

**KOMBİNATORİK TASARIMLARDAN ELDE EDİLEN KODLAR
ÜZERİNE BİR ÇALIŞMA**

Fikri GÖKPINAR

**DOKTORA TEZİ
İSTATİSTİK**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**KASIM 2007
ANKARA**

Fikri GÖKPINAR tarafından hazırlanan KOMBİNATORİK TASARIMLARDAN
ELDE EDİLEN KODLAR ÜZERİNE BİR ÇALIŞMA adlı bu tezin Doktora tezi
olarak uygun olduğunu onaylarım.

Prof. Dr. Hülya BAYRAK

.....

Tez Danışmanı, İstatistik Anabilim Dalı

Bu çalışma, jürimiz tarafından oy birliği ile İstatistik Anabilim Dalında Doktora tezi
olarak kabul edilmiştir.

Prof.Dr. Zülal GÜNGÖR

.....

Endüstri Mühendisliği, Gazi Üniversitesi

Prof.Dr. Hülya BAYRAK

.....

İstatistik, Gazi Üniversitesi

Prof.Dr. Fikri ÖZTÜRK

.....

İstatistik, Ankara Üniversitesi

Prof.Dr. Reşat KASAP

.....

İstatistik, Gazi Üniversitesi

Prof.Dr. Hasan BAL

.....

İstatistik, Gazi Üniversitesi

Tarih: 21/11/2007

Bu tez ile G.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu Doktora derecesini
onamıştır.

Prof. Dr. Nermin ERTAN

.....

Fen Bilimleri Enstitüsü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Fikri GÖKPINAR

**KOMBİNATORİK TASARIMLARDAN ELDE EDİLEN KODLAR ÜZERİNE
BİR ÇALIŞMA
(Doktora Tezi)**

Fikri GÖKPINAR

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
Kasım 2007**

ÖZET

Kombinatorik yapıların birçok uygulama alanları vardır. Bunların en önemlilerinden biri kodlama teorisidir. Kodlama teorisi kesikli matematiğin en çok çalışılan sahalarından biri olmuştur. Burada temel amaç biri kanalda bozulan “gelen kod kelimeyi” çözümlemektir. Tezde, bazı kombinatorik yapılar kullanılarak Optimal ve Maksimal Kodların elde edilişi verilmiştir. Bu kodların doğru çözümleme ve çözümleme zamanları bakımından bir karşılaştırılması yapılmış ve dijital fotoğraflar üzerine uygulaması gösterilmiştir. Ayrıca, Maksimal kodlar için geliştirilen alternatif bir çözümleme algoritması verilmiştir. Son olarak, çok değişkenli dağılımlardan istatistiksel olarak sayı üretilmesi üzerinde durulmuştur.

Bilim Kodu : 205.1.066
Anahtar Kelimeler : Hadamard matrisleri, Çözülebilir tasarımlar, Optimal Kod, Maksimal kod, Ortogonal Dizimler
Sayfa Adedi : 135
Tez Yöneticisi : Prof.Dr. Hülya BAYRAK

A STUDY ON CODES OBTAINED FROM COMBINATORICAL DESIGNS
(Ph.D. Thesis)

Fikri GÖKPINAR

GAZİ UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY
December 2007

ABSTRACT

Combinatorial Structures have many application areas. One of the most important of these areas is coding theory. Coding theory is the one of the most popular area of discrete mathematics. In coding theory, main purpose is to decode the defective “coming” in channel. In this thesis, Obtaining of maximal and optimal codes were given using some combinatorial structures. These codes were compared according to true decoding ratio and decoding times and an application were shown on digital photographs. Besides, an alternative decoding algorithm for maximal codes and number generating from multivariate distribution was also given.

Science Code : 205.1.066

**Key Words : Hadamard Matrices, Resolvable Designs, Optimal Code,
Maximal Code, Ortogonal Arrays**

Page Number : 135

Adviser : Prof.Dr. Hülya BAYRAK

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren Hocam Prof. Dr. Hülya BAYRAK'a ve manevi destekleriyle beni hiçbir zaman yalnız bırakmayan anne ve babama teşekkürü bir borç bilirim.

İÇİNDEKİLER

Sayfa

ÖZET	iv
ABSTRACT	v
TEŞEKKÜR	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ	x
ŞEKİLLERİN LİSTESİ	xii
SİMGELER VE KISALTMALAR	xvi
1. GİRİŞ	1
2. TASARIM TEORİSİ	7
2.1. t-tasarımlar	7
2.2. Dengeli Tamamlanmamış Blok Tasarımlar	9
2.3. Çözülebilir Tasarımlar	14
2.4. Hadamard matrisleri ve Hadamard tasarımları	16
2.4.1. Hadamard matrislerinin temel tanımı ve bazı özellikleri	17
2.4.2. Hadamard matrislerinin bazı kuruluş metotları	18
2.4.3. Hadamard tasarımlar	24
2.5. Sonlu Geometriler	25
2.5.1. Projektif geometri	26
2.5.2. Afin geometri	27
2.6. Latin Kareler ve Ortogonal Dizimler	29
2.6.1. Latin kareler	29
2.6.2. Ortogonal dizimleri	31

Sayfa

3. KODLAMA TEORİSİ	34
3.1. En Çok Olabilirlik Çözümlemesi.....	36
3.2. Eşit Uzaklıklı Kodlar	41
3.3. Sabit Ağırlıklı Kodlar	43
4. TASARIMLARDAN ELDE EDİLEN KODLARIN KURULUŞU.....	46
4.1. Hadamard 2-tasarımlar ve Bu Tasarımlara İlişkin Kodlar	46
4.2. Eşit Uzaklıklı Kodlar, Çözümenebilir Tasarımlar ve Afın Geometri.....	49
4.3. Sabit Ağırlıklı Kodlar ve 2-tasarımlar	51
4.4. Projektif Geometri $PG(n,2)$ Kullanılarak Kod Oluşturma ve Çözümleme Algoritması.....	53
5. KARŞILAŞTIRMA	61
6. TASARIMLARDAN ELDE EDİLEN KODLARIN DOĞRU ÇÖZÜMLEME ORANLARI	86
6.1. Kodların Genel Doğru Çözümleme Oranları.....	86
6.2. Hadamard Matrisinden Oluşturulan Maksimal Kodların Doğru Çözümleme Oranları	89
6.3. Afın Çözülebilir Tasarımlardan Elde Edilen Optimal Eşit Ağırlıklı Kodların Doğru Çözümleme Oranları	95
6.4. 2-tasarımlardan Elde Edilen Sabit Ağırlıklı Kodların Doğru Çözümleme Oranları	103
7. HADAMARD MATRİSLERİ İÇİN YENİDEN DÜZENLENMİŞ EN ÇOK OLABİLİRLİK ALGORİTMASI.....	107
7.1. C_1 kodu için Yeniden Düzenlenmiş En Çok Olabilirlik Algoritması.....	108
7.2. C_2 kodu için Yeniden Düzenlenmiş En Çok Olabilirlik Algoritması.....	110
7.3. C_3 kodu için Yeniden Düzenlenmiş En Çok Olabilirlik Algoritması.....	113
8. OPTİMAL KODLARA DAYALI SAYI ÜRETME TEKNİKLERİ.....	116

Sayfa

8.1. Sayı Üretme	117
8.1.1. Optimal kodlardan sayı üretme	117
8.1.2. Latin hiperküplerden sayı üretme.....	118
8.1.3. Optimal koddan elde edilen latin hiperküplerden sayı üretme.....	119
8.2. Çok Değişkenli Normal Dağılım için Olasılık Hesaplaması.....	121
9. SONUÇ	126
KAYNAKLAR	128
ÖZGEÇMİŞ	132

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 4.1. $PG(2,2)$ den türetilen ikili lineer kod	54
Çizelge 4.2. Çözümleme algoritması	55
Çizelge 4.3. $PG(3,2)$ nin hiperdüzlemleri	55
Çizelge 4.4. $PG(3,2)$ nin düzlemleri	56
Çizelge 4.5. $PG(3,2)$ den elde edilen kod	56
Çizelge 5.1. Renk tonları ve bunlara ait sayılar	61
Çizelge 6.1. Çözülebilir 2 tasarımlar ve karşılık gelen kodlar	97
Çizelge 6.2. Eşit uzaklıklı kodların doğru düzeltme olasılığı	101
Çizelge 6.3. 2-tasarımlardan elde edilen sabit ağırlıklı kodlar.....	104
Çizelge 6.4. Sabit ağırlıklı kodların doğru düzeltme olasılığı	106
Çizelge 7.1. Yeniden düzenlenmiş en çok olabilirlik algoritması	107
Çizelge 7.2. H_8 den elde edilen C_1 için yeniden düzenlenmiş en çok olabilirlik algoritması	108
Çizelge 7.3. H_{12} den elde edilen C_1 için yeniden düzenlenmiş en çok olabilirlik algoritması	109
Çizelge 7.4. H_{16} den elde edilen C_1 için yeniden düzenlenmiş en çok olabilirlik algoritması	109
Çizelge 7.5. C_1 için ML ve MML için çözümleme zamanları	110
Çizelge 7.6. H_{12} den elde edilen C_2 için yeniden düzenlenmiş en çok olabilirlik algoritması	111
Çizelge 7.7. H_{16} den elde edilen C_2 için yeniden düzenlenmiş en çok olabilirlik algoritması	112
Çizelge 7.8. C_2 için ML ve MML için çözümleme zamanları	113

Çizelge	Sayfa
Çizelge 7.9. H_8 den elde edilen C_3 için yeniden düzenlenmiş en çok olabilirlik algoritması	113
Çizelge 7.10. H_{12} den elde edilen C_3 için yeniden düzenlenmiş en çok olabilirlik algoritması.....	114
Çizelge 7.11. H_{12} den elde edilen C_3 için yeniden düzenlenmiş en çok olabilirlik algoritması.....	114
Çizelge 7.12. C_3 için ML ve MML için çözümleme zamanları	115

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 1.1. Basit bir iletişim sistemi	3
Şekil 3.1. İkili simetrik kanal	37
Şekil 4.1. PG(2,2)'nin doğruları.....	54
Şekil 4.2. Hata oranı 0.05 iken çözümüleme zamanları.....	59
Şekil 4.3. Hata oranı 0.10 iken çözümüleme zamanları.....	59
Şekil 4.4. Hata oranı 0.15 iken çözümüleme zamanları.....	60
Şekil 5.1. Bir dijital fotoğrafın pikselleri ve bunlara karşılık gelen sayılar	62
Şekil 5.2. 80×60 piksel boyutunda dijital fotoğraf	66
Şekil 5.3. C ₃ kodunu kullanarak çözümlenen 80×60 piksel boyutunda dijital fotoğraflar	66
Şekil 5.4. C ₁ kodunu kullanarak çözümlenen 80×60 piksel boyutunda dijital fotoğraflar.....	67
Şekil 5.5. C ₂ kodunu kullanarak çözümlenen 80×60 piksel boyutunda dijital fotoğraflar.....	67
Şekil 5.6. 120×90 piksel boyutunda dijital fotoğraf	68
Şekil 5.7. C ₃ kodunu kullanarak çözümlenen 120×90 piksel boyutunda dijital fotoğraflar.....	68
Şekil 5.8. C ₁ kodunu kullanarak çözümlenen 120×90 piksel boyutunda dijital fotoğraflar.....	68
Şekil 5.9. C ₂ kodunu kullanarak çözümlenen 120×90 piksel boyutunda dijital fotoğraflar.....	69
Şekil 5.10. 160×120 piksel boyutunda dijital fotoğraf.....	69
Şekil 5.11. C ₃ kodunu kullanarak çözümlenen 160×120 piksel boyutunda dijital fotoğraflar.....	70

Şekil	Sayfa
Şekil 5.12. C ₁ kodunu kullanarak çözümlenen 160×120 piksel boyutunda dijital fotoğraflar.....	70
Şekil 5.13. C ₂ kodunu kullanarak çözümlenen 160×120 piksel boyutunda dijital fotoğraflar.....	71
Şekil 5.14. C ₄ kodunu kullanarak çözümlenen 80×60 piksel boyutunda dijital fotoğraflar.....	72
Şekil 5.15. C ₄ kodunu kullanarak çözümlenen 120×90 piksel boyutunda fotoğraflar.....	72
Şekil 5.16. C ₄ kodunu kullanarak çözümlenen 160×120 piksel boyutunda fotoğraflar.....	72
Şekil 5.17. M=16 iken C ₁ ,C ₂ ve C ₃ için çözümleme zamanlarının karşılaştırılması	73
Şekil 5.18. M=24 iken C ₁ ,C ₂ ve C ₃ için çözümleme zamanlarının karşılaştırılması	74
Şekil 5.19. M=48 iken C ₁ ,C ₂ ve C ₃ için çözümleme zamanlarının karşılaştırılması	74
Şekil 5.20. M=64 iken C ₁ , C ₂ ve C ₃ için çözümleme zamanlarının karşılaştırılması	75
Şekil 5.21. M=16 iken C ₁ ,C ₂ ,C ₃ ve C ₄ için çözümleme zamanlarının karşılaştırılması	76
Şekil 5.22. M=64 iken C ₁ ,C ₂ ,C ₃ ve C ₄ için çözümleme zamanlarının karşılaştırılması	76
Şekil 5.23. C ₄ 'in M=16, 64, 256 iken çözümleme zamanlarının karşılaştırılması....	77
Şekil 5.24. C ₂ 'in M=16, 24, 32, 48, 64 iken çözümleme zamanlarının karşılaştırılması	77
Şekil 5.25. C ₃ 'in M=16, 24, 32, 48, 64 iken çözümleme zamanlarının karşılaştırılması	78
Şekil 5.26. C ₁ 'in M=16, 24, 32, 48, 64 iken çözümleme zamanlarının karşılaştırılması	78
Şekil 5.27. C ₂ 'in M=16, 24, 32, 48 iken doğru çözümleme oranına ilişkin grafik ...	79

Şekil	Sayfa
Şekil 5.28. C3'in M=16, 24, 32, 48 iken doğru çözümleme oranına ilişkin grafik ...	80
Şekil 5.29. C1'in M=16, 24, 32, 48 iken doğru çözümleme oranına ilişkin grafik ...	81
Şekil 5.30. M=16 iken C1, C2 ve C3'ün doğru çözümleme oranına ilişkin grafik ...	82
Şekil 5.31. M=24 iken C1, C2 ve C3'ün doğru çözümleme oranına ilişkin grafik ...	82
Şekil 5.32. M=32 iken C1, C2 ve C3'ün doğru çözümleme oranına ilişkin grafik ...	83
Şekil 5.33. M=48 iken C1, C2 ve C3'ün doğru çözümleme oranına ilişkin grafik ...	83
Şekil 5.34. M=16 iken C1, C2, C3 ve C4'ün doğru çözümleme oranına ilişkin grafik	84
Şekil 5.35. M=64 iken C1, C2, C3 ve C4'ün doğru çözümleme oranına ilişkin grafik	85
Şekil 6.1. Kod kelime uzunluğu 16 iken herhangi bir kod kelimenin doğru çözümleme olasılığını gösteren grafik	87
Şekil 6.2. C ₁ için Kod kelime uzunluğu 15, 79, 199, 999, 4999, 19999 iken herhangi bir kod kelimenin doğru çözümleme olasılığı gösteren grafik....	90
Şekil 6.3. C ₁ için Doğru çözümlenecek vektörlerin oranına ilişkin grafik	91
Şekil 6.4. C ₂ için Doğru çözümlenecek vektörlerin oranına ilişkin grafik	93
Şekil 6.5. C ₃ için Kod kelime uzunluğu 14, 78, 198, 998, 4998, 19998 iken herhangi bir kod kelimenin doğru çözümleme olasılığını gösteren grafik	94
Şekil 6.6. C ₃ için doğru çözümlenecek vektörlerin oranına ilişkin grafik	95
Şekil 6.7. C ₄ kodu için herhangi bir kod kelimenin doğru çözümleme olasılığını gösteren grafik.....	98
Şekil 6.8. C ₅ için herhangi bir kod kelimenin doğru çözümleme olasılığını veren grafik	102
Şekil 6.9. C ₅ için doğru çözümlenecek vektörlerin oranına ilişkin grafik	103
Şekil 6.10. C ₆ için herhangi bir kod kelimenin doğru çözümleme olasılığını gösteren grafik.....	105

Şekil	Sayfa
Şekil 8.1. MC Yöntemiyle iki değişken için alınan 9 çaplı bir örnek.....	116
Şekil 8.2. OK yöntemiyle iki değişken için alınan 9 çaplı bir örnek	118
Şekil 8.3. LH yöntemiyle iki değişken için alınan 9 çaplı bir örnek.....	119
Şekil 8.4. OALH yöntemiyle iki değişken için alınan 9 çaplı bir örnek.....	120
Şekil 8.5. $r=0.1$ için $P(X < x_0)$ olasılığının OK, OKLH, LH ve MC tahminleri.....	124
Şekil 8.6. $r=0.9$ için $P(X < x_0)$ olasılığının OK, OKLH, LH ve MC tahminleri.....	125

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
F_q	q Üzerinde Cisim
$QR(q)$	Kuadratik Artık Kümesi
v	Tasarımdaki Nokta Sayısı
b	Tasarımdaki Blok Sayısı
k	Her Bloktaki Nokta Sayısı
r	Her Noktanın Gözüktüğü Blok Sayısı
λ	Her Nokta Çiftinin Gözüktüğü Blok Sayısı
M	Koddaki Kod kelime Sayısı
n	Kod kelime Uzunluğu
d	Minimum Uzaklık
Kısaltmalar	Açıklama
AG	Afin Geometri
DTBD	Dengeli Tamamlanmamış Blok Tasarımı
ISK	İkili Simetrik Kanal
LH	Latin Hiperküp
MC	Monte Carlo
ML	En Çok Olabilirlik Algoritması
MML	Yeniden Düzenlenmiş En Çok Olabilirlik Alg.
<i>MOLS</i>	İkişerli Ortogonal Latin Kareler
OA	Ortogonal Tasarımlar
OK	Optimal Kod
OKLH	Optimal Koddan elde edilmiş Latin Hiperküp
PG	Projektif Geometri

1. GİRİŞ

Klasik kombinatoriyel matematik (ya da kombinatorik) nesneleri belirli kurallara göre düzenleme problemiyle ilgilenir. Kombinatoriyel problemlerde esas, istenilen kurallara göre bir düzenlemenin yapılıp yapılamayacağıdır. Örnek olarak 10 tane basketbol takımının oynayacağı bir turnuvanın birbirleriyle birer kez oynayacak şekilde düzenlenmesi verilebilir. Bazen probleme özel bir koşul da konulabilir; örneğin, test edilmek istenen 5 araba tekerleği için 5 araba üzerinde hangi şekilde sıralanacağı problemi üzerine fazladan her tekerlek her arabada bir kez yer alacağı gibi bir koşul konulabilir [Comtet,1974].

Bir diğer önemli problem, eğer böyle bir tasarım varsa kaç farklı yoldan yapılabileceğidir. İstenilen kurallar basit ise (örneğin n kişilik sınıftan seçilecek k kişilik bir kurul) düzenlemenin varlığı açık olacağından ilgilenilen durum bütün düzenlemenin kaç tane olacağıdır. Fakat, bazen düzenlemenin varlığı bu kadar açık görünmez. Örneğin, Kirkman'ın okul kızları ya da Euler'in subay problemleri düzenlemenin açıkça görülmediği en önemli kombinatorik problemlerden ikisidir [Graham ve ark., 1995].

En önemli kombinatoriyel tasarım türleri t -tasarımlar, dengeli tamamlanmamış blok tasarımlar, ikişerli ortogonal dengeli tasarımlar, Latin kareler, ortogonal Dizimler ve bunlarla ilişkili olarak Sonlu Geometri, Çizge teorisidir. Tasarım teorisinin temel sorusu verilen parametrelerle tasarımın var olup olmadığıdır. Bununla ilgili var olma sonuçları kadar var olmama sonuçları da mevcuttur. Bugün çalışılan tasarımların birçok türü temel olarak matematiksel bulmacalar ve zeka oyunlarına dayanmaktadır [Merris, 2003].

Sonlu projektif ve afin geometriler eğer boyut en azından üç ise çeşitli yollardan özel tasarımlar gibi yorumlanabilir. O nedenle kombinatoriyel yapıların analizi için olan metodlar bu geometrileri çalışmak için de kullanılabilir. Böyle bir metot, isabet matrislerinin kullanımıdır. İsbet matrisleri ve onların genelleştirmelerinin, tasarımlar ve daha genel yapılar üzerine olan çeşitli araştırmalarda zorunlu olduğu

ispatlanmıştır. Sonlu geometriler, uygulamaların çeşitli alanlarında bulunabilirler. Ayrıca istatistikte de geniş bir uygulama alanı görülür.

Tasarım teorisi lineer cebir, grup, halka, cebir, sayı teorisini araç olarak kullanır. Tasarım teorisinin temel içeriği basit olmasına rağmen, tasarımları kurmakta kullanılan matematik çeşitli ve zengindir.

Kombinatoriyel tasarım teorisinin doğuşu 18.nci Yüzyıla uzanır. 1782’de Euler, 1847’de Kirkman ve 1853’de Steiner’in problemlerine kadar dayanmasına rağmen 20. yüzyılın başına Fisher ve arkadaşlarının 1920’lerdeki deney tasarımı üzerindeki çalışmalarından sonra matematiksel bir disiplin olarak kabul görmüştür.

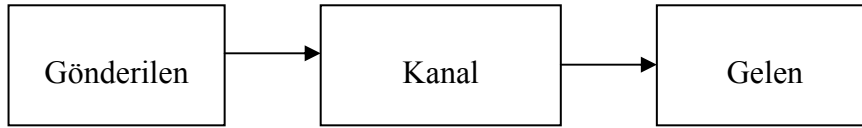
Kombinatoriyel problemler üç ana başlıkta toplanır:

- 1) Klasik Kombinatoriyel Problemler: Seçim Problemleri, Binom Katsayıları, Sıralama Problemleri, Kombinatoriyel Eşitsizlikler v.b.
- 2) Kombinatoriyel Konfigürasyonlar: Blok Tasarımlar, Fark Kümeleri, Latin Kareler, İlişki Matrisleri, Hadamard Matrisleri, Sonlu Geometri v.b.
- 3) Çizge(Graph) Teorisi

Bu konular kendi aralarında ilişkiye sahiptir.

Bu tezin içeriği kombinatoriyel konfigürasyonlarla kesikli matematiğin önemli çalışma alanı olan hata düzeltme kodlarını ihtiva eder. İstatistiksel deney tasarımı da Kombinatoriyel Konfigürasyonların bir uygulama alanıdır.

Kodlama teorisi, 1940’ların sonunda ortaya çıkmasıyla birlikte kesikli matematiğin en önemli konularından olmuştur ve birçok alanda kullanılmaktadır. En önemli uygulama alanı iletişimidir. Şekil 1.1 basit bir iletişim sistemini vermektedir.



Şekil 1.1. Basit bir iletişim sistemi

Burada gönderilen resim, yazı yada ses olabilir. Kanal ise radyo bağlantısı, telefon hattı yada fiber optik kablo olabilir. Kodlama teorisin temel amaçlarından biri kanalda bozulan “geleni” çözümlenektir.

Kodlama teorisinin temelinde “kirli bir kanaldan daha iyi nasıl veri iletilir?” problemi bulunur. Bu problemin incelenmesi sonucu iletişim ve bilgisayar teknolojilerinde birçok ilerleme kaydedilmiştir. En önemli uygulamalarından biri CD’lerin çizik ve tozlarla bozulmuş olabilecek okuma bozukluklardan en az etkilenmesini sağlamaktır. Buradaki çizik ve tozlar, hata diye adlandırılır. Bununla ilgili telefon ve radyolardaki cızırtı, dijital fotoğraflardaki bozukluklar, örnek olarak verilebilir.

İlk ve en önemli kodlardan biri Morse kodudur. Bu kodda sembol olarak sadece nokta ve çizgiler vardır.

Kodlama teorisinin temel mantığını anlamak için küçük bir örnek düşünölsün. 0 ve 1lerden oluşan bir alfabe mevcut olsun. 00 ve 11 gibi iki kodkekiye dikkate alınsın. Bu kod kelimelerden 11 kirli bir kanal üzerinden gönderildiği düşünölsün. Ve ilk hanede bir hata yapıldığı varsayılınsın. Dolayısıyla alıcıya kod kelime 01 şeklinde gelecektir. Burada gelen kod kelime bir hata yapıldığı bilinmektedir. İşte böyle bir koda hata fark etme kodu (error detecting code) adı verilir. Fakat dikkat edilirse, alıcı hata yapıldığını fark etmekte fakat hatanın hangi sembolde yapıldığını anlayamamaktadır. Bunu önlemek için şöyle bir yol geliştirilebilir. Kodun uzunluğu 2 olarak değil 3 olarak belirlenir. Böylece mevcut 2 kod kelime 000 ve 111 şeklinde olur. Dolayısıyla meydana gelecek bir hatada alıcı kod kelimedeki hatayı sadece fark etmez aynı zamanda düzeltebilir. Örneğin 001 gelen kod kelimesi 000 yakın olduğundan bu kod kelime olarak çözölür.

Kodlama teorisi bu çerçevede daha çok hata düzeltmek için çeşitli sınırlar ve bunları sağlayan kodlar geliştirmiştir.

Kodlama teorisinde iki temel amaç vardır:

- 1) Sabit kod kelime uzunluğu ve kodun minimum uzaklığı biliniyorken kod kelime sayısını değerini maksimize etmek.
- 2) Sabit kod kelime uzunluğu ve kodun kod kelime sayısını biliniyorken minimum uzaklığı değerini maksimize etmek.

Son yıllarda, hızlı bilgisayar varlığıyla kombinatoriyel konfigürasyonun kodlama teorisine uygulanması hız kazanmıştır. Kombinatoriyel konfigürasyonlar optimal hata düzelten Kodlar bulunmasında kullanılırken, kodlama teorisi, çeşitli kombinatoriyel yapıların oluşturulmasında ve analizinde kullanılır.

Bu tezde incelenecek olan maksimal kodların kuruluşunda kullanılan en önemli yapı Hadamard matrisleri ve Hadamard tasarımlarıdır. Hadamard matrislerinin modern iletişim konusunda birçok uygulamaları vardır [Yarlagadda ve Hershey, 1997]. Maksimal hata düzeltme kodlarının Hadamard matrislerini temel alan kuruluş teknikleri Bose ve Shrikande (1959), Plotkin (1960) ve Peterson (1961)e dayanır. Ayrıca Hadamard matrisleri Walsh fonksiyonuna [Walsh, 1923] dayalı olarak sinyal modellemesinde kullanılır [Harmurth, 1960]. Bu fonksiyon kullanılarak sıkıştırılmış resimdeki verinin gönderilmesi ve şifrelenmesi de mümkündür [Jain, 1989]. Hadamard matrisine dayalı maksimal kodlar 1960lı yıllarda uzay araştırmalarında Jüpiter, Satürn, Mars Uranüs, Neptün ve aylarının yüksek çözünürlüklü alınan fotoğraflarının doğru olarak çözümlenmesinde kullanıldı. Bu kodların kullanılmasından önce gönderilen kanalların sinyallerinin oldukça düşük olmasından dolayı sadece alınan fotoğrafların gönderilmesi çok uzun zaman alıyor ayrıca gönderilen fotoğraflar oldukça hatalı oluyordu. Hadamard matrislerinin böyle güçlü kodların kuruluşlarındaki etkisi nedeniyle Hadamard matrislerinin oluşturulmasında birçok teknik geliştirilmiştir [Ang ve Ark. (2003), Baumert ve Hall (1965), Turyn (1972)].

Bir diğ er  nemli kod t r  olan optimal e ituzaklıkl  kodların ilk kurulu u Preece(1967) tarafından verilmi tir. Ayrıca bu kodların   z lebilir tasarımlardan kurulmasını ilk olarak Semakov ve Zinoviev (1968) vermi tir. Daha sonra bu kurulu  tekniğı Tonchev (1988) ve Sinha (1994) tarafından geli tirilmi tir. Sinha ve Mitra (1999)   z lebilir ve i i e dengeli tamamlanmamı  blok tasarımı kullanarak yakla ık optimal kodları vermi lerdir.   z lebilir ve i i e dengeli tamamlanmamı  blok tasarımı kullanarak yakla ık optimal kodların bir diğ er kurulu  Miao ve Kageyama (1999) tarafından verilmi tir.

Diğ er  nemli bir maksimal kod t r  Maksimal Sabit ağırlıkl  kodlardır. Sabit ağırlıkl  kodlarda i in maksimal kod sınırı Johnson (1962) tarafından verilmi tir. Bu kodların kurulu u i in bazı ko ulları saėlayan, dengeli tamamlanmamı  blok tasarımları kullanılmaktadır [Tonchev, 1988]. Daha sonra bu sınırın Optik Ortogonal kodlarla ili kisi ortaya  ıkarılmı tır [Chung ve diğ erleri.,1989]. Bu sınır bu kodlar i in minimum uzaklık $d>1$ durumu i in geli tirilmi tir [Omranı ve Ark., 2004]. Ayrıca Optik Ortogonal kodların Projektif d zlemlerle ili kisi Miyamoto ve Ark. (2004) tarafından ortaya konmu tur. Optimal kodların kurulu larında kullanılan bir diğ er kombinatoriyel yapı da ortogonal dizimlerdir [Tonchev, 2005].

Bu tezin ikinci b l m nde, Hadamard 2- tasarım,   z lebilir tasarım, Ortogonal Latin Kareler, Ortogonal dizimler, Simetrik 2-tasarımlar, Projektif ve Afin Geometri yapıları verilerek bunların  zellikleri incelenmi tir.

    nc  b l m de kodlama teorisindeki temel tanımlar ve temel   z mlleme algoritması olan en  ok olabilirlik algoritması verildikten sonra maksimal kod ve optimal kod, en  ok olabilirlik algoritması ele alınmı  ve bunlara ait sınırlar ve  zellikler incelenmi tir.

D rd nc  b l m de, ikinci b l m de verilen Hadamard 2- tasarım,   z lebilir tasarım, Ortogonal Latin Kareler, Ortogonal dizimler, Simetrik 2-tasarımlar, Projektif ve Afin Geometriden faydalanarak,     nc  b l m de tanıtılan maksimal kod ve optimal kod kodların elde edili i incelenmi tir. Ayrıca en  ok olabilirlik

algoritmasına Perspektif geometri kullanılarak bir çözümleme algoritması geliştirilmiştir.

Beşinci bölümde, Hadamard tasarımlardan ve çözülebilir tasarımlardan elde edilen kodların en çok olabilirlik çözümleme yöntemi kullanarak farklı boyutlardaki dijital fotoğraflara uygulanmış ve belirli hata oranları göz önüne alınarak çözümleme zamanları ve doğru çözümleme oranları karşılaştırılmıştır. Ayrıca ikinci kısmında bu kodlar daha farklı kod kelime sayıları, hata oranları ve çözümleme zamanları için karşılaştırılmış ve hangi durumlarda birbirlerine üstün oldukları belirlenmiştir.

Altıncı bölümde, kodların olasılık yapıları, hata düzeltme oranları genel olarak incelenmiş ve bunlar verilen grafiklerle yorumlanmıştır. Daha sonra optimal ve maksimal kodlar için bu yapılar tekrar ele alınarak incelenmiştir.

Yedinci bölümde, Hadamard tasarımlarından elde edilen maksimal kodlar için en çok olabilirlik algoritması yeniden düzenlenmiştir. Bu algoritma 8, 12 ve 16 mertebeli Hadamard matrislerinden elde edilen Hadamard tasarımlardan elde edilen maksimal kodlar için klasik en çok olabilirlik algoritmasına göre karşılaştırılmıştır ve üstün olduğu durumlar incelenmiştir.

Sekizinci bölümde, Optimal kodlar ve modifikasyonlarını kullanarak çok değişkenli dağılımlardan veri üretme teknikleri incelenmiş ve klasik Monte Carlo yöntemine göre göreceli etkinliği karşılaştırılmıştır.

2. TASARIM TEORİSİ

2.1. t -tasarımlar

t -tasarımları, kombinatoriyel yapılardır. t -tasarımlarında terminoloji kullanımı matematiğin ana görüşüne daha yakındır. Bundan dolayı iyi bir matematik bilgisini gerektirir.

t -tasarımları, tasarım teorisindeki en önemli yapılardır ve gelişmiş bilgisayarlar olmasına rağmen birçok t -tasarımını oluşturmak zordur. Çünkü bütün kabul edilebilir parametreler için hala genel bir formül yoktur.

Bu tasarımlar, aynı zamanda kombinatoriyel optimizasyon problemleridir. Bu nedenle istatistiksel deneyler, network tasarımları gibi uygulamalı problemlerde yapı maliyetlerini minimum yapmada önemlidir. t -tasarımları, 1969'dan beri askeri taktiklerde, bilgisayar çiplerinin tasarımlarında ve özellikle istatistikte (tıp ve ziraat deneylerinde, nükleer araştırmalarda, demoskopide (demoscropy), kalite kontrolünde, ...vb.) sık kullanılmaktadır.

$X=\{X_1, X_2, \dots, X_v\}$ sonlu noktalar kümesi $D=\{B_1, B_2, \dots, B_b\}$ X 'in k -elemanlı altkümelerinin sonlu ailesi olan bloklar kümesi olsun. X 'in t -boyutlu her altkümesi D 'in tam olarak λ bloğunda varsa D 'ye t -(v, k, λ) tasarımı ya da t -tasarım adı verilir. $t=k$ için tasarıma tam (yada trivial) denir. (X, D) çifti arasındaki ilişki (incidence) A matrisiyle gösterildiğinde $A=(a_{ij})_{v \times b}$ aşağıdaki gibi tanımlanır.

$$a_{i,j} = \begin{cases} 1 & x_i \in B_j \\ 0 & x_i \notin B_j \end{cases} \quad (2.1)$$

Burada A 'ya ilişki matrisi adı verilir. İki tasarımın izomorfik olması için birinin ilişki matrisinin bir permütasyonuyla diğer düzenin ilişki matrisinin elde edilebilmesi gerekir.

2.1. Teorem

Her t -tasarım aynı zamanda $s \leq t$ için bir s tasarımıdır. Bu tasarımın parametreleri;

$$\lambda_s = \lambda \frac{(v-s)(v-s-1)\dots(v-t+1)}{(k-s)(k-s-1)\dots(k-t+1)} \quad (2.2)$$

dir [Hall, 1986].

Eş. 2.2'ye dikkat edilirse;

$$\lambda_i = \lambda_{i+1} \frac{(v-i)}{(k-i)} \quad (2.3)$$

olduğu görülür. Ayrıca $\lambda_0=b$ ve $\lambda_t=k$ alındığında;

$$vr=bk \quad (2.4)$$

eşitliği elde edilir.

t -(v,k,λ) tasarımının varlığı için $(k-i)(k-i-1)\dots(k-t+1)$ 'in $(v-i)(v-i-1)\dots(v-t+1)$ ifadesini ($i=0,1,\dots,t-1$) tam bölmesi gerekir.

2.1. Örnek

3-(8,4,1) tasarımına ait bloklar aşağıdaki gibi olsun.

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

Bu durumda ilişki matrisi aşağıda verildiği gibidir.

$$A = \begin{matrix} & B_1 & B_2 & B_3 & B_4 & B_5 & B_6 & B_7 & B_8 & B_9 & B_{10} & B_{11} & B_{12} & B_{13} & B_{14} \\ \begin{matrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \\ X_8 \end{matrix} & \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

İlişki matrisine dikkat edildiğinde, tasarımın aynı zamanda 2-(8,4,3) tasarımı olduğu görülür.

2.2. Dengeli Tamamlanmamış Blok Tasarımlar

Dengeli Tamamlanmamış Blok Tasarımlar (Balanced Incompleted Block Designs-DTBT) t -tasarımların özel bir hali olup ($t=2$) parametreleri v, b, r, k, λ şeklindedir ve 2-tasarımlarlar 2-(v, k, λ) ya da (v, b, r, k, λ) olarak da ifade edilir. DTBT'lar tasarım teorisinin en önemli yapılarındadır. Birçok uygulama alanları bulunmaktadır. Birçok özel hali olmasıyla birlikte geometri, kodlama teorisi ve çizge teorisi ile yakın ilişkisi vardır. Bu tasarımların parametreleri arasında Eş. 2.5'teki ilişki vardır.

$$\left. \begin{array}{l} vr = bk \\ \lambda(v-1) = r(k-1) \end{array} \right\} \quad (2.5)$$

Eş. 2.5'e göre;

$$\left. \begin{array}{l} \lambda(v-1) \equiv 0 \pmod{k-1} \\ \lambda v(v-1) \equiv 0 \pmod{k(k-1)} \end{array} \right\} \quad (2.6)$$

olacağı aşıkardır.

2.2. Örnek

$v=9$, $b=12$, $k=3$, $r=4$, $\lambda=1$ parametrelili bir $2-(9,3,1)$ tasarımının blokları ve ilişki matrisi aşağıdaki gibi olsun.

$$B_1=\{1,2,3\}, B_2=\{4,5,6\}, B_3=\{7,8,9\}, B_4=\{1,4,7\}, B_5=\{2,5,8\}, B_6=\{3,6,9\}, \\ B_7=\{1,5,9\}, B_8=\{2,6,7\}, B_9=\{3,4,8\}, B_{10}=\{1,6,8\}, B_{11}=\{2,4,9\}, B_{12}=\{3,5,7\}$$

$$A = \begin{matrix} & B_1 & B_2 & B_3 & B_4 & B_5 & B_6 & B_7 & B_8 & B_9 & B_{10} & B_{11} & B_{12} \\ \begin{matrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \\ X_8 \\ X_9 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{matrix}$$

$v=9$, $b=12$, $k=3$, $r=4$, $\lambda=1$ olduğundan Eş. 2.5'deki eşitliğinin sağlandığı görülür.

Aşağıda verilen Fisher Eşitsizliği 2-tasarımlar için gerek koşulu vermektedir.

2.2. Teorem (Fisher Eşitsizliği)

Herhangi bir $2-(v,k,\lambda)$ tasarımı için $k < v$ olmak üzere;

$$b \geq v \tag{2.7}$$

eşitsizliği sağlanır [Beth ve Ark., 1999].

$b=v$ (dolayısıyla $k=r$) olduğunda böyle bir tasarıma simetriktir denir.

2.1. Tanım

2-tasarım için aşağıdakiler denktir.

- a) $b=v$
- b) $r=k$
- c) herhangi iki blok λ ortak noktaya sahiptir.
- d) herhangi iki bloğun ortak nokta sayısı sabittir [Cameron ve Lint, 1996].

Aşağıda verilen Bruck-Ryser-Chowala Teoremi (BRC teoremi) herhangi bir 2-tasarımın varlığı için gerek koşulu vermektedir.

2.3. Teorem (Bruck-Ryser-Chowala Teoremi)

Herhangi bir 2- (v, k, λ) 'ın varlığı için aşağıdaki koşullar sağlanmalıdır.

v çift ise $\Rightarrow (k-\lambda)$ tam karedir.

v tek ise $\Rightarrow (x, y, z) \neq 0$ olmak üzere $x^2 = (k - \lambda)y^2 + (-1)^{\frac{v-1}{2}} \lambda z^2$ 'dir [Cameron ve Lint, 1996].

Bu teorem $\lambda=1$ özel durumu için Bruck ve Ryser (1949) tarafından ispatlanmıştır. Genel durum ispatı ise Chowla ve Ryser (1950)'a aittir.

Calderbank (1994) Bruck-Ryser-Chowla teoreminin simetrik olmayan tasarımlara bir genelleştirmesini vermiştir.

2.3. Örnek

$v=16, k=10, \lambda=6$, BRC teoremine göre $k-\lambda=4$ tam kare olduğundan simetrik tasarım mevcuttur Ayrıca yine BRC teoremine göre $v=7, k=3, \lambda=1$ simetrik tasarımı $(x,y,z)=(1,1,1)$ için vardır.

2.4. Teorem

$k \leq v-2$ olacak şekilde (v, k, λ) parametrelili DTBT olsun. Bu durumda $2-(v, v-k, b-2r+\lambda)$ tasarımı vardır [Mathon ve Rosa,1996].

Teorem 2.6yı sağlayan tasarıma orijinal tasarımın dual tasarımı denir.

2.4 Örnek

Örnek 2.2'deki tasarımı tekrar ele alınsın. İlişki matrisinde 1 yerine 0, 0 yerine 1 konursa elde edilen yapı orijinal tasarımın dual tasarımıdır.

$$A' = \begin{matrix} & \begin{matrix} B_1 & B_2 & B_3 & B_4 & B_5 & B_6 & B_7 & B_8 & B_9 & B_{10} & B_{11} & B_{12} \end{matrix} \\ \begin{matrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \\ X_8 \\ X_9 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \end{matrix}$$

olmak üzere yeni ilişki matrisinin parametreleri $v'=9, b'=12, r'=12-4=8, k'=9-3=6, \lambda'=12-2 \times 4+1=5$ elde edilir. Burada A' dual tasarımın ilişki matrisidir.

2.2. Tanım

(X, D) dengeli tamamlanmamış blok tasarımı ve $B_0 \in D$ olan herhangi bir blok olsun. Nokta kümesi B_0 'daki noktalar, blok kümesi ise;

$$\{B \cap B_0 : B \in D, B \neq B_0\}$$

olan tasarıma türetilmiş DTBT denir. Ayrıca nokta kümesi $X - B_0$ 'daki noktalar ve blok kümesi;

$$\{B - B_0 : B \in D, B \neq B_0\}$$

olan tasarıma ise artık DTBT denir [Beth ve Ark., 1999].

2.5. Teorem

$2-(v, k, \lambda)$ parametrelili DTBT dikkate alınsın. Bu tasarımdan elde edilecek türetilmiş DTBT'in parametreleri $2-(k, \lambda, \lambda - 1)$ şeklinde, artık DTBT'in parametreleri ise $2-(v - k, k - \lambda, \lambda)$ şeklindedir [Beth ve Ark., 1999].

2.4. Örnek

$2-(7, 4, 2)$ tasarımına ait ilişki matrisi

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

şeklinde verilsin. Buradan elde edilen türetilmiş tasarımının ilişki matrisi ($B_0=B_1$ için);

$$A' = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

şeklinde olur. A' nin tanımladığı tasarım, 2-(4,2,1) tasarımıdır.

Benzer şekilde artık tasarımının ilişki matrisi ($B_0=B_1$ için);

$$A'' = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

şeklinde olur. A'' nin tanımladığı tasarım ise 2-(3,2,2) tasarımını verir.

2.3. Çözülebilir Tasarımlar

Çözülebilir tasarımlar 2-tasarımların özel bir halidir. En ünlü çözülebilir tasarımlardan biri Kirkman'ın okul kızları problemidir. Bu problemde 15 tane kız öğrencinin haftanın 5 günü 3'erli gruplar halinde okula gitmesini ve her öğrenci çiftinin tam olarak 1 kez beraber okula gidecek bir tasarım oluşturulmak istenmektedir. Bu tasarımda dikkat edilecek nokta 15'in 3'ün katı olmasıdır. Bu koşul; yani tüm öğrenci sayısının gruplardaki öğrenci sayısına bölünebilmesi, bir başa deyişle, nokta sayısı v 'nin blok genişliği k 'ya tam olarak bölünmesidir. Bu noktadan hareketle çözülebilir $2-(v,k,\lambda)$ tasarımlarının temelinde k 'nın v 'yi böldüğü tasarımlar vardır denilebilir. $v|k$ ise Fisher eşitsizliği aşağıdaki gibi genelleştirilebilir.

2.6. Teorem

Bir $2-(v, k, \lambda)$ tasarımı $k < v$ ve $v|k$ ise Fisher eşitsizliği;

$$b \geq v + r - 1$$

şeklinde güçlendirilebilir [Tonchev, 2004].

v, k, λ, r, b parametrelili bir $2-(v, k, \lambda)$ tasarım göz önüne alınsın. Bu tasarımı bloklar $S_0, \dots, S_{r-1}$ şeklinde her birinde $b|r$ blok olacak şekilde ayrılabiliriyorsa ayrıca her nokta S_i 'deki bloklarda tam olarak 1 defa bulunursa S_i gruplarına paralel sınıflar denir. Böyle bir tasarım çözülebilir tasarımdır. Ayrıca farklı paralel sınıflardaki herhangi iki blok tam olarak μ noktada kesişiyorsa böyle bir tasarıma afın çözülebilir tasarım denir.

μ şöyle hesaplanabilir.

$B_{i1}, \dots, B_{in}$ $i = 0, 1, \dots, r-1$ olmak üzere S_i 'nin blokları ve $n = v/k = b/r$ olsun. B_{01} bloğu için e_{ij} ; B_{01} ile B_{ij} arasındaki ortak noktaların sayısını versin ($i = 1, \dots, r-1$ $j = 1, 2, \dots, n$).

B_{01} 'in her noktası $r-1$ tane daha başka bloklarda gözüktür ve her bir $S_1, \dots, S_{r-1}$ 'in elemanıdır. Bu durumda ; $\sum_{i,j} e_{ij} = k(r-1)$ dir. Dolayısıyla

$$\mu = \frac{1}{n(r-1)} \sum e_{ij} = \frac{k}{n} = \frac{k^2}{v}$$

$$\mu = \frac{k^2}{v}$$

şeklindedir.

2.7. Teorem

v, k, λ, r, b parametrelili çözülebilir tasarım $b \geq v+r-1$ koşulunu sağlar. Eğer tasarım afin çözülebilir ise $b=v+r-1$ eşitliğini sağlar [Tonchev, 2005].

2.5. Örnek

Örnek 2.2’de 2-(9,3,1) tasarımı aşağıdaki şekilde paralel sınıflara ayrılabilir.

$$PS(1)=\{B_1=\{1,2,3\}, B_2=\{4,5,6\}, B_3=\{7,8,9\}\}$$

$$PS(2)=\{B_4=\{1,4,7\}, B_5=\{2,5,8\}, B_6=\{3,6,9\}\}$$

$$PS(3)=\{B_7=\{1,5,9\}, B_8=\{2,6,7\}, B_9=\{3,4,8\}\}$$

$$PS(4)=\{B_{10}=\{1,6,8\}, B_{11}=\{2,4,9\}, B_{12}=\{3,5,7\}\}$$

Burada $\mu=3^2/9=1$ ve $b=9+4-1=12$ olduğundan tasarım afin çözülebilirdir denir. Burada PS, paralel sınıfları tanımlanmaktadır.

2.4. Hadamard Matrisleri ve Hadamard Tasarımları

Hadamard matrisleri tasarım teorisinin bir başka önemli yapısıdır. Hadamard matrislerinin Projektif Geometri, Dengeli Tamamlanmamış Blok Tasarımları, ortogonal dizimler ve maksimal kodları gibi birçok yapıyla yakın ilişkilidir. Bu matrisler ilk olarak Sylvester (1867) ve Hadamard (1893) tarafından ortaya konulmuştur.

Bu bölümde birçok alanda uygulaması bulunan Hadamard matrisleri ve kombinatoriyel tasarımlara uygulanışı incelenecektir.

2.4.1. Hadamard matrislerinin temel tanımı ve bazı özellikleri

$n \times n$ boyutlu ve elemanları $+1$ ve -1 'den oluşan bir H matrisi $HH^T = nI$ koşulunu sağlıyorsa, H matrisine *Hadamard matrisi* denir.

2.5. Örnek

Aşağıdaki matrisler 2 ve 4 mertebeli hadamard matrislerine birer örnek olarak verilebilir.

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Hadamard matrisinin tanımından HH^T matrisinin köşegen elemanları n köşegen dışı elemanları ise 0 olduğu düşünülürse Hadamard matrisinin satırlarının ortogonal olduğu söylenebilir. Ayrıca Hadamard matrisinin ilk satır ve sütun elemanları 1 ise böyle bir Hadamard matrisine standard biçim denir. Aşağıdaki teorem, Hadamard matrisinin varlığı için gerekli koşulu verir.

2.8. Teorem

$n > 2$ mertebeli bir Hadamard matrisi var ise bu durumda n , 4 ile tam olarak bölünür [Hall, 1986].

İspat

$H = (h_{ij})$ $n > 2$ mertebeden normalleştirilmiş HM olsun. $1 \leq i \leq n$ için r_i H 'in i .satırını gösterebilir. $i \geq 2$ ise $r_1 \bullet r_i = 0$ olduğu ve r_1 n tane "1" içerdiği için her r_i satırı $n/2$ tane "1" ve $n/2$ tane "-1" içerir $2 \leq i \leq n$. Burada n çifttir. Ayrıca,

$$\begin{aligned}
a &= \left| \left\{ j : h_{2,j} = h_{3,j} = 1 \right\} \right|, \\
b &= \left| \left\{ j : h_{2,j} = 1, \quad h_{3,j} = -1 \right\} \right|, \\
c &= \left| \left\{ j : h_{2,j} = -1, \quad h_{3,j} = 1 \right\} \right|, \\
d &= \left| \left\{ j : h_{2,j} = h_{3,j} = -1 \right\} \right|
\end{aligned}$$

olmak üzere, aşağıdaki eşitlikler elde edilir.

$$\begin{aligned}
a + b + c + d &= n \\
r_1 \bullet r_2 = 0 \text{ olduğundan } a + b - c - d &= 0 \\
r_1 \bullet r_3 = 0 \text{ olduğundan } a - b + c - d &= 0 \\
r_2 \bullet r_3 = 0 \text{ olduğundan } a - b - c + d &= 0
\end{aligned}$$

Bu sistem, $a = b = c = d = \frac{n}{4}$ olan tek çözüme sahiptir. a, b, c ve tamsayı olduğundan $n \equiv 0 \pmod{4}$ durumunda olmalıdır.

2.4.2. Hadamard matrislerinin bazı kuruluş metotları

Hadamard matrisleri için birçok kuruluş metodu vardır. Bu bölümde bu kuruluş yöntemlerinden en önemlileri olan Sylvester ve Paley kuruluşu verilecektir.

Sylvester kuruluşu

$A=(a_{ij})$ ve $B=(b_{ij})$ $m_1 \times m_2$ ve $n_1 \times n_2$ boyutlu matris olsunlar. A ve B 'nin kronoker çarpımı, $A \otimes B$, $m_1 n_1 \times m_2 n_2$ boyutlu ve A matrisindeki a_{ij} elemanının yerine $a_{ij}B$ matrisi konularak elde edilen matristir.

2.9. Teorem

A ve B hadamard matrisleri ise $A \otimes B$ matrisi de hadamard matrisidir. Dolayısıyla m ve n mertebeli hadamard matrisi var ise mn mertebeli Hadamard matrisi de vardır [Tressler, 2004].

2.7. Örnek

Örnek 2.6'deki H_2 ve H_4 matrislerini tekrar ele alınsın. Hadamard Matrislerinin kronecker çarpımı aşağıda görüldü gibi yeni bir Hadamard matrisidir.

$$\begin{aligned}
 H_2 \otimes H_4 &= \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \\ \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} & - \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \end{bmatrix}
 \end{aligned}$$

Yukarıdaki kuruluş Sylvester kuruluşu yada Kronecker kuruluşu olarak bilinir. Ayrıca Sylverter matrisi H_2 üzerinde Sylvester kuruluşunu kullanarak oluşturulur.

2.3. Tanım

S_0 , [1] matrisi ve

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

olsun. Sylvester matrisi S_n , $S_n = H_2 \otimes S_{n-1}$ şeklinde tanımlanan $2^n \times 2^n$ boyutlu matristir [Tressler, 2004].

Paley kuruluşu

Paley kuruluşundan önce kuadratik artık kavramını vermek gerekir.

F_q , q üzerinde cisim ve q tek ve asal olsun. Kuadratik artık kümesi $QR(q)$ aşağıdaki gibi tanımlanır.

$$QR(q) = \{z^2 : z \in F_q, z \neq 0\}$$

ve

$$QNR(q) = F_q - \{QR(q) \cup 0\}$$

Buradan F_q üzerinde tanımlı kuadratik karakter fonksiyonu;

$$\chi_q(x) = \begin{cases} 0 & x = 0 \\ 1 & x \in QR(q) \\ -1 & x \in QNR(q) \end{cases}$$

şeklinde verilir.

2.1. Yardımcı Teorem

$q \equiv 3 \pmod{4}$ olacak şekilde q asal sayı olsun, bu durumda $q+1$ mertebeli Hadamard matrisi vardır [Tressler, 2004].

Ayrıca bu matris,

$$h_{i,j} = \begin{cases} 1 & i = \infty \text{ yada } j = \infty \\ \chi_q(j-i) & i, j \in \mathbb{F}_q, i \neq j \\ -1 & i = j, i \neq \infty \end{cases}$$

biçimindedir. Burada ∞ ilk satır ve sütun indisleridir.

2.2. Yardımcı Teorem

$q \equiv 1 \pmod{4}$ olacak şekilde q asal sayı olsun, bu durumda $2(q+1)$ mertebeli Hadamard matrisi vardır [Tressler, 2004].

B matrisi;

$$b_{i,j} = \begin{cases} 1 & i = \infty \text{ yada } j = \infty, i \neq j \\ \chi_q(j-i) & i, j \in \mathbb{F}_q, i \neq j \\ 0 & i = j \end{cases}$$

alınsın. C matrisi, B matrisinin elemanlarından $+1, -1, 0$ 'ın yerine sırasıyla

$$\begin{pmatrix} + & + \\ + & - \end{pmatrix} \quad \begin{pmatrix} - & - \\ - & + \end{pmatrix} \quad \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

elemanlarının konmasıyla elde edilir. D matrisi ise B matrisinin elemanlarından $+1, -1, 0$ 'ın yerine sırasıyla

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \quad \begin{pmatrix} + & - \\ - & - \end{pmatrix}$$

elemanlarının konmasıyla elde edilir. $A = C + D$ olarak elde edilen matris $(2q+2)$ mertebeli Hadamard matrisidir.

2.8. Örnek

F_5 cismindeki kuadratik artıklar $QR(q)=\{1,4\}$ 'dir. Dolayısıyla B matrisi;

$$B = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & -1 & -1 & 1 \\ 1 & 1 & 0 & 1 & -1 & -1 \\ 1 & -1 & 1 & 0 & 1 & -1 \\ 1 & -1 & -1 & 1 & 0 & 1 \\ 1 & -1 & -1 & -1 & 1 & 0 \end{bmatrix}$$

şeklindedir. Buradan C matrisi

$$C = \begin{bmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} + & + \\ + & - \end{pmatrix} & \begin{pmatrix} + & + \\ + & - \end{pmatrix} & \begin{pmatrix} + & + \\ + & - \end{pmatrix} & \begin{pmatrix} + & + \\ + & - \end{pmatrix} & \begin{pmatrix} + & + \\ + & - \end{pmatrix} \\ \begin{pmatrix} + & + \\ + & - \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} + & + \\ + & - \end{pmatrix} & \begin{pmatrix} - & - \\ - & + \end{pmatrix} & \begin{pmatrix} - & - \\ - & + \end{pmatrix} & \begin{pmatrix} + & + \\ + & - \end{pmatrix} \\ \begin{pmatrix} + & + \\ + & - \end{pmatrix} & \begin{pmatrix} + & + \\ + & - \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} + & + \\ + & - \end{pmatrix} & \begin{pmatrix} - & - \\ - & + \end{pmatrix} & \begin{pmatrix} - & - \\ - & + \end{pmatrix} \\ \begin{pmatrix} + & + \\ + & - \end{pmatrix} & \begin{pmatrix} - & - \\ - & + \end{pmatrix} & \begin{pmatrix} + & + \\ + & - \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} + & + \\ + & - \end{pmatrix} & \begin{pmatrix} - & - \\ - & + \end{pmatrix} \\ \begin{pmatrix} + & + \\ + & - \end{pmatrix} & \begin{pmatrix} - & - \\ - & + \end{pmatrix} & \begin{pmatrix} - & - \\ - & + \end{pmatrix} & \begin{pmatrix} + & + \\ + & - \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} + & + \\ + & - \end{pmatrix} \\ \begin{pmatrix} + & + \\ + & - \end{pmatrix} & \begin{pmatrix} + & + \\ + & - \end{pmatrix} & \begin{pmatrix} - & - \\ - & + \end{pmatrix} & \begin{pmatrix} - & - \\ - & + \end{pmatrix} & \begin{pmatrix} + & + \\ + & - \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \end{bmatrix}$$

dir. Benzer şekilde;

2.4.3. Hadamard tasarımlar

Hadamard matrislerinden simetrik ve simetrik olmayan dengeli tamamlanmamış blok tasarımlar elde edilebilir. Böyle tasarımlara hadamard tasarımlar adı verilir.

2.10. Teorem

$m > 1$ olacak şekilde simetrik $(4m-1, 2m-1, m-1)$ -düzeninin var olması için gerek ve yeter koşul $4m$ mertebeli Hadamard matrisinin var olmasıdır [Hall,1986].

2.9. Örnek

$m=8$ mertebeli bir Hadamard matrisi aşağıdaki gibi olsun.

$$H_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

Buradan ilk satır ve sütunu atıp geri kalan matristeki -1 yerine 0 koyarak elde edilen matris;

$$A = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

şeklindedir. Bu matrise dikkat edilecek olursa simetrik $(7,3,1)$ -tasarımıdır.

Tanım 2.2 ve Teorem 2.5'den faydalanarak artık ve türetilmiş tasarımlar üzerine aşağıdaki sonuç verilebilir.

2.1. Sonuç

$m > 2$ mertebeli bir Hadamard matrisi mevcut ise aşağıdaki tasarımlar vardır.

1. $(4m-1, 2m-1, m-1)$ DTBT
2. $(4m-1, 2m, m)$ DTBT
3. $(2m-1, m-1, m-2)$ DTBT
4. $(2m, m, m-1)$ DTBT (Hall, 1986).

2.10. Örnek

Örnek 2.9'daki A ilişki matrisinde verilen tasarım ele alınsın. Bu tasarımdan, Sonuç 2.1'deki 2.nci sonucu elde etmek için 1 ile 0'ların yerlerine değiştirilirse elde edilen tasarım simetrik 2- $(7,4,2)$ DTBT'dir. 3. ve 4. durumlar için ise tasarımın artık ve türetilmiş tasarımını bulmak yeterlidir.

2.5. Sonlu Geometriler

19ncu yüzyıl matematikçileri, matematiğin birçok disiplini ile ilgilenmişlerdir. Bunların en önemlilerinden biri sonlu geometridir. Geometri 20. yüzyılın başlarında doruğa ulaşmış ve daima aksiyomatik temelli konuların prototipi olarak düşünülmüştür.

G.Fano bir doğru üzerinde noktaların sonsuz sayısının varlığını kapsayan aksiyomlara karar verme problemiyle ilgilenmiştir. Tüm aksiyomları sağlayan fakat doğru üzerinde sonsuz sayıda olmayan noktalar sistemlerini bulmuştur. Böylece sonlu geometri doğmuştur.

Bütün geometriler projektif geometriden elde edilen alt geometriler gibi düşünülebilirler. Geometriler ile tasarımlar arasında birçok ilişki vardır. Sonlu geometrilerden simetrik DTBT, t -tasarımlar, ortogonal dizimler, ikişerli ortogonal Latin kareler, çözülebilir tasarımlar gibi çok önemli kombinatorik tasarımlar elde edilir.

2.5.1. Projektif geometri

q asal yada asalın katı ve V_n , $GF(q)$ üzerinde n boyutlu vektör uzayı olsun. V 'nin $(i+1)$ -boyutlu alt uzayına geometrinin i -flat'ı denir. a ve b flatlarının ilişkili olması için $a \subset b$ ya da $b \subset a$ olması gerekir. Bu flat ve aralarındaki ilişkiler q mertebeli ve n boyutlu projektif geometri $PG(n, q)$ ile verilir. 0-flat noktalar 1-flat doğrular $(n-1)$ -flat ise n boyutlu hiperdüzlemi ifade eder.

2.4. Tanım (Gauss Polinomu)

$PG(n, q)$ 'daki i -flat sayısı

$$\begin{bmatrix} n+1 \\ i+1 \end{bmatrix} = \frac{(q^{n+1}-1)(q^{n+1}-q)\cdots(q^{n+1}-q^i)}{(q^{i+1}-1)(q^{i+1}-q)\cdots(q^{i+1}-q^i)}$$

dir [Beutelspacher and Rosenbaum, 1998].

Bu durumda nokta ve n -boyutlu hiperdüzlemlerin sayısı $\frac{(q^{n+1}-1)}{(q-1)}$ kadardır.

2.11. Örnek

3 boyutlu 2 mertebeli $PG(3,2)$ 'ye ait 3- boyutlu hiperdüzlemler ve üzerindeki noktalar aşağıdaki gibidir.

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

Burada hiperdüzlemler, bloklar gibi düşünüldüğünde 2-(15,7,3) simetrik tasarımının isabet matrisi olduğu görülür.

Aşağıdaki teoremden Projektif geometriden simetrik 2-(v,k,λ) tasarımının elde edilmesini verir.

2.11. Teorem

$q > 1$ ve $n > 1$ tamsayıları için;

$$2 - \left(\frac{q^{n+1} - 1}{q - 1}, \frac{q^n - 1}{q - 1}, \frac{q^{n-1} - 1}{q - 1} \right)$$

parametrelerine sahip simetrik 2-(v,k,λ) vardır [Cameron and Lint,1996].

2.5.2. Afın geometri

Afın geometri projektif geometriden elde edilen bir yapıdır. Bu geometrinin birçok tasarım ve kod türleriyle ilişkisi vardır. Afın geometri, kısaca projektif geometriden

bir hiperdüzlemin ve içerdiği noktaların atılmasıyla elde edilir. Bu durumda nokta sayısı q^n ve n -boyutlu hiperdüzlemlerin sayısı $\frac{(q^{n+1}-q)}{(q-1)}$ şeklindedir.

2.12. Örnek

Örnek 2.11 tekrar düşünölsün. Bu yapıdan, 1.nci hiperdüzlem ve bu hiperdüzleme ait noktalar çıkarılsın. Bu durumda bir AG(3,2)ye ait doğrular ve üzerindeki noktalar;

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

şeklinde olacaktır. Göröldüğü gibi $2^3=8$ noktalı ve $(2^4-2)/(2-1)=14$ hiperdüzlemlı bir afın geometri elde edilir. Bu, aynı zamanda 2-(8,4,3) tasarımıdır.

2.15. Teorem

$q>1$ ve $n>1$ tamsayıları için;

$$2-\left(q^n, q^{n-1}, \frac{q^{n-1}-1}{q-1}\right)$$

parametrelerine sahip 2-tasarım vardır [Cameron ve Lint,1996].

2.6. Latin Kareler ve Ortogonal Dizimler

Matematikte, bir matris türü olarak bilinen Latin Kareler, istatistikte sıkça başvurulanan deney düzenleme yöntemlerinden biridir.

Latin kareler, değişkenlerin tamamen farklı iki yolla, satır ve sütunlarla gruplanmasından dolayı genellikle deneysel tasarımlarda kullanılırlar. Bu çiftli gruplamanın etkisi satır ve sütunlar arasındaki farklılıklardan kaynaklanan hataların küçültülmesi içindir.

Matematikte bir diğer önemli matris türü olan ortogonal dizimler istatistikte başvurulanan deney düzenleme yöntemlerinden biridir. Ortogonal dizimler kendilerine uyan yapıdan başka bir yapıya geçişte kullanılır. Bu tasarımlarda işlem çiftlerinin birlikte görülme sayısı her denemede farklıdır.

Bose ve Kishen (1940), ortogonal dizimler ve simetrik faktöriyel tasarımlarda etki karışımı teorisi arasındaki ilişkiyi kurmuşlardır (temelde sonlu projektif geometriyi kullanmışlardır) [Bose ve Bush, 1952].

Kishen (1942), ortogonal Latin kareler ve Latin kareler kavramını tanıtmıştır ve Latin kareleri Latin küplere, hiperküplere ayrıca, ortogonal Latin küplere genelleştirmiştir [Raghavarao, 1971].

2.6.1. Latin kareler

n satır ve n sütunu olan bir karenin elemanları $\{1,2,\dots,n\}$ kümesinin elemanlarından oluşsun ve her sayı, her satır ve sütunda 1 kez gözüksün. Böyle bir tasarıma n . mertebeden *latin kare* adı verilir. aynı mertebeden iki latin karenin ortogonal olması için herhangi $x,y \in \{1,2,\dots,n\}$ için $a_{ij}=x$ ve $b_{ij}=y$ olan tek bir (i,j) olmalıdır. Latin karelerin kümesinin ortogonal olması için her çift karenin ikişerli ortogonal olması gerekir. m tane n . mertebeden *ikişerli ortogonal Latin kareler* (*MOLS*=Mutally

Ortogonal Latin Squares) verilmiş olsun. Ortogonallik kare içindeki tekrar sıralanmaktan etkilenmez. Ortogonallikten dolayı $m \leq n-1$ olur. $m=n-1$ ise MOLS'a *tam* denir.

2.13. Örnek

4.mertebeden bir Latin kare aşağıdaki gibi yazılabilir.

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \\ 3 & 4 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix}$$

Ayrıca böyle bir Latin kareye ortogonal olan bir başka Latin kare,

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \\ 2 & 1 & 4 & 3 \\ 3 & 4 & 1 & 2 \end{bmatrix}$$

şeklindedir. Burada dikkat edilecek nokta, herhangi $x, y \in \{1, 2, \dots, n\}$ için $a_{ij}=x$ ve $b_{ij}=y$ olan tek bir (i, j) olmasıdır. Bu iki Latin kareye ikisine de ortogonal olan bir başka Latin kare eklenecek olursa $m=4-1=3$ olacağından böyle bir MOLS'e tam denir ve aşağıdaki gibi verilir.

$$\left\{ \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \\ 3 & 4 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \\ 2 & 1 & 4 & 3 \\ 3 & 4 & 1 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 2 \\ 4 & 3 & 2 & 1 \\ 2 & 1 & 4 & 3 \end{bmatrix} \right\}$$

Afın Düzlemlerin elde edilişi için n .nci mertebeden Latin kare tam seti kullanılabilir. n .nci mertebeden Latin kare tam seti $L_1, L_2, \dots, L_{n-1}$ şeklinde olsun. Burada, noktaları

$X = \{1, 2, \dots, n\} \times \{1, 2, \dots, n\}$ olan Afin düzlem oluşturulacaktır. Afin Düzlemin blokları aşağıdaki gibidir.

$$1 \leq x \leq n-1, 1 \leq k \leq n \text{ için}$$

$$A_{x,k} = \{(i, j) : L_x(i, j) = k\}.$$

$$\text{ve } 1 \leq k \leq n \text{ için}$$

$$A_{n,k} = \{(i, j) : 1 \leq j \leq n\}$$

Böylece;

$$\mathcal{A} = \{A_{x,k} : 1 \leq x \leq n+1, 1 \leq k \leq n\}$$

şeklinde oluşur. Burada $(X, \mathcal{A})$ yapısına n .nci mertebeden afin düzlemi denir.

2.6.2. Ortogonal dizimler

Ortogonal dizimler $OA(N, k, s, t)$ şeklinde tanımlanır. N sütun sayısı, k satır sayısı (kısıt sayısı), s düzey sayısı ve t kuvveti gösterir. Böyle bir ortogonal tasarım $k \times N$ boyutlu ve tüm elemanları $0, 1, \dots, s-1$ 'den oluşan bir matristir. Bu matrisin her $t \times N$ matrisinin sütun vektörleri λ kez tekrar eder. λ ortogonal dizimin indeksidir. $N = \lambda s^t$ 'dir [Hedayat Sloane and Stufken, 1999].

2.14. Örnek

$\lambda=1$ olan $OA(9, 4, 3, 2)$ 'a bir örnek olarak aşağıdaki matris verilebilir.

$$OA(9,4,3,2) = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \\ 0 & 1 & 2 & 1 & 2 & 0 & 2 & 0 & 1 \\ 0 & 1 & 2 & 2 & 0 & 1 & 1 & 2 & 0 \end{bmatrix}$$

Ayrıca iki ortogonal diziden birbirlerine bir geçiş varsa bu ortogonal dizimlere denk denir. $OA(N,k,s,t)$ den bir satır atılırsa $OA(N,k-1,s,t)$ elde edilir.

Aşağıdaki teorem ortogonal dizimlerin satır sayısı için bir üst sınır verir.

2.13. Teorem

$OA(N,k,s,2)$ 'nün varlığı için;

$$k \leq \frac{s^2\lambda - 1}{s - 1}$$

olmalıdır. $k = \frac{s^2\lambda - 1}{s - 1}$ ise ortogonal dizim tamdır [Bose and Bush, 1942].

2.15. Örnek

$OA(9,4,3,2)$ ortogonal dizimi aşağıdaki gibi olsun.

$$A^T = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \\ 0 & 1 & 2 & 2 & 0 & 1 & 1 & 2 & 0 \\ 0 & 1 & 2 & 1 & 2 & 0 & 2 & 0 & 1 \end{pmatrix}$$

$k = (3^2 \times 1 - 1) / (3 - 1) = 4$ olduğundan A tam ortogonal dizimdir.

Ayrıca Hadamard matrisleri ile ortogonal dizimler arasında yakın ilişki vardır. 4λ mertebeli Hadamard matrisindeki ilk satır elemanlarını çıkararak elde edilen matris $(4\lambda, 4\lambda-1, 2, 2)$ parametrelili ortogonal dizim olur.

Bir ortogonal dizimde $t=1$ olursa böyle bir ortogonal dizime Latin hiperküp denir. Diğer bir deyişle $OA(N, k, s, 1)$ parametereli bir ortogonal dizim bir Latin hiperküptür.

3. KODLAMA TEORİSİ

Kodlara ihtiyaç duyulmasının birçok sebebi vardır. Bunlardan en önemlisi gönderilecek verinin gizliliği ya da yetkisiz kişiler tarafından erişiminin engellenmesidir. Böyle kodlar Kriptoloji konusuna dâhildirler. Bir başka sebep bilgiyi sıkıştırarak bilgisayarın sabit diskinde kapladığı alanı küçültmektir. Böyle kodlar, bilgisayar teknolojisinde karşılaşılan “zip” kodlarıdır. Bir başka sebep, bilginin gönderilmesindeki güvenilirliğin artırılmasıdır. Diğer bir deyişle gönderim sırasında oluşabilecek hataları düzeltilmesinin maksimum düzeye çıkarılmasıdır. Böylece yanlış yorumlanma minimuma indirgenmiş olur. Kodlama teorisinde amaç, kod kelimelerde meydana gelen hataların tespiti ve mümkünse düzeltilmesidir.

Kodlama teorisinde düzeltilecek hataların sayısının maksimum olması istenir. Bu sebepten tam kodlar geliştirilmiştir. Fakat tam kodu bulmak her zaman mümkün olmadığından alternatif olarak optimal yada maksimal kod oluşturulur. Bu bölümde optimal ve maksimal kodlarının bazı türleri verilecektir. Bu nedenle bu kodlara ilişkin temel tanımlar vermek gerekir.

$GF(q)$, q mertebeli Galois Cismi olmak üzere, V_n , $GF(q)$ üzerinde vektör uzayı olsun. $x=(x_1, x_2, \dots, x_n)$ ve $y=(y_1, y_2, \dots, y_n)$ V_n 'in elemanı olmak üzere aralarındaki hamming uzaklığı $\rho(x, y)$;

$$\rho(x, y) = \left| \{i \mid x_i \neq y_i\} \right| \quad (3.1)$$

Şeklinde olup, x ve y vektörleri arasındaki farklı noktaların sayısıdır.

Bir q -ary C kodu V_n 'in bir alt kümesidir. C 'deki vektörler kod kelimeler olarak adlandırılır. Tüm $x, y \in C$ çiftleri arasındaki uzaklıkların en küçüğüne *minimum uzaklık* adı verilir ve aşağıdaki gibi tanımlanır.

$$d = \min \{ \rho(x, y) : x, y \in C \} \quad (3.2)$$

Minimum uzaklığı d olan bir C kodu için düzeltilebilecek maksimum hata sayısı $(d-1)/2$ dir. Kod kelimelerin bir iletişim kanalından bilgi gönderimi için kullanıldığı ve $x \in C$ kod kelimesinin gönderim sırasında gürültüden dolayı $y \in V_n$ şeklinde ulaştığı varsayalım. Alıcı y kod kelimesini eğer $\rho(x,y) \leq (d-1)/2$ ise x olarak çözer. Eğer $\rho(x',y) \leq (d-1)/2$ olan başka bir x' kod kelimesi varsa y kod kelimesi x' olarak çözülür. Burada dikkat edilecek nokta hem $\rho(x',y) \leq (d-1)/2$ hem de $\rho(x,y) \leq (d-1)/2$ olamayacağıdır. Bunun sebebi,

$$\rho(x,x') \leq \rho(x,y) + \rho(x',y) \leq d-1$$

şeklinde verilen üçgen eşitsizliği ile açıklanabilir Aşağıdaki örnek minimum uzaklığın hesaplanmasını verir.

3.1. Örnek

Aşağıdaki gibi bir C koduna sahip olunsun.

$$C = \begin{pmatrix} 10101 \\ 01110 \\ 11100 \\ 10111 \end{pmatrix}$$

C matrisinin satırları C_1, C_2, C_3 ve C_4 kod kelimeler olmak uzaklıklar,

$$\rho(C_1, C_2)=4$$

$$\rho(C_1, C_3)=3$$

$$\rho(C_1, C_4)=1$$

$$\rho(C_2, C_3)=2$$

$$\rho(C_2, C_4)=3$$

$$\rho(C_3, C_4)=3$$

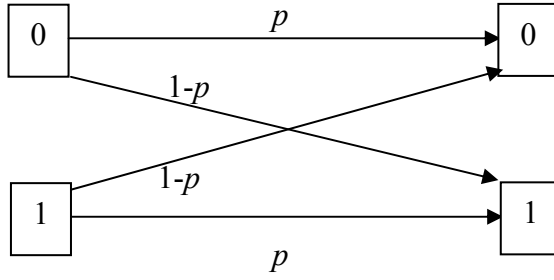
şeklinde. Minimum uzaklık $d=1$ olur. Dolayısıyla C kodu $(5,4,1)$ parametrelidir.

Bu ise minimum uzaklığın d olmasına aykırıdır. Böyle çözümlmeye en çok olabilirlik çözümlmesi denir. $C \subseteq V_n$ olmak üzere minimum uzaklık d , $|C|=M$ ve uzunluğu n olan bir koda (n, M, d) kodu denir.

3.1. En Çok Olabilirlik Çözümlmesi

Kodlama teorisinde önemli konularından biri çözümlmedir. Çözümlme temelde alıcının kanalda bozulmuş mesajın düzeltilme işlemidir. Bunun için birçok yöntem geliştirilmiştir. En çok olabilirlik, tamamlanmamış en çok olabilirlik ve sendrom çözümlmesi birer örnek olarak verilebilir. En çok olabilirlik çözümlmesi kısaca kanalda bozulan vektörü, koddaki kod kelimelerle uzaklıklar bakımından karşılaştırarak kodun minimum uzaklığından küçük olan kod kelimeyi(varsa) bulur ve vektörü o kod kelime olarak çözümler. Eğer böyle bir kod kelime yoksa yeniden gönderim gerekir. Tamamlanmamış en çok olabilirlik algoritmasında, en çok olabilirlik algoritmasına benzer şekilde koddaki kod kelimelerle uzaklıklar bakımından karşılaştırarak vektöre en yakın kod kelimeyi bulur ve vektörü o kod kelime olarak çözer. Fakat bu yöntemin bazı sakıncaları vardır. Minimum uzaklık dikkate alınmadığından vektöre aynı uzaklıkta birden çok kod kelime olabilir. Bu durumda tamamlanmamış en çok olabilirlik algoritması bu iki kod kelimedenden rasgele birini seçer. Bu da vektörü doğru kod kelimeye çözümlmeyi zorlaştırır. Sendrom çözümlmesi ise bunlara en çok olabilirlik ve tamamlanmamış en çok olabilirlik çözümlmesine benzeyen fakat sadece lineer kodlar için kullanılan bir çözümlme algoritmasıdır.

Çözümlme teknikleri dikkate alındığında kodlama teorisinde en önemli çözümlme tekniği En çok olabilirlik çözümlmesidir (EÇÖÇ). Bu bölümle bu tekniği kullanarak çözümlme yapılışı verilecektir. Bu teknik için önce, ikili simetrik kanaldan (İSK) bahsetmek gerekir. İkili simetrik kanal, sadece ikili (binary) kodlar için kullanılan bir kanal türüdür. Bu kanala benzer birçok tane kanal vardır. ikili simetrik kanalı aşağıdaki şekil ile ifade edilsin.



Şekil 3.1. İkili simetrik kanal

Burada p , İSK'dan gönderilen kod kelimenin dijitin doğru olarak iletilmesi olasılığıdır. Ayrıca $0.5 < p \leq 1$ olduğu varsayılır aksi halde kanal kullanılmaz ya da $0.5 < p \leq 1$ olacak şekilde düzenlenir. $0.5 < p \leq 1$ değilse, kod kelimenin bir dijitin yanlış gönderilmesi olasılığının doğru gönderilmesi olasılığından daha büyük olacağı anlamına gelir. Bu da istenilmeyen bir durumdur. Bu aralığı 1 yerine 0, 0 yerine 1 göndererek $0.5 < p \leq 1$ olması sağlanabilir. 5 uzunluklu bir kelimenin doğru olarak gönderilmesi olasılığı p^5 'tir. Fakat 2 hatayla gönderilmesi olasılığı $10p^3(1-p)^2$ şeklindedir.

En çok olabilirlik çözümlemesi aşağıdaki örnek üzerinde anlatılacaktır.

3.2. Örnek

(10,15,3) parametrel C kodu ve kod kelimelere karşılık gelen harfler aşağıdaki gibi olsun.

$$C = \begin{pmatrix} \begin{matrix} A \\ E \\ O \\ İ \\ C \\ K \\ T \\ S \\ U \\ V \\ D \\ F \\ L \\ N \\ P \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \end{pmatrix}$$

$d=3$ olduğundan $e=1$ dir. Harfler yukarıdaki gibi şifrelensin. Böyle şifrelenen bir alfabe için İSTATİSTİK kelimesini şifreledikten sonra İSK geçirip çözmeye çalışılsın.

Şifrelenmiş mesaj;

$$\begin{matrix} \text{İ} \\ \text{S} \\ \text{T} \\ \text{A} \\ \text{T} \\ \text{İ} \\ \text{S} \\ \text{T} \\ \text{İ} \\ \text{K} \end{matrix} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

şeklinde olur. İSK çıktıktan sonra bazı hatalar olduğu varsayılınsın. Alıcıya gelen şifrelenmiş mesaj;

$$\begin{matrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{matrix}$$

şeklinde olacaktır. Gelen mesajdaki vektörler ile C 'deki kod kelimeler arasındaki uzaklık 1 yada 0 ise vektör o kod kelime olarak çözümlenir. Eğer hiçbirine 1 yada 0 uzaklıkta değilse o kod kelime çözümlenemez ve tekrar gönderim istenir. Buna göre ilk kelimeye baktıldığında $d(1101010010, 0101010010)=1$ olduğundan ilk kod kelime U olarak çözümlenir. Benzer şekilde diğerleri yapıldığında;

U	0	1	0	1	0	1	0	0	1	0
S	1	1	1	1	1	1	1	0	0	0
T	0	0	1	0	1	1	0	1	0	0
A	0	1	0	1	0	1	0	1	0	1
*	*	*	*	*	*	*	*	*	*	*
İ	1	1	1	0	0	0	0	1	1	1
P	0	0	1	0	1	1	0	0	1	1
T	0	0	1	0	1	1	0	1	0	0
İ	1	1	1	0	0	0	0	1	1	1
K	1	0	0	0	0	1	1	1	1	0

olarak çözümlenir. Burada * ile gösterilen ifade vektörün hiçbir kod kelimeye 1 ya da daha az uzaklıkta olmadığından dolayı çözümlenemediğini ifade eder.

Kodlama teorisi bu çerçevede daha çok hata düzeltmek için çeşitli sınırlar ve bunları sağlayan kodların geliştirilmesinde rol oynamıştır. Daha çok hata düzeltmek için; ya kod kelime sayısını ya da minimum uzaklığı maksimize etmek gerekir.

$M(n,d)$, sabit n ve d için M 'nin maksimum değeri, $d=d(n,M)$ ise sabit n ve M için M 'nin maksimum değeri olmak üzere;

Kodlama teorisinde iki temel amaç vardır:

- 1) sabit n ve d değerleri için $M=M(n,d)$ değerini maksimize etmek,
- 2) sabit M ve d değerleri için $d=d(n,M)$ değerini maksimize etmek.

$M(n,d)$ ve $d=d(n,M)$ fonksiyonları için çeşitli sınırlar geliştirilmiştir. Bunların en önemlilerden biri Plotkin sınırıdır. Plotkin sınırı $q=2$ (2-düzeyle) olan kodlar için geçerlidir.

Aşağıdaki teorem, kodlardaki kod kelime sayısına ilişkin üst sınırı(Plotkin Sınırı) verir.

3.1. Teorem (Plotkin Sınırı)

C , $n < 2d$ olan bir (n, M, d) kodu olsun.

d çift ve $2d > n$ ise

$$M(n, d) \leq 2 \left\lceil \frac{d}{2d - n} \right\rceil \quad (3.3)$$

$$M(2d - 1, d) \leq 2d \quad (3.4)$$

$$M(2d, d) \leq 4d \quad (3.5)$$

d tek ve $2d + 1 > n$ ise;

$$M(n, d) \leq 2 \left\lceil \frac{d + 1}{2d + 1 - n} \right\rceil \quad (3.6)$$

$$M(2d + 1, d) \leq 4d + 4 \quad (3.7)$$

dir [Hoffman ve Ark., 1992].

Eş.3.3- Eş. 3.7'deki ifadelerde eşitliği sağlayan kodlara maksimal kod denir.

3.3. Örnek

Bir $(7, 6, 3)$ kodu aşağıdaki gibi olsun.

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Burada $n=2 \times 3 + 1$ olduğundan ve Eş. 3.7'ye göre $M(7,3)=16 > 6$ olup bu kod maksimal değildir. Ancak;

$$C' = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

kodu $(7,8,4)$ parametrelili olup ve Eş. 3.3'e göre $M(7,4)=2[2 \times 4 / (2 \times 4 - 7)]=8=M$ olduğundan C' kodu maksimaldir.

3.2. Eşit Uzaklıklı Kodlar

Eşit uzaklıklı kodlar, kod kelimeler arasındaki uzaklıklar birbirine eşit olan kodlardır. Eşit uzaklıklı kodlarda kod kelime sayısı maksimize edilerek optimal kod elde edilir.

C , q -düzeyli (n, M, d) parametrelili kod olsun. Her hangi bir kod için minimum uzaklık kod kelimeler arası ortalama uzaklığı geçemez.

$$d \leq \bar{d} = \frac{1}{M(M-1)} \sum_{x \in C} \sum_{y \in C} \rho(x, y) \quad (3.9)$$

Eğer $\bar{d}$ tamsayı ve her kod kelime arasındaki uzaklık $\bar{d}$ ise böyle bir koda *eşit uzaklıklı kod* denir.

Aşağıdaki teorem eşit uzaklıklı kodların minimum uzaklığı d için bir üst sınır verir.

3.2. Teorem

Eşit uzaklıklı her (n, M, d, q) kodu için aşağıdaki eşitsizlik geçerlidir.

$$d \leq d_{maks} = nM(q-1)/(M-1)q$$

Eşitliğin sağlanması için M 'nin q 'nın katı ve $0, 1, \dots, q-1$ sembollerinin $M \times n$ boyutlu kod kelimeler matrisinin her sütununda $t=M/q$ defa bulunması gerekir [Tonchev, 1988].

$d=d_{maks}$ ise eşit uzaklıklı koda optimal denir ve parametreleri aşağıdaki gibidir.

$$n, M=qt, d = nt(q-1)/(qt-1) \quad (3.10)$$

3.4. Örnek

Bir $(4, 9, 3, 3)$ parametrelili kod aşağıdaki gibi olsun.

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 2 & 2 & 2 \\ 1 & 0 & 2 & 1 \\ 1 & 1 & 0 & 2 \\ 1 & 2 & 1 & 0 \\ 2 & 0 & 1 & 2 \\ 2 & 1 & 2 & 0 \\ 2 & 2 & 0 & 1 \end{bmatrix}$$

Burada $d=4 \times 3(3-1)/(3 \times 3-1)=3=d_{maks}$ olduğundan bu kod Teorem 3.2'ye göre optimal koddur.

3.3. Sabit Ağırlıklı Kodlar

Sabit ağırlıklı kodlar, içerdiği tüm kod kelimelerin ağırlıklarının sabit olduğu; yani kod kelimelerdeki 0ların sayısının birbirine eşit olduğu kodlardır. Bu kodlar için Johnson sınırı kullanılarak maksimal kodlar elde edilir. Bu bölümde sadece $q=2$ ve ağırlıkları sabit olan kodlar göz önüne alınacaktır. Sebebi, burada verilen Johnson sınırının sadece bu tarz kodlar için var olmasıdır. n, d ve w parametreleri verilmişken kod kelimelerin maksimum sayısı $A(n, d, w)$ (sabit n, w ve d için M 'nin maksimum değeri) ile ilgilenilecektir.

3.3. Teorem

Sabit n, w ve d için M 'nin maksimum değeri $A(n, d, w)$ için aşağıdakiler geçerlidir [Tonchev, 1988].

$$A(n, 2\delta-1, w) = A(n, 2\delta, w) \quad (3.11)$$

$$A(n, 2\delta, w) = A(n, 2\delta, n-w) \quad (3.12)$$

$$A(n, 2\delta, w) = 1, \quad w < \delta \quad (3.13)$$

$$A(n, 2\delta, w) = \lfloor n/\delta \rfloor \quad (3.14)$$

Eş. 3.14'ün genellenmiş haline *Johnson Sınırı* denir ve Teorem 3.4'te verildiği gibidir.

Teorem 3.4 ve Teorem 3.5 sabit ağırlıklı kodlarda kod kelime sayısının bir üst sınırını verir.

3.4. Teorem (Johnson Sınırı)

$$A(n, 2\delta, w) \leq \left\lceil \frac{\delta n}{w^2 - wn + \delta n} \right\rceil \quad (3.15)$$

[Huffman ve Pless, 2003].

3.5. Teorem

$$A(n, 2\delta, w) \leq \left\lceil \frac{n}{w} A(n-1, 2\delta, w-1) \right\rceil \quad (3.16)$$

[Huffman ve Pless, 2003].

Teorem 3.4 ve Teorem 3.5 birleştirilirse, Teorem 3.6'daki sonuca varılır.

3.6. Teorem

$$A(n, 2\delta, w) \leq \left\lceil \frac{n}{w} \left\lceil \frac{n-1}{w-1} \dots \frac{n-w+\delta}{\delta} \right\rceil \dots \right\rceil \quad (3.17)$$

olur [Huffman ve Pless, 2003].

Eş. 3.15 ve Eş. 3.17'deki tamsayı fonsiyonları kaldırılırsa;

$$A(n, 2\delta, w) \leq \frac{\delta n}{w^2 - wn + \delta n} \quad (3.18)$$

$$A(n, 2\delta, w) \leq \frac{n}{w} \frac{n-1}{w-1} \dots \frac{n-w+\delta}{\delta} \quad (3.19)$$

elde edilir.

3.5. Örnek

Aşağıdaki sabit ağırlıklı kod göz önüne alınsın.

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

Burada $w=4$, $\delta=2$, $n=7$, $M=7$ dir. $A(7,4,4)=2 \times 7 / (4^2 - 4 \times 7 + 2 \times 7) = 7 = M$ olduğundan Johnson sınırı sağlanır. Dolayısıyla bu koda maksimal koddur denilir. Bundan sonraki bölümde Bölüm 2’de verilen tasarımlardan faydalananarak bu bölümdeki kodların elde edilişi üzerinde durulacaktır.

4. TASARIMLARDAN ELDE EDİLEN KODLARIN KURULUŞU

Kombinatoriyel yapılar, 1970'lerden sonra özellikle son 10 yıl içinde optimal hata düzelten kodlar için kullanılmıştır. Bu bölümde Bölüm 2'de verilen kombinatorik tasarımlar ve Bölüm 3'de verilen maksimal ve optimal kodlar arasındaki ilişkiler ve bu yapılar arasındaki geçişler verilecektir.

4.1. Hadamard 2-tasarımlar ve Bu Tasarımlara İlişkin Kodlar

Hadamard matrisleri Walsh fonksiyonuna (Walsh,1923) dayalı olarak sinyal modellemesinde kullanılır [Harmurth, 1960]. Ayrıca bu fonksiyon kullanılarak sıkıştırılmış resimdeki verinin gönderilmesi ve şifrelenmesi de mümkündür [Jain,1989]. Hadamard matrisine dayalı maksimal kodlar 1960lı yıllarda uzay araştırmalarında Jüpiter, Satürn, Mars Uranüs, Neptün ve aylarının yüksek çözünürlüklü alınan fotoğraflarının doğru olarak çözümlenmesinde kullanıldı.

Bu bölümde Hadamard 2- tasarımlardan 3 tane maksimal kodun bulunuşu verilecektir. Ayrıca bu kodların Plotkin sınırını sağladığı da gösterilecektir..

4.1. Teorem

$z = 4\lambda$ ve $A, 2-(4\lambda-1, 2\lambda, \lambda)$ parametrelili Hadamard tasarımının ilişkili matrisi olmak üzere;

1) $C_1 = \begin{pmatrix} 0 \cdots 0 \\ A \end{pmatrix}$ kodu $(z-1, z, z/2)$ parametrelili maksimal koddur.

2) C_2 kodu C_1 ve C_1^c dan meydana gelen $(z-1, 2z, z/2-1)$ parametrelili maksimal koddur.

3) C_3 0'la başlayan kod kelimelerin silinmesiyle elde edilen maksimal $(z-2, z/2, z/2)$ kodudur [Tonchev, 1988].

Bu şekilde hadamard matrisinden elde edilen üç kod için Plotkin sınırı sağlandığından Plotkin sınırı eşitlik olarak aşağıdaki teoremdaki gibi verilebilir.

4.2. Teorem

Her λ için $2-(4\lambda-1, 2\lambda, \lambda)$ Hadamard tasarımı var ve

d çift ise;

$$M(n, d) = 2 \left\lceil \frac{d}{2d-n} \right\rceil \quad d \leq n < 2d \quad (4.2)$$

$$M(2\lambda, \lambda) = 4\lambda \quad (4.3)$$

dir.

d tek ise;

$$M(n, d) = 2 \left\lceil \frac{d+1}{2d+1-n} \right\rceil \quad d \leq n < 2d+1 \quad (4.4)$$

$$M(2d+1, d) = 4d+4 \quad (4.5)$$

dir [Tonchev, 1988].

4.1. Örnek

$\lambda = 2$ olan $2 - (7, 4, 2)$ tasarım ele alındığında A ilişki matrisi aşağıdaki gibi olsun.

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Bu Hadamard tasarımından elde edilen C_1 kodu $(7, 8, 4)$, C_2 kodu $(7, 16, 3)$ ve C_3 kodu $(6, 4, 4)$ şeklindedir. Bu kodlara dikkat edildiğinde tümü Teorem 3.1'i (Plotkin sınırı) sağlamaktadır, dolayısıyla bu kodların üçü de maksimaldir. Ayrıca C_2 dikkat edilecek olursa, elde edilen kodun maksimal olmasının yanı sıra lineer olan bu kod Hamming sınırını sağladığından tam koddur.

Ayrıca tasarımdan elde edilen kodlara dikkat edilsin. C_1 kodu $(7, 8, 4)$ parametrelidir. $n=7$ ve $d=4$ olduğundan dolayı Eş. 4.1 kullanılır. Bu eşitlikten kodun maksimal olması için;

$$M(7, 4) = 2 \left\lfloor \frac{4}{2 \cdot 4 - 7} \right\rfloor = 2 \cdot 4 = 8$$

olmalıdır. Elde edilen kodda $M=4$ olduğundan kod maksimaldir. C_2 kodu için ise $n=7$ ve $d=3$ olduğundan Eş. 4.5 kullanılacaktır. Dolayısıyla

$$M(7, 3) = 4 \cdot 3 + 4 = 16$$

olur. C_2 kodundaki kod kelime sayısı $M=16$ olduğundan maksimaldir. Son olarak C_3 kodunu ele aldığımızda, d çift ve $d \leq n < 2d$ olduğundan, yine Eş. 4.1 kullanılacaktır. Bu eşitlikten faydalanarak;

$$M(7,4) = 2 \left\lfloor \frac{4}{2 \cdot 4 - 6} \right\rfloor = 2 \cdot 2 = 4$$

elde edilir. Dolayısıyla C_3 kodu da maksimaldir.

4.2. Eşit Uzaklıklı Kodlar, Çözülebilir Tasarımlar ve Afin Geometri

Bir diğer önemli kod türü olan optimal eşituzaklıklı kodların ilk kuruluşu Preece (1967) tarafından verilmiştir. Kombinatoriyel tasarımların en önemlilerinden olan çözülebilir tasarımlar ve afin çözülebilir tasarımlar optimal kod oluşturmak için kullanılabilir. Bu kodların çözülebilir tasarımlardan kurulmasını ilk olarak Semakov ve Zinoviev (1968) vermiştir. Daha sonra bu kuruluş tekniği Tonchev (1988) ve Sinha (1994) tarafından geliştirilmiştir. Sinha ve Mitra (1999) çözülebilir ve içiçe dengeli tamamlanmamış blok tasarımını kullanarak yaklaşık optimal kodları vermişlerdir. Çözülebilir ve içiçe dengeli tamamlanmamış blok tasarımını kullanarak yaklaşık optimal kodların bir diğer kuruluş yöntemi Miao ve Kageyama (1999) tarafından verilmiştir.

Semakov ve Zinoviev (1968, 1969), tarafından verilen aşağıdaki teoremler Teorem 2.8'deki çözülebilir ve Teorem 2.9'daki afin çözülebilir 2-tasarımlardan Teorem 3.2'i sağlayan optimal kodların bulunuşu gösterir.

4.3. Teorem

(n, M, d, q) optimal eşit uzaklıklı kodunun varlığı ancak ve ancak aşağıdaki parametrelere sahip çözülebilir 2-tasarımın varlığıyla sağlanır.

$$v = M, \quad r = n, \quad k = t, \quad \lambda = n - d, \quad b = nq$$

[Semakov ve Zinoviev, 1968].

Eğer tasarım afin çözülebilir ise bu durumda Teorem 4.4 geçerlidir.

4.4. Teorem

$n = \frac{(qt-1)}{(q-1)}$ uzunluğunda her optimal eşit uzaklıklı kod aşağıdaki parametrelere sahip çözülebilir tasarıma denktir [Semakov ve Zinoviev, 1968].

$$v = qt, \quad k = t, \quad \lambda = \frac{(t-1)}{(q-1)}$$

Eğer $q=t$ ise özel olarak Teorem 4.5 geçerlidir.

4.5. Teorem

$q = t$ iken,

$$n = q + 1, \quad M = q^2, \quad d = q,$$

karşılık gelen tasarım

$$v = q^2, \quad k = q, \quad \lambda = 1$$

olur. Bu da optimal eşit uzaklıklı $(q+1, q^2, q, q)$ kodu, q mertebeli afin düzleme karşılık gelir [Semakov ve Zinoviev, 1969].

4.2. Örnek

Aşağıda AG(2,3) parametrelili Afin geometri A ve Teorem 4.5'e göre buradan elde edilen optimal kod C verilmiştir.

$$A = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 2 & 2 & 2 \\ 1 & 0 & 2 & 1 \\ 1 & 1 & 0 & 2 \\ 1 & 2 & 1 & 0 \\ 2 & 0 & 1 & 2 \\ 2 & 1 & 2 & 0 \\ 2 & 2 & 0 & 1 \end{bmatrix} = C$$

A' 'dan elde edilen C kodunun parametreleri Teorem 3.2'de yerine koyulduğunda

$$n = 4 \quad q + 1 = 3 + 1 = 4 \quad M = q^2 = 3^2 = 9 \quad d = q = 3$$

$$d_{maks} = 4 \cdot 9(3 - 1) / (9 - 1)3 = 3$$

olarak bulunur. $d = d_{maks}$ olduğundan elde edilen kodun optimal eşit uzaklıklı kod olduğu söylenebilir.

4.3. Sabit Ağırlıklı Kodlar ve 2- tasarımlar

Diğer bir önemli kod türünde sabit ağırlıklı kodlarda için maksimal kod sınırı Johnson (1962) tarafından verilmiştir. Bu kodların kuruluşu için bazı koşulları sağlayan, dengeli tamamlanmamış blok tasarımları kullanılmaktadır [Tonchev, 1988]. Daha sonra bu sınırın Optik Ortogonal kodlarla ilişkisi ortaya çıkarılmıştır [Chung ve Ark., 1989]. Ve bu sınır sabit ağırlıklı kodlar için minimum uzaklık $d > 1$ durumu için geliştirilmiştir [Omranı ve Ark., 2004]. Optik Ortogonal kodların Projektif

düzlemlerle ilişkisi Miyamoto ve Ark. (2004) tarafından ortaya konmuştur. Ayrıca Optimal kodların kuruluşlarında kullanılan bir diğer kombinatoriyel yapı da ortogonal dizimlerdir [Tonchev, 2005].

Bu bölümde sabit ağırlıklı kodlardaki kod kelimelerin üst sınırının belirlenmesi 2-tasarım yardımıyla yapılacaktır.

4.6. Teorem

Eş. 3.18'deki

$$A(n, 2\delta, w) \leq \frac{\delta n}{w^2 - wn + \delta n}$$

eşitsizliğinin sağlanması için gerek ve yeter koşul $b=n$, $r=w$ ve $\lambda=w-\delta$ olan bir 2-tasarımın varlığıdır [Tonchev, 1998].

4.3. Örnek

Örnek 2.1'deki 2-(8,4,3) tasarımını ele alınsın.

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Burada $b=14$ ve $r=7$ olduğundan $n=b=14$, $w=r=7$, $\delta=w-\lambda=7-3=4$ olur ve;

$$A(14, 8, 7) \leq \frac{4 \times 14}{7^2 - 7 \times 14 + 4 \times 14} = 8 = M$$

elde edilir. A matrisinin satırlarını C kodunun kod kelimeleri olarak düşünüldüğünde bu kod maksimal koddur.

4.4. Projektif Geometri $PG(n,2)$ Kullanılarak Kod Oluşturma ve Çözümleme Algoritması

Bu bölümde Projektif geometri kullanarak ençok olabilirlik algoritmasına alternatif çözümleme algoritması verilecektir. Cameron (1996), $PG(2,2)$ 'yi kullanarak maksimal (ayrıca tam) kod oluşturmuştur ve $PG(2,2)$ 'den oluşturulan bu kodun çözümleme algoritmasını da elde etmiştir. Bu bölümde bu algorithmadan faydalananarak $PG(3,2)$ den kod üretilecek, buna ait çözümleme algoritması verilecek ve klasik en çok olabilirlik yöntemine göre karşılaştırılacaktır. Bu çözümleme algoritmasını verebilmek için kodu Projektif geometriyi kullanarak oluşturmak gerekir. Kuruluş yöntemi anlatıldıktan sonra çözümleme algoritması verilecektir. Projektif geometriyi kullanarak oluşturulan maksimal lineer kod aşağıdaki gibidir.

Tanım 2.4 ve Teorem 2.13'den $PG(n,2)$ 'de $2^{n+1}-1$ nokta ve hiperdüzlem, her hiperdüzlem 2^n-1 nokta, her nokta 2^n-1 farklı hiperdüzlemde olduğunu ve her hiperdüzlem çifti $2^{n-1}-1$ farklı noktada kesişir. $PG(n,2)$ den oluşturulacak kodun toplam 2^{n+2} tane kod kelimesi vardır. Bu kod kelimelerin uzunluğu $2^{n+1}-1$, minimum uzaklığı 2^n-1 ve dolayısıyla düzeltebileceği maksimum hata sayısı $2^{n-1}-1$ olacaktır.

Bu bilgiler ışığında $PG(n,2)$ den kod şu şekilde oluşturulur. $2^{n+1}-1$ kod kelime 2^n-1 ağırlığında ve $2^{n+1}-1$ kod kelime 2^n ağırlığındadır. Diğer iki kod kelime ise $2^{n+1}-1$ ve 0 ağırlığındadır yani tümü 1 ve tümü 0 olan kod kelimelerdir.

2^n-1 ağırlığındaki kod kelimelerdeki 1'lerin yerleri projektif geometrideki hiperdüzlemlerin noktalarının koordinatlarından oluşur. Benzer şekilde, 2^n ağırlığındaki kod kelimelerdeki 0'lerin yerleri projektif geometrideki hiperdüzlemlerin noktalarının koordinatlarından meydana gelir.

Çizelge 4.2. Çözümleme algoritması

1'lerin sayısı	Çözümleme kuralı
0	Hata yoktur. Hiçbir değişiklik yapılmaz.
1	1'in bulunduğu yer hatadır. $1 \rightarrow 0$
2	Bu iki noktanın bulunduğu doğrudaki 3.ncü nokta bulunur ve $0 \rightarrow 1$ yapılır
3	Eğer bir doğru üzerinde ise hata yoktur. Aksi halde 0'ların yerine bakılır. İçerdiği doğru dışında kalan nokta hatadır ve $0 \rightarrow 1$ yapılır
4	0'ların yerine 1, 1'lerin yerine 0 yazılarak ilk 4 adımdaki işlemlerden biri yapılır.
5	
6	
7	

Bu bölümün geri kalanında ise Gökpınar (2007)'de $PG(3,2)$ den elde edilen kodu üretip çözümleme algoritması verilecektir. İlk önce $PG(3,2)$ verilsin.

Çizelge 4.3. $PG(3,2)$ 'nin hiperdüzlemleri

	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15
X1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
X2	1	1	1	0	0	0	0	1	1	1	1	1	0	0	0
X3	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1
X4	1	0	0	1	1	0	0	1	1	0	0	0	1	1	0
X5	1	0	0	1	1	0	0	0	0	1	1	1	0	0	1
X6	1	0	0	0	0	1	1	1	1	0	0	0	0	0	1
X7	1	0	0	0	0	1	1	0	0	1	1	1	1	1	0
X8	0	1	0	1	0	1	0	1	0	1	1	0	1	0	0
X9	0	1	0	1	0	1	0	0	1	0	0	1	0	1	1
X10	0	1	0	0	1	0	1	1	0	1	1	0	0	1	1
X11	0	1	0	0	1	0	1	1	1	0	0	1	1	0	0
X12	0	0	1	0	1	1	0	0	1	1	0	0	0	1	0
X13	0	0	1	0	1	1	0	1	0	0	1	1	1	0	1
X14	0	0	1	1	0	0	1	0	1	1	0	0	1	0	1
X15	0	0	1	1	0	0	1	1	0	0	1	1	0	1	0

Çizelge 4.3'de verilen $PG(3,2)$ 'nin düzlemlerinin noktaları aşağıdaki gibidir.

Çizelge 4.4. $PG(3,2)$ 'nin düzlemleri

Düzlem	Noktalar
1	1,2,3,4,5,6,7
2	1,2,3,8,9,10,11
3	1,2,3,12,13,14,15
4	1,4,5,8,9,14,15
5	1,4,5,10,11,12,13
6	1,6,7,8,9,12,13
7	1,6,7,10,11,14,15
8	2,4,6,8,10,13,15
9	2,4,6,9,11,12,14
10	2,5,7,8,10,12,14
11	2,5,7,9,11,13,15
12	3,4,7,8,11,13,14
13	3,4,7,9,10,12,15
14	3,5,6,8,11,12,15
15	3,5,6,9,10,13,14

Buradan elde edilecek kodun 15 kod kelimesi 7 ağırlığında ve 15 kod kelimesi 8 ağırlığındadır. Diğer iki kod kelime ise 15 ve 0 ağırlığındadır yani tümü 1 ve tümü 0 olan kod kelimelerdir. Dolayısıyla elde edilecek kod Çizelge 4.5'deki gibi olacaktır. Bu kodun parametreleri $M=32$, $n=15$, $d=7$ şeklindedir.

Çizelge 4.5. $PG(3,2)$ den elde edilen kod

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	1	1	1	1	0	0	1	1	0
3	0	1	0	0	1	0	1	1	0	1	0	1	0	1
4	0	0	1	0	1	1	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	1	1	0	1
6	1	0	0	1	1	0	0	1	1	0	0	0	0	1
7	0	1	0	0	1	0	1	0	1	0	1	0	1	0
8	1	1	1	1	1	1	1	0	0	0	0	0	0	0
9	1	1	1	0	0	0	0	0	0	0	0	1	1	1
10	0	1	0	1	0	1	0	1	0	1	0	0	1	0
11	1	0	0	1	1	0	0	0	0	1	1	1	1	0
12	0	0	1	0	1	1	0	0	1	1	0	0	1	1
13	0	0	1	1	0	0	1	1	0	0	1	0	1	1
14	0	0	1	1	0	0	1	0	1	1	0	1	0	0
15	1	1	1	0	0	0	0	1	1	1	1	0	0	0
16	1	0	0	0	0	1	1	0	0	1	1	0	0	1
17	0	1	1	1	1	0	0	0	0	1	1	0	0	1

Çizelge 4.5. (Devam) $PG(3,2)$ den elde edilen kod

18	1	0	1	1	0	1	0	0	1	0	1	0	1	0	1
19	1	1	0	1	0	0	1	0	1	1	0	0	1	1	0
20	1	0	1	0	1	0	1	1	0	1	0	0	1	0	1
21	0	1	1	0	0	1	1	0	0	1	1	1	1	0	0
22	1	0	1	1	0	1	0	1	0	1	0	1	0	1	0
23	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
24	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
25	1	0	1	0	1	0	1	0	1	0	1	1	0	1	0
26	0	1	1	0	0	1	1	1	1	0	0	0	0	1	1
27	1	1	0	1	0	0	1	1	0	0	1	1	0	0	1
28	1	1	0	0	1	1	0	0	1	1	0	1	0	0	1
29	1	1	0	0	1	1	0	1	0	0	1	0	1	1	0
30	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
31	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0
32	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bu koda ait çözümleme kuralı ise aşağıdaki gibi olacaktır.

Gelen vektörün tümü 0 ise hiçbir hata olmamış demektir ve kod kelime 0 vektörü olarak çözülür. Benzer şekilde, 1,2 ve 3 tane 1 varsa, $d=3$ olduğundan, bu kod kelimeler 0 olarak çözülür.

Eğer 4 tane 1 varsa ve bunlar $PG(3,2)$ 'nin düzlemleri üzerindeki noktalar ise diğer 3 nokta bulunur ve buradaki 0 lar yerine 1 yazılarak düzeltilir. Bu üç nokta hatadır. Eğer bu 4 nokta aynı düzlem üzerinde değilse 3ten fazla hata meydana gelmiştir ve bu vektör çözülmez. Yeniden gönderim istenir.

Eğer 5 tane 1 varsa ve bu noktalar bir düzlem üzerinde ise düzlem üzerindeki diğer 2 nokta bulunur. Bu noktalardaki 0'lar 1 yapılarak vektör çözülür. Eğer bu noktalar düzlem üzerinde değilse 0'lara bakılır 0lar bir düzlemi içeriyorsa düzlemin dışında kalan 0'ların yerine 1 konulara düzeltme yapılır. Bu koşullardan ikisi de sağlanmıyorsa bu vektör düzeltilemez. Yeniden gönderim istenir.

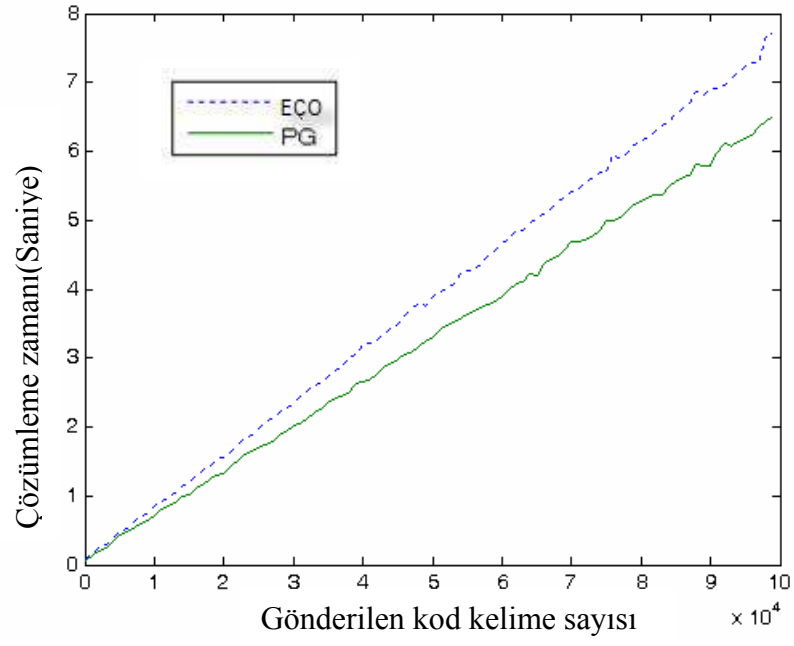
Eğer 6 tane 1 varsa ve bu noktalar bir düzlem üzerinde ise düzlem üzerindeki diğer 1 nokta bulunur. Bu noktalardan 5'i doğru üzerinde 1 tanesi değilse, buradaki 1, 0

olarak düzeltilir ve düzlemin sahip olduğu diğer 2 nokta da 1 olarak düzeltilir. Eğer bu noktalar düzlem üzerinde değilse 0'lara bakılır. 0'lar bir düzlemi içeriyorsa düzlemin dışında kalan 0'ların yerine 1 konularak düzeltme yapılır. Bu koşullardan üçü de sağlanmıyorsa bu vektör düzeltilemez. Yeniden gönderim istenir.

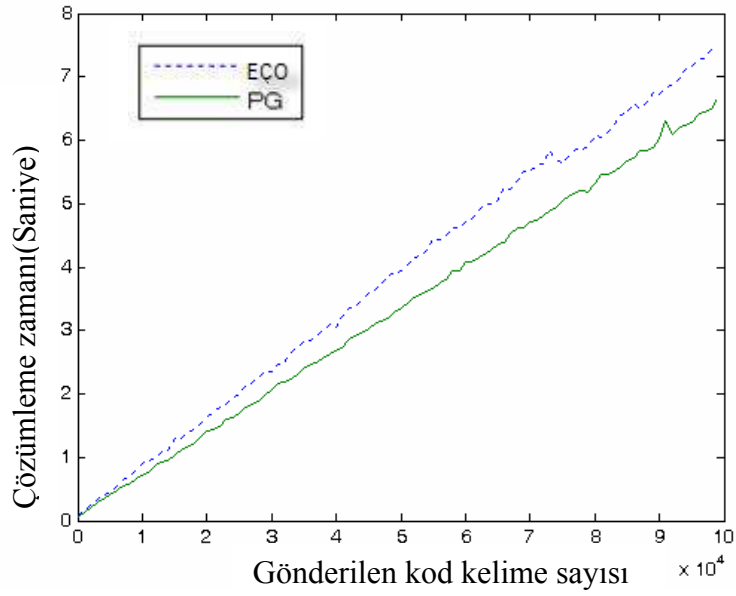
Eğer 7 tane 1 varsa ve bu noktalar bir düzlem oluşturuyorsa bu gönderim doğrudur. Eğer 6 tanesi bir düzlem üzerinde ve biri değilse düzlem üzerindeki diğer nokta bulunur ve bu nokta düzeltilir. Eğer bu noktalar düzlem üzerinde değilse 0 lara bakılır 0lar bir düzlemi içeriyorsa düzlemin dışında kalan 0ların yerine 1 konulara düzeltme yapılır. Eğer 0'lardan 6'sı bir düzlem oluşturuyor ise düzlem üzerindeki diğer nokta bulunarak 0 yerine 1 konur. Bu nokta ve geri kalan nokta hatadır. Bu koşullardan dördü de sağlanmıyorsa bu vektör düzeltilemez. Yeniden gönderim istenir.

Eğer 0'ların sayısı 8 veya daha fazla ise 0'lar ile 1'ler yer değiştirilerek aynı işlem yapılır.

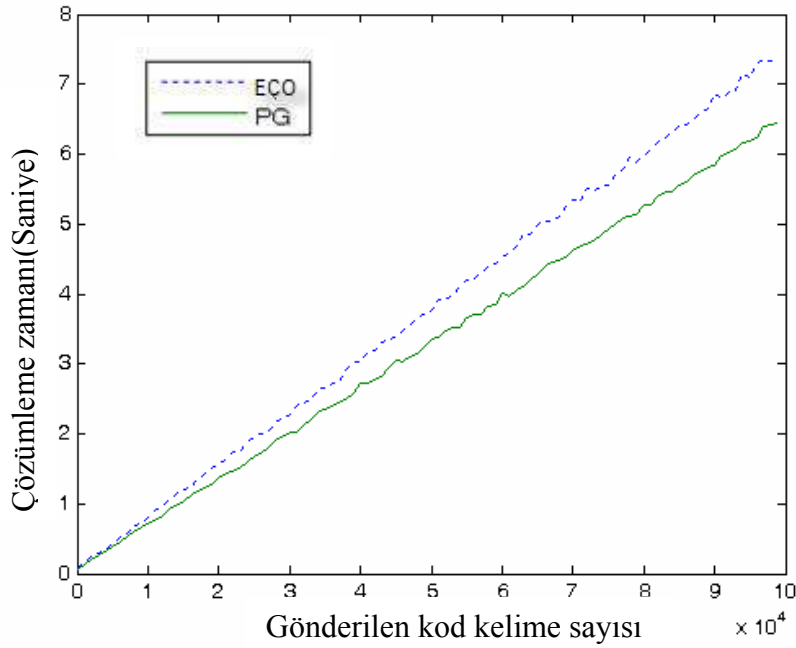
Bu bölümün geri kalanında En çok olabilirlik(EÇO) ve Projektif Geometri(PG) çözümleme yöntemlerini farklı gönderilen kod kelime sayısı ve hata oranları göz önüne alınarak karşılaştırma yapılacaktır. Dikkate alınan kod kelime sayıları 1000(1000)10000 ve hata oranları 0.05, 0.10, 0.15 şeklindedir. Çizelge 4.5 dikkate alınarak, 00...0 kelimesinin karşılığı 1 sayısı ve bu şekilde devam ederek 11...1 kelimesinin karşılığı 32 sayısı olarak alınacaktır. 1-32 arasında rasgele sayılar üretilerek ve karşılık gelen kod kelimeler verilen hata oranları dikkate alınarak gönderilecektir ve bunları çözümleme zamanları Şekil 4.2-Şekil 4.4'de olduğu gibidir.



Şekil 4.2. Hata oranı 0.05 iken çözümleme zamanları



Şekil 4.3. Hata oranı 0.10 iken çözümleme zamanları



Şekil 4.4. Hata oranı 0.15 iken çözümleme zamanları

Şekil 4.2- Şekil 4.4'den görüldüğü gibi *PG* kullanılan kodun çözümleme zamanları ile EÇOyu kullanan kodun çözümleme zamanlarına göre belirgin bir şekilde hızlıdır. Bu fark gönderilen kod kelime sayısı arttıkça yükselmektedir. Aralarındaki fark yaklaşık olarak %20 civarındadır. 0.05 hata oranında doğru çözümleme oranı 0.995, 0.10 hata oranında doğru çözümleme oranı 0.95 ve 0.15 hata oranında is doğru çözümleme oranı ise 0.85 civarındadır. Doğru çözümleme oranını %95 olarak kabul edilirse bu kod %10 hata oranına kadar kullanılabilir.

Bu algoritmanın Hadamard matrislerinden oluşturulan maksimal kodlar için yeniden düzenlenmesi ve en çok olabilirlik algoritmasına göre karşılaştırılması 7.nci bölümde verilmiştir.

5. KARŞILAŞTIRMA

Bu bölümde, Hadamard 2-tasarımlardan elde edilen 3 farklı maksimal kod ve ikili ortogonal Latin kare tam setinden oluşturulan Afin yapısındaki çözülebilir tasarımımdan elde edilen optimal kodlar ençok olabilirlik çözümlemesi yöntemiyle karşılaştırılmış ve çeşitli boyutlardaki dijital fotoğraflar üzerine uygulaması yapılmıştır.

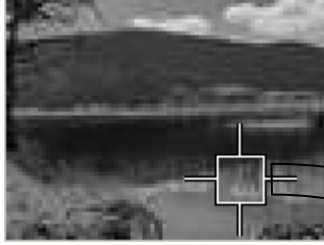
Kullanılan dijital fotoğraflar 256 renk gri tonlamalı fotoğraflardır. 80×60 , 120×90 ve 160×120 piksel boyutlu fotoğraflar ele alınmıştır.

Bir dijital fotoğraf piksellerden oluşur. 256 renk gri tonlama dijital fotoğraflarda siyah ile beyaz arasında toplam 256 renk tonu dikkate alınır. Her bir renk tonu bir sayıya karşılık gelir. 0 en koyu siyaha karşılık gelirken 255 en açık beyaza karşılık gelmektedir. Aradaki tonlar 0 ile 255 sayıları arasında değerler almaktadır. Bu numaralandırmaya ilişkin çizelge aşağıda verilmiştir.

Çizelge 5.1. Renk tonları ve bunlara ait sayılar

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71
72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87
88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103
104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119
120	121	122	123	124	125	126	127
128	129	130	131	132	133	134	135
136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151
152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167
168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183
184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199
200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215
216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231
232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247
248	249	250	251	252	253	254	255

Bu çizelgeye göre bir dijital fotoğraf için örnek Şekil 5.1(a) ve Şekil 5.1(b) verilebilir.



(a)

42	39	36	37	39	39	36	31	30	33	37	42
39	43	41	42	45	46	32	49	51	38	73	69
45	53	52	55	65	82	64	82	67	53	83	79
70	80	80	69	80	106	76	96	68	64	95	91
97	102	99	83	89	110	68	86	61	71	111	112
106	104	98	97	102	112	74	87	72	80	124	131
109	107	100	103	113	113	98	99	97	81	122	133
111	115	112	121	132	114	121	109	125	84	122	135
107	117	119	147	156	120	137	116	147	95	134	147
114	118	120	138	151	153	143	141	147	142	128	128
117	121	124	112	124	127	121	122	131	130	119	105
114	116	118	107	115	114	104	103	109	107	98	101

(b)

Şekil 5.1. Bir dijital fotoğrafın pikselleri ve buna karşılık gelen sayılar (a) Bir dijital fotoğraf (b) Dijital fotoğrafın pikselleri ve bunlara karşılık gelen sayılar

Bir 256 renk gri tonlama dijital fotoğrafta 256 renk tonu olduğundan kullanılacak koddaki kod kelime sayısı toplam 256 olmak zorundadır. Bu durumda kodlar bu koşula göre seçilmelidir. Hadamard tasarımından elde edilen ve C_1 , C_2 ve C_3 olarak nitelendirilecek kodların üçü de bu koşulu sağlamaktadır. Bu kodların seçilmesinin

bir diğer sebebi maksimal olmalarıdır. Yine benzer şekilde elde edilen ikili ortogonal Latin kare'den elde edilen Afin Geometri yapısındaki çözülebilir tasarımından elde edilen ve C_4 diye adlandırılan optimal kodda bu koşul sağlanmaktadır. Bu durumda tüm kodlarda kod kelime sayısı $M=256$ olacağından kullanılacak tasarımlar ve bunlardan elde edilecek kodlar aşağıdaki gibidir. A , $2-(4\lambda-1, 2\lambda, \lambda)$ ilişki matrisi olsun. C_1 Bölüm 4'te verildiği gibi aşağıdaki gibi oluşturulan $(4\lambda-1, 4\lambda, 2\lambda)$ parametrelili maksimal koddur.

$$C_1 = \begin{pmatrix} 0 \cdots 0 \\ A \end{pmatrix}$$

Bu durumda $n=M=256$ olur. Yani kodun parametreleri $(255, 256, 128)$ şeklinde olacaktır. Bu kodun düzeltebileceği maksimum hata sayısı, $d=128$ olduğundan, $e = \lfloor (128-1)/2 \rfloor = 63$ dir. Bu kod için 256 mertebeli H_{256} Hadamard matrisi kullanılacaktır. Benzere şekilde C_1 ve C_1^c 'den oluşan C_2 koduna ilişkin parametreler $(4\lambda-1, 8\lambda, 2\lambda-1)$ şeklinde olduğundan ve $M=256$ olacağından kod $(127, 256, 63)$ parametrelidir. Burada düzeltebilecek maksimum hata sayısı $e = \lfloor (63-1)/2 \rfloor = 31$ olacaktır. Bu kod için 128 mertebeli H_{128} Hadamard matrisi kullanılacaktır. Benzer şekilde oluşturulacak son kod ise C_1 'deki 0 ile başlayan kod kelimelerin silinmesiyle elde edilen C_3 kodudur. Bu kodun parametreleri $(4\lambda-2, 2\lambda, 2\lambda)$ şeklindedir. $M=256$ olması için bu kodun parametreleri $(510, 256, 256)$ şeklindedir. Bu durumda düzeltebileceği maksimum hata sayısı $e = \lfloor (256-1)/2 \rfloor = 127$ dir.

Son olarak ele alınacak bir diğer kod, yukarıda verilen kodlardan farklı olarak optimal olmakla beraber aynı zamanda ikili kod değildir. q mertebeli Afin düzleme karşılık gelen çözülebilir tasarımından elde edilen bu kod $(q+1, q^2, q)$ parametrelerine sahip q -düzeylidir. Burada $M=256$ olacağından $q=16$ olmalıdır. Elde edilen kod $(17, 256, 16)$ parametrelili 16-düzeylidir. Bu durumda $e = \lfloor (16-1)/2 \rfloor = 7$ olarak elde edilir. Verilen bu bilgiler ışığında kullanılan algoritma kısaca aşağıdaki gibidir.

A) Kodların Oluşturulması

A1) C_1 kodunun oluşturulması

- 1) 256 mertebeli H_{256} hadamard matrisi oluşturulur.
- 2) Oluşturulan H_{256} matrisinin ilk satır ve sütunu silinir.
- 3) Geri kalan matriste $-1 \rightarrow 1$ ve $1 \rightarrow 0$ yazılır. Elde edilen A matrisi $2-(4\lambda-1, 2\lambda, \lambda)$ parametrelili simetrik tasarımıdır.
- 4) A matrisine 0lardan oluşan uygun boyutlu bir satır vektörü eklenir. Teorem 4.1-1'ye göre bu şekilde oluşturulan matrisin satırları C_3 kodunun kod kelimeleridir.

A2) C_2 kodunun oluşturulması

- 1) 128 mertebeli H_{128} hadamard matrisi için A1deki (2)-(4) işlemleri tekrarlanır.
- 2) Teorem 4.1-2'ye göre elde elden C_1 ve tümleyeni C_1^T matrislerinden oluşan matrisin satırları C_2 kodunun kod kelimeleridir.

A3) C_3 kodunun oluşturulması

- 1) 512 mertebeli H_{512} hadamard matrisi için A1deki (2)-(4) işlemleri tekrarlanır.
- 2) Teorem 4.1-3'ye göre elde edilen C_1 matrisinden 0la başlayan elemanlar silinerek elde edilen matrisin satırları C_3 kodunun kod kelimeleridir.

A4) C_4 kodunun oluşturulması

- 1) 16. mertebeden ikili ortogonal tam seti MAPLE paket programı kullanılarak oluşturulur ve gerekli dönüştürmeler kullanılarak MATLAB paket programına aktarılır.
- 2) Buradan oluşturulan Afin düzlemin doğruları (Aynı zamanda Çözülebilir tasarımın blokları) ise Bölüm 2.6.1'de verildiği gibi oluşturulur.

3) Teorem 4.5'te verildiği gibi her paralel sınıf için 100...0 yerine 0, 010...0 yerine 1 vb. yazılarak oluşturulan matrisin satırları C_4 kodunun kod kelimelerini oluşturur.

B) Dijital Fotoğraf Piksellerinin Kod kelimelere Dönüştürülmesi (Şifreleme, Encoding)

- 1) İlk önce $a \times b$ piksel boyutlu dijital fotoğrafın toplam $a \cdot b$ pikselin renk tonlarına ait sayılar atanır.
- 2) Her kod kelime 0–255 arası bir sayıya karşılık geldiğinden 1. adımda ataman sayılara karşılık gelen kod kelimelere dönüştürülür.
- 3) Son olarak elimizde $a \cdot b$ tane kod kelimedenden oluşan bir mesaj oluşur.

C) Gönderme

C1) C_1, C_2, C_3 kodları için;

- 1) Bu kodlar ikili olduklarından mesajın kod kelimelerinin her biti için bernoulli dağılımından $p=1$ - hata oranı kullanılarak hatalar vektörü oluşturulur.
- 2) Oluşturulan bu hata vektörleri mod 2'ye göre kod kelimeler ile toplanır. Elde edilen yapı kanaldan geçen mesajın bozulmuş halidir.

C2) C_4 kodu için;

- 1) Bu kodlar ikili olmadıklarından mesajın kod kelimelerinin her biti için deneme sayısı 1 olan çokterimli dağılımından $p_0=1$ -hata oranı $p_1=\dots=p_{15}=(\text{hata oranı})/15$ kullanılarak hatalar vektörü oluşturulur.
- 2) Oluşturulan bu hata vektörleri mod 16'ye göre kod kelimeler ile toplanır. Elde edilen yapı kanaldan geçen mesajın bozulmuş halidir.

D) Çözümleme

- 1) Bozulmuş mesajın her bir vektörü (gönderilirken bozulmuş kod kelimeler) için tüm kod kelimeler arasındaki hamming uzaklığı bulunur.
- 2) Bulunan uzaklıkların en küçüğü tespit edilir
- 3) Bu uzaklık minimum uzaklık d' den küçükse o kod kelime olarak çözülür.
- 4) Bu uzaklık minimum uzaklık d' den büyükse o vektör çözülmez.
- 5) Çözülünen vektörlerin kod kelimelerine karşılık gelen sayılar atanır.
- 6) 1-5 adımlarındaki işlemler bozulmuş mesajdaki tüm vektörlere uygulanır.
- 7) Elde edilen sayıları koordinatlarına koyarak çözülünen kod oluşturulur.

Bu bilgiler ışığında ilk olarak 80×60 , 120×90 ve 160×120 piksel fotoğrafları incelenir. 80×60 piksel boyutlu gönderilen fotoğraf Şekil 5.2'deki gibi olsun.



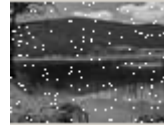
Şekil 5.2. 80×60 piksel boyutunda dijital fotoğraf



(a)



(b)



(c)



(d)

Şekil 5.3. C_1 kodunu kullanarak çözülünen 80×60 piksel boyutunda dijital fotoğraflar (a) bit başına hata oranı 0.05 iken (b) bit başına hata oranı 0.15 iken (c) bit başına hata oranı 0.20 iken (d) bit başına hata oranı 0.25 iken

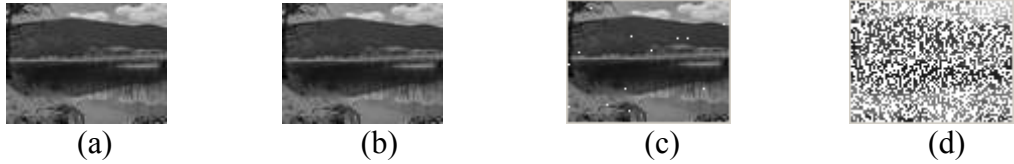
C_1 kodu dikkate alınıp Şekil 5.3(a)- Şekil 5.3(b) incelendiğinde 0.05 ile 0.15 bit başına hata oranında bu kodun dijital fotoğrafı hatasız çözümlediği görülür. Ancak Şekil 5.3(c)'de bit başına hata oranı 0.20'ye çıktığında hatalı sonuçlar verdiği (doğru çözümleme oranı 0.97) görülmektedir. Elde edilen sonuç iyi bir çözümleme oranı sayılmasına rağmen Şekil 5.3(d)'de bit başına hata oranı 0.20'den 0.25'e çıktığında

doğru çözümleme oranı çok fazla düşmüştür (doğru çözümleme oranı 0.51). Bu kod için hata oranı 0.20 civarı olmalıdır, çözümleme zamanı ise 57 saniyedir.



Şekil 5.4. C_2 kodunu kullanarak çözümlenen 80×60 piksel boyutunda dijital fotoğraflar a) bit başına hata oranı 0.05 iken b) bit başına hata oranı 0.15 iken c) bit başına hata oranı 0.20 iken d) bit başına hata oranı 0.25 iken

Benzer şekilde C_2 kodu incelendiğinde Şekil 5.4(a)'da 0.05 bit başına hata oranı verildiğinde fotoğrafı hatasız olarak düzeltilmiştir. Şekil 5.4'(b) için kullanılan 0.15 bit başına hata oranı için çok az (doğru çözümleme oranı 0.9991) hatalı çevirdiği görülmüştür. Fakat Şekil 5.4'(c) oldukça hatalı sonuçlar vermiştir (doğru çözümleme oranı 0.9145). Özellikle bit başına hata oranı 0.25'e çıktığında C_1 kodunda olduğu gibi doğru çözümleme oranı çok fazla düşmektedir (doğru çözümleme oranı 0.51). Bu kod için hata oranı 0.15 civarı olmalıdır. Ayrıca çözümleme zamanı ise 24 saniyedir.



Şekil 5.5. C_3 kodunu kullanarak çözümlenen 80×60 piksel boyutunda dijital fotoğraflar. (a) bit başına hata oranı 0.05 iken (b) bit başına hata oranı 0.15 iken (c) bit başına hata oranı 0.20 iken (d) bit başına hata oranı 0.25 iken

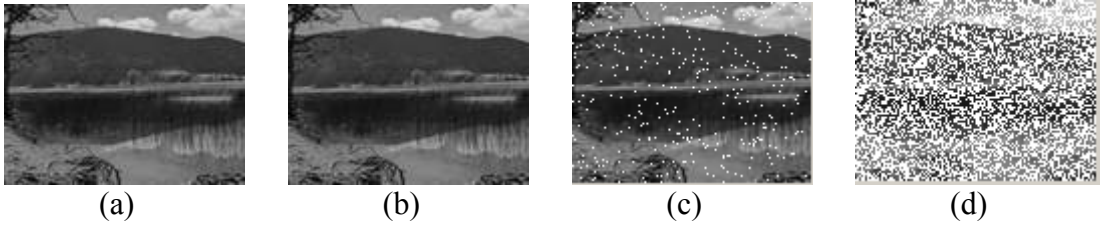
Son olarak C_3 kodunu ele alındığında Şekil 5.3(a)-Şekil 5.3(b) incelenirse 0.05 ile 0.15 bit başına hata oranında bu kodun dijital fotoğrafı hatasız çözümlediği görülür. 0.20 bit başına hata oranında bu kodun dijital fotoğrafı yüksek doğru çözümleme oranıyla çözümlediği gözükmemektedir (doğru çözümleme oranı 0.9973). Fakat diğer iki kodda olduğu gibi bit başına hata oranı 0.25 için doğru çözümleme oranı oldukça düşmektedir. Ayrıca bu kod için çözümleme zamanı 168 saniyedir.

Bu problem için dikkate alınan Hadamard matrisinden üretilen kodlarda bit başına hata oranı 0.05 ve 0.15 iken tatmin edici sonuçlar vermektedir. Bu durumda en hızlı çözümleme yapan kod olan C_2 kodu seçilmesi daha uygun olur. Bit başına hata oranı 0.25 iken; hızlı çözümleme yapılan bir kod isteniyorsa C_1 kodu, daha doğru bir çözümleme yapan bir kod isteniyorsa C_3 kodu tercih edilir. Bit başına hata oranı 0.25 ise üç kod da tercih edilmez.

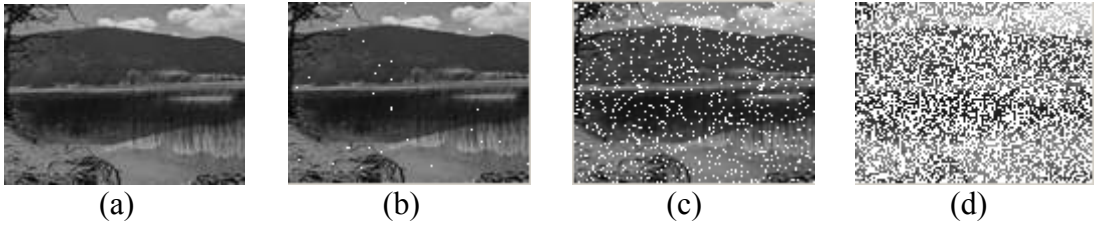
Şekil 5.6'daki 120×90 piksel resim dikkate alınsın.



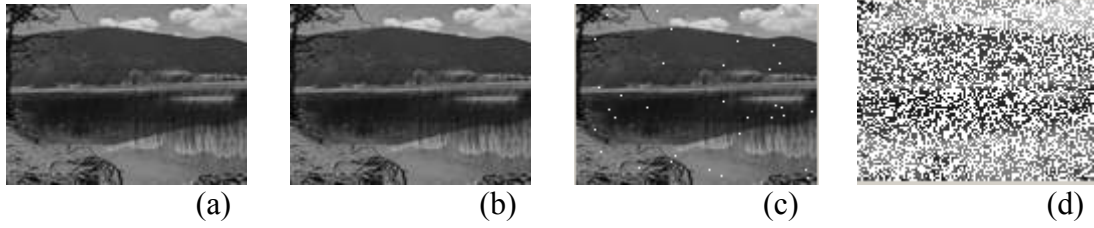
Şekil 5.6. 120×90 piksel boyutunda dijital fotoğraf



Şekil 5.7. C_1 kodunu kullanarak çözümlenen 120×90 piksel boyutunda dijital fotoğraflar. a) bit başına hata oranı 0.05 iken b) bit başına hata oranı 0.15 iken c) bit başına hata oranı 0.20 iken d) bit başına hata oranı 0.25 iken



Şekil 5.8. C_2 kodunu kullanarak çözümlenen 120×90 piksel boyutunda dijital fotoğraflar a) bit başına hata oranı 0.05 iken b) bit başına hata oranı 0.15 iken c) bit başına hata oranı 0.20 iken d) bit başına hata oranı 0.25 iken



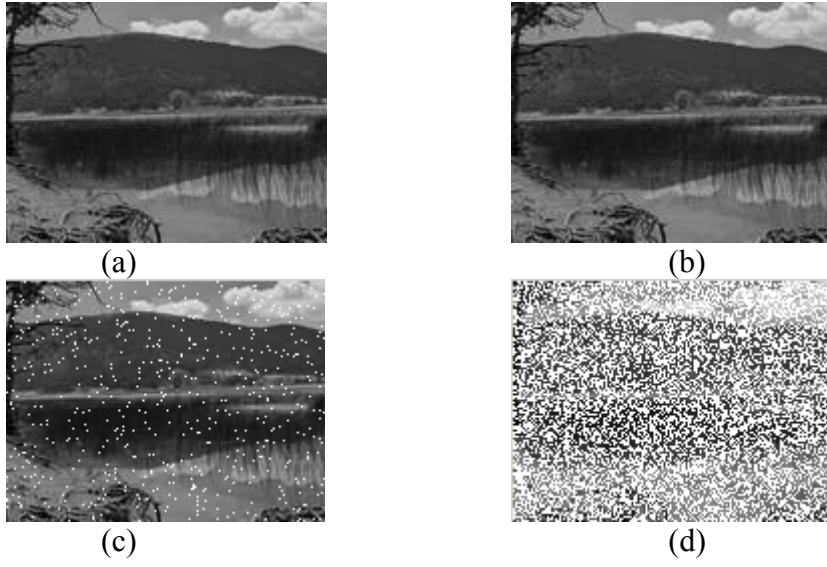
Şekil 5.9. C_3 kodunu kullanarak çözümlenen 120×90 piksel boyutunda fotoğraflar
a) bit başına hata oranı 0.05 iken b) bit başına hata oranı 0.15 iken c) bit başına hata oranı 0.20 iken d) bit başına hata oranı 0.25 iken

Şekil 5.7-Şekil 5.9 dikkate alındığında 80×60 piksel boyutlu fotoğraflar için var olan sonuçlara benzer sonuçlar vermektedir. Bu kodlar için yine bit başına hata oranı 0.05 ve 0.15 iken doğru çözümleme oranları şekillerden de görüleceği gibi çok iyidir. Bu durumda diğerleri içinde en hızlı (C_2 kodu için 54.56 saniye; C_3 için 371.84 saniye, C_1 için ise 126.10 saniye) kod C_2 kodudur. Bit başına hata oranı 0.25 iken şekillerden de görüleceği gibi üç kod da iyi sonuç vermediğinden hiçbir kod da tercih edilmez.

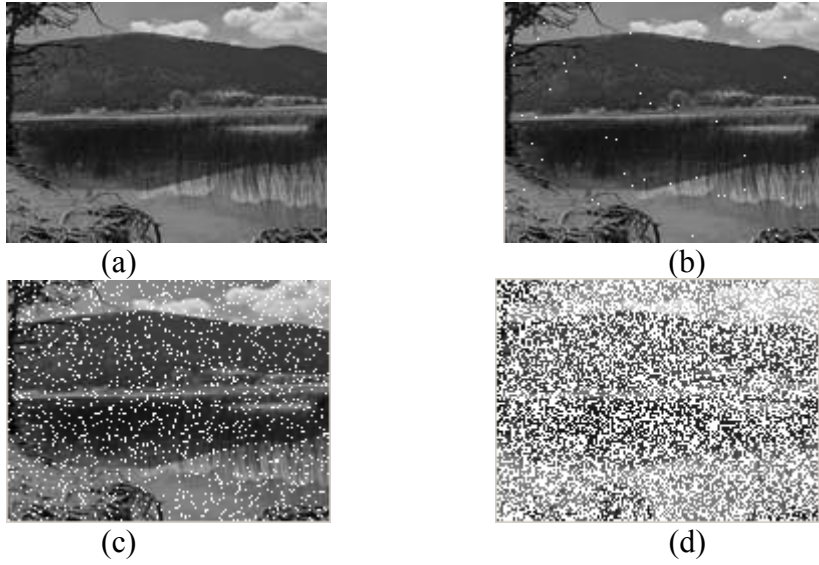
Şekil 5.10'daki 160×120 piksel resim dikkate alınsın.



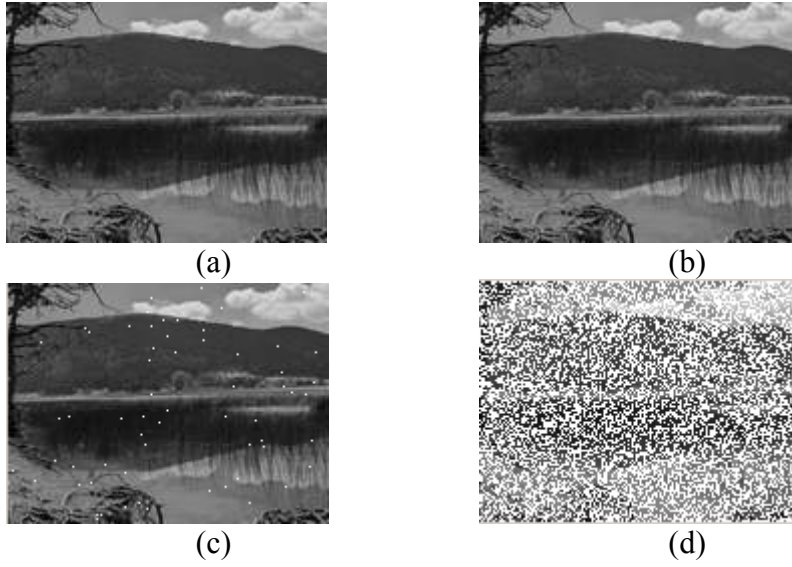
Şekil 5.10. 160×120 piksel boyutunda dijital fotoğraf



Şekil 5.11. C_1 kodunu kullanarak çözümlenen 160×120 piksel boyutunda fotoğraflar
a) bit başına hata oranı 0.05 iken b) bit başına hata oranı 0.15 iken c) bit başına hata oranı 0.20 iken d) bit başına hata oranı 0.25 iken



Şekil 5.12. C_2 kodunu kullanarak çözümlenen 160×120 piksel boyutunda fotoğraflar
a) bit başına hata oranı 0.05 iken b) bit başına hata oranı 0.15 iken c) bit başına hata oranı 0.20 iken d) bit başına hata oranı 0.25 iken



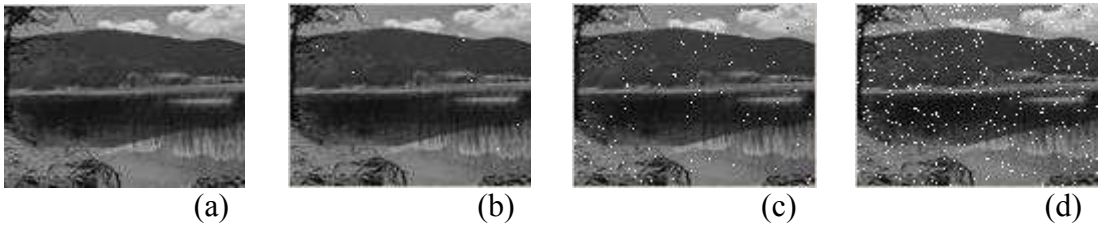
Şekil 5.13. C_3 kodunu kullanarak çözümlenen 160×120 piksel boyutunda fotoğraflar
a) bit başına hata oranı 0.05 iken b) bit başına hata oranı 0.15 iken c) bit başına hata oranı 0.20 iken d) bit başına hata oranı 0.25 iken

Şekil 5.11-Şekil 5.13 dikkate alındığında 80×60 ve 120×90 piksel boyutlu fotoğraflar için olan yorumların benzerleri söylenebilir. Bu üç kod için bit başına hata oranları düşükken üç kod da iyi sonuçlar verir fakat C_2 (C_2 kodu için 104.12 saniye; C_3 için 750.48 saniye, C_1 için ise 237.46 saniye) hepsinden daha hızlı olduğundan C_2 tercih edilir. Bit başına hata oranı artıkça üç kodun doğru oranları düştüklerinden üç kodda tercih edilmez. Özellikle bit başına hata oranı 0.25 iken üç kod için iyi sonuçlar çıkmadığını rahatlıkla görebiliriz

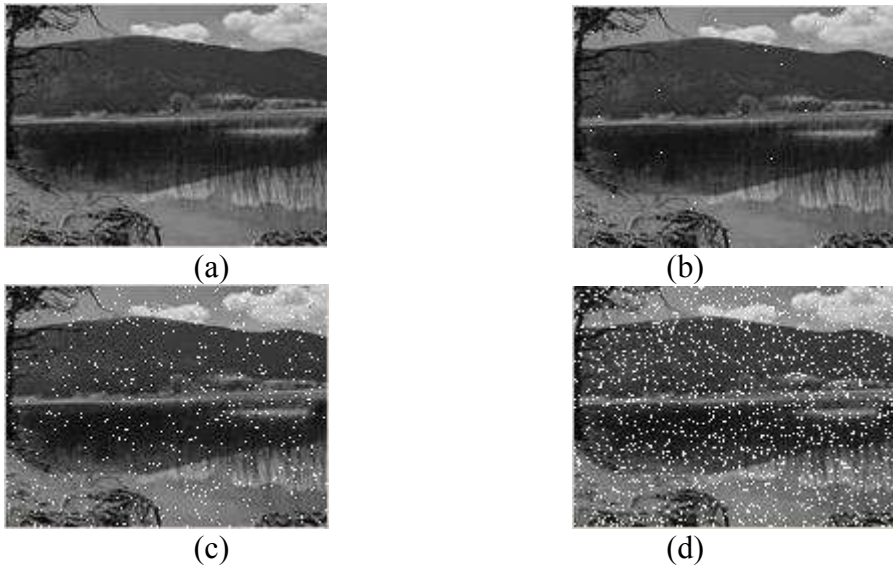
Farklı boyuttaki her fotoğraf incelendiğinde C_2 'in 0.05 ile 0.15'de daha uygun sonuçlar verdiği görülmüştür. Fakat bit başına hata oranı 0.20 iken kodu kullananın tercihinin kalmıştır. Bit başına hata oranı 0.25 iken üç kodda anlamlı sonuç vermeyecektir.



Şekil 5.14. C_4 kodunu kullanarak çözümlenen 80×60 piksel boyutunda dijital fotoğraflar a) bit başına hata oranı 0.05 iken b) bit başına hata oranı 0.15 iken c) bit başına hata oranı 0.20 iken d) bit başına hata oranı 0.25 iken



Şekil 5.15. C_4 kodunu kullanarak çözümlenen 120×90 piksel boyutunda fotoğraflar a) bit başına hata oranı 0.05 iken b) bit başına hata oranı 0.15 iken c) bit başına hata oranı 0.20 iken d) bit başına hata oranı 0.25 iken

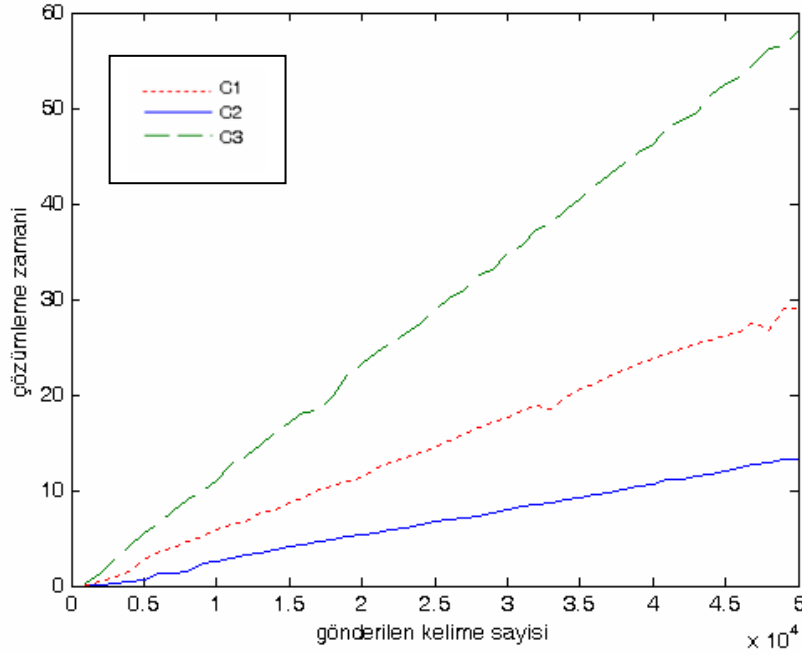


Şekil 5.16. C_4 kodunu kullanarak çözümlenen 160×120 piksel boyutunda fotoğraflar a) bit başına hata oranı 0.05 iken b) bit başına hata oranı 0.15 iken c) bit başına hata oranı 0.20 iken d) bit başına hata oranı 0.25 iken

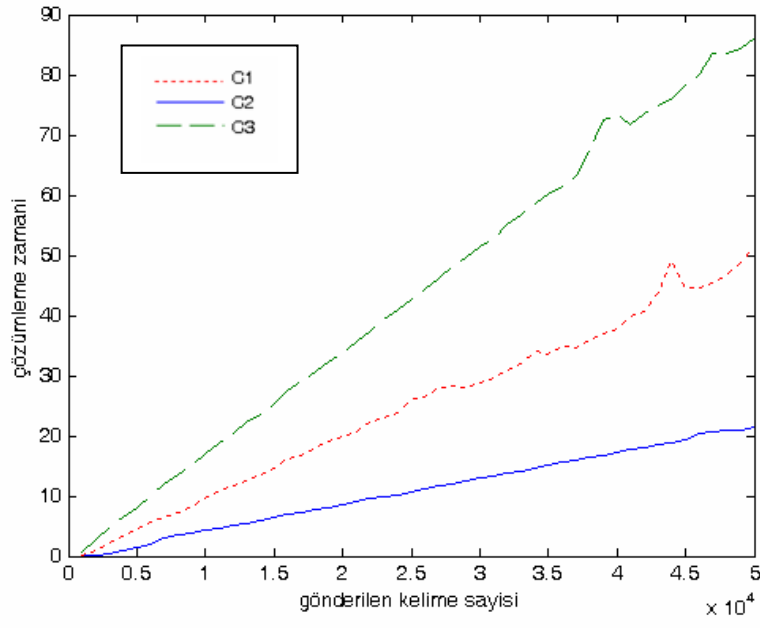
Şekil 5.14(a)-Şekil 5.16(a) fotoğrafları dikkate alındığında bit başına hata oranı 0.05 iken doğru çözümleme oranı diğer üç kod gibi 1dir. Bit başına hata oranı 0.15 iken doğru çözümleme oranı biraz düşmekle beraber oldukça tatmin edicidir (doğru

çözümleme oranı=0.9901). Yine hata oranı 0.20 için doğru çözümleme oranı 0.975 civarındadır. Bu kodun diğer kodlar arasındaki göze çarpan en büyük farklılığı bit başına hata oranının 0.25 olduğu zaman gerçekleşmektedir. Bu durumda diğer kodlar 0.50 civarında doğru çözümleme oranı verirken bu kod 0.93 civarında doğru çözümleme oranı vermektedir. Bu ise kimi durumlarda oldukça tatminkârdır. Aynı zamanda bu kod için zamanlar 80×60 için 8 saniye, 120×90 için 10 saniye 160×120 için ise 13 saniye civarındadır. Bu zamanlar ise her üç kodun zamanlarından daha iyidir. Böylece C_4 'ün diğer 3 koddan daha uygun sonuçlar verdiği söylenebilir.

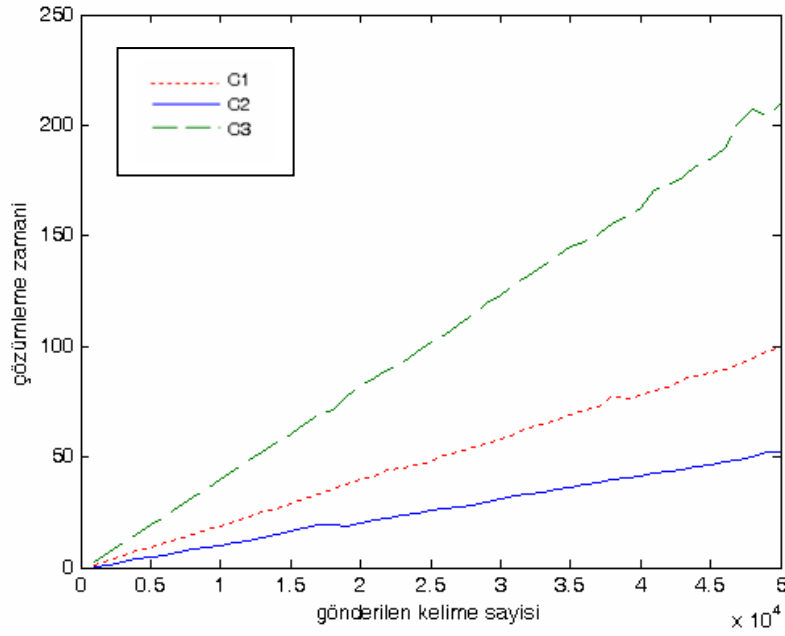
Fakat burada 256 renk gri tonlama fotoğrafları dikkate alındığından sadece $M=256$ için karşılaştırma yapıldı. Bu özel durumu ortadan kaldırmak için farklı M değerleri ve farklı bit başına hata oranları için bu kodları karşılaştırmak gerekir. Geri kalan kısımda gönderilen kod kelime sayısı için 1000:1000:50000 değerleri, bit başına hata oranı için ise 0.01:0.01:0.50 değerleri dikkate alınmış ayrıca Hadamard matrisinden üretilen kodlar (C_1 , C_2 , C_3) için $M=16$, 24, 32, 48, 64 değerleri, çözülebilir tasarımdan oluşturulan C_4 kodu için ise $M=16$, 64, 256 olan kodlar dikkate alınmıştır.



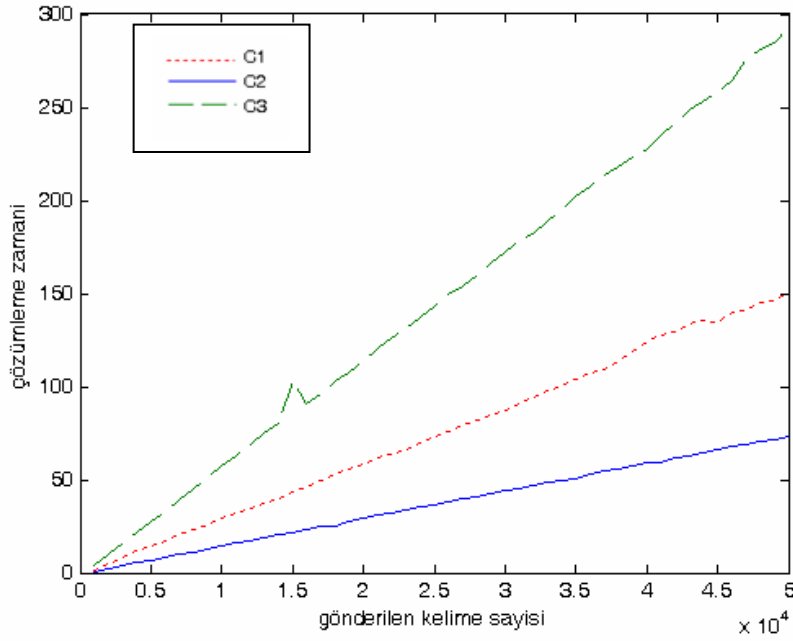
Şekil 5.17. $M=16$ iken C_1 , C_2 ve C_3 için çözümleme zamanlarının karşılaştırılması



Şekil 5.18. $M=24$ iken C_1 , C_2 ve C_3 için çözümleme zamanlarının karşılaştırılması

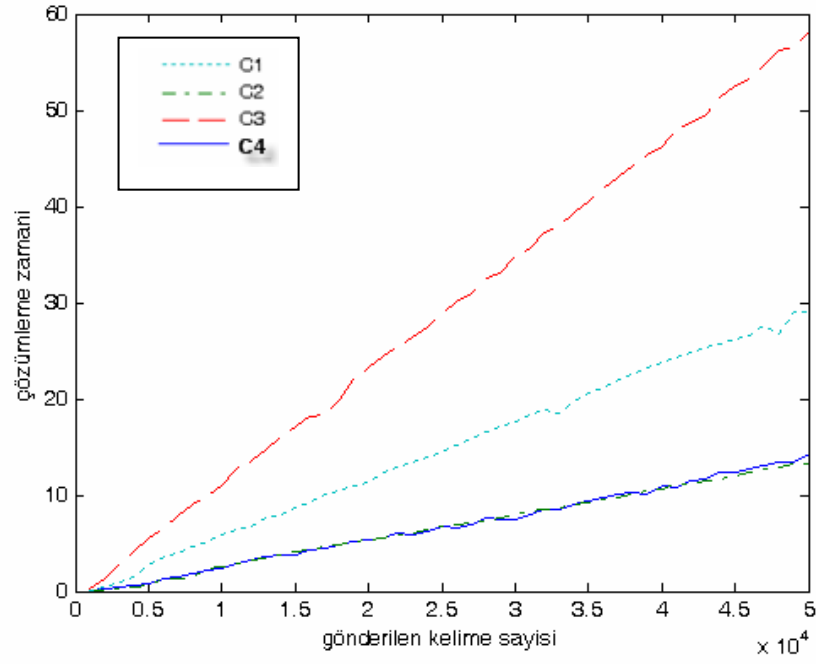


Şekil 5.19. $M=48$ iken C_1 , C_2 ve C_3 için çözümleme zamanlarının karşılaştırılması

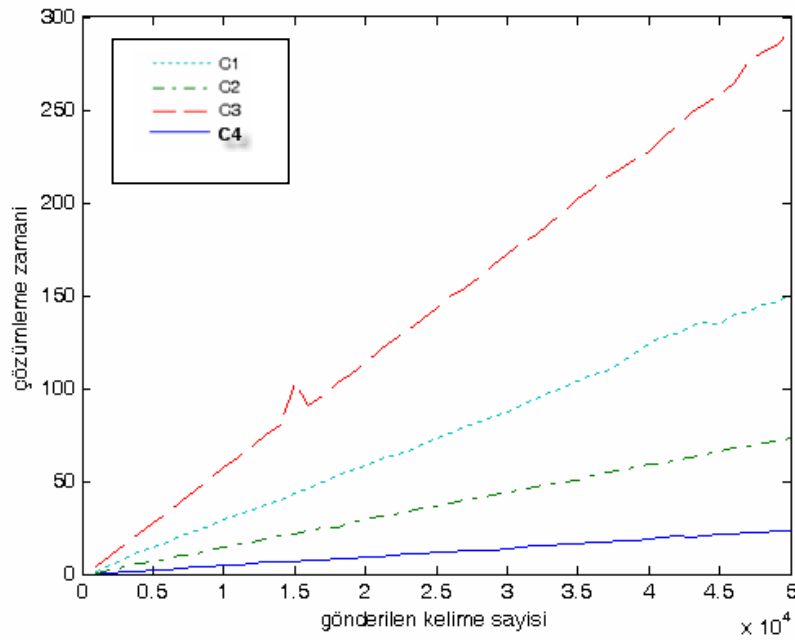


Şekil 5.20. $M=64$ iken C_1 , C_2 ve C_3 için çözümleme zamanlarının karşılaştırılması

Şekil 5.17–Şekil 5.20 dikkate alındığında gönderilen kod kelime sayısı az iken 3 kod kelime sayısı arasında çözümleme zamanları açısından fark oldukça azdır. Fakat her kod kelime sayısı M için gönderilen kelime sayısı artıkça C_2 kodu C_3 kodundan 5–6 kat kadar, C_1 kodundan ise 2–3 kat daha hızlı çözümleme yapmaktadır. Ayrıca C_1 kodu C_2 kodundan 2-2.5 kat daha hızlı olduğu görülmektedir.

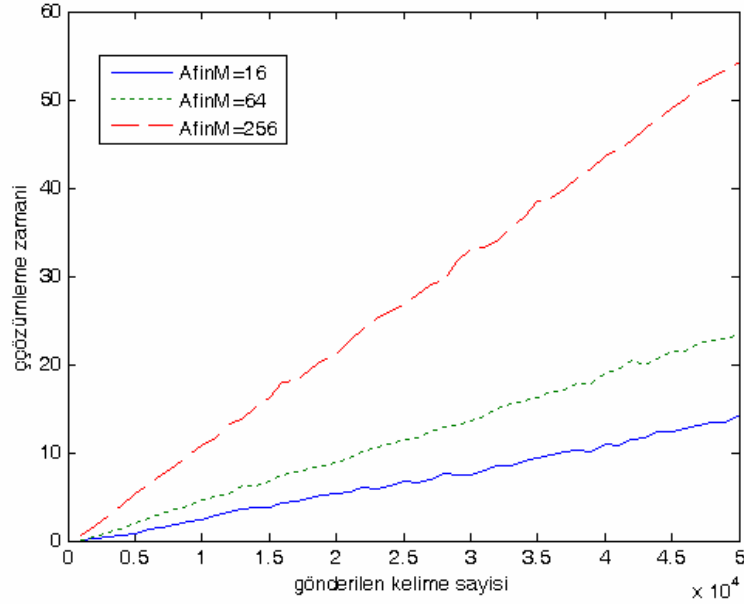


Şekil 5.21. $M=16$ iken C_1 , C_2 ve C_3 ve C_4 için çözümleme zamanlarının karşılaştırılması

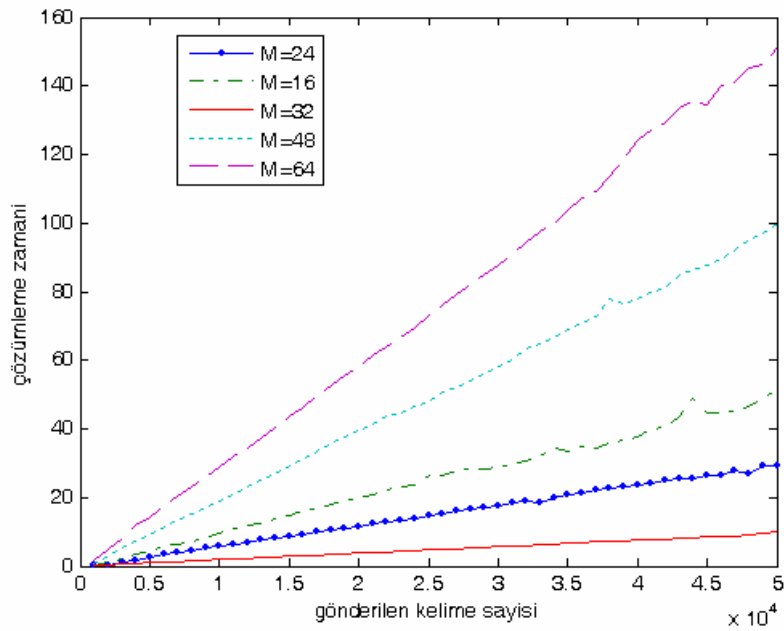


Şekil 5.22. $M=64$ iken C_1 , C_2 ve C_3 ve C_4 için çözümleme zamanlarının karşılaştırılması

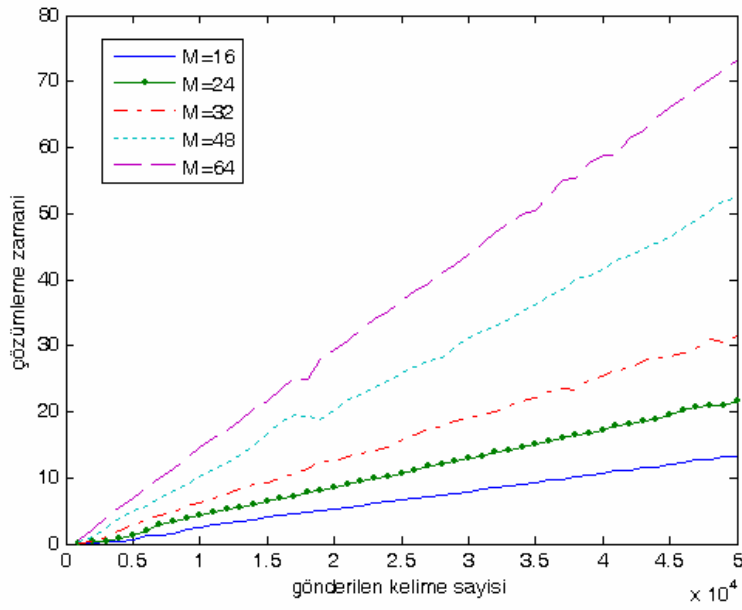
Şekil 5.21 incelendiğinde C_4 kodunun C_3 ve C_1 ten daha iyi sonuçlar vermesine rağmen C_2 ile aynı sonuçları vermektedir. Fakat Şekil 5.22 incelendiğinde C_2 ile C_4 arasında C_4 'ün lehine belirgin bir fark olduğu görülmektedir.



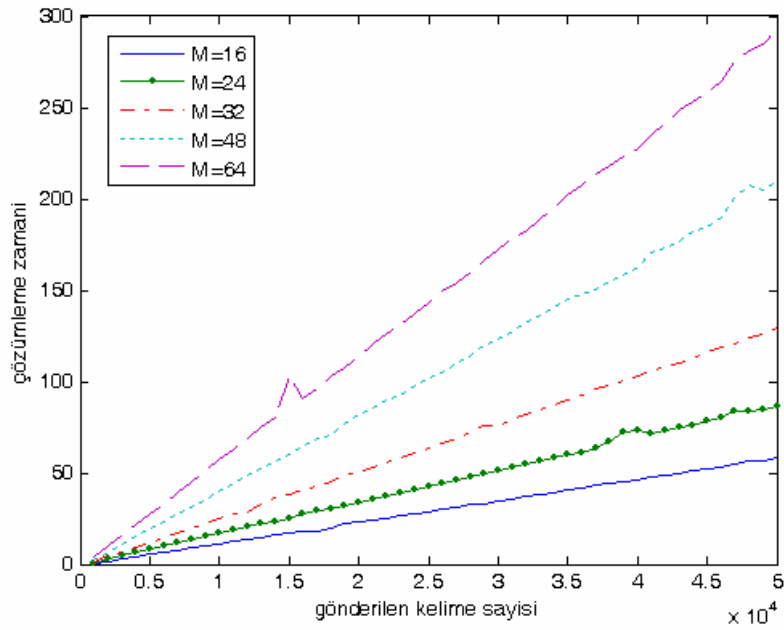
Şekil 5.23. C_4 'in $M=16, 64, 256$ iken çözümüleme zamanlarının karşılaştırılması



Şekil 5.24. C_1 'in $M=16, 24, 32, 48, 64$ iken çözümüleme zamanlarının karşılaştırılması

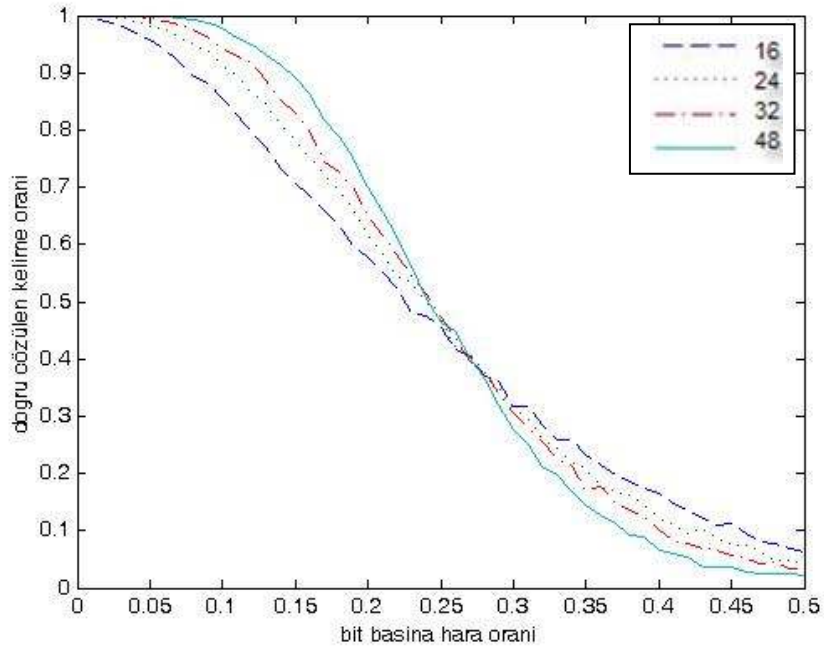


Şekil 5.25. C_2 'in $M=16, 24, 32, 48, 64$ iken çözümüleme zamanlarının karşılaştırılması



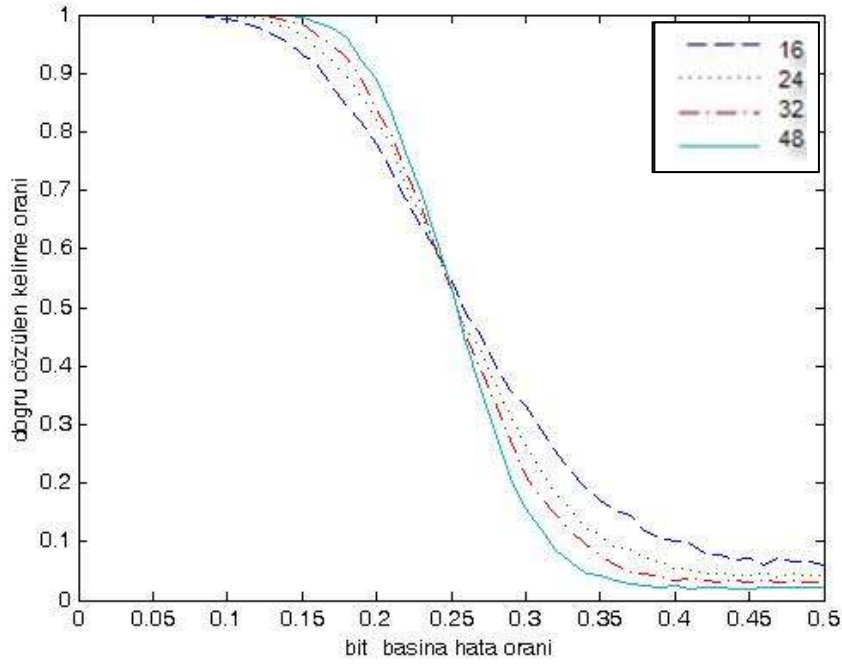
Şekil 5.26. C_3 'in $M=16, 24, 32, 48, 64$ iken çözümüleme zamanlarının karşılaştırılması

Şekil 5.23– Şekil 5.26 incelendiğinde kod kelime sayısı arttıkça her kod için hem kod kelime sayısı olan M hem de gönderilen kelime sayısı arttıkça kodların çözümüleme süreleri oldukça artmaktadır.



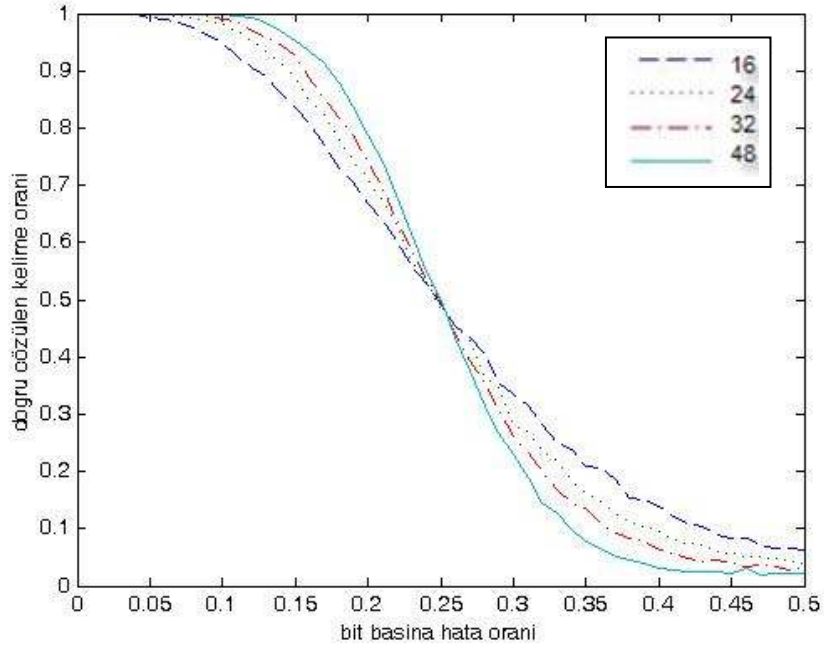
Şekil 5.27. C_2 'in $M=16, 24, 32, 48$ iken doğru çözümleme oranına ilişkin grafik

Şekil 5.27 ele alındığında doğru çözümleme oranının alt limitini 0.95 şeklinde düşünülürse $M=16$ için bu limit bit başına hata oranı 0.05'in üstüne çıktığında doğru düzeltme oranı 0.95'in altına düştüğü görülür. Diğer M değerleri için incelendiğinde M arttıkça bit başına hata oranının sınırının 0.15'e kadar yükseldiği görülür. Ayrıca her kod kelime sayısı için belli bir bit başına hata oranından sonra oldukça hızlı bir biçimde doğru çözümleme oranının düştüğü görülmüştür.



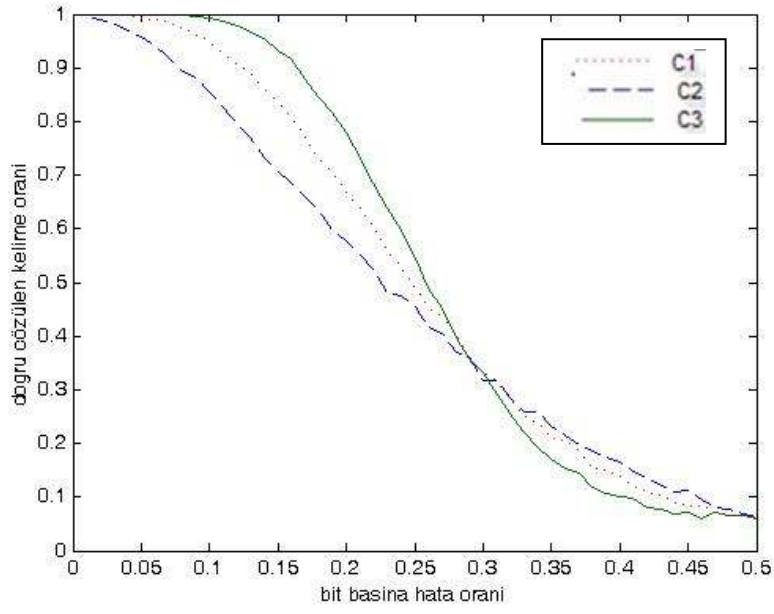
Şekil 5.28. C_3 'in $M=16, 24, 32, 48$ iken doğru çözümleme oranına ilişkin grafik

Şekil 5.28 ele alındığında doğru çözümleme oranının alt limitini 0.95 şeklinde düşünüldüğünde $M=16$ için bu limit bit başına hata oranı 0.13'in üstüne çıktığında doğru düzeltme oranı 0.95'in altına düştüğü görülmüştür. Diğer M değerleri için incelendiğinde, M arttıkça bit başına hata oranının sınırının 0.18'e kadar yükseldiği görülür. Buradaki artış C_2 koduyla aynı değildir. Ayrıca yine bu kodda her kod kelime sayısı için belli bir bit başına hata oranından sonra oldukça hızlı bir biçimde doğru çözümleme oranının düştüğü tespit edilmiştir.

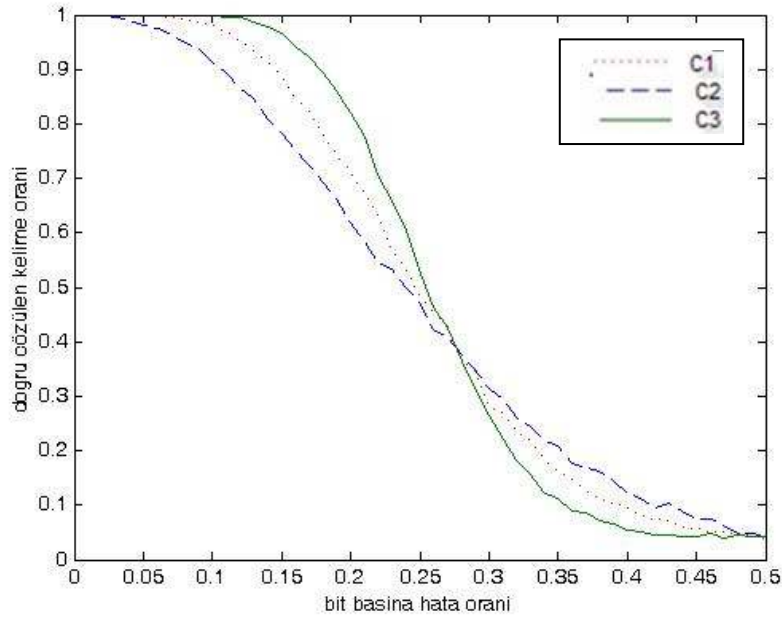


Şekil 5.29. C_1 'in $M=16, 24, 32, 48$ iken doğru çözümleme oranına ilişkin grafik

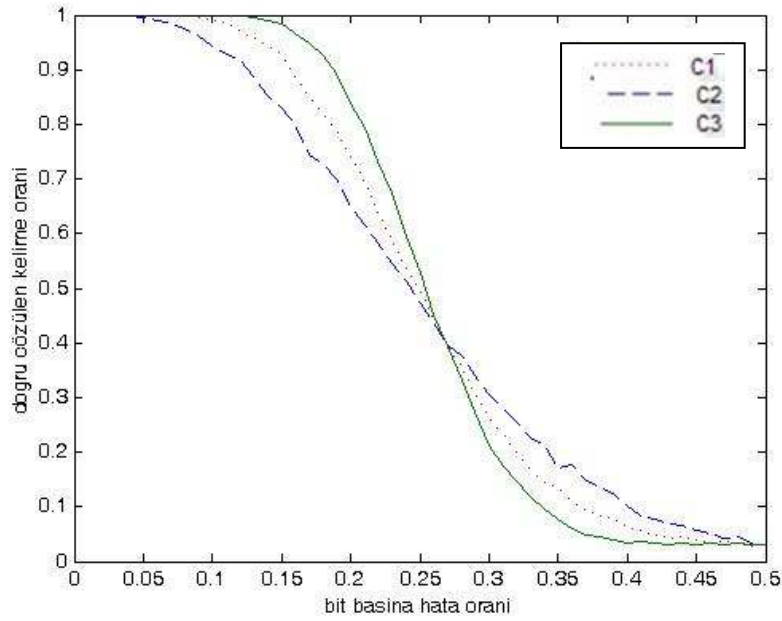
Son olarak Şekil 5.29 ele alındığında doğru çözümleme oranının alt limitini 0.95 şeklinde düşünüldüğünde, $M=16$ için bu limit bit başına hata oranı 0.10'in üstüne çıktığında doğru düzeltme oranı 0.95'in altına düştüğü görülür. Diğer M değerleri için incelendiğinde M arttıkça bit başına hata oranının sınırının 0.16'a kadar yükseldiği saptanmıştır. Bu artış C_2 kodundan az, C_3 kodundan fazladır. Ayrıca yine bu kod için her kod kelime sayısı için belli bir bit başına hata oranından sonra oldukça hızlı bir biçimde doğru çözümleme oranı düşmüştür.



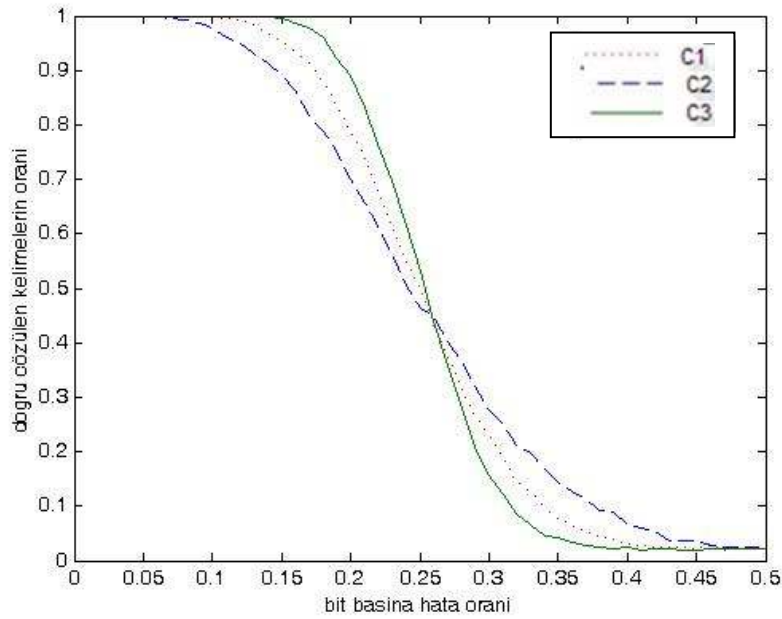
Şekil 5.30. $M=16$ iken C_1 , C_2 ve C_3 'ün doğru çözümleme oranına ilişkin grafik



Şekil 5.31. $M=24$ iken C_1 , C_2 ve C_3 'ün doğru çözümleme oranına ilişkin grafik



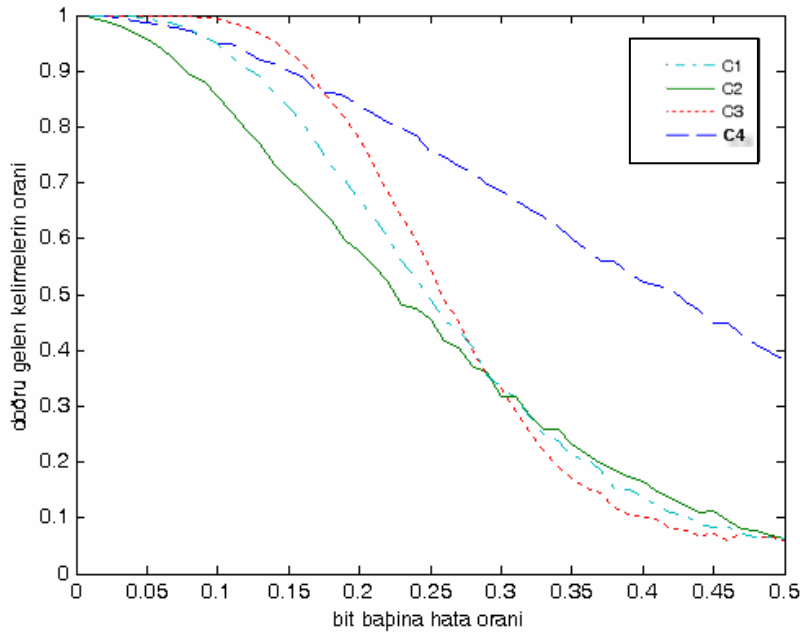
Şekil 5.32. $M=32$ iken C_1 , C_2 ve C_3 'ün doğru çözümleme oranına ilişkin grafik



Şekil 5.33. $M=48$ iken C_1 , C_2 ve C_3 'ün doğru çözümleme oranına ilişkin grafik

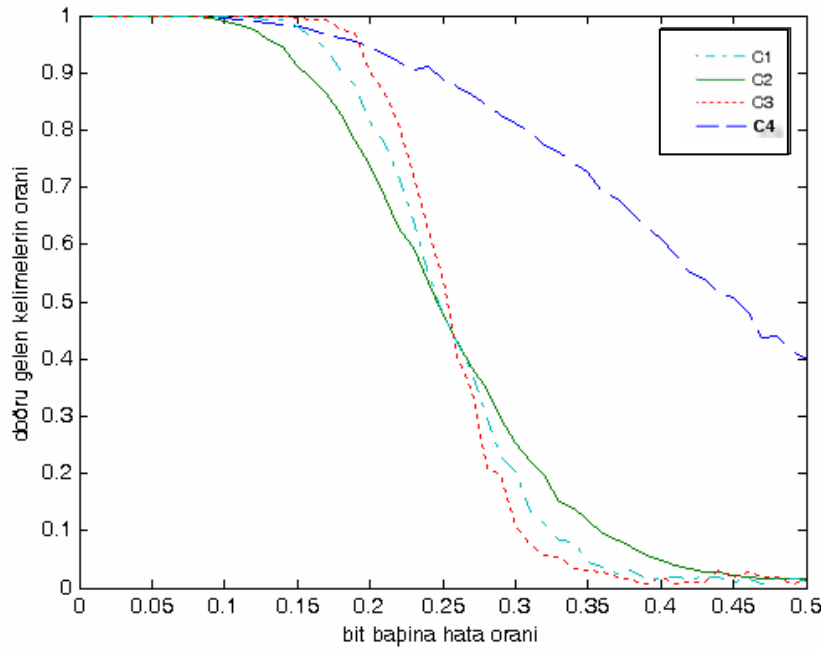
Şekil 5.30 incelendiğinde kod kelime sayısı $M=16$ iken doğru çözümleme oranının alt limitini 0.95 olarak düşünülürse, bit başına hata oranının C_2 kodu için 0.06, C_3 için 0.16 ve C_1 için ise 0.12 olarak elde edilir. Şekil 5.31'i incelendiğinde ise kod kelime sayısı $M=24$ bu limit için bit başına hata oranının C_2 kodu için 0.07 C_3 için

0.17 ve C_1 için ise 0.13 olarak elde edilir. Şekil 5.32'yi dikkate alındığında, ise kod kelime sayısı $M=32$ dir. Bu limit için bit başına hata oranının C_2 kodu için 0.11 C_3 için 0.18 ve C_1 için ise 0.16 olarak elde edilir. Son olarak $M=48$ alındığında bu limit için bit başına hata oranının C_2 kodu için 0.12, C_3 için 0.18 ve C_1 için ise 0.16 olarak elde edilir.



Şekil 5.34. $M=16$ iken C_1 , C_2 , C_3 ve C_4 'ün doğru çözümleme oranına ilişkin grafik

Şekil 5.34'e göre kod kelime sayısı $M=16$ iken doğru çözümleme oranının alt limitini yine 0.95 olarak düşünülürse, bit başına hata oranının C_2 kodu için 0.06 C_3 için 0.16 ve C_1 için ise 0.12 olarak elde edilirken C_4 kodu için bu oran 0.12 olarak elde edilmiştir. C_4 kodu C_1 koduyla aynı sonucu vermiş fakat C_3 den daha düşük bir sınır değeri vermiştir. Ancak doğru çözümleme oranının alt limitini yine 0.90 olarak düşünülürse, elde edilen C_4 ile C_2 'nin sınır değeri aynı olmakta ve bundan sonraki değerlerde C_4 daha iyi sonuç vermektedir. Fakat kabul edilebilecek doğru çözümleme oranının alt limitinin 0.90 olduğunu düşünülürse bu sonuç bir anlam ifade etmez.



Şekil 5.35. $M=64$ iken C_1 , C_2 , C_3 ve C_4 'ün doğru çözümleme oranına ilişkin grafik

Son olarak Şekil 5.35'i incelense kod kelime sayısı $M=64$ iken doğru çözümleme oranının alt limitini yine 0.95 olduğu düşünülürse, bit başına hata oranının C_2 kodu için 0.14 C_1 için 0.19 ve C_1 için ise 0.17 olarak elde edilirken C_4 kodu için bu oran C_3 'le aynı bulunmuştur. Doğru çözümleme oranını 0.95'in altında bir değeri limit olarak alırsa bu durumda C_4 kodu çok daha üstün olarak kabul edilebilir.

Sonuç olarak eğer kullanılacak kodun kod kelimeleri düşük ve bit başına hata oranı düşük ise hem doğru çözme oranı hem de hızlı çözme bakımından C_2 kodu diğerlerine göre daha iyidir. Ancak kodun kod kelimeleri düşük ise fakat bit başına hata oranı yüksek ise doğru çözme oranı bakımından C_3 kodu tercih edilebilir. Özellikle kodun kod kelimeleri yüksek ve aynı zamanda bit başına hata oranı yüksek ise C_4 kodu kullanılır. Fakat C_4 kodunun en büyük dezavantajı sadece M 'in 2^4 , 2^6 veya 2^8 gibi 2'nin çift kuvvetleri için var olmasıdır. Diğer 3 kod ise M , 4'ün katı olduğu herhangi bir değer için bulunabilir.

6. TASARIMLARDAN ELDE EDİLEN KODLARIN DOĞRU ÇÖZÜMLEME ORANLARI

Bu bölümde, Hadamard matrisinden oluşturulan maksimal kodların, afin çözülebilir tasarımdan elde edilen optimal eşit uzaklıklı kodların ve 2-tasarımlardan elde edilen sabit ağırlıklı kodların doğru çözümleme oranları ve doğru çözümleme oranlarının karşılaştırılması verilecektir.

6.1. Kodların Genel Doğru Çözümleme Oranları

Kodların doğru çözümleme oranlarından önce bazı temel tanımları verilecektir.

p : Bit başına hata oranı

E : Hata vektörü

x : Bir kod kelimedede meydana gelen hatalı bit sayısı

E hata vektörünün uzunluğu, n 'dir. Hata vektöründeki 0'dan farklı elemanların sayısı, kod kelimedede meydana gelen hatalı bit sayısını verir. Bu elemanların yerleri ise bu bitlerin yerlerini verir. Örneğin (0 1 1 0 0 0 0) hata vektörü; 2 hata olduğunu ve bu hataların 2. ve 3. bitlerde meydana geldiğini belirtir. Eğer $q=2$ ise, x 'in dağılımı n, p parametrelili Binom dağılımı olduğu açıktır.

$$P(x) = \binom{n}{x} p^x (1-p)^{n-x} \quad x=0,1,\dots,n$$

Bu ifade herhangi bir kod kelimenin x bitinde hata olması olasılığını verir.

Böylece herhangi bir kod kelimenin hatasız gönderilmesi olasılığı;

$$P(0) = \binom{n}{0} p^0 (1-p)^n = (1-p)^n$$

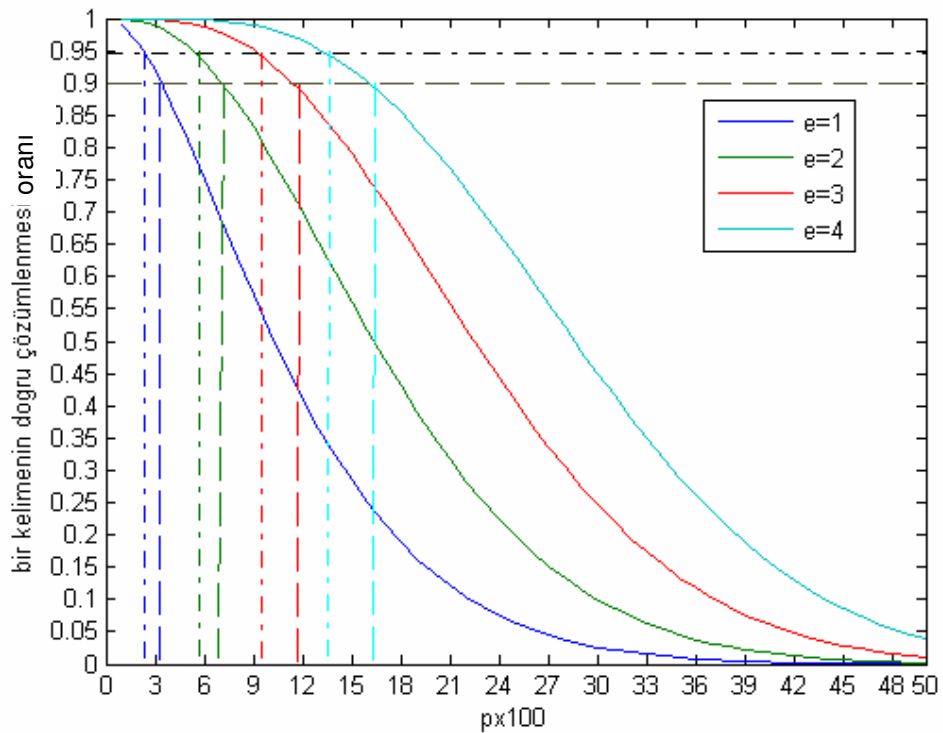
dır. Benzer şekilde herhangi bir kod kelimenin doğru çözümlenebilecek kadar hata yapılması olasılığı;

$$P = \binom{n}{0} p^0 (1-p)^{n-0} + \binom{n}{1} p^1 (1-p)^{n-1} + \dots + \binom{n}{e} p^e (1-p)^{n-e} = \sum_{i=0}^e \binom{n}{i} p^i (1-p)^{n-i}$$

olarak hesaplanır.

6.1. Örnek

$n=16$, $e=1:4$ ve $p= 0.01: 0.01:0.50$. olarak alınsın. İstenen P değerleri (herhangi bir kodun doğru çözümlenebilecek kadar hata yapılması olasılığı) 0.90 ve 0.95 alınsın. Bu durumda Şekil 6.1 elde edilir.



Şekil 6.1. Kod kelime uzunluğu 16 iken herhangi bir kod kelimenin doğru çözümleme olasılığını gösteren grafik

Şekil 6.1 dikkate alındığında herhangi bir kod kelimenin bitlerinin %95'inin doğru gönderilmesi için hata düzeltme sayısı $e=1$ alındığında $p<0.02$ olmalıdır. %90'nın doğru gönderilmesi için $p<0.03$ olmalıdır. $e=2$ olduğunda %95'inin doğru gönderilmesi için hata düzeltme oranı $p<0.55$ olmalıdır. %90'nın doğru gönderilmesi için $p<0.07$ olmalıdır. $e=3$ için %95 ve %90'nın doğru gönderilmesi için sırasıyla $p<0.095$ ve $p<0.115$ olmalıdır. $e=4$ için %95 ve %90'nın doğru gönderilmesi için sırasıyla $p<0.135$ ve $p<0.165$ olmalıdır.

Benzer şekilde herhangi bir kodun doğru çözümlenemeyecek kadar hata yapılması olasılığı;

$$\begin{aligned} & \binom{n}{e+1} p^{e+1} (1-p)^{n-e-1} + \binom{n}{e+2} p^{e+2} (1-p)^{n-e-2} + \dots + \binom{n}{n} p^n (1-p)^{n-n} \\ &= \sum_{i=e+1}^n \binom{n}{i} p^i (1-p)^{n-i} = 1 - P \end{aligned}$$

olarak hesaplanır.

Burada önemli bir faktör kod kelime sayısıdır. Toplam mümkün farklı gelebilecek vektör sayısı ise q^n 'dir. Bu durumda herhangi bir (n, M, d) kodunda doğru düzeltilebilecekler vektörlerin tüm vektörlere oranı

$$\frac{M \sum_{x=0}^e \binom{n}{x} (q-1)^x}{q^n} = \frac{M \sum_{x=0}^e \binom{n}{x} (2-1)^x}{2^n} = \frac{M \sum_{x=0}^e \binom{n}{x}}{2^n}$$

olur.

n, M, d, e parametreleri birbirlerinden bağımsız değildir. $e=\lceil (d-1)/2 \rceil$ şeklindedir. Benzer şekilde M ile d (dolayısıyla e) ters orantılıdır. Ayrıca kodlar, daha önceki bölümlerde verilen Plotkin, Johnson, Hamming gibi sınırlarla kısıtlanmıştır.

Dolayısıyla bu durumları dikkate alarak değerlendirme yapmak daha uygun olur. Bu karşılaştırma için 3. bölümde verilen maksimal ve optimal kodlar gözönüne alınacaktır. .

6.2. Hadamard Matrisinden Oluşturulan Maksimal Kodların Doğru Çözümleme Oranları

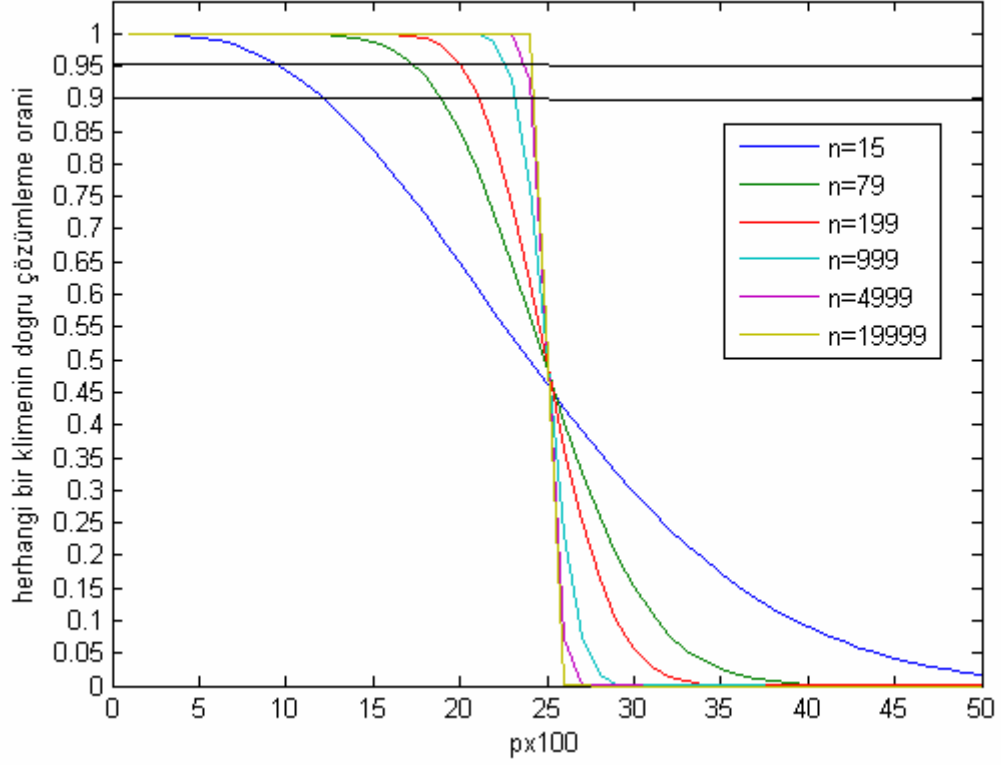
Teorem 4.1’de verilen C_1 kodu dikkate alınsın. C_1 kodu, $(4\lambda-1, 4\lambda, 2\lambda)$ parametrelili maksimal koddur. Dolayısıyla $e=[(2\lambda-1)/2]=\lambda-1$ olacaktır. Buradan herhangi bir kod kelimenin doğru çözümlenebilecek kadar hata yapılması olasılığı;

$$P = \sum_{i=0}^e \binom{n}{i} p^i (1-p)^{n-i} = \sum_{i=0}^{\lambda-1} \binom{4\lambda-1}{i} p^i (1-p)^{4\lambda-1-i}$$

olarak hesaplanır.

6.2. Örnek

$n=15, 79, 199, 999, 4999, 19999$ ve $p=0.01:0.01:0.50$ olsun. İstenen P değerleri(herhangi bir kodun doğru çözümlenebilecek kadar hata yapılması olasılığı) 0.90 ve 0.95 olsun. Bu durumda C_1 koduna ait grafik Şekil 6.2’deki gibi elde edilir.

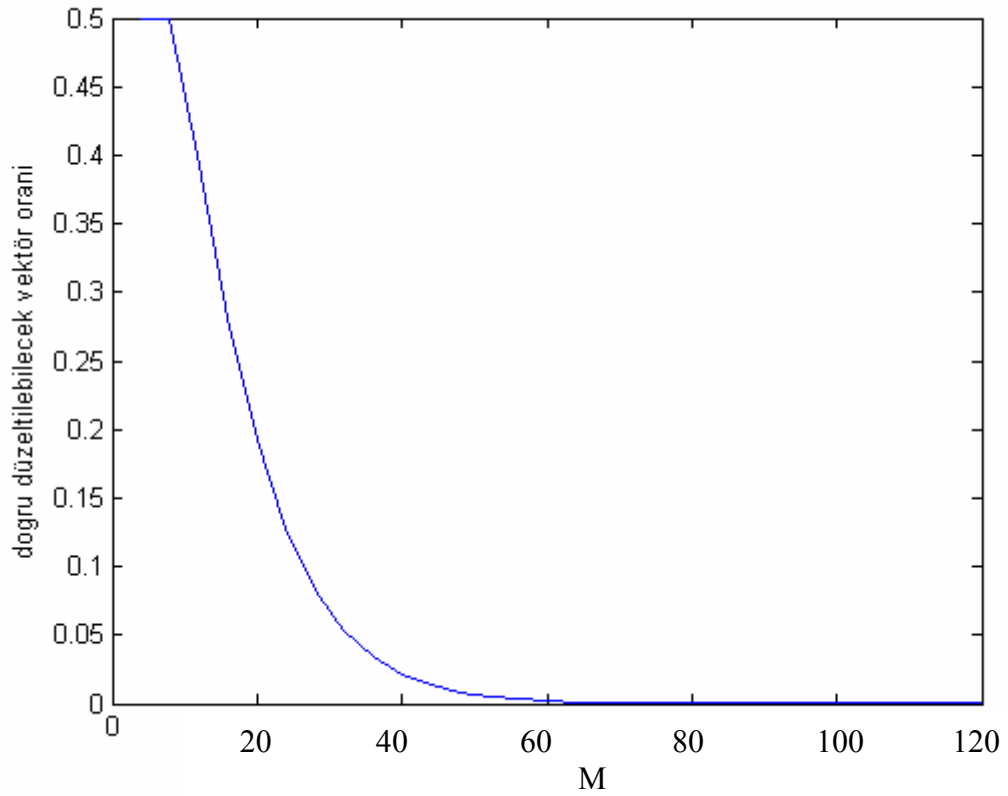


Şekil 6.2. C_1 için Kod kelime uzunluğu 15, 79, 199, 999, 4999, 19999 iken herhangi bir kod kelimenin doğru çözümleme olasılığı gösteren grafik

Şekil 6.2 dikkate alındığında kod kelime uzunluğu olan n arttıkça 0.95 ve 0.90 için kullanılacak bit başına hata oranının üst limiti artmaktadır. Fakat bu artış n 'in artışına bağlı olmaksızın 0.25'i geçmemektedir. Dikkat edilecek olursa, C_1 , p değeri 0.25'e yaklaştıkça doğru çözümleme oranı çok hızlı bir düşüş göstermekte ve 0.50'ye çok yakın değerler almaktadır. Bu durum tüm n değerleri için geçerlidir. Bunun sebebi aşağıdaki gibi açıklanabilir:

Kullanılan dağılım parametreleri $4\lambda-1$ ve p olan bir binom dağılımıdır. Dolayısıyla X değişkenin beklenen değeri $E(X)=(4\lambda-1)p$ 'dir. $p=0.25$ alındığında $E(X)=\lambda-0.25$ olacaktır. Ayrıca n arttıkça binom dağılımın normal dağılıma yaklaşacağından $\lim_{n \rightarrow \infty} P(X \leq E(X)) \rightarrow 0.5$ alınabilir. Burada bulunmak istenen olasılık $P(X \leq \lambda-1)$ dir. bu olasılık λ tamsayı olduğundan $P(X \leq \lambda-1) = P(X \leq \lambda-0.25) = P(X \leq E(X))$ olur. Dolayısıyla $\lim_{n \rightarrow \infty} P(X \leq \lambda-1) \rightarrow 0.5$ yazılabilir.

C_1 kodunda doğru düzeltilebilecek vektörlerin tüm vektörlere oranı şekil 6.3de verilmiştir.



Şekil 6.3. C_1 için Doğru çözümlenecek vektörlerin oranına ilişkin grafik

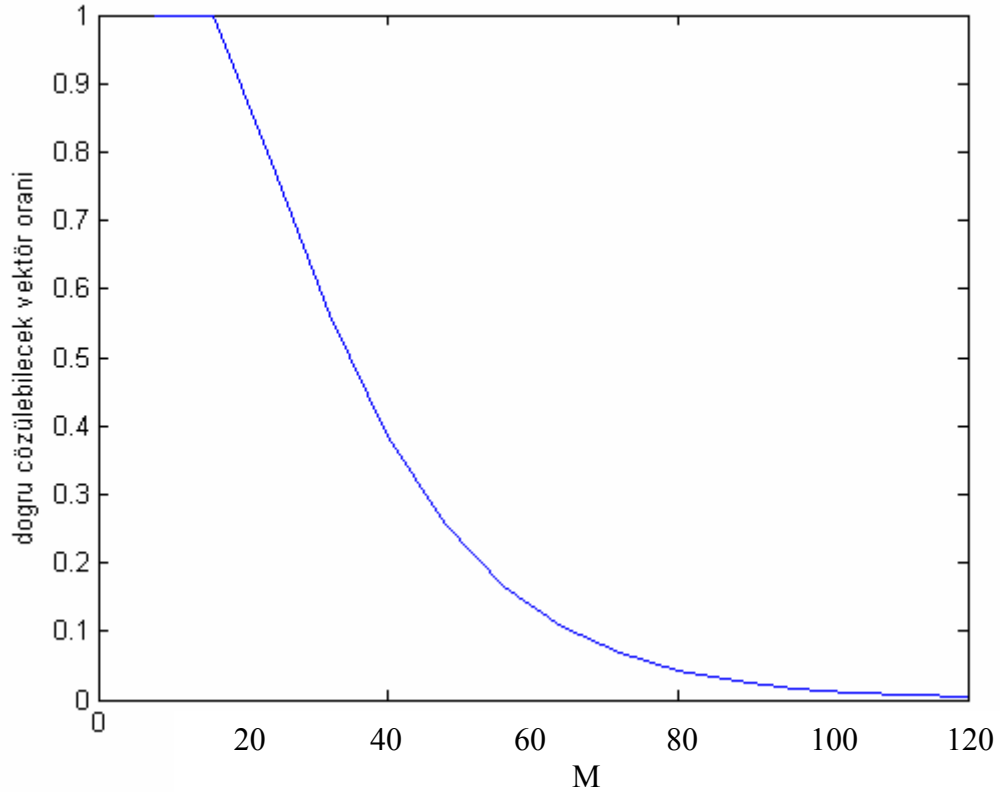
Şekil 6.3 dikkate alındığında C_1 , M arttıkça (dolayısıyla n) doğru çözümlenecek vektörlerin oranı hızlı biçimde düşmekte ve $M=32$ 'den sonra 0.05'in altına inmektedir. Bu çeşit kodlarda eğer p değeri 0.25'e (n büyük olduğunda, aksi halde bu sınır daha aşağıda olacaktır) çok yakın değilse bir sorun yaratmayacaktır. Çünkü bu tarz maksimal kodlarda, kodun düzeltilebileceği maksimum hata sayısı oldukça yüksektir. Eğer, p oldukça büyükse bu durum şu şekilde bir sorun yaratır. Giden kod kelimenin çözümlenemeyecek bölgeye düşme olasılığı oldukça artar. Bu durumda gelen mesajda bazı kod kelimelerin çözümlenememesine neden olur. Bununla beraber kod kelimenin yanlış çözümlenmesi olasılığı oldukça düşüktür.

C_2 kodu için ise şu yorumlar yapılabilir. C_2 kodu $(4\lambda-1, 8\lambda, 2\lambda-1)$ parametrelili maksimal koddur. Dolayısıyla $e=[(2\lambda-1-1)/2]=\lambda-1$ olacaktır. Böylece, C_1 koduna benzer şekilde herhangi bir kod kelimenin doğru çözümlenebilecek kadar hata yapılması olasılığı;

$$P = \sum_{i=0}^e \binom{n}{i} p^i (1-p)^{n-i} = \sum_{i=0}^{\lambda-1} \binom{4\lambda-1}{i} p^i (1-p)^{4\lambda-1-i}$$

olacaktır. Herhangi bir kod kelimenin doğru çözümlenebilecek kadar hata yapılması olasılığını belirlemede C_1 kodu ile benzer yorumlar yapılabilir. Şöyle ki; her ne kadar minimum uzaklık d farklı olsa da düzeltilebilecek bit sayısı $e=[(d-1)/2]=[(2\lambda-1-1)/2]=\lambda-1$ dir. Dolayısıyla C_2 'in düzeltilebilecek bit sayısı C_1 ile aynıdır. Bu kodun C_1 'den esas farklılığı kod kelime sayısıdır. C_1 'in kod kelime sayısı 4λ iken C_2 kodunun kod kelime sayısı $M=8\lambda$ 'dır. Bu durumun avantajı, iki kod benzer bir problemde kullanılacağı zaman C_1 kodunun C_2 kodundan 2 kat büyük mertebeli Hadamard matrisi kullanılacağından C_2 kodunun daha hızlı çözümü yapmasıdır. Çünkü kullanılacak binom dağılımının deneme sayısı (n =kod kelime uzunluğu) parametresi C_2 'inin yaklaşık olarak yarısı kadar olacaktır. Dolayısıyla bu durum C_2 'yi daha hızlı hale getirecektir.

Ayrıca C_2 kodunda doğru düzeltilebilecek vektörlerin tüm vektörlere oranı incelendiğinde Şekil 6.4 elde edilir.



Şekil 6.4. C_2 için Doğru çözümlenecek vektörlerin oranına ilişkin grafik

Şekil 6.4 dikkate alındığında, C_1 'e benzer şekilde C_2 , M arttıkça (dolayısıyla n) doğru çözümlenecek vektörlerin oranı hızlı biçimde düşmektedir. $M=76$ 'dan sonra 0.05'in altına inmektedir. Bu sınır C_1 'e göre oldukça büyüktür. Burada C_1 için söylenenler geçerlidir.

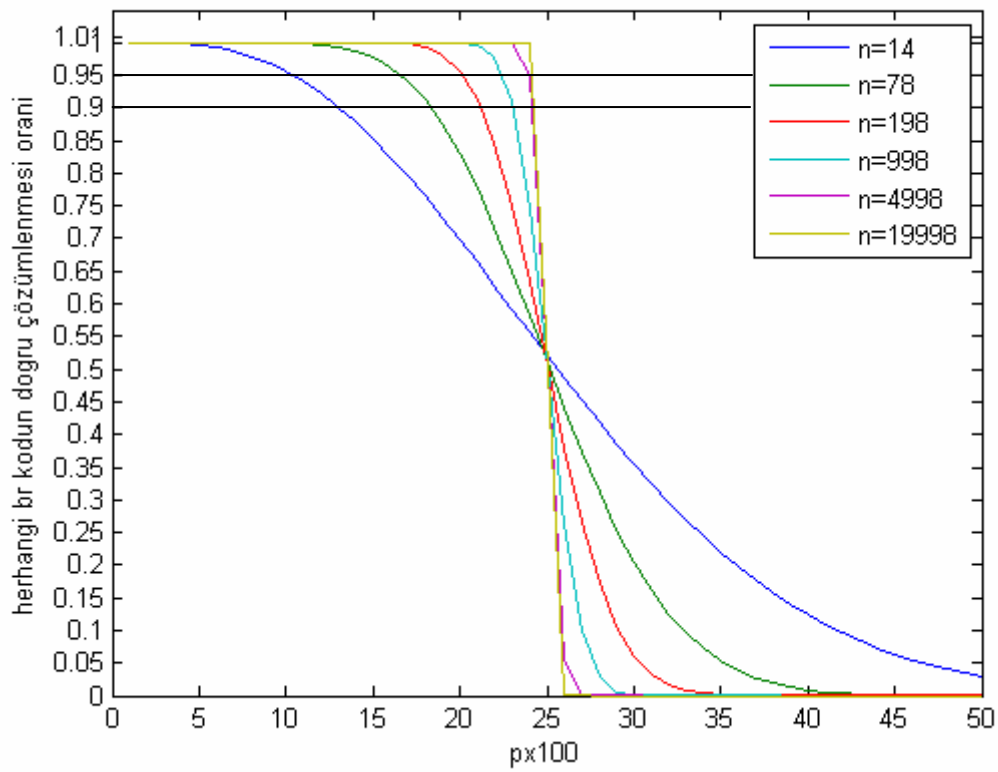
Son olarak C_3 kodu için şu yorumlar yapılabilir. C_3 kodu $(4\lambda-2, 2\lambda, 2\lambda)$ parametrelili maksimal koddur. Dolayısıyla $e=[(2\lambda-1)/2]=\lambda-1$ olacaktır. Buradan C_1 ve C_2 koduna benzer şekilde herhangi bir kod kelimenin doğru çözümlenebilecek kadar hata yapılması olasılığı;

$$P = \sum_{i=0}^e \binom{n}{i} p^i (1-p)^{n-i} = \sum_{i=0}^{\lambda-1} \binom{4\lambda-2}{i} p^i (1-p)^{4\lambda-1-i}$$

olarak hesaplanır.

6.2. Örnek

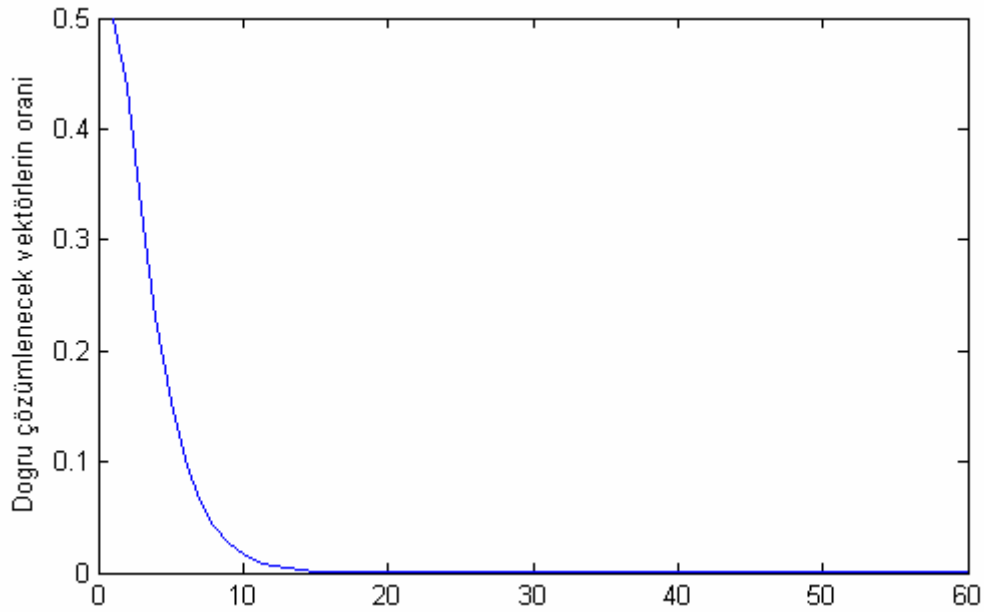
C_1 kodu için farklı bit başına hata oranları için doğru çözümlenebilecek kadar hata yapılması olasılıkları bulunmak istenilsin. $n=14, 78, 198, 998, 4998, 19998$ ve $p=0.01:0.01:0.50$ alınsın. İstenen P değerleri (herhangi bir kodun doğru çözümlenebilecek kadar hata yapılması olasılığı) 0.90 ve 0.95 olsun. Bu durumda C_1 koduna ait grafik şekil 6.5deki gibi elde edilir.



Şekil 6.5. C_3 için kod kelime uzunluğu 14, 78, 198, 998, 4998, 19998 iken herhangi bir kod kelimenin doğru çözümleme olasılığını gösteren grafik

Şekil 6.5 dikkate alındığında, C_1 ve C_2 koduna benzer olarak kod kelime uzunluğu n arttıkça 0.95 ve 0.90 için kullanılacak bit başına hata oranının üst limiti artmaktadır. Yine C_2 de olduğu gibi temel farklılık M değerinde oluşmaktadır.

Ayrıca C_3 kodunda doğru düzeltilebilecek vektörlerin tüm vektörlere oranı incelenecek olursa Şekil 6.6 elde edilir.



Şekil 6.6. C_3 için doğru çözümlenecek vektörlerin oranına ilişkin grafik

Şekil 6.6 dikkate alındığında, C_1 ve C_2 'ye benzer şekilde C_3 , M arttıkça (dolayısıyla n) doğru çözümlenecek vektörlerin oranı hızlı biçimde düşmekte ve $M=8$ 'den sonra 0.05'in altına inmektedir. Bu sınır C_1 ve C_2 'ye göre oldukça küçüktür. C_1 ve C_2 'ye benzer şekilde bu tarz maksimal kodlarda, kodun düzeltebileceği maksimum hata sayısı oldukça yüksektir. Eğer, p büyükse bu durum, giden kod kelimenin çözümlenemeyecek bölgeye düşme olasılığı artacağından sorun yaratır.

Genel olarak bakacak olursak her üç kod için de kod kelime sayısı arttıkça doğru çözümleme olasılıklarının arttığını söyleyebiliriz. Fakat kod kelime sayısını ne kadar artırılırsa artırılсын bit başına hata oranı 0.25'e yaklaştıkça sonuçlar doğru çözümleme oranı düşmekte ve bu oran 0.25'i geçtikçe oran oldukça düşmektedir.

6.3. Afin Çözülebilir Tasarımdan Elde edilen Optimal Eşit Uzaklıklı Kodların Doğru Çözümleme Oranları

İncelenecek optimal eşit uzaklıklı kodlar, Bölüm 6.2'de incelenen Hadamard matrisinden elde edilen maksimal kodlardan farklı olarak $q \geq 2$ olan kodlardır.

Dolayısıyla herhangi bir kod kelimenin hata vektörünün dağılımı $n, 1-p, p/(q-1), \dots, p/(q-1)$ parametrelili çokterimli dağılımı olacaktır.

$$P(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) = \binom{n}{x_1, x_2, \dots, x_q} (1-p)^{x_1} \left(\frac{p}{q-1}\right)^{x_2} \dots \left(\frac{p}{q-1}\right)^{x_q} \quad \begin{array}{l} x_1 + x_2 + \dots + x_q = n \\ x_i = 0, 1, \dots, n \\ i = 1, 2, \dots, q \end{array}$$

X_1 , 0 gelenlerin sayısı, yani hatasız gönderilen bit sayısını tanımlasın. Hata vektöründeki 0'dan farklı elemanların sayısı, kod kelimedede meydana gelen hatalı bit sayısını, yerleri ise bu bitlerin yerlerini verir.

Kullanılan en çok olabilirlik çözümlemesinde dikkate alınan bit'in hatalı olup olmayışına bağlı olarak bu dağılım, 0 gelen bit ve 0 gelmeyen bit olarak düşünülebilir. Dolayısıyla dağılım n ve p parametrelili binom dağılımıdır. Yani,

$$P(x) = \binom{n}{x_2 + x_3 + \dots + x_q} p^{x_2 + x_3 + \dots + x_q} (1-p)^{n - (x_2 + x_3 + \dots + x_q)} = \binom{n}{x} p^x (1-p)^{n-x}$$

dir. Burada kullanılacak ilk kodun parametreleri çözülebilir 2-tasarımın parametreleriyle ilişkili olarak;

$$n=r, M=v, d=b-\lambda, q=b/r$$

şeklindedir.

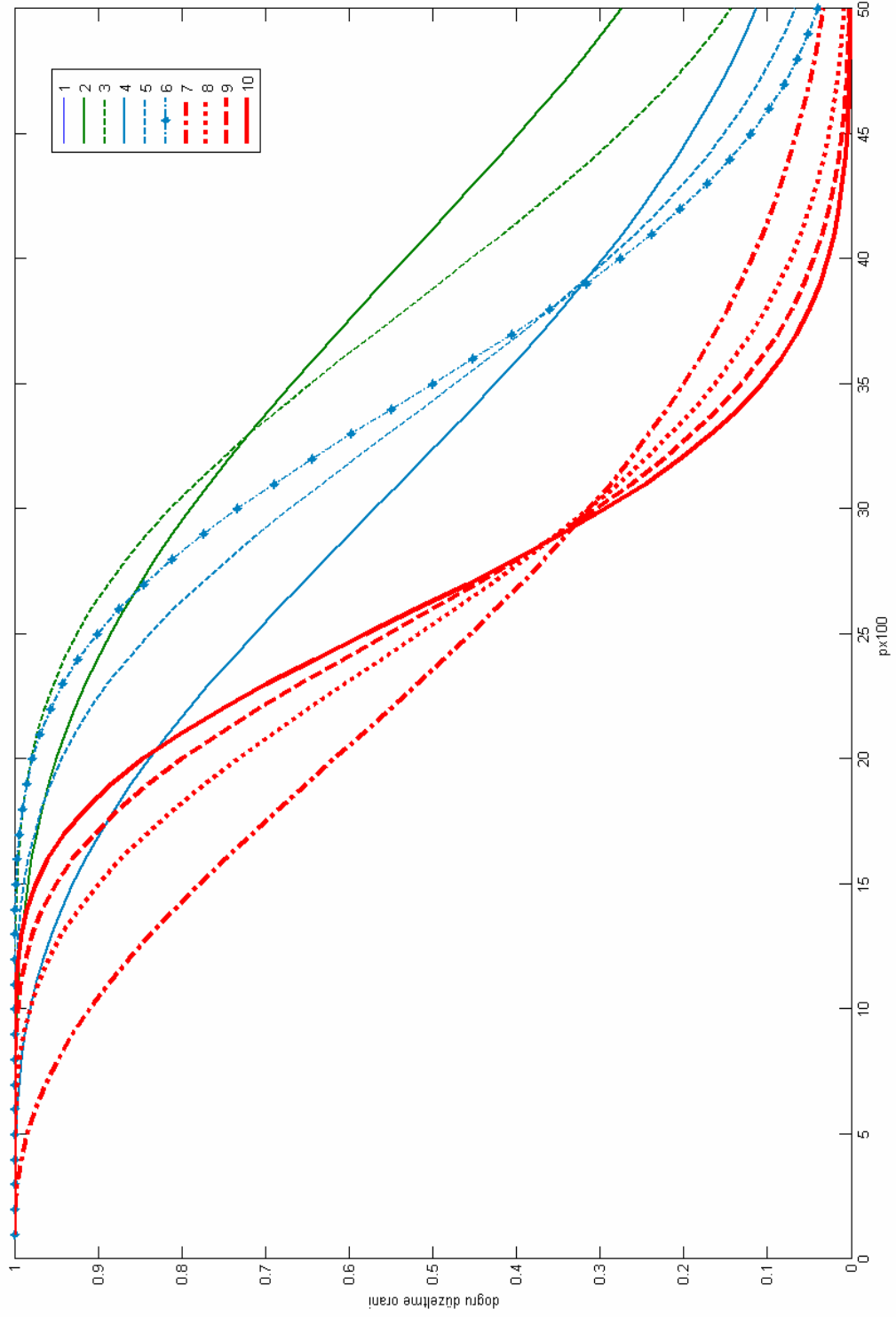
6.3. Örnek

Farklı parametrelili çözülebilir tasarımlardan elde edilen optimal kodları dikkate alınacaktır.

İncelenecek tasarımlar ve bunlara karşılık gelen kodlar Çizelge 6.1'deki gibi olsunlar. Bu tasarımlar, MATLAB 7.1 de yazılan bir program parçasıyla tasarımın varlığı için gerek ve yeter koşulları dikkate alarak elde edilmiştir.

Çizelge 6.1. Çözülebilir 2-tasarımlar ve karşılık gelen kodlar

Tasarım/ Kod no	v	b	k	r	λ		n	M	d	q	t
1	12	66	2	11	1		11	12	10	6	2
2	12	44	3	11	2		11	12	9	4	3
3	12	88	3	22	4		22	12	18	4	3
4	12	33	4	11	3		11	12	8	3	4
5	12	66	4	22	6		22	12	16	3	4
6	12	99	4	33	9		33	12	24	3	4
7	12	22	6	11	5		11	12	6	2	6
8	12	44	6	22	10		22	12	12	2	6
9	12	66	6	33	15		33	12	18	2	6
10	12	88	6	44	20		44	12	24	2	6



Şekil 6.7. C_5 kodu için herhangi bir kod kelimenin doğru çözümleme olasılığını gösteren grafik

Çizelge 6.1'deki 1. ve 2. kodlarda kod kelime uzunluğu n aynıdır. Birinci kodda $q=6$ iken 2. kodda $q=4$ 'dür. Minimum uzaklık d birinci kodda 10 ikinci kodda 9'dur. fakat ikisinin de maksimum hata düzeltebilme sayısı $e=4$ 'dür. Dolayısıyla bu iki kodun eğrileri çakışmıştır. Bu kodlarda %95 ve %90lık doğru çözümleme için p 'in üst sınırları sırasıyla 0.19 ve 0.24'dir. İkinci ve üçüncü kodda $q=4$ 'tür. Bu iki kodun farklılığı n ve d değerleridir. 2.kodda $n=11$ ve $d=9$, 3. kodda ise $n=22$ ve $d=18$ 'dir. 3. koda ait %95 ve %90lık doğru çözümleme için p 'in üst sınırları sırasıyla 0.23 ve 0.27 dir. Bu iki kod karşılaştırıldığında 3. kodun her iki doğru çözümleme oranlarında (%95 ve %90) sırasıyla p 'nin üst sınırını 0.04 ve 0.03 kadar daha yüksek vermektedir. 4. 5. ve 6. kod ($q=3$) incelendiğinde genel olarak $q=4$ olan 2. ve 3. koda göre p 'nin üst sınırı için daha düşük sonuç vermektedir. %95 doğru çözümleme oranında 6. kod, 3. koda yakın sonuç vermekte iken %90 doğru çözümleme oranında daha düşük sonuç vermektedir. 5. kod ise 2. koda %95 doğru çözümleme oranında 2. koda yakın sonuçlar vermektedir. Benzer şekilde %90 doğru çözümleme oranında daha düşük sonuç vermektedir. Fakat genel olarak $q=3$ olan kodlar $q=4$ ve $q=6$ olan kodlar daha düşük olasılıklar vermektedir. Yine, $q=2$ kodlar incelendiğinde benzer sonuçlar verdiği ve $q=6$, $q=4$ ve $q=3$ olan kodlardan daha düşük olasılıklar elde edildiği görülür. Dolayısıyla q arttıkça sabit çözümleme oranında kod daha yüksek bit başına hata oranına müsaade etmektedir. Genel olarak, aynı M düzeyinde q arttıkça doğru çözümleme oranı arttığı söylenebilir. Ayrıca aynı M ve q düzeyinde d arttıkça, n artmasına rağmen kodların daha uygun sonuçlar verdiği yorumu yapılabilir.

Burada $q \geq 2$ olacağından toplam mümkün farklı gelebilecek vektör sayısı q^n 'dir. Bu durumda herhangi bir $(n, M, d)_q$ kodunda doğru düzeltilebilecek vektörlerin tüm vektörlere oranı;

1. kod için;

$$\frac{M \sum_{x=0}^e \binom{n}{x} (q-1)^x}{q^n} = \frac{12 \sum_{x=0}^{(10-1)/2} \binom{11}{x} (6-1)^x}{6^n} = \frac{12 \sum_{x=0}^4 \binom{11}{x} 5^x}{6^n} = 0.0076$$

2. kod için;

$$\frac{M \sum_{x=0}^e \binom{n}{x} (q-1)^x}{q^n} = \frac{12 \sum_{x=0}^{(9-1)/2} \binom{11}{x} (4-1)^x}{4^n} = \frac{12 \sum_{x=0}^4 \binom{11}{x} 3^x}{4^n} = 0.0907$$

3. kod için;

$$\frac{M \sum_{x=0}^e \binom{n}{x} (q-1)^x}{q^n} = \frac{12 \sum_{x=0}^{(18-1)/2} \binom{22}{x} (4-1)^x}{4^n} = \frac{12 \sum_{x=0}^8 \binom{22}{x} 3^x}{4^n} = 0.0017$$

⋮

10. kod için;

$$\frac{M \sum_{x=0}^e \binom{n}{x} (q-1)^x}{q^n} = \frac{12 \sum_{x=0}^{(24-1)/2} \binom{44}{x} (2-1)^x}{2^n} = \frac{12 \sum_{x=0}^{11} \binom{44}{x}}{2^n} = 0.0076$$

elde edilir. Sonuçlar Çizelge 6.2’de görülmektedir.

Çizelge 6.2. Eşit uzaklıklı kodların doğru düzeltme olasılığı

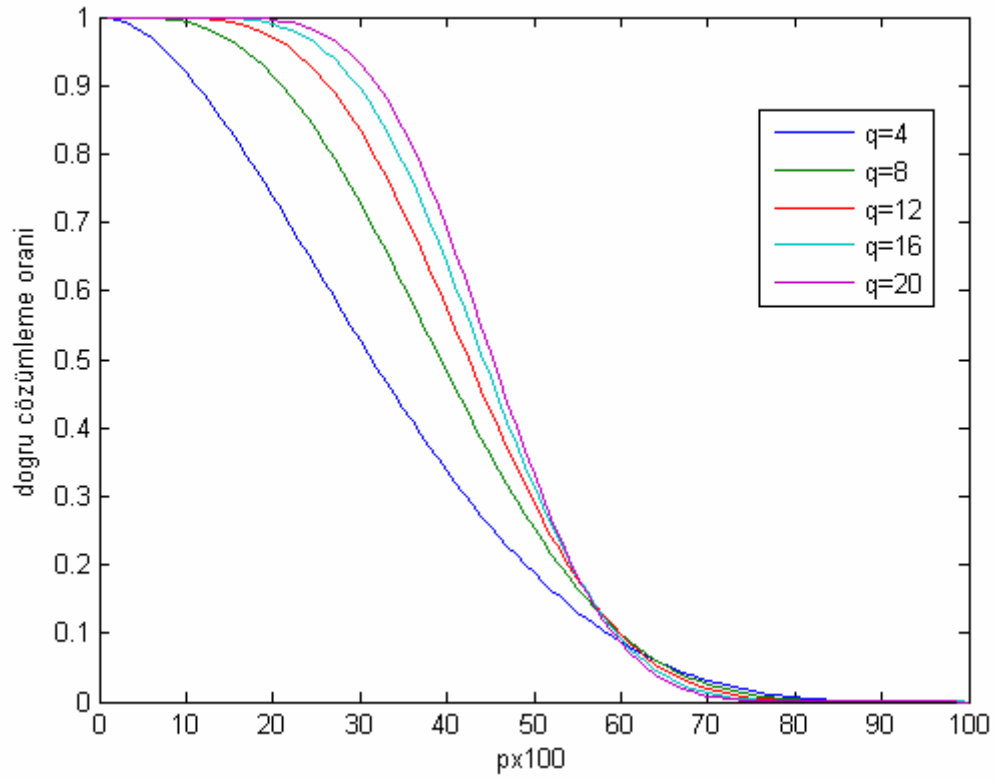
Kod no	n	M	d	q	t	Doğru düzeltilebilecek vektörlerin tüm vektörlere oranı
1	11	12	10	6	2	0.0076
2	11	12	9	4	3	0.0907
3	22	12	18	4	3	0.0017
4	11	12	8	3	4	0.0143
5	22	12	16	3	4	0.0003
6	33	12	24	3	4	0.00001
7	11	12	6	2	6	0.3926
8	22	12	12	2	6	0.1014
9	33	12	18	2	6	0.0276
10	44	12	24	2	6	0.0076

Çizelge 6.2'ye dikkat edildiğinde q sabit tutulup n ve d artırıldığında kodun doğru düzeltilebilecek vektörlerin tüm vektörlere oranı gittikçe düşmektedir. Benzer şekilde q azaldıkça kodun doğru düzeltilebilecek vektörlerin tüm vektörlere oranı gittikçe artmaktadır.

Kullanılacak ikinci kod $q=t$ iken $v = q^2$, $k = q$, $\lambda = 1$ parametrelili afin çözülebilir tasarıma karşılık gelen $n = q + 1$, $M = q^2$, $d = q$ parametereli koddur.

6.3. Örnek

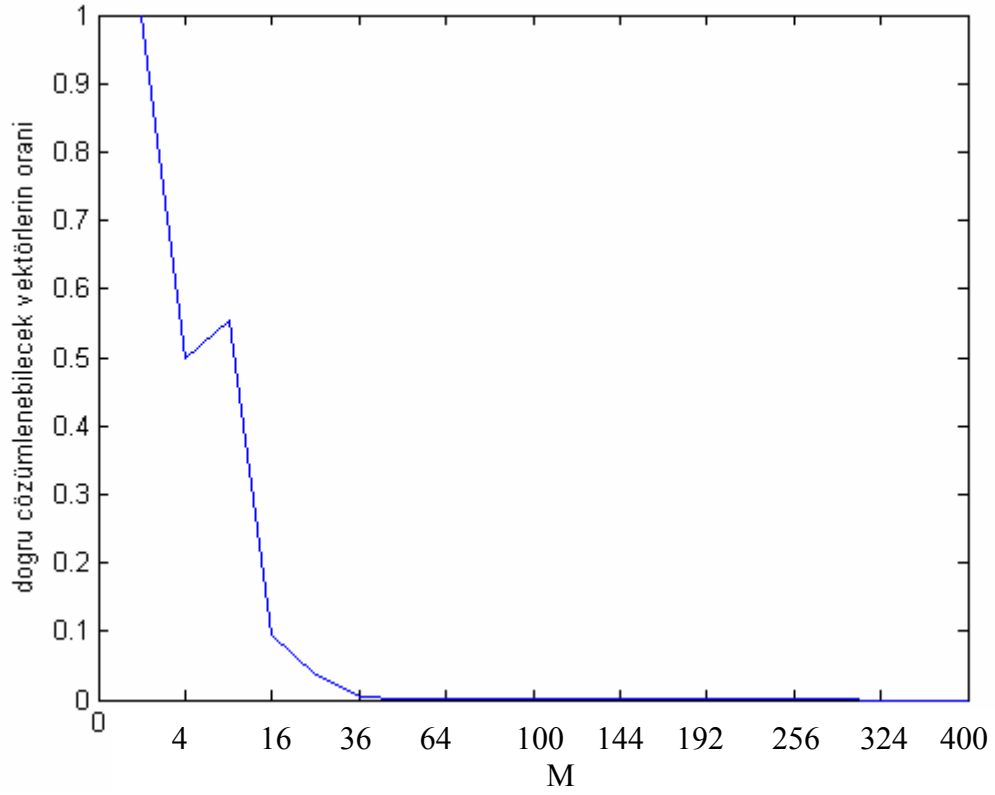
$q=4:4:20$ parametrelili ve afin çözülebilir tasarımlardan elde edilen optimal kodlar dikkate alınsın.



Şekil 6.8. C_4 için herhangi bir kod kelimenin doğru çözümleme olasılığını veren grafik

Şekil 6.8'de görüldüğü gibi, q arttıkça (dolayısıyla n , M ve d) doğru çözümleme olasılığı artmaktadır. Ancak bu artış miktarı giderek azalmaktadır.

Ayrıca C_4 kodunda doğru düzeltilebilecek vektörlerin tüm vektörlere oranı incelendiğinde Şekil 6.9 elde edilir.



Şekil 6.9. C_4 için doğru çözümlenecek vektörlerin oranına ilişkin grafik

Şekil 6.9'da M arttıkça (dolayısıyla q) doğru çözümlenecek vektörlerin oranının çok hızlı biçimde düştüğü görülmektedir.

6.4. 2-tasarımlardan Elde Edilen Sabit Ağırlıklı Kodların Doğru Çözümleme Oranları

Bu bölümde, sabit ağırlıklı kodlardaki kod kelimelerin üst sınırının bulunuşu 2-tasarım yardımıyla elde edilen Maksimal Sabit ağırlıklı kodlar incelenecektir. Bunun için ilk önce MATLAB 7.1 de yazılan bir program parçasıyla 2- tasarımlar ve bunlara ilişkin kodlar oluşturulacaktır.

Ele alınacak kod 2-tasarımlarla ilişkili ve $A(n, 2\delta, w) \leq \frac{\delta n}{w^2 - wn + \delta n}$ koşulunu sağlayan ve parametreleri $n=b$, $w=r$ ve $\delta=w-\lambda$ dolayısıyla $d=2(w-\lambda)$ olan maksimal koddur.

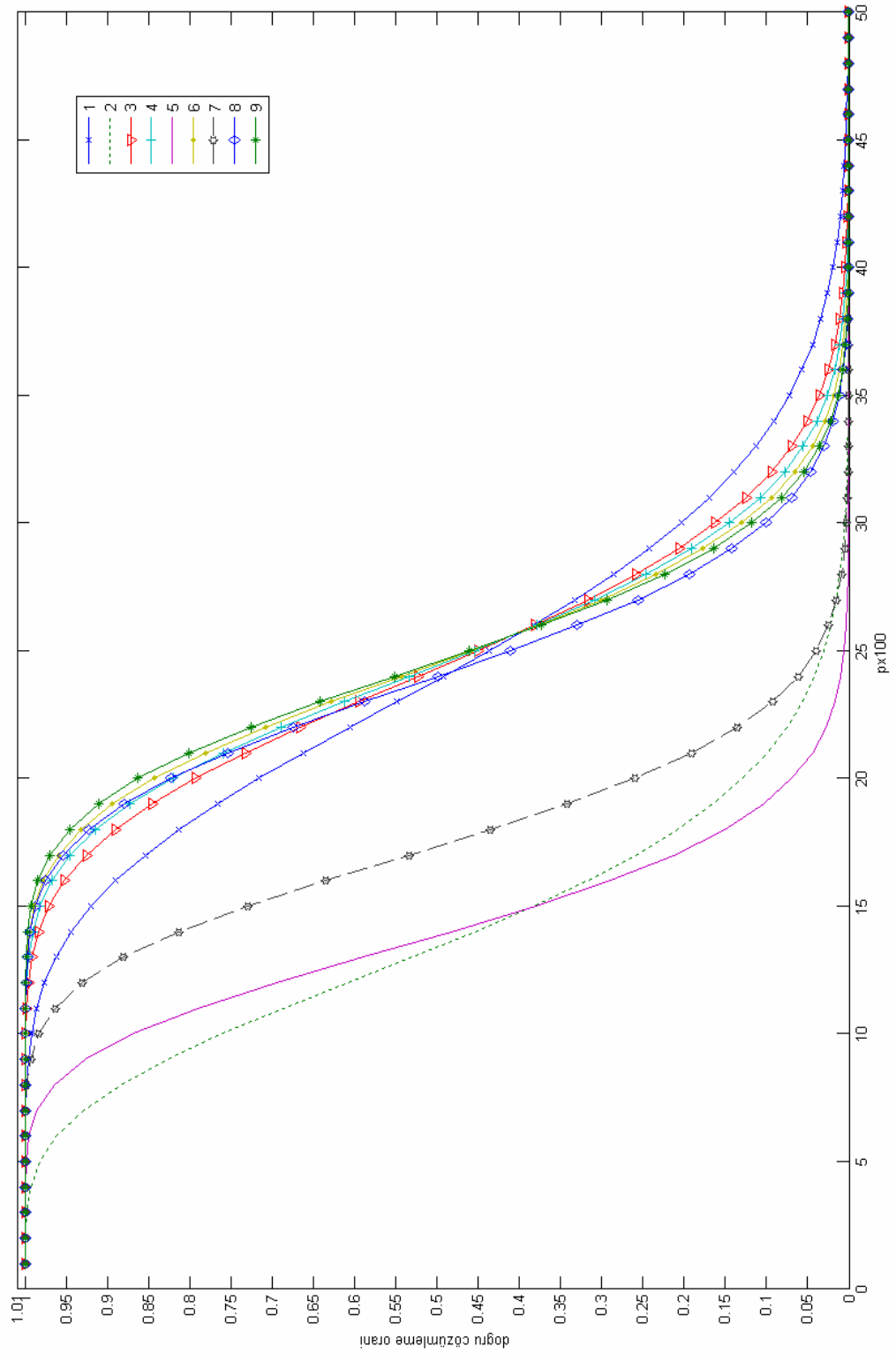
6.4. Örnek

Farklı parametrelili (kod kelimeler sayısı M sabit tutularak) ve 2-tasarımlardan elde edilen maksimal sabit ağırlıklı kodlar dikkate alınacaktır.

Tasarımlar ve bunlara karşılık gelen kodlar Çizelge 6.3'deki gibi olsunlar. Bu parametreler, MATLAB 7.1 de yazılan bir program parçasıyla tasarımın varlığı için gerek ve yeter koşullar sağlatılarak elde edilmiştir.

Çizelge 6.3. 2-tasarımlardan elde edilen sabit ağırlıklı kodlar

Tasarım/ Kod no	v	b	k	r	λ		n	M	d	w	δ
1	36	36	15	15	6		36	36	18	15	9
2	36	42	6	7	1		42	36	12	7	6
3	36	60	15	25	10		60	36	30	25	15
4	36	72	15	30	12		72	36	36	30	18
5	36	84	6	14	2		84	36	24	14	12
6	36	84	15	35	14		84	36	42	35	21
7	36	90	8	20	4		90	36	32	20	16
8	36	90	14	35	13		90	36	44	35	22
9	36	96	15	40	16		96	36	48	40	24



Şekil 6.10. C_6 için herhangi bir kod kelimenin doğru çözümleme olasılığını gösteren grafik

Şekil 6.10 dikkat edilecek olursa incelenen 9 kod 2 gruba ayrılmıştır. Birinci grupta 2, 5, 7 no'lu kodlar diğer grupta ise 1, 3, 4, 6, 8, 9 no'lu kodlar vardır. İkinci gruptaki kodların %95 doğru çözümleme oranında p 'nin üst sınırı yaklaşık olarak 0.16-0.19 arasında değişmekte iken birinci grupta bu oran 0.6-0.12 arasındadır. Bu oldukça farklı bir sonuçtur. Benzer şekilde, ikinci gruptaki kodların %90 doğru çözümleme oranında p 'nin üst sınırı yaklaşık olarak 0.14-0.18 arasında değişmekte iken birinci grupta bu oran 0.8-0.13 arasındadır. Bu gruplara dikkat edildiğinde, ikinci gruptaki kodlarda d/n daima 0.5 şeklinde iken birinci grupta bu oran daha düşüktür. Dolayısıyla ikinci gruptaki kodların doğru çözümleme oranları daha yüksektir.

Çizelge 6.4. Sabit ağırlıklı kodların doğru düzeltme olasılığı

Kod no	n	M	d	w	δ	Doğru düzeltilebilecek vektörlerin tüm vektörlere oranı
1	36	36	18	15	9	0.0215000000
2	42	36	12	7	6	0.0000100000
3	60	36	30	25	15	0.0007000000
4	72	36	36	30	18	0.0001400000
5	84	36	24	14	12	0.00000000001
6	84	36	42	35	21	0.0000200000
7	90	36	32	20	16	0.0000000080
8	90	36	44	35	22	0.0000069000
9	96	36	48	40	24	0.0000055984

Çizelge 6.4'e göre, n arttıkça genel olarak doğru düzeltilebilecek vektörlerin tüm vektörlere oranı düşmektedir. Aynı n düzeyinde d parametresi büyük olan kodun doğru düzeltilebilecek vektörlerin tüm vektörlere oranı daha büyük olmaktadır.

7. HADAMARD MATRİSLERİ İÇİN YENİDEN DÜZENLENMİŞ EN ÇOK OLABİLİRLİK ALGORİTMASI

Bu bölümde, Bölüm 3’de ele alınan çözümleme algoritması (en çok olabilirlik çözümleme algoritması) yeniden düzenlenerek Hadamard tasarımlardan elde edilen maksimal kodlara uygulanacak ve çözümleme zamanları bakımından karşılaştırılacaktır. Bunun için, daha önceden Bölüm 4.4’de anlatılan ve Cameron(1996)’un verdiği $PG(2,2)$ ’yi kullanarak oluşturduğu çözümleme algoritmasından yola çıkılmıştır.

Ayrıca 8,12 ve 16 mertebeli Hadamard matrislerinden elde edilen tasarımlardan bulunan maksimal kodlar için çözümleme algoritması oluşturulacaktır. Bunun için ilk önce Çizelge 4.2’deki algoritma yeniden düzenlensin. Çizelge 4.2’ye dikkat edildiğine ağırlığı 0 ve 1 olan vektörler, kod kelimeler ile arasındaki uzaklıklar dikkate alınmaksızın, doğrudan 0 vektörü olarak çözümlenir. Eğer ağırlık 2 ile 4 arası ve ağırlığı 3 olan kod kelimeler ile arasındaki uzaklıkların biri 1 veya daha küçük ise o kod kelimeye çözümlenir. Eğer ağırlık 3 ile 5 arası ve ağırlığı 4 olan kod kelimeler ile arasındaki uzaklıkların biri 1 veya daha küçük ise o kod kelimeye çözümlenir. Eğer ağırlık 6 veya 7 ise, kod kelimeler ile arasındaki uzaklıklar dikkate alınmaksızın, doğrudan 1 vektörü olarak çözümlenir. Bu algoritma Çizelge 7.1’deki gibi özetlenebilir.

Çizelge 7.1. Yeniden düzenlenmiş en çok olabilirlik algoritması

1’lerin sayısı	Çözümleme kuralı
0	0 olarak çözümlenir.
1	
2	3 ağırlıklı kod kelimeler ile arasındaki uzaklıklara bakılır ve bu uzaklıkların biri 1 veya daha küçük ise o kod kelimeye çözümlenir.
3	
4	
3	4 ağırlıklı kod kelimeler ile arasındaki uzaklıklara bakılır ve bu uzaklıkların biri 1 veya daha küçük ise o kod kelimeye çözümlenir.
4	
5	
6	1 olarak çözümlenir.
7	

Burada dikkat edileceği üzere ağırlıkları 0,1,6,7 olan vektörlerin kod kelimeler arasındaki uzaklıklara bakılmaksızın çözümleme yapıldığından ve diğerleri için 16 yerine sadece 7 tane kod kelime ile uzaklıklarına bakmak yeterli olduğundan, çözümleme zamanı düşecektir. Dolayısıyla gönderilen kod kelimeler daha hızlı çözümlenecektir.

Bu algoritma C_1 kodu için aşağıdaki gibi düzenlenir.

7.1. C_1 Kodu İçin Yeniden Düzenlenmiş En Çok Olabilirlik Algoritması

İlk önce H_8 den elde edilen C_1 kodu için bu algoritma düzenlensin. Bu kodun parametreler (7,8,4) şeklindedir, dolayısıyla $e=1$ olacaktır. Çizelge 4.1'deki algoritma dikkate alındığında projektif geometriden elde edilen bu kodda 1 tane 0 ağırlıklı, 1 tane 7 ağırlıklı, 7 tane 3 ve 7 tane 4 ağırlıklı kod kelime olduğu görülmektedir. Fakat H_8 den elde edilecek C_1 kodunun yapısı gereği 1 tane 0 ağırlıklı ve 7 tane 4 ağırlıklı kod kelime olacaktır. Dolayısıyla çözümleme algoritması şu şekilde yazılabilir. $e=1$ ve kod kelimelerin 1 tanesi 0 vektörü olduğundan, ağırlığı 1 yada 0 olan vektörleri 0 kod kelimesi olarak çözümlenir. Eğer ağırlıklar 3 yada 5 ise ve herhangi bir kod kelime ile uzaklık 1 ise o kod kelime olarak çözümlenir. Eğer ağırlık 4 ve herhangi bir kod kelime ile arasındaki uzaklık 0 ise o kod kelime olarak çözümlenir. Eğer ağırlık 2, 6 ya da 7 ise hiçbir kod kelimeyle uzaklığı 1'den küçük olamayacağından çözülemezdir. Dolayısıyla bu algoritmayı Çizelge 7.2'de verildiği gibi elde edilir.

Çizelge 7.2. H_8 den elde edilen C_1 için yeniden düzenlenmiş en çok olabilirlik algoritması

1'lerin sayısı	Çözümleme kuralı
0,1	0 olarak çözümlenir.
2,6,7	Çözülemez.
3,5	Herhangi bir kod kelime ile uzaklık 1 ise o kod kelime olarak çözümlenir.
4	Herhangi bir kod kelime ile uzaklık 0 ise o kod kelime olarak çözümlenir.

H_{12} den elde edilen C_1 kodu için Çizelge 4.2'deki algoritma düzenlensin. Bu kodun parametreleri (11,12,6) şeklindedir, dolayısıyla $e=2$ olacaktır. H_{12} 'den elde edilen C_1 kodunun yapısı gereği 1 tane 0 ağırlıklı ve 11 tane 6 ağırlıklı kod kelime olacaktır. $e=2$ ve kod kelimelerin 1 tanesi 0 vektörü olduğundan, ağırlığı 0, 1 veya 2 olan vektörleri 0 kod kelimesi olarak çözümlenir. Eğer ağırlıklar 5 yada 7 ise ve herhangi bir kod kelime ile uzaklık 1 ise o kod kelime olarak çözümlenir. Eğer ağırlıklar 4 yada 8 ise ve herhangi bir kod kelime ile uzaklık 2 ise o kod kelime olarak çözümlenir. Eğer ağırlık 6 ve herhangi bir kod kelime ile arasındaki uzaklık 0 ise o kod kelime olarak çözümlenir. Eğer ağırlık 3, 9, 10 ya da 11 ise hiçbir kod kelimeyle uzaklığı 2'den küçük olamayacağından çözülemezdir. Bu algoritma Çizelge 7.3'deki gibi düzenlenir.

Çizelge 7.3. H_{12} den elde edilen C_1 için yeniden düzenlenmiş en çok olabilirlik algoritması

1'lerin sayısı	Çözümleme kuralı
0,1,2	0 olarak çözümlenir.
3,9,10,11	Çözülemez.
4,8	Herhangi bir kod kelime ile uzaklık 2 ise o kod kelime olarak çözümlenir.
5,7	Herhangi bir kod kelime ile uzaklık 1 ise o kod kelime olarak çözümlenir.
6	Herhangi bir kod kelime ile uzaklık 0 ise o kod kelime olarak çözümlenir.

H_8 ve H_{12} den elde edilen C_1 koduna paralel olarak H_{16} dan elde edilen C_1 kodu için çözümleme algoritması Çizelge 7.4'deki gibi olacaktır.

Çizelge 7.4. H_{16} den elde edilen C_1 için yeniden düzenlenmiş en çok olabilirlik algoritması

1'lerin sayısı	Çözümleme kuralı
0,1,2,3	0 olarak çözümlenir.
4,12,13,14,15	Çözülemez.
5,11	Herhangi bir kod kelime ile uzaklık 3 ise o kod kelime olarak çözümlenir.
6,10	Herhangi bir kod kelime ile uzaklık 2 ise o kod kelime olarak çözümlenir.
7,9	Herhangi bir kod kelime ile uzaklık 1 ise o kod kelime olarak çözümlenir.
8	Herhangi bir kod kelime ile uzaklık 0 ise o kod kelime olarak çözümlenir.

C_1 kodu için Yeniden düzenlemiş en çok olabilirlik algoritması ile klasik en çok olabilirlik algoritmasını işlemci zamanı bakımından karşılaştırmak için 10000 tane rasgele üretilmiş kod kelime belirli bit başına hata oranları (0.01, 0.10, 0.20) uygulanıp çözümlene hızları karşılaştırılmıştır. Simülasyon çalışması için $2 \times 1.83\text{GHz}$ çift çekirdek işlemcili ve 1 GB RAM'a sahip PC kullanılmıştır.

Çizelge 7.5. C_1 için ML ve MML için çözümlene zamanları

	0.01		0.10		0.20	
	ML	MML	ML	MML	ML	MML
H_8	3.6563	3.4219	3.3906	3.0938	2.4688	2.1563
H_{12}	7.6563	6.5000	6.5156	5.7188	4.7969	4.2656
H_{16}	9.8125	9.0313	9.2669	8.6406	7.9469	6.4371

Çizelge 7.5'e dikkat edilecek olursa kod kelime sayısı arttıkça çözümlene zamanları artmaktadır. Ayrıca bit başına hata oranı arttıkça çözümlene zamanı düşmektedir. Bunun sebebi gelen vektörlerde çözümlenemeyecek kadar çok hatalı olan vektörlerin daha çok olmasıdır. Burada dikkat edilecek bir nokta, örneğin H_8 için, bit başına hata oranı 0.01 iken ML ile MML arasında %7 gibi bir fark var iken bit başına hata oranı arttıkça bu fark %14'e kadar çıkmaktadır. Bunun sebebi çözümlenemeyecek kadar çok hatalı olan vektörleri daha kolay tespit etmesidir. Dikkat edilecek diğer önemli nokta kod kelime sayısı arttıkça ML ile MML arasındaki fark %23'e kadar çıkmaktadır. Bunun sebebi çözümlenemeyecek kadar çok hatalı olan vektörlerin sayısının artmasıdır.

7.2. C_2 Kodu İçin Yeniden Düzenlenmiş En Çok Olabilirlik Algoritması

C_2 kodunun yapısı projektif geometriden elde edilen kodlara oldukça benzemektedir. Özellikle 2^n ($n > 2$ o.ü.) gibi 2 'nin kuvveti olan hadamard matrislerinden elde edilen C_2 kodları ile $PG(n-1,2)$ parametrelili Projektif geometriden elde edilen kodlar aynı parametreye ve yapıya sahiptir. Örneğin 8 mertebeli bir Hadamard matrisinden elde edilen kod ile $PG(2,2)$ parametrelili Projektif geometriden elde kodların parametreleri

$(7,16,3)_2$ şeklindedir. Yine benzer şekilde 16 mertebeli bir Hadamard matrisinden elde edilen kod ile $PG(3,2)$ parametrelili Projektif geometriden elde kodların parametreleri $(15,32,7)_2$ şeklindedir. Dolayısıyla H_8 den elde edilen maksimal kod için Çizelge 7.1'i kullanılabilir. H_{12} 'den elde edilen maksimal kodun parametreleri $(11,24,5)_2$ şeklindedir. Bu kodda 1 tane 0 ağırlıklı, 11 tane 5 ağırlıklı, 11 tane 6 ağırlıklı ve 1 tane 11 ağırlıklı kod kelime olacaktır. H_{12} den elde edilen kodun H_8 den elde edilen koddan en büyük farkı; H_8 den elde edilen kod hem maksimal hem de tam kod iken, H_{12} den elde edilen kod maksimal olması, dolayısıyla her gelen vektörü çözümleyememesidir. Çizelge 7.1'e dikkat edilecek olursa herhangi bir durumda kod kelimenin çözümlenememe gibi bir durumu yoktur. Bu koşullar altında çözümleme algoritması H_{12} ye göre düzenlensin. $d=5$ olduğundan $e=2$ olacaktır. Dolayısıyla 0,1,2 ağırlıklı gelen vektörler 0 olarak ve 9,10,11 ağırlıklı gelen vektörler 1 olarak çözümlenir. 3, 4, 5, 6, 7 ağırlıklı gelen vektörler için herhangi bir 5 ağırlıklı kod kelime ile uzaklığı 2 veya daha küçükse o kod kelime olarak çözümlenir. 4, 5, 6, 7, 8 ağırlıklı gelen vektörler için herhangi bir 6 ağırlıklı kod kelime ile uzaklığı 2 veya daha küçükse o kod kelime olarak çözümlenir. Bu algoritmayı Çizelge 7.6'daki gibi özetlenir.

Çizelge 7.6. H_{12} den elde edilen C_2 için yeniden düzenlenmiş en çok olabilirlik algoritması

1'lerin sayısı	Çözümleme kuralı
0, 1, 2	<u>0</u> olarak çözümlenir.
3,4,5,6,7	5 ağırlıklı kod kelimeler ile arasındaki uzaklıklara bakılır ve bu uzaklıkların biri 2 veya daha küçük ise o kod kelimeye çözümlenir.
4,5,6,7,8	6 ağırlıklı kod kelimeler ile arasındaki uzaklıklara bakılır ve bu uzaklıkların biri 3 veya daha küçük ise o kod kelimeye çözümlenir.
9,10,11	<u>1</u> olarak çözümlenir.

Son olarak H_{16} dan elde edilen C_2 kodunu ele alınsın. H_{12} den elde edilen maksimal kodun parametreleri $(15,32,7)_2$ şeklindedir. Bu kodda 1 tane 0 ağırlıklı, 15 tane 7 ağırlıklı, 15 tane 8 ağırlıklı ve 1 tane 15 ağırlıklı kod kelime olacaktır. Bu koşullar altında çözümleme algoritmasını H_{12} ye göre düzenleyelim. $d=5$ olduğundan $e=2$ olacaktır. Dolayısıyla 0, 1, 2, 3 ağırlıklı gelen vektörleri 0 olarak ve 12, 13, 14, 15 ağırlıklı gelen vektörleri 1 olarak çözümleyebiliriz. 4, 5, 6, 7, 8, 9, 10 ağırlıklı gelen vektörler için herhangi bir 7 ağırlıklı kod kelime ile uzaklığı 3 veya daha küçükse o kod kelime olarak çözülür. 5, 6, 7, 8, 9, 10, 11 ağırlıklı gelen vektörler için herhangi bir 8 ağırlıklı kod kelime ile uzaklığı 3 veya daha küçükse o kod kelime olarak çözülür. Bu algoritma Çizelge 7.7’de verildiği gibidir.

Çizelge 7.7. H_{16} den elde edilen C_2 için yeniden düzenlenmiş en çok olabilirlik algoritması

1’lerin sayısı	Çözümleme kuralı
0, 1, 2,3	<u>0</u> olarak çözülür.
4,5,6,7,8,9,10	7 ağırlıklı kod kelimeler ile arasındaki uzaklıklara bakılır ve bu uzaklıkların biri 3 veya daha küçük ise o kod kelimeye çözülür.
5,6,7,8,9,10,11	8 ağırlıklı kod kelimeler ile arasındaki uzaklıklara bakılır ve bu uzaklıkların biri 3 veya daha küçük ise o kod kelimeye çözülür.
12,13,14,15	<u>1</u> olarak çözülür.

C_2 kodu için Yeniden düzenlemiş en çok olabilirlik algoritması ile klasik en çok olabilirlik algoritması ile işlemci zamanı bakımından karşılaştırmak için C_1 koduna benzer şekilde 10000 tane rasgele üretilmiş kod kelime belirli bit başına hata oranları(0.01, 0.10, 0.20) uygulanıp çözümleme hızları karşılaştırılmıştır. Bu karşılaştırmaya ait işlemci zamanları Çizelge 7.8’de verilmiştir.

Çizelge 7.8. C_2 için ML ve MML için çözümleme zamanları

	0.01		0.10		0.20	
	ML	MML	ML	MML	ML	MML
H_8	4.4531	3.7969	3.9844	3.3906	3.7500	2.7656
H_{12}	8.0625	6.7188	7.9531	6.2188	7.5469	5.7813
H_{16}	12.1406	10.8281	11.8438	10.5469	10.1094	8.7969

Burada C_1 'e benzer şekilde kod kelime sayısı arttıkça çözümleme zamanları artmaktadır. Yine C_1 'e benzer şekilde bit başına hata oranı arttıkça çözümleme zamanı düşmektedir. Aynı parametrelili kodlar için bit başına hata oranı arttıkça ML ile MML arasında çözümleme zamanları bakımından fark oldukça artmaktadır. Genel olarak bakıldığında çözümleme zamanları arasındaki farklar %35'e kadar çıkmaktadır ki; bu, oldukça yüksek bir rakamdır.

7.3. C_3 Kodu İçin Yeniden Düzenlenmiş En Çok Olabilirlik Algoritması

İlk olarak H_8 den elde edilen maksimal kod için yeniden düzenlenmiş en çok olabilirlik algoritmasını oluşturulsun. Bu kodun parametreleri $(6,4,4)_2$ şeklindedir, yani $e=1$ dir ve kodun yapısına dikkat edildiğinde bütün kod kelimeler 3 ağırlıklıdır. C_1 ve C_2 kodları dikkate alındığında bu kodun çözümleme algoritması daha kolaydır. $e=1$ olduğundan gelen vektörün ağırlığı 2 veya 4 ve herhangi bir kod kelimeyle uzaklığı 1 ise o kod kelime olarak çözülür. Eğer gelen vektörün ağırlığı 3 ve herhangi bir kod kelimeyle uzaklığı 0 ise o kod kelime olarak çözülür. Diğer durumlarda yani ağırlık 0, 1, 5, 6 ise kod kelime çözülmez denilir. Bu algoritma Çizelge 7.9'daki gibi özetlenir.

Çizelge 7.9. H_8 den elde edilen C_3 için yeniden düzenlenmiş en çok olabilirlik algoritması

1'lerin sayısı	Çözümleme kuralı
0,1,5,6	Çözülmez.
2,4	herhangi bir kod kelimeyle uzaklığı 1 ise o kod kelime olarak çözülür.
3	herhangi bir kod kelimeyle uzaklığı 0 ise o kod kelime olarak çözülür.

H_{12} den elde edilen maksimal kodun parameterleri $(10,6,6)_2$ dir. Burada tüm kod kelimeler 5 ağırlıklı ve $e=2$ olacaktır. Dolayısıyla eğer gelen vektörün ağırlığı 3 veya 7 ve herhangi bir kod kelimeyle uzaklığı 2 ise o kod kelime olarak çözümlenir. Eğer gelen vektörün ağırlığı 4 veya 6 ve herhangi bir kod kelimeyle uzaklığı 1 ise o kod kelime olarak çözümlenir. eğer gelen vektörün ağırlığı 5 ve herhangi bir kod kelimeyle uzaklığı 0 ise o kod kelime olarak çözümlenir. Diğer durumlarda gelen vektör çözümlenemez. Bu algoritma Çizelge 7.10'daki gibi özetlenebilir.

Çizelge 7.10. H_{12} den elde edilen C_3 için yeniden düzenlenmiş en çok olabilirlik algoritması

1'lerin sayısı	Çözümleme kuralı
0,1,2,8,9,10	Çözümlenemez.
3,7	Herhangi bir kod kelimeyle uzaklığı 2 ise o kod kelime olarak çözümlenir.
4,6	Herhangi bir kod kelimeyle uzaklığı 1 ise o kod kelime olarak çözümlenir.
5	Herhangi bir kod kelimeyle uzaklığı 0 ise o kod kelime olarak çözümlenir.

H_{16} den elde edilen C_3 maksimal kodunun çözümleme algoritması H_8 ve H_{12} 'den elde edilen C_3 kodunun çözümleme algoritmasına benzer şekilde Çizelge 7.11'deki gibi olacaktır.

Çizelge 7.11. H_{12} den elde edilen C_3 için yeniden düzenlenmiş en çok olabilirlik algoritması

1'lerin sayısı	Çözümleme kuralı
0,1,2,3,11,12,13,14,15	Çözümlenemez.
5,11	Herhangi bir kod kelimeyle uzaklığı 3 ise o kod kelime olarak çözümlenir.
6,10	Herhangi bir kod kelimeyle uzaklığı 2 ise o kod kelime olarak çözümlenir.
7,9	Herhangi bir kod kelimeyle uzaklığı 1 ise o kod kelime olarak çözümlenir.
8	Herhangi bir kod kelimeyle uzaklığı 0 ise o kod kelime olarak çözümlenir.

C_3 kodu için Yeniden düzenlemiş en çok olabilirlik algoritması ile klasik en çok olabilirlik algoritması işlemci zamanı bakımından karşılaştırmak için C_1 ve C_2 kodlarına benzer şekilde 10000 tane rasgele üretilmiş kod kelime belirli bit başına hata oranları (0.01, 0.10, 0.20) uygulanıp çözümleme hızları karşılaştırılmıştır. Bu karşılaştırmaya ait işlemci zamanları Çizelge 7.11’de verilmiştir.

Çizelge 7.12. C_3 için ML ve MML için çözümleme zamanları						
	0.01		0.10		0.20	
	ML	MML	ML	MML	ML	MML
H_8	2.1875	1.6875	2.0156	1.4531	1.5469	1.2031
H_{12}	6.9375	5.8750	5.6406	5.1719	4.1563	3.9531
H_{16}	9.4375	8.4531	8.6406	7.9063	6.9531	6.1875

Burada; diğer iki kodda olduğu gibi iki algoritma arasında çözümleme zamanları bakımından oldukça belirgin fark vardır. İki algoritma da bit başına hata oranı arttıkça çözümleme zamanlamaları azalmaktadır. Diğer iki koddan belirgin farklılık kod kelime sayısı arttıkça iki algoritma arasındaki çözümleme zamanları bakımından farkın oranının azalmasıdır. Bunun sebebi kodun sadece sabit ağırlıklı kod kelimelerden oluşmasıdır.

Bu karşılaştırmalar sonucu yeniden düzenlenmiş en çok olabilirlik çözümleme algoritması, klasik en çok olabilirlik çözümleme algoritmasına göre en az %7 daha hızlı çözümleme yaparken bu oranın %35’e kadar çıktığı görülmüştür. Dolayısıyla Hadamard tasarımlardan elde edilen maksimal kodlar söz konusu olduğunda bu algoritmanın daha avantajlı olduğu görülmüştür.

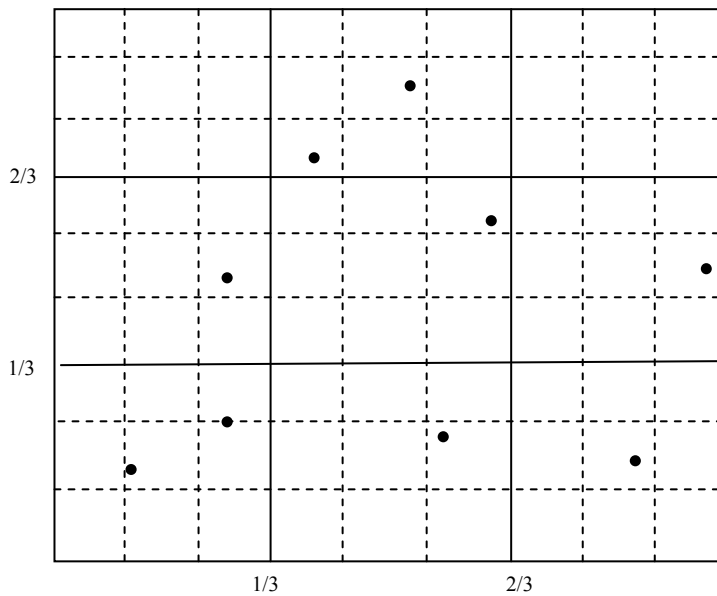
8. OPTİMAL KODLARA DAYALI SAYI ÜRETME TEKNİKLERİ

Bazı kombinatorik yapıları kullanarak rasgele sayı üretilebilir. Bu üretilen sayılardan tahmin ediciler klasik MC yöntemine göre daha etkindir.

Bu bölümde, Bölüm 2, 3 ve 4'deki Latin Hiperküpler ve Optimal Kodlar dikkate alınarak çok değişkenli dağılımlardan sayı üretilecek ve bu sayılardan yığına ait parametreler tahmin edilecektir. Ayrıca bu yöntemlerin klasik MC yöntemine göre etkinliği karşılaştırılacaktır.

Dikkate alınan yöntemler Latin Hiperküp (LH), Optimal Kod (OK) ve Optimal Koddan elde edilen Latin Hiperküplerdir (OKLH).

Klasik Monte Carlo (MC) Yöntemiyle sayı üretildiğinde sayıların dağılımı aşağıdaki gibi olabilir.



Şekil 8.1. MC yöntemiyle iki değişken için alınan 9 çaplı bir örnek

Şekil 8.1'den görüldüğü gibi alan 9 eşit parçaya bölündüğünde her parçaya sayı düşmeyebiliyor. Bunu engellemek için bazı yöntemler geliştirilmiştir. Bunlar en önemlilerinden biri Ortogonal dizime bağlı olarak sayı üretmedir.

Kullanılacak yöntem temel olarak Owen (1992)'deki ortogonal dizimlerden sayı üretme tekniğine dayanmaktadır. Bu yöntemler; Optimal kodlardan sayı üretme, Latin hiperküplerden sayı üretme ve Optimal kodlardan elde edilen Latin hiperküplerden sayı üretme şeklindedir.

8.1. Sayı Üretme

8.1.1. Optimal kodlardan sayı üretilmesi

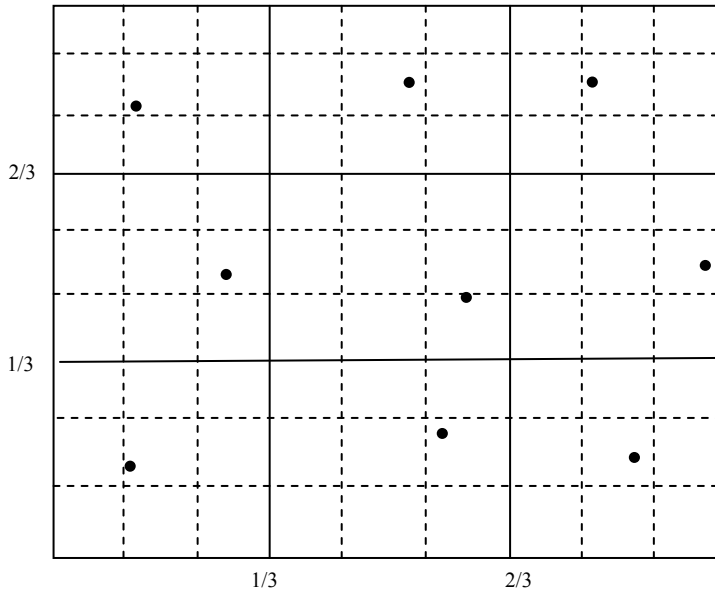
Optimal kod (OK) örneği (n, M, d, q) kodundan elde edilir. Bu kodların herhangi 2 sütunu q tane sembolün mümkün tüm ikili kombinasyonlarını içermektedir. Bu özellikten faydalanarak OK örneği

$$x_{ij} = \frac{\pi_j(a_{ij}) + u_{ij}}{q} \quad i = 1, 2, \dots, n, j = 1, 2$$

gibi yazılır. Burada a_{ij} , OK'nın herhangi iki sütununun elemanları, u_{ij} (0,1) aralığında üretilen rasgele sayılar ve π_j , $0, 1, \dots, q-1$ elemanlarının düzgün rasgele permütasyonlarıdır [Sandor ve Andras, 2004]. Elde edilen x_{ij} 'ler (0,1) aralığındadır ve düzgün dağılıma sahiplerdir.

8.1. Örnek

Örnek 4.2'deki $(9, 4, 3)_3$ optimal kodun ilk iki satırını dikkate alınsın. Buradan iki değişkene ait toplam 9 tane sayı üretilenektir. Bu sayıların dağılımı aşağıdaki gibi olur.



Şekil 8.2. OK yöntemiyle iki değişken için alınan 9 çaplı bir örnek

Şekil 8.2’den görüleceği gibi OK örnekleme alanı $1/9$ olan her kareye 1 örnek düşmesini garanti eder MC yönteminde ise böyle bir durumdan her zaman için söz edilmez. Dolayısıyla OK yönteminde yığını daha iyi bir temsil söz konusudur.

8.1.2. Latin Hiperküplerden sayı üretilmesi

$OA(s, k, s, I)$ parametrelili bir Latin Hiperküp mevcut olsun. Böyle bir yapıdan seçilecek örnek OK örneği gibi her $1/9$ luk kareye 1 örnek düşmesini garanti etmez. Fakat her iki değişken teker teker ele alındığında örneklerin aralıklara dağılımı MCya ve OKya göre daha iyidir. LH’yi kullanarak elde edilen örnek

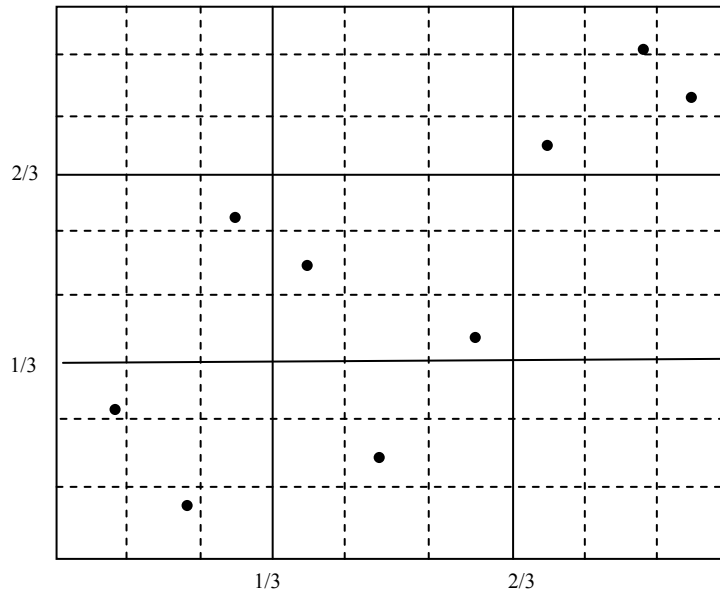
$$x_{ij} = \frac{\pi_j(a_{ij}) + u_{ij}}{s} \quad i = 1, 2, \dots, s, j = 1, 2$$

gibi yazılır. Burada OK ya benzer şekilde a_{ij} LH’nin herhangi iki sütununun elemanları, u_{ij} $(0,1)$ aralığında üretilen rasgele sayılardır. Fakat OKdan farklı olarak $\pi_j, 0, 1, \dots, s-1$ elemanlarının düzgün rasgele permütasyonlarıdır [Sandor ve Andras, 2004]. Burada elde edilen x_{ij} ’ler $(0,1)$ aralığındadır ve düzgün dağılıma sahiplerdir.

8.2. Örnek

Aşağıdaki gibi bir LH olsun. Buradan elde edilecek örnek şöyle olacaktır.

$$OA(9,2,9,1) = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 0 & 5 & 4 & 1 & 3 & 6 & 8 & 7 \end{bmatrix}$$



Şekil 8.3. LH yöntemiyle iki değişken için alınan 9 çaplı bir örnek

Şekil 8.3'den görüleceği gibi LH örnekleme, alanı 1/9 olan her kareye 1 örnek düşmesini garanti etmemekte fakat değişkenler tek tek ele alındığında, yığını MC ve OK yöntemlerinden daha iyi temsil ettiği açıktır.

8.1.3. Optimal koddan elde edilen latin hiperküplerden sayı üretilmesi

Bu örnekleme tasarımı, Tang (1993) tarafından geliştirilen Ortogonal dizimlerden elde edilen Latin hiperküp örnekleme tasarımının özel bir durumu olarak düşünülebilir. Optimal koddan elde edilen Latin hiperküp (OKLH) örnekleme tasarımı kısaca şu şekildedir. İlk önce Bölüm 8.1.1'deki gibi her bir sütuna düzgün rasgele permütasyon uygulanır. Daha sonra her sütundaki k elemanları yerine $k(q+1)$, $k(q+1)+1, \dots, k(q+1)+q=(k+1)(q+1)-1$ elemanlarının rasgele permütasyonu konulur.

Elde edilen matris OK'ya dayalı LH matrisidir. Bu matrsten elde edilecek rasgele örnek;

$$x_{ij} = \frac{a_{ij} + u_{ij}}{s} \quad i = 1, 2, \dots, s, j = 1, 2$$

şeklinde olacaktır.

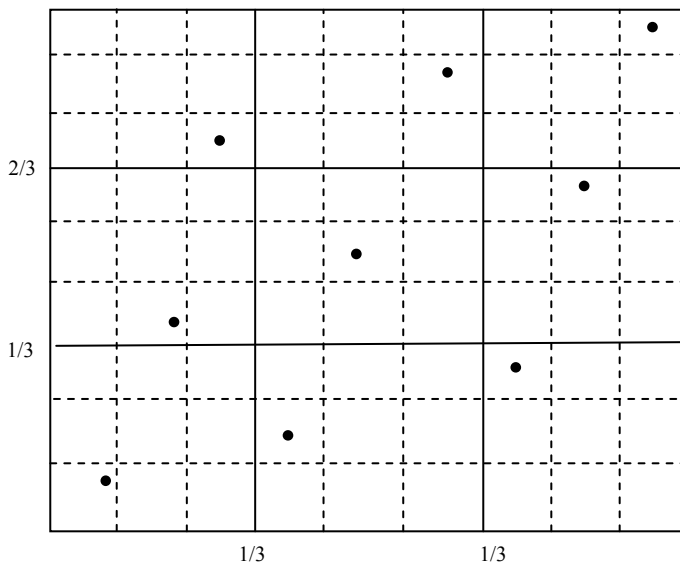
Bunun için bir örnek aşağıdaki gibidir.

8.3. Örnek

Örnek 4.2'deki optimal kodun ilk iki sütununu tekrar dikkate alalım. Bu OK'dan üretilecek LH aşağıdaki gibi olacaktır.

$$OKLH = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 0 & 3 & 6 & 1 & 4 & 7 & 2 & 5 & 8 \end{bmatrix}$$

OKLH'den elde edilecek sayıların dağılımı aşağıdaki gibi olur.



Şekil 8.4. OALH yöntemiyle iki değişken için alınan 9 çaplı bir örnek

Şekil 8.4'den görüldüğü gibi her 1/9 luk alanlardan birer örnek alınmıştır aynı zamanda değişkenler teker teker incelendiğinde LH kadar yığını iyi temsil ettiği görülmektedir.

Bu bölümün geri kalanında bu üç yöntemi klasik Monte Carlo yöntemine göre karşılaştırmak amacıyla çok değişkenli normal dağılımlardan rasgele sayı üretilip belirli bir sabit vektörden küçük olması olasılığının beklenen değeri ve varyansları üzerinde durulacaktır.

Bunun için ilk önce üretilen sayılarla iki değişkenli Normal dağılım için olasılık hesaplaması üzerinde durulacaktır.

8.2. Çok Değişkenli Normal Dağılım İçin Olasılık Hesaplaması

$X \sim N(\mu, \Sigma)$ olmak üzere $p = P(X < x_0)$ olasılığı hesaplanmak istensin. Burada x_0 , reel sayıların r boyutlu bir vektörü, $v = x_0 - \mu$ ve T, Σ 'nın alt üçgensel kök matrisi olsun. Böylece,

$$p = P(X < x_0) = P(Te < v), \quad e \sim N(0, I_r)$$

olur. T matrisi;

$$T = \begin{bmatrix} t_{11} & 0 & \cdots & 0 \\ t_{21} & t_{22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ t_{n1} & t_{n2} & \cdots & t_{nn} \end{bmatrix}$$

ve $v = (v_1, v_2, \dots, v_r)$ olsun. elde edilecek integral;

$$p = \int_D \phi(e_1) \dots \phi(e_r) de_1 \dots de_r \quad (8.1)$$

şeklindedir. D ile gösterilen tanım bölgesi;

$$D = \left\{ (e_1, \dots, e_r) \in \mathbb{R}^r : e_1 < \frac{v_1}{t_{11}}, e_2 < \frac{v_2 - t_{21}e_1}{t_{22}}, \dots, e_r < \frac{v_r - t_{r1}e_1 - \dots - t_{rr-1}e_{r-1}}{t_{rr}} \right\}$$

Olacaktır. Burada ϕ standart Normal dağılımı temsil eder. $e < b$ ve $e \sim N(0,1)$ ise e 'leri, $(0,1)$ aralığında tekdüze dağılımdan secilen u 'ları $e = \Phi^{-1}(u\Phi(b))$ fonksiyonunda kullanarak elde edilir [Börsch-Supan ve Hajivassiliou, 1993, Vijverberg, 1997]. Burada. Φ Standard Normal dağılım fonksiyonudur. Aşağıdaki dönüşümü kullanarak;

$$\begin{aligned} e_1 &= \Phi^{-1} \left(u_1 \Phi \left(\frac{v_1}{t_{11}} \right) \right) \\ e_2 &= \Phi^{-1} \left(u_2 \Phi \left(\frac{v_2 - t_{21}e_1}{t_{22}} \right) \right) \\ &\dots \\ e_r &= \Phi^{-1} \left(u_r \Phi \left(\frac{v_r - t_{r1}e_1 - \dots - t_{rr-1}e_{r-1}}{t_{rr}} \right) \right) \end{aligned}$$

Eş. 8.1'deki integrali;

$$p = \Phi \left(\frac{v_1}{t_{11}} \right) \int_{(0,1)^{r-1}} \Phi \left(\frac{v_2 - t_{21}e_1}{t_{22}} \right) \dots \Phi \left(\frac{v_r - t_{r1}e_1 - \dots - t_{rr-1}e_{r-1}}{t_{rr}} \right) du_1 \dots du_{r-1} \quad (8.2)$$

şeklinde olacaktır. Böyle elde edilen bir tahmin ediciye Geweke, Hajivassilou ve Keane(GHK) simülatörü [Genz, 1992] yada geri dönüşlü önem örnekleme (GDÖÖ-RIS) [Vijverberg, 1997] olarak adl andırılır. Çalışmada $r=3$ durumu dikkate alınacaktır.

Burada, üç değişkenli normal dağılımdan sayı üretilip farklı kovaryans matrisleri altında $P(X < x_0)$ olasılıklarının dört yönteme göre tahminleri karşılaştırılacaktır.

Ayrıca OK, LH ve OKLH tahmin edicilerinin MC tahmin edicisine göre göreceli etkinlik değerleri elde edilerek birbirleriyle karşılaştırılacaktır. Göreceli etkinlik değerleri;

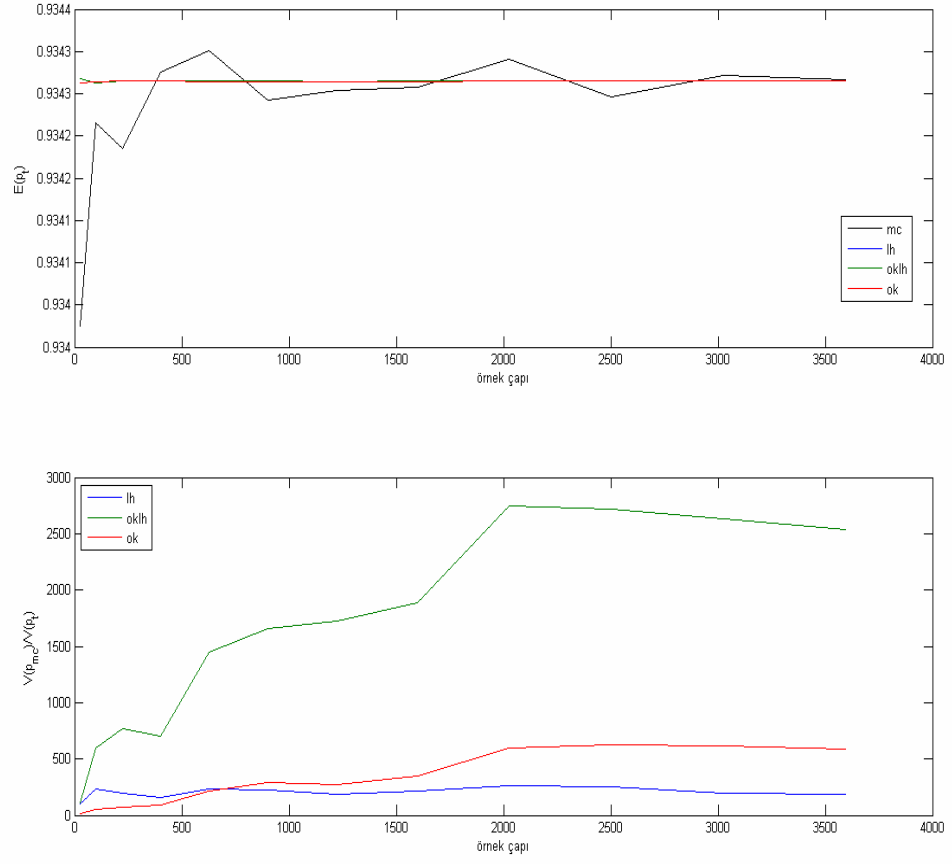
$$GE = \frac{V(p_{MC})}{V(p_t)}$$

şeklinde elde edilir. Burada $V(p_{MC})$ ve $V(p_t)$ değerleri sırasıyla MC ile LH, OK, OKLH tahmin edicilerinin varyanslarıdır.

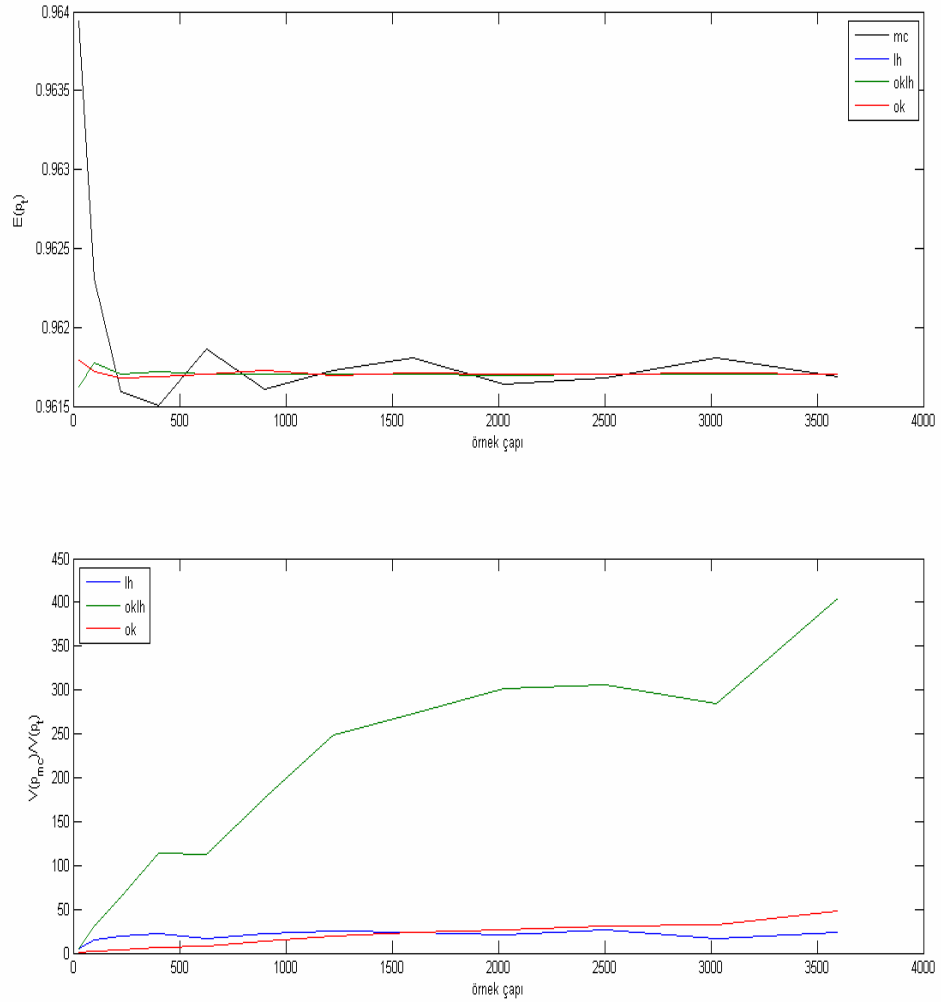
Karşılaştırma için X ortalama vektörü $\mu=(4,4,4)$ ve kovaryans matrisi;

$$C = \begin{bmatrix} 1 & r & r \\ r & 1 & r \\ r & r & 1 \end{bmatrix}$$

olan normal dağılım dikkate alınarak $r=0.1$ ve $r=0.9$ için $x_0=(6,6,6)$ alınmıştır.



Şekil 8.5. $r=0.1$ için $P(X < x_0)$ olasılığının OK, OKLH, LH ve MC tahminleri



Şekil 8.6. $r=0.9$ için $P(X < x_0)$ olasılığının OK, OKLH, LH ve MC tahminleri

Şekil 8.5-Şekil 8.6 dikkate alındığında dört yöntemin de sapmasız olduğu görülmektedir. Ayrıca OKLH her iki durumda da diğer yöntemlere göre çok daha üstün sonuçlar verdiği görülebilir. Her iki grafikten görülebileceği gibi belli bir örnek çapına kadar LH, OK'dan daha etkin iken, belli bir örnek çapından sonra OK LH'dan daha iyi sonuç vermektedir.

9. SONUÇ

Bu tez çalışmasında, bazı kombinatorik tasarımları kullanarak elde edilen maksimal optimal kodlar ele alınarak özellikleri incelenmiş ve uygulamaları yapılmıştır. Bu özellikler orijinal örnekler üzerinde uygulanmıştır.

Bu kombinatorik tasarımlardan, Hadamard 2-düzenlerden elde edilen 3 farklı maksimal kod ve ikili ortogonal Latin kare tam setinden oluşturulan Afin yapısındaki çözülebilir düzeninden elde edilen optimal kodlar en çok olabilirlik çözümlemesi yöntemiyle karşılaştırılmış ve çeşitli boyutlardaki dijital fotoğraflar üzerine uygulaması yapılmıştır.

Ayrıca farklı kod kelime sayısı olan M değerleri ve farklı bit başına hata oranları için bu kodları karşılaştırmak için gönderilen kod kelime sayısı için 1000:1000:50000 değerleri, bit başına hata oranı için ise 0.01:0.01:0.50 değerleri dikkate alınarak Hadamard tasarımından elde edilen maksimal ve afin çözülebilir tasarımlardan elde edilen optimal kodlar üretilmiştir. Ayrıca Hadamard tasarımından üretilen kodlar (C_1 , C_2 , C_3) için $M=16, 24, 32, 48, 64$ değerleri, çözülebilir tasarımdan oluşturulan C_4 kodu için ise $M=16, 64, 256$ olan kodlar dikkate alınmıştır. Bunun sonucunda eğer kullanılacak kodun kod kelimeleri düşük ve bit başına hata oranı düşük ise hem doğru çözme oranı hem de hızlı çözme bakımından C_2 kodu diğerlerine göre daha iyidir. Ancak kodun kod kelimeleri düşük ise fakat bit başına hata oranı yüksek ise doğru çözme oranı bakımından C_3 kodu tercih edilebilir. Özellikle kodun kod kelimeleri yüksek ve aynı zamanda bit başına hata oranı yüksek ise C_4 kodu kullanılır. Fakat C_4 kodunun en büyük dezavantajı sadece M 'in 2^4 , 2^6 veya 2^8 gibi 2'nin çift kuvvetleri için var olmasıdır. Diğer 3 kod ise M , 4'ün katı olduğu herhangi bir değer için bulunabilir.

Karşılaştırılan bu kodların hangi durumlarda birbirlerine üstünlük sağladığı belirlenmiştir.

İlave olarak, Hadamard 2-tasarımlardan elde edilen 3 farklı maksimal kod ve ikili ortogonal Latin kare tam setinden oluşturulan Afin yapısındaki çözülebilir düzeninden elde edilen 2 optimal kod ve 2-tasarımlardan elde edilen maksimal sabit ağırlıklı kodlar olasılık yapılar teorik olarak incelenmiş ve bazı koşullar altında karşılaştırılmış ve üstün oldukları noktalar belirtilmiştir

PG kullanılan kodun çözümleme zamanları ile klasik EÇOyu kullanan kodun çözümleme zamanlarına göre belirgin bir şekilde hızlı olduğu bölüm 4te gösterilmiştir. Bu fark gönderilen kod kelime sayısı arttıkça yükselmektedir. Aralarındaki fark yaklaşık olarak %20 civarındadır. 0.05 hata oranında doğru çözümleme oranı 0.995, 0.10 hata oranında doğru çözümleme oranı 0.95 ve 0.15 hata oranında ise doğru çözümleme oranı 0.85 civarındadır. Doğru çözümleme oranını %95 olarak kabul edilirse bu kod %10 hata oranına kadar kullanılabilir.

Daha sonra Bölüm 7’de bu algorithmadan faydalanarak en çok olabilirlik çözümleme algoritması yeniden düzenlenmiş ve Hadamard tasarımlardan elde edilen maksimal kodlara uygulanacak ve çözümleme zamanları bakımından karşılaştırılmıştır. Bu karşılaştırma sonucu yeniden düzenlenmiş en çok olabilirlik çözümleme algoritması, klasik en çok olabilirlik çözümleme algoritmasına göre %7 ile %35 arası daha hızlı çözümleme yaptığı görülmüştür. Dolayısıyla Hadamard tasarımlardan elde edilen maksimal kodlar söz konusu olduğunda bu algoritma kullanılabilir.

Son olarak, bazı kombintorik tasarımlardan bulunan kodlardan elde edilen sayı üretme yöntemleri verilmiştir. Verilen yöntemlerden elde edilen tahmin ediciler MC yöntemine göre oldukça etkin olduğu görülmüştür.

KAYNAKLAR

- Ang, R., Seberry, J., Wysocki, B.J., Wysocki, T.A., "Application of nega-cyclic matrices to generate spreading sequences", *ISCTA'2003*, Ambleside, U.K., 46-68 (2003).
- Baumert, L.D., Hall, Jr. M., "Hadamard matrices of the Williamson type", *Math. Comput.*, 19:442-447 (1965).
- Beutelspacher, A. and Rosenbaum, U., "Projective Geometry", *Cambridge University Press*, Cambridge, 31-69 (1998).
- Beth, T., Junknikel, D., Lenz, H., "Design Theory", *Cambridge University Press*, Cambridge, 76-110 (1999).
- Bose, R.C., "A note on the resolvability of balanced incomplete block designs", *Sankhya*, 6:105-110 (1942).
- Bose, R.C. and Bush, K. A., "Orthogonal Arrays", *Ann. Math. Stat.*, 23(4):508-524 (1952).
- Bose, R.C. and Kishen, K., "On the problem of confounding in the general symmetrical factorial design", *Sankhyā*, 5:21-36 (1940).
- Bose, R.C., Shrikhande, S.S., "A note on result in the theory of code construction", *Inf. and Control*, 2:183-194 (1959).
- Börsch-Supan, A. and Hajivassiliou V.A., "Smooth unbiased multivariate probability simulators for maximum likelihood estimation of limited dependent variable models", *Journal of Econometrics*, 58:347-368 (1993).
- Bruck, R.H. and Ryser H.J., "The nonexistence of certain finite projective planes", *Canadian J. Math.*, 1:88-93 (1949).
- Cameron, P.J., "Combinatorics: Topics, Techniques, Algorithms", *Cambridge University Press*, Cambridge, 110-125 (1996).
- Cameron, P.J., Lint J.H., "Design, Graph, Codes and their links", *Cambridge University Press*, Cambridge, 20-37 (1996).
- Chowla, S., Ryser, H.J., "Combinatorial Problems", *Canadian J. Math.*, 2:93-99 (1950).
- Chung, F., Salehi, J.A., Wei, V. K., "Optical orthogonal codes: Design, analysis, and applications", *IEEE Trans. Information Theory*, 35: 595-604 (1989).

Comtet, L., “Advanced Combinatorics: The Art of Finite and Infinite Expansions”, *Rev. Enl. Ed. Dordrecht*, Netherlands, 52-73 (1974).

Genz, A., “Numerical computation of the multivariate normal probabilities”, *Journal of Computational and Graphical Statistics*, 1: 141–150 (1992).

Gökpınar, F., “A simulation study on decoding of binary linear codes using projective geometry $PG(3,2)$ ”, *G.U. Journal of Science*, 20(3):127-132 (2007).

Graham, R.L. Grötschel, M., Lovasz, L., “Handbook of Combinatorics, vol. 1”, *North-Holland* Amsterdam, 46-85 (1995).

Graham, R.L. Grötschel, M., Lovasz, L., “Handbook of Combinatorics, vol. 2”, *North-Holland*, Amsterdam, 26-67 (1995).

Hadamard, J., “Résolution d'une question relative aux déterminants”, *Bull. Sci. Math.*, 17:240-246 (1893).

Hall, J.R., “Combinatorial Theory”, *Wiley*, New York, 105-117 (1986).

Harmuth, H.F., “Transmission of information by orthogonal functions”, *Springer-Verlag*, Berlin, 48-95 (1960).

Hedayat, A.S., Sloane, N.J.A, Stufken, J., “Orthogonal Arrays”, *Springer*, New York, 15-39 (1999).

Hoffman, D.G., Leonard, D.A., Lindner, C.C., Phelps, K.T., Rodger, C.A., Wall, J.R., “Coding Theory”, *Markel Deccer, Inc.*, New York, 24-57 (1992).

Huffman, W.C., and Pless, V., “Fundamentals of Error-Correcting Codes”, *Cambridge University Press*, Cambridge, 76-87 (2003).

Jain, A.K., “Fundamentals of digital image processing”, *Prentice-Hall*, Englewood Cliffs, 73-78 (1989).

Johnson, S.M., “A new upper bound for error-correcting codes”, *IRE Trans. Information Theory*, 203–207 (1962).

Kirkman, T.P., “On a Problem in combinatorics”, *Cambridge and Dublin Math.*, 2:191-204 (1847).

Kishen, K., “On Latin and hypergraecolatin cubes and hypercubes”, *Current Science*, 11:98-99 (1942).

Mathon, R., Rosa, A., “ $2-(v,k,\lambda)$ Designs of Small Order”, CRC Handbook of Combinatorial Designs, 1st ed., Colbourn C.J. and Dinitz J.H, *CRC Press*, Boca Raton, 3-41 (1996).

Merris, R., “Combinatorics”, *Wiley*, New York, 117-120 (2003).

Miao, Y., Kageyama, S., “Two classes of q -ary codes based on group divisible association schemes”, *Discrete Mathematics*, 195:269-276(1999).

Miyamoto N., Mizuno H., Shinohara S., “Optical orthogonal codes obtained from conics on finite projective planes”, *Finite Fields and Their Applications*, 10:405–411 (2004).

Omran, R., Moreno, O., Kumar, P. V., “Optimal optical orthogonal codes with $\lambda > 1$ ”, *Proc. Int. Symposium on Information Theory*, 366-378 (2004).

Owen, A.B., “Orthogonal arrays for computer experiments, integration and visualization”, *Statistica Sinica*, 2: 439–452 (1992).

Peterson, W.W., “Error correcting codes”, *The M.I.T. Press*, New York, 56-87 (1961).

Plotkin, M., “Binary codes with specified minimum distance”, *IRE Transactions*, 6: 445-450 (1960).

Preece, D.A., “Nested balanced incomplete block designs”, *Biometrika*, 54:479-486 (1967).

Raghavarao, D., “Constructions and combinatorial problems in design of experiments”, *Wiley*, New York, 126-137 (1971).

Sandor, Z. and Andras, P., “Alternative sampling methods for estimating multivariate normal probabilities”, *Journal of Econometrics*, 120:207-234 (2004).

Semakov, N.V., Zinoviev, V.A., “Equidistant q -ary codes with maximal distance and resolvable incomplete block designs”, *Problemi peredatchi informatsii*, 4(2):3-10 (1968).

Semakov, N.V., Zinoviev, V.A., “Equidistant codes and tactical configurations”, *Problemi peredatchi informatsii*, 5(3): 28-37(1969).

Sinha, K., “A class of q -ary codes”, *Discrete Mathematics*, 126:439-940 (1994).

Sinha, K., Mitra, R.K., “Construction of nested balanced block designs, rectangular designs and q -ary codes”, *Annals of Combinatorics*, 3:71-80(1999).

Sylvester, J., “Thoughts on Orthogonal Matrices, Simultaneous Sign-Successions, and Tessellated Pavements in Two or More Colours, with Applications to Newton's Rule, Ornamental Tile-Work, and the Theory of Numbers”, *Phil. Mag.*, 34:461-475 (1867).

Tang, B., “Orthogonal array-based Latin hypercubes”, *Journal of the American Statistical Association*, 88: 1392–1397 (1993).

Tonchev, V.D., “Combinatorial Configurations Designs, Codes, Graphs”, *Longman*, London, 55-76 (1988).

Tonchev, V.D., “Codes and Designs”, Handbook of Coding Theory, Pless V.S. and Huffman W.C., 1st ed., *Elsevier Science B.V.*, New York, 1229-1267 (1998).

Tonchev V.D., “Codes”, CRC Handbook of Combinatorial Designs, Colbourn C.J. and Dinitz J.H., *CRC Press*, Boca Raton, 517-543 (1996).

Tonchev, V.D., “Affine designs and linear orthogonal arrays”, *Discrete Math.* 294:219-222 (2005).

Tressler, E., “A survey of the Hadamard Conjecture”, Ms. Thesis, *Faculty of the Virginia Polytechnic*, Institute and State University, 72-78 (2004).

Turyn, R.J., “An infinite class of Williamson matrices”, *J. Combin. Theory Ser. A*, 12:319–321 (1972).

Vijverberg, W.P.M., “Monte Carlo evaluation of multivariate normal probabilities”, *Journal of Econometrics*, 76: 281–307 (1997).

Walsh, J.L., “A closed set of normal orthogonal functions”, *American J. of Mathematics*, 55:5–24 (1923).

Yarlagadda, R.K., Hershey, J.E., “Hadamard Matrix Analysis and Synthesis: with Applications to Communications and Signal/Image Processing”, *Kluwer*, New York, 64-76 (1997).

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : GÖKPINAR, Fikri
 Uyruğu : K.K.T.C. ve T.C.
 Doğum tarihi ve yeri : 13.04.1978 / Lefkoşa
 Medeni hali : Bekar
 Telefon : 0 (312) 202 14 83
 e-mail : fikri@gazi.edu.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Yüksek lisans	Gazi Üniversitesi /İstatistik Bölümü	2002
Lisans	Gazi Üniversitesi/ İstatistik Bölümü	1999
Lise	Türk Maarif Koleji	1995

İş Deneyimi

Yıl	Yer	Görev
2000-2007	Gazi Üniversitesi	Araştırma Görevlisi

Yabancı Dil

İngilizce, Almanca

Yayınlar

- ÖZDEMİR Y.A., GÖKPINAR F., 2007, A new Formula For Inclusion Probabilities in Median Ranked Set Sampling, Communications in Statistics-Theory and Methods.
- EBEGİL M., GÖKPINAR F., EKNI M., 2007, Bühlmann Kredibilite Modelinde Rassal Etki Modelinin Kullanılması üzerine bir çalışma, Sakarya Üniversitesi Fen Edebiyat Fakültesi Fen-Edebiyat dergisi,9(1):67-82.

3. GÖKPINAR F., 2007, A simulation study on decoding of binary linear codes using projective geometry PG(3,2), G.U. Journal of Science, 20(3):127-132.
4. ÖZDEMİR Y.A., GÖKPINAR F., 2006, A Generalized Formula for Inclusion Probabilities in Ranked Set Sampling, Hacettepe Journal of Mathematics and Statistics, Vol 36(1), 89-99
5. ÖZDEMİR Y.A., GÖKPINAR F., 2006, Hypothesis Testing For The Population Mean Using Unbiased Ranked Set Sampling Designs, International Journal of Pure and Applied Mathematics, Vol. 31 No. 4, 2006, 501-513
6. EBEGİL M., GÖKPINAR F., EKİNİ M., 2006, A Simulation Study on the Shrinkage Estimators, Hacettepe Journal of Mathematics and Statistics
7. GÖKPINAR F., 2006, Hadamard Tasarımlardan Elde Edilen Maksimal Kodların Karşılaştırılması ve Bir Uygulama, 5.İstatistik Günleri Sempozyumu, Ankara Üniversitesi, Antalya.
8. GÖKPINAR F., 2006, Çözülebilir Tasarımlardan Elde Edilen Optimal Kodların Olasılık Yapıları, Gazi Üniversitesi Matematik Çalıştayı
9. ÇEREZCİ E.T., GÖKPINAR F., 2005, Comparing Linear Programing Models used in the Least Absolute Value Method, Hacettepe Journal of Mathematics and Statistics, Vol. 34, 95-102
10. ERKMEN E., ATAÇ M.S., GÜNGÖR N., AKALIN V., GÖKPINAR F., 2005, Comparison Of Patient Satisfaction With Complete Dentures After Vestibuloplasty And Implant Retained Overdentures In Extremely Resorbed Mandibles: A Clinical Investigation, Journal of Gazi University Faculty of Dentistry, 22(2):89-96.
11. GÖKPINAR F., 2005, Sürekli Değişkenler İçeren Grafiksel Modellerde Kullanılan Sapma ve F-İstatistiklerinin Karşılaştırılması, İstatistik Araştırma Dergisi, Vol. 04, No. 2:55-64.
12. BAYRAK H., GÖKPINAR F., 2005, Koşullu Bağımsızlık Yapılarında Grafiksel Modeller(Kıtap).
13. GÖKPINAR F., ÖZDEMİR Y.A., A.Alptekin ESİN, 2005, Comparing The Efficiency Of The Estimators For The Population Mean Under Different Designs Of Ranked Set Sampling, G.U. Journal Of Science, 18(3) 187-197.
14. TEKEL N., VEZİROĞLU F., KOCABAY C., GÜMGÜM, ERKMEN E., GÖKPINAR F., 2005, The Evaluation of different Measure Scales For Postoperative Pain and Trismus after the Removal of Impacted Mandibular Third Molar: A Preliminary Report, Turkish Association of Oral and Maxillofacial Surgery, 13th International Scientific Congress, 29 May- 2 June, Antalya Turkey

15. GÖKPINAR F., BAYRAK H., 2005, Simetrik 2-dizaynlar ve Maksimal kodlar, 4. İstatistik kongresi, Bildiriler Kitabı, 275-280, 8-12 Mayıs, Antalya.
16. GÖKPINAR F., BAYRAK H., 2005, Eşit Uzaklıklı Kodlar, Çözülebilir Dizaynlar ve Afin Geometri, Bildiriler Kitabı, 595-600, 4. İstatistik kongresi, 8-12 Mayıs, Antalya.
17. ŞAHİN S.T., GÖKPINAR F., 2004, Sonradan Tabakalamada Üstten Kesilmiş Çoklu Ters Örneklem Tahmin Edicisi İle Diğer Tahmin Edicilerin Karşılaştırılması, 4. İstatistik günleri 2004, 121-126, Bildiriler kitabı, Dokuz Eylül Üniversitesi.
18. ÖZDEMİR Y.A, GÖKPINAR F., A.Alptekin ESİN, 2004, Sıralı Küme Örneklemesi Tasarımlarının Etkinliklerinin Karşılaştırılması Üzerine Bir Çalışma, 4. İstatistik günleri 2004, 127-134, Bildiriler kitabı, Dokuz Eylül Üniversitesi
19. GÖKPINAR F., BAYRAK H., 2004, On The Projective Geometry and Linear Codes, 4.İstatistik günleri 2004, 393-399, Dokuz Eylül Üniversitesi
20. ÖZKAYA B., ÖZKAYA H., BAYRAK H., GÖKPINAR F.,2004, Erişte kalitesine kurutma işlemlerinin etkileri, 60-66, Geleneksel Gıdalar Sempozyumu 2004.
21. ÖZKAYA H., ÖZKAYA B., BAYRAK H., GÖKPINAR F.,2004, Bulgurun fitik asit içeriğine prosesin etkisi, 54-59, Geleneksel Gıdalar Sempozyumu 2004.
22. ÇELEBİOĞLU S., OLMUŞ H., GÖKPINAR F., ÖRKÇÜ, H.H., 2004, Ankaranın başkent oluşunun 80. yılında nüfus istatistik üzerine bir çalışma, 59-74, Cumhuriyetin 80. yılında her yönüyle Ankara, Gazi Üniversitesi
23. GÖKPINAR F., BAYRAK H., 2004, Kesikli Değişkenler için Grafiksel Modeller, Anadolu Ün. Bilim ve teknolojileri Dergisi, Vol.5 No:2, 195-200
24. ERKMEN E., YÜCEL E., KUZU I., GÖKPINAR F., ATAÇ M.S., 2004, Comparison of effectiveness of two different allogeic resorbable membranes used for guided bone regeneration in defective mandibles: an experimental study, Journal of Hacettepe Faculty of Dentistry, Vol.28(4), 16-25.
25. BAYRAK H., GÖKPINAR F., ÖZKAYA B., 2003, Sürekli Değişkenler içeren Grafiksel modeller, İSTATİSTİK ARAŞTIRMA DERGİSİ, cilt 2, No 1, 79-88
26. BAYRAK H., GÖKPINAR F., 2003, Koşullu Gauss Dağılımı ve Etkileşimleri, İSTATİSTİK ARAŞTIRMA DERGİSİ, cilt 2, No 3, 11-20.

27. GÖKPINAR F., BAYRAK H., 2003, Sürekli değişkenler arasındaki koşullu bağımsızlıkların Grafiksel modeller ile bulunması, Ekonometri Sempozyumu Bildiriler CD'si, Gazi Üniversitesi
28. GÖKPINAR F., BAYRAK H., 2003, Karma Değişkenler İçeren Modellerde Koşullu bağımsızlık Yapılarının İncelenmesi, 213-216, 3. İstatistik kongresi, 16-20 Nisan, Antalya.
29. GÖKPINAR F., BAYRAK H., 2002, Conditional Independence Structures Of Graphical Models, JOURNAL OF QAFQAZ UNIVERSITY, No 9, 225-233
30. ÖZKAYA B., BAYRAK H., GÖKPINAR F., 2002, Ekmeğin Fitik Asit Miktarına Etkili Bazı Faktörlere Grafiksel Model İle Yaklaşım Hububat 2002, 543-552(2).
31. ÖZKAYA B., BAYRAK H., GÖKPINAR F., 2002, Unun Ekmeklik Kalitesini Etkileyen Bazı Faktörler Arasındaki ilişkilerin Kovaryans Seçimli Model İle Belirlenmesi, Hububat 2002, 557(1)-555.
32. GÖKPINAR F., BAYRAK H., 2002, Karma Değişkenler İçin Grafiksel Modeller, 3. İstatistik günleri Hacettepe Üniversitesi

Hobiler

Sinema, Bilgisayar teknolojileri, Basketbol