

**ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

**KOMBİNATORİYAL TASARIMLAR VE KESİRLİ TEKRARLAMA
KODLARI**

Ceren KARTAL

MATEMATİK ANABİLİM DALI

**ANKARA
2017**

Her hakkı saklıdır

TEZ ONAYI

Ceren KARTAL tarafından hazırlanan “**Kombinatoryal Tasarımlar ve Kesirli Tekrarlama Kodları**” adlı tez çalışması 10.01.2017 tarihinde aşağıdaki jüri tarafından oy birliği ile Ankara Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı’nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman: Doç. Dr. Oktay ÖLMEZ

Jüri Üyeleri:

Başkan: Doç. Dr. Murat ŞAHİN
Ankara Üniversitesi Matematik Anabilim Dalı

Üye: Doç. Dr. Oktay ÖLMEZ
Ankara Üniversitesi Matematik Anabilim Dalı

Üye: Yrd. Doç. Dr. Hakan Özgür AYDOĞMUŞ
Ankara Sosyal Bilimler Üniversitesi Ekonomi Anabilim Dalı

Yukarıdaki sonucu onaylarım

Prof. Dr. Atila YETİŞEMİYEN

Enstitü Müdürü

ETİK

Ankara Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez içindeki bütün bilgilerin doğru ve tam olduğunu, bilgilerin üretilmesi aşamasında bilimsel etiğe uygun davranıldığı, yararlandığım bütün kaynakları atıf yaparak belirttiğimi beyan ederim.

10.01.2017

Ceren KARTAL

ÖZET

Yüksek Lisans Tezi

KOMBİNATORİYAL TASARIMLAR VE KESİRLİ TEKRARLAMA KODLARI
Ceren KARTAL

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Matematik Anabilim Dalı

Danışman: Doç. Dr. Oktay ÖLMEZ

Bu tez beş bölümden oluşmaktadır.

Bu tezde, dağıtılmış depolama sistemleri için paketler arasında kodlama yapmayan ve tam (tablo tabanlı) onarım işlemi sunan tekrarlama kodları ele alınmıştır. Bu kodlar, sistemdeki bir depolama düğümünde veri kaybı yaşanması halinde, erişim sağlanan diğer düğümlerden sadece gerekli paketleri indirerek erişim sağlanamayan verinin yeniden oluşturulmasını sağlamaktadır. İncelenen sistem mimarisinin dış kısmı bir maksimum uzaklıkla ayrılabilen (MDS) kod ve bu kodu takip eden bir iç kesirli tekrarlama (FR) kodundan oluşmaktadır. Tezdeki çalışmalarda, kesirli tekrarlama kodlarının kombinatoriyal tasarımlar ile olan ilişkileri ortaya konularak incelenmiştir. Kesirli tekrarlama kodlarının, projektif düzlemler, Steiner sistemleri ve çözülebilir kombinatoriyal tasarımlarından yararlanılarak elde edilen örneklerine yer verilmiştir.

Ocak 2017, .48 sayfa

Anahtar Kelimeler: Dağıtılmış depolama sistemleri, kesirli tekrarlama kodları, kombinatoriyal tasarımlar

ABSTRACT

Master Thesis

COMBINATORIAL DESIGNS AND FRACTIONAL REPETITION CODES

Ceren KARTAL

Ankara University
Graduate School of Natural and Applied Sciences
Department of Mathematics

Supervisor: Assoc. Prof. Dr. Oktay ÖLMEZ

This thesis consists of five chapters.

In this thesis, we consider the design of regenerating codes for distributed storage systems with an exact (table-based) repair process that is also uncoded. These codes designed to regenerate a failed node by simply downloading specified packets from the surviving nodes. The system design of our regenerating codes consist of an outer maximum distance separable (MDS) code followed by an inner fractional repetition (FR) code. Fractional repetition codes, specify the placement of symbols on storage nodes. In our work, we examine FR codes, by establishing their connection with combinatorial designs. We present constructions of FR codes based on projective planes, Steiner systems and resolvable combinatorial designs. We also examined the system properties of our constructions.

January 2017, 48 pages

Key Words: Distributed storage system, fractional repetition code, combinatorial designs

TEŞEKKÜR

Çalışmalarında bilgi ve tecrübeleriyle yön gösteren, hatalarımı hoşgörüle karşılayan ve çalışma tutkusuyla ilham veren değerli danışman hocam Doç. Dr. Oktay ÖLMEZ'e (Ankara Üniversitesi Matematik Anabilim Dalı) en içten saygı ve minnetlerimi sunarım.

Çalışmalarımıza katılarak özellikle de çok değerli sorularıyla farklı bakış açısı getirerek düşünce ufkumu geliştiren değerli hocam Doç. Dr. Murat ŞAHİN'e (Ankara Üniversitesi Matematik Anabilim Dalı) teşekkür ederim.

Beni cebir çalışmaya yönlendirerek severek çalıştığım bu konularla tanışmama vesile olan, değerli hocam Prof. Dr. Yusuf YAYLI'ya (Ankara Üniversitesi Matematik Anabilim Dalı) teşekkür ederim.

Kardeşten de öte bir bağla bağlı olduğum dostum Ebru ERBEK'e, beni her zorlukta motive ederek bana inancıyla güç veren dostum Elif BOLAT'a, akademik araştırmalara başladığım ilk zamanlardaki ev arkadaşlığıma ve dağınıklığıma sabreden dostum Kübra ÖZ'e ve lisansüstü eğitim arkadaşlarıma ayrıca teşekkür ederim.

Bu tez 114F246 No'lu TÜBİTAK-3001 "Başlangıç Ar-Ge Projeleri Destekleme Programı" tarafından desteklenmiştir. TÜBİTAK'a teşekkürlerimi sunarım.

Ceren KARTAL

Ankara, Ocak 2017

İÇİNDEKİLER

TEZ ONAY SAYFASI

ETİK.....	i
ÖZET.....	ii
ABSTRACT	iii
TEŞEKKÜR	iv
ŞEKİLLER DİZİNİ	vii
ÇİZELGELER DİZİNİ	viii
1. GİRİŞ	1
2. KOMBİNATORİYAL TASARIM TEORİSİ.....	4
2.1 Kombinatoriyal Tasarımlar	4
2.2 Oluşum Matrisi	7
2.3 Simetrik Tasarımlar.....	8
2.3.1 İkildüzlem tasarımlar	10
2.3.2 n. dereceden projektif düzlemler	12
2.3.3 Derecesi bir asal sayının kuvveti olan projektif düzlemlerin inşası	13
2.4 Steiner Üçlü Sistemler	16
2.5 Çözülebilir Tasarımlar	18
3. KODLAMA TEORİSİ.....	20
3.1 Temel Tanımlar	20
3.2 Lineer Kodlar	21
3.3 Üreteç Matrisi ve Kontrol Matrisi.....	21
3.4 Lineer Kodlama ile Hata Düzeltme	22
3.5 Maksimum Uzaklıkla Ayrılabilen Kodlar	27
3.6 MDS Kodların Bir Uygulaması	28
3.7 Silinim Kodları Kullanarak Onarım	28
4. KESİRLİ TEKRARLAMA KODLARININ DAĞITILMIŞ DEPOLAMA SİSTEMLERİNE BİR UYGULAMASI.....	31
4.1 Dağıtılmış Depolama Sistemleri.....	31
4.2 Kesirli Tekrarlama Kodları	34

4.3 Kesirli Tekrarlama Kodlarının Dağıtılmış Depolama Sistemlerine Bir	
Uygulaması	38
5. TARTIŞMA VE SONUÇ.....	44
KAYNAKLAR	46
ÖZGEÇMİŞ.....	48



ŞEKİLLER DİZİNİ

Şekil 2.1	Fano düzlemi.....	5
Şekil 2.2	(9,3,1) afin düzlem tasarımı	9
Şekil 2.3	(11,5,2) ikilidüzlem tasarımın bir gösterimi	11
Şekil 2.4	Noktaları bir boyutlu altuzaylar olarak ifade edilen Fano düzlemi	15
Şekil 2.5	K_7 'nin üçlü parçalanışı	18
Şekil 3.1	Silinim kodlarının güvenliğini tekrarlamayla karşılaştırma	29
Şekil 3.2	Silinim kod ile dosyayı oluşturma	30
Şekil 3.3	Silinim kod ile kaybolan düğümü onarım.....	30
Şekil 4.1	Bu şekilde $(n, k, d) = (5,3,4)$ DSS verilmiştir.....	32
Şekil 4.2	Bu şekilde $(n, k, d) = (4,2,2)$ DSS verilmiştir.....	33
Şekil 4.3	Bu şekilde $(7,3,3,1)$ DSS verilmiştir.....	39
Şekil 4.4	Depolanan kodlanmış paketler.....	42
Şekil 4.5	$n = 28, k = 15, r = 3, \theta = 28, \alpha = 3, \rho = 3$ parametrelili bir lokal FR.....	43

ÇİZELGELER DİZİNİ

Çizelge 2.1	n. dereceden projektif düzlemlerin durumları.....	13
Çizelge 2.2	X bir boyutlu altuzaylar A iki boyutlu altuzaylar.	15
Çizelge 2.3	Derecesi 2 olan projektif düzlemlerin blokları.	15
Çizelge 2.4	Yürüyüş programı.	16
Çizelge 3.1	Yalanı yakalama oyunu.....	26
Çizelge 4.1	k = 7 için yapılan dosya boyutu hesaplamasında $\mu = 27$	40
Çizelge 4.2	k = 7 için yapılan dosya boyutu hesaplamasında $\mu = 28$	40
Çizelge 5.1	$d \geq k$ ve $\beta = 1$ şartlarını sağlayan sonuçlar	44
Çizelge 5.1	Kronecker çarpımından elde edilen sonuçlar.....	45

1. GİRİŞ

Büyük ölçekli veri depolama sistemleri giderek daha popüler hale gelmektedir. Manyetik diskler gibi düşük maliyetli depolama ortamları bu sistemlerin Facebook, Youtube ve benzeri sosyal paylaşım sitelerinde kullanımlarını artırmıştır. Bulut depolama sistemleri de son derece popüler bir başka uygulamadır. Bu sistemlerde bireysel kullanıcılar ve kuruluşlar depolama ihtiyaçlarını Microsoft ve Amazon gibi şirketlerden karşılayabilmektedirler. Bu bulut depolama sağlayıcıları, büyük depolama merkezlerinde sürekli eklenen ve güncellenen veri akışını sağlayarak hizmet vermektedir. Örneğin her gün Youtube paylaşım sitesine yaklaşık 72 saatlik uzunluğunda video ve Facebook sitesine yaklaşık 500 terabaytlık yeni verinin eklendiği tahmin edilmektedir. Facebook birkaç yüz beta baytlık veriyi kendi depolama düğümlerinde yönetmektedir. Bu astronomik miktardaki veri devasa depolama sistemin içinde dünyanın herhangi bir yerinde olan kullanıcılar tarafından sorunsuz bir şekilde kullanılmak üzere saklanılır.

Dağıtılmış depolama sistemleri büyük ölçekteki verinin dağıtılmış bir şekilde depolanması için kullanılır. Bu sistemler için tasarlanan rejenerasyon kodların incelenmesi ilk olarak Dimakis tarafından ele alınmıştır (Dimakis vd. 2010). Bu çalışmalarında Rastgele Ağ kodlamasıyla fonksiyonel tamir yapabilen sistemler üzerinde sonuçlar elde etmişlerdir. Ayrıca Dimakis (Dimakis vd. 2010) çalışmasında depolama kapasitesi, onarım ve bant genişliği arasındaki ilişkiyi enformasyon teorik bir eğri ile açıklamıştır. Bu eğri üzerindeki iki önemli nokta göze çarpmaktadır. Noktalardan birincisi minimum bant genişliği rejenerasyonu (MBR) ve ikincisi minimum depolama rejenerasyonu (MSR) olarak adlandırılmıştır. MBR noktasında sistemde toplam bant genişliği minimize edilirken, MSR noktasında amaç depolanan paket sayısını belirten parametresini minimize etmektir. Fonksiyonel tamir yapabilen sistemler dışında tam onarım yapabilen sistemler üzerindeki modellemeler ve çözümler (Rashmi vd. 2009, El Rouayheb ve Ramchandran 2010, Rashmi vd. 2011, Suh ve Ramchandran 2011, Tamo vd. 2011, Papailiopoulos vd. 2012, Shah vd.

2012, Tian vd. 2013, Ölmez ve Ramamoorthy 2016) referanslarında bulunabilir. Bu sistemlerdeki amaç bir depolama düğümünde kayıp yaşanması halinde, kayıp verinin tam bir kopyasının yeniden oluşturulmasını sağlamaktır. (Rashmi vd. 2011, Suh ve Ramchandran 2011, Tamo vd. 2011, Shah vd. 2012, Papailiopoulos vd. 2012) çalışmalarındaki rejenerasyon kodlar, MSR ve (Rashmi vd. 2009, El Rouayheb ve Ramchandran 2010, Rashmi vd. 2011, Shah vd. 2012, Ölmez ve Ramamoorthy 2016) çalışmalarındaki kodlar MBR noktasında çalışmaktadır.

Paketler arasında kodlama yapmayan ve tam (tablo tabanlı) onarım işlemi sunan rejenerasyon kodlar (El Rouayheb ve Ramchandran 2010, Shah vd. 2012, Shum ve Hu 2012, Hu vd. 2013) çalışmalarında ele alınmıştır. Bu kodlar, sistemdeki bir depolama düğümünde hasar olması durumunda hasar görmüş verinin yeniden oluşturulmasını erişim sağlanan diğer düğümlerden sadece gerekli paketleri indirerek sağlamaktadır. (El Rouayheb ve Ramchandran 2010) çalışmasında sistem mimarisinin dış kısmı bir maksimum uzaklıkla ayrılabilen (MDS) kod ve bu kodu takip eden bir iç kesirli tekrarlama (FR) kodundan oluşan rejenerasyon kodlar kullanılarak tasarlanmıştır. Tasarlanan bu kodlar kombinatoriyal tasarım teorisinden faydalanılarak elde edilmiştir. (Ko ve Gill 2011) referansta bu çalışmayı takip ederek çizgilerden yararlanarak kodlar inşa edilmiştir. Ancak bu çalışma dosya boyutu hesaplamasından bağımsızdır. Tasarımlar ayrıca (Silberstein vd. 2015) çalışmalarında da rejenerasyon kod inşa etmek için kullanılan temel bir yapıdır.

Dağıtılmış depolama sistemlerinde gözönünde bulundurulması gereken önemli sistem özelliklerinden bir tanesi onarım sırasında ulaşılması gereken hayatta kalan depolama düğümlerinin sayısıdır. Bu özelliğin detaylı bir biçimde ele alındığı rejenerasyon kodlar lokal rejenerasyon kodlar olarak literatürde yer almıştır (Oggier ve Datta 2011, Gopalan vd. 2012, Papailiopoulos ve Dimakis 2012, Kamath vd. 2013, Ölmez ve Ramamoorthy 2013, Kamath vd. 2014). Lokal kesirli tekrarlama kodları ilk olarak (Papailiopoulos ve Dimakis 2012, Kamath vd. 2014) çalışmalarında ele alınmış ve bu kodların minimum uzaklıkları detaylı olarak incelenmiştir.

Bu tezde kombinatorial tasarımlardan elde edilen kesirli tekraralama kodları ve dağıtılmış depolama sistemlerine yer verilmiştir. Örnek olarak Steiner sistemler, projektif geometriler, çözülebilir tasarımlar, afin çözülebilir tasarımların üzerinde incelemelerde bulunulmuştur.



2. KOMBİNATORİYAL TASARIM TEORİSİ

Kombinatoryal tasarım teorisi; sonlu küme sistemlerinin özelliklerini araştıran ayrık matematiğin bir alt dalıdır. Bu alandaki temel problemlerin çoğu tasarımların varlığıyla ilgili sorulardır. Belirli özelliklere sahip tasarımların varlığını araştırmak ilgi çekici ve zor bir problemdir. Tasarım teorisi 20.yy'ın başlarından itibaren matematiksel bir disiplin olarak görülmeye başlanmıştır. Tarihsel gelişiminde ise tasarım teorisi; turnuva programlama, piyangolar, matematiksel biyoloji, iletişim ağları, algoritma tasarımı ve analiz etme, kriptoloji alanlarındaki problemlere çözümler üretmiştir. Tasarım teorisi ile ilgili problemlerin çözümünde lineer cebir, gruplar, halkalar, cisimler, sonlu geometri ve sayılar teorisi ile ilgili matematiksel sonuçlar sıklıkla kullanılmaktadır.

2.1 Kombinatoryal Tasarımlar

Tanım 2.1.1 X sonlu bir küme ve \mathcal{A} , X 'in boştan farklı alt kümelerinin bir koleksiyonu olsun. (X, \mathcal{A}) ikilisine bir **kombinatoryal tasarım** adı verilir.

Dolayısıyla X kümesinin bir elemanına "nokta", \mathcal{A} kümesinin bir elemanına "blok" (doğru) adı verilir.

Tanım 2.1.2 Bir (X, \mathcal{A}) tasarımı için aşağıdaki özellikler sağlanıyorsa; (X, \mathcal{A}) tasarımına (v, k, λ) -**tasarımı** adı verilir:

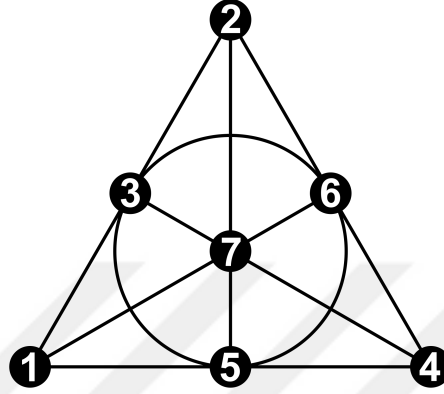
1. $|X| = v$,
2. her blok k tane nokta içerir,
3. herhangi bir nokta ikilisini içeren tamı tamına λ blok vardır.

Örnek 2.1.1 $(7, 3, 1)$ -tasarımının noktaları ve blokları:

$X = \{1, 2, 3, 4, 5, 6, 7\}$ ve

$\mathcal{A} = \{123, 145, 167, 246, 257, 347, 356\}$ şeklindedir.

Bu tasarım Fano düzlemi olarak da adlandırılır. Fano düzlemi şekil 2.1’de resmedilmiştir.



Şekil 2.1 Fano düzlemi

Aşağıdaki teoremden bir kombinatoriyal tasarımın parametreleri arasındaki ilişkiler belirtilecektir.

Teorem 2.1.1 b tane bloğa sahip bir (v, k, λ) -tasarımının parametreleri aşağıdaki eşitlikleri sağlar.

$$\lambda(v - 1) = r(k - 1) \quad (2.1)$$

$$bk = vr \quad (2.2)$$

İspat 2.1.1 x bir nokta ve r_x , x noktasını içeren blokların sayısı olsun. Seçilen herhangi bir x noktası için,

$$\{(y, A) : x \neq y \in A \in \mathcal{A} \text{ ve } x \in A\}$$

kümesinin eleman sayısı iki farklı biçimde sayalım. Birinci bileşen x 'ten farklı herhangi bir nokta olabileceği için birinci bileşeni seçmek için $v - 1$ tane seçenek vardır.

Birinci bileşen y olarak seçildikten sonra ikinci bileşen, x ve y noktalarının her ikisini birden içeren bir blok olarak seçilmelidir; yani x ve y 'yi içeren λ seçenek vardır.

Böylece;

$$|\{(y, A) : x \neq y \in A \in \mathcal{A} \text{ ve } x \in A\}| = (v - 1) \lambda$$

olur.

İkinci bileşen x 'i içeren herhangi bir blok olabileceğinden, ikinci bileşen için r_x seçenek vardır. A bu bloklardan biri olsun. İkinci bileşen seçildiğinde, birinci bileşen A 'nın x 'ten farklı herhangi bir noktası olabilir. Dolayısıyla birinci bileşen için $k - 1$ seçenek vardır. Böylece;

$$|\{(y, A) : x \neq y \in A \in \mathcal{A} \text{ ve } x \in A\}| = r_x (k - 1)$$

olur.

Buradan,

$$(v - 1) \lambda = r_x (k - 1)$$

eşitliği ortaya çıkar $r_x = r$ olur.

Şimdi de $bk = vr$ eşitliğini ispat edelim. Bu eşitliği ispatlamak için,

$$\{(x, A) : x \in A \in \mathcal{A}\}$$

kümesinin eleman sayısı iki farklı biçimde sayılacaktır.

Birinci bileşen için v seçenek vardır. v seçenek arasından bir x seçildiğinde, ikinci bileşen, x 'i içeren herhangi bir blok olabilir. Dolayısıyla ikinci bileşen için r seçenek vardır. Böylece;

$$|\{(x, A) : x \in A \in \mathcal{A}\}| = vr$$

olur.

İkinci bileşen için b seçenek vardır, b tane seçenek arasından A seçildiğinde, birinci bileşen A 'daki herhangi bir nokta olabilir ve bunlardan k tane vardır. Böylece;

$$|\{(x, A) : x \in A \in \mathcal{A}\}| = bk$$

olur. Buradan

$$vr = bk$$

eşitliği ortaya çıkar.

Örnek 2.1.2 $X = \{1, 2, 3, 4, 5, 6\}$

$\mathcal{A} = \{123, 124, 135, 146, 156, 236, 245, 256, 345, 346\}$

ile verilen (X, \mathcal{A}) ikilisi bir $(6, 3, 2)$ -tasarımıdır.

Bu tasarımın parametreleri $v = 6, b = 10, r = 5, k = 3, \lambda = 2$ dir.

2.2 Oluşum Matrisi

Tanım 2.2.1 $X = \{x_1, \dots, x_v\}$ ve $\mathcal{A} = \{A_1, \dots, A_b\}$ olmak üzere, aşağıdaki kuralla tanımlanan $v \times b$ tipindeki $M = (m_{i,j})$ matrisine

$$m_{i,j} = \begin{cases} 1, & x_i \in A_j \\ 0, & x_i \notin A_j \end{cases}$$

(X, \mathcal{A}) ikilisinin **oluşum matrisi** denir.

(v, b, r, k, λ) tasarımının M oluşum matrisi aşağıdaki özellikleri sağlar:

1. Her $i \in 1, \dots, v$ için $S_i = \{j : m_{i,j} = 1\}$ kümesinin eleman sayısı k 'dir.
2. Her $j \in 1, \dots, b$ için $K_j = \{i : m_{i,j} = 1\}$ kümesinin eleman sayısı r 'dir.
3. Her $i \neq i'$ için $|S_i \cap S_{i'}| = \lambda$ 'dir.

Örnek 2.2.1 $X = \{1, 2, 3, 4, 5, 6\}$ ve $\mathcal{A} = \{123, 124, 135, 146, 156, 236, 245, 256, 345, 346\}$ şeklindeki $(6, 3, 2)$ -tasarımın bir oluşturma matrisi 6×10 tipindeki aşağıdaki matristir:

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Teorem 2.2.1 M matrisi girdileri 0 ve 1 girdilerinden oluşan $v \times b$ tipinde bir matris, I_v $v \times v$ tipinde birim matris ve J_v $v \times v$ tipinde tüm girdileri 1 olan bir matris olsun. \mathbf{u}_n ; n uzunluğunda 1 girdilerinden oluşan bir vektörü ifade etsin. Oluşturma matrisi M olan bir (v, b, r, k, λ) - tasarımı aşağıdaki denklemleri sağlar:

$$MM^T = \lambda J_v + (r - \lambda) I_v$$

$$\mathbf{u}_v M = k \mathbf{u}_b.$$

Bir (v, b, r, k, λ) - tasarımının varlığı için önemli bir gerekli şart aşağıdaki teoremden ifade edilmiştir.

Teorem 2.2.2 (Fisher Eşitsizliği) Bir (v, b, r, k, λ) -tasarımının parametreleri

$$b \geq v$$

eşitsizliğini sağlar.

2.3 Simetrik Tasarımlar

Projektif düzlemde nokta sayısı ve doğru sayısına eşittir ve noktalar doğrular, doğrular noktalar olarak görülebilirler. Dolayısıyla herhangi iki doğrunun kesişiminde

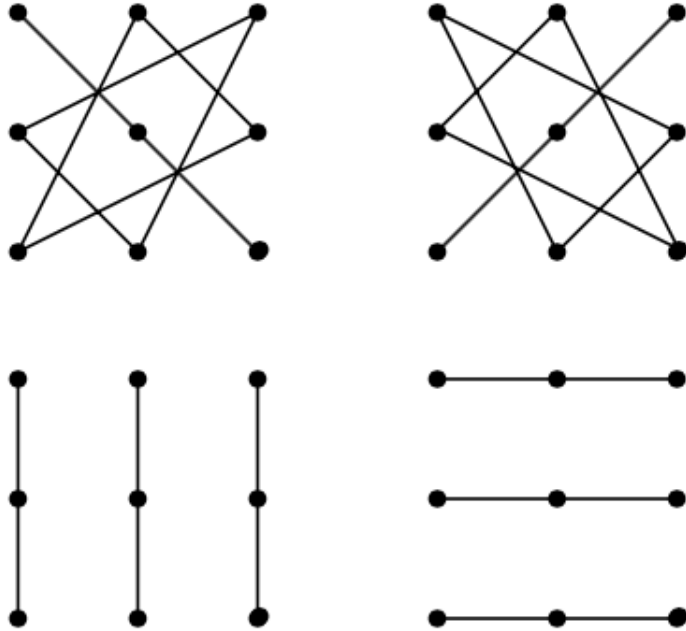
tek nokta olduğu gibi herhangi iki noktadan geçen tek doğru vardır. Bu kolaylığı sağlayan özel tasarımlar vardır. Bu tasarımlar simetrik tasarımlar olarak bilinirler. Fakat her tasarım için bu özellik sağlanmaz. Örneğin afin düzlem projektif düzlem-den bir doğru çıkarılarak elde edilen bir düzlemdir dolayısıyla nokta sayısı doğru sayısına eşit değildir. Örnek olarak aşağıdaki afin düzlem örneği incelenebilir.

Örnek 2.3.1 $X = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$\mathcal{A} = \{123, 456, 789, 147, 258, 369, 159, 267, 348, 168, 249, 357\}$

ile verilen (X, \mathcal{A}) ikilisi bir $(9, 3, 1)$ -tasarımdır. Gösterim olarak şekil 2.2'de verilmiştir.

Tasarımın 12 bloğundan sekizi doğru ve dördü üçgen şeklinde resmedilmiştir. Ayrıca blokların dört sınıfa ayrıldığını ve her bir sınıfın, tasarımın tüm noktalarını içerdiğini gözlemleyebiliriz.



Şekil 2.2 $(9,3,1)$ Afin düzlem tasarımı

Tanım 2.3.1 (Simetrik tasarım) Bir (v, b, r, k, λ) -tasarımı $b = v$ ($r = k$) özelliğini sağlıyorsa bu tasarımlara simetrik tasarım denir.

Uyarı 2.1 Simetrik tasarımlarda, tasarımların oluşum matrisleri simetrik olmak zorunda değildir.

Tasarıma simetrik denmesinin sebebi; noktalar ve bloklar arasındaki ilişkinin simetrik özellikler taşımasıdır. Noktalar ve bloklar için aşağıdaki şartlar sağlanır:

1. her blok k noktaya sahiptir,
2. her nokta k blokta yer alır,
3. herhangi iki farklı nokta λ blokta bulunur,
4. herhangi iki farklı blok λ noktada kesişir.

Eğer simetrik bir (v, b, r, k, λ) - tasarımı nokta blok, bloklar da nokta yapılırsa yine simetrik bir (v, b, r, k, λ) tasarımı elde edilir.

2.3.1 İkildüzlem tasarımlar

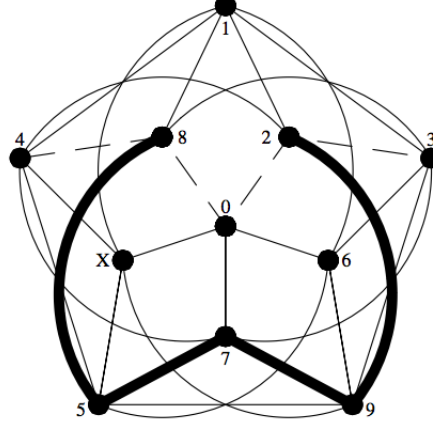
Simetrik $(v, k, 2)$ -tasarımına ikildüzlem tasarımlar denir. İkildüzlemler için aşağıdaki aksiyomlar sağlanır:

1. Herhangi iki farklı nokta tam olarak iki doğru üzerinde bulunur ve
2. Herhangi iki farklı doğru tam olarak iki noktada kesişir.

Simetrik $(v, k, 2)$ - tasarımı $k = 3, 4, 5, 6, 9, 11, 13$ olduğu durumlarda ikildüzlemlerin varlığı bilinmektedir.

Örnek 2.3.2 $(11, 5, 2)$ ikildüzlem tasarımı simetrik bir tasarımdır. Noktalar \mathbb{Z}_{11} 'in elemanlarıdır ve bloklar aşağıdaki tabloda verilmiştir.

$$\begin{array}{lll}
B_1 = \{1, 3, 4, 5, 9\} & B_2 = \{2, 4, 5, 6, X\} & B_3 = \{3, 5, 6, 7, 0\} \\
B_4 = \{4, 6, 7, 8, 1\} & B_5 = \{5, 7, 8, 9, 2\} & B_6 = \{6, 8, 9, X, 3\} \\
B_7 = \{7, 9, X, 0, 4\} & B_8 = \{8, X, 0, 1, 5\} & B_9 = \{9, 0, 1, 2, 6\} \\
B_X = \{X, 1, 2, 3, 7\} & B_0 = \{0, 2, 3, 4, 8\}
\end{array}$$



Şekil 2.3 $(11, 5, 2)$ İkilibesgen tasarımının bir gösterimidir. Düzgünbeşgenin köşeleri üzerindeki 1, 3, 4, 5, 9 noktaları B_1 doğrusunu oluşturur. B_0 doğrusu kesikli çizgiyle B_5 doğrusu koyu renkle gösterilmiştir. Bu iki doğrunun 0 noktası etrafında saat yönünde 72° döndürülmesiyle sırasıyla B_9, B_3, B_7, B_8 ve B_2, B_4, B_X, B_6 doğruları elde edilir.

Bu tasarımın oluşum matrisi aşağıda verilmiştir.

$$M = \begin{pmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0
\end{pmatrix}$$

2.3.2 n. dereceden projektif düzlemler

Tanım 2.3.2 (Projektif Düzlem) $n \geq 2$ olsun. Simetrik $(n^2 + n + 1, n + 1, 1)$ -tasarımına derecesi n olan bir projektif düzlem adı verilir.

En küçük projektif düzlem, derecesi iki olan şekil 2.1 'de resmedilen Fano düzlemidir. Projektif düzlemler bugüne kadar sınıflandırılmamıştır ve hala sınıflandırma yapacak kadar güçlü yöntemlerin bilinmediği düşünülmektedir. Şimdiye kadar derecesi asal bir sayının kuvveti olmayan projektif düzlem bulunamamıştır.

Bruck-Ryser-Chowla Teoremi n . dereceden projektif düzlemlerin hangi durumda var olmadığını ifade eden tek genel teoremdir.

Teorem 2.3.1 (Bruck-Ryser-Chowla) (X, \mathcal{A}) tasarımı bir simetrik (v, k, λ) -tasarımı olsun.

i. Eğer v çiftse, $k - \lambda$ bir tam karedir.

ii. Eğer v tekse,

$$z^2 = (k - \lambda)x^2 + (-1)^{\frac{v-1}{2}} \lambda y^2$$

denkleminin tamsayılar da $(0, 0, 0)$ 'dan farklı bir çözümü vardır.

Bruck-Ryser-Chowla Teoremi n . dereceden bir projektif düzlem için düşünülürse; $v = n(n + 1) + 1$ tektir ve $\frac{v-1}{2} = \frac{n(n+1)}{2}$ olur. Simetrik tasarımlarda v tek durumu için aşağıdaki Diophantine denklemleri ele alınmalıdır.

i. $z^2 = nx^2 + y^2$ eğer $n \equiv 0, 3 \pmod{4}$ ise

ii. $z^2 = nx^2 - y^2$ eğer $n \equiv 1, 2 \pmod{4}$ ise

Teorem 2.3.2 n . dereceden bir projektif düzlem varsa ve $n \equiv 1, 2 \pmod{4}$ ise n iki tam karenin toplamı olarak yazılabilir.

Teorem 2.3.1'e göre derecesi 6 olan bir projektif düzlem yoktur. $10 = 1^2 + 3^2$ olmasına rağmen derecesi 10 olan bir projektif düzlemin olmadığı bilgisayar yardımıyla 1989'da ispatlanmıştır (Lam 1989). Dolayısıyla teoremin tersi doğru değildir.

r pozitif tamsayı ve p bir asal sayı olmak üzere; bilinen tüm projektif düzlemler, derecesi $n = p^r$ şeklinde bir asalın kuvvetidir. Çizelge 2.1'de n . dereceden projektif düzlemler için kısa bir liste verilmiştir.

Çizelge 2.1 n . dereceden projektif düzlemlerin durumları

n	$durum$	n	$durum$
2	$\exists!$	11	\exists
3	$\exists!$	12	?
4	$\exists!$	13	$\exists (13 = 2^2 + 3^2)$
5	$\exists!$	14	$\nexists (BRC)$
6	$\nexists (BRC)$	15	?
7	$\exists!$	16	$\exists 7$
8	$\exists 3$	17	\exists
9	$\exists \geq 4$ (açık)	18	?($18 = 3^2 + 3^2$)
10	$\nexists (?10 = 1^2 + 3^2)$	19	\exists

2.3.3 Derecesi bir asal sayının kuvveti olan projektif düzlemlerin inşası

q bir asal sayının kuvveti ve $|\mathbb{F}_q| = q$ ve $\mathcal{V} = \mathbb{F}_q^3$ olsun. (X, \mathcal{A}) tasarımının noktaları ve blokları

$X = \{C : C \subseteq \mathcal{V}, \text{boyut}(C) = 1\}$ ve

$\mathcal{A} = \{B : B \subseteq \mathcal{V}, \text{boyut}(B) = 2\}$ olsun.

$(x_1, x_2, x_3) \in \mathcal{V}$ vektörü $x_1x_2x_3$ ile gösterilecektir.

- (X, \mathcal{A}) ikilisinin derecesi q olan bir projektif düzlem olduğunu gösterelim.

$000 \in C$ ve $|C| = q$ 'dir. Böylece bir-boyutlu altuzayların sayısı;

$$|X| = \frac{q^3 - 1}{q - 1} = q^2 + q + 1$$

olarak hesaplanır.

$B \in \mathcal{A}$ iki-boyutlu altuzaylarını alalım. $|B| = q^2$ 'dir. Blokların içerdiği nokta sayısını hesaplamak için, iki-boyutlu alt uzayın kapsadığı bir-boyutlu altuzaylar sayılacaktır. Dolayısıyla;

$$\frac{q^2 - 1}{q - 1} = q + 1$$

blokların içerdiği nokta sayısı olacaktır.

Son olarak, $C_1 \neq C_2$ olmak üzere; $C_1, C_2 \in X$ alınsın. C_1 ve C_2 bir-boyutlu altuzayları tek bir B iki-boyutlu altuzayında bulunur. C_1 ve C_2 noktalarını barındıran tek bir B bloğu vardır. Buradan $\lambda = 1$ olacaktır.

Sonuç olarak aşağıdaki teorem elde edilir.

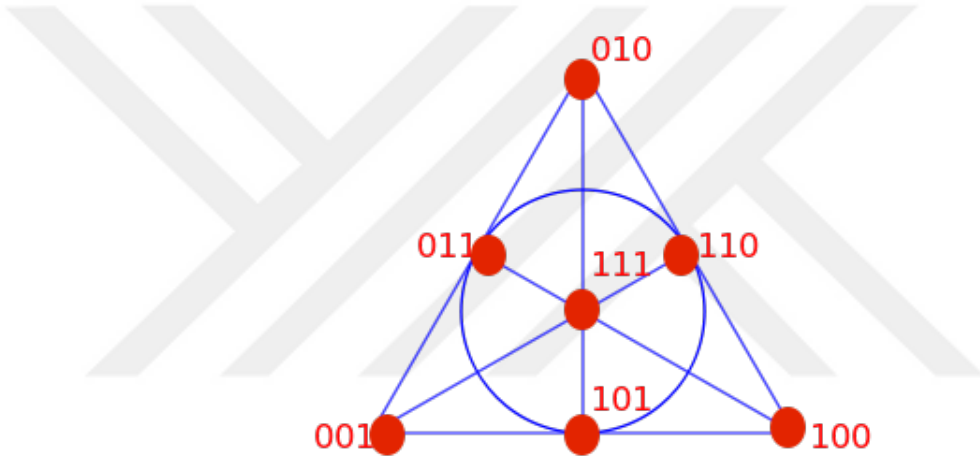
Teorem 2.3.3 $q \geq 2$ olsun. q , asalın her kuvveti için q . dereceden bir $(v, k, \lambda) = (q^2 + q + 1, q + 1, 1)$ projektif düzlem vardır.

Örnek 2.3.3 \mathbb{F}_2 sonlu cismi üzerinde derecesi 2 olan bir projektif düzlemi inşa edelim.

\mathbb{F}_2^3 'ün bir-boyutlu ve iki-boyutlu altuzayları çizelge 2.2'de verilmiştir. Bir boyutlu altuzaylar noktaları ve iki boyutlu altuzaylar doğruları ifade etmek üzere elde edilen projektif düzlem tasarımı Fano düzlemine karşılık gelir. Bu tasarım şekil 2.4'te resmedilmiştir. Oluşan projektif düzlemin blokları ise çizelge 2.3'te verilmiştir.

Çizelge 2.2 X , bir-boyutlu altuzaylar ve \mathcal{A} , iki-boyutlu altuzaylar

X	\mathcal{A}
$C_1 = \{000, 001\}$	$B_1 = \{000, 001, 010, 011\}$
$C_2 = \{000, 010\}$	$B_2 = \{000, 001, 110, 111\}$
$C_3 = \{000, 100\}$	$B_3 = \{000, 001, 100, 101\}$
$C_4 = \{000, 011\}$	$B_4 = \{000, 101, 110, 011\}$
$C_5 = \{000, 101\}$	$B_5 = \{000, 010, 100, 110\}$
$C_6 = \{000, 110\}$	$B_6 = \{000, 010, 101, 111\}$
$C_7 = \{000, 111\}$	$B_7 = \{000, 100, 011, 111\}$



Şekil 2.4 Noktaları bir- boyutlu altuzaylar olarak ifade edilen Fano düzlemi

Çizelge 2.3 Derecesi 2 olan projektif düzlemin blokları

$A_{B_1} = \{C_1, C_2, C_4\}$
$A_{B_2} = \{C_1, C_6, C_7\}$
$A_{B_3} = \{C_1, C_3, C_5\}$
$A_{B_4} = \{C_4, C_5, C_6\}$
$A_{B_5} = \{C_2, C_5, C_6\}$
$A_{B_6} = \{C_2, C_5, C_7\}$
$A_{B_7} = \{C_3, C_4, C_7\}$

2.4 Steiner Üçlü Sistemleri

Tasarım teorisinin geçmişinde sorulan soruların teorisinin gelişmesinde önemli bir yeri vardır. Bunlardan 1850’de Kirkman’ın sorduğu Kirkman öğrenci problemi olarak bilinen problem de ayrıca önemlidir. Kirkman öğrenci problemi blok tasarımları adı verilen bir konunun gelişmesine katkı sağlamıştır.

Kirkman Öğrenci Problemi şöyledir: Bir öğretmen 15 öğrencisini haftanın yedi günü yürüyüşe çıkarsın. Öğrenciler 3’erli beş sıra halinde yürüsünler. Bu yürüyüş programını herhangi iki öğrenci yalnızca bir gün aynı sırada yürüyecek şekilde ayarlamak mümkün müdür?

Aşağıdaki yürüyüş programı bu sorunun yanıtının mümkün olduğu çözümlerden biridir. 15 öğrenci 1’den 15’e kadar sayılarla ifade edilerek çizelge 2.4 oluşturulmuştur.

Çizelge 2.4 Kirkman öğrenci yürüyüş programı

Pazartesi	Salı	Çarşamba	Perşembe	Cuma	Cumartesi	Pazar
1 2 3	1 4 5	1 6 7	1 8 9	1 10 11	1 12 13	1 14 15
4 8 12	2 8 10	2 9 11	2 12 15	2 13 14	2 4 6	2 5 7
5 10 14	3 13 15	3 12 14	3 5 6	3 4 7	3 9 10	3 8 11
6 11 13	6 9 14	4 10 15	4 11 14	5 9 12	5 11 15	4 9 13
7 9 15	7 11 12	5 8 13	7 10 13	6 8 15	7 8 14	6 10 12

Tanım 2.4.1 (Steiner Üçlü Sistemi) (v, k, λ) -tasarımlarda parametrelerin $k = 3$ ve $\lambda = 1$ olduğu durumlarda bu tasarım $(v, 3, 1)$ Steiner üçlü sistemi olarak adlandırılır.

Açık bir şekilde bloklar üçlülerdir ve herhangi birbirinden farklı nokta çifti tam

olarak bir blok tarafından içerilir.

Örnek 2.4.1 $v = 7$,

$$\mathcal{A} = \{\{1, 2, 4\}, \{2, 3, 5\}, \{3, 4, 6\}, \{4, 5, 7\}, \{5, 6, 1\}, \{6, 7, 2\}, \{7, 3, 1\}\}$$

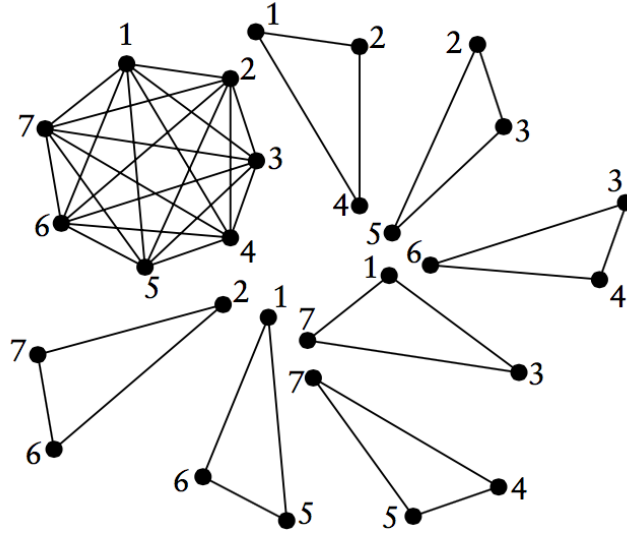
Bu tasarım $(7, 3, 1)$ Steiner üçlü sistemi örneğidir.

Tanım 2.4.2 (Çizge) V köşe olarak adlandırılan nokta veya düğüm kümesi ve E kenar olarak adlandırılan V nin iki elemanlı alt kümelerinden oluşan küme koleksiyonu olmak üzere $\Gamma = (V, E)$ ikilisine çizge denir.

Tanım 2.4.3 (Tamçizge) Birbirinden farklı herhangi iki köşesi bağlantılı olan çizgelere tamçizge denir. n köşesi olan bir tamçizge K_n ile gösterilir.

Örnek 2.4.2 $V = \{1, 2, 3, 4\}$ ve $E = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\}$ ile verilen $\Gamma = (V, E)$ çizgesi bir tamçizgedir.

Örnek 2.4.3 Steiner üçlü sistemlerini tamçizgelerle yorumlamak da mümkündür. $(7, 3, 1)$ Steiner üçlü sistemi K_7 tamçizgesinin üçgenlere parçalanmasına denktir (Şekil 2.5). Her kenar yalnızca bir üçgende bulunduğundan her sayı çifti yalnızca bir üçlüde olacaktır.



Şekil 2.5 K_7 nin Üçlü Parçalanışı

Örnek 2.4.4 Çizelge 2.4'ten yararlanarak her günün her sırası bir üçlü olarak görüldüğünde Kirkman öğrenci probleminin her çözümü $(15, 3, 1)$ Steiner üçlü sistemi örneğidir.

Aşağıdaki teorem hangi v 'ler için Steiner üçlü sistemlerinin var olduğunu ifade etmektedir.

Teorem 2.4.1 (Kirkman 1847)

(v, k, λ) Steiner üçlü sisteminin varlığı için gerek ve yeter şart $v \equiv 1$ ya da $3 \pmod{6}$ 'dır. Ayrıca bu durumda blok sayısı

$$b = \frac{v(v-1)}{6}$$

dır.

2.5 Çözülebilir Tasarımlar

Tanım 2.5.1 (Çözülebilir Tasarım) (X, \mathcal{A}) ikilisi bir (v, b, r, k, λ) -tasarımı olsun. $P \subset \mathcal{A}$ kümesinin birbirinden farklı her $A_i \in P$ ve $A_j \in P$ elemanı için; $A_i \cap A_j = \emptyset$

ve $\cup_{j:A_j \in P} A_j = X$ eşitlikleri sağlanıyorsa P kümesine bir **paralel sınıf** adı verilir. \mathcal{A} kümesi b/k tane paralel sınıfa ayrıştırılabiliyorsa (X, \mathcal{A}) tasarımı çözülebilirdir. Bu özelliği sağlayan (v, k, λ) -tasarımlara **çözülebilir tasarım** denir.

Örnek 2.5.1 Örnek 2.3.1'de noktaları ve blokları verilen $(9, 3, 1)$ -tasarımın bloklarının dört paralel sınıfa ayrıldığı ve her bir sınıfın, tasarımın tüm noktalarını içerdiği gözlemlenebilir. Bu paralel sınıflar aşağıdaki verilmiştir.

$$P_1 = \{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}\}$$

$$P_2 = \{\{1, 4, 7\}, \{2, 5, 8\}, \{3, 6, 9\}\}$$

$$P_3 = \{\{1, 5, 9\}, \{2, 6, 7\}, \{3, 4, 8\}\}$$

$$P_4 = \{\{1, 6, 8\}, \{2, 4, 9\}, \{3, 5, 7\}\}$$

Örnek 2.5.2 Kirkman öğrenci problemi en ünlü çözülebilir tasarım örneğidir. Çizelge 2.4'ten gözlemlendiğinde bu problemin çözümünde her günün bir paralel sınıfa karşılık geldiği görülür.

3. KODLAMA TEORİSİ

Bilginin iletilmesi veya depolanmasında kullanılan sistemler pratikte tek başına güvenli değildir. Bu sistemlerde bilgi; gürültü veya parazit denilen ve bilginin orijinal halini değiştiren bir takım karıştırıcılara maruz kalabilir. Kodlama teorisinin amacı bu hataları tespit etmek ve düzeltmektir. Günümüzde kompakt disk kayıtlarında hatayı en aza indirmek, telefon hatları üzerinden veya uydudan düzgün bilgi akışını sağlamak gibi çeşitli uygulamaları vardır.

3.1 Temel Tanımlar

Tanım 3.1.1 \mathbb{F}_q , eleman sayısı q olan bir sonlu cisim olsun. $V = \mathbb{F}_q^n$, \mathbb{F}_q üzerinde n boyutlu vektör uzayı olsun. Her $v, w \in \mathbb{F}_q^n$ için vektör toplama ve skalerle vektör çarpımı

$$+ : \mathbb{F}_q^n \times \mathbb{F}_q^n \longrightarrow \mathbb{F}_q^n$$

$$v + w \longmapsto (v_1 + w_1, \dots, v_n + w_n)$$

$$\cdot : \mathbb{F}_q \times \mathbb{F}_q^n \longrightarrow \mathbb{F}_q^n$$

$$\lambda v \longmapsto (\lambda v_1, \dots, \lambda v_n)$$

şeklinde tanımlansın.

\mathbb{F}_q^n vektör uzayının elemanları $v = \{v_1, v_2, \dots, v_n\}$ şeklindedir. $v, w \in \mathbb{F}_q^n$ ve $\lambda \in \mathbb{F}_q$ için toplama ve çarpma işlemleri sırasıyla

$v + w = (v_1 + w_1, \dots, v_n + w_n)$, $\lambda v = (\lambda.v_1, \dots, \lambda.v_n)$ şeklinde yapılır (Ling ve Xing 2004).

Örnek 3.1.1 $\mathcal{C} = \{000, 101, 010, 111\}$, \mathbb{F}_2 üzerinde bir vektör uzayıdır.

Tanım 3.1.2 V vektör uzayının boştan farklı bir \mathcal{C} alt kümesi aynı vektör toplama ve skalerle çarpım altında bir vektör uzayı ise \mathcal{C} ye V nin bir **alt vektör uzayı** denir (Ling ve Xing 2004).

Lemma 3.1.1 \mathbb{F}_q üzerinde bir V vektör uzayının boştan farklı bir \mathcal{C} alt kümesinin bir altuzay olması için gerek ve yeter şart $\forall x, y \in \mathcal{C}$ ve $\forall \lambda, \mu \in \mathbb{F}_q$ için $\lambda x + \mu y \in \mathcal{C}$ olmasıdır (Ling ve Xing 2004).

3.2 Lineer Kodlar

Tanım 3.2.1 $V = \mathbb{F}_q^n$ vektör uzayının boştan farklı bir \mathcal{C} altuzayına \mathbb{F}_q üzerinde n uzunluğunda bir **lineer kod** denir. \mathcal{C} 'nin elemanlarına **kod kelimesi** denir.

Tanım 3.2.2 \mathbb{F}_q üzerinde uzunluğu n , boyutu k olan bir \mathcal{C} lineer kodu, \mathbb{F}_q^n uzayının bir alt uzayıdır. $[\mathbf{n}, \mathbf{k}]_q$ şeklinde gösterilir.

$q = 2$ iken \mathbb{F}_2 üzerinde bir koda ikili kod; $q = 3$ iken \mathbb{F}_3 üzerinde bir koda üçlü kod denir.

3.3 Üreteç Matrisi ve Kontrol Matrisi

Tanım 3.3.1 Bir \mathcal{C} lineer kodu, \mathbb{F}_q^n uzayının bir alt uzayı olduğundan bir bazı vardır. \mathcal{C} nin bir $\{c_1, c_2, \dots, c_k\}$ bazındaki vektörleri satır kabul eden G matrisine, \mathcal{C} kodunun üreteç matrisi denir.

\mathcal{C} kodunun üreteç matrisi; $G = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{bmatrix}_{k \times n}$ ile gösterilir. Dolayısıyla \mathcal{C} deki her kod kelimesi G 'nin satırlarının bir lineer kombinasyonu olarak yazılabilir (Vermani 1996).

Tanım 3.3.2 \mathcal{C}^\perp dual kodun üreteç matrisi, \mathcal{C} lineer kodunun kontrol matrisidir. H ile gösterilen bu matris $\forall x \in \mathcal{C}$ için $H \cdot x^T = 0$ eşitliğini sağlar (Vermani 1996).

Tanım 3.3.3 \mathcal{C} üreteç matrisi G olan bir $[n, k]_q$ -kod olsun.

1. $G = [I_k | A]_{k \times n}$ ise G standart formdadır. I_k , $k \times k$ tipinde birim matris; A $k \times (n - k)$ tipinde bir matristir.
2. $H = [A^T | I_{n-k}]_{(n-k) \times n}$ ise H standart formdadır (Vermani 1996).

3.4 Lineer Kodlama ile Hata Düzeltme

Hata düzeltici kodların ardındaki düşünce, kod kelimelerini birbirlerinden yeterince farklı seçip, gönderilen kelimenin birkaç terimi yanlış iletilse bile, iletilen kelimenin hala tanınabilir olmasını, yani asıl kelimeye diğer kod kelimelerinden daha yakın olmasını sağlamaktır. Bu yakınlığı ölçmek için bir uzaklık fonksiyonu kullanılır.

Tanım 3.4.1 (Hamming Uzaklık) $x = (x_1, x_2, \dots, x_n)$ ve $y = (y_1, y_2, \dots, y_n) \in \mathbb{F}_q^n$ için

$$d(x_i, y_i) = \begin{cases} 1, & x_i \neq y_i \\ 0, & x_i = y_i \end{cases}$$

şeklinde tanımlı olmak üzere; x ve y arasındaki Hamming uzaklık:

$d(x, y) = d(x_1, y_1) + d(x_2, y_2) + \dots + d(x_n, y_n)$ dir (Ling ve Xing 2004).

Örnek 3.4.1 \mathbb{F}_2 üzerinde $x = (1011100)$ ve $y = (0111110)$ için $d(x, y) = 3$ 'tür.

Tanım 3.4.2 (Hamming Ağırlık) \mathbb{F}_q^n 'nin bir elemanı olan x kod kelimesinin Hamming ağırlığı sıfırdan farklı koordinatlarının sayısı $w(x) = |\{i : x_i \neq 0\}|$ olarak tanımlanır (Ling ve Xing 2004).

Örnek 3.4.2 \mathbb{F}_2 üzerinde $x = (1011100)$ için $w(x) = 4$ 'tür.

Tanım 3.4.3 (Minimum Uzaklık) \mathcal{C} bir kod olsun. \mathcal{C} 'nin minimum uzaklığı,

$d_{\min}(\mathcal{C}) = d(\mathcal{C}) = \min \{d(x, y) \mid x \neq y, \forall x, y \in \mathcal{C}\}$ şeklinde tanımlanır (Ling ve Xing 2004).

Minimum uzaklığı d olan bir $[n, k]_q$ -kod, $[n, k, d]_q$ ile gösterilir.

Tanım 3.4.4 (Minimum Ağırlık) Bir \mathcal{C} kodunun minimum ağırlığı sıfırdan farklı tüm kod kelimelerinin ağırlıklarının en küçüğüdür ve $w(\mathcal{C})$ ile gösterilir.

Lemma 3.4.1 Her $x, y \in \mathbb{F}_q^n$ için $d(x, y) = w(x - y)$ dir (Ling ve Xing 2004).

Teorem 3.4.1 \mathcal{C} , \mathbb{F}_q üzerinde bir lineer kod ise $d(\mathcal{C}) = w(\mathcal{C})$ dir (Ling ve Xing 2004).

Örnek 3.4.3 $\mathcal{C} = \{0000, 1000, 0100, 1100\}$ lineer kodunu alalım.

$$w(1000) = 1,$$

$$w(0100) = 1,$$

$$w(1100) = 2.$$

olup $w(\mathcal{C}) = 1$ 'dir. Teorem 3.4.1'den $d(\mathcal{C}) = 1$ olur.

Teorem 3.4.2 \mathcal{C} , \mathbb{F}_q üzerinde bir lineer $[n, k, d]$ -kod ise $d \leq n - k + 1$ eşitsizliği sağlanır (Vermani 1996).

Teorem 3.4.3 Bir \mathcal{C} lineer kodunun minimum uzaklığı d ise bu kod $\lfloor \frac{d-1}{2} \rfloor$ hatayı düzeltebilir ve $\lfloor d - 1 \rfloor$ hatayı tespit edebilir.

Örnek 3.4.4 $k = 4$ uzunluğundaki mesajlar, aşağıdaki G üreteç matrisi kullanılarak

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}_{4 \times 7}$$

$n = 7$ uzunluğunda kod kelimeleri elde edilir. Elde edilen $[7, 4, 3]_2$ kodu teorem 3.4.3'ten $d=3$ olduğundan tek hatayı düzeltir ve herhangi iki hatayı tespit eder.

Mesaj		Kod Kelimesi	Mesaj		Kod Kelimesi
0000	→	0000000	0110	↦	0110010
0001	→	0001011	0101	↦	0101110
0010	→	0010111	0011	↦	0011100
0100	→	0100101	1110	↦	1110100
1000	→	1000110	1101	↦	1101000
1100	→	1100011	1011	↦	1011010
1010	→	1010001	0111	↦	0111001
1001	→	1001101	1111	↦	1111111

Örnek 3.4.5 (Yalanı Yakalama Oyunu)

Oyunun kuralı şöyledir; 1'den 15'e kadar (15 dahil) bir sayı tutulsun ve aşağıdaki sorulara verilen cevaplar için en fazla bir kere yalan söyleme hakkına sahip olunsun.

Aşağıdaki sorulara verilen cevaplara göre hangi sayının tutulduğu ve hangi soruda yalan söylendiği bulunabilir.

Sorular

1. Tuttuğunuz sayı bu kümenin içinde mi $\{4, 6, 7, 8, 9, 10, 12, 15\}$?
2. Tuttuğunuz sayı bu kümenin içinde mi $\{1, 5, 7, 9, 10, 11, 13, 15\}$?
3. Tuttuğunuz sayı bu kümenin içinde mi $\{1, 2, 6, 10, 11, 12, 14, 15\}$?
4. Tuttuğunuz sayı bu kümenin içinde mi $\{2, 3, 7, 8, 11, 12, 13, 15\}$?
5. Tuttuğunuz sayı bu kümenin içinde mi $\{1, 3, 4, 9, 12, 13, 14, 15\}$?
6. Tuttuğunuz sayı bu kümenin içinde mi $\{2, 4, 5, 8, 10, 13, 14, 15\}$?
7. Tuttuğunuz sayı bu kümenin içinde mi $\{3, 5, 6, 8, 9, 11, 14, 15\}$?

Her soru için verilen cevaplara göre sırasıyla evetse 1 ve hayırsa 0 yazılarak 7 elemanlı bir dizi elde edilir.

Elde edilen bu dizi çizelge 3.1'de aranır. Eğer yalan söylenmemişse elde edilen dizi çizelge 3.1'deki i. satırda bulunur ve bu satır tutulan sayıyı verir. Eğer yalan söylenmişse; cevaplardan elde edile diziye en çok benzeyen i.satır tutulan sayıyı ve dizinin bileşenin farklı olduğu j. sütun hangi soruda yalan söylendiğini ifade eder.

Eldeki dizinin çizelgeki satırlara en benzeyenini bulmak; dizinin sırasıyla tüm girdilerini çizelgedeki her bir dizinin girdileri ile tek tek karşılaştırarak farklı girdi sayısının

en az olanını bulmak demektir. Çizelge 3.1 $(7, 3, 1)$ -tasarımının ve onun tümleyeninin oluşum matrisinden oluşmuştur. Tümleyenin oluşum matrisi, bir (v, k, λ) -tasarımının oluşum matrisindeki 1 ve 0 girdilerinin yerleri değiştirilerek oluşturulan matristir.

Bir \mathcal{C} kodunun kod kelimeleri bir matrisin satırları olarak yazılabilir. Ayrıca bir tasarımın oluşum matrisinin satırları bir kodun kod kelimeleri olarak görülebilir.

Örnek 3.4.5'teki yalan yakalama oyunu $[7, 4, 3]_2$ kodunun tek hatayı düzeltme özelliğinden yararlanılarak kurgulanmıştır.

Çizelge 3.1 $(7, 3, 1)$ -tasarımının ve tümleyeninin oluşum matrislerinin alt alta yazılmasıyla oluşturulmuş çizelge

	1	2	3	4	5	6	7
1	0	1	1	0	1	0	0
2	0	0	1	1	0	1	0
3	0	0	0	1	1	0	1
4	1	0	0	0	1	1	0
5	0	1	0	0	0	1	1
6	1	0	1	0	0	0	1
7	1	1	0	1	0	0	0
8	1	0	0	1	0	1	1
9	1	1	0	0	1	0	1
10	1	1	1	0	0	1	0
11	0	1	1	1	0	0	1
12	1	0	1	1	1	0	0
13	0	1	0	1	1	1	0
14	0	0	1	0	1	1	1
15	1	1	1	1	1	1	1

Ayşe ve Burcu bu oyunu oynasınlar. Burcu'nun sırasıyla sorduğu sorulara Ayşe 1011100 dizisinden oluşan cevabı verir. Burcu bu diziyi çizelge 3.1'de arandığında 12. satırda aynısını görür. Bu durumda Ayşe yalan söylememiştir ve tuttuğu sayı

12'dir.

Şimdi de Ayşe Burcu'ya aynı soruları sırasıyla sorsun. Burcu'nun cevaplarından oluşan dizi 1001101'dir. Ayşe bu diziye çizelge 3.1'de arandığında bu diziye en çok benzeyen yalnızca 1. girdisi farklı olan 3. satırı görecektir. Bu durumda Burcu 3 sayısını tutmuştur ve 1. soruda yalan söylemiştir.

3.5 Maksimum Uzaklıkla Ayrılabilen Kodlar

Tanım 3.5.1 C bir $[n, k, d]_q$ lineer kod olsun. $d = n - k + 1$ eşitliğini sağlayan C koduna Maksimum Uzaklıkla Ayrılabilen Kod (MDS) denir (Vermani 1996).

MDS kodlar büyük ölçüde hata düzeltme ve tespit etme yeteneğine sahip olduklarından lineer kodların önemli bir sınıfıdır.

Teorem 3.5.1 C bir $[n, k, d]_q$ lineer kod olsun. Aşağıdaki önermeler denktir:

1. C bir MDS koddur.
2. C lineer kodunun üreteç matrisinin herhangi k sütunu lineer bağımsızdır.
3. C lineer kodunun kontrol matrisinin herhangi $n - k$ sütunu lineer bağımsızdır (Vermani 1996).

Uyarı 3.1 Verilen her (n, k) ikilisi için bir MDS kod vardır. Bu kodun üreteç matrisi Vandermonde matrisi ile verilir.

$\mathbf{a} = (a_1, \dots, a_n)$, n uzunluğunda girdileri birbirlerinden farklı bir vektör olsun.

$$G_k(\mathbf{a}) = \begin{pmatrix} 1 & \cdots & 1 \\ a_1 & \cdots & a_n \\ \vdots & \ddots & \vdots \\ a_1^{k-1} & \cdots & a_n^{k-1} \end{pmatrix}.$$

şeklinde tanımlı Vandermonde matrisi MDS kodun bir üreteç matrisidir.

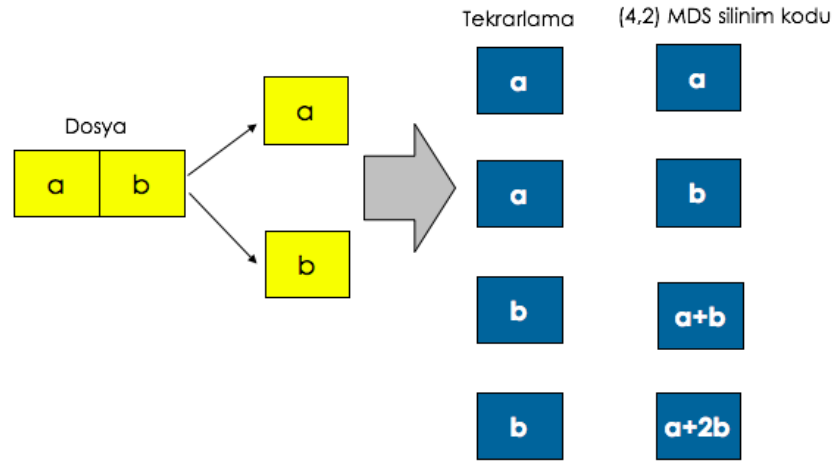
3.6 MDS Kodların Bir Uygulaması

Veri depolama sistemleri günümüz teknolojisinin vazgeçilmez bir parçası haline gelmiştir. Bu sistemlerde, sürekli eklenen ve güncellenen veri saklanarak kullanıcıların hizmetine sunulur.

Silinim kodlar depolama sistemlerinde veri depolamada etkin bir şekilde kullanılmaktadır. Bu sayede depolanan veri olası hasarlara karşı verinin kopyalarının saklanmasıyla daha güvenli şekilde korunur. Bu karşılaştırma şekil 3.1'den gözlemlenebilir. Dosya $2n$ uzunluğunda \mathbb{F}_q üzerinde $(x_1, \dots, x_n, y_1, \dots, y_n)$ biçiminde bir vektör ile gösterilsin. Dosyayı iki parçaya bölelim. Bu parçalar n uzunluğunda a ve b vektörleri olsun. Dosyanın a ve b parçalarının kopyalayarak saklanması durumunda herhangi iki parçaya ulaşılarak dosya tekrar oluşturulamayabilir. Fakat bir tür silinim kod olan $(4, 2)$ -MDS kod ile kodlama yapılarak depolanan kodlanmış verilerden herhangi ikisine ulaşılarak dosya oluşturulur. Böylece depolama düğümlerine saklanan veri olası hasarlara karşı daha güvenli şekilde korunmuş olur.

3.7 Silinim Kodları Kullanarak Onarım

Bir \mathcal{M} dosyası k parçaya bölünür ve elde edilen k tane parçanın kodlanmasıyla n tane kodlanmış parça elde edilir. (n, k) -MDS kod kullanılarak kodlanan veri n depolama düğümlerine depolanır.



Şekil 3.1 Silinim kodlarının güvenliğini tekrarlama ile karşılaştırma

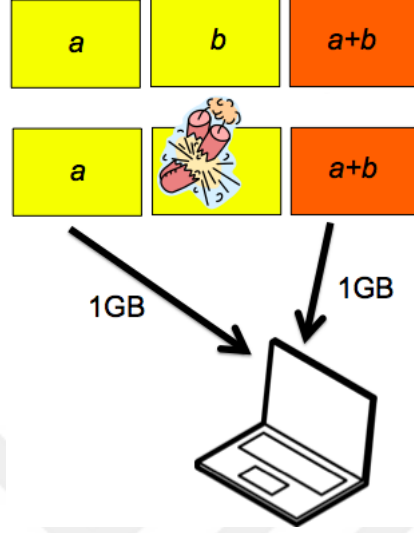
$a + b$; n uzunluğunda iki vektör toplamını ve $a + 2b$; a vektörü ile b vektörünün 2 katının toplamını ifade etmektedir.

Bir veri depolama sisteminde veri (n, k) -MDS kod kullanılarak depolansın. Kodlanarak depolanan n tane depolama düğümünden $n - k$ tanesine zarar gelmesi durumunda hayatta kalan k depolama düğümüne ulaşarak dosya oluşturulabilir. Dosyayı onarma işlemi MDS kod özelliğinden yararlanılarak yapılır.

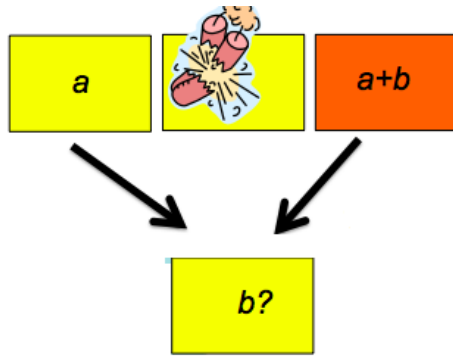
Şekil 3.2’de 2 GB’lık veri $(3, 2)$ -MDS kod kullanılarak 3 depolama düğümüne depolanmıştır. 3 depolama düğümünden herhangi birinde hasar oluşsun. Hayatta kalan 2 depolama düğümüne ulaşarak 2 GB’lık dosya oluşturulur.

Sistemdeki bir depolama düğümünde kayıp yaşanması halinde, erişim sağlanan diğer düğümlerden sadece gerekli paketleri indirilerek kayıp verinin yeniden oluşturulması sağlanır. Şekil 3.3’te b kayıp verisinin tam bir kopyası yeniden oluşturulmuştur.

Onarım sırasında paketler arasında kodlama yapmayan ve tam (tablo tabanlı) onarım işlemi sunan bu sistemde amaç bir depolama düğümünde kayıp yaşanması halinde, kayıp verinin tam bir kopyasının yeniden oluşturulmasını sağlamaktır.



Şekil 3.2 Silinim kod ile dosyayı oluşturma



Şekil 3.3 Silinim kod ile kaybolan düğümü onarım

4. KESİRLİ TEKRARLAMA KODLARININ DAĞITILMIŞ DEPOLAMA SİSTEMLERİNE BİR UYGULAMASI

Dağıtılmış depolama sistemleri büyük ölçekteki verinin dağıtılmış bir şekilde depolanması için kullanılır. Bu depolanan devasa miktardaki verinin her an erişilebilir olmasını sağlayacak matematiksel modeller geliştirmek önemlidir. Bu bölümde, kombinatorial tasarımlar yardımıyla elde edilen matematiksel bir model hakkında yapılan çalışmalara yer verilmiştir.

4.1 Dağıtılmış Depolama Sistemleri

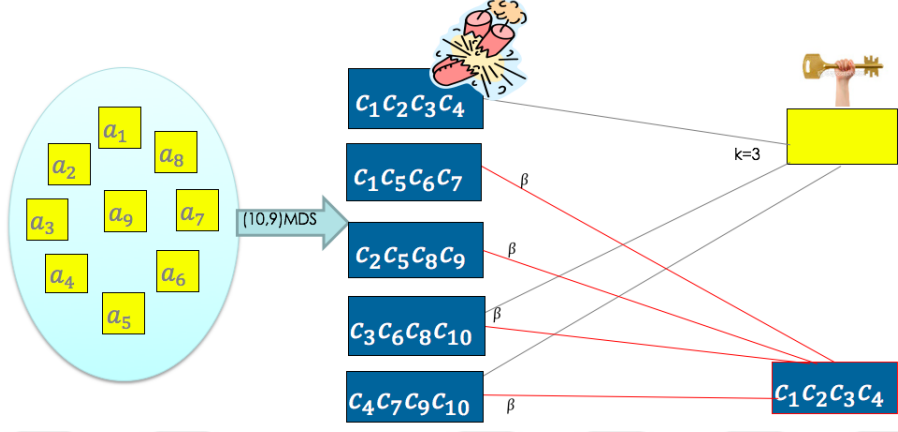
Tanım 4.1.1 (Dağıtılmış Depolama Sistemi) (n, k, d, α) -dağıtılmış depolama sistemi (DSS) boyutu M olan bir dosyayı n tane düğümde depolamak için kullanılan bir yöntemdir.

Bu sistemde her bir depolama düğümü α paket depolayabilmektedir. Sistemdeki bir kullanıcı herhangi k tane depolama düğümüne erişerek depolanan dosyayı oluşturabilir. Bu özellik depolama sisteminin maksimum uzaklıkla ayrılabilme (MDS) özelliği olarak adlandırılır. Belirli bir depolama düğümünde veride kayıp olduğunda sistem erişilebilir herhangi d tane depolama düğümünün her birinden β paket indirerek bu veriyi onarır.

Örnek olarak parametreleri $(n, k, d) = (5, 3, 4)$ olan bir DDS şekil 4.1'de verilmiştir (Papailiopoulos vd. 2012).

Örnek 4.1.1 \mathbb{F}_q^9 ile q elemanlı \mathbb{F}_q sonlu cismi üzerindeki 9 boyutlu vektör uzayını gösterelim. Bu örnekte, 5 depolama düğümünde boyutu $M = 9$ paket olan bir $(a_1, \dots, a_9) \in \mathbb{F}_q^9$ dosyasını depolamak için bir kodlama metodu tanıtılacaktır. Öncelikle dosya $(10, 9)$ MDS kod uygulanarak $c_i = a_i, i = 1, \dots, 9$ ve $c_{10} = \sum_{i=1}^9 a_i$

paketleri elde edilir. Kodlanmış bu paketler daha sonra aşağıdaki gibi $n = 5$ depolama düğümüne yerleştirilir.



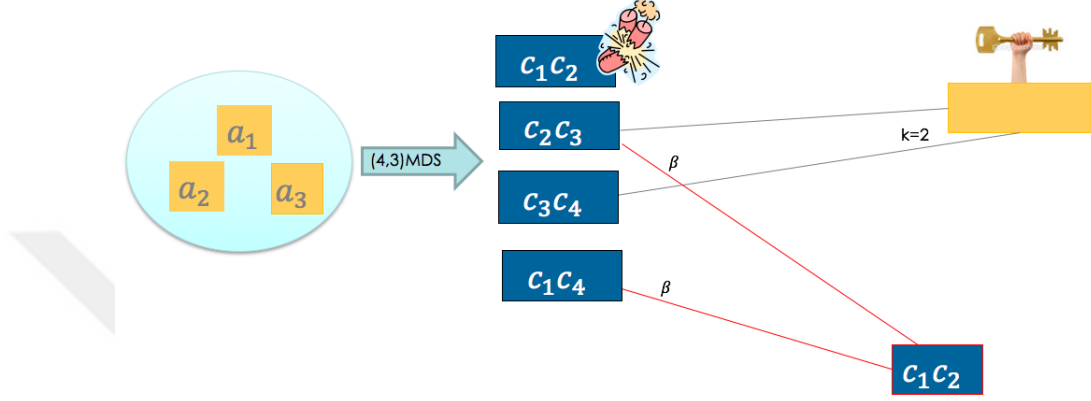
Şekil 4.1 Bu şekilde $(n, k, d) = (5, 3, 4)$ DSS verilmiştir.

Bu yerleştirme bir kesirli tekrarlamaya kodu tarafından belirlenmektedir. Burada kodlanmış her bir paket iki kopyasını sistem içerisinde depolamıştır. Bir kullanıcı herhangi $k = 3$ depolama düğümüne ulaşarak dosyadaki veriye ulaşabilir. Örneğin bir kullanıcı $c_1c_2c_3c_4$, $c_3c_6c_8c_{10}$, $c_4c_7c_9c_{10}$ düğümlerine ulaşsın. $c_1, c_2, c_3, c_4, c_6, c_7, c_8, c_9, c_{10}$ paketlerini elde edecektir. $c_i = a_i, i = 1, \dots, 9$ ve $c_{10} = \sum_{i=1}^9 a_i$ olduğundan dosyadaki c_5 paketi hesaplanabilir. Böylece $(a_1, \dots, a_9) \in \mathbb{F}_q^9$ dosyasına ulaşılır.

Kayıp yaşanan bir düğümün onarımı erişilebilen diğer $d = 4$ düğümün her birinden birer paket indirilerek yapılmaktadır. Örneğin $c_1c_2c_3c_4$ düğümünde veri kaybı yaşanması durumunda sistem $c_1c_5c_6c_7$, $c_2c_5c_8c_9$, $c_3c_6c_8c_{10}$, $c_4c_7c_9c_{10}$ düğümlerine ulaşarak her birinden sırasıya c_1, c_2, c_3, c_4 paketlerini indirerek kayıp yaşanan depolama düğümünün bir kopyasını oluşturur.

Örnek 4.1.1'de verilen kodlama süreçleri matrisler yardımıyla ifade edilebilir. Bunun için daha basit bir örnek incelenecektir.

Örnek 4.1.2 Bu örnekte, her biri 2 paket depolayabilen 4 depolama düğümünden oluşan bir dağıtılmış depolama sistemini ele alalım. 3 paket boyutundaki dosyaya ilk olarak (4, 3) MDS kod uygulanacaktır. Bu işlem sonucunda $c_i = a_i, i = 1, \dots, 3$ ve $c_4 = \sum_{i=1}^3 a_i$ kodlanmış paketleri elde edilir. Kodlanmış bu paketler daha sonra aşağıdaki gibi $n = 4$ depolama düğümüne yerleştirilir.



Şekil 4.2 $(n, k, d) = (4, 2, 2)$ DSS verilmiştir.

Bir kullanıcı herhangi $k = 2$ depolama düğümüne ulaşarak dosyadaki veriyi oluşturabilir. Kayıp yaşanan bir düğümün onarımı tablo tabanlı olarak $d = 2$ adet düğümüne erişilip her birinden $\beta = 1$ adet paket indirilerek yapılmaktadır. Dosya $a = (a_1, a_2, a_3)$ vektörü ile gösterilsin. Aşağıdaki matrisler yardımıyla MDS ve kesirli tekrarlı kodlama süreçleri özetlenecektir.

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad F = \left[\begin{array}{cc|cc|cc|cc} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{array} \right]$$

- a vektörü ile G matrisin çarpımı olan $c = aG = (a_1, a_2, a_3, a_1 + a_2 + a_3)$ vektörünün bileşenleri ile kodlanmış paketler elde edilir.
- c vektörü ile F blok matrisinin çarpımı olan $w = cF = (c_1, c_2, c_2, c_3, c_3, c_4, c_1, c_4)$ vektörünün bileşenleri ikişer ikişer bloklandığında paketlerin depolama düğümlerine yerleştirilmesi elde edilir.

Uygulanan matematiksel modelde öncelikle istenen sayıda paket elde edebilmek için MDS kodlama yapılmaktadır. Kodlanmış paketlerin sayısı θ olsun. Bu kodlanmış paketlerin her birinin ρ tane kopyası n tane depolama düğümünde ve her bir düğüm α paket içerecek biçimde depolanmaktadır. Eğer bir depolama düğümü erişim dışı kalırsa, sistem bu düğümü erişilebilir olan d tane düğüm noktasının her birinden β paket indirerek onarmaktadır. Örneklerde $\beta = 1$ durumuna yer verilmiştir.

$[n]$ ile $\{1, 2, \dots, n\}$ kümesi gösterilsin.

Tanım 4.1.2 (β -onarılabilirlik) Her $i = 1, \dots, d$ için $V_i \subset \Omega = [\theta]$ olsun. Eğer $d\beta$ elemanlı bir $A \subset \Omega$ kümesi için;

1. Her $i = 1, \dots, d$ için $B_i \subset A, |B_i| = \beta$ ve $B_i \subseteq V_i$

2. $\bigcup_{i=1}^d B_i = A$

şartlarını sağlayan B_i 'ler bulunabiliyorsa A kümesine $V = \{V_1, \dots, V_d\}$ kümesi tarafından β -onarılabilir denir.

4.2 Kesirli Tekrarlama Kodları

Tanım 4.2.1 (FR Kodlar) $V = \{V_1, \dots, V_n\}$ kümesi $\Omega = [\theta]$ kümesinin altkümelerinin bir kümesi olsun. Eğer $C = (\Omega, V)$ ikilisi

(a) $|V_i| = \alpha$.

(b) Ω kümesinin her bir elemanı tam olarak V kümesinin ρ elemanı tarafından içerilir.

- (c) V^+ ile V kümesinin herhangi bir $(n - \delta)$ elemanlı altkümesini gösterelim. $V^- = V \setminus V^+$ olarak tanımlansın. Her $V_j \in V^-$ kümesi V^+ kümesinin d -elemanlı bir alt kümesi tarafından β -onarılabilir.

özelliklerini sağlıyor ise $C = (\Omega, V)$ ikilisine *Kesirli Tekrarlama* (FR) kodu adı verilir.

Yukarıdaki kesirli tekrarlar kodu tekrarlar derecesi ρ ve tamir bant genişliği β olan (n, k, d, α) parametrelili dağıtılmış depolama sistemlerinin tasarımında kullanılmıştır. Ayrıca sistem mimarisinde Tanım 4.2.1 de yer alan (c) maddesi önemli bir yere sahiptir. Bu aşağıdaki örnek ile açıklanacaktır.

Örnek 4.2.1 $\Omega = \{1, 2, 3, 4, 5, 6\}$ kümesini ve bu kümenin alt kümelerinin kümesi olan

$$V = \{\{1, 2, 3\}, \{2, 3, 4\}, \{4, 5, 6\}, \{1, 5, 6\}\}$$

ve

$$W = \{\{1, 2, 3\}, \{3, 4, 5\}, \{2, 5, 6\}, \{1, 4, 6\}\}$$

kümelerini ele alalım. Burada hem V hem de W Tanım 4.2.1 de verilen (a) ve (b) maddelerini sağlamaktadırlar. $\{1, 2, 3\} \cap \{4, 5, 6\} = \emptyset$ olduğu için $\{1, 2, 3\}$ kümesi V kümesi tarafından 1-onarılabilir değildir. Sonuç olarak $C = (\Omega, V)$ ikilisi bir kesirli tekrarlar kodu olamaz. Fakat $C = (\Omega, W)$ ikilisi bir kesirli tekrarlar kodudur.

(c) maddesinde yer alan δ parametresi tasarlanan sistemlerin bir başka önemli parametresidir. ρ_{res} ile (c) maddesinin sağlanmasını olası kılan en büyük δ değerini gösterebiliriz. Tasarlanan dağıtılmış depolama sistemi ρ_{res} adet depolama düğümünde veri kaybı yaşanmasına rağmen sorunsuz bir biçimde çalışmaya devam edecektir. Dağıtılmış depolama sistemlerinde dosya boyutu ölçülmesi önem arz eden bir metriktir. Bu metriğin belirlenmesi incelenen matematiksel modeldeki en zorlayıcı problemdir.

Dosya Boyutu

$$\mathcal{M} = \min_{I \subset [n], |I|=k} \left| \bigcup_{i \in I} V_i \right| \quad \text{ve}$$

Kod Oranı

$$R_C = \frac{\mathcal{M}}{n\alpha}.$$

olarak tanımlanır. $\beta = 1$ and $k \leq d$ parametreleri ile verilen kesirli tekrarlar kodları (Rouayheb ve Ramchandran 2010) tarafından ele alınmıştır. Bu çalışmadaki kodlar (c) maddesini sağlamakta ve $\rho_{res} = \rho - 1$ dir. Fakat $\beta > 1$ ve $k > d$ durumlarında çalışabilen kodlar üzerine bir çalışma yapılmamıştır.

Tanım 4.2.2 (FR Kodların Oluşum Matrisi) $\Omega = [\theta]$ and $V = \{V_1, V_2, \dots, V_n\}$ olmak üzere, aşağıdaki kuralla tanımlanan $\theta \times n$ tipindeki $N = (n_{i,j})$ matrisine

$$N_{i,j} = \begin{cases} 1, & i \in V_j; \\ 0, & i \notin V_j. \end{cases}$$

$\mathcal{C} = (\Omega, V)$ kesirli tekrarlar kodunun bir oluşum matrisi denir.

Tanım 4.2.3 (Dual FR Kod) C bir kesirli tekrarlar kodu ve N matrisi bu kodun oluşum matrisi olsun. N^T (N matrisinin transpozu) oluşum matrisinden elde edilen C^T kesirli tekrarlar koduna C kodunun duali adı verilir.

Örnek 4.1.1'de yapılan dış MDS kodlamasının oranı 9/10 dur. Bu örnekte verilen dağıtılmış depolama sistemi sadece bir depolama düğümünde veri kaybı yaşanması durumunda onarım işlemleri gerçekleştirebilmektedir. Fakat elde edilen depolama sistemi iki düğümde erişim problemi yaşanması durumunda bile dosyayı kullanıcıların erişimine sunmaktadır. d_{\min} tane depolama düğümünde aynı anda veri kaybı yaşanması durumunda dosya bir kullanıcı tarafından oluşturulamaz.

Tanım 4.2.4 (En kısa uzaklık) Bir dağıtılmış depolama sisteminin en kısa uzaklığı (d_{\min}) dosyanın bir kullanıcı tarafından oluşturulamayacağını garanti eden de-

polama düğümlerinin erişim sağlanamayan en küçük elemanlı altkümesinin kardinalitesi olarak tanımlanır.

(Kamath vd. 2014) referansında bu metrik için bir üst sınır verilmiştir.

Lemma 4.2.1 (Singleton üst sınırı) M boyutlu bir dosyanın saklandığı en kısa uzaklığı d_{\min} olan (n, k, d, α) parametrelili bir dağıtılmış depolama sistemi aşağıdaki eşitsizliği sağlar.

$$d_{\min} \leq n - \left\lceil \frac{M}{\alpha} \right\rceil + 1.$$

$\alpha > 1$ ve $k > d$ durumunda (Gopalan vd. 2012) ve (Papailiopoulos ve Dimakis 2012) referanslarında en kısa uzaklık için aşağıdaki üst sınır verilmiştir.

Lemma 4.2.2

$$d_{\min} \leq n - \left\lceil \frac{M}{\alpha} \right\rceil - \left\lceil \frac{M}{d\alpha} \right\rceil + 2.$$

- Eğer $d \geq k$ eşitsizliği sağlanırsa $\left\lceil \frac{M}{d\alpha} \right\rceil = 1$ olur. Bu durumda Lemma 4.2.2 verilen üst sınır Singleton üst sınırın aynısı olur.
- Eğer $k = \left\lceil \frac{M}{\alpha} \right\rceil$ ise sistem Singleton üst sınırını sağlar.
- Benzer biçimde, eğer $k = \left\lceil \frac{M}{\alpha} \right\rceil + \left\lceil \frac{M}{d\alpha} \right\rceil - 1$ eşitsizliği sağlanırsa sistem Lemma 4.2.2 de verilen üst sınırı sağlar.

Dosya boyutu hesaplamalarında aşağıdaki temel ilkeye sıklıkla başvurulacaktır.

Lemma 4.2.3 A_1, A_2, \dots, A_n kümeleri verilsin. $\mathcal{I} \subseteq [n]$ olmak üzere $A_{\mathcal{I}} = \bigcap_{j \in \mathcal{I}} A_j$ şeklinde tanımlansın. Bu durumda

$$|A_1 \cup A_2 \cup \dots \cup A_n| = \sum_{\emptyset \neq \mathcal{I} \subseteq [n]} (-1)^{|\mathcal{I}|+1} |A_{\mathcal{I}}|$$

eşitliği sağlanır.

Yukarıdaki eşitlikten aşağıdaki eşitsizlik elde edilebilir.

$$|A_1 \cup A_2 \cup \dots \cup A_n| \geq \sum_{i=1}^n |A_i| - \sum_{i < j} |A_i \cap A_j| \quad (4.1)$$

Kesirli tekrarlar kodlarının mimarisinin inşa edilmesinde temel olarak kombinatoriyal tasarımlar teorisine başvurulmuştur.

(v, b, r, k, λ) -tasarımının parametreleri bu bölümden itibaren sırasıyla $(\theta, n, \rho, \alpha, \lambda)$ parametreleri olarak alınacaktır.

4.3 Kesirli Tekrarlar Kodlarının Dağıtılmış Depolama Sistemlerine Bir Uygulaması

Bir $(\theta, \rho, \alpha, \lambda)$ -tasarımı β -onarılabilirlik özelliğine sahipse dağıtılmış depolama sistemlerinin sistem mimarisinde kesirli tekrarlar kodu olarak kullanılabilir. Örneğin, $\beta = 1$ olduğu durumda sonlu projektif düzlemler sistem mimarisinde kullanılabilir.

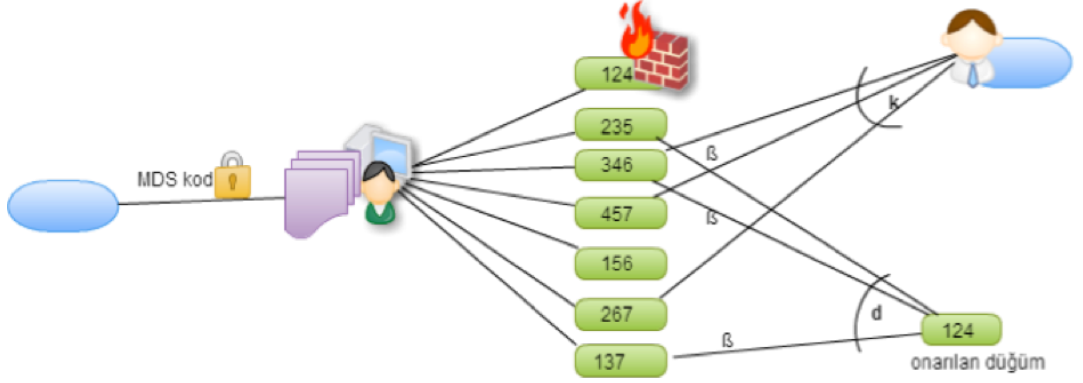
Projektif düzlemlerin sahip olduğu geometrik özellikler dosya boyutu hesaplamalarının sade ve basit olarak ele alınmasına olanak vermektedir. Bu geometrik özellikler:

1. birbirinden farklı iki noktayı içeren tek blok vardır
2. herhangi iki farklı blok tek bir noktada kesişir

olarak sıralanabilir.

Fano düzleminin bloklarından ve noktalarından yararlanarak şekil 4.3'teki depolama sistemi elde edilebilir.

- Tasarlanan sistem 1-onarılabilir.



Şekil 4.3 Bu şekilde $(n, k, d, \alpha) = (7, 3, 3, 3)$ -dağıtılmış depolama sistemi verilmiştir.

- Depolama düğümlerinden birisinde yaşanan erişim problemi; üç tane erişim sağlanabilen depolama düğümünün her birinden birer paket indirilerek giderilebilir.
- Dağıtılmış depolama sistemimizde bazı parametreler açık bir şekilde gözlemlenebilir. Bu parametreler $\alpha = 3$, $d = 3$, $\beta = 1$ ve $n = 7$ dir.
- Sistem ihtiyaçlarına göre k parametresi sistem mimarisinin tasarımcısı tarafından belirlenecektir. k parametresinin değeri dosya boyutu ve uygulanması gereken dış MDS kodlama ile yakından ilişkilidir.
- Örneğin sistemde bir kullanıcı herhangi $k = 3$ tane depolama düğümüne erişerek dosyayı oluşturabilir. Öncelikle dosyanın kaç tane paketten oluştuğunun yani dosya boyutunun belirlenmesi gerekmektedir. $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4, \mathcal{B}_5, \mathcal{B}_6$ ve \mathcal{B}_7 depolama düğümleri olmak üzere \mathcal{M} değeri

$$\mathcal{M} = \min_{I \subset [n], |I|=3} \left| \bigcup_{i \in I} \mathcal{B}_i \right|$$

eşitliği yardımıyla hesaplanır.

- Herhangi $k = 3$ düğüm noktasına erişen kullanıcı en az $9 - \binom{3}{2} = 6$ kodlanmış pakete ulaşabilecektir. Bundan dolayı dosyanın 6 pakete bölünmesi ve sisteme $(7, 6)$ -MDS ile başlanması gerekmektedir.

Çizelge 4.1 $k = 7$ için yapılan dosya boyutu hesaplamasında $\mathcal{M} = 27$ 'dir.

{0, 1, 2}	{0, 3, 4}	{0, 5, 6}	{0, 8, 7}	{0, 9, 10}
{0, 11, 12}	{0, 13, 14}	{1, 3, 5}	{1, 4, 7}	{8, 1, 6}
{1, 11, 9}	{1, 10, 13}	{1, 12, 14}	{9, 2, 3}	{2, 4, 6}
{2, 10, 5}	{2, 14, 7}	{8, 2, 12}	{2, 11, 13}	{3, 11, 6}
{3, 12, 7}	{8, 3, 13}	{10, 3, 14}	{4, 5, 13}	{8, 9, 4}
{4, 10, 12}	{11, 4, 14}	{11, 5, 7}	{8, 5, 14}	{9, 12, 5}
{10, 6, 7}	{9, 6, 14}	{12, 13, 6}	{9, 13, 7}	{8, 10, 11}

Çizelge 4.2 $k = 7$ için yapılan dosya boyutu hesaplamasında $\mathcal{M} = 28$ 'dir.

{1, 11, 6}	{1, 2, 5}	{2, 3, 6}	{9, 5, 6}	{3, 11, 5}
{7, 13, 5}	{11, 4, 13}	{2, 12, 7}	{3, 4, 7}	{8, 4, 5}
{8, 11, 7}	{4, 12, 6}	{8, 6, 14}	{12, 5, 14}	{8, 3, 13}
{8, 9, 12}	{9, 10, 13}	{1, 12, 13}	{0, 9, 7}	{9, 2, 11}
{0, 8, 2}	{9, 4, 14}	{10, 11, 14}	{0, 11, 12}	{0, 3, 14}
{8, 1, 10}	{10, 3, 12}	{1, 3, 9}	{0, 10, 5}	{0, 13, 6}
{1, 14, 7}	{2, 4, 10}	{2, 13, 14}	{0, 1, 4}	{10, 6, 7}

Steiner sistemler dağıtılmış depolama sistemlerinde kullanılabilirler. Bu durumda sistem parametreleri:

$$\rho = \frac{\theta - 1}{\alpha - 1}, \quad n = \frac{(\theta - 1)\theta}{(\alpha - 1)\alpha}, \quad d = \alpha \text{ ve } \beta = 1$$

şeklinde hesaplanır.

Bilgisayar ile yapılan gözlemlerde dosya boyutunun kesirli tekrarlar kodlarının parametrelerinden bağımsız olup, sistemin oluşturulmasında kullanılan kombinatoriyal tasarımların özelliklerinden etkilendiği ortaya konulmuştur. Bunu birbirine denk olmayan iki adet Kirkmann öğrenci probleminin çözümünü veren $(15, 3, 1)$ Steiner üçlü sisteminde gözlemleyebiliriz.

$k = 7$ için yapılan dosya boyutu hesaplamalarında $\mathcal{M}_1 = 27$ ve $\mathcal{M}_2 = 28$ olduğu gözlemlenir. Bu tasarımlar sayesinde Steiner sistemlerden elde edilemeyen parametrelerdeki kesirli tekrarlar da elde edilmiştir.

Çözülebilir tasarımdan elde edilen kesirli tekrarlar koduna çözülebilir kesirli tekrarlar kodu denir. Çözülebilir tasarımların depolama sisteminde kullanılmasının Steiner üçlü sistemlerin kullanılmasından daha avantajlı olduğu durumlar vardır. Örneğin bir paralel sınıftaki depolama düğümlerinin her birinde aynı anda yaşanabilecek erişim problemi, en az bir tane paralel sınıfın erişilebilir olduğu durumda kolaylıkla çözülebilir. Çözülebilir olmayan Steiner üçlü sistemleri aynı anda yaşanan çoklu erişim problemlerini çözmekte yetersiz kalacaklardır.

Örnek 4.3.1 Çözülebilir tasarımlara aşağıda blokları belirtilen tasarım örnek verilebilir.

{1, 2, 3, 4}	{5, 6, 7, 8}	{9, 10, 11, 12}	{13, 14, 15, 16}
{1, 5, 9, 13}	{2, 6, 10, 14}	{3, 7, 11, 15}	{4, 8, 12, 16}
{1, 6, 11, 16}	{2, 5, 12, 15}	{3, 8, 9, 14}	{4, 7, 10, 13}
{1, 7, 12, 14}	{2, 8, 11, 13}	{3, 5, 10, 16}	{4, 6, 9, 15}

Tanım 4.3.1 (Kronecker Çarpım) A ; $m \times r$ ve B ; $p \times q$ tipinde matrisler olsun. A matrisi ile B matrisinin *Kronecker çarpımı* $A \otimes B$ ile gösterilip aşağıdaki biçimde tanımlıdır.

$$\begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1r}B \\ a_{21}B & a_{22}B & \cdots & a_{2r}B \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mr}B \end{pmatrix}.$$

Sırasıyla $(n_1, \theta_1, \alpha_1, \rho_1)$ ve $(n_2, \theta_2, \alpha_2, \rho_2)$ parametreleri ile verilen $\mathcal{C}_1 = (\Omega_1, V_1)$ ve $\mathcal{C}_2 = (\Omega_2, V_2)$ kesirli tekrarlar kodlarının Kronecker çarpımları da bir kesirli tekrarlar kodudur.

Tanım 4.3.2 (Lokal Kesirli Tekrarlama Kodu) $\mathcal{C} = (\Omega, V)$ kodu (n, k, d, α) parametrelili bir dağıtılmış depolama sisteminin sistem mimarisinde kullanılan bir kesirli tekrarlama kodu olsun. Eğer $d < k$ eşitsizliği sağlanıyor ise \mathcal{C} kesirli tekrarlama koduna lokal kesirli tekrarlama kodu adı verilir.

Tanım 4.3.3 (Lokal Altyapı) $\mathcal{C} = (\Omega, V)$ bir lokal kesirli tekrarlama kodu ve $V_i \in V$ olsun. $S(V_i) \in V$ ile V_i depolama düğümüne erişim sağlanamadığında sistemin bağlantı kurduğu depolama düğümlerinin kümesini gösterelim. $S(V_i)$ kümesine V_i depolama düğümünün lokal altyapısı adı verilir.

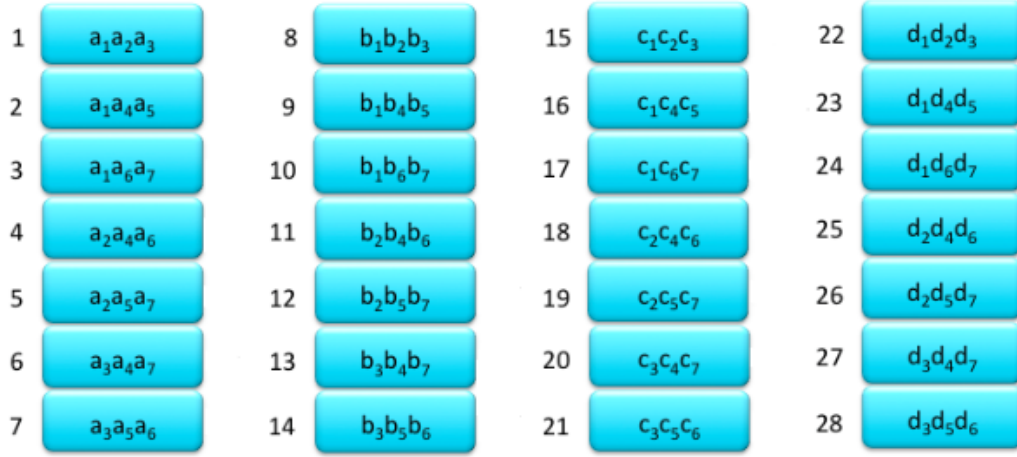
Örnek 4.3.2 (Kronecker çarpımdan elde edilen bir lokal kod) $\Omega = \{1, 2, 3\}$ ve $V = \{V_1 = \{1, 2\}, V_2 = \{2, 3\}, V_3 = \{1, 3\}\}$ olmak üzere $\mathcal{C} = (\Omega, V)$ kodu bu ikiliden elde edilen kesirli tekrarlama kodu olsun. N matrisi bu kodun ölçüm matrisi olmak üzere 3×3 birim matris ile N matrisinin Kronecker çarpımı $\tilde{N} = I \otimes N$ matrisinden elde edilen kesirli tekrarlama kodu lokal bir kod olarak kullanılabilir. Depolama sistemini oluştururken öncelikle bir $(9, 5)$ MDS kod uygulanır. Bu sistemde herhangi $k = 4$ depolama düğümüne erişen bir kullanıcı dosyayı oluşturabilecektir.



Şekil 4.4 Depolanan kodlanmış paketler

Bir depolama düğümünde yaşanan erişim problemi $d = 2 < k$ tane depolama düğümünün her birinden $\beta = 1$ paket indirilerek yapılabilir. Bu kod en fazla 5 adet depolama düğümünde aynı anda erişim problemi yaşanmasında bile dosyanın sorunsuz olarak oluşturulmasını sağlamaktadır. Bu yüzden d_{\min} parametresi Lemma 4.2.2 de verilen üst sınıra ulaşmaktadır.

Örnek 4.3.3 Şekil 4.5'te her lokal altyapısı Fano düzlemi olan bir lokal kod örneği verilmiştir. Buradaki lokal kod 17 paketten oluşan bir dosya için tasarlanmıştır. Herhangi $k = 15$ depolama ulaşan bir kullanıcı bu dosyayı oluşturabilir. Elde edilen lokal kesirli tekrarlı kodunun parametreleri $n = \theta = 28$, $\alpha = 3$, $\rho = 3$ ve $d = 3 < k$ olarak hesaplanır.



Şekil 4.5 $n = 28, k = 15, r = 3, \theta = 28, \alpha = 3, \rho = 3$ parametrelili bir lokal kesirli tekrarlı kodu.

5. TARTIŞMA VE SONUÇ

Kombinatoryal tasarımlar kesirli tekrarlar kodlarının oluşturulmasında kullanılmıştır. İncelenen modelde; bir depolama düğümü zarar gördüğünde bu depolama düğümünü tamir etmek için erişilebilir d depolama düğümünün her birinden β boyutunda paket indirilmektedir. Önerilen bu modelin işlevsel olabilmesi için kesirli tekrarlar kodu için kullanılan tasarım β -onarılabilirlik özelliğine sahip olmalıdır.

Dosya boyutunun hesaplanması önemli ve zor bir problemdir. Cebirsel hesaplamalarda etkili olan programlama dillerinde geliştirilen algoritmalarla yararlanarak dosya boyutu hesaplamaları yapılmıştır. Bu algoritmalar gelecek çalışmalarımızda kullanılacak bilgi birikiminin önemli bir parçasını oluşturmaktadır. Ayrıca kullanılan algoritmalar, dosya boyutunun dağıtılmış depolama sistemlerinin (n, k, α, d) ile verilen parametre değerlerinden bağımsız olduğunun anlaşılması konusunda öncü olmuştur. İleride yapılacak olan çalışmalarda tasarımların dosya boyutunu etkileyen özelliklerinin anlaşılması ve listelenmesi üzerine bilgi birikimi oluşturulmuştur. (Olmez ve Ramamoorthy 2016) çalışmalarında Steiner sistemler için yapılan dosya boyutu hesaplamaları çizelge 5.1-5.2'de verilmiştir.

Çizelge 5.1 $d \geq k$ ve $\beta = 1$ şartlarını sağlayan sonuçlar

Kullanılan Metot	$(n, \theta, \alpha, \rho)$	\mathcal{M}	k
Steiner Sistemler	$(n, \theta, \alpha, \frac{\theta-1}{\alpha-1})$	$\geq k\alpha - \binom{k}{2}$	$1 \leq k \leq \alpha$
Dual Steiner Sistemler	$(n, \theta, \frac{\theta-1}{\rho-1}, \rho)$	$\geq k\alpha - \binom{k}{2}$	$1 \leq k \leq \alpha$

Çizelge 5.2 Kronecker çarpımdan elde edilen sonuçlar

Baz alınan kod	Uygulanan Metot	$(n, \theta, \alpha, \rho)$	β	\mathcal{M}	k
Dual Steiner Sistemler	\mathcal{C}^T ile kendisinin Kronecker çarpımı	$(\tilde{n}^2, \tilde{\theta}^2, \tilde{\alpha}^2, \tilde{\rho}^2)$	$\tilde{\alpha}$	$k\tilde{\rho}^2 - \tilde{\rho}\binom{k}{2}$	$1 \leq k \leq \tilde{\rho}$

Kombinatoryal tasarımların özelliklerinin araştırılması dağıtılmış depolama sistemlerinin sistem mimarisini tasarlamak için önem arz etmektedir. Fakat elde edilen depolama sistemlerinin parametrik özelliklerini tasarımların parametrelerinden yararlanarak önceden tahmin etmek zor bir olgudur. Örneğin bu olgu dosya boyutu hesaplamalarında ortaya çıkan zorlu bir araştırma problemidir. Tezde gerçekleştirilen çalışmalar bu alandaki problemlerin daha iyi anlaşılmasına ve bu konuda ileride yürütülecek araştırmaların şekillenmesinde faydalı olmuştur.

KAYNAKLAR

- Colbournand, C. J. and Dinitz J.H. 2010. Handbook of Combinatorial Designs CRC press.
- Dimakis, A. G., Godfrey, P.B., Wu, Y., Wainwright, M. J. and Ramchandran, K. 2010.
- El Rouayheb, S. and Ramchandran, K. 2010. Fractional repetition codes for repair in distributed storage systems, In Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on (pp. 1510-1517). IEEE.
- Ernvall, T. 2012. The existence of fractional repetition codes, arXiv:1201.3547.
from combinatorial designs. IEEE Transactions on Information Theory, 62(4), 1565-1591.
- Gopalan, P. Huang, C. Simitci, H. and Yekhanin, S. 2012. On the locality of codeword
- Hu, Y., Lee, P. P. and Shum, K.W. 2013. Analysis and construction of functional regenerating codes with uncoded repair for distributed storage systems, In INFOCOM, 2013 Proceedings IEEE (pp. 2355-2363). IEEE.
interference alignment. Information Theory, IEEE Transactions on, 57(3), 1425-1442.
- Kamath, G.M., Prakash, N., Lalitha, V. and Kumar, P.V. 2014. Codes with local regeneration and erasure correction. Information Theory, IEEE Transactions on, 60(8), 4637-4660.
- Kamath, G.M.,N. Silberstein, N. Prakash, A.S. Rawat, V. Lalitha, Koyluoglu, O.O. Kumar, P.V. and Vishwanath, S. 2013. Explicit MBR all symbol locality codes, in IEEE Intl. Symposium on Info. Th. 504–508.
- Koo, J. C. and Gill, J.T. 2011. Scalable constructions of fractional repetition codes in distributed storage systems, In Communication, Control, and Computing Allerton, 2011 49th Annual Allerton Conference on (pp. 1366-1373). IEEE.
- Michelsoni, R., Marelli, A. and Eshghi, K. 2012. Inside solid state drives (SSDs) (Vol. 37). Springer Science and Business Media.
- Network coding for distributed storage systems. IEEE Transactions on Information Theory 56(9), 4539-4551.
- Oggier, F. and Datta, A. 2011. Self-repairing homomorphic codes for distributed storage systems. In INFOCOM, 2011 Proceedings IEEE (pp. 1215-1223). IEEE.
- Ölmez O. 2013. Symmetric Designs and Difference Sets, Journal of Combinatorial Designs, doi: 10.1002/jcd.21354.
- Ölmez O. and Ramamoorthy, A. 2012. Repairable replication-based storage systems using resolvable designs. In Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on (pp. 1174-1181). IEEE.
- Ölmez, O. and Ramamoorthy, A. 2013. Replication based storage systems with local repair. In Network Coding 2013 International Symposium on (pp. 1-6). IEEE.

- Ölmez, O. and Ramamoorthy, A. 2016. Fractional repetition codes with flexible repair
- Papailiopoulos, D. S., Luo, J., Dimakis, A. G., Huang, C. and Li, J. 2012. Simple regenerating codes: Network coding for cloud storage. In INFOCOM, 2012 Proceedings IEEE (pp. 2801-2805). IEEE.
- Papailiopoulos, D.S. and Dimakis, A.G. 2012. Locally repairable codes, IEEE Int. Symposium on Info. Th., 2012, pp. 2771 –2775.
- Rashmi, K.V., Shah, N.B. and Kumar, P.V. 2011. Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction. Information Theory, IEEE Transactions on, 57(8), 5227-5239.
- Rashmi, K.V., Shah, N.B., Kumar, P.V. and Ramchandran, K. 2009. Explicit construction of optimal exact regenerating codes for distributed storage. In Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on (pp. 1243-1249). IEEE.
- Shah, N.B., Rashmi, K.V., Kumar, P.V. and Ramchandran, K. 2012. Distributed storage codes with repair-by-transfer and nonachievability of interior points on the storage-bandwidth tradeoff". Information Theory, IEEE Transactions on, 58(3), 1837-1852.
- Shah, N.B., Rashmi, K.V., Kumar, P.V. and Ramchandran, K. 2012. Interference alignment in regenerating codes for distributed storage: Necessity and code constructions. Information Theory, IEEE Transactions on, 58(4), 2134-2158.
- Shum, K.W. and Hu, Y. 2012. Functional-repair-by-transfer regenerating codes, In Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on (pp. 1192-1196). IEEE.
- Silberstein, N. and Etzion, T. 2014. Optimal fractional repetition codes based on graphs and designs. arXiv preprint arXiv:1401.4734.
- Srijan, A., Gupta M.K. and Gulliver, T.A. 2013. Enumerating Some Fractional Repetition Codes arXiv preprint arXiv:1303.6801.
- Suh, C. and Ramchandran, K. 2011. Exact-repair MDS code construction using interference alignment. Information Theory, IEEE Transactions on, 57(3), 1425-1442.
- Suh, C. and Ramchandran, K. 2011. Exact-repair MDS code construction using symbols, IEEE Trans. on Info. Th., vol. 58, no. 11, pp.6925–6934
- Tamo, I., Wang, Z. and Bruck, J. 2011. MDS array codes with optimal rebuilding, In Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on (pp. 1240-1244). IEEE. ISO 690
- Tian, C., Aggarwal, V. and Vaishampayan, V. A. 2013. Exact-repair regenerating codes via layered erasure correction and block designs. In Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on (pp. 1431-1435). IEEE.

ÖZGEÇMİŞ

Adı Soyadı : Ceren KARTAL

Doğum Yeri :Yenimahalle

Doğum Tarihi : 10.03.1991

Medeni Hali : Bekar

Yabancı Dili : İngilizce

Eğitim Durumu

Lise : Pursaklar Ayyıldız Anadolu Lisesi, Ankara (2010)

Lisans : Ankara Üniversitesi Fen Fakültesi Matematik Bölümü (2014)

Yüksek Lisans : Ankara Üniversitesi Fen Bilimleri Enstitüsü Matematik
Anabilim Dalı (Şubat 2015-Ocak 2017)