АВТОРСКА СПРАВКА

за научните приноси
на

Цонка Стефанова Байчева

по конкурс за професор по научната специалност 01.01.12 Информатика (компютърни подходи в изследването на шумозащитни кодове)

Общият брой на научните публикации е 63. За конкурса са представени 26 научни публикации. Шест от тях ([1]-[6]) са представени при получаване на академичната длъжност "доцент". Останалите 20 работи са публикувани след това. 14 от статиите са в списания с имапкт фактор с общ IF=18,772.

От гледна точка на практическото приложение на шумозащитните кодове е необходимо не само да се покаже съществуването на код с определени характеристики, но и да се конструира самия код. Интерес представлява и пълната класификация на кодове със зададени параметри и определянето на техните основни шумозащитни характеристики, което позволява избора на подходящ за всяка конкретна ситуация код. Особено актуални и интересни са тези резултати в последните години, когато все по-често се правят софтуерни реализации на кодирането и декодирането.

В огромната част от случаите използването на чисто математически подходи за решаването на тези задачи не може да доведе до търсения резултат. Една възможност за решаването на такъв тип задачи е използването на компютър. Директната им атака с компютър обаче позволява да се получат решения само за ограничени по обем входни данни. Причината за това е във факта, че експоненциално нарастващата им трудност бързо стопява предимството на компютъра да работи многократно по-бързо от човека и да помни големи количества информация. За по-големи входни данни се налага за всяка отделна задача да се провеждат предварителни математически изследвания, които макар да не решават директно въпроса, силно съкращават възможностите, които трябва да се изследват с компютър. Друга възможност е разработването на все по-бързи и икономични на памет алгоритми, което изисква отличното познаване и умелото използване на свойствата на изследваните математически обекти и възможностите на компютъра. Така се оформя една все по-успешна в последните две десетилетия хибридна математико-компютърна стратегия, с помощта на която бяха решени някои трудни алгоритмично разрешими задачи в различни области на математиката.

Представените за конкурса статии използват този подход за изследването на класове шумозащитни кодове имащи добри характеристики по отношение на откриване и коригиране на грешки. В работите [10] и [22] е използвана написаната от Илия Буюклиев програма за класификация на кодове Q-EXTENSION. Всички останали компютърни резултати са

получени с написан от кандидата софтуер основно на C++ и Maple. Те са в няколко направления.

1. Радиус на покритие на класове линейни кодове (статии 1, 2, 6, 8, 10, 11, 22) В тези статии са разгледани конкретни задачи за намиране на радиусите на покритие на някои класове линейни кодове. Радиусът на покритие е основен параметър на кода. Ако кодът се използва за поправяне на грешки и се декодира винаги до най-близката кодова дума, радиусът му на покритие е максималното тегло на коригируем вектор-грешка, а ако се използва за компресиране на данни е мярка за максималното им изкривяване. Радиусът на покритие ни дава мярка и за това дали един код е максимален (към него да не могат да се прибавят още кодови думи без да се намали минималното му разстояние), което е възможно само, ако радиусът на покритие е строго по-малък от минималното разстояние на кода.

В статията си от 1986 година Sloane представя един нов подход, който улеснява пресмятането на радиуса на покритие на двоичен линеен код. Той се основава на определянето на въведения в същата статия нормализиран радиус на покритие на код. За кодове с фиксирана размерност са дадени горни и долни граници за него, а също така са дадени формули за определяне радиусите на покритие на двоичните кодове с размерности до 4 и произволна дължина. По-късно Великова е използвала същия подход, за да определи радиусите на покритие на кодовете с размерност 2 над GF(4).

В [1] са разгледани всички троични линейни кодове с размерности 2 и 3. Въведено е понятието нормализиран радиус на покритие за троичен линеен код и са изведени долни и горни граници за него. Направена е класификация на всички троични кодове с размерности 2 и 3 според кратностите на различните стълбове, съставящи пораждащите им матрици. С аналитични методи, които използват изведените горни и долни граници, са определени стойностите на нормализирания радиус на покрите за част от класифицираните кодове, а за останалите са използвани компютърни пресмятания. Изведени са формули за определянето на радиуса на покритие на троичен линеен код с произволна дължина и размерност 2 или 3 и на троичните кодове с ко-размерности до 8 и до фиксирани дължини. По-късно в [22] са изследвани троичните проективни кодове с размерности 4 и 5. С програмата Q-EXTENSION са класифицирани всички проективни троични кодове с размерност 4 и тези с размерност 5 с дължини до 15. Определени са основни характеристики на тези кодове като групи от автоморфизми, теглови спектри и радиуси на покритие.

Интересен изследователски проблем, свързан с радиуса на покритие на код, е определянето на стойностите на функцията $t_q[n,k]$ - минималния радиус на покритие на линеен q-ичен $[n,k]$ код за фиксирани дължини и размерности. Най-интензивно е изследвана функцията $t_2[n,k]$ и получените точни стойности или горни и долни граници за нея са представени в Таблица 7.1 от книгата „Покриващи кодове“, излязла през 1997 година. Чрез предложения в [8] метод за конструиране на кодове с радиус на покритие равен на долната

граница за $t_2[n,k]$ като се използват особености на пораждащите им матрици са решени първите шест отворени случая от Таблица 7.1.

В работата на Graham и Sloane от 1985 година са определени стойностите на функцията $t_2[n,k]$ за кодове с размерности до 5. Изведени са и горни граници за $t_2[n,k]$ за кодове с размерности 6 и 7. В [11] са определени минималните радиуси на покритие на всички двоични линейни кодове с размерност 6 като е показано, че той съвпада с границата от работата на Graham и Sloane. От решаващо значение за получаване на резултата беше разработеният евристичен алгоритъм, който бързо определя долна граница на радиуса на покритие на линеен код. Изведена е и горна граница за функцията $t_2[n,k]$ за кодове с размерности 8 и 9. Класифицирани са всички двоични линейни кодове с дължини до 15 и размерности до 5 и е пресметнат радиусът им на покритие. На базата на тази класификация е представена конструкция на кодове с минимален радиус на покритие, произволна дължина и размерност до 5. Показано е, че съществува единствен $[19,6,7]$ код с минимален радиус на покритие. Като е използван този резултат, известния от преди факт, че $[14,6,5]$ кода с минимален радиус на покритие е единствен, и че двата кода са нормализирани и всичките им координати са приемливи е дадена конструкция на кодове с минимален радиус на покритие, произволна дължина и размерност 6.

Специален случай в теорията на кодирането са съвършените кодове. Тези кодове могат да коригират всички грешки, които откриват, т.е. радиусът им на покритие и радиусът на сферичната им опаковка са еднакви. Още през 1973 година е напълно решен въпросът за какви параметри съществуват такива кодове, но за съжаление те са много малко. Едно естествено продължение на изследванията е да се разглеждат квази-съвършени кодове, т.е. кодове, за които радиусът на сферичната опаковка и радиусът на покритие се различават с 1. Квази-съвършените кодове са много широко изследвани, но всички известни примери на такива кодове съдържаха само по един код за съответните дължина и размерност. В [10] е даден отговор на въпросът колко рестриктивно е свойството на един код да е квази-съвършен, т.е. има ли нееквивалентни квази-съвършени кодове с фиксирани дължина и размерност като са класифицирани всички двоични и троични квази-съвършени кодове с размерности съответно до 9 и 6. Получени са и частични резултати за двоичните квази-съвършени кодове с размерности до 14 и за троичните квази-съвършени кодове с размерности до 13. Предложена е модификация на един от основните методи за определяне на радиуса на покритие на линеен код, с която се определят само тези кодове, които имат предварително зададен радиус на покритие. Получените в това изследване резултати показват, че за всяка размерност има само по няколко възможни дължини, за които могат да съществуват квази-съвършени кодове. За някои от параметрите бяха намерени стотици и хиляди кодове, което показва че това не е толкова рестриктивно свойство. А следователно и класификацията на всички възможни параметри на квази-съвършени кодове би била многократно по-тежка задача в сравнение с тази за съвършените кодове.

До нашата работа имаше само няколко известни примера на квази-съвършени кодове с минимално разстояние по-голямо 5. В нея се дават примери на още такива кодове и по този начин се отговаря на отворения проблем, поставен в статията на Etzion и Mounits, където се предлага да се намерят нови или да се докаже несъществуването на други квази-съвършени кодове с d > 5.

В обзорната си статия от 1985 година, Cohen, Karpovski, Mattson, Jr. и Shatz поставят като отворен проблем определянето на радиусите на покритие на някои класове линейни кодове. Като отговор Downie и Sloane определят радиусите на покритие на двоичните циклични кодове с дължини до 31. Великова и Манев продължават изследването за двоичните циклични кодове с дължини 33, 35 и 39. По-късно Dougherty и Janwa пресмятат радиусите на покритие на всички двоични циклични кодове с дължини, ненадминаващи 64, и размерности до 28. Резултатите и в трите работи са получени с използването на компютър. В [2] са определени радиусите на покритие на всички троични циклични кодове с дължини до 25. Направена е характеризация на тези кодове, които са 165 на брой. Определени са горни и долни граници за радиусите им на покритие. За 41 от разглежданите кодове стойностите за една от горните и една от долните граници съвпадат, т.е. радиусите им на покритие се определят точно. За част от останалите (12 кода) са използвани различни математически съображения, за да бъдат определени радиусите им на покритие, а за другите - компютърни пресмятания.

[6] е продължение на изследванията от [2]. Там са разгледани троичните негациклични кодове с четни дължини до 26. Фокусът е само върху негацикличните кодове с четни дължини, защото тези с нечетни дължини са еквивалентни на циклични кодове. Направена е класификация на всички такива кодове и са пресметнати минималните им разстояния, спектрите им и радиусите им на покритие. Като резултат от това са уточнени 7 стойности на функцията $t_3[n,k]$, а за други три случая са подобрени горните граници.

Накрая ще отбележим, че получените в горните работи резултати са полезни и в още едно отношение. То е свързано с факта, че като знаем радиуса на покритие на някои кодове от тях можем да получим нови като използваме различни конструкции като външна директна сума, удължаване и скъсяване на код, слепване на кодове. Новите кодове ще бъдат с точно известен радиус на покритие или ще знаем граница за него.

2. Поведение на шумозащитни кодове при откриване и коригиране на грешки (статии 7, 16, 17, 18, 25)

Когато искаме да намерим $[n,k]$ код за откриване и коригиране на грешки за някое приложение, най-добрият избор би бил код, чиято стойност на вероятността за неоткрита грешка е минимална (оптимален код), ако знаем вероятността за грешка на канала ε. За съжаление, дори и да знаем стойността на ε, не е известен общ метод за конструиране на оптимални кодове различен от пълното претърсване. Следователно, полезно е да имаме критерии, които да ни помагат да определим дали един код е добър за контрол на грешки.

Кодът C се нарича t-подходящ (или само подходящ когато t = 0 и той се използва само за откриване на грешки), ако вероятността му за неоткрита грешка е монотонна функция в интервала $\varepsilon \in [0, \frac{q-1}{q}]$. Поверката дали един код е подходящ за контрол на грешки е трудна задача и определянето на класове кодове, които са подходящи, представлява значителен интерес, както от теоретична гледна точка, така и от гледна точка на приложенията. Освен това, ако имаме подходящи кодове, от тях могат да се конструират нови, също подходящи кодове.

В [7] е изследван троичният $[13, 7, 5]$ квадратично-остатъчен код, показано е, че е подходящ за откриване и коригиране на грешки и са представени два ефективни алгоритъма за декодирането му.

Дефинираните над $GF(9)$ кодове на Reed-Solomon RS(10;9) и Glynn Gl(10;9) са разгледани в [16]. Тези кодове са нееквивалентни $[10, 5, 6]$ MDS кодове. Обикновено оптималните според някакъв критерий кодове не могат да се разширяват. Разглежданите кодове, обаче, се разширяват до $[12, 6, 9]$ такива и всички разширени кодове са almost MDS, но не и near MDS. Пресметнати са тегловите разпределения на лидерите на съседни класове на $[10, 5, 6]$ кодовете и те съвпадат напълно. А тъй като стойността на радиусите им на покритие е 4, то вероятностите им за неоткрита грешка след коригиране на до 4 грешки са еднакви.

Интересен въпрос за линейни кодове с еднакви основни параметри (дължина, размерност, минимално разстояние и радиус на покритие) е доколко прецизно тези параметри определят поведението им при отриване и коригиране на грешки. За да се демонстрира как различините параметри на един линеен код влияят върху поведението му при контрол на грешки, в [17] са използвани три класа двоични линейни кодове с параметри $[15, 3, 7]$, $[15, 3, 8]$ и $[16, 3, 8]$ и на тяхна база са показани различни аспекти от оценката на поведението им. Накрая е даден отговор и на отворен изследователски проблем 5.1 поставен в книгата на MacWilliams и Sloane „The Theory of Error-Correcting codes“ (Глава 5, стр. 132) като е показано, че спектърът на лидерите на съседни класове на един линеен код не определя еднозначно спектърът на лидерите на съседни класове на дуалния му. Следователно, за тези спектри немогат да се изведат тъждества подобни на тъждествата на MacWilliams за спектрите на кодовете и на дуалните им.

Като са използвани класификациите на двоични циклични кодове, на двоични кодове с максимално минимално разстояние, на двоични CRC кодове и на троични циклични и негациклични кодове в [18] и [25] са пресметнати тегловите им разпределения, разпределенията на лидерите на съседните им класове и тегловите разпределения на самите съседни класове. С помощта на програма, написана на Maple, са определени всички кодове, които не са t-подходящи. Пакетът Maple беше избран тъй като проверката дали кодът е t-подходящ става за линейно време, т.е. не ни е нужно бързодействие на програмата, а системата предлага възможности за работа с дроби. Така сравненията на стойностите на вероятността за неоткрита грешка на всяка стъпка е максимално прецизно тъй като сравняваме самите

дроби, а не тяхно представяне с мантиса и порядък както е в C++, например.

Много често в практическите приложения се интересуваме от използването на кода в някакъв подинтервал на интервала от вероятности за грешка $\varepsilon \in [0, \frac{q-1}{q}]$. Резултатите, получени в [18] и [25], позволяват да направим заключения и за това дали кодът е подходящ в някой подинтервал на интервала $[0, \frac{q-1}{q}]$. Тази проверка се прави лесно при наличието на пресметнатите в двете работи данни и дава възможност кодове, които не са подходящи в целия интервал $[0, \frac{q-1}{q}]$ да бъдат използвани в подинтервалите, в които са подходящи.

3. Скъсени циклични кодове (статии 3, 5, 9, 19, 23, 24)

Комуникационните системи добавят в края на информационните последователности допълнителни битове, които осигуряват коректното предаване на данните. Много често за това се използват скъсени двоични циклични кодове, наречени CRC кодове. Практика при избора на CRC код за някое приложение е да се избере стандартизиран такъв, като се счита, че той ще е достатъчно добър. Оказва се, обаче, че това в голяма част от случаите не е така. Например, някои от стандартизираните 16-битови CRC кодове имат поведение при контрол на грешки, което е по-лошо от това на други нестандартизирани 16-битови CRC кодове. Една от причините е, че в първите години са били използвани изключително хардуерни имплементации на кодирането и декодирането и използването на пораждащ полином с по-малко ненулеви коефициенти е водело до по-евтина имплементация. Друга причина е, че липсват достатъчно данни, позволяващи да се оцени поведението на стандартизираните кодове и да се сравни с това на други, нестандартизирани такива. В [19] и [23] на базата на някои от най-често използваните стандартизирани кодове показваме, че подходът да се избира стандартизиран код или такъв с неразложим пораждащ полином или пораждащ полином, имащ множител (x+1) (т.е. код, който открива всички грешки с нечетно тегло) в много случаи е неудачен. Решения на тези проблеми са предложени в следващите работи, в които са изследвани CRC кодове с до 16 проверочни бита.

При стандартизираната технология за предаване на цифрови данни ATM (Asynchronous Transfer Mode) с 8-битов CRC код се защитават хедърите, които носят информацията необходима за насочване на данните през мрежата. В [3] са разгледани всички двоични полиноми от 8 степен, които са подходящи да бъдат използвани като пораждащи полиноми на 8-битови CRC кодове. За всички разглеждани кодове са определени характеристиките им: монотонност на функцията на вероятността за неоткрита грешка, добър, подходящ, минимално разстояние и радиус на покритие и са предтавени в таблици, така че всеки два кода могат да бъдат лесно сравнявани директно. Пресметнати са също и радиусите на покритие и тегловите разпределения на лидерите на съседни класове на всички изследвани кодове. Сравнени са стойностите на вероятността за неоткрита грешка за всички изследвани кодове на дължина 40 (дължината, на която работи ATM стандарта). Оказва се, че има два кода, които имат по-добро поведение от ATM стандарта на дължина 40. При стойности на вероятността за неоткрита грешка между 0.019331 и 0.2 поведението на първия код е с до

18% по-добро от това на ATM кода. Поведението на втория код е с до 5% по-добро от това на ATM кода като има същото минимално разстояние $d = 4$ и радиус на покритие $R = 3$ като него.

В [5], подобно на [3], се разглеждат всички полиноми от 16-та степен, които са подходящи да бъдат използвани като пораждащи полиноми на CRC кодове. Изследват се породените от тях кодове с дължини до 1024 и се сравняват със съществуващи 16-битови стандартизирани CRC кодове. CRC кодовете с 16 проверочни символа са едни от най-често използваните в практиката кодове и преди нашата работа имаше доста изследвания върху тях. Предимството на подходът, който използваме ние, е че изследваме всички кодове с дължини от 18 до 1024 и така определяме най-добрите по отношение на вероятността за неоткрита грешка. Освен това, имаме информация за реда на полинома, за минималното разстояние на породените от него кодове и дали те удовлетворяват достатъчните условия за добър и подходящ за откриване на грешки код. Отново, както и в [3], предлагаме кодове, които са по-добри от стандартизираните до момента. От лична кореспонденция с разработчици на комуникационни системи знаем за многобройни приложения на предложените в тази работа кодове в практиката. Особено често те се използват от компании, правещи разработки за железопътния и градския транспорт.

Направените в предишните две работи изследвания, както и тези на други автори, безспорно показват, че някои приложения използват CRC кодове, които имат поведение много по-лошо от най-добрите известни такива кодове. Всички тези изследвания, обаче, предлагат най-добри полиноми за фиксирани дължини и вероятности за грешка на канала, но не дават всички необходими данни, които да позволят на разработчиците на комуникационни системи сами да правят сравнение между полиномите и да избират най-подходящия за тяхното конкретно приложение. За да предоставим цялата нужна информация за сравняване на поведението при контрол на грешки на CRC кодове с до 10 проверочни символа, в [9] и [24] ние изследваме всички полиноми до 10 степен, които са подходящи да бъдат използвани като пораждащи полиноми на такива кодове. Първо правим списък на всички такива полиноми като изключваме от него реципрочните, тъй като пораждат еквивалентни кодове и определяме реда на тези полиноми. След това пресмятаме всички необходими данни, които позволяват сравняването на поведението при откриване и коригиране на грешки на тези кодове за линейно време. За да избегнем сравняването на всички полиноми от зададена степен при избора на най-подходящия, ние предлагаме лесна процедура от 4 стъпки, която дава възможност да се сравняват само няколко от най-добрите кодове.

Много от комуникационните протоколи не налагат ограничения за дължината на съобщенията, които се защитават, т.е. кодирането със CRC кода се прилага към съобщения с дължина многократно надвишаваща реда на пораждащия полином. Решението в този случай е да се използва повторение на оригиналния код. Кодиращите и декодиращите процедури за тези кодове са същите както и при CRC кодовете, но тяхното минимално разстояние

е 2. В [9] е изведена формула за пресмятане на кодовите думи с тегло 2 в такива кодове. Този резултат е много полезен при оценката на вероятността им за неоткрита грешка.

4. Някои характеристики на шумозащитни кодове, свързани с техните възможности за контрол на грешки (статии 4, 13, 15, 20, 21)

(n, M, d) код е множество от M двоични думи с дължина n и минимално разстояние по-голямо или равно на d. За фиксирани стойности на n и d, с $A(n, d)$ се означава максималното цяло число M, такова че съществува (n, M, d) код. До публикуването на [4] и [20] бяха известни всички стойности на $A(n, d)$ за $n \leq 15$ с изключение на $A(10, 3)$ и $A(11, 3)$. От конструкции, направени през 1965 година от Julin се знаеше, че $A(10, 3) \geq 72$ и $A(11, 3) \geq 144$. Определянето на точните стойности за тези функции е посочено като отворен изследователски проблем 2.4 в книгата на MacWilliams и Sloane "The Theory of Error-Correcting codes". В цитираните по-горе работи ние доказваме, че $A(10, 3) = 72$ и $A(11, 3) = 144$ и има 562, респективно 7398 неекивалентни такива кода. От съществено значение за ефективността на алгоритъма за класификация, който използваме, е наблюдението, че (n, M, d) код може да бъде скъсен и да се получи $(n - 1, M', d)$ код с $M' \geq M/2$. Така започваме с класифицирането на скъсените кодове и след това получаваме всички техни неекивалентни разширявания като отправната ни точка са двата $(4, 2, 3)$ кода. Друг важен момент е възможността да отхвърляме еквивалентните кодове, така че да не ги разглеждаме многократно в процеса на търсене. За целта преформулираме проблема за еквивалентност на кодове в проблем за еквивалентност на графи и използваме програмата за изоморфизъм на графи `nauty`, написана от Brendan McKay.

Свойството на един код да е нормализиран е въведено през 1985 от Graham и Sloane. Това свойство трябва да притежават кодовете, за да могат да участват в конструкцията смесена директна сума (amalgamated direct sum (ADS)), представена в същата работа. Целта на тази конструкция е да се получат кодове с колкото е възможно по-малък радиус на покритие в сравнение с други кодове със същата дължина и размерност. Интересен въпрос в този контекст е да се определи кои кодове са нормализирани. След работата на Graham и Sloane са направени доста изследвания с цел да се да се отговори на този въпрос. В [13] са обобщени резултатите за известните параметри, за които двоичните кодове са нормализирани, доказано е, че всички двоични кодове с дължини 16, 17 и 18 и ко-размерност 10 са нормализирани и е направена класификация на тези кодове. Показани са и примери как получените класификационни резултати могат да се използват за конструиране на кодове с минимален радиус на покритие.

Възможностите за коригиране на грешки на голяма част от блоковите кодове, разглеждани в литературата, се описват в контекста на коректното приемане на цялото съобщение. Съществуват, обаче, много приложения, при които някои от позициите на съобщението са по-важни от други. Линейни кодове, които защитават някои от позициите на съобщението срещу по-голям брой грешки отколкото други негови позиции се наричат линейни кодове

с неравномерна защита от грешки (linear unequal error protection (LUEP)). В [15] разглеждаме кодове с неравномерна защита на един от информационните символи като следваме дефиницията на Dunning и Robbins. Те въвеждат така наречения отделящ вектор, за да оценят възможностите за коригиране на грешки на линеен LUEP код. Основната задача е да се намери LUEP код с фиксирана размерност и отделящ вектор такъв, че дължината му да е минимална, а следователно скоростта му да е максимална. За целта използваме компютърно търсене с предложен от нас алгоритъм, с който са пресметнати отделящите вектори на троичните циклични и нега-циклични кодове с дължини до 26 и минимално разстояние поне 3.

Известно е, че за един линеен $[n, k, d]$ код всички грешки с тегло $t \leq (d-1)/2$ са коригируеми по единствен начин. Съществуват, обаче, грешки с тегло по-голямо от t, които са също коригируеми по единствен начин. Това са случите когато съседните класове с тегла по-големи от t имат единствен лидер. Естествени и важни въпроси в този случай са: кои са грешките, които са коригируеми по единствен начин; колко са те за предварително зададено тегло; какво е най-голямото тегло на грешка, която може да се коригира по единствен начин? В [21] даваме отговор на тези въпроси за двоичните циклични кодове с дължина до 31, двоичните кодове с максимално минимално разстояние с дължини до 33 и всички троични циклични и негациклични кодове с дължини до 22. Предимство на това изследване е, че не само определяме точните стойности на теглата на коригируемите по единствен начин грешки, но посочваме и броя на единствените лидери в съседните класове, т.е. колко от всички грешки с дадено тегло са коригируеми по единствен начин.

5. Оптимални оптични ортогонални кодове и свързаните с тях комбинаторни структури (статии 12, 14, 26)

Оптичните ортогонални кодове (OOC) въведени от Chung, Salehi и Wei са фамилии от двоични последователности с определени авто- и крос-корелационни свойства, които осигуряват много високи скорости на комуникация през оптични CDMA комуникационни мрежи. Използването на тези кодове позволява на голям брой потребители да предават данни асинхронно с необходимата скорост и надеждност. Те имат приложения също в мобилните радиосистеми, комуникациите с разпръснат спектър и прескачане на честота, радари и сонари, конструиране на последователности за M-активни-от-T потребители за канали с обратна връзка където са възможни конфликти.

Оптичните ортогонални кодове са свързани и с много други комбинаторни структури. (v, k, λ) оптичните ортогонални кодове са и двоични циклично-пермутационни (v, k, λ) константно тегловни кодове (CPCW). Всеки (v, k, λ) OOC е еквивалентен и на $(\lambda+1)-(v, k, \lambda)$ строго цикличен частичен дизайн, където $\lambda \geq 1$.

В статиите от този раздел за пръв път са направени пълни класификации на оптимални оптични ортогонални кодове и на свързаните с тях комбинаторни обекти. В [12] и [26] са класифицирани до изоморфизъм оптималните $(v, 4, 1)$ OOC (двоичните $(v, 4, 1)$ CPCW

кодове) с $v \leq 76$ и цикличните $2-(73,4,1)$ и $2-(76,4,1)$ дизайни. Тъй като съвършените $(v,4,1)$ CPCW кодове са еквивалентни на $(v,4,1)$ циклични разностни фамилии, то получаваме класификация и за $(73,4,1)$ цикличните разностни фамилии. В основата на алгоритъма за класификация е известната техника за търсене с връщане с тест за минималност на частичните решения, представена в книгата на Kaski и Östergård "Classification algorithms for codes and designs". За ефективното осъществяване на теста за минималност подреждаме всички възможни блокове според въведена от нас лексикографска наредба и действието на автоморфизмите на цикличната група от ред v.

В работата на Chu и Colbourn е представена таблица на оптималните $(v,4,2)$ OOC с $v \leq 44$ като авторите конструират по един код за всяко v с помощта на алгоритъм, основан на задачата за максимална клика в граф. В [14] ние правим пълна класификация на $(v,4,2,1)$ кодовете с $v \leq 75$ и $v \neq 71$. Първо, като използваме подход, подобен на този от [12] и [26], класифицираме с точност да мултипликативна еквивалентност всички $(v,4,2,1)$ OOC с $v \leq 75$ и $v \neq 71$. Ще отбележим, че в този случай класификацията до мултипликативна еквивалентност е по-сложна, тъй като кодовите думи са от различен тип и това трябва да се вземе под внимание при построяването на алгоритъма. Освен това се налагат допълнителни проверки, защото някои от колекциите от транслации на множества от кодови думи са мултипликативно нееквивалентни, но изоморфни.

Някои от резултатите в [12], [14] и [27] са получени с помощта на паралелни програми, изпълнени на българския суперкомпютър BlueGene/P. През 1993 година Karp и Zhang доказват, че търсенето с връщане се имплементира на паралелен компютър с ускорение близко до оптималното, което прави разработените от нас последователни програми подходящи за пренаписването им като паралелни. Реализираните паралелни версии използват търсене с връщане с глобален контрол и MPI за осъществяване на комуникацията между процесорите. Те бяха използвани само за някои по-специфични случаи, които не изискват много памет, но броя на възможните множества, които трябва да се тестват, е голям.

Накрая ще отбележим, че голяма част от получените класификационните резултати са достъпни в интернет, което ги прави удобни за ползване от всички, които се интересуват от тях. Останалите могат да се получат от автора при поискване.

Юни, 2012 Подпис: