |
| 84 | 19 | [92] | 39424 |
| 85 | 21 | [166] | 16 |
| 86 | 21 | [167] | 1 |
| 87 | 22 | [166] | 2 |
| 88 | 19 | [125] | 1263616 |
| 89 | 21 | [93] | 305 |
| 90 | 22 | [167] | 8 |
| 91 | 22 | [166] | 8 |
| 92 | 21 | [92] | 7221248 |
| 93 | 23 | [93] | 6 |
| 94 | 23 | [167] | 1 |
| 95 | 23 | [166] | 6 |
| 96 | 21 | [189] | 261242880 |
| 97 | 24 | [166] | 1 |
| 98 | 24 | [167] | 1 |
| 99 | 24 | [93] | 75 |
| 100 | 22 | [92] | 1839104 |
| 101 | 25 | [93] | 1 |
| 102 | 25 | [167] | 3 |
| 103 | 25 | [93] | 6 |
| 104 | 22 | [125] | $\geq 1$ |
| 105 | 26 | [93] | $\geq 1$ |
| 106 | 26 | [167] | $\geq 1$ |
| 107 | 26 | [93] | $\geq 1$ |
| 108 | 24 | [92] | $\geq 1$ |
| 109 | 27 | [93] | $\geq 1$ |
| 110 | 27 | [167] | $\geq 1$ |
| 111 | 28 | [93] | 2 |
| 112 | 24 | [189] | $\geq 1$ |
| 113 | 28 | [93] | $\geq 1$ |
| 114 | 28 | [167] | $\geq 1$ |
| 115 | 28 | [93] | 4 |
| 116 | 25 | [92] | $\geq 1$ |
| 117 | 29 | [93] | $\geq 1$ |
| 118 | 29 | [189] | $\geq 1$ |
| 119 | 29 | [93] | $\geq 1$ |

**Таблица 6.7: Оптимални CAC с $k = 4$ и $n \leq 152$**

| n | M(n,4) | bb | Reff | CAC |
|---|---|---|---|---|
| 16 | 2 | 2 | [234] | 6 |
| 17 | 2 | 2 | [234] | 2 |
| 18 | 2 | 2 | [234] | 11 |
| 19 | 2 | 2 | [234] | 4 |
| 20 | 3 | 3 | [234] | 5 |
| 21 | 3 | 3 | [193], [234] | 1 |
| 22 | 3 | 3 | [234] | 2 |
| 23 | 2 | 2 | [234] | 37 |
| 24 | 3 | 3 | [234] | 36 |
| 25 | 3 | 3 | [234] | 12 |
| 26 | 4 | 4 | [234] | 1 |
| 27 | 3 | 3 | [234] | 20 |
| 28 | 4 | 4 | [193], [234] | 15 |
| 29 | 4 | 4 | [234] | 1 |
| 30 | 4 | 5 | | 78 |
| 31 | 4 | 5 | | 6 |
| 32 | 5 | 6 | | 1 |
| 33 | 4 | 4 | [193] | 32 |
| 34 | 5 | 6 | | 12 |
| 35 | 6 | 6 | [193] | 4 |
| 36 | 5 | 6 | | 135 |
| 37 | 6 | 6 | | 1 |
| 38 | 6 | 7 | | 4 |
| 39 | 5 | 5 | [193] | 60 |
| 40 | 6 | 7 | | 251 |
| 41 | 6 | 6 | | 3 |
| 42 | 6 | 7 | | 53 |
| 43 | 6 | 7 | | 50 |
| 44 | 7 | 8 | | 23 |
| 45 | 6 | 6 | [214] | 1778 |
| 46 | 7 | 8 | | 22 |
| 47 | 7 | 7 | | 2 |
| 48 | 7 | 8 | | 130 |
| 49 | 8 | 8 | | 4 |
| 50 | 7 | 9 | | 4972 |
| 51 | 7 | 8 | | 121 |
| 52 | 8 | 9 | | 189 |
| 53 | 7 | 8 | | 2530 |
| 54 | 8 | 9 | | 3 |
| 55 | 8 | 10 | | 318 |
| 56 | 8 | 10 | | 9110 |
| 57 | 8 | 8 | [193] | 82 |
| 58 | 9 | 10 | | 9 |
| 59 | 9 | 9 | | 3 |
| 60 | 9 | 10 | | 423 |
| 61 | 9 | 10 | | 5 |
| 62 | 9 | 11 | | 1594 |
| 63 | 9 | 11 | | 32 |
| 64 | 10 | 11 | | 11 |
| 65 | 10 | 11 | | 6 |
| 66 | 10 | 11 | | 3 |
| 67 | 10 | 11 | | 12 |
| 68 | 10 | 12 | | 3970 |
| 69 | 11 | 11 | [193] | 1 |
| 70 | 11 | 12 | | 15 |
| 71 | 10 | 11 | | 2386 |
| 72 | 11 | 12 | | 14 |
| 73 | 11 | 12 | | 3 |
| 74 | 11 | 13 | | 971 |
| 75 | 11 | 13 | | 128 |
| 76 | 12 | 13 | | 15 |
| 77 | 12 | 12 | [193] | 16 |
| 78 | 11 | 13 | | 221613 |
| 79 | 12 | 13 | | 3 |
| 80 | 13 | 14 | | 2 |
| 81 | 12 | 14 | | 4 |
| 82 | 12 | 14 | | 18805 |
| 83 | 12 | 13 | | 1849 |
| 84 | 13 | 14 | | 1 |
| 85 | 13 | 15 | | 159 |
| 86 | 13 | 15 | | 608 |
| 87 | 13 | 13 | [193] | 40 |
| 88 | 14 | 15 | | 37 |
| 89 | 13 | 14 | | 1916 |
| 90 | 14 | 15 | | 8 |
| 91 | 14 | 15 | [193] | 28 |
| 92 | 14 | 16 | | 5387 |
| 93 | 15 | 15 | [193] | 2 |
| 94 | 14 | 16 | | 13946 |
| 95 | 14 | 16 | | 25532 |
| 96 | 14 | 16 | | 130749 |
| 97 | 15 | 16 | | 2 |
| 98 | 15 | 17 | | 205 |
| 99 | 14 | 17 | | 1252877 |
| 100 | 16 | 17 | | 45 |
| 101 | 15 | 16 | | 1014 |
| 102 | 15 | 17 | | 92768 |
| 103 | 16 | 17 | | 1 |
| 104 | 17 | 18 | | 12 |
| 105 | 16 | 18 | | 100 |
| 106 | 16 | 18 | | 7100 |
| 107 | 16 | 17 | | 643 |
| 108 | 16 | 18 | | 43485 |
| 109 | 16 | 18 | | 20426 |
| 110 | 17 | 19 | | 1089 |
| 111 | 17 | 18 | | 29 |
| 112 | 17 | 19 | | 50202 |
| 113 | 17 | 18 | | 748 |
| 114 | 17 | 19 | | 44261 |
| 115 | 17 | 20 | | 408423 |
| 116 | 18 | 20 | | 2036 |
| 117 | 17 | 20 | | $\geq 1$ |
| 118 | 18 | 20 | | $\geq 1$ |
| 119 | 19 | 19 | | 8 |
| 120 | 18 | 20 | | $\geq 1$ |
| 121 | 18 | 20 | | $\geq 1$ |
| 122 | 19 | 21 | | 109 |
| 123 | 19 | 20 | | 20 |
| 124 | 20 | 21 | | 1 |
| 125 | 19 | 21 | | 13761 |
| 126 | 19 | 21 | | $\geq 1$ |
| 127 | 19 | 19 | [193] | $\geq 1$ |
| 128 | 20 | 22 | | 756 |
| 129 | 19 | 21 | | $\geq 1$ |
| 130 | 20 | 22 | | $\geq 1$ |
| 131 | 20 | 21 | | $\geq 1$ |
| 132 | 20 | 22 | | $\geq 1$ |
| 133 | 21 | 22 | | 16 |
| 134 | 21 | 23 | | $\geq 1$ |
| 135 | 20 | 23 | | $\geq 1$ |
| 136 | 22 | 23 | | 164 |
| 137 | 21 | 22 | | $\geq 1$ |
| 138 | 21 | 23 | | $\geq 1$ |
| 139 | 23 | 23 | | 1 |
| 140 | 22 | 24 | | $\geq 1$ |
| 141 | 23 | 23 | | 1 |
| 142 | 22 | 24 | | $\geq 1$ |
| 143 | 22 | 23 | | $\geq 1$ |
| 144 | 22 | 24 | | $\geq 1$ |
| 145 | 22 | 25 | | $\geq 1$ |
| 146 | 23 | 25 | | $\geq 1$ |
| 147 | 24 | 25 | | 2 |
| 148 | 23 | 25 | | $\geq 1$ |
| 149 | 23 | 24 | | $\geq 1$ |
| 150 | 23 | 25 | | $\geq 1$ |
| 151 | 23 | 25 | | $\geq 1$ |
| 152 | 25 | 26 | | 12 |

**Таблица 6.8: Оптимални CAC с $k = 5$ и $n \leq 150$**

| n | M(n,5) | bb Reff | CAC | n | M(n,5) | bb Reff | CAC | n | M(n,5) | bb Reff | CAC |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 25 | 2 | 4 | 8 | 67 | 7 | 8 | 31 | 109 | 11 | 13 | $\geq 1$ |
| 26 | 2 | 4 | 1 | 68 | 7 | 9 | 28 | 110 | 12 | 14 | 3278478 |
| 27 | 2 | 4 | 2 | 69 | 8 | 9 | 8 | 111 | 12 | 14 | 70367 |
| 28 | 2 | 4 | 4 | 70 | 8 | 9 | 5644 | 112 | 11 | 14 | $\geq 1$ |
| 29 | 2 | 3 | 11 | 71 | 7 | 8 | 2594 | 113 | 12 | 14 | 846 |
| 30 | 3 | 4 | 4 | 72 | 7 | 9 | 16873 | 114 | 12 | 15 | $\geq 1$ |
| 31 | 3 | 3 | 1 | 73 | 8 | 9 | 288 | 115 | 13 | 15 | 126761 |
| 32 | 2 | 4 | 94 | 74 | 8 | 10 | 42 | 116 | 13 | 15 | 26 |
| 33 | 2 | 5 | 192 | 75 | 8 | 10 | 3600 | 117 | 14 | 15 | 200 |
| 34 | 2 | 5 | 371 | 76 | 8 | 10 | 26 | 118 | 12 | 15 | $\geq 1$ |
| 35 | 4 | 5 | 3 | 77 | 8 | 10 | 1178517 | 119 | 13 | 15 | 210 |
| 36 | 3 | 5 | 31 | 78 | 8 | 10 | 12675 | 120 | 12 | 15 | $\geq 1$ |
| 37 | 4 | 4 | 1 | 79 | 8 | 9 | 795 | 121 | 13 | 15 | 259 |
| 38 | 3 | 5 | 26 | 80 | 8 | 10 | 33414 | 122 | 13 | 16 | 2011 |
| 39 | 3 | 5 | 83 | 81 | 9 | 11 | 2226 | 123 | 14 | 16 | 1 |
| 40 | 4 | 5 | 15 | 82 | 8 | 11 | 56447 | 124 | 14 | 16 | 2 |
| 41 | 3 | 5 | 205 | 83 | 8 | 10 | 65970 | 125 | 14 | 16 | 2232 |
| 42 | 5 | 6 | 1 | 84 | 8 | 11 | 58003106 | 126 | 13 | 16 | $\geq 1$ |
| 43 | 3 | 5 | 714 | 85 | 9 | 11 | 32302 | 127 | 14 | 15 | 12 |
| 44 | 5 | 5 [193] | 1 | 86 | 9 | 11 | 508 | 128 | 13 | 16 | $\geq 1$ |
| 45 | 6 | 6 [215] | 10 | 87 | 10 | 11 | 34 | 129 | 14 | 17 | $\geq 1$ |
| 46 | 4 | 6 | 124 | 88 | 9 | 11 | 518 | 130 | 15 | 17 | 276 |
| 47 | 5 | 5 | 2 | 89 | 9 | 11 | 25634 | 131 | 15 | 16 | 16 |
| 48 | 4 | 6 | 1231 | 90 | 10 | 12 | 24 | 132 | 14 | 17 | $\geq 1$ |
| 49 | 5 | 7 | 14 | 91 | 10 | 12 | 660 | 133 | 14 | 17 | $\geq 1$ |
| 50 | 6 | 7 | 18 | 92 | 11 | 11 [193] | 1 | 134 | 14 | 17 | $\geq 1$ |
| 51 | 5 | 7 | 20 | 93 | 10 | 12 | 6205 | 135 | 15 | 17 | 3171 |
| 52 | 6 | 7 | 1 | 94 | 10 | 12 | 30 | 136 | 14 | 17 | $\geq 1$ |
| 53 | 5 | 6 | 77 | 95 | 11 | 12 | 4498 | 137 | 15 | 17 | 4 |
| 54 | 6 | 7 | 18 | 96 | 9 | 12 | $\geq 1$ | 138 | 14 | 18 | $\geq 1$ |
| 55 | 6 | 7 | 422 | 97 | 12 | 12 | 2 | 139 | 15 | 17 | 662 |
| 56 | 6 | 7 | 7 | 98 | 10 | 13 | $\geq 1$ | 140 | 15 | 18 | $\geq 1$ |
| 57 | 6 | 8 | 14 | 99 | 12 | 13 | 136 | 141 | 15 | 18 | $\geq 1$ |
| 58 | 7 | 7 [193] | 1 | 100 | 11 | 13 | $\geq 1$ | 142 | 15 | 18 | $\geq 1$ |
| 59 | 6 | 7 | 12 | 101 | 10 | 12 | $\geq 1$ | 143 | 15 | 17 | $\geq 1$ |
| 60 | 6 | 8 | 4706 | 102 | 10 | 13 | $\geq 1$ | 144 | 14 | 18 | $\geq 1$ |
| 61 | 6 | 7 | 378 | 103 | 10 | 12 | $\geq 1$ | 145 | 16 | 19 | $\geq 1$ |
| 62 | 6 | 8 | 1255 | 104 | 10 | 13 | $\geq 1$ | 146 | 15 | 19 | $\geq 1$ |
| 63 | 8 | 8 [215] | 20 | 105 | 11 | 14 | $\geq 1$ | 147 | 16 | 19 | $\geq 1$ |
| 64 | 6 | 8 | 1791 | 106 | 13 | 13 [193] | 1 | 148 | 16 | 19 | $\geq 1$ |
| 65 | 7 | 9 | 651 | 107 | 11 | 13 | $\geq 1$ | 149 | 16 | 18 | $\geq 1$ |
| 66 | 7 | 9 | 139 | 108 | 11 | 14 | $\geq 1$ | 150 | 16 | 19 | $\geq 1$ |

**Таблица 6.9: Оптимални CAC с $k = 6$ и $n \leq 122$**

| n | M(n,6) | bb | Reff | CACs | n | M(n,6) | bb | Reff | CACs | n | M(n,6) | bb | Reff | CACs |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 36 | 2 | 4 | | 175 | 65 | 4 | 6 | | 380 | 94 | 8 | 10 | | 2 |
| 37 | 2 | 3 | | 2 | 66 | 5 | 7 | | 209 | 95 | 9 | 9 | [214] | 1 |
| 38 | 2 | 4 | | 28 | 67 | 4 | 6 | | 913 | 96 | 8 | 10 | | 252 |
| 39 | 2 | 4 | | 99 | 68 | 5 | 7 | | 3 | 97 | 8 | 9 | | 5 |
| 40 | 2 | 5 | | 124 | 69 | 5 | 7 | | 41 | 98 | 8 | 10 | | 65882 |
| 41 | 2 | 4 | | 45 | 70 | 5 | 8 | | 541 | 99 | 8 | 10 | | 3425 |
| 42 | 3 | 5 | | 30 | 71 | 5 | 7 | | 40 | 100 | 8 | 11 | | 43 |
| 43 | 3 | 4 | | 1 | 72 | 5 | 8 | | 7621 | 101 | 8 | 10 | | 27 |
| 44 | 2 | 5 | | 1151 | 73 | 5 | 7 | | 73 | 102 | 9 | 11 | | 8 |
| 45 | 2 | 5 | | 1343 | 74 | 6 | 8 | | 24 | 103 | 8 | 10 | | 4959 |
| 46 | 2 | 5 | | 2597 | 75 | 6 | 8 | | 6 | 104 | 8 | 11 | | 14403 |
| 47 | 2 | 4 | | 1259 | 76 | 6 | 8 | | 5 | 105 | 9 | 11 | | 7 |
| 48 | 3 | 5 | | 348 | 77 | 8 | 8 | [215] | 5 | 106 | 9 | 11 | | 14 |
| 49 | 3 | 5 | | 36 | 78 | 6 | 8 | | 914 | 107 | 8 | 10 | | 39621 |
| 50 | 4 | 6 | | 1 | 79 | 6 | 7 | | 16 | 108 | 9 | 11 | | 93 |
| 51 | 3 | 6 | | 184 | 80 | 6 | 9 | | 23 | 109 | 9 | 10 | | 2 |
| 52 | 3 | 6 | | 357 | 81 | 7 | 9 | | 1 | 110 | 9 | 12 | | 36 |
| 53 | 3 | 5 | | 60 | 82 | 6 | 9 | | 832 | 111 | 9 | 12 | | 18527 |
| 54 | 4 | 6 | | 44 | 83 | 6 | 8 | | 51 | 112 | 9 | 12 | | 16618 |
| 55 | 3 | 5 | | 433 | 84 | 8 | 9 | | 216 | 113 | 9 | 11 | | 90 |
| 56 | 5 | 6 | | 2 | 85 | 7 | 8 | | 10 | 114 | 10 | 12 | | 23 |
| 57 | 4 | 6 | | 6 | 86 | 7 | 9 | | 5 | 115 | 11 | 11 | | 1 |
| 58 | 4 | 6 | | 20 | 87 | 7 | 9 | | 45 | 116 | 9 | 12 | | 10788 |
| 59 | 3 | 5 | | 3387 | 88 | 7 | 9 | | 412 | 117 | 9 | 12 | | 305801 |
| 60 | 4 | 7 | | 929 | 89 | 6 | 8 | | 11848 | 118 | 10 | 12 | | 86 |
| 61 | 4 | 6 | | 18 | 90 | 7 | 10 | | 2527 | 119 | 10 | 12 | | 1830796 |
| 62 | 4 | 7 | | 565 | 91 | 8 | 10 | | 9766 | 120 | 10 | 13 | | 16 |
| 63 | 5 | 7 | | 3 | 92 | 7 | 10 | | 476 | 121 | 12 | 13 | | 5 |
| 64 | 4 | 7 | | 1624 | 93 | 8 | 10 | | 4 | 122 | 10 | 13 | | 294 |

Нашите резултати не са в противоречие с нито един известен ни предишен резултат, по-точно:

- Във всички случаи, когато стойността на $M(n,k)$ беше известна, с нашите програми получихме същата стойност.
- Всички намерени от нас стойности на $M(n,k)$ са по-малки или равни на тези, които пресмятаме според трите граници от [193], [214], [215].
- Конструираният в [214] $(95,6)$ CAC е еквивалентен на намерения от нас и се оказва, че няма друг CAC с тези параметри.
- Класификацията на оптималните CAC с тегло 3 и дължина $n \leq 98$, за които $3 \nmid n$ съответства на класификацията на оптималните $(n,3,2,1)$ OOC (Следствие 6.4.3) с дължина до 98 от [22].
- Класификацията на оптималните CAC с тегло 4 и дължина $n \leq 99$, за които $2 \nmid n$ съответства на класификацията на оптималните $(n,4,3,1)$ OOC (Следствие 6.4.3) с дължина до 99 от [21].

Резултатите в тази част са получени съвместно с Цонка Байчева и са публикувани в [T14], [T15] и [T16].

**Таблица 6.10: Оптимални CAC с $k = 7$ и $n \leq 96$**

| n | M(n,7) | bb | Reff | CACs |
|---|---|---|---|---|
| 49 | 2 | 5 | | 396 |
| 50 | 2 | 5 | | 1 |
| 51 | 2 | 5 | | 3 |
| 52 | 2 | 5 | | 7 |
| 53 | 2 | 4 | | 22 |
| 54 | 2 | 5 | | 36 |
| 55 | 2 | 5 | | 89 |
| 56 | 3 | 5 | | 34 |
| 57 | 3 | 5 | | 1 |
| 58 | 2 | 5 | | 726 |
| 59 | 2 | 4 | | 778 |
| 60 | 2 | 5 | | 5711 |
| 61 | 2 | 5 | | 2112 |
| 62 | 2 | 6 | | 7479 |
| 63 | 4 | 6 | | 6 |
| 64 | 3 | 6 | | 66 |

| n | M(n,7) | bb | Reff | CACs |
|---|---|---|---|---|
| 65 | 4 | 6 | | 1 |
| 66 | 3 | 6 | | 12 |
| 67 | 3 | 5 | | 57 |
| 68 | 3 | 6 | | 70 |
| 69 | 3 | 6 | | 44 |
| 70 | 4 | 6 | | 13 |
| 71 | 3 | 5 | | 405 |
| 72 | 4 | 6 | | 42 |
| 73 | 4 | 6 | | 1 |
| 74 | 4 | 7 | | 3 |
| 75 | 3 | 7 | | 2717 |
| 76 | 4 | 7 | | 5 |
| 77 | 5 | 7 | | 5 |
| 78 | 4 | 7 | | 22 |
| 79 | 4 | 6 | | 81 |
| 80 | 4 | 7 | | 442 |

| n | M(n,7) | bb | Reff | CACs |
|---|---|---|---|---|
| 81 | 4 | 7 | | 910 |
| 82 | 5 | 7 | | 1 |
| 83 | 4 | 6 | | 287 |
| 84 | 5 | 7 | | 38 |
| 85 | 6 | 8 | | 1 |
| 86 | 4 | 8 | | 2187 |
| 87 | 4 | 8 | | 1932 |
| 88 | 7 | 8 | | 1 |
| 89 | 5 | 7 | | 3 |
| 90 | 5 | 8 | | 269 |
| 91 | 8 | 8 | [215] | 6 |
| 92 | 5 | 8 | | 210 |
| 93 | 5 | 8 | | 98 |
| 94 | 5 | 8 | | 51 |
| 95 | 6 | 8 | | 10 |
| 96 | 5 | 8 | | 2197 |

**Таблица 6.11: Плътни оптимални CAC с $k = 3$ и $n \leq 111$**

| n | M(n,3) | TCAC |
|---|---|---|
| 13 | 3 | 1 |
| 15 | 4 | 1 |
| 16 | 3 | 2 |
| 17 | 4 | 1 |
| 18 | 4 | 2 |
| 19 | 4 | 2 |
| 20 | 4 | 3 |
| 21 | 4 | 5 |
| 23 | 5 | 1 |
| 24 | 5 | 8 |
| 25 | 6 | 1 |
| 27 | 6 | 2 |
| 28 | 6 | 5 |
| 29 | 7 | 1 |
| 30 | 7 | 2 |
| 31 | 7 | 2 |
| 32 | 7 | 2 |
| 33 | 8 | 3 |
| 35 | 8 | 1 |
| 36 | 8 | 30 |
| 37 | 9 | 1 |
| 39 | 10 | 2 |
| 40 | 8 | 195 |
| 41 | 10 | 1 |
| 42 | 10 | 1 |
| 43 | 10 | 2 |
| 44 | 9 | 308 |

| n | M(n,3) | TCAC |
|---|---|---|
| 47 | 11 | 1 |
| 48 | 10 | 1602 |
| 49 | 11 | 22 |
| 51 | 13 | 4 |
| 52 | 11 | 621 |
| 53 | 13 | 1 |
| 54 | 13 | 2 |
| 56 | 12 | 170 |
| 57 | 14 | 5 |
| 59 | 14 | 2 |
| 60 | 13 | 7702 |
| 61 | 15 | 1 |
| 63 | 15 | 46 |
| 64 | 13 | 101136 |
| 65 | 16 | 6 |
| 66 | 16 | 1 |
| 67 | 16 | 4 |
| 68 | 15 | 200 |
| 69 | 17 | 7 |
| 71 | 17 | 3 |
| 72 | 16 | 1333 |
| 73 | 17 | 80 |
| 75 | 19 | 4 |
| 76 | 16 | 377203 |
| 77 | 18 | 78 |
| 78 | 19 | 1 |
| 79 | 19 | 5 |

| n | M(n,3) | TCAC |
|---|---|---|
| 80 | 17 | 209575 |
| 81 | 19 | 1758 |
| 83 | 20 | 4 |
| 84 | 19 | 2464 |
| 85 | 21 | 10 |
| 87 | 22 | 2 |
| 88 | 19 | 39552 |
| 89 | 21 | 125 |
| 90 | 22 | 3 |
| 91 | 22 | 4 |
| 92 | 20 | 200224 |
| 93 | 23 | 6 |
| 95 | 23 | 2 |
| 96 | 21 | 3411597 |
| 97 | 24 | 1 |
| 99 | 24 | 40 |
| 100 | 22 | 40928 |
| 101 | 25 | 1 |
| 102 | 25 | 1 |
| 103 | 25 | 5 |
| 104 | 22 | $\geq$5000000 |
| 105 | 26 | 22 |
| 107 | 26 | 2 |
| 108 | 24 | $\geq$3600000 |
| 109 | 27 | 2 |
| 111 | 28 | 2 |

**Таблица 6.12: Плътни оптимални CAC с $k = 4$ и $n \leq 118$**

| n | M(n,3) | TCAC |
|---|---|---|
| 17 | 2 | 1 |
| 20 | 3 | 2 |
| 24 | 3 | 2 |
| 25 | 3 | 1 |
| 28 | 4 | 4 |
| 30 | 4 | 2 |
| 32 | 5 | 1 |
| 34 | 5 | 1 |
| 35 | 6 | 1 |
| 36 | 5 | 13 |
| 37 | 6 | 1 |
| 38 | 6 | 2 |
| 39 | 5 | 1 |
| 40 | 6 | 24 |
| 41 | 6 | 2 |
| 42 | 6 | 3 |
| 43 | 6 | 11 |
| 44 | 7 | 6 |
| 45 | 6 | 66 |
| 46 | 7 | 6 |
| 47 | 7 | 1 |
| 48 | 7 | 24 |
| 49 | 8 | 1 |
| 50 | 7 | 350 |
| 51 | 7 | 4 |
| 52 | 8 | 18 |
| 53 | 7 | 225 |
| 55 | 8 | 37 |
| 56 | 8 | 310 |

| n | M(n,4) | TCAC |
|---|---|---|
| 57 | 8 | 3 |
| 58 | 9 | 1 |
| 59 | 9 | 1 |
| 60 | 9 | 57 |
| 61 | 9 | 4 |
| 62 | 9 | 302 |
| 63 | 9 | 2 |
| 64 | 10 | 6 |
| 65 | 10 | 5 |
| 67 | 10 | 4 |
| 68 | 10 | 629 |
| 69 | 11 | 0 |
| 70 | 11 | 9 |
| 71 | 10 | 440 |
| 72 | 11 | 11 |
| 73 | 11 | 3 |
| 74 | 11 | 294 |
| 75 | 11 | 38 |
| 76 | 12 | 5 |
| 77 | 12 | 1 |
| 78 | 11 | 26106 |
| 79 | 12 | 3 |
| 80 | 13 | 2 |
| 82 | 12 | 3797 |
| 83 | 12 | 470 |
| 84 | 13 | 1 |
| 85 | 13 | 57 |
| 86 | 13 | 206 |
| 87 | 13 | 2 |

| n | M(n,4) | TCAC |
|---|---|---|
| 88 | 14 | 4 |
| 89 | 13 | 517 |
| 90 | 14 | 8 |
| 91 | 14 | 1 |
| 92 | 14 | 572 |
| 94 | 14 | 3827 |
| 95 | 14 | 4386 |
| 96 | 14 | 22577 |
| 97 | 15 | 2 |
| 98 | 15 | 72 |
| 99 | 14 | 160321 |
| 100 | 16 | 21 |
| 101 | 15 | 337 |
| 102 | 15 | 17812 |
| 104 | 17 | 2 |
| 105 | 16 | 11 |
| 106 | 16 | 2950 |
| 107 | 16 | 245 |
| 108 | 16 | 10812 |
| 109 | 16 | 6676 |
| 110 | 17 | 692 |
| 112 | 17 | 4858 |
| 113 | 17 | 231 |
| 114 | 17 | 9480 |
| 115 | 17 | 95927 |
| 116 | 18 | 340 |
| 117 | 17 | 162282 |
| 118 | 18 | 2455 |
| 120 | 18 | 45890 |

**Таблица 6.13: Плътни оптимални CAC с $k = 5$ и $n \leq 118$**

| n | M(n,5) | TCAC |
|---|---|---|
| 43 | 3 | 1 |
| 45 | 6 | 1 |
| 47 | 5 | 1 |
| 50 | 6 | 1 |
| 54 | 6 | 1 |
| 55 | 6 | 7 |
| 56 | 6 | 2 |
| 60 | 6 | 23 |
| 61 | 6 | 10 |
| 62 | 6 | 5 |
| 63 | 8 | 2 |
| 64 | 6 | 15 |
| 65 | 7 | 11 |
| 66 | 7 | 4 |
| 67 | 7 | 2 |
| 69 | 8 | 1 |
| 70 | 8 | 8 |
| 71 | 7 | 13 |
| 72 | 7 | 64 |

| n | M(n,5) | TCAC |
|---|---|---|
| 73 | 8 | 15 |
| 75 | 8 | 34 |
| 77 | 8 | 362 |
| 78 | 8 | 47 |
| 79 | 8 | 19 |
| 80 | 8 | 59 |
| 81 | 9 | 4 |
| 82 | 8 | 39 |
| 83 | 8 | 285 |
| 84 | 8 | 1302 |
| 85 | 9 | 88 |
| 86 | 9 | 1 |
| 88 | 9 | 2 |
| 89 | 9 | 84 |
| 91 | 10 | 6 |
| 93 | 10 | 9 |
| 95 | 11 | 13 |
| 96 | 9 | 4214 |
| 97 | 12 | 2 |

| n | M(n,5) | TCAC |
|---|---|---|
| 98 | 10 | 25 |
| 99 | 12 | 2 |
| 101 | 10 | 468 |
| 102 | 10 | 1942 |
| 103 | 10 | 2104 |
| 104 | 10 | 2028 |
| 105 | 11 | 1521 |
| 107 | 11 | 67 |
| 108 | 11 | 58 |
| 109 | 11 | 372 |
| 110 | 12 | 18 |
| 111 | 12 | 94 |
| 112 | 11 | 1064 |
| 113 | 12 | 6 |
| 114 | 12 | 13 |
| 115 | 13 | 44 |
| 117 | 14 | 2 |
| 118 | 12 | 239 |

**Таблица 6.14: Плътни оптимални CAC с $k = 6$ и $n \leq 119$**

| n | M(n,6) | TCAC |
|---|---|---|
| 42 | 3 | 1 |
| 62 | 4 | 1 |
| 66 | 5 | 1 |
| 69 | 5 | 1 |
| 72 | 5 | 1 |
| 77 | 8 | 1 |
| 84 | 8 | 5 |
| 88 | 7 | 2 |
| 89 | 6 | 2 |

| n | M(n,6) | TCAC |
|---|---|---|
| 90 | 7 | 1 |
| 91 | 8 | 20 |
| 94 | 8 | 1 |
| 96 | 8 | 2 |
| 98 | 8 | 45 |
| 99 | 8 | 9 |
| 100 | 8 | 3 |
| 103 | 8 | 4 |
| 104 | 8 | 13 |

| n | M(n,6) | TCAC |
|---|---|---|
| 107 | 8 | 38 |
| 108 | 9 | 2 |
| 111 | 9 | 1 |
| 112 | 9 | 6 |
| 116 | 9 | 4 |
| 117 | 9 | 60 |
| 119 | 10 | 9 |

**Таблица 6.15: Плътни оптимални CAC с $k = 7$ и $n \leq 95$**

| n | M(n,7) | TCAC |
|---|---|---|
| 60 | 2 | 1 |
| 63 | 4 | 2 |
| 91 | 8 | 1 |

## 6.5 Силни антиконфликтни кодове

### 6.5.1 Въведение

Докато CAC се използват в комуникационните системи с множествен достъп без обратна връзка, но със налична синхронизация по време (слот), то в случаите, когато такава синхронизация няма, се използват кодове, наречени *силни антиконфликтни кодове (SCAC)* [254],[255]. Когато няма синхронизация по време (слот) участниците в комуникацията не знаят кога започва слота и е възможно един пакет данни да попадне едновременно в два съседни слота.

**Пример 6.5.1.** *Да разгледаме антиконфликтния код от пример 6.4.1, но със слот-асинхронен канал и нека вторият участник е започнал малко по-рано. Тогава*

| *слот* | *0* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* | *11* | *12* | *13* | *14* | *15* | *16* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *участник 1* | | ===| ===| ===| | | | | | | | | | | | | |
| *участник 2* | == | = | | ==| = | | ==| == | | | | | | | | | |
| *участник 3* | | | ===| | | | ===| | | | ===| | | | | | |

*В тази ситуация само третият участник ще комуникира успешно. Един от начините да избегнем припокриването е да удвоим броя на слотовете и да разглеждаме код с дължина* 34 *с кодови думи {2,4,6}, {0,6,12}, {4,12,20}, {0,10,20}, получени чрез удвояване на елементите на кодовите думи на използвания CAC с дължина* 17*. Новият код е вече SCAC (виж Теорема 6.5.3).*

Както и при CAC, броят на кодовите думи в един такъв код определя броя на потенциалните участници в комуникацията през канала и затова особен интерес представляват кодовете с максимален брой думи (***оптимални кодове***) при фиксирани параметри.

В този раздел класифицираме с точност до мултипликативна еквивалентност оптималните SCAC за $k = 3, 4$ и 5 активни потребители и малки дължини.

Разглеждаме $A, B \subset Z_n$, такива, че $A \pm B = \{a \pm b \ (mod\ n) | a \in A, b \in B\}$. Означаваме с $P(n,k)$ множеството от всички $k$-елементни подмножества на $Z_n$. Всеки елемент $x \in P(n,k)$ съответства на двоична последователност с дължина $n$ и тегло $k$. Всеки *силен антиконфликтен код (SCAC)* с дължина $n$ и тегло $k$ може да бъде разглеждан като подмножество на $P(n,k)$. Елементите му наричаме *кодови думи*.

За всяка кодова дума $X \in P(n,k)$ означаваме *множеството от разликите* между всеки два елемента на $X$ с

$$d(X) = \{a - b \ (mod\ n) | a, b \in X\},$$

*множеството от ненулевите разлики* в $X$ с

$$d^*(X) = d(X) \backslash \{0\}$$

и с $d^*(X)'$ множеството

$$d^*(X)' = d^*(X) \cup (d^*(X) + \{1\}) \cup (d^*(X) - \{1\}).$$

Тогава CAC и SCAC можем да дефинираме чрез $d$, $d^*$ и $d^{*'}$.

**Дефиниция 6.5.1.** *Силен антиконфликтен код (SCAC) с дължина $n$ и тегло $k$ (за $k$ едновременно активни потребители) е подмножество $C = \{X_1, \ldots, X_s\} \subset P(n,k)$ удовлетворяващо условието, че за всяко $j \neq k$,*

$$d^*(X_j)' \cap d(X_k) = \emptyset.$$

Означаваме с $CAC(n,k)$ ($SCAC(n,k)$) класа от всички CAC (SCAC) с дължина $n$ и тегло $k$, а с $M(n,k)$ ($M_S(n,k)$) максималния размер (брой кодови думи) на кодовете от $CAC(n,k)$ ($SCAC(n,k)$). Код с максималния размер се нарича *оптимален*.

**Пример 6.5.2.** *SCAC с дължина* 55 *и тегло* 4.

$X_1 = \{0,11,22,33\}$ $d(X_1) = \{0,11,22,33,44\}$
$d^*(X_1)' = \{10,11,12,21,22,23,32,33,34,43,44,45\}$
$X_2 = \{0,2,4,6\}$ $d(X_2) = \{0,2,4,6,49,51,53\}$
$d^*(X_2)' = \{1,2,3,4,5,6,7,48,49,50,51,52,53,54\}$
$X_3 = \{0,13,27,41\}$ $d(X_3) = \{0,13,14,27,28,41,42\}$
$d^*(X_3)' = \{12,13,14,15,26,27,28,29,40,41,42,43\}$
$X_4 = \{0,8,17,25\}$ $d(X_4) = \{0,8,9,17,25,30,38,46,47\}$
$d^*(X_4)' = \{7,8,9,10,16,17,18,24,25,26,29,30,31,37,38,39,45,46,47,48\}$

За $X \in P(n,k)$ дефинираме

$$d^+(X) = d^*(X) + \{0,1\}.$$

Тогава

**Твърдение 6.5.1.** *([254])*
*$C = \{X_1, X_2, \ldots, X_M\} \in SCAC(n,k)$ тогава и само тогава, когато*
*(i) $\{1, n-1\} \cap d^*(X_j) = \emptyset$ за всяко $j$;*
*(ii) $d^+(X_j) \cap d^+(X_k) = \emptyset$ за всяко $j \neq k$.*

**Пример 6.5.3.** *SCAC с дължина* 55 *и тегло* 4.

$X_1 = \{0,11,22,33\}$ $d^+(X_1) = \{11,12,22,23,33,34,44,45\}$
$X_2 = \{0,2,4,6\}$ $d^+(X_2) = \{2,3,4,5,6,7,49,50,51,52,53,54\}$
$X_3 = \{0,13,27,41\}$ $d^+(X_3) = \{13,14,15,27,28,29,41,42,43\}$
$X_4 = \{0,8,17,25\}$ $d^+(X_4) = \{8,9,10,17,18,25,26,30,31,38,39,46,47,48\}$

**Дефиниция 6.5.2.** *Две кодови думи $X$ и $Y$ са **еквивалентни** ако $d(X) = d(Y)$.*

**Пример 6.5.4.** *Нека $n = 30$. Кодовите думи $\{0,6,12\}$ и $\{0,6,18\}$ са еквивалентни, $d = \{0,6,12,18,24\}$.*

**Дефиниция 6.5.3.** *Два SCAC с дължина $n$ и тегло $k$ са мултипликативно еквивалентни ако могат да бъдат получени един от друг чрез прилагане на автоморфизъм на $Z_n$ и замяна на кодова дума с някоя от нейните транслации.*

**Дефиниция 6.5.4.** *Код с максималния размер $(M_S(n,k))$ за дадените параметри се нарича* ***оптимален****.*

Оптималните кодове позволяват на най-голям брой потребители да комуникират успешно по канала.

За първи път SCAC са разглеждани в [254], където е изведена и горна граница за размера на такъв код и са конструирани SCAC с четни дължини.

**Теорема 6.5.2.** ([254]) *За $k \leq n \leq 2k^2$*

$$M_S(n,k) = 1.$$

**Теорема 6.5.3.** ([254]) *Ако съществува $C \in CAC(n,k)$ с $s$ кодови думи, то съществува $C' \in SCAC(2n,k)$ с $s$ кодови думи.*

**Теорема 6.5.4.** ([254, Th. 7]) *Нека $x$ дели $n$ и $f(x,k) = x-1-x(\lceil k/x \rceil + \lceil 2k/x \rceil + 3k)$. Разглеждаме множество $S = \{x_1, x_2, \ldots x_t\}$, където $f(x_i,k) > 0$ и $x_i$ и $x_j$ са делители на $n$, които са взаимно прости ($gcd(x_i,x_j) = 1$), $i \neq j$. Нека $\Phi(S,k) = \sum_{i=1}^{t} f(x_i,k)$. Дефинираме $F(n,k) = \Phi(S,k)$ ако $\Phi(S',k) \leq \Phi(S,k)$ за всяко друго множество $S' = \{x_1, x_2, \ldots x_{t'}\}$ от взаимно прости делители на $n$. Тогава за $n \geq 2k^2$*

$$M_S(n,k) \leq \left\lfloor \frac{n-2+F(n,k)}{3k-3} \right\rfloor$$

Изследванията на SCAC продължават в [252], [255] и [257], където [257] се отнася до използването на SCAC и CAC, докато [252] и [255] са посветени на SCAC с $k = 3$. В тях авторите независимо получават почти еднакви горни граници за броя на кодовите думи на SCAC с тегло три и четни дължини, които подобряват Теорема 6.5.4 когато $k = 3$.

**Теорема 6.5.5.** ([255, Th. 7], подобно в [252, Th. 18, Th. 19]) *Нека $n \geq 18$. Тогава*

$$M_S(n,3) \leq \begin{cases} n/8 & ako\ n \equiv 0\ (mod\ 8), \\ (n-4)/8 & ako\ n \equiv 4\ (mod\ 8), \\ (n+2)/8 & ako\ n \equiv 6\ (mod\ 24), \\ (n-2)/8 & ako\ n \equiv 2, 10, 18\ (mod\ 24), \\ (n-6)/8 & ako\ n \equiv 14, 22\ (mod\ 24). \end{cases}$$

Доказано е, че тази граница е плътна за SCAC с дължина $n \equiv 4\ (mod\ 8)$ [255, Cor.2] и когато $n$ удовлетворява условията на [255, Cor.2, Cor.5] и [252, Th.20, Th.25, Th.26]. Във всички тези случаи отбелязваме със (*) съответните $M_S(n,3)$ в Таблица 6.16.

Преди настоящото изследване $M_S(n,k)$ и примери за оптимални $(n,k)$ SCAC не бяха известни за $k = 3, 4, 5$ и голям брой малки дължини. Това ни мотивира за нашата класификация.

### 6.5.2 Допълнителните условия и използваните алгоритми

Поради връзката на SCAC с CAC и, съответно, с OOC, използваме модификация на алгоритъма, описан в раздел 6.4.2 за конструиране на всички кодове с $M_S(n,k)$ кодови думи, а на алгоритъма от част 6.2 за установяване на несъществуването на $(n,k)$ SCAC с $M_S(n,k)+1$ кодови думи.

Промените в двата алгоритъма засягат основно *възможните кодови думи* (които генерираме и подреждаме преди да започне търсенето), проверяването на повече условия при добавяне на следваща кодова дума и теста за минималност на частичните решения.

**Възможни кодови думи**:

От дефиниция 6.5.1 следва, че множеството от разликите на кодова дума $X$ на SCAC с дължина $n$ не може да съдържа 1 или $n-1$, т.е.

$$\{1, n-1\} \cap d^*(X) = \emptyset.$$

Отначало конструираме и запазваме всички $k$-елементни подмножества на $P(n,k)$, чиито множества от разликите не съдържат 1 или $n-1$, като ги подреждаме според мощността на множествата от разликите. Ако има няколко $k$-подмножества с еднакви множества от разликите, оставяме само лексикографски най-малкото от тях (Дефиниция 6.5.2). Запазваме и съответните множества $d^*$. Множеството от ненулевите разлики $d^*(X)$ е симетрично, т.е. ако $z \in d^*(X)$, то $n-z \in d^*(X)$. Затова пазим само разликите, които не са по-големи от $n/2$. Въпреки, че Твърдение 6.5.1 дава по-лесен начин за проверка на условията, на които трябва да отговаря един SCAC, ние не съхраняваме множеството $d^+(X)$, защото то не е симетрично и, съответно пазенето на $d^+$ разликите изисква двойно повече памет.

**Тест за минималност на частичните решения**:

Ако $C \in SCAC(n,k)$, то $C \in CAC(n,k)$. Затова можем да дефинираме мултипликативна еквивалентност на SCAC по същия начин като за CAC. Проблемът е, че класът на еквивалентност на един SCAC не е затворен относно прилагането на автоморфизъм на $Z_n$. Последното води до CAC, не всички от които са SCAC.

**Пример 6.5.5.** *Кодът {{0,8,16}, {0,2,4}, {0,6,19}, {0,10,21}} е* $(32,3)$ *SCAC. Той е еквивалентен на кода* $\{\{0,8,16\},\{0,1,2\},\{0,6,12\},\{0,7,14\}\}$*, получен от него с прилагане на автоморфизма на* $Z_{32}$ *който се изразява чрез умножение по* 3 *на елементите на кодовите думи и транслации, както следва:*

$\{0,8,16\} \overset{*3}{\to} \{0,16,24\} \overset{+16}{\to} \{0,8,16\}$

$\{0,2,4\} \overset{*3}{\to} \{0,6,12\}$

$\{0,6,19\} \overset{*3}{\to} \{0,18,25\} \overset{+14}{\to} \{0,7,14\}$

$\{0, 10, 21\} \overset{*3}{\to} \{0, 30, 31\} \overset{+2}{\to} \{0, 1, 2\}$

*Полученият код, обаче, е CAC, но не е SCAC, защото* $7 \in d^+(\{0, 6, 12\})$ *и* $7 \in d^+(\{0, 7, 14\})$.

Затова не е възможно да отхвърляме еквивалентни частични решения, така както правим при CAC и OOC. Даже ако еквивалентното частично решение отговаря на условията за SCAC, то може да се окаже неразширимо до SCAC, докато текущото да е разширимо. По тази причина тест за мимималност не прилагаме на частичните, а само на пълните решения.

### 6.5.3 Класификация - сравнение с известните конструкции и граници

Класификационните резултати са представени в таблици 6.16, 6.17 и 6.18. Дължината на кода е дадена в първата колона, предварително известните граници - във втората, $M_S(n, k)$ в третата и броят на мултипликативно нееквивалентните SCAC в четвъртата. Горните граници за $k = 3$ и четни $n$ са изчислени според Теорема 6.5.5, а във всички останали случаи според Теорема 6.5.4. Долните граници $M_{S_l}(n, k)$ са получени с Теорема 6.5.3 като са използвани размерите на съответните оптимални CAC, дадени в [T14]. Не можем, обаче, да сравним броя на $(n/2, k)$ CAC от [T14] с броя на $(n, k)$ SCAC с четни $n$ и $M_S(n, k) = M_{S_l}(n, k)$, защото в [T14] не е дефинирана еквивалентност на кодови думи (Дефиниция 6.5.2), което в някои случаи води до по-голям брой CAC.

**Пример 6.5.6.** *Два* $(15, 3)$ *CAC с по* 4 *кодови думи са получени в [T14], докато тук построяваме единствен (30,3) SCAC с* 4 *кодови думи. Ако и тук не бяхме ползвали дефиниция 6.5.2, щяхме да получим двата кода: {{0,10,20}, {0,2,4}, {0,8,16}, {0,6,12}} и {{0,10,20}, {0,2,4}, {0,8,16}, {0,6,18}} (виж пример 6.5.4).*

Точните стойности на $M_S(n, k)$ за $k = 3$ са отбелязани със звезда (*) ако са уточнени в [252] или [255]. Размерът на оптималните кодове съвпада с долната граница $M_{S_l}(n, k)$ за 83% от разглежданите четни дължини, $M_S(n, k) = M_{S_l}(n, k)+1$ в повечето от останалите случаи на четни $n$ и само $M_S(80, 3) = M_{S_l}(80, 3) + 2$ с единствен $(80, 3)$ SCAC от 10 кодови думи: {0,2,4}, {0,6,43}, {0,8,16}, {0,10,45}, {0,12,24}, {0,14,47}, {0,18,49}, {0,20,40}, {0,22,51}, {0,26,53}.

Горната граница от Теорема 6.5.4 се достига само от размера на $(25, 3)$ CAC с кодови думи {0,2,4}, {0,6,12}, {0,8,16}. За $k = 3$ и четни $n$, обаче, горната граница от Теорема 6.5.5 достигат 78% от разглежданите кодове с четни дължини. Докато долната граница за четни дължини е конструктивна, представените тук оптимални SCAC с нечетни дължини са първите примери за оптимални кодове с дадените параметри. Обикновено в даден диапазон от близки кодови дължини размерът на оптималните кодове с нечетни дължини е по-малък от този на съседните четни.

**Пример 6.5.7.** *Разглеждаме* $k = 4$ *и диапазона от кодови дължини от* $n = 87$ *до* $n = 91$. *Най-малката четна дължина със* 7 *кодови думи е* $n = 88$, *докато най-*

*малката нечетна с код със* 7 *кодови думи е* $n = 91$*, като съществува единствен* $(91, 4)$ *SCAC с кодови думи {0,6,12,18}, {0,4,33,62}, {0,10,37,64}, {0,14,35,70}, {0,16,41,66}, {0,2,45,47}, {0,8,31,39}.*

Всъщност най-атрактивните кодове в даден диапазон от кодови дължини са тези с максимален размер при сравнително малка дължина и обикновено техният брой е сравнително малък.

Всички кодове са свободно достъпни онлайн. Те могат да бъдат изтеглени от http://www.moi.math.bas.bg/∼svetlana и да бъдат използвани както в директни практически приложения, така и за по-нататъшни изследвания.

Резултатите в този раздел са съвместни с Цонка Байчева и са публикувани в [T17].

**Таблица 6.16: Оптимални SCAC с $k = 3$ и $n \leq 122$. Долни граници от [254], горни граници за четни $n$ от [255] и [252], за нечетни $n$ от [254].**

| n | bound | $M_S$ | num | n | bound | $M_S$ | num | n | bound | $M_S$ | num |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 18 | 2 | 2 | 1 | 53 | $\leq 8$ | 6 | 3 | 88 | 9 - 11 | 10 | 145 |
| 19 | $\leq 2$ | 2 | 1 | 54 | 6 - 7 | 6 | 82 | 89 | $\leq 14$ | 10 | 7 |
| 20 | * 2 | 2 | 3 | 55 | $\leq 8$ | 6 | 12 | 90 | 11 | 11 | 4 |
| 21 | $\leq 3$ | 2 | 6 | 56 | 6 - 7 | 6 | 166 | 91 | $\leq 15$ | 10 | 69 |
| 22 | 2 | 2 | 6 | 57 | $\leq 9$ | 6 | 210 | 92 | * 11 | 11 | 3 |
| 23 | $\leq 3$ | 2 | 8 | 58 | 7 | 7 | 2 | 93 | $\leq 15$ | 10 | 2428 |
| 24 | 3 | 3 | 1 | 59 | $\leq 9$ | 6 | 301 | 94 | 11 | 11 | 47 |
| 25 | $\leq 3$ | 3 | 1 | 60 | * 7 | 7 | 13 | 95 | $\leq 15$ | 10 | 2194 |
| 26 | * 3 | 3 | 2 | 61 | $\leq 9$ | 6 | 1163 | 96 | 10 - 12 | 11 | 548 |
| 27 | $\leq 4$ | 3 | 5 | 62 | 7 | 7 | 22 | 97 | $\leq 15$ | 10 | 14976 |
| 28 | * 3 | 3 | 7 | 63 | $\leq 10$ | 7 | 44 | 98 | 11 - 12 | 12 | 1 |
| 29 | $\leq 4$ | 3 | 8 | 64 | 7 - 8 | 7 | 109 | 99 | $\leq 16$ | 11 | 301 |
| 30 | * 4 | 4 | 1 | 65 | $\leq 10$ | 7 | 16 | 100 | * 12 | 12 | 2 |
| 31 | $\leq 4$ | 3 | 23 | 66 | 8 | 8 | 10 | 101 | $\leq 16$ | 11 | 198 |
| 32 | 3 - 4 | 4 | 2 | 67 | $\leq 10$ | 7 | 210 | 102 | 13 | 13 | 4 |
| 33 | $\leq 5$ | 4 | 6 | 68 | * 8 | 8 | 4 | 103 | $\leq 16$ | 11 | 1624 |
| 34 | * 4 | 4 | 8 | 69 | $\leq 11$ | 8 | 6 | 104 | 11 - 13 | 12 | 163 |
| 35 | $\leq 5$ | 4 | 5 | 70 | 8 | 8 | 15 | 105 | $\leq 17$ | 12 | 14 |
| 36 | * 4 | 4 | 40 | 71 | $\leq 11$ | 8 | 1 | 106 | 13 | 13 | 3 |
| 37 | $\leq 5$ | 4 | 20 | 72 | 8 - 9 | 8 | 741 | 107 | $\leq 17$ | 12 | 15 |
| 38 | 4 | 4 | 51 | 73 | 1$\leq$ 11 | 8 | 4 | 108 | * 13 | 13 | 14 |
| 39 | $\leq 6$ | 4 | 111 | 74 | 9 | 9 | 2 | 109 | $\leq 17$ | 12 | 126 |
| 40 | 4 - 5 | 4 | 114 | 75 | $\leq 12$ | 8 | 785 | 110 | 13 | 13 | 36 |
| 41 | $\leq 6$ | 4 | 162 | 76 | * 9 | 9 | 4 | 111 | $\leq 18$ | 13 | 6 |
| 42 | 5 | 5 | 11 | 77 | $\leq 12$ | 8 | 1040 | 112 | 12 - 14 | 13 | 84 |
| 43 | $\leq 6$ | 5 | 1 | 78 | 10 | 10 | 2 | 113 | $\leq 18$ | 13 | 1 |
| 44 | * 5 | 5 | 9 | 79 | $\leq 12$ | 8 | 3874 | 114 | 14 | 14 | 19 |
| 45 | $\leq 7$ | 5 | 23 | 80 | 8 - 10 | 10 | 1 | 115 | $\leq 18$ | 13 | 8 |
| 46 | 5 | 5 | 38 | 81 | $\leq 13$ | 9 | 103 | 116 | * 14 | 14 | 3 |
| 47 | $\leq 7$ | 5 | 32 | 82 | 10 | 10 | 3 | 117 | $\leq 19$ | 13 | 817 |
| 48 | 5 - 6 | 5 | 325 | 83 | $\leq 13$ | 9 | 122 | 118 | 14 | 14 | 91 |
| 49 | $\leq 8$ | 5 | 131 | 84 | * 10 | 10 | 12 | 119 | $\leq 19$ | 13 | 745 |
| 50 | * 6 | 6 | 2 | 85 | $\leq 13$ | 10 | 1 | 120 | 13 - 15 | 14 | 524 |
| 51 | $\leq 8$ | 6 | 1 | 86 | 10 | 10 | 39 | 121 | $\leq 19$ | 13 | 5161 |
| 52 | * 6 | 6 | 7 | 87 | $\leq 14$ | 10 | 15 | 122 | 15 | 15 | 2 |

**Таблица 6.17: Оптимални SCAC с $k = 4$ и $n \leq 133$. Границите са от [254].**

| n | bound | $M_S$ | num | n | bound | $M_S$ | num | n | bound | $M_S$ | num |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 32 | 2 - 3 | 2 | 5 | 66 | 4 - 7 | 5 | 1 | 100 | 7 - 11 | 7 | 27411 |
| 33 | $\leq 3$ | 2 | 3 | 67 | $\leq 7$ | 4 | 4288 | 101 | $\leq 11$ | 7 | 2641 |
| 34 | 2 - 3 | 2 | 21 | 68 | 5 - 7 | 5 | 42 | 102 | 7 - 11 | 7 | 13845 |
| 35 | $\leq 3$ | 2 | 15 | 69 | $\leq 7$ | 5 | 9 | 103 | $\leq 11$ | 7 | 8098 |
| 36 | 2 - 4 | 2 | 63 | 70 | 6 - 7 | 6 | 4 | 104 | 8 - 11 | 8 | 536 |
| 37 | $\leq 3$ | 2 | 50 | 71 | $\leq 7$ | 5 | 15 | 105 | $\leq 11$ | 8 | 3 |
| 38 | 2 - 4 | 2 | 116 | 72 | 5 - 8 | 6 | 5 | 106 | 7 - 11 | 8 | 1 |
| 39 | $\leq 4$ | 2 | 133 | 73 | $\leq 7$ | 6 | 2 | 107 | $\leq 11$ | 8 | 24 |
| 40 | 3 - 4 | 3 | 5 | 74 | 6 - 8 | 6 | 4 | 108 | 8 - 12 | 8 | 268 |
| 41 | $\leq 4$ | 3 | 2 | 75 | $\leq 8$ | 6 | 7 | 109 | $\leq 11$ | 8 | 14 |
| 42 | 3 - 4 | 3 | 4 | 76 | 6 - 8 | 6 | 7 | 110 | 8 - 12 | 8 | 6240 |
| 43 | $\leq 4$ | 3 | 3 | 77 | $\leq 8$ | 5 | 1939 | 111 | $\leq 12$ | 8 | 174 |
| 44 | 3 - 5 | 3 | 27 | 78 | 5 - 8 | 6 | 26 | 112 | 8 - 12 | 8 | 11646 |
| 45 | $\leq 5$ | 3 | 21 | 79 | $\leq 8$ | 6 | 60 | 113 | $\leq 12$ | 8 | 1454 |
| 46 | 2 - 5 | 3 | 44 | 80 | 6 - 9 | 6 | 625 | 114 | 8 - 12 | 8 | 7235 |
| 47 | $\leq 5$ | 3 | 35 | 81 | $\leq 9$ | 6 | 158 | 115 | $\leq 12$ | 8 | 9816 |
| 48 | 3 - 5 | 3 | 164 | 82 | 6 - 9 | 6 | 365 | 116 | 9 - 13 | 9 | 33 |
| 49 | $\leq 5$ | 3 | 103 | 83 | $\leq 9$ | 6 | 540 | 117 | $\leq 13$ | 8 | 11820 |
| 50 | 3 - 5 | 3 | 329 | 84 | 6 - 9 | 6 | 1139 | 118 | 9 - 13 | 9 | 71 |
| 51 | $\leq 5$ | 3 | 272 | 85 | $\leq 9$ | 6 | 2299 | 119 | $\leq 13$ | 8 | 40427 |
| 52 | 4 - 5 | 4 | 6 | 86 | 6 - 9 | 6 | 3635 | 120 | 9 - 13 | 9 | 577 |
| 53 | $\leq 5$ | 4 | 4 | 87 | $\leq 9$ | 6 | 2842 | 121 | $\leq 13$ | 9 | 2 |
| 54 | 3 - 6 | 4 | 9 | 88 | 7 - 9 | 7 | 59 | 122 | 9 - 13 | 9 | 180 |
| 55 | $\leq 6$ | 4 | 8 | 89 | $\leq 9$ | 6 | 8846 | 123 | $\leq 13$ | 9 | 9 |
| 56 | 4 - 6 | 4 | 63 | 90 | 6 - 10 | 7 | 10 | 124 | 9 - 13 | 9 | 6316 |
| 57 | $\leq 6$ | 4 | 23 | 91 | $\leq 9$ | 7 | 1 | 125 | $\leq 13$ | 9 | 490 |
| 58 | 4 - 6 | 4 | 81 | 92 | 7 - 10 | 7 | 71 | 126 | 9 - 14 | 9 | 2005 |
| 59 | $\leq 6$ | 4 | 71 | 93 | $\leq 10$ | 7 | 1 | 127 | $\leq 13$ | 9 | 634 |
| 60 | 4 - 6 | 4 | 544 | 94 | 7 - 10 | 7 | 207 | 128 | 10 - 14 | 10 | 15 |
| 61 | $\leq 6$ | 4 | 200 | 95 | $\leq 10$ | 7 | 75 | 129 | $\leq 14$ | 9 | 3156 |
| 62 | 4 - 6 | 4 | 770 | 96 | 7 - 10 | 7 | 1109 | 130 | 10 - 14 | 10 | 127 |
| 63 | $\leq 7$ | 4 | 490 | 97 | $\leq 10$ | 7 | 146 | 131 | $\leq 14$ | 9 | 15290 |
| 64 | 5 - 7 | 5 | 1 | 98 | 8 - 10 | 8 | 4 | 132 | 10 - 14 | 10 | 3 |
| 65 | $\leq 7$ | 4 | 2203 | 99 | $\leq 11$ | 7 | 817 | 133 | $\leq 14$ | 10 | 6 |

**Таблица 6.18: Оптимални SCAC с $k = 5$ и $n \leq 148$. Границите са от [254].**

| n | bound | $M_S$ | num | n | bound | $M_S$ | num | n | bound | $M_S$ | num |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 50 | 2 - 4 | 2 | 8 | 83 | $\leq 6$ | 4 | 49 | 116 | 7 - 9 | 7 | 3 |
| 51 | $\leq 4$ | 2 | 1 | 84 | 5 - 7 | 5 | 3 | 117 | $\leq 9$ | 6 | 2573 |
| 52 | 2 - 4 | 2 | 5 | 85 | $\leq 7$ | 4 | 470 | 118 | 6 - 9 | 6 | 10138 |
| 53 | $\leq 4$ | 2 | 8 | 86 | 3 - 7 | 4 | 355 | 119 | $\leq 9$ | 6 | 9809 |
| 54 | 2 - 4 | 2 | 20 | 87 | $\leq 7$ | 4 | 338 | 120 | 6 - 10 | 6 | 50721 |
| 55 | $\leq 5$ | 2 | 99 | 88 | 5 - 7 | 5 | 3 | 121 | $\leq 10$ | 6 | 24067 |
| 56 | 2 - 4 | 2 | 66 | 89 | $\leq 7$ | 5 | 6 | 122 | 6 - 10 | 6 | 47841 |
| 57 | $\leq 4$ | 2 | 89 | 90 | 6 - 7 | 6 | 1 | 123 | $\leq 10$ | 7 | 1 |
| 58 | 2 - 4 | 2 | 175 | 91 | $\leq 7$ | 5 | 4 | 124 | 6 - 10 | 6 | 112206 |
| 59 | $\leq 4$ | 2 | 242 | 92 | 4 - 7 | 5 | 9 | 125 | $\leq 10$ | 6 | 218330 |
| 60 | 3 - 5 | 3 | 4 | 93 | $\leq 7$ | 5 | 18 | 126 | 8 - 10 | 8 | 6 |
| 61 | $\leq 4$ | 3 | 1 | 94 | 5 - 7 | 5 | 32 | 127 | $\leq 10$ | 6 | 261767 |
| 62 | 3 - 5 | 3 | 3 | 95 | $\leq 8$ | 6 | 4 | 128 | 6 - 10 | 6 | 354417 |
| 63 | $\leq 5$ | 3 | 1 | 96 | 4 - 8 | 5 | 192 | 129 | $\leq 10$ | 7 | 66 |
| 64 | 2 - 5 | 2 | 1919 | 97 | $\leq 7$ | 5 | 65 | 130 | 7 - 11 | 7 | 3231 |
| 65 | $\leq 5$ | 3 | 21 | 98 | 5 - 8 | 6 | 2 | 131 | $\leq 10$ | 6 | 1114230 |
| 66 | 2 - 5 | 3 | 17 | 99 | $\leq 8$ | 5 | 244 | 132 | 7 - 11 | 7 | 620 |
| 67 | $\leq 5$ | 3 | 27 | 100 | 6 - 8 | 6 | 36 | 133 | $\leq 10$ | 8 | 6 |
| 68 | 2 - 5 | 3 | 23 | 101 | $\leq 8$ | 5 | 868 | 134 | 7 - 11 | 7 | 912 |
| 69 | $\leq 5$ | 3 | 14 | 102 | 5 - 8 | 5 | 1688 | 135 | $\leq 11$ | 7 | 112 |
| 70 | 4 - 6 | 4 | 3 | 103 | $\leq 8$ | 6 | 13 | 136 | 7 - 11 | 7 | 891 |
| 71 | $\leq 5$ | 3 | 100 | 104 | 6 - 8 | 6 | 5 | 137 | $\leq 11$ | 7 | 1740 |
| 72 | 3 - 6 | 4 | 2 | 105 | $\leq 8$ | 6 | 56 | 138 | 8 - 11 | 8 | 99 |
| 73 | $\leq 5$ | 3 | 279 | 106 | 5 - 8 | 6 | 40 | 139 | $\leq 11$ | 7 | 2939 |
| 74 | 4 - 6 | 4 | 5 | 107 | $\leq 8$ | 6 | 52 | 140 | 8 - 11 | 8 | 763 |
| 75 | $\leq 6$ | 4 | 3 | 108 | 6 - 9 | 6 | 228 | 141 | $\leq 11$ | 7 | 7498 |
| 76 | 3 - 6 | 4 | 2 | 109 | $\leq 8$ | 6 | 33 | 142 | 7 - 11 | 8 | 5 |
| 77 | $\leq 6$ | 4 | 9 | 110 | 6 - 9 | 6 | 2485 | 143 | $\leq 12$ | 7 | 17286 |
| 78 | 3 - 6 | 4 | 3 | 111 | $\leq 9$ | 6 | 486 | 144 | 7 - 12 | 8 | 8 |
| 79 | $\leq 6$ | 4 | 27 | 112 | 6 - 9 | 6 | 554 | 145 | $\leq 12$ | 8 | 6 |
| 80 | 4 - 6 | 4 | 56 | 113 | $\leq 9$ | 6 | 482 | 146 | 8 - 12 | 8 | 4100 |
| 81 | $\leq 6$ | 4 | 54 | 114 | 6 - 9 | 6 | 2092 | 147 | $\leq 12$ | 8 | 1030 |
| 82 | 3 - 6 | 4 | 63 | 115 | $\leq 9$ | 6 | 5325 | 148 | 8 - 12 | 8 | 2574 |

# Възможности за бъдеща работа

Много често интересните отворени случаи за близки до успешно разгледаните параметри се оказват невъзможни за класификация по абсолютно същия начин. Обикновено причината е в бързодействието на софтуера, но може да бъде и в значително по-големия брой на решенията, което затруднява както самото им получаване и изследване, така и практическата им приложимост.

Що се отнася до бързодействието, авторът е оптимист относно по-нататъшното успешно използване на модификации на описаните класификационни алгоритми. Опитът показва, че въпреки лошата си теоретична оценка, те имат големи възможности за повишаване на ефективността чрез отразяването на подходящи математически и други съображения. Що се отнася до големия брой решения, там най-доброто е да бъдат потърсени критерии за конструиране само на *представителна* част от тях. Тук ще изредим съвсем малка част от отворените проблеми, близки до разгледаните в дисертацията.

Все още са отворени огромен брой случай на дизайни с не много големи параметри, за които няма класификационни резултати, а за някои дори и въпросът за съществуването не е решен. Все още не са класифицирани, например, Щайнеровите системи от тройки от ред 21 и не е ясно дали сред тях има двойно-разрешими.

Напоследък $t$-спредовете и $t$-паралелизмите на проективни пространства намират все повече приложения при кодове, използвани в компютърните мрежи, а конструктивна класификация е направена за много малко параметри, като изобщо няма класификации на $t$-спредове и $t$-паралелизми за $t > 2$. Отворени са и чисто фундаментални въпроси като този за съществуването на други транзитивни 2-паралелизми, освен тези в $PG(5, 2)$. Теоретични конструкции на паралелизми за $q > 2$ има само в $PG(2^m - 1, q)$ и единственият пример за размерност, която не можем да представим като $2^m - 1$ са наскоро намерени паралелизми в $PG(5, 3)$ [85].

В последно време се работи много върху оптични ортогонални кодове, чиито думи могат да имат няколко различни дължини и тегла. За такива кодове засега няма известни класификационни резултати.

Убеждение на автора е, че съществуват и много други обекти, дори на пръв поглед съвсем различни от разглежданите в дисертацията, за чиято класификация могат да бъдат използвани алгоритми, подобни на описаните тук.

Разбира се, някои от повдигнатите в дисертацията въпроси, които са решени в нея за даден частен случай, в бъдеще биха могли да намерят и теоретично решение, например въпросът за това дали двойните на дизайна от точките и правите на проективните равнини от прост ред са уникално-разделими. Възможно е, също така, представените тук конкретни резултати да се окажат полезни за намирането на теоретични доказателства за съществуване на нови безкрайни фамилии от тези обекти.

# Библиография


[1] Ц. Байчева, *Оптимални кодове за контрол на грешки и оптични комуникации*, Дисертация за присъждане на научната степен доктор на науките, ИМИ, София, 2015 г.

[2] И. Буюклиев, *Алгоритмични подходи за изследване на линейни кодове*, Дисертация за присъждане научната степен доктор на математическите науки по научна специалност информатика, ИМИ, София, 2008.

[3] С. Железова, *Изследване и класификация на двойно-разрешими дизайни*, Дисертация за присъждане научната и образователна степен доктор, ИМИ, София, 2010.

[4] С.Н. Капралов, Алгоритми за генериране и изследване на орбитни матрици, Юбилейна научна сесия *100 години академик Л. Чакалов* (1986), 70 – 78.

[5] С.Н. Капралов, Комбинаторни 2-(28,8,4) дизайни с автоморфизми от ред 3, фиксиращи една точка, *Математика и математическо образование* (1987), 453 – 458.

[6] С.Н. Капралов, Симетрични 2-(31,10,3) дизайни с автоморфизми от ред 3, *Математика и математическо образование* (1988), 260 – 264.

[7] С.Н. Капралов, *Комбинаторни конфигурации и групи от автоморфизми*, Дисертация за присъждане научната степен кандидат на математическите науки, 1989, София.

[8] К. Манев, *Увод в дискретната математика*, 3-то издание, КЛМН, София, 2003.

[9] З. Матева, *Конфигурации с адамарови дизайни*, Дисертация за присъждане на образователната и научна степен доктор по научна специалност информатика, ТУ Варна, 2011.

[10] Ф. Соловьева, С. Топалова, О группах автоморфизмов совершенных двоичных кодов и систем Штейнера, *Проблемы передачи информаций* **36** (4) (2000), 53 – 58.
F. Solov'eva, S. Topalova, On the automorphism group of perfect binary codes and Steiner triple systems, *Problems of Information Transmission* **36** (4) (2000), 331 – 335.

[11] В.Д. Тончев, *Комбинаторни конфигурации. Дизайни, кодове, графи*, Наука и изкуство, София 1984.
V.D. Tonchev, *Combinatorial configurations*, Longman Scientific and Technical, New York, 1988.

[12] В.Д. Тончев, *Комбинаторни структури и кодове*, Университетско издателство Климент Охридски, София 1988.

[13] С. Топалова, Перечисление 2-(21,5,2) блок-схем с автоморфизмами нечетного простого порядка, *Дискретный анализ и исследование операций* (серия 1) **5** (1) (1998), 64 – 81.

[14] С. Топалова, *Конструиране и изследване на комбинаторни дизайни със зададени автоморфизми*, Дисертация за присъждане научната и образователна степен доктор, ИМИ, София, 1998.

[15] R.J.R. Abel, M. Buratti, Some progress on (v, 4, 1) difference families and optical orthogonal codes, *J. Comb. Theory A* **106** (2004), 59 – 75.

[16] R.P. Anstee, M.Jr. Hall, J. Thompson, Planes of order 10 do not have collineation of order 5, *J. Comb. Theory A* **29** (1980), 39 – 58.

[17] L. Bader, G. Lunardon, Desarguesian spreads, *Ricerche mat.* **60** (2011), 15 – 37.

[18] T. Baicheva, S.Topalova, Optimal (v,4,2,1) optical orthogonal codes with small parameters, *J. Comb. Des.* **20** (2) (2012), 142 – 160.

[19] T. Baicheva, S. Topalova, Classification results for (v, k, 1) cyclic difference families with small parameters, *Mathematics of Distances and Applications*, M. Deza, M. Petitjean, K. Markov eds., In International book series: Information Science and Computing, book **25**, (2012), 24 – 30.

[20] T. Baicheva, S. Topalova, Optimal (v,5,2,1) optical orthogonal codes of small v, *Appl. Algebr. Eng. Comm.*, **24** (3-4), (2013), 165 – 177.

[21] T. Baicheva, S. Topalova, Maximal $(v, 4, 3, 1)$ optical orthogonal codes for asynchronous transmission, In: Proc. Scientitfic Jubilee Conference 25 Years Faculty of Mathematics and Informatics, Veliko Tarnovo, Bulgaria, (2015), 80 – 85.

[22] T. Baicheva, S. Topalova, Classification of maximal optical orthogonal codes of weight 3 and small lengths. *Serdica Journal of Computing* **9** (1) (2015), 83 – 92. ISSN:1312-6555.

[23] T. Baicheva, S. Topalova, Classification of optical orthogonal codes and spreads by backtrack search on a parallel computer, Proc. Annual Workshop Coding Theory and Applications, Gabrovo, Bulgaria, (2011), 7 – 10.

[24] R. Baker, Partitioning the planes of $AG_{2m}(2)$ into 2-designs, *Discrete Math.*, **15** (1976), 205 – 211.

[25] J. Barát, A. Del Fra, S. Innamorati and L. Storme, Minimal blocking sets in PG(2,8) and maximal partial spreads in PG(3,8), *Design. Code. Cryptogr.* **31** (2004) 15 – 26.

[26] Th. Beth, D. Jungnickel, H. Lenz, *Design Theory*, Cambridge University Press, 1993.

[27] A. Betten, The packings of $PG(3, 3)$, *Design. Code. Cryptogr.* **79** (3) (2016), 583 – 595.

[28] A. Beutelspacher, On parallelisms in finite projective spaces, *Geometriae Dedicata*, **3** (1) (1974), 35 – 45.

[29] V.K. Bhargava, J.M. Stein, $(v, k, \lambda)$ configurations and self-dual codes, *Information and Control*, **28** (1975), 352 – 355.

[30] N. Biggs, *Finite groups of automorphisms*, Cambridge University Press, 1971.

[31] I.C.M. Bird, A.D. Keedwell, Design and applications of optical orthogonal codes - a survey, *Bull. Inst. Comb. Appl.* **11** (1994), 21 – 44.

[32] S. Bitan, T. Etzion, Constructions for optimal constant weight cyclically permutable codes and difference families, *IEEE T. Inform. Theory*, **41** (1) (1992) 77 – 87.

[33] A. Blokhuis, A.E. Brouwer, H.A. Wilbrink, Blocking sets in PG(2,p) for small p, and partial spreads in PG(3,7), *Adv. Geom.*, Special Issue (2003) 245 – 253.

[34] I. Bouyukliev, V. Fack, J. Winne, Hadamard matrices of order 36 and double-even self-dual [72,36,12] codes, *Proc. of EUROCOMB'2005*, Berlin, Germany, (2005), 93 – 98.

[35] I. Bouyukliev, On the binary projective codes with dimension 6, *Discrete Appl. Math.* **154** (12) (2006),1693 – 1708.

[36] 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* **2** World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, (2007), 126 – 151.

[37] S. Bouyuklieva, I. Bouyukliev, An algorithm for classifcation of binary self-dual codes, *IEEE Trans. Inform. Theory*, **58** (2012), 3933 – 3940.

[38] I. Bouyukliev, M. Dzhumalieva-Stoeva, On an algorithm for classification of binary self- dual codes with minimum distance four, Proc. Intern. Workshop on Algebraic and Combinatorial Coding Theory (ACCT 2012) (2012), 105 – 110.

[39] I. Bouyulkiev, M. Dzhumalieva-Stoeva, V. Monev, Classification of Binary Self-Dual Codes of Length 40, *IEEE Trans. Inform. Theory*, **61** (8) (2015), 4253 – 4258.

[40] Z. Božikov, Symmetric designs with parameters (69,17,4) and $F_{39}$ as a group of automorphisms, *J. Comb. Des.* **6** (4) (1998), 231 – 233.

[41] E.F. Brickell, V. Wei, Optical orthogonal codes and cyclic block designs, *Congr. Numer.* **58** (1987), 175 – 182.

[42] R. Bruck, Construction Problems of Finite Projective Planes, *Proc. Conf. Combinatorics*, University of North Carolina Press, (1967), 427 – 514.

[43] R.H. Bruck, H.J. Ryser, The nonexistence of certain finite projective planes, *Canadian J. Math.* **1** (1949), 88 – 93.

[44] A.A. Bruen, Partial spreads and replaceable nets, *Canadian J. Math.* **23** (1971), 381 – 392.

[45] A.A. Bruen and J. Thas, Partial spreads, Packings and Hermitian manifolds in $PG(3, q)$, *Math. Z.* **151** (1976), 207 – 214.

[46] M. Buratti, Cyclic designs with block size 4 and related optimal optical orthogonal codes, *Design. Code. Cryptogr.* **26** (2002), 111 – 125.

[47] M. Buratti, K. Momihara, A. Pasotti, New results on optimal $(v, 4, 2, 1)$ optical orthogonal codes, *Design. Code. Cryptogr.*, **58** (2011), 89 – 109.

[48] M. Buratti, A. Pasotti, Further progress on difference families with block size 4 or 5, *Design. Code. Cryptogr.* **56** (2010), 1 – 20.

[49] G. Butler, An inductive schema for computing conjugacy classes in permutation groups, *Math. Comput.*, **62** (1994), 363–383.

[50] A.R. Camina, A survey of the automorphism groups of block designs, *J. Comb. Des.* **2** (2) (1994), 79 – 100.

[51] J. Cannon, D. Holt, Computing conjugacy class representatives in permutation groups. *J. Algebra*, **300** (1) (2006), 213 – 222.

[52] J. Cannon, B. Souvignier, On the computation of conjugacy classes in permutation groups, In: W. Küchlin (Ed.), *Proc. of Intern. Symp. on Symbolic and Algebraic Computation*, Maui, 1997, ACM New York, NY, USA, (1997), 392 – 399.

[53] Y. Chang, R. Fuji-Hara, Y. Miao, Combinatorial constructions of optimal optical orthogonal codes with weight 4, *IEEE T. Inform. Theory*, **49** (2003), 1283 – 1292.

[54] Y. Chang, L. Ji, Optimal $(4up, 5, 1)$ optical orthogonal codes, *J. Comb. Des.* **12** (2004), 346 – 361.

*Discrete Math.* **279** (1–3) (2004), 135 – 151.

[55] Chen K., Wei R., Zhu L., Existence of $(q, 7, 1)$ difference families with $q$ a prime power, *J. Comb. Des.* **10** (2002), 126 – 138.

[56] Chen K., Zhu L., Existence of $(q, 6, 1)$ difference families with $q$ a prime power, *Design. Code. Cryptogr.* **15** (1998), 167 – 173.

[57] K. Chen, L. Zhu, Existence of $(q, k, 1)$ difference families with $q$ a prime power and $k = 4, 5$. *J. Comb. Des.* **7** (1999), 21 – 30.

[58] S. Chowla, H.J. Ryser, Combinatorial problems, *Canadian J. Math.* **2** (1950), 93 – 99.

[59] W. Chu and S. W. Golomb, A new recursive construction for optical orthogonal codes, *IEEE T. Inform. Theory*, **49** (11) (2003), 3072 – 3076.

[60] W. Chu, C.J. Colbourn, Optimal (n, 4, 2)- OOC of small order, *Discrete Math.* **279** (2004), 163 – 172.

[61] F.R.K. Chung, J.A. Salehi, V.K. Wei, Optical orthogonal codes: design, analysis and applications, *IEEE T. Inform. Theory* **35** (1989), 595 – 604.

[62] M. Cohen, C. Colbourn, L. Ives, A. Ling, Kirkman triple systems of order 21 with nontrivial automorphism group, *Math. Comput.* **71** (238) (2002), 873 – 881.

[63] C. J. Colbourn, On cyclic Steiner systems $S(2, 6, 91)$, *Abstracts Amer. Math. Soc.* **2** (1981).

[64] C. J. Colbourn, Embedding partial Steiner triple system is NP-complete, *J. Combin. Theory A* **35** (1983), 100 – 105.

[65] C. J.Colbourn, J. H. Dinitz editors, *Handbook of Combinatorial Designs*, 2nd edition, (Discrete mathematics and its applications, K. Rosen (eds.)), CRC Press, Boca Raton, FL., 2007.

[66] C.J. Colbourn, J.H. Dinitz, D.R. Stinson, Applications of combinatorial designs to communications, cryptography, and networking, In J.D. Lamb, D.A. Preece (eds.) *Surveys in combinatorics*, Cambridge University Press, London, 1999, 37 – 100.

[67] C.J. Colbourn, S.S. Magliveras, and D.R. Stinson, Steiner triple systems of order 19 with nontrivial automorphism group, *Math. Comput.* **59** (1992), 283 – 295.

[68] C.J. Colbourn, S.S. Magliveras, R.A. Mathon, Transitive Steiner and Kirkman triple systems of order 27, *Math. Comput.* **58** (1992), 441 – 449.

[69] C.J. Colbourn, R. Mathon, Steiner systems, *Handbook of Combinatorial Designs*, CRC Press, New York, 1996, 66 – 75.

[70] M.J. Colbourn and R.A. Mathon, On cyclic Steiner 2-designs, *Ann. Discrete Math.* **7** (1980), 215 – 253.

[71] C.J. Colbourn and A. Rosa, *Triple Systems*, Clarendon Press, Oxford, 1999.

[72] J. Cooper, J. Milas, W.D. Wallis, Hadamard equivalence, In: D.A. Holton, J. Seberry (eds), Combinatorial Mathematics, *Lect. Notes Math.*, **686**, Springer, Berlin, Heidelberg, 1978.

[73] R. Craigen, Hadamard matrices and designs, *Handbook of Combinatorial Designs*, Boca Raton, FL.,CRC Press, 1996, 370 - 377.

[74] R. Craigen, W. Wallis, Hadamard matrices, 1893-1993, Congressus Numeratum, 97 (1993), 99 – 129.

[75] R. Craigen, H. Kharaghani, Weaving Hadamard matrices with maximum excess and classes with small excess, *J. Comb. Des.* **12** (2004), 233 – 255.

[76] D. Crnković, On some symmetric designs with classical parameters, *Acta Math. Acad. Paedagog. Nyházi.* **22** (1) (2006), 5-18 (electronic), http://www.emis.de/journals/AMAPN.

[77] R.H.F. Denniston, Some packings of projective spaces, *Atti Accad. Naz. Lincei Rend. Cl. Sci. Fis. Mat. Natur.* **52** (8) (1972), 36 – 40.

[78] R.H.F. Denniston, Cyclic packings of the projective space of order 8, *Atti Accad. Naz. Lincei Rend.* **54** (1973), 373 – 377.

[79] St. Dougherty, T.A. Gulliver, M. Harada, Extremal binary self-dual codes, *IEEE T. Inform. Theory* **43** (6) (1997), 2036 – 2047.

[80] J. Doyen and R.M. Wilson, Embeddings of Steiner triple systems, *Discrete Math.* **5** (1973), 229 – 239.

[81] J. Eisfeld, L. Storme, (Partial) t-spreads and minimal t-covers in finite projective spaces, Lecture notes from the Socrates Intensive Course on Finite Geometry and its Applications, Ghent, April (2000).

[82] S. S. Epp, *Discrete Mathematics with Applications*, 4th edition, Brooks/Cole (2011), ISBN: 9780495391326.

[83] T. Etzion, N. Silberstein, Codes and designs related to lifted MRD codes, *IEEE T. Inform. Theory* **59** (2) (2013), 1004 – 1017.

[84] T. Etzion, A. Vardy, Error-correcting codes in projective space, *IEEE T. Inform. Theory* **57** (2) (2011), 1165 – 1173.

[85] T. Etzion, A. Vardy, Automorphisms of codes in the Grassmann scheme. arXiv:1210.5724 (October 2012).

[86] I. A. Faradžev, Constructive enumeration of combinatorial objects, in Problèmes Combinatoires et Théorie des Graphes, (Université d'Orsay, July 9 – 13, 1977), CNRS, Paris, (1978), 131 – 135.

[87] W. Feit, Automorphisms of symmetric ballanced incomplete block designs, *Math. Z.* **118** (1970), 40 – 49.

[88] V. Felsch, J. Neubüser, An algorithm for the computation of conjugacy classes and centralizers in p-groups, In: E.W. Ng (Ed.), Symbolic and Algebraic Computation, Marseille, 1979. In: *Lect. Notes Comput. Sc.* **72**, Springer-Verlag, Berlin, 1979, 452 – 465.

[89] R.A. Fisher, An examination of the different possible solutions of a problem in incomplete blocks, *Ann. Eugenics* **10** (1940), 52 – 75.

[90] J.W. Freeman, Reguli and pseudo-reguli in $PG(3, s^2)$, *Geometriae Dedicata* **9** (1980), 267 – 280.

[91] T. C. Frenz, *Computing Techniques for the Enumeration of Cyclic Steiner Systems*, Ph.D. Thesis, Rochester Institute of Technology School of Computer Science (1989).

[92] H. Fu, Y. Lin, M. Mishima, Optimal conflict-avoiding codes of even length and weight 3, *IEEE T. Inform. Theory* **56** (11) (2010), 5747 – 5756.

[93] H. Fu, Y. Lo, and K. Shum, Optimal conflict-avoiding codes of odd length and weight three, *Design. Code. Cryptogr.* **72** (2) (2014), 289 – 309.

[94] R. Fuji-Hara, Mutually 2-orthogonal resolutions of finite projective space, *Ars Combinatoria* **21** (1986), 163 – 166.

[95] R. Fuji-Hara, Y. Miao, Optical orthogonal codes: Their bounds and new optimal constructions, *IEEE T. Inform. Theory* **46** (2000), 2396 – 2406.

[96] GAP - Groups, Algorithms, Programming - a System for Computational Discrete Algebra (http://www.gap-system.org/)

[97] M. Genma, M. Mishima, M. Jimbo, Cyclic resolvability of cyclic Steiner 2-designs, *J. Comb. Des.* **5** (1997), 177 – 187.

[98] S. Georgiou, C. Koukouvinos, J. Seberry, Hadamard matrices, orthogonal designs and construction algorithms, In Designs 2002: Further Computational and Constructive Design Theory (W.D. Wallis (eds.)), Kluwer Academic Publishers, Norwell, Massachusetts, (2003), 133 – 205.

[99] S. Georgiou, C. Koukouvinos, S. Stylianou, On good matrices, skew Hadamard matrices and optimal designs, *Comput. Stat. Data An.* **41** (1) (2002), 171 – 184.

[100] S. Georgiou, I. Kotsireas, C. Koukouvinos, Astronomical Bounds for finding inequivalent Hadamard Matrices, *Mathematics of Computer Algebra and Analysis*, University of Waterloo, Waterloo, Ontario, June 4-6, 2005.

[101] P. Gibbons, P. Östergård, Computational methods in design theory, *In: Handbook of Combinatorial Designs, C. Colbourn, J. Dinitz (eds.)*, 2nd edition, (Discrete mathematics and its applications, K. Rosen (eds.)), CRC Press, Boca Raton, FL., (2007), 755 –782.

[102] E.N. Gilbert, Cyclically permutable error-correcting codes, *IEEE T. Inform. Theory*, **9** (1963), 175 – 180.

[103] N.A. Gordon, R. Shaw and L.H. Soicher, Classification of partial spreads in $PG(4, 2)$, Research Report (2004), 63, http://www.hull.ac.uk/maths/people/rs/staffdetails.html.

[104] A. Gruner and M. Huber, New Combinatorial Construction Techniques for Low-Density Parity-Check Codes and Systematic Repeat-Accumulate Codes, *IEEE T. Commun.* **60** (9) (2012), 2387 – 2395.

[105] M. Hall Jr., A survey of difference sets, *Proc. Amer. Math. Soc.*, **7** (1956), 975 – 986.

[106] M. Hall Jr., *Combinatorial theory*, J.Wiley & Sons, New York, 1986.

[107] N. Hamada, On the p-rank of the incidence matrix of a balanced or partially balanced incomplete block design and its applications to error-correcting codes, *Hiroshima Math. J.* **3** (1973), 153 - 226.

[108] H. Hanani, Balanced incomplete block designs and related designs, *Discrete Math.* **11** (1975), 255 – 369.

[109] M. Harada, H. Kimura, On extremal self-dual codes, *Math. J. Okayama Univ.* **37** (1995), 1 – 14.

[110] M. Harada, C. Lam and V. D. Tonchev, Symmetric (4,4)-nets and generalized Hadamard matrices over groups of order 4, *Design. Code. Cryptogr.* **34** (2005), 71 – 87.

[111] O. Heden, A maximal partial spread of size 45 in $PG(3,7)$, *Design. Code. Cryptogr.* **22** (2001), 331 – 334.

[112] O. Heden, S. Marcugini, F. Pambianco and L. Storme, On the non-existence of a maximal partial spread of size 76 in $PG(3,9)$, *Ars Combinatoria* **89** (2008), 369 – 382.

[113] O. Heden, G. Faina, S. Marcugini and F. Pambianco, The maximal size of a maximal partial spread in $PG(3,9)$, In: Trends in Incidence and Galois Geometry:a Tribute to Giuseppe Tallini, *Quaderni di Matematica* (F. Mazzocca, N. Melone adn D. Olanda (eds.)), **19** (2009), 77-112.

[114] M. Huber, Combinatorial Designs for Authentication and Secrecy Codes, *Foundations and Trends in Communications and Information Theory* **5** (6) (2010), 581 – 675.

[115] D. Hughes, F. Piper, *Design Theory*, Cambridge University Press, 1975.

[116] A. Hulpke, Conjugacy classes in finite permutation groups via homomorphic images, *Math. Comput.* **69** (232) (2004), 1633 – 1651.

[117] B. Huppert, *Endliche Gruppen I*, Springer-Verlag, 1967.

[118] N. Ito, On Hadamard groups, *J. Algebra* **168** (1994), 981 – 987.

[119] N. Ito, J.S. Leon, J.Q. Longyear, Classification of 3-(24,12,5) designs and 24-dimensional Hadamard matrices, *J. Comb. Theory A* **31** (1981), 66 – 93.

[120] Z. Janko, V.D. Tonchev, Cyclic 2-$(91,6,1)$ designs with multiplier automorphisms, *Discrete Math.* **97** (1991), 265–268.

[121] Z. Janko, Tr. Van Trung, Construction of a new symmetric block design for $(78,22,6)$ with the help of tactical decompositions, *J. Comb. Theory A* **40** (1985), 451 – 455.

[122] Z. Janko, Coset Enumeration in Groups and Constructions of Symmetric Designs, *Annals of Discrete Mathematics* **52** (1992), 275 – 277.

[123] V. Jha, N.L. Johnson, The classification of spreads in $PG(3,q)$ admitting linear groups of order $q(q+1)$, II. Even order, *Advances in Geometry*, Special Issue (2003), 271 – 313.

[124] V. Jha, N.L. Johnson, The classification of spreads in $PG(3,q)$ admitting linear groups of order $q(q+1)$, I. Odd order, *J. Geometry* **81** (2004), 46 – 80.

[125] M. Jimbo, M. Mishima, S. Janiszewski, A.Y. Teymorian, V.D. Tonchev, On conflict-avoiding codes of length $n = 4m$ for three active users, *IEEE T. Inform. Theory* **53** (8) (2007), 2732 – 2742.

[126] N.L. Johnson, Subplane Covered Nets, *Monographs and Textbooks in Pure and Applied Mathematics* **222** Marcel Dekker, New York, 2000.

[127] N.L. Johnson, Some new classes of finite parallelisms, *Note Math.* **20** (2) (2000/01), 77 – 88.

[128] N.L. Johnson, Two-Transitive Parallelisms, *Design. Code. Cryptogr.* **22** (2) (2001), 179 – 189.

[129] N.L. Johnson, Parallelisms of projective spaces, *J. Geom.* **76** (2003), 110–182.

[130] N.L. Johnson, Spreads in $PG(3,q)$ admitting several homology groups of order $q+1$, *Note di Matematica* **24** (2) (2005), 9 – 39.

[131] N.L. Johnson, *Combinatorics of Spreads and Parallelisms*, Series: Chapman & Hall Pure and Applied Mathematics, CRC Press, 2010.

[132] N.L. Johnson, A. Montinaro, The doubly transitive $t$-parallelisms, *Results Math.* **2** (2008), 75–89.

[133] N.L. Johnson, A. Montinaro, The transitive t-parallelisms of a finite projective space, *Adv. Geom.* **12** (2012), 401–429.

[134] N.L. Johnson, R. Pomareda, Transitive Partial Parallelisms of Deficiency One, *Eur. J. Combin.* **23** (8) (2002), 969 – 986.

[135] S.J. Johnson and S.R. Weller, Resolvable 2-Designs for Regular Low-Density Parity-Check Codes, *IEEE T. Commun.* **51** (9) (2003), 1413 – 1419.

[136] R. Julian, R. Abel, S. Costa, N. Finizio, Directed-ordered whist tournaments and $(v,5,1)$ difference families: existence results and some new classes of Z-cyclic solutions, *Discrete Applied Mathematics* 143 (2004), 43–53.

[137] D. Jungnickel, Quasimultiples of projective and affine planes, *J.Geometry* **26** (1986), 172 – 181.

[138] D. Jungnickel, On the existence of small quasimultiples of affine and projective planes of arbitrary order,*Discrete Math.* **85** (2) (1990), 177 – 189.

[139] S. Kageyama, A survey of resolvable solutions of balances incomplete block designs, *Rev. Inst. Internat. Stat.* **40** (1972), 269 – 273.

[140] S.N. Kapralov, 2-$(10,4,4)$ designs with automorphisms of order 2 fixing no point or block, *Proc. First Intern. Workshop on Algebraic and Combinatorial Coding Theory, Varna, Bulgaria* (1988), 112 - 115.

[141] S.N. Kapralov, The uniqueness of a group-divisible symmetric design on 30 points, *Mathematics and Education in Mathematics* (1989).

[142] S.N. Kapralov, I.N. Landgev, V.D. Tonchev, 2-$(25,10,6)$ designs invariant under the dihedral group of order 10, *Annals of Discrete Mathematics* **34** (1987), 301 – 306.

[143] S.N. Kapralov, S. Topalova, On the Steiner Triple Systems of order 21 with automorphisms of order 3, *Proc. Third International Workshop on Algebraic and Combinatorial Coding Theory, V.Voda, Bulgaria* (1992), 105 – 108.

[144] S.N. Kapralov, S. Topalova, Enumeration of 2-$(21, 6, 3)$ Designs with Automorphisms of Order 7 or 5, *Ars Combinatoria* **48** (1998), 135 - 146.

[145] H. Kharaghani and B. Tayfeh-Rezaie, On the classification of Hadamard matrices of order 32, http://www3.interscience.wiley.com/journal/123249643/abstract

[146] R.M. Karp, Reducibility among combinatorial problems, in Miller, R. E.; Thatcher, J. W., Proceedings of a Symposium on the Complexity of Computer Computations, Plenum Press (1972), 85 – 103.

[147] R.M. Karp, Y Zhang, Randomized parallel algorithms for backtrack search and branch-and-bound computation, *J. Assoc. Comput. Mach.* **40** (3) (1993), 765 – 789.

[148] P.Kaski, Isomoprph-free exhaustive generation of combinatorial designs, Helsinki University of Technology Laboratory for Theoretical Computer Science, Research Reports 70, 2002.

[149] P. Kaski, P. Östergård, Miscellaneous Classification Results for 2-Designs, *Discrete Math.* **280** (2004), 65 – 75.

[150] P. Kaski, P. Östergård, The Steiner Triple Systems of order 19, *Mathematics of Comput.* **73** (2004), 2075 – 2092.

[151] P. Kaski, P. Östergård, *Classification algorithms for codes and designs*, Springer, Berlin, 2006.

[152] H. Kimura, Hadamard matrices of order 28 with automorphism groups of order 2, *J. Comb. Theory A* **43** (1986), 98 – 102.

[153] H. Kimura, New Hadamard matrix of order 24, *Graphs and Combin.* **5** (1989), 235 – 242.

[154] H. Kimura, Classification of Hadamard matrices of order 28, *Discrete Math.* **133** (1994), 171 – 180.

[155] I. Kotsireas, Ch. Koukouvinos, J. Seberry, Hadamard ideals and Hadamard matrices with two circulant cores, *Eur. J. Combin.* **27** (5) (2006), 658 – 668.

[156] E.S. Kramer, S.S. Magliveras, R. Mathon, The Steiner systems $S(2, 4, 25)$ with nontrivial automorphism group, *Discrete Math.* **77** (1989), 137 – 157.

[157] C. Lam, S. Lam, V.D. Tonchev, Bounds on the number of Affine, Symmetric and Hadamard Designs and Matrices, *J. Comb. Theory A* **92** (2000), 186 – 196.

[158] C. Lam, S. Lam, V.D. Tonchev, Bounds on the number of Hadamard designs of even order, *J. Comb. Des.* **9** (2001), 363 – 378.

[159] C. Lam and Y. Miao, On Cyclically Resolvable Cyclic Steiner 2-Designs, *J. Comb. Theory A* **85** (1999), 194 – 207.

[160] C. Lam and Y. Miao, Cyclically resolvable cyclic Steiner triple systems of order 21 and 39, *Discrete Math.* **219** (2000), 173 – 185.

[161] E.S. Lander, Symmetric designs: an algebraic approach, *London Math. Soc. Lecture Note* **74**, Cambridge University Press, Cambridge, 1983.

[162] I. Landgev, On symmetric 2-$(41, 16, 6)$ designs invariant under the Frobenius group of order 10, *C. R. Acad. Bulg. Sci.* **40** (5) (1987), 29 – 31.

[163] I. Landgev, V.D. Tonchev, Automorphisms of 2-$(22, 8, 4)$ designs, *Annals of Discrete Math.* **77** (1989), 177 – 189.

[164] I. Landjev, S. Topalova, Symmetric 2-$(61, 16, 4)$ designs invariant under the dihedral group of order 10, *Serdica Journal of Computing* **23** (1997), 1001 – 1008.

[165] J. Leon, An algorithm for computing the automorphism group of a Hadamard matrix, *J. Comb. Theory A* **27** (1979), 289 – 306.

[166] V. I. Levenshtein, Conflict-avoiding codes and cyclic triple systems, *Probl. of Inform. Transm.* **43** (3) (2007), 199 – 212.

[167] V. I. Levenshtein, and V. D. Tonchev, Optimal conflict-avoiding codes for three active users, Proc. IEEE Internat. Symposium on Inform. Theory, Adelaide (2005), 535 – 537.

[168] Y. Lin, M. Mishima, J. Satoh, M. Jimbo, Optimal equi-difference conflict-avoiding codes of odd length and weight three, *Finite Fields Appl.* **26** (2014), 49 – 68.

[169] S. Ma, Y. Chang, A new class of optimal optical orthogonal codes with weight five, *IEEE Trans. Inform. Theory* **50** (2004), 1848 – 1850.

[170] S. Ma, Y. Chang, Constructions of optimal optical orthogonal codes with weight five, *J. Combin. Designs* **13** (2005), 54 – 69.

[171] W. Ma, C. Zhao, D. Shen, New optimal constructions of conflict-avoiding codes of odd length and weight 3, *Design. Code. Cryptogr.* **73** (3) (2014), 791 – 804.

[172] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error - Correcting Codes*, North Holland, 1977.

[173] F. Manganiello, E. Gorla, and J. Rosenthal, Spread codes and spread decoding in network coding, Proc. of International Symposium on Information Theory, Toronto, Ontario, Canada (2008), 881 – 885.

[174] H.B. Mann, A note on balanced incomplete block designs, *Ann. Math. Stat.* **40** (1969), 679 – 680.

[175] S. Martirosyan and A. J. Han Vinck, A construction for optical orthogonal codes with correlation 1, *IEICE Trans. Fundamentals* **E85-A** (1) (2002), 269 – 272.

[176] Z. Mateva, Doubles of Hadamard 2-$(15, 7, 3)$ designs, Proc. of Eleventh Intern. Workshop on Algebraic and Combinatorial Coding Theory, Pamporovo, Bulgaria, (2008), 210 – 214.

[177] Z. Mateva, Constructing a Canonical form of a Matrix in Several Problems about Combinatorial Designs, *Serdica Journal of Computing* 2 (4) (2008), 349 – 368.

[178] R. Mathon, Symmetric 2-$(31, 10, 3)$ designs with non-trivial automorphism group, *Ars Combinatoria* **25** (1988), 171-183.

[179] R. Mathon, K. Phelps, A. Rosa, A class of Steiner triple systems of order 21 and associated Kirkman systems, *Math. Comput.* **37** (1981), 209 – 222.

[180] R. Mathon, A. Rosa, The 4-rotational Steiner and Kirkman triple systems of order 21, *Ars Combinatoria* **17**A (1984), 221 – 250.

[181] R. Mathon, A. Rosa, Some results on the existence and enumeration of BIBDs, *Math. Report* **125** (1985), Dept.of Math.&Stat, McMaster univ., 33.

[182] R.Mathon, A. Rosa, 2-$(v, k, \lambda)$ designs of small order, *In: Handbook of Combinatorial Designs, C. Colbourn, J. Dinitz (eds.)*, 2nd edition, (Discrete mathematics and its applications, K. Rosen (eds.)), CRC Press, Boca Raton, FL., (2007), 25 - 57.

[183] R. Mathon, E. Spence, On 2-$(45, 12, 3)$ designs, *J. Comb. Des.* **4** (3) (1996), 155 – 175.

[184] I. Matulic-Bedenic, K. Horvatic-Baldasar, E. Kramer, Construction of new symmetric designs with parameters (66,26,10), *J. Comb. Des.* **3** (6) (1995), 405 – 410.

[185] V.C. Mavron, T. McDonough and V.D. Tonchev, On affine designs and Hadamard designs with line spreads, *J. Discrete Math.* **308** (2008), 2742 – 2750.

[186] B. McKay, Hadamard equivalence via graph isomorphism, *J. Discrete Math.* **27** (1979), 213 - 214.

[187] McKay B.D., *Nauty* user's guide (version 1.5), Technical Report TR-CS-90-02, Computer Science Department, Australian National University, 1990.

[188] M. Mecky, J. Neubüser, Some remarks on the computation of conjugacy classes of soluble groups, *Bull. Austral. Math. Soc.* **40** (1989), 281 – 292.

[189] M. Mishima, H. Fu, S. Uruno, Optimal conflict-avoiding codes of length $n \equiv 0$ (*mod* 16) and weight 3, Design. Code. Cryptogr. **52**(3), 275–291 (2009).

[190] M. Mishima and M. Jimbo, Some types of cyclically resolvable cyclic Steiner 2-designs, *Congr.Numer.* **123** (1997), 193 – 203.

[191] M. Mishima and M. Jimbo, Recursive constructions for cyclic quasiframes and cyclically resolvable cyclic Steiner 2-designs, *Discrete Math.* **211** (2000) 135 – 152.

[192] K. Momihara, Necessary and sufficient conditions for tight equi-difference conflict avoiding codes of weight 3. *Design. Code. Cryptogr.* **45** (3) (2007), 379 – 390.

[193] K. Momihara,, M. Müler, J. Saton, and M. Jimbo, Constant weight conflict-avoiding codes, *SIAM J. Discr. Math.* **21** (4) (2007), 959 – 979.

[194] O. Moreno, Z. Zhang, P. V. Kumar, V. A. Zinoviev, New constructions of optimal cyclically permutable constant weight codes, *IEEE T. Inform. Theory* **41** (2) (1995), 448 – 455.

*Ars Combin.* **3** (1977), 233 – 250.

[195] M. Muzychuk, A solution of the isomorphism problem for circulant graphs, *Proc. London Math. Soc.* **88** (3) (2004), 1 – 41.

[196] Q. A. Nguyen, L. Gyöfri, J. L. Massey, Constructions of binary constant weight cyclic codes and cyclically permutable codes, *IEEE T. Inform. Theory* **38** (3) (1992), 940 – 949.

[197] P. Östergård, T. Baicheva and E. Kolev, Optimal Binary One-error-correcting Codes of Length 10 have 72 Codewords, *IEEE T. Inform. Theory* **45** (4) (1999), 1229 – 1231.

[198] R.E.A.C. Paley, On orthogonal matrices, *J. Math. Phys. MIT* **12** (1933), 311 – 320.

[199] P. Pálfy, Isomorphism problem for relational structures with a cyclic automorphism, *Eur. J. Combin.* **8** (1987) 35 – 43.

[200] E.T. Parker, On collineations of symmetric designs, *Proc. Amer. Math. Soc.* **8** (1957), 350 – 351.

[201] T. Penttila, B. Williams, Regular packings of $PG(3,q)$*Eur. J. Combin.* **19** (6) (1998), 713 – 720.

[202] F. Piper, M. Walker, Linear Ciphers and Spreads, *J. Cryptol.* **1** (1989), 185 – 188.

[203] A. Prince, Parallelisms of $PG(3,3)$ invariant under a collineation of order 5, In: Mostly Finite Geometries, Iowa City, 1996, N.Johnson, (eds.), *Lect. Notes Pure Appl.* **190**, Marcel Dekker, New York (1997), 383–390.

[204] A. Prince, The cyclic parallelisms of $PG(3,5)$, *Eur. J. Combin.* **19** (5) (1998), 613–616.

[205] A. Rahilly, On the line structure of designs, *Discrete Math.* **92** (1991), 291 – 303.

[206] R. C. Read, Every one a winner; or, How to avoid isomorphism search when cataloguing combinatorial configurations, *Ann. Discrete Math.* **2** (1978), 107 – 120.

[207] C. Reid, A. Rosa, Steiner systems $S(2,4,v)$ - a survey, *Electron. J. Comb.* (2010), DS18.

[208] J. Sarmiento, Resolutions of $PG(5,2)$ with point-cyclic automorphism group, *J. Comb. Des.* **8** (1) (2000), 2 – 14.

[209] J. Sarmiento, On point-cyclic resolutions of the 2-$(63,7,15)$ design associated with $PG(5,2)$, *Graph. Combinator.* **18** (3) (2002), 621 – 632.

[210] J. Seberry , M. Yamada, Hadamard matrices, sequences, and block designs, *In: Contemporary Design Theory, a collection* of surveys (J.H.Dinitz and D.R.Stinson (eds.)), Wiley, 1992, 431 - 560.

[211] N.V. Semakov and V.A. Zinoviev, Equidistant $q$-ary codes with maximal distance and resolvable balanced incomplete block designs, *Probl. Inf. Transm.* **4** (2) (1968), 1 – 7.

[212] S.S. Shrikhande, N.M. Singhi, Construction of geomatroids, *Utilitas Mathematica* **8**) (1975), 187 - 192.

[213] R. Shull, The classification of projective planes of order 9 possessing collineations of order 5, *Algebras, groups and geometries* **3** (1985), 365 – 379.

[214] K. W. Shum, S. W. Wong, A tight asymptotic bound on the size of constant-weight conflict-avoiding codes, *Design. Code. Cryptogr.* **57** (1) (2010), 1 – 14.

[215] K. W. Shum,, S. W. Wong, and C. S. Chen, A general upper bound on the size of constant-weight conflict avoiding codes, *IEEE T. Inform. Theory* **56** (7) (2010), 3265 – 3276.

[216] N. Silberstein, Coding Theory and Projective Spaces, Ph.D. thesis, Israel Institute of Technology, Haifa (2011).

[217] L. Soicher, Computation of Partial Spreads, web preprint, http://www.maths.qmul.ac.uk/∼ leonard/partialspreads (2000),

[218] E. Spence, Symmetric $(31, 10, 3)$-designs with a non-trivial authomorphism of odd order, *JCMCC* **10** (1991), 51 – 64.

[219] E. Spence, Symmetric $(41, 16, 6)$-designs with a nontrivial authomorphism of odd order, *J. Comb. Des.***1** (3) (1993), 193 – 211.

[220] E. Spence, Classification of Hadamard Matrices of Orders 24 and 28, *Discrete Math.* **140** (1995), 185 – 243.

[221] E. Spence, Regular two-graphs on 36 vertices, *Linear Algebra Appl.* **226-228** (1995), 459 – 497.

[222] D. Stinson, *Combinatorial Designs: Constructions and Analysis*, Springer-Verlag, New York, (2004).

[223] D. Stinson, S. Vanstone, Orthogonal packings in PG(5,2), *Aequationes Math.* **31** (1) (1986), 159–168.

[224] D. R. Stinson, R. Wei, J. Yin, Packings, *In: Handbook of Combinatorial Designs, C. Colbourn, J. Dinitz (eds.)*, 2nd edition, (Discrete mathematics and its applications, K. Rosen (eds.)), CRC Press, Boca Raton, FL., (2007), 549 – 555.

[225] L. Storme, Finite Geometry, *In: Handbook of Combinatorial Designs, C. Colbourn, J. Dinitz (eds.)*, 2nd edition, (Discrete mathematics and its applications, K. Rosen (eds.)), CRC Press, Boca Raton, FL., (2007), 702–729.

[226] Y. Tang, J.Yin, Combinatorial constructions for a class of optimal OOCs, *Science in China (A)* **45** (2002), 1268-1275.

[227] V.D. Tonchev, Hadamard matrices of order 28 with automorphisms of order 13, *J. Comb. Theory A* **35** (1983), 43 – 57.

[228] V.D. Tonchev, Hadamard matrices of order 28 with automorphisms of order 7, *J. Comb. Theory A* **40** (1985), 62 – 81.

[229] V.D. Tonchev, Hadamard matrices of order 36 with automorphisms of order 17, *Nagoya Math.J.* **104** (1986), 163 – 174.

[230] V.D. Tonchev, Quasi-symmetric 2-(31,7,7) designs and a revision of Hamada's conjecture, *J. Comb. Theory A* **42** (1986), 104 – 110.

[231] Tonchev V.D., Steiner triple systems of order 21 with automorphisms of order 7, *Ars Combinatoria* **23** (1987), 93-96.

[232] Tonchev V.D., Symmetric 2-(31,10,3) designs with automorphisms of order 7, *Annals of Discrete Math.* **34** (1987), 461 – 464.

[233] V.D. Tonchev, Symmetric designs without ovals and extremal self-dual codes, *Annals of Discrete Math.* **37** (1988), 451 – 458.

[234] V.D. Tonchev, *Tables of Conflict-Avoiding Codes*, available online at `http://www.math.mtu.edu/~tonchev/CAC.html`.

[235] V.D. Tonchev, On Affine Designs and GMW Difference Sets, *In: Finite Geometries, Groups, and Computation*, A. Hulpke, R. Liebler, T. Penttila and A. Seress (eds.), Walter de Gruyter, Berlin, (2006), 237 – 245.

[236] V.D. Tonchev, Steiner systems for two-stage disjunctive testing, *J. Comb. Optim.* **15** (1) (2008), 1 – 6.

[237] V.D. Tonchev and S.A. Vanstone, On Kirkman triple systems of order 33, *Discrete Math.* **106/107** (1992), 493 – 496.

[238] S. Topalova, S. Zhelezova, On transitive parallelisms of $PG(3,4)$, *Appl. Algebr. Eng. Comm.* **24** (3-4) (2013), 159 – 164.

[239] S. Topalova, S. Zhelezova, On point-transitive and transitive deficiency one parallelisms of $PG(3,4)$, *Designs Codes and Cryptography* **75** (1) (2015), 9 – 19.

[240] S. Topalova, S. Zhelezova, New regular parallelisms of $PG(3,5)$, *J. Comb. Des.* **24** (10) (2016), 473 – 482.

[241] S. Topalova, S. Zhelezova, Types of spreads and duality of the parallelisms of $PG(3,5)$ with automorphisms of order 13, *Design. Code. Cryptogr.* (2018) https://doi.org/10.1007/s10623-018-0558-2.

[242] Tr. van Trung, The existence of symmetric block designs with parameters $(41,16,6)$ and $(66,26,10)$, *J. Comb. Theory A* **33** (1982), 201 – 204.

[243] E.R. Van Dam, Classification of spreads of $PG(3,4) \setminus PG(3,2)$, *Design. Code. Cryptogr.* **3** (1993), 193 – 198.

[244] X. Wang, Y. Chang, Further results on $(v,4,1)$-perfect difference families, *Discrete Math.* **310** (13 – 14) (2010), 1995 – 2006.

[245] S. Wang, L. Wang, J. Wang, A new class of optimal optical orthogonal codes with weight six, Proc. Seventh International Workshop of Signal Design and its Applications in Communications, April (2016).

[246] M. Weller, Construction of classes of subgroups of small index in $p$-groups, *Arch. Math.* **68** (2) (1997), 89 – 99.

[247] S.H. Whitesides, Fixed point free collineations of order 7 in projective planes of order 9, *Algebras, Groups and Geometries* **2** (1985), 564 – 578.

[248] H. Wielandt, *Finite permutation groups*, Academic Press, 1964.

[249] R.M. Wilson, Nonisomorphic Steiner triple systems, *Math. Z.* **135** (1974), 303 – 313.

[250] S.L. Wu, H.L. Fu, Optimal tight equi-difference conflict-avoiding codes of length $n = 2^k \pm 1$ and weight 3, *J. Comb. Des.* **21** (6) (2013), 223 – 231.

[251] J. Yin, Some combinatorial constructions for optical orthogonal codes, *Discrere Math.* **185** (1998), 201-219.

[252] Z. Yu and J. Wang, Strongly conflict-avoiding codes with weight three, *Wireless Pers. Commun.* **84** (2015), 153 – 165.

[253] G. Zaitsev, V. Zinoviev, N. Semakov, Interrelation of Preparata and Hamming codes and extension of Hamming codes to new double-errorcorrecting codes, Proc. of Second Intern. Symp. on Information Theory (Armenia, USSR, 1971), Budapest, Academiai Kiado, (1973), 257 – 263.

[254] Y. Zhang, K.W. Shum, and W.S. Wong, Strongly conflict-avoiding codes, *SIAM J. Discrete Math.* **25** (3) (2011), 1035 – 1053, .

[255] Y. Zhang, Y. Lo, and W. S. Wong, Optimal strongly conflict-avoiding codes of even length and weight three, *Design. Code. Cryptogr.* **79** (2) (2016), 367 – 382.

[256] Y. Zhang, M. Peng, S. Yang, A clique-based online algorithm for constructing optical orthogonal codes, *Appl. Soft Comput.* **47** (2016), 21 – 32.

[257] Y. Zhang, J. Wei, M. Zhang, A. Zou, and F. Shu, A performance comparison between CACs and SCACs based topology-transparent scheduling, in *Intern. Conference on Wireless Communications and Signal Processing (WCSP)*, Nanjing, Jiangsu, China, October 15 – 17 (2015), 1–5.

[258] S. Zhelezova, Parallel class intersection matrices of orthogonal resolutions, *Serdica Journal of Computing* **2** (4) (2008), 321 – 330.

[259] S. Zhelezova, Cyclic parallelisms of $PG(5,2)$, *Math. Balkanica* **24** (1-2) (2010), 141–146.

[260] S. Zhelezova, A method for classification of doubly resolvable designs and its application, *Serdica Journal of Computing* 5 (2011), 273 – 308.

[261] S. Zhelezova, On regular parallelisms of PG(3,5) with automorphisms of order 5, *Science. Business. Society* **1** (6) (2016), 3 – 5.

[262] S. Zhelezova, A method for constructing splitting (v,u x c, lambda) BIBDs, *Cybernetics and Information Technologies* **18** (5) (2018), 87 – 94.

**Публикации по дисертацията**


[T1] S. Topalova, Symmetric 2-$(69, 17, 4)$ designs with automorphisms of order 13, *Journal of Statistical Planning and Inference* **95** (2001), 335 – 339. ISSN 0378-3758, SJR(2001) = 0.711, ISI IF (2001) = 0.366.

[T2] S. Topalova, Classification of Hadamard matrices of order 44 with automorphisms of order 7, *Discrete Mathematics* **260** (2003), 275 – 283. ISSN 0012-365X, SJR(2003) = 0.775, ISI IF (2003) = 0.303.

[T3] S. Topalova, STS(21) with automorphisms of order 3 with 3 fixed points and 7 fixed blocks, *Mathematica Balkanica* **18** (2004), 215 – 220. ISSN 0205-3217.

[T4] V. Fack, S. Topalova, J. Winne, R. Zlatarski, Enumeration of the doubles of the projective plane of order 4, *Discrete Mathematics* **306** (2006) 2141 – 2151. ISSN 0012-365X, SJR(2006) = 0.867, ISI IF (2006) = 0.347.

[T5] P. Kaski, P. Östergård, S. Topalova, R. Zlatarski, Steiner Triple Systems of Order 19 and 21 with Subsystems of Order 7, *Discrete Mathematics* **308** (2008), 2732 – 2741. ISSN 0012-365X, SJR(2008) = 0.894, ISI IF (2008) = 0.502.

[T6] Z. Mateva, S. Topalova, Hadamard 2-(63,31,15) designs invariant under the dihedral group of order 10, *Discrete Mathematics* **309** (6) (2009), 1347 – 1356. ISSN 0012-365X, SJR(2009) = 0.914, ISI IF (2009) = 0.548.

[T7] Z. Mateva, S. Topalova, Line spreads of $PG(5, 2)$, *Journal of Combinatorial Designs* **17** (1) (2009), 90 – 102. ISSN 1063-8539, SJR(2009)=0.996, ISI IF (2009) = 0.709.

[T8] S. Topalova, S. Zhelezova, 2-spreads and transitive and orthogonal 2-parallelisms of PG(5,2), *Graphs and Combinatorics* **26** (5) (2010), 727 – 735. ISSN 0911-0119, SJR(2010)=0.729, ISI IF (2010) = 0.242.

[T9] T. Baicheva, S. Topalova, Classification of optimal $(v, 4, 1)$ binary cyclically permutable constant weight codes and cyclic $S(2, 4, v)$ designs with $v \leq 76$, *Problems of Information Transmission* **47** (3) (2011), 224 – 231. ISSN 0032 - 9460, SJR(2011)=0.467, ISI IF (2011) = 0.484.

[T10] T.P. McDonough, R. Shaw, S. Topalova, Classification of book spreads in $PG(5, 2)$, *Note di Matematica* **33** (2), (2013), 43-64. ISSN: 1123-2536, e-ISSN: 1590-0932, SJR(2013) = 0.162.

[T11] R. Shaw, S. Topalova, Book spreads in $PG(7, 2)$, *Discrete Mathematics* **330** (2014), 76 – 86. ISSN: 0012-365X, SJR(2014) = 1.03, SJR(2014) = 0.974, IF (2014) = 0.557.

[T12] S. Topalova, Conjugates for finding the automorphism group and isomorphism of design resolutions, *Serdica Journal of Computing* **10** (1), ИМИ, (2016), 79 – 92. ISSN:1312-6555.

[T13] S. Topalova, On the resolutions of cyclic Steiner triple systems with small parameters, *Journal of Algebra Combinatorics Discrete Structures and Applications* **3** (3) (2016), 201 – 208. ISSN: 2148-838X (online).

[T14] T. Baicheva and S. Topalova, Optimal conflict-avoiding codes for 3, 4 and 5 active users, *Problems of Information Transmission* **53** (1) (2017), 42 – 50. ISSN: 0032-9460 (print version) ISSN: 1608-3253 (electronic version), SJR(2017): 0.353, ISI IF(2017):0.359.

[T15] T. Baicheva and S. Topalova, Classification of optimal conflict-avoiding codes of weights 6 and 7, *Electronic Notes in Discrete Mathematics* **57**, Elsevier, (2017), 9 – 14, ISSN:1571-0653, SJR(2017):0.262.

[T16] T. Baicheva and S. Topalova, On tight optimal conflict-avoiding codes for 3, 4, 5 and 6 active users, *Cybernetics and Information Technologies* **18** (5) (2018), 5-11. ISSN:1311-9702, SJR(2017):0.204.

[T17] T. Baicheva and S. Topalova, Classification of strongly conflict-avoiding codes, *IEEE Communications Letters* **22** (12) (2018), 2415 – 2418, Print ISSN: 1089-7798, Electronic ISSN: 1558-2558, SJR(2017):0.589, IF(2017) = 2.723.

[T18] T. Baicheva and S. Topalova, Classification of optimal $(v, k, 1)$ binary cyclically permutable constant weight codes with $k = 5$, 6 and 7 and small lengths, *Designs Codes and Cryptography* (2018), Online first https://doi.org/10.1007/s10623-018-0534-x. ISSN: 0925-1022, e-ISSN: 1573-7586, SJR(2017) = 0.549, IF(2017) = 1.114.

**Цитирания на публикациите по дисертацията**
(Подредбата е според вида на изданието с цитата и по години.)

**Цитирания в списания с импакт фактор:**

T1. **S. Topalova**, Symmetric 2-$(69, 17, 4)$ designs with automorphisms of order 13, *Journal of Statistical Planning and Inference* **95** (2001), 335 – 339.

1. Kaski P., Isomorph-free exhaustive generation of designs with prescribed groups of automorphisms, *SIAM Journal on Discrete Mathematics* 19 (3) (2005), 664-690.
2. Krčadinac V., Nakić A., Pavčević M. O., The Kramer – Mesner method with tactical decompositions: some new unitals on 65 points, Journal of Combinatorial Designs 19 (2011), 290 – 303.

T2. **S. Topalova**, Classification of Hadamard matrices of order 44 with automorphisms of order 7, *Discrete Mathematics* **260** (2003), 275 – 283.

3. Goodwin V., Yorgov V., New extremal self-dual doubly-even binary codes of length 88, *Finite Fields and Their Applications* 11 (1) ( 2005), 1-5.
4. Kaski P., Isomorph-free exhaustive generation of designs with prescribed groups of automorphisms, *SIAM Journal on Discrete Mathematics* 19 (3) (2005), 664-690.

T2c. **S. Topalova**, Hadamard matrices of order 44 with automorphisms of order 7, Proceedings of the Seventh International Workshop on Algebraic and Combinatorial Coding Theory, Bansko, Bulgaria (2000), 305-310.

5. Gulliver A., Harada M., Kim J., Construction of new extremal self-dual codes, *Discrete Mathematics* 263 (1 - 2) (2003) 81 – 91.
6. Carlach J.-C., Otmani A., A systematic construction of self-dual codes , *IEEE Transactions on Information Theory* 49 (11) (2003), 3005 – 3009.

T5. **P. Kaski, P. Östergård, S. Topalova, R. Zlatarski**, Steiner Triple Systems of Order 19 and 21 with Subsystems of Order 7, *Discrete Mathematics* **308** (2008), 2732 – 2741.

7. Eslami Z. ,Classification of designs with nontrivial automorphism groups, *Journal of Combinatorial Designs* 14 (6) (2006) , 479 - 489.
8. Armanious, M. H., On Steiner Quasigroups of Cardinality 21, *Ars Combinatoria* 104 (2012), 417 – 430.
9. Ihringer, F., Meagher, K., Mikls-Manickam-Singhi conjectures on partial geometries, *Des. Codes Cryptogr.* 86 (6) (2018), 1311–1327.

T7. **Z. Mateva, S. Topalova**, Line spreads of $PG(5, 2)$, *Journal of Combinatorial Designs* **17** (1) (2009), 90 – 102.

10. Draisma J., Shaw R., Some noteworthy alternating trilinear forms, *Journal of Geometry* 105 (1) (2014), 167 – 176.
11. Kiermaier, M., Kurz, S. and Wassermann, A., The order of the automorphism group of a binary q-analog of the Fano plane is at most two, *Des. Codes Cryptogr.* 86 (2) (2018), 239 – 250.
12. Honold, T., Kiermaier, M. and Kurz, S. , Classification of large partial plane spreads in PG(6,2) and related combinatorial objects, Journal of Geometry (2019) 110: 5.

T8. **S. Topalova, S. Zhelezova**, 2-spreads and transitive and orthogonal 2-parallelisms of PG(5,2), *Graphs and Combinatorics* **26** (5) (2010), 727 – 735.

13. Etzion T., Silberstein N., Codes and designs related to lifted MRD codes, *IEEE Transactions on Information Theory* 59 (2) (2013), 1004 – 1017.

14. Johnson N.L., Montinaro A., The transitive t-parallelisms of a finite projective space, *Advances in Geometry* 12 (3) (2012), 401 – 429.

15. A. Betten, The packings of $PG(3,3)$, *Design. Code. Cryptogr.* 79 (3) (2016), 583 – 595.

16. Etzion, T., Storme, L., Galois geometries and coding theory, *Des. Codes Cryptogr.* 78 (1) (2016), 311–350.

17. Huang, LP. and Lv, B., Cores and Independence Numbers of Grassmann Graphs, *Graphs and Combinatorics* 33 (6) (2017), 1607 – 1620.

T9. **T. Baicheva, S. Topalova**, Classification of optimal $(v, 4, 1)$ binary cyclically permutable constant weight codes and cyclic $S(2, 4, v)$ designs with $v \leq 76$, *Problems of Information Transmission* **47** (3) (2011), 224 – 231.

18. Fujiwara Y., Self-synchronizing pulse position modulation with error tolerance, *IEEE Trans. Inform. Theory* 59 (9) (2013), 5352-5362.

### Цитирания в реферирани и индексирани в Scopus списания без импакт фактор

T2. **S. Topalova**, Classification of Hadamard matrices of order 44 with automorphisms of order 7, *Discrete Mathematics* **260** (2003), 275 – 283.

19. Georgiou S., Koukouvinos C., Lappas E., Extremal doubly-even self-dual codes from Hadamard matrices, *Journal of Discrete Mathematical Sciences and Cryptography* 9(2) (2006), 331-339, SJR(2006)=0.176.

20. Georgiou S., Kotsireas I., Koukouvinos C., Inequivalent Hadamard matrices of order 2n constructed from Hadamard matrices of order n, *Journal of Combinatorial Mathematics and Combinatorial Computing* 63 (2007), 65-79.

T5. **P. Kaski, P. Östergård, S. Topalova, R. Zlatarski**, Steiner Triple Systems of Order 19 and 21 with Subsystems of Order 7, *Discrete Mathematics* **308** (2008), 2732 – 2741.

21. Li X., Xu Z., Chou W., Method of Construction and Enumeration Steiner Triple Systems with Order 19, *Key Engineering Materials* 474 – 476 (2011), 1205 – 1208, SJR(2011)=0.177.

22. Xu Z., Li X., Chou W., Two Isomorphic 9 Order Steiner Triple Systems Large Sets, *Applied Mechanics and Materials* 427 (2013), 1237–1240, SJR(2013)=0.132.

23. A. Drapal, T. S. Griggs, On cosets in Steiner loops, *Buletinul Academiei de Stiinte a Republicii Moldova. Matematica* 1(80) (2016), 118 – 124, SJR(2016)=0.218.

24. Li X., Xu Z., Chou W., A new method of constructing Steiner triple systems, *Chinese Control and Decision Conference* 2010, 3760 – 3764, SJR(2010)=0.11.

25. Li X., Xu Z., Chou W., A method of constructing Kirkman triple system of higher order, *Chinese Control and Decision Conference (CCDC)* 2010, 2318 – 2322, SJR(2010)=0.11.

26. Li X., Xu Z., Chou W., A new method of constructing Kirkman triple system, *Chinese Control and Decision Conference* 2011 , 4237 – 4242, SJR(2011)=0.12.

27. Li X., Xu Z., Chou W., Construction and enumeration of Steiner triple systems with order $s \times t$, *Chinese Control and Decision Conference (CCDC)* 2011, 4243 – 4247, SJR(2011)=0.12.

28. Xu Z., Li X., Chou W., New method of constructing triple system with order $s \times t$, 24th *Chinese Control and Decision Conference (CCDC)* 2012, 3757 – 3761, SJR(2012)=0.123.

29. Xu Z., Li X., Chou W., Decomposition of $K_{2n}$ into $n-1$ Hamiltonian cycles and a perfect matching $M_i$, 24th *Chinese Control and Decision Conference (CCDC)* 2012, 3762 – 3767, SJR(2012)=0.123.

T8. **S. Topalova, S. Zhelezova**, 2-spreads and transitive and orthogonal 2-parallelisms of PG(5,2), *Graphs and Combinatorics* **26** (5) (2010), 727 – 735.

30. Silberstein N., Etzion T., Codes and designs related to lifted MRD codes , *IEEE International Symposium on Information Theory Proceedings (ISIT)* 2011, St. Petersburg, 2288 - 2292, SJR(2011)=0.673.

**Цитирания в монографии и колективни томове с научно рецензиране:**

T1. **S. Topalova**, Symmetric 2-$(69, 17, 4)$ designs with automorphisms of order 13, *Journal of Statistical Planning and Inference* **95** (2001), 335 – 339.

31. Kaski P., Ostergard P., *Classification Algorithms for Codes and Designs*, **Springer Verlag**, Berlin Heidelberg, 2006 ( p.396).

32. Gibbons P., Ostergard P., Computational Methods in Design Theory, in: C.Colbourn and J.Dinitz (Eds.), *The CRC Handbook of Combinatorial Designs*, **CRC Press**, Boca Raton, FL., 2007, 755-783.

33. Mathon R., Rosa A., 2-(v, k, $\lambda$ ) designs of small order, in: C.Colbourn and J.Dinitz (Eds.), *The CRC Handbook of Combinatorial Designs*, **CRC Press**, Boca Raton, FL., 2007, 25-58.

T2. **S. Topalova**, Classification of Hadamard matrices of order 44 with automorphisms of order 7, *Discrete Mathematics* **260** (2003), 275 – 283.

34. Georgiou S., Koukouvinos C., Seberry J., Hadamard matrices, orthogonal designs and construction algorithms, in *Designs 2002: Further Computational and Constructive Design Theory* (Ed. W.D. Wallis), **Kluwer Academic Publishers**, Norwell, Massachusetts, (2003), 133–205.

35. Kaski P., Ostergard P., *Classification Algorithms for Codes and Designs*, **Springer Verlag**, Berlin Heidelberg, 2006 (p.396).

36. Gibbons P., Ostergard P., Computational Methods in Design Theory, in: C.Colbourn and J.Dinitz (Eds.), *The CRC Handbook of Combinatorial Designs*, **CRC Press**, Boca Raton, FL., 2007, 755–783.

T7. **Z. Mateva, S. Topalova**, Line spreads of $PG(5, 2)$, *Journal of Combinatorial Designs* **17** (1) (2009), 90 – 102.

37. Honold T., Kiermaier M., Kurz S., Partial Spreads and Vector Space Partitions. In: Greferath M., Pavčević M., Silberstein N., Vázquez-Castro M. (eds) *Network Coding and Subspace Designs. Signals and Communication Technology*, **Springer**, Cham, 2018, 131 – 170.

38. V. Tonchev, Finite geometry designs, codes and Hamada's conjecture, In: *Information Security, Coding Theory and Related Combinatorics*, D. Crnkovič and V. Tonchev (eds), **IOS Press**, 2011, 437 – 448.

### Цитирания в реферирани в други известни бази рецензирани списания без импакт фактор

T1. **S. Topalova**, Symmetric 2-$(69, 17, 4)$ designs with automorphisms of order 13, *Journal of Statistical Planning and Inference* **95** (2001), 335 – 339.

39. Zhelezova S., A method for classification of doubly resolvable designs and its application, *Serdica Journal of Computing* 5(3) (2011), 273-308 (**zbMATH**).

T2. **S. Topalova**, Classification of Hadamard matrices of order 44 with automorphisms of order 7, *Discrete Mathematics* **260** (2003), 275 – 283.

40. Zhelezova S., A method for classification of doubly resolvable designs and its application, *Serdica Journal of Computing* 5(3) (2011), 273-308 (**zbMATH**).

T3. **S. Topalova**, STS(21) with Automorphisms of order 3 with 3 fixed points and 7 fixed blocks, *Mathematica Balkanica* **18** (2004), 215 – 220.

41. Zhelezova S., A method for classification of doubly resolvable designs and its application, *Serdica Journal of Computing* 5(3) (2011), 273-308 (**zbMATH**).

T4. **V. Fack, S. Topalova, J. Winne, R. Zlatarski**, Enumeration of the doubles of the projective plane of order 4, *Discrete Mathematics* **306** (2006) 2141 – 2151.

42. Zhelezova S., A method for classification of doubly resolvable designs and its application, *Serdica Journal of Computing* 5(3) (2011), 273-308 (**zbMATH**).

T5. **P. Kaski, P. Östergård, S. Topalova, R. Zlatarski**, Steiner Triple Systems of Order 19 and 21 with Subsystems of Order 7, *Discrete Mathematics* **308** (2008), 2732 – 2741.

43. Armanious M.H., On the classes of Steiner loops of small orders, *Notes on Number Theory and Discrete Mathematics* 17(2) (2011), 52-68 (**zbMATH**).

44. Zhelezova S., A method for classification of doubly resolvable designs and its application, *Serdica Journal of Computing* 5(3) (2011), 273-308 (**zbMATH**).

T7. **Z. Mateva, S. Topalova**, Line spreads of $PG(5, 2)$, *Journal of Combinatorial Designs* **17** (1) (2009), 90 – 102.

45. Zhelezova S., A method for classification of doubly resolvable designs and its application, *Serdica Journal of Computing* 5(3) (2011), 273-308 (**zbMATH**).

### Цитирания в други рецензирани списания

T7. **Z. Mateva, S. Topalova**, Line spreads of $PG(5, 2)$, *Journal of Combinatorial Designs* **17** (1) (2009), 90 – 102.

46. Zhelezova S., Cyclic parallelisms of PG(5,2), *Mathematica Balkanica* 24 (1-2) (2010), 141-146.

47. P Ranjan, N Spencer, A Unified Approach to Factorial Designs with Randomization Restrictions, *Calcutta Statistical Association Bulletin* 65 (1–4) (2013).

**Цитирания в сборници от международни конференции:**

T4. **V. Fack, S. Topalova, J. Winne, R. Zlatarski**, Enumeration of the doubles of the projective plane of order 4, *Discrete Mathematics* **306** (2006) 2141 – 2151.

48. Mateva Z., Doubles of Hadamard 2-(15,7,3) designs, Eleventh International Workshop on Algebraic and Combinatorial Coding Theory, June 16-22, 2008, Pamporovo, Bulgaria pp. 210-214.

T5. **P. Kaski, P. Östergård, S. Topalova, R. Zlatarski**, Steiner Triple Systems of Order 19 and 21 with Subsystems of Order 7, *Discrete Mathematics* **308** (2008), 2732 – 2741.

49. Zhelezova S., PCIMs in constructing doubly resolvable designs, Proceedings of the Fifth International Workshop on Optimal Codes and related topics, White Lagoon, Bulgaria (2007), 260 - 266.

50. Zhaodi Xu , Xiaoyi Li, Wanxi Chou, Construction and counting of t+1 Steiner triple, International Conference on Applied Science and Engineering Innovation (ASEI 2015), 2015.

51. Wei Tian , Xiaoyi Li, Wanxi Chou, Different configurations of 25-order structure of Steiner triple systems of counting, International Conference on Applied Science and Engineering Innovation (ASEI 2015), 2015

**Цитирания в дисертации в чужбина**

T1. **S. Topalova**, Symmetric 2-$(69, 17, 4)$ designs with automorphisms of order 13, *Journal of Statistical Planning and Inference* **95** (2001), 335 – 339.

52. Kaski P., Algorithms for classification of combinatorial objects, Dissertation for the degree of Doctor of Science in Technology, Department of Computer Science and Engineering, Helsinki University of Technology, 2005.

T2. **S. Topalova**, Classification of Hadamard matrices of order 44 with automorphisms of order 7, *Discrete Mathematics* **260** (2003), 275 – 283.

53. Kaski P., Algorithms for classification of combinatorial objects, Dissertation for the degree of Doctor of Science in Technology, Department of Computer Science and Engineering, Helsinki University of Technology, 2005.

T2c. **Topalova S.**, Hadamard matrices of order 44 with automorphisms of order 7, Proceedings of the Seventh International Workshop on Algebraic and Combinatorial Coding Theory, Bansko, Bulgaria (2000), 305-310.

54. Otmani A., Codes Cortex et construction de codes auto-duaux optimaux, THESE pour obtenir le grade de docteur de l'universite de Limoges, discipline : mathematiques et ses applications, Universite de Limoges U.F.R. de sciences et techniques, 2002.

T5. **P. Kaski, P. Östergård, S. Topalova, R. Zlatarski**, Steiner Triple Systems of Order 19 and 21 with Subsystems of Order 7, *Discrete Mathematics* **308** (2008), 2732 – 2741.

55. Gugisch R., Konstruktion von Isomorphieklassen orientierter Matroide, Dissertation zur Erlangung des Grades eines Doktors der Naturwissenschaften, Universitat Bayreuth, 2005.

56. Degraer J., Isomorph-free exhaustive generation algorithms for association schemes, PhD thesis, Ghent University, Belgium, 2007.

57. Winne J., Software tools for combinatorial algorithms, Ph.D. thesis, Ghent University, 2007.

T6. **Z. Mateva, S. Topalova**, Hadamard 2-(63,31,15) designs invariant under the dihedral group of order 10, *Discrete Mathematics* **309** (6) (2009), 1347 – 1356.

58. Clark D. C., Applications of Finite Geometries to Designs and Codes, Ph. D. thesis, Michigan Technological University, 2011.

T7. **Z. Mateva, S. Topalova**, Line spreads of $PG(5,2)$, *Journal of Combinatorial Designs* **17** (1) (2009), 90 – 102.

59. Clark D. C., Applications of Finite Geometries to Designs and Codes, Ph. D. thesis, Michigan Technological University, 2011.

T8. **S. Topalova, S. Zhelezova**, 2-spreads and transitive and orthogonal 2-parallelisms of PG(5,2), *Graphs and Combinatorics* **26** (5) (2010), 727 – 735.

60. Silberstein N., Coding Theory and Projective Spaces, Ph. D. thesis, Israel Institute of Technology, Haifa, 2011.

61. Chowdhury A. N., Shadows and Intersections, Ph. D. Thesis, University of California, San Diego, 2012.

### Цитирания в дисертации в България

T1. **S. Topalova**, Symmetric 2-$(69, 17, 4)$ designs with automorphisms of order 13, *Journal of Statistical Planning and Inference* **95** (2001), 335 – 339.

62. Матева З., Конфигурации с адамарови дизайни, Дисертация за присъждане на образователната и научна степен доктор по научна специалност информатика, ТУ Варна, 2011.

T2. **S. Topalova**, Classification of Hadamard matrices of order 44 with automorphisms of order 7, *Discrete Mathematics* **260** (2003), 275 – 283.

63. Буюклиев И., Алгоритмични подходи за изследване на линейни кодове, Дисертация за присъждане научната степен доктор на математическите науки по научна специалност информатика, ИМИ, София, 2008.

64. Матева З., Конфигурации с адамарови дизайни, Дисертация за присъждане на образователната и научна степен доктор по научна специалност информатика, ТУ Варна, 2011.

T3. **S. Topalova**, STS(21) with Automorphisms of order 3 with 3 fixed points and 7 fixed blocks, *Mathematica Balkanica* **18** (2004), 215 – 220.

65. Железова С., Изследване и класификация на двойно разрешими дизайни, Дисертация за присъждане на образователната и научна степен доктор по научна специалност информатика, ИМИ, София, 2009.

66. Матева З., Конфигурации с адамарови дизайни, Дисертация за присъждане на образователната и научна степен доктор по научна специалност информатика, ТУ Варна, 2011.

T4. **V. Fack, S. Topalova, J. Winne, R. Zlatarski**, Enumeration of the doubles of the projective plane of order 4, *Discrete Mathematics* **306** (2006) 2141 – 2151.

67. Матева З., Конфигурации с адамарови дизайни, Дисертация за присъждане на образователната и научна степен доктор по научна специалност информатика, ТУ Варна, 2011.

T5. **P. Kaski, P. Östergård, S. Topalova, R. Zlatarski**, Steiner Triple Systems of Order 19 and 21 with Subsystems of Order 7, *Discrete Mathematics* **308** (2008), 2732 – 2741.

68. Железова С., Изследване и класификация на двойно разрешими дизайни, Дисертация за присъждане на образователната и научна степен доктор по научна специалност информатика, ИМИ, София, 2009.

69. Матева З., Конфигурации с адамарови дизайни, Дисертация за присъждане на образователната и научна степен доктор по научна специалност информатика, ТУ Варна, 2011.

T6. **Z. Mateva, S. Topalova**, Hadamard 2-(63,31,15) designs invariant under the dihedral group of order 10, *Discrete Mathematics* **309** (6) (2009), 1347 – 1356.

70. Буюклиев И., Алгоритмични подходи за изследване на линейни кодове, Дисертация за присъждане научната степен доктор на математическите науки по научна специалност информатика, ИМИ, София, 2008.

T7. **Z. Mateva, S. Topalova**, Line spreads of $PG(5,2)$, *Journal of Combinatorial Designs* **17** (1) (2009), 90 – 102.

71. Железова С., Изследване и класификация на двойно разрешими дизайни, Дисертация за присъждане на образователната и научна степен доктор по научна специалност информатика, ИМИ, София, 2009.

## Означения и съкращения

| | | |
|---|---|---|
| OOC | $\longrightarrow$ | оптичен ортогонален код, оптични ортогонални кодове<br>optical orthogonal code(s) |
| CAC | $\longrightarrow$ | антиконфликтен код, антиконфликтни кодове<br>conflict avoiding code(s) |
| SCAC | $\longrightarrow$ | силен антиконфликтен код, силни антиконфликтни кодове<br>strongly conflict avoiding code(s) |
| CPCW | $\longrightarrow$ | циклично-пермутационен константно-тегловен (код),<br>циклично-пермутационни константно-тегловни (кодове)<br>cyclically permutable constant weight (code(s)) |
| $STS(n)$ | $\longrightarrow$ | Щайнерова система от тройки от ред $n$<br>Steiner triple system of order $n$ |