

**T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

AĞ CEBİRSEL KODLARI VE UYGULAMALARI

AYŞEGÜL BAYRAM

**YÜKSEK LİSANS TEZİ
MATEMATİK MÜHENDİSLİĞİ ANABİLİM DALI**

**DANIŞMAN
PROF. DR. İRFAN ŞİAP**

İSTANBUL, 2012

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

AĞ CEBİRSEL KODLARI VE UYGULAMALARI

Ayşegül BAYRAM tarafından hazırlanan tez çalışması 30/01/2012 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Matematik Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Prof. Dr. İrfan ŞİAP

Yıldız Teknik Üniversitesi

Jüri Üyeleri

Prof. Dr. İrfan ŞİAP

Yıldız Teknik Üniversitesi



Doç. Dr. Gürsel YEŞİLOT

Yıldız Teknik Üniversitesi



Yrd. Doç. Dr. Bahattin YILDIZ

Fatih Üniversitesi



ÖNSÖZ

Eğitim hayatım boyunca beni özveri ile yetiştiren tüm öğretmen ve öğretim üyesi hocalarıma, bilhassa bu tezi hazırlamamda tecrübesini ve zamanını esirgemeyen çok değerli hocam Prof. Dr. İrfan ŞİAP' a ve burs vererek eğitimime destek sağlayan TÜBİTAK'a şükranlarımı sunarım.

Ayrıca bugüne kadar benden maddi ve manevi desteklerini esirgemeyen sevgili aileme, özellikle canım babama teşekkür ederim.

Ocak, 2012

Ayşegül BAYRAM

İÇİNDEKİLER

	Sayfa
ÖNSÖZ	iii
İÇİNDEKİLER.....	iv
SİMGE LİSTESİ.....	vi
KISALTMA LİSTESİ	viii
ÇİZELGE LİSTESİ	xi
BÖLÜM 1	
GİRİŞ	1
1.1 Literatür Özeti	1
1.2 Tezin Amacı	1
1.3 Hipotez	2
BÖLÜM 2	
GRAF KAVRAMI VE LİNEER KODLAR	3
2.1 Graf Kavramı (Çizge Teorisi)	3
2.2 Lineer Kodlar	12
2.2.1 Self Dual ve Self Ortogonal (Kendine Dual ve Kendine Dik) Kodlar	14
2.2.2 Üreteç ve Kontrol Matrisleri	15
2.2.3 Hamming Uzaklık- Hamming Ağırlık.....	19
2.2.4 Lineer Kodların Denklığı	23
2.3 Ağ Bağlantı Kodları ve Tanımlar	23
2.4 Ağ Bağlantı Kodlamasının Faydaları	26
BÖLÜM 3	
AĞ BAĞLANTILARI	27
3.1 Devirsiz Ağ Bağlantıları.....	27
3.1.1 Devirsiz Ağ Bağlantıları ve Lineer Ağ Bağlantı Kodları.....	27
3.2 Devirli Ağ Bağlantıları	36
3.2.1 Devirli Ağ Bağlantıları ve Lineer Ağ Bağlantı Kodları.....	36
3.3 Ağ Bağlantı Kodlamasına Cebirsel Bir Yaklaşım	40
3.3.1 Cebirsel Modelleme	43
3.3.2 Genel Ağ Bağlantı Kodlama Problemi	52
BÖLÜM 4	
HATA DÜZELTEBİLEN AĞ BAĞLANTI KODLARI	58

4.1 Genel Hata Düzeltilebilen Ağ Bağlantıları	58
4.2 Lineer Ağ Bağlantı Kodlarının Hata Düzeltme Kapasitesi.....	59
4.3 Lineer Ağ Bağlantı Kodlarının Hata Belirleme Kapasitesi.....	67
BÖLÜM 5	
SONUÇ VE ÖNERİLER	70
ÖZGEÇMİŞ	72

SİMGE LİSTESİ

y	Alınan vektör
$F(x, z)$	Alınan vektörler
w	Alınan vektörlerin ağırlığı
Φ	Alınan vektörlerin kümesi
E	Ayrıt kümesi
$Y(.)$	Ayrıtlar üzerindeki işlemler
c	Bağlantı
$R(.)$	Bağlantının hızı
$\mathfrak{Z}$	Bağlantı kümesi
$Z(.)$	Çıktı işlemlerinin kümesi
$\mathcal{G}$	Çıktı işlemlerinin sayısı
T	Düğüm
V	Düğüm kümesi
$C^\perp$	Dual kod
F_e	e ayrıtı üzerindeki işlemler
$\overline{F_e}$	e ayrıtının girdi işlemleri
$C(e)$	e kanalının kapasitesi
$\overline{\beta_e}$	e ayrıtı üzerindeki yerel enkodlama fonksiyonu
k_e	e kanalına etki eden yerel kodlama çekirdeği
$\tilde{k}_e$	e kanalına etki eden yerel kodlama dönüşümü
f_e	e kanalı üzerindeki global kodlama çekirdeği
$\tilde{f}_e$	e kanalı üzerindeki global kodlama dönüşümü
$\overline{F}$	F sonlu cisminin kapanışı
$GF(.)$	Galois cismi
G	Graf
$\mu(.)$	Hayali kanallardaki işlem sayısı
$\zeta_{x(.)}$	Hayali kanallardaki işlem kümesinin eleman sayısı
$X(.)$	Hayali kanallardaki işlemlerin kümesi
w	Hayali kanalların sayısı
ρ	Hata şekli

ρ^*	Hata şekli ile ilişkilendirilen ayırtların kümesi
z	Hata vektörü
Σ	Hata vektörlerinin kümesi
e	Kanal
d	Minimum uzaklık
n_s	S Kaynak düğümünün çıktı ayırtlarının sayısı
Z^*	Sıfırdan farklı tamsayıların kümesi
F	Sistem dönüşüm matrisi
K	Sistem için bitişiklik matrisi
M	Sistem transfer matrisi
F	Sonlu cisim
Ψ	Özel tanımlı fonksiyon
V_T	Özel tanımlı küme
A	Özel tanımlı matris
$\aleph$	Yönlendirilmiş çizgi grafi
ε	Yönlendirilmiş çizgi grafin ayırtlarının kümesi
ν	Yönlendirilmiş çizgi grafin düğümlerinin kümesi
F^w	w Boyutlu sonlu cisim
ρ_z	z Hata vektörünün sıfırdan farklı bileşen yerlerinin sayısı

KISALTMA LİSTESİ

ARPANET	Advanced Research Projects Agency Network
<i>MWD</i>	Minimum weight decoder

ŞEKİL LİSTESİ

	Sayfa
Şekil 2. 1 Graf	3
Şekil 2. 2 Basit graf	4
Şekil 2. 3 Devirli graf.....	4
Şekil 2. 4 (a) bağlı graf, (b) bağlantısız graf	5
Şekil 2. 5 (a) Graf, (b) Alt graf	5
Şekil 2. 6 Yol.....	6
Şekil 2. 7 Kapalı yol	6
Şekil 2. 8 Yönlendirilmiş graf	8
Şekil 2. 9 Yönlendirilmiş izomorf graflar	9
Şekil 2. 10 Graftaki köprü	9
Şekil 2. 11 e, u' dan v' ye bitişik	10
Şekil 2. 12 Bitişiklik matrisi verilen graf.....	10
Şekil 2. 13 Çakışım matrisi verilen graf	11
Şekil 2. 14 Mincut' ı hesaplanan graf	12
Şekil 2. 15 Devirsiz ağ bağlantısı.....	25
Şekil 2. 16 Ağ bağlantısında xOR kodlaması.....	25
Şekil 3. 1 Kelebek ağ bağlantısı	28
Şekil 3. 2 Yerel kodlama çekirdekleri verilen ağ bağlantısı	30
Şekil 3. 3 Ağ kodlaması	31
Şekil 3. 4 Devirli ağ bağlantısında kodlama	37
Şekil 3. 5 Çözümsüz devirli ağ bağlantısı	38
Şekil 3. 6 Cebirsel ağ kodlaması	42
Şekil 3. 7 Girdi ve çıktıları verilen ağ bağlantısı	43
Şekil 3. 8 İşaretlenmiş çizgi grafi	46
Şekil 3. 9 Graf ve grafa ait işaretlenmiş çizgi grafi.....	47
Şekil 3. 10 Ağ bağlantılarında n uzunluğundaki yolların sayısı.....	49
Şekil 3.11 Dönüşüm matrisi verilen ağ bağlantısı	51
Şekil 3. 12 İki kaynaklı iki alıcı düğümlü ağ bağlantısı, uygun işaretlenmiş çizgi graf	53
Şekil 3. 13 Çözüme sahip ağ bağlantısı.....	56
Şekil 4. 1 Hatalı ağ bağlantısında kodlama.....	60
Şekil 4. 2 c ağırlıklı hata yuvarları.....	62

Şekil 4. 3 $[7,4,3]_2$ Hamming kod için ağ bağlantısı	63
Şekil 4. 4 Ayrık yuvarlar	65
Şekil 4. 5 Lineer olmayan ağ bağlantısı	68

ÇİZELGE LİSTESİ

	Sayfa
Tablo 4. 1 Yerel kodlama dönüşümleri.....	68

AĞ CEBİRSEL KODLARI VE UYGULAMALARI

Ayşegül Bayram

Matematik Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Prof. Dr. İrfan ŞİAP

Bilgi çağında yaşadığımız bu günlerde bilginin depolanması ve transferi sırasında meydana gelebilecek bilgi kayıplarını korumak ya da yanlış iletilmesini düzeltmek için kodlama kullanılmaktadır. Bilgi paylaşımının yapıldığı araçların başında bilgisayarlar ve donatıları gelmektedir. Bilgisayar aracılığı ile bilgi paylaşımı yapabilmek için bilgisayar ağına sahip olmak gerekir. Bilginin paylaşımı kurulan ağlarda kodlama ile gerçekleştirilir.

Bu tezde öncelikli olarak ağ bağlantısı ve ağ bağlantı kodlamasının tanımı ve tarihi hakkında bilgi verilmiştir. Daha sonra ağ bağlantıları, çizge teorisi ve ağ bağlantı kodlaması ile ilgili temel tanım ve teoremler verilmiştir. Ağ bağlantı kodlamasının nasıl gerçekleştiğini anlayabilmek için ağ bağlantı kodlamasının nasıl gerçekleştiği ile ilgili ifadeler ve örneklere yer verilmiştir. Özellikle lineer ağ bağlantı kodları ve dönüşüm matrisleri tanıtılmıştır. Hangi şartlar altında ağ kodlama işleminin gerçekleşeceği ve istenilen bağlantıların sağlanacağı belirtilmiştir. Ayrıca ağ bağlantı kodlarında hata tespit etme ve düzeltme ile ilgili temel tanım ve teoremler ifade edilmiştir. Son olarak sonuç ve öneriler kısmı oluşturulmuştur.

Anahtar Kelimeler: Ağ bağlantısı, ağ bağlantı kodlaması, graf teorisi, lineer kodlar, hata düzeltme, hata belirleme.

NETWORK ALGEBRAIC CODES AND APPLICATIONS

Ayşegül BAYRAM

Department of Mathematics Engineering

MSc. Thesis

Advisor: Prof. Dr. İrfan ŞİAP

Nowadays, it is an important issue to store or transmit information from one point to another without any loss or error. Therefore, some efficient coding algorithms have been used to reduce the error or loss. Primarily, information can be shared by computers and there must be a computer network. In these networks, information is shared by using some different coding algorithms.

In the first part of this thesis, an introduction will be given about the definition and history of the network connection and its coding algorithm which is studied in this work. Then, some basic definitions and theorems which are related to networks, graph theory and network coding are explained. To understand how network coding is implemented, some explanations and examples are used. Especially, linear network codes and transformation matrices are introduced. It is specified under which conditions network coding processing and connections which is determined, is occurred. In addition, there are also some basic definitions and theorems about error detection and correction in network connection coding algorithms.

Consequently, there are some results and suggestions about network coding.

Key words: Network, network coding, graph theory, linear codes, error correction, error detection.

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE

1.1 Literatür Özeti

1969 yılında Amerika’ da ARPANET adında ilk bilgisayar ağı hazırlanmıştır. Ağ bağlantılarında veriler, sayısal verilerin kodlanmış bilgi veya verinin sistemler arasında aktarılmasını kapsar. Veri haberleşme yolu üzerinden yönlendirilmeli cihazlar arasında elektrik işaretleri ya da bit katarları ile aktarılır. Verinin bu sayısal tanımlamaları sırasında pek çok protokol imzalanmıştır.

Ağ bağlantı kodlaması ilk olarak 1999’ da R.W Yueng ve N.Cai tarafından “Distributed Source Coding For Satellite Communication” [1] adlı makalede tanıtılmış daha sonra da 2000 de Ahlsvedw. Cai, Li ve Yueng tarafından “ Network Information Flow” [2] de detaylı olarak tanımlanmıştır. Ağ bağlantı kodlamasındaki pek çok çalışma, ağ bağlantısındaki özel durumlar için gerçekleştirilmiştir: Rastgele lineer ağ bağlantı kodları; T. Ho, R. Koetter, M. Medard, D. R. Karger ve M. Effros tarafından “The Benefits Of Coding Over Routing in Randomized Setting” [3] adlı makalede rastgele kodlama metotları ile her düğümde güncellenen kaynak işlemlerinin katsayılarını bir vektör olarak tanımlamışlardır.

1.2 Tezin Amacı

Bu tez kodlama teorisi alanında çalışan veya yeni çalışmaya başlayan herkesin yararlanabilmesi amacıyla hazırlanmıştır. Bu çalışmada graf (çizge) teorisi ve lineer kodlar anlaşılır şekilde anlatılmış ve çok sayıda örnek verilmiş. Ayrıca son zamanlarda

zerinde durulan ađ bađlantı kodlarına dikkat ekmek ve bu konu ile ilgilenenlerin tek kaynakta gerekli tm temel bilgileri bulabilmeleri amalanmıřtır.

1.3 Hipotez

Ađ bađlantı kodları, inřa řekilleri ve ađ bađlantısında bilgi iletiminin nasıl olduđu incelenmiřtir. Ađ bađlantı kodlarına cebirsel bir yaklařımda bulunulmuřtur. Ayrıca ađ bađlantı kodlama probleminin zm iin sonlu cisimler zerinde gerekli řartlar verilmiř ve rnekler ile zme sahip olmayan ađ bađlantıları verilmiřtir.

Ađ bađlantılarında meydana gelebilecek hata ve veri kayıplarına karřı hata belirleme ve hata dzeltme iin gerekli tanım ve teoremler ifade edilmiřtir.

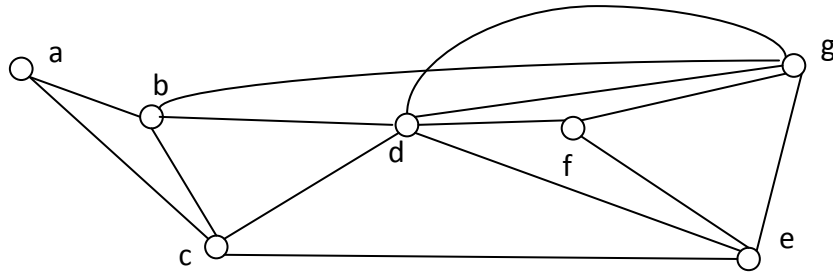
GRAF KAVRAMI ve LİNEER KODLAR

Bu alt bölümde, ağ bağlantılarının geometrik yapısını temsil eden ve önemli rol oynayan çizge (graf) kavramına ilaveten kodlama teorisinin önemli bir ailesi olan lineer kodlar ile ilgili temel tanım ve teoremler verilecektir. Buradaki temel kavramlar ve sonuçlar için ana kaynak olarak Graphs An Introductory Approach, 1990 [4], Coding Theory First Course 2004 [5] ve Network Coding Theory [6] kitabı kullanılmıştır.

2.1 Graf Kavramı (Çizge Teorisi)

Tanım 2.1 Graf; düğüm diye adlandırılan noktaların ve noktalar arasındaki ayırıt diye adlandırılan çizgilerden oluşan koleksiyona denir. Daha belirgin tanımı; V nin elemanları düğüm ve E nin elemanları ayırıt olarak adlandırılmak üzere graf; V düğüm kümesi ve V' nin iki elemanlı alt kümelerinden oluşan E kümesinden oluşur. $G = (V, E)$ ile gösterilir.

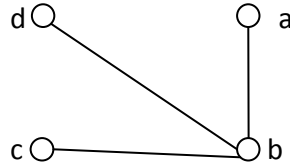
Örnek 2.1



Şekil 2. 1 Graf

Yukarıda şekilde gösterilen düğüm ve ayrıtların kümelerini kontrol edersek $V = \{a, b, c, d, e, f, g\}$ ve $E = \{ab, ac, bc, bd, bg, cd, ce, de, df, dg, ef, eg, fg\}$ olur.

Örnek 2.2



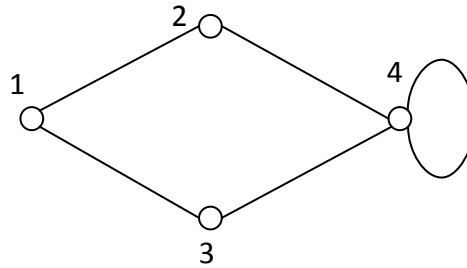
Şekil 2. 2 Basit graf

Yukarıdaki şekilde düğüm ve ayrıtların kümesini kontrol edersek;

$V = \{a, b, c, d\}$ ve $E = \{ab, bc, bd\}$ olduğu görülür.

Tanım 2.2 $G = (V, E)$ grafında bir düğümden kendisine bir ayrıt çiziliyor ise buna döngü denir. Döngü içermeyen graflara basit graf denir.

Örnek 2.3

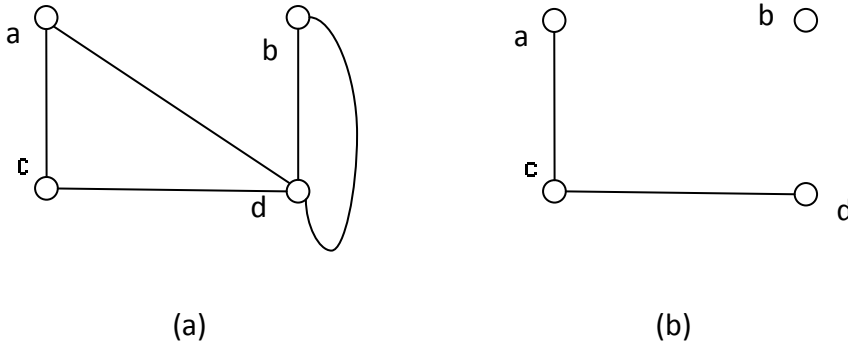


Şekil 2. 3 Devirli graf

Yukarıdaki şekildeki grafta düğümlerin kümesi $V = \{1, 2, 3, 4\}$ ve ayrıtların kümesi $E = \{12, 13, 24, 34, 44\}$ dir. $44 \in E$ olduğu için bu graf bir döngü içerir.

Tanım 2.3 Bir graftaki tüm düğümler en az bir ayrıt ile birbirine bağlı ise bu grafa bağlı graf, en az bir düğüm diğerleri ile bağlı değil ise bu grafa bağlantısız graf denir.

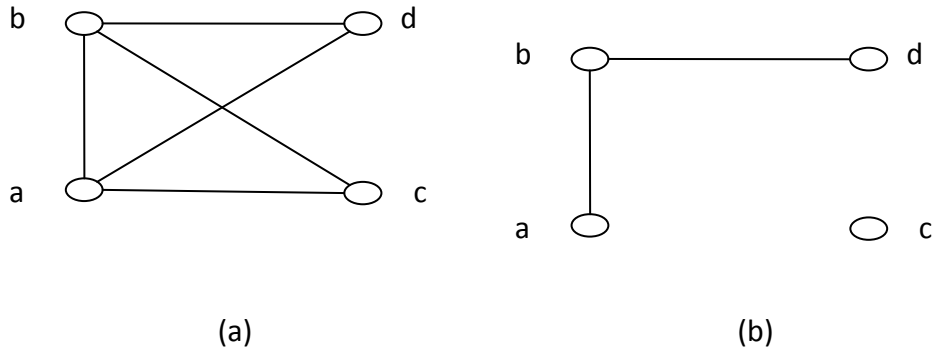
Örnek 2.4



Şekil 2. 4 (a) bağlı graf, (b) bağlantısız graf

Tanım 2.4 $G = (V, E)$ bir graf olmak üzere tüm düğümleri V' ye ve tüm ayrıtları E' ye ait olan graflara G' nin alt grafı denir.

Örnek 2.5



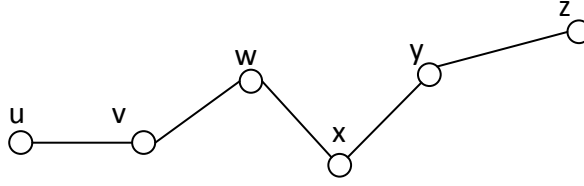
Şekil 2. 5 (a) Graf, (b) Alt graf

(a) da $G = (V, E)$ grafın da düğüm kümesi $V = \{a, b, c, d\}$ ve ayrıt kümesi $E = \{ab, ac, ad, bc, bd\}$ olmak üzere (b) de $G' = (V', E')$ alt grafının düğüm kümesi $V' = \{a, b, c, d\}$ ve ayrıt kümesi $E' = \{ab, bd\}$ dir.

Tanım 2.5 G döngüsüz bir graf ve v , G de bir düğüm olsun. v ile ilişkilendirilen ayrıtların sayısına v ' nin derecesi denir ve $\deg(v)$ ile gösterilir.

Tanım 2.6 G grafında k uzunluğundaki bir yürüyüş diye k tane ayrıtın bir dizisine denir. G grafındaki bir yürüyüş $uvw...yzu$ formundaysa yani başlangıç ve bitiş düğümü aynı ise bu yürüyüşe kapalı yürüyüş denir.

Örnek 2.6

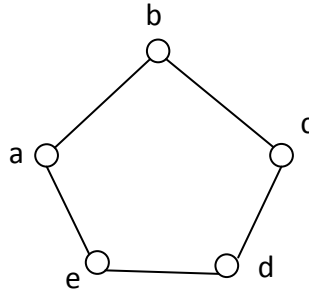


Şekil 2. 6 Yol

uv, vw, wx, xy, yz formundaki ayrıtlar için u' dan v' ye bir yol $uvwxyz$ biçiminde gösterilir.

Tanım 2.7 Bir yürüyüşteki tüm ayrıtlar (düğümlerin olması gerekli değil) farklı ise bu yürüyüşe iz denir. Ayrıca tüm düğümler de farklı ise iz' e yol denir. Eğer kapalı bir yürüyüşteki ayrıtlar farklı ise kapalı bir iz, düğümlerde farklı ise bir devir olur.

Örnek 2.7



Şekil 2. 7 Kapalı yol

Tanım 2.8 $G_1 = (V_1, E_1)$ grafindan $G_2 = (V_2, E_2)$ grafinı izomorfizma V_1' den V_2' ye bire bir f fonksiyondur öyle ki E_1 deki her bir $\{a, b\}$ ayrıtı için $\{f(a), f(b)\}$ de E_2' nin bir ayrıtıdır. İzomorfik grafların ayrıt ve düğüm sayıları eşittir. Ayrıca birbirine karşılık gelen düğümlerin dereceleri eşittir.

Teorem 2. 1 İzomorfizma bağıntısı bir denklik bağıntısıdır.

Tanım 2.9 Bir graf kenarları birbirini kesmeyecek şekilde çizilebiliyorsa bu grafa düzlemsel graf denir.

Tanım 2.10 Devir içermeyen bağlantılı graflara ağaç denir.

Teorem 2.2 n düğümlü bir ağacın $n-1$ ayrıtı vardır.

İspat: İspatı n üzerinden tümevarım ile yapalım. $n=1$ için ağacın herhangi bir ayrıtı yoktur. $n=2$ için ağacın bir ayrıtı vardır. $n=k$ için hipotez doğru olsun. $n=k+1$ için G de x ve y düğümler ve $\{x, y\}$ ayrıtı olsun. Eğer $\{x, y\}$ ayrıtını silersek G grafi iki farklı bağlı G_1 ve G_2 grafına ayırırız. Böylece G_1 ve G_2 grafları çevirim içermeyen bağlı iki ağaç olur ve sırası ile düğüm sayıları k_1 ve k_2 , k dan küçüktür. Kabulden G_1' in k_1-1 ve G_2' nin k_2-1 ayrıtı vardır. Sonuç olarak G' nin ayrıt sayısı $k_1-1+k_2-1+1=k_1+k_2-1=k+1-1=k$ olur.

Tanım 2.11 G bir düzlemsel graf olsun. G' nin ayrıtları ile sınırlandırılmış bölgelere yüz denir. Her yüz en az 3 ayrıta sahiptir.

Teorem 2.3 (Euler Formülü) Düzleme çizilmiş bir G grafi için; v düğümlerin sayısı, e ayrıtların sayısı ve f' de yüzlerin sayısı olmak üzere düzlemsel graf için

$$v-e+f=2 \text{ dir.}$$

İspat: İspatı e için tümevarımla yapalım. Kabul edelim ki $e=0$ olsun. Böylece sadece bir tane düğüm ve sadece bir tane yüz vardır. Buradan $v-e+f=1-0+1=2$ elde edilir. $e=1$ ise iki olasılık vardır ya $v=1, e=1, f=2$ ya da $v=2, e=1, f=1$ dir. Her iki durumda da $v-e+f=2$ dir. Şimdi teoremin n ayrıttan daha az graflar için doğru olduğunu kabul edelim. Eğer G bir ağaç ise düğüm sayısı $v=e+1$ ve $f=1$ dir. O halde $v-e+f=e+1-e+1=2$ elde edilir. Eğer G bir ağaç değil ise G de en azından bir devir vardır. G' de ki herhangi bir devri seçelim ve bir ayrıtını silelim. Oluşan yeni grafin $n-1$ tane ayrıtı vardır. Kabulden teorem sağlanır. Ayrıca $f-1$ yüzü vardır. Böylece $v-(e-1)+(f+1)=v-e+1=2$ dir.

Teorem 2.4 e ayrıtların ve v düğümlerin sayısını göstermek üzere 3 veya daha fazla düğümü olan bağlı düzlemsel graf için

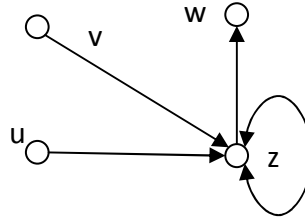
$$e \leq 3v-6 \text{ dir.}$$

İspat: Kabul edelim ki G en fazla e ayrıta sahip ve f tane yüzü olsun. f yüzünü sınırlayan ayrıtlar e olmak üzere (e, f) ikililerini düşünelim. Her bir e ayrıtı için e' nin sınırladığı yüz vardır ve bu yüz ayrıt ikililerinin toplam sayısı $2e'$ den daha azdır. Öte

yandan G basit graf olduğu için her yüz en fazla 3 ayrıt ile sınırlandırılır. Böylece yüz ayrıt çiftlerinin toplam sayısı en az $3f'$ ye eşit ya da $3f'$ den fazladır. Dolayısıyla $3f \leq 2e$ olur. Öte yandan $f = 2 + e - v$ dir. Buradan $3f \leq 2e$, $6 + 3e - 3v \leq 2e$ ve $e \leq 3v - 6$ olur.

Tanım 2.12 $G = (V, E)$ grafı için eğer V' nin düğümleri u ve v ise uv ye u' dan v' ye yönlendirilmiş ayrıt denir. $G = (V, E)$ grafı düğümler ve yönlendirilmiş ayrıtlardan oluşuyor ise yönlü graf denir.

Örnek 2.8

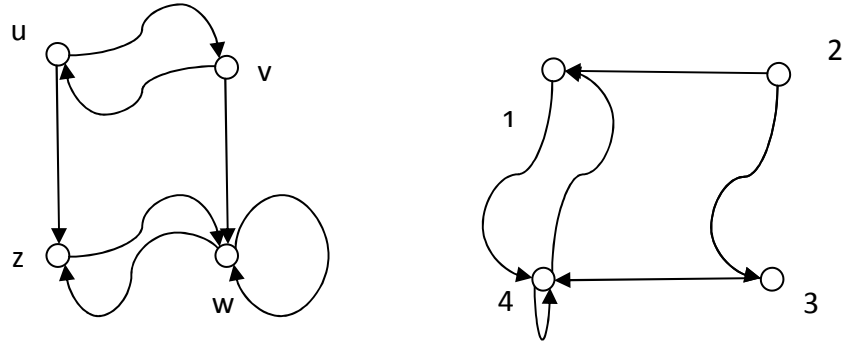


Şekil 2. 8 Yönlendirilmiş graf

$G = (V, E)$ olmak üzere $V = \{u, v, w, z\}$ ve $E = \{uz, vz, wz, zz\}$ dir.

Tanım 2.13 G_1 ve G_2 iki farklı yönlendirilmiş graf olsun. Eğer G_1' in ayrıtları ve G_2' nin ayrıtları arasında birebir eşleme var yani; herhangi bir düğüme giren ayrıtlar ile diğer graftaki uygun düğümlere giren ayrıtlar arasında birebir bir eşleme var ise G_1 ve G_2' ye izomorf yönlendirilmiş graflar denir.

Örnek 2.9

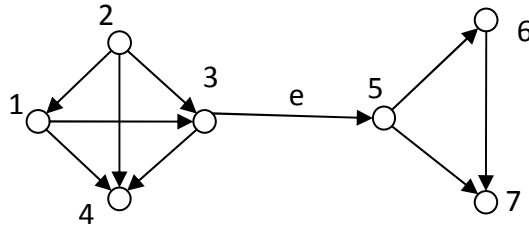


Şekil 2. 9 Yönlendirilmiş izomorf graflar

$$\begin{array}{c} G_1 \\ \updownarrow \\ G_2 \end{array} : \begin{array}{c} u \\ \updownarrow \\ 2 \end{array} \begin{array}{c} v \\ \updownarrow \\ 3 \end{array} \begin{array}{c} w \\ \updownarrow \\ 4 \end{array} \begin{array}{c} z \\ \updownarrow \\ 1 \end{array} \quad \text{ve} \quad \begin{array}{c} G_1 \\ \updownarrow \\ G_2 \end{array} : \begin{array}{c} uv \\ \updownarrow \\ 23 \end{array} \begin{array}{c} wz \\ \updownarrow \\ 41 \end{array} \dots \begin{array}{c} ww \\ \updownarrow \\ 44 \end{array} \quad \text{dir.}$$

Tanım 2.14 G grafi bağlı bir graf olsun. Eğer bir e ayrıtının G grafindan atılması ile graf bağlantısız oluyorsa e ayrıtına köprü denir.

Örnek 2.10

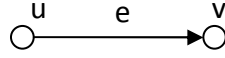


Şekil 2. 10 Graftaki köprü

$G = (V, E)$ grafinda $V = \{1, 2, 3, 4, 5, 6, 7\}$ ve $E = \{13, 14, 21, 23, 24, 34, 35, 56, 57, 67\}$ olmak üzere $e = 35$ ayrıtının silinmesi ile graf iki farklı grafa ayrılır. O halde e ayrıtına köprü denir.

Tanım 2.15 u ve v bir G grafinin düğümleri olsun. Eğer u ve v düğümleri bir e ayrıtı ile birbirlerine bağlanmış ise u ve v ye bitişiktirler denir. Eğer e ayrıtı, u' dan v doğru yönlendirilmişse e' ye u' dan v' ye bitişiktir denir.

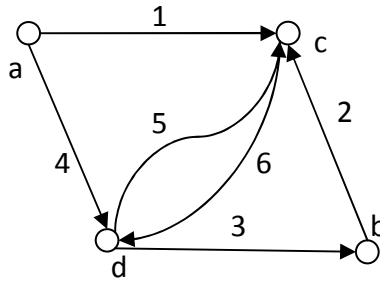
Örnek 2.11



Şekil 2. 11 e , u ' dan v ' ye bitişik

Tanım 2.16 G yönlendirilmiş, çevrimsiz n düğümlü graf ve $1, 2, \dots, n$; G' nin düğümleri olsun. Bir M , $n \times n$ boyutlu matrisinin i . satırı ve j . sütunu i düğümünden j düğümüne olan ayrıtların kümesinin eleman sayısını gösteriyor olsun. M , matrisine G grafının bitişiklik matrisi denir.

Örnek 2.12



Şekil 2. 12 Bitişiklik matrisi verilen graf

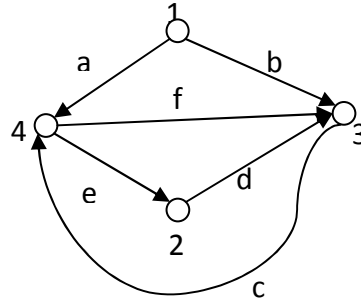
Yukarıdaki grafın bitişiklik matrisi; $M = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$ dir.

Tanım 2.17 G yönlendirilmiş, döngüsüz n düğümlü graf, $1, 2, \dots, n$; G' nin düğümleri ve $a_1, a_2, \dots, a_m$, G' nin ayrıtları olmak üzere; $n \times m$ boyutlu $I(G)$ matrisi şu şekilde tanımlasın.

$$I(i, j) = \begin{cases} 1, & j, i \text{ düğümüne giriyor ise} \\ -1, & j, i \text{ düğümünden çıkıyor ise} \\ 0, & \text{diğer} \end{cases}$$

$I(G)$ matrisine G grafının çakışım matrisi denir.

Örnek 2.13



Şekil 2. 13 Çakışım matrisi verilen graf

G grafi yukarıdaki şekildeki gibi verilsin. G' nin çakışım matrisi I olmak üzere;

$$I = \begin{pmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 1 & -1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & -1 & -1 \end{pmatrix} \text{ dir.}$$

Tanım 2.18 $G = (V, E)$ bir graf olsun. $G = (V, E)$ grafindaki ayrıtlara bir pozitif tam sayı değeri atanarak oluşturulan grafa ağırlıklı graf denir. Ayrıtlara atanan sayılar ayrıtların taşıdıkları yüklerin ağırlıkları olarak düşünülebilir.

Tanım 2.19 $G = (V, E)$ bir graf olsun. S ve R , E ayrıt kümesinin iki alt kümesi olsun. R' den S' i ayıran kesim (cut); R' den S' yi ayıran ayrıtların kümesidir. Başka bir ifade ile; R ve S arasındaki kesim (cut);

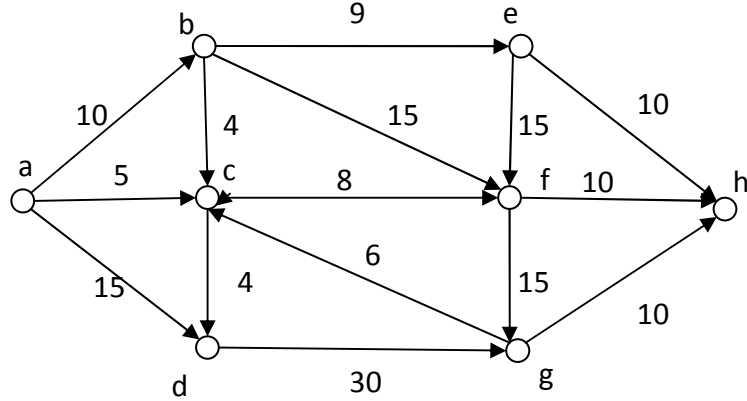
$E_C = \{e \in E : e \in \text{out}(i) \cap \text{int}(j); i \in S, j \notin R\}$ kümesidir. Burada $\text{out}(i)$ ile i düğümünden çıkan ayrıtların kümesi ve $\text{int}(j)$ ile j düğümüne giren ayrıtların kümesi gösterilir.

Tanım 2.20 $G = (V, E)$ bir ağırlıklı graf olsun. S ve R , E ayrıt kümesinin iki alt kümesi olsun. R' den S' yi ayıran kesimin ağırlığı kesimdeki ayrıtların ağırlıkları toplamına eşittir.

Bir $G = (V, E)$ grafi için kesim tek değildir pek çok kesim olabilir.

Tanım 2.21 $G = (V, E)$ bir ağırlıklı graf olsun. S ve R , E ayrıt kümesinin iki alt kümesi olsun. R' den S' yi ayıran kesimin minimum ağırlığına grafın $\min cut'$ ı denir ve $\min cut(G)$ ile gösterilir.

Örnek 2.14



Şekil 2. 14 Mincut' ı hesaplanan graf

Yukarıdaki graf için a ve h düğümlerini ayıran kesimler içinde minimum ağırlık 30 dur. Dolayısıyla $\min cut(G) = 30$ olur.

2.2 Lineer Kodlar

Tanım 2.22 $F_q = \{0, 1, \dots, q-1\}$, q elemanlı sonlu bir cisim olsun. F_q üzerinde uzunluğu n , eleman sayısı M olan bir C kodu, F_q^n uzayının bir alt kümesidir. Kodun herhangi bir elemanına kod söz denir. Böyle bir kod $(n, M)_q$ – kod şeklinde gösterilir.

Tanım 2.23 (Lineer Kod) F_q üzerinde uzunluğu n , boyutu k olan bir C lineer kodu, F_q^n uzayının bir alt uzayıdır. Böyle bir kod $[n, k]_q$ – kod şeklinde gösterilir. Özel olarak, $q = 2$ için, F_2 üzerinde bir koda ikili kod, $q = 3$ için, F_3 üzerinde bir koda üçlü kod denir.

Örnek 2.15 $q = 2$ için $C = \{0000, 0001, 0010, 0011\} \subseteq F_2^4$ kodu bir ikili lineer koddur.

Örnek 2.16 $q = 3$ için $C = \{000, 110, 011, 220, 022, 201, 212, 121, 102, \dots\} \subseteq F_3^3$ kodu bir üçlü lineer koddur.

Tanım 2.24 C , F_q üzerinde bir lineer kod olsun.

- C' nin dual kodu $C^\perp$; $C^\perp = \{x \in F_q^n : \langle x, c \rangle = 0, \forall c \in C\}$ şeklinde tanımlanır.
- C kodunun boyutu, C' nin vektör uzayı olarak boyutudur, $\text{boy}(C)$ ile gösterilir.

Teorem 2.5 C , F_q üzerinde n uzunluğunda boyutu k olan bir lineer kod olsun.

$$(i) |C| = q^{\text{boy}(C)} \quad (M = q^k),$$

$$(ii) C^\perp \text{ bir lineer koddur ve } \text{boy}(C) + \text{boy}(C^\perp) = n,$$

$$(iii) (C^\perp)^\perp = C.$$

İspat: (i) C kodunun, vektör uzayı olarak bir bazı $\{c_1, c_2, \dots, c_k\}$ olsun. O halde, $C = \{\lambda_1 c_1 + \lambda_2 c_2 + \dots + \lambda_k c_k : \lambda_1, \lambda_2, \dots, \lambda_k \in F_q\}$ yazılır. $|F_q| = q$ olduğundan her bir λ_i için q seçenek vardır. O halde C' nin q^k elemanı vardır.

$$(ii) C^\perp = \{x \in F_q^n : \langle x, c \rangle = 0, \forall c \in C\}, F_q^n \text{ vektör uzayının bir alt uzayıdır:}$$

$$1) x, y \in C^\perp \text{ olsun. Yani, } \langle x, c \rangle = 0, \forall c \in C \text{ ve } \langle y, c \rangle = 0, \forall c \in C.$$

$$\forall c \in C; \langle x + y, c \rangle = \langle x, c \rangle + \langle y, c \rangle = 0, \forall c \in C \Rightarrow x + y \in C^\perp.$$

$$2) x \in C^\perp \text{ olsun. Yani, } \langle x, c \rangle = 0, \forall c \in C.$$

$$\forall c \in C, \alpha \in F_q; \langle \alpha x, c \rangle = \alpha \langle x, c \rangle = 0 \Rightarrow \alpha x \in C^\perp.$$

1) ve 2) den $C^\perp$, F_q^n vektör uzayının bir alt uzayıdır, yani bir lineer koddur.

$\text{boy}(C) + \text{boy}(C^\perp) = n$; $C = \{0\}$ ise F_q^n deki tüm vektörler C' ye diktir. Dolayısıyla

$F_q^n = C^\perp$. $\text{boy}(C) = k \geq 1$ olsun. C' nin bir bazı $\{c_1, c_2, \dots, c_k\}$ olsun.

$x \in C^\perp \Leftrightarrow \langle c_1, x \rangle = \langle c_2, x \rangle = \dots = \langle c_k, x \rangle = 0$. Yani, $\{c_1, c_2, \dots, c_k\}$ bazının elemanlarını

satır kabul eden matris A olsun. $A = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{pmatrix} \Rightarrow Ax^T = 0$. Burada k lineer bağımsız

denklem ve n bilinmeyen var. Böyle bir sistemin $n - k$ tane çözümü vardır. Dolayısıyla $C^\perp$ uzayının boyutu $n - k$ dir.

(iii) $c \in C$ olsun. Bu durumda $\langle c, d \rangle = c \cdot d = 0, \forall d \in C^\perp \Rightarrow \langle d, c \rangle = d \cdot c = 0 \Rightarrow c \in (C^\perp)^\perp$

Öte yandan C bir $[n, k]$ -kod olduğundan $C^\perp$ $[n, n - k]$ -koddur. $\left| (C^\perp)^\perp \right| = q^k = |C|$.

Buradan, $(C^\perp)^\perp = C$ elde edilir.

Örnek 2.17 $C = \{000000, 000001, 000010, 000011, 000100, 000101, 000110, 000111\}$

kodu için $M = 8 = 2^k \Rightarrow k = 3$ tür.

$C^\perp = \{x \in F_2^6 : \langle x, c \rangle = 0, \forall c \in C\}, x = (x_1, x_2, x_3, x_4, x_5, x_6) \in C^\perp :$

$\langle x, 000000 \rangle = 0, \langle x, 000001 \rangle = 0, \langle x, 000010 \rangle = 0, \langle x, 000011 \rangle = 0, \langle x, 000100 \rangle = 0,$
 $\langle x, 000101 \rangle = 0, \langle x, 000110 \rangle = 0, \langle x, 000111 \rangle = 0.$

$x_6 = 0, x_5 = 0, x_5 + x_6 = 0, x_4 = 0, x_4 + x_6 = 0, x_4 + x_5 = 0, x_4 + x_5 + x_6 = 0.$

$C^\perp = \{(r, s, t, 0, 0, 0) : r, s, t \in F_2\} =$

$C^\perp = \{000000, 100000, 010000, 110000, 001000, 101000, 011000, 111000\}.$

2.2.1 Self Dual ve Self Ortogonal (Kendine Dual ve Kendine Dik) Kodlar

Tanım 2.25 C, F_q üzerinde bir lineer kod olsun. Eğer $C \subseteq C^\perp$ ise C , ye kendine ortogonal (dik) kod, eğer $C = C^\perp$ ise C , ye kendine dual kod denir.

- Kendine dual bir kodun uzunluğu çifttir ve $k = \frac{n}{2}$ dir.

- $C = C^\perp \Rightarrow k = n - k \Rightarrow 2k = n \Rightarrow k = \frac{n}{2}$ olur.

• C, F_q^n üzerinde bir $[n, k]_q$ – kod olsun. C ' nin kendine dual olması için gerek ve yeter şart C ' nin kendine dik ve $k = \frac{n}{2}$ olmasıdır. $C \subseteq C^\perp, \quad k = \frac{n}{2},$
 $|C| = |C^\perp| \Rightarrow C = C^\perp.$

2.2.2 Üreteç ve Kontrol Matrisleri

Tanım 2.26 (Üreteç Matris) Bir $C [n, k]_q$ – kodu, F_q^n uzayının bir alt uzayıdır. Dolayısıyla bir bazı vardır. C ' nin bir bazı $\{c_1, c_2, \dots, c_k\}$ olsun. Bazdaki vektörleri satır kabul eden G matrisine, C kodunun üreteç matrisi denir.

$$G = \begin{pmatrix} -c_1 - \\ -c_2 - \\ \vdots \\ -c_k - \end{pmatrix}_{k \times n}$$

C deki her kodsöz G ' nin satırlarının bir F_q lineer kombinezonu olarak yazılabilir.

Örnek 2.18 $G = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}_{2 \times 4}$ matrisinin F_2 üzerinde ürettiği C kodu $2^2 = 4$ elemanlıdır. C bir $[4, 2]_2$ – koddur.

$$C = \{\alpha c_1 + \beta c_2 : \alpha, \beta \in F_2\} = \{0000, 1001, 0111, 1110\}.$$

Örnek 2.19 $G = \begin{pmatrix} 1 & 0 & 1 & 2 & 1 \\ 1 & 1 & 0 & 2 & 0 \end{pmatrix}_{2 \times 5}$ matrisinin F_3 üzerinde ürettiği C kodu $3^2 = 9$ elemanlıdır. C bir $[5, 2]_3$ – koddur.

$$C = \{\alpha_1 (10121) + \alpha_2 (11020) : \alpha_1, \alpha_2 \in F_3\}.$$

$$C = \{00000, 10121, 01020, 11111, 20212, 02010, 22222, 12101, 21202\} \text{ olur.}$$

• Bir C lineer kodunun kendine ortogonal olduğunu ispat etmek için, C ' nin üreteç matrisindeki satırların ikişerli dik olduğunu göstermek yeterlidir. Gerçekten; C ' den c, d keyfi kodsözlerini alalım. Bu durumda; $c_1, c_2, \dots, c_k, G$ üreteç matrisinin satırları olmak üzere,

$c = \alpha_1 c_1 + \alpha_2 c_2 + \dots + \alpha_k c_k$, $d = \beta_1 c_1 + \beta_2 c_2 + \dots + \beta_k c_k$ şeklinde yazılır. Buradan,

$\langle c, d \rangle = \sum_{i,j} \alpha_i \beta_j \langle c_i, c_j \rangle$. Eğer her bir $\langle c_i, c_j \rangle = 0$ ise o zaman $\langle c, d \rangle = 0$ dır.

Örnek 2.20 Varsa uzunluğu 4 olan kendine dual bir kod inşa ediniz.

Çözüm: $G = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$ matrisi ile elde edilen $C = \{0000, 0110, 1001, 1111\}$

kodu 4 uzunluğunda kendine dual bir koddur.

• Herhangi $n \in N$ için $2n$ uzunluğunda kendine dual bir ikili kod vardır.

$G = (I_n | I_n)$ üreteç matrisine sahip bir kod $[2n, n]_2$ - koddur. $(e_i | e)(e_j | e_j) = 0, i \neq j$,

$i = j \Rightarrow (e_i | e)(e_j | e_j) = e_i \cdot e_j + e_i \cdot e_j = 1 + 1 = 0$.

C aynı zamanda self dualdir.

• Yukarıdaki durum $q \neq 2$ için doğru değildir.

Örnek 2.21 F_3 üzerinde uzunluğu 2 olan kendine dual bir kod var mıdır?

Çözüm: Mümkün olan tüm üreteç matrisleri:

$[10], [11], [12]$ dir. $(1,0) \cdot (1,0) = 1 \neq 0, (1,1) \cdot (1,1) = 2 \neq 0, (1,2) \cdot (1,2) = 2 \neq 0$. Dolayısıyla

F_3 üzerinde kendine dual olan 2 uzunluğunda kod yoktur. Ancak F_3 üzerinde örneğin

$[12]'$ nin ürettiği kod kendine dualdir.

Örnek 2.22 $q = 3$ için $S = \{12101, 20110, 01122, 11010\}$ ve $\langle S \rangle = C$ ise C kodunu yazınız.

Çözüm: $S = \{12101, 20110, 01122, 11010\}'$ nin elemanlarını satır kabul eden G matrisini yazalım.

$$G = \begin{pmatrix} 1 & 2 & 1 & 0 & 1 \\ 2 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 2 & 2 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

$C = \{00000, 12100, 01120, 00001, 21200, 02210, 00002, 10220, 12101, 01121, 20110, 21202, 02212, 10221, 20112, 11010, 12102, 22020, 21201, 01122, 02211, 01121, 20112, 20111, 11012, 22021, 10222, \}$

bir $[5,3]_3$ – koddur.

Tanım 2.27 (Parite Kontrol Matrisi) Bir C lineer kodunun parite kontrol matrisi dualinin üreteç matrisidir.

$$C = \{x \in F_q^n : Hx^T = 0\}.$$

Örnek 2.23 $H = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}_{3 \times 4}$ matrisinin parite kontrol matrisi olduğu ikili C

kodunu bulalım.

$$C = \left\{ (x_1, x_2, x_3, x_4) \in F_2^4 : H \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \right\}.$$

$$x_1 + x_3 + x_4 = 0$$

$$x_2 + x_4 = 0, \quad x_1 = 0, \quad x_2 = x_3 = r, \quad x_4 = -r \Rightarrow C = \{(0rrr) : r \in F_2\} = \{0000, 1111\}.$$

$$x_3 + x_4 = 0$$

Tanım 2.28 C' nin üreteç matrisi $G = [I_k \mid A]_{k \times n}$ olan bir $[n, k]_q$ – kod olsun. Bu durumda G nin standart formda olduğu söylenir. Burada, I_k , $k \times k$ boyutunda birim matris; A ise $k \times (n - k)$ boyutlu bir matristir.

Teorem 2.6 Bir C $[n, k]_q$ – kodunun üreteç matrisi standart formda verilmişse

$$H = [-A^T \mid I_k]$$
 matrisi, C için bir kontrol matrisidir.

Örnek 2.24 Bir C $[4, 2]_3$ – kodu

$$G = \begin{bmatrix} 1 & 0 & 2 & 2 \\ 0 & 1 & 2 & 1 \end{bmatrix}$$
 üreteç matrisi ile verilsin.

Bu durumda C kodunun farklı bir üreteç matrisini H kontrol matrisi ile belirleyiniz.

Çözüm:

$$G = \begin{pmatrix} 1 & 0 & 2 & 2 \\ 0 & 1 & 2 & 1 \end{pmatrix} \rightarrow A^t = \begin{pmatrix} 2 & 2 \\ 2 & 1 \end{pmatrix} \rightarrow -A^t = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \rightarrow H = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{pmatrix} \text{ dır. Buradan}$$

$$C = \{(x_1, x_2, x_3, x_4) \in F_3^4 : Hx^T = 0\} \Rightarrow \begin{cases} x_1 + x_2 + x_3 = 0 \\ x_1 + 2x_2 + x_4 = 0 \end{cases}$$

$$\Rightarrow x_3 = r, x_4 = s, x_1 = r + s, x_2 = r - s.$$

$$\Rightarrow C = \{(r + s, r + 2s, r, s) : r, s \in F_3\}. G' = \begin{pmatrix} 1 & 2 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \text{ matrisi } C \text{ için başka bir üreteç}$$

matristir.

Örnek 2.25 $G = \begin{pmatrix} 2 & 1 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 2 & 1 & 2 \\ 2 & 0 & 0 & 2 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 2 \end{pmatrix}$ matrisini standart hale getirip üçlü bir C kodu

için kontrol matrisi bulalım.

Çözüm:

$$G = \begin{pmatrix} 2 & 1 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 2 & 1 & 2 \\ 2 & 0 & 0 & 2 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 2 \end{pmatrix} \xrightarrow{H_{12}(1), H_{32}(1)} \begin{pmatrix} 1 & 1 & 1 & 2 & 1 & 2 \\ 0 & 0 & 1 & 2 & 1 & 2 \\ 2 & 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 1 & 2 & 1 & 2 \end{pmatrix} \xrightarrow{H_{12}, H_1\left(\frac{1}{2}\right)} \begin{pmatrix} 1 & 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 1 & 2 & 1 & 2 \\ 1 & 1 & 1 & 2 & 1 & 2 \\ 0 & 0 & 1 & 2 & 1 & 2 \end{pmatrix} \xrightarrow{H_{21}(2), H_{23}(2)} \begin{pmatrix} 1 & 0 & 0 & 1 & 2 & 0 \\ 1 & 1 & 1 & 2 & 1 & 2 \\ 0 & 0 & 1 & 2 & 1 & 2 \end{pmatrix}$$

$$\xrightarrow{H_{12}(1), H_{32}(1)} \begin{pmatrix} 1 & 0 & 0 & 1 & 2 & 0 \\ 0 & 1 & 0 & 2 & 1 & 0 \\ 0 & 0 & 1 & 2 & 1 & 2 \end{pmatrix} \Rightarrow A = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 2 & 1 & 2 \end{pmatrix} \Rightarrow -A^T = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 2 \\ 0 & 0 & 2 \end{pmatrix} \Rightarrow H = \begin{pmatrix} 2 & 1 & 1 & 1 & 0 & 0 \\ 1 & 2 & 2 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & 0 & 1 \end{pmatrix}$$

Örnek 2.26 $H = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}_{3 \times 7}$ kontrol matrisi ile verilen ikili kodu bulunuz.

Çözüm: C bir $[7, 4]_2$ -koddur. $C = \{x \in F_2^7 : Hx^T = 0\}, x = (x_1, x_2, x_3, x_4, x_5, x_6, x_7) \in C$

$$\begin{cases} x_1 + x_4 + x_5 + x_7 = 0 \\ x_2 + x_4 + x_6 + x_7 = 0 \\ x_3 + x_5 + x_6 + x_7 = 0 \end{cases} \Rightarrow \begin{cases} x_4 = a, x_5 = b, x_6 = s, x_7 = r, x_1 = a + b + r, x_2 = a + s + r, x_3 = r + s + b \end{cases}$$

$$C = \{(a + b + r, a + s + r, r + s + b, a, b, s, r) : a, b, s, r \in F_2\}.$$

$$a(1101000) + b(1010100) + s(0110010) + r(1110001)$$

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \text{ matrisi } C \text{ için bir üreteç matrisidir. } C \text{ ikili Hamming}$$

koddur: $[7,4,3]_2$.

Tanım 2.29 (Tekrarlı Kod) n uzunluğunda tekrarlı ikili kod bir $[n,1]_2$ -koddur:

$$\{00\dots 0, 11\dots 1\}.$$

$$G = \begin{pmatrix} 1 & \underbrace{1\dots 1}_A \\ \vdots & \vdots \end{pmatrix}_{1 \times n} \Rightarrow H = \begin{pmatrix} 1 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & 0 & 0 & \dots & 1 \end{pmatrix} \Rightarrow C = \{(x_1, x_2, \dots, x_n) : Hx^T = 0\} \Rightarrow$$

$$\left. \begin{array}{l} x_1 + x_2 = 0 \\ x_1 + x_3 = 0 \\ \vdots \\ x_1 + x_n = 0 \end{array} \right\} C = \{(rr\dots r) : r \in F_2\}$$

2.2.3 Hamming Uzaklık- Hamming Ağırlık

Tanım 2.30 (Hamming Uzaklık) $x = (x_1, x_2, \dots, x_n), y = (y_1, y_2, \dots, y_n) \in F_q^n$ için x ile y arasındaki Hamming uzaklık $d_H(x, y) = d_H(x_1, y_1) + d_H(x_2, y_2) + \dots + d_H(x_n, y_n)$ dir. Burada,

$$d_H(x_i, y_i) = \begin{cases} 1, & x_i \neq y_i \\ 0, & x_i = y_i \end{cases} \text{ biçiminde tanımlanır. Yani; } d_H(x, y) = |\{i : x_i \neq y_i\}| \text{ dir.}$$

Örnek 2.27 F_2 üzerinde $x = (10011111), y = (10110001) \Rightarrow d_H(x, y) = 4$.

F_3 üzerinde $x = (00222101), y = (11212210) \Rightarrow d_H(x, y) = 6$.

Teorem 2.7 d_H, F_q^n üzerinde bir metriktir. Yani $\forall x, y, z \in F_q^n$:

$$(i) \quad d_H(x, x) = 0,$$

$$(ii) \quad d_H(x, y) \geq 0 \text{ ve } d_H(x, y) = 0 \Leftrightarrow x = y,$$

$$(iii) \quad d_H(x, y) = d_H(y, x),$$

$$(iv) \quad d_H(x, z) \leq d_H(x, y) + d_H(y, z).$$

Tanım 2.31 F_q üzerinde n uzunluğunda bir C kodunun minimum uzaklığı,

$d_{\min}(C) = \min\{d_H(x, y) : x \neq y, x, y \in C\}$ şeklinde tanımlanır. Minimum uzaklığı d olan bir $[n, k]_q$ – kod, $[n, k, d]_q$ – kod ile gösterilir.

Tanım 2.32 (Hamming Ağırlığı) Bir $x \in F_q^n$ vektörünün Hamming ağırlığı, $w_H(x)$, x ' in sıfırdan farklı koordinatlarının sayısıdır. Yani, $w_H(x) = d_H(x, 0)$ dir.

Örnek 2.28 F_2 üzerinde $x = (10011101) \Rightarrow w_H(x) = 5$.

F_3 üzerinde $x = (10221101) \Rightarrow w_H(x) = 6$.

Teorem 2.8 $x, y \in F_q^n$ için $d_H(x, y) = w_H(x - y)$ dir.

İspat: $x, y \in F_q^n$ olsun. $d_H(x, y) = d_H(x_1, y_1) + d_H(x_2, y_2) + \dots + d_H(x_n, y_n)$
 $= d_H(x_1 - y_1, 0) + d_H(x_2 - y_2, 0) + \dots + d_H(x_n - y_n, 0)$
 $= w_H(x_1 - y_1) + w_H(x_2 - y_2) + \dots + w_H(x_n - y_n)$
 $= w_H(x - y)$

Teorem 2.9 Eğer C , F_q üzerinde bir lineer kodsaa $d_{\min}(C) = w_{\min}(C)$.

İspat:

$$\begin{aligned} d_{\min}(C) &= \min\{d_H(x, y) : x, y \in C, x \neq y\} \\ &= \min\{w_H(x - y) : x, y \in C, x \neq y\} \\ &= \min\{w_H(c) : c \in C, c \neq 0\} \\ &= w_{\min}(C). \end{aligned}$$

Teorem 2.10 $x, y \in F_2^n$ olsun. Bu durumda,

$$w_H(x + y) = w_H(x) + w_H(y) - 2w_H(x * y),$$

$x * y$: bileşensel (componentwise) çarpım.

Örnek 2.29 $x = (11101011), y = (10100010) \Rightarrow x * y = (10100010)$

$$x * x = x, \quad x * y = y * x, \quad x * 1 = x, \quad x * 0 = 0.$$

Teorem 2.11 Eğer C kendine dik ikili bir kodsı C deki her kodsöz çift ağırlıklıdır ve $(11\dots 1) \in C^\perp$ in içindedir.

İspat: $w_H(x) = \langle x, x \rangle = x \cdot x \pmod{2}$. C kendine dik olduğundan,

$w_H(x) = \langle x, x \rangle = x \cdot x = 0 \pmod{2}$. Böylelikle her kodsöz çift ağırlıklıdır. Her kodsöz çift ağırlıklı olduğundan, $\forall c \in C, \bar{1} \cdot c = w_H(c) \equiv 0 \pmod{2}$. Dolayısıyla $\bar{1} \in C^\perp$.

Not: Teoremin tersi her zaman doğru değildir.

Örnek 2.30 C , aşağıdaki üreteç matrisi ile üretilen bir kod olsun. C deki tüm kodsözler çift ağırlıklıdır: $G = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$, $C = \{110, 011, 101, 000\}$. Fakat C kendine dik değildir. $(110)(011) = 1 \neq 0$.

Not: Eğer G 'nin satırları çift ağırlıklı ise C 'nin tüm elemanları çift ağırlıklıdır.

$$\text{İspat: } G = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{pmatrix}, \quad C = \{c_1, c_2, \dots, c_k, c_1 + c_2, \dots, c_1 + c_2 + \dots + c_k\}$$

$$\begin{aligned} w_H(c_i + c_j) &= w_H(c_i) + w_H(c_j) - 2w_H(c_i * c_j) \\ &= w_H(c_i) + w_H(c_j) \pmod{2} \\ &= 0 \pmod{2}. \end{aligned}$$

Teorem 2.12 Eğer C kendine dik üçlü bir kodsı, her bir kod sözünün ağırlığı 3 ile bölünür.

Teorem 2.13 Eğer bir C üçlü kodundaki kodsözlerin ağırlığı 3 ile bölünebiliyorsa, C kendine diktir.

Teorem 2.14 C bir ikili lineer kod olsun.

- i. C kendine dik ve C nin üreteç matrisinin satırlarının ağırlıkları 4 ile bölünebiliyorsa C deki tüm kod sözlerin ağırlığı 4 ile bölünür.
- ii. Eğer C deki her kodsözün ağırlığı 4 ile bölünebiliyorsa, C kendine diktir.

İspat:

$$i. \quad G = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{pmatrix} \quad w_H(x_i) \equiv 0 \pmod{4}, \quad x_i \cdot x_j = 0 \quad (F_2 \text{ de})$$

Tümevarımdan, $w_H(x_i + x_j)$ 'yi incelemek yeterlidir:

$$w_H(x_i + x_j) = w_H(x_i) + w_H(x_j) - 2w_H(x_i * x_j) \equiv 0 \pmod{4}$$

$$ii. \quad x, y \in C;$$

$$0 = w_H(x + y) = w_H(x) + w_H(y) - 2w_H(x * y) \Rightarrow 2w_H(x * y) \equiv 0 \pmod{4}$$

$$\Rightarrow w_H(x * y) \equiv 0 \pmod{2}$$

$$\Rightarrow x \cdot y \equiv 0 \pmod{2}$$

Böylece C kendine diktir.

Teorem 2.15 C kodunun minimum uzaklığının d olması için gerek ve yeter koşul kontrol matrisinin d lineer bağımlı sütunu olması fakat $d-1$ lineer bağımlı sütununun olmamasıdır.

Örnek 2.31 C bir $[5,2]_2$ -kodunun kontrol matrisi $H = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}$ olsun.

$H_1 + H_3 = 0$, yani 1. ve 3. sütunlar lineer bağımlıdır, ancak herhangi 1 sütun aldığımızda lineer bağımsız çıkar. O halde, $d(C) = 2$ dir.

Örnek 2.32 $H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$ olsun. 2., 3., 4. sütunların toplamı sıfırdır. Ancak 2

sütun aldığımızda lineer bağımsızdır. O halde, $d(C) = 3$ tür.

Örnek 2.32 $H = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}$. $H_1 + H_3 + H_4 = 0$. O halde, $d(C) = 3$ tür.

2.2.4 Lineer Kodların Denkliği

Tanım 2.33 C_1 ve C_2 iki $(n, M)_q$ – kod olsun. Eğer aşağıda verilen yollardan herhangi biri veya ikisiyle bu iki koddan birinden diğeri elde ediliyorsa C_1 ve C_2 ‘ye denk kodlar denir:

- i. Kod sözlerin n bileşeninin permütasyonu,
- ii. Belli yerdeki bileşenlerin sıfırdan farklı bir skalerle çarpımı.

Örnek 2.34 $C_1 = \{0000, 0011, 0110, 0101\}$, $C_2 = \{0000, 0011, 1001, 1010\}$ kodları $(2, 1, 4, 3)$ permütasyonuna göre denktir.

Teorem 2.16 Herhangi bir C lineer kodu standart formda olan üreteç matrisi ile C' lineer koduna denktir.

Örnek 2.35 $G = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}$ matrisinde sütunlara $(1, 4, 5, 2, 3, 6)$

permütasyonu uygulanırsa $G' = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$ denk kodunun üreteç matrisi

elde edilir.

2.3 Ağ Bağlantı Kodları ve Tanımlar

Çağımızda bilgi çok hızlı üretilen ve en yenisine sahip olan kurum ve kuruluşlara bilimsel ve teknolojik açıdan üstünlükler sağlayan bir unsur haline gelmiştir. Bilgi aynı zamanda hızlı bir şekilde tüketilmektedir. Bunun için eldeki verilerin en hızlı ve güvenli bir biçimde kullanılması gerekmektedir. Bilgi üretimi ve işleminin en verimli olarak gerçekleştirildiği bilgisayar sistemlerinin bağlanarak bilginin iletildiği ve paylaşıldığı yapılara bilgisayar ağları denilmektedir. İki ya da daha fazla bilgisayarın birbiriyle

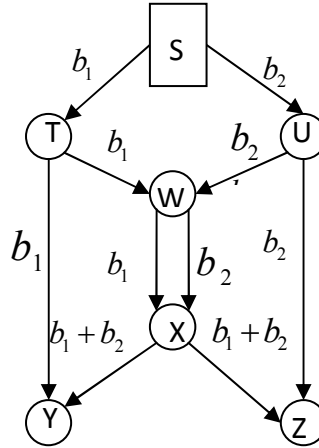
kaynaklarını paylaşması bilgisayar ağı ile gerçekleşir. Bilgisayar ağı; veri paylaşımı, haberleşme, bilgisayar kaynaklarının paylaşımı, yazılımın paylaşımı, yüksek iletim hızının sağlanması, bilgisayar ağının ekonomik olarak genişletilmesi, merkezi yönetim, organizasyon yapısının geliştirilmesi gibi avantajlar sağlar.

Ağ kodlaması en basit tanımıyla bir bilişim kuramı ve kodlama teorisi alanıdır. Ağ kodlama bir ağ içinde en fazla bilgi akışının sağlandığı bir yöntemdir. Ağ bağlantı kodlamasının birçok tanımı vardır. “Network Information Flow” adlı makalelerinde Ahlswedw. N. Cai, S. R. Li ve R. W. Yueng” ağ bağlantı kodlamasını ağ bağlantılarındaki düğümlerde yapılan kodlama” olarak tanımlamıştır. Burada kodlama girdilerden çıktılara yapılan keyfi dönüşümler olarak ifade edilir. Bu, ağ bağlantı kodlamasının en genel tanımıdır. Ağ bağlantı kodlamasının diğer bir tanımı; hatasız ağ bağlantılarındaki düğümlerde yapılan kodlamadır. Bu, ağ bağlantı kodlamasını gürültülü kanallardaki kodlamadan ayırır. Üçüncü olarak ağ bağlantı kodlaması paket kodlamasındaki düğümlerde yapılan kodlamadır. (Burada veriler paketlere bölünür ve kodlama paketlerin içerisine uygulanır.)

Bir iletişim ağ bağlantısı olarak sonlu bir yönlendirilmiş graftan bahsedebiliriz. Burada bir düğümden diğerine çoklu ayrıtlar göndermek serbesttir. Gelen ayrıtı olmayan düğümlere kaynak düğümler, herhangi diğer düğümlere de kaynak olmayan düğümler denir. Herhangi bir ayrıt, kanal olarak adlandırılır ve birim zamandaki veri aktarımı için iletişim bağlantısını temsil eder. Bir X düğümünden Y düğümüne yönlendirilmiş kanal XY ile gösterilir.

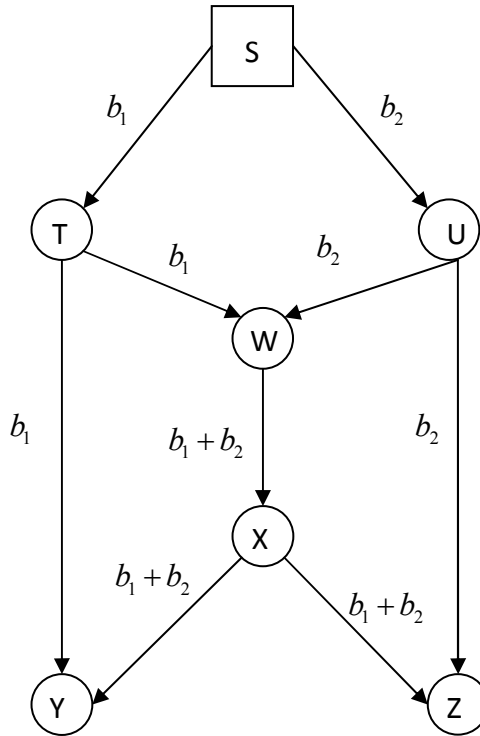
Tanım 2.34 (İletişim ağ bağlantısı) Çoklu ayrıtların bir düğümden diğerlerine ulaşmasına izin veren yönlendirilmiş bir grafa ağ bağlantısı denir. Graftaki her bir ayrıt; birim zamandaki bir verinin kapasitesi ile iletişim kanalı sunar.

Örnek 2.36



Şekil 2. 15 Devirsiz ağ bağlantısı

Bir ağ bağlantısında ne kadar mesajın iletilebileceği ve ne kadar hızlı iletilebileceği ile ilgilenilir.



Şekil 2. 16 Ağ bağlantısında xOR kodlaması

Yukarıdaki S kaynağından b_1 ve b_2 bitleri Y ve Z düğümlerine iletilirken W düğümü alınan b_1 ve b_2 bitlerini *exclusive-OR* bit, $b_1 \oplus b_2$ olarak çıkarır ve W düğümü X

düğümüne $b_1 \oplus b_2$ bitini aktarır. Öte yandan X düğümü Y ve Z düğümlerine $b_1 \oplus b_2$ aktarır. O halde Y düğümü b_1 ve $b_1 \oplus b_2$ bitleri alarak b_2 'yi benzer şekilde Z düğümü b_2 ve $b_1 \oplus b_2$ bitlerini alarak b_1 bitini dekodlar.

- *exclusive – OR* bitinin oluşması basit bir kodlamadır.

Tanım 2.36 (Kaynak düğüm) Ağ bağlantısındaki girdi ayrıtı olmayan düğümlere kaynak düğüm denir. Her bir ağ bağlantısında en az bir kaynak düğüm vardır.

Her bir T düğümü için, T düğümüne giren tüm girdi kanallarının kümesini $in(T)$ ile, T düğümünden çıkan tüm çıktı kanallarının kümesini $out(T)$ ile gösterelim. Genelliği bozmaksızın kaynak düğümü S ile göstereceğiz. Böylelikle $in(S)$ hayali kanalların kümesidir öyle ki bu kanalları üreten düğüm yoktur. Bu kanallar daima bağımsız ve sayısı w dır.

2.4 Ağ Bağlantı Kodlamasının Faydaları

Ağ bağlantı kodlanmasında bilgi tek bir kaynaktan aynı anda pek çok alıcıya gönderilebilir (multicast). Ağ bağlantı kodlamasının pek çok avantajı vardır. Bunlar; iş çıkarım oranı: Ağ bağlantı kodlamasına ilgiyi arttıran asıl neden bir ağın kapasitesini çoklu akışlarla arttırabilmesidir. “Network Information Flow” [2] adlı makalelerinde Ahlswedw. Cai, Li ve Yueng çok ünlü bir iş çıkarım örneği olan kelebek ağ bağlantısı örneğini vermişlerdir. Bu örnekte tek bir kaynak düğüm ve iki alıcı düğüm vardır ve alıcı düğümlerin ikisi de kaynaktaki mesajı bilmek isterler. Örnekte uygun gönderim işlemleri (xor) ile sağlanır. Dayanıklılık; ağ bağlantı kodlaması geleneksel yöntemle benzer şekilde bilgi paketlerini alır ve kodlanmış paketlere dönüştürür. Bu noktada kodlanmış her bir paket eşit öneme sahiptir. Yeterli sayıda kodlanmış paket alındığı takdirde dekodlama işlemi de başarıyla gerçekleştirilebilir. Ağ bağlantı kodlaması paket kayıplarına karşı dayanıklılığı sağlar. Karmaşıklık; karmaşık bir ağ bağlantısı yapısı içinde alt ağ bağlantılarında optimal çözüm sağlanır. Gizlilik; ağ bağlantı kodlamasında alıcılar tarafından alınan paketlerin tam olarak nerden gönderildiği her zaman bilinmeyebilir. Bu durum ağ bağlantısının inşasına bağlıdır.

BÖLÜM 3

AĞ BAĞLANTILARI

Bu alt bölümde ağ bağlantıları ile ilgili tanım ve teoremler, ağ bağlantıları üzerinde kodlama ve ağ kodlamasına cebirsel bir yaklaşım sunulmuştur. Bu bölüm oluşturulurken Network Coding Theory, 2005 [6] kitabından, An Algebraic to Network Coding [7] adlı makaleden yararlanılmıştır.

3.1 Devirsiz Ağ Bağlantıları

3.1.1 Devirsiz Ağ Bağlantıları ve Lineer Ağ Bağlantı Kodları

Tanım 3.1 İletişim ağ bağlantısı; çoklu ayrıtların bir düğümden diğerine ulaşmasına izin veren yönlendirilmiş bir graftır. Graftaki her bir ayrıt birim zamandaki bir verinin kapasitesi ile iletişim kanalı sunar. Bir T düğümü için $in(T)$ ile T düğümüne giren ayrıtların kümesi, $out(T)$ ile T düğümünden çıkan ayrıtların kümesini gösterir. Böylece $in(S)$ hayali kanalların kümesidir öyle ki bu kanalları üreten küme yoktur. Eğer iletişim ağ bağlantısında hiç bir yönlendirilmiş devir yoksa bu ağ bağlantısı devirsiz ağ bağlantısı olarak adlandırılır. Bundan sonra kanal ve ayrıt ifadelerini aynı anlamda kullanacağız.

- Bir veri birimi bazı F sonlu cisimlerin elemanları olarak temsil edilebilir.

Bir veri birimi bit ise $F = GF(2)$ dir. Bir mesaj w veri biriminden oluşur. Bu yüzden w boyutlu sütun vektör $x \in F^w$ dir. S kaynak düğümü bir x mesajını üretir ve her bir kanalda bir sembol aktararak onu gönderir. Ağ bağlantısındaki mesaj; ağ

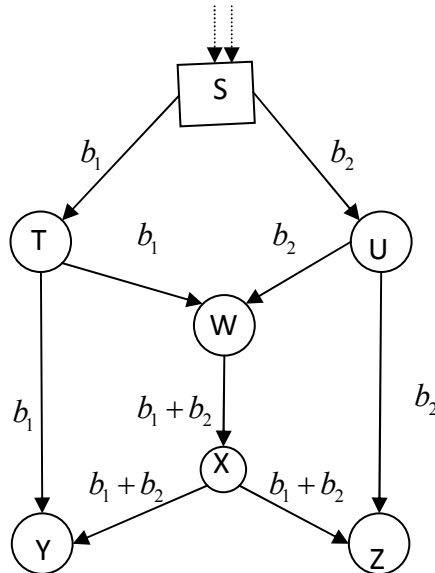
bağlantısındaki her bir e kanalındaki $\tilde{f}_e(x) \in F$ sembollerinin aktarılmasıyla başarılı. Bu mesajın kodlama fonksiyonu alınan sembollerin, tüm girdi kanallarından her bir çıktı kanallarına dönüşümüdür.

Tanım 3.2 (Devirsiz Bir Ağ Bağlantısındaki Ağ Kodunun Yerel Tanımı) F bir sonlu cisim ve w pozitif bir tamsayı olsun. Devirsiz bir ağ bağlantısındaki F değerli w boyutlu bir ağ bağlantı kodu; ağ bağlantısındaki her bir T düğümü ve $e \in out(T)$ kanalı için $\tilde{k}_e : F^{in(T)} \rightarrow F$ yerel kodlama dönüşümünden oluşur.

Tanım 3.3 (Devirsiz Bir Ağ Bağlantısındaki Ağ Kodunun Global Tanımı) F bir sonlu cisim ve w pozitif bir tamsayı olsun. Devirsiz bir ağ bağlantısındaki F değerli w boyutlu bir ağ bağlantı kodu; ağ bağlantısındaki her bir T düğümü ve $e \in out(T)$ kanalı için; $\tilde{k}_e : F^{in(T)} \rightarrow F$ yerel kodlama dönüşümü ve $\tilde{f}_e : F^w \rightarrow F$ global kodlama dönüşümünden oluşur öyle ki; her bir T düğümü ve $e \in out(T)$ için $\tilde{f}_e$ tek türlü olarak $(\tilde{f}_d(x), d \in in(T))$ dönüşümünden oluşur ve $\tilde{k}_e : (\tilde{f}_d(x), d \in in(T)) \rightarrow \tilde{f}_e(x)$ dir. (3.1)

w hayali kanalı için $\tilde{f}_e ; F^w$ uzayından w farklı koordinatlara bir izdüşümüdür.

Örnek 3.1 [6]



Şekil 3. 1 Kelebek ağ bağlantısı

Yukarıdaki şekilde $x = (b_1, b_2)$, $[GF(2)]^2$ de bir vektör olsun. Aşağıdaki global kodlama dönüşümleri ile 2 boyutlu ikili kodu temsil eder. Burada OS, OS' şeklindeki hayali kanalı temsil eder. $e = OS, ST, TW, TY$ için, $\tilde{f}_e(x) = b_1$; $e = OS', SU, UW, UZ$ için $\tilde{f}_e(x) = b_2$ $e = WX, XY, XZ$ için $\tilde{f}_e(x) = b_1 \oplus b_2$ dir. Uygun yerel kodlama dönüşümleri; $\tilde{k}_{ST}(b_1, b_2) = b_1, \tilde{k}_{SU}(b_1, b_2) = b_2, \tilde{k}_{TW}(b_1) = \tilde{k}_{TY}(b_1) = b_1,$
 $\tilde{k}_{UW}(b_2) = \tilde{k}_{UZ}(b_2) = b_2, \tilde{k}_{WX}(b_1, b_2) = b_1 \oplus b_2$
 dir.

Global kodlama dönüşümü $\tilde{f}_e$ lineer olduğu zaman; $\tilde{f}_e(x) = xf_e$ ile temsil edilir. Burada f_e , w boyutlu sütun vektörü ve x , S tarafından üretilen w boyutlu sütun mesajı temsil eder. Benzer şekilde $e \in out(T)$ için $\tilde{k}_e(y)$ lineer olduğu zaman $\tilde{k}_e(y) = yk_e$ ile temsil edilir. Burada $y \in F^{in(T)}$, T düğümü tarafından $in(T)$ boyutlu alınan sütun vektördür; w boyutlu F değerli bir devirsiz iletişim ağ bağlantısında tüm yerel kodlama dönüşümleri lineer ise tüm global kodlama dönüşümleride lineerdir. Çünkü; global kodlama dönüşümleri yerel kodlama dönüşümlerinin fonksiyonel kompozisyonlarıdır. Bunun tersi her zaman doğru değildir.

Tanım 3.4 (Devirsiz Ağ Bağlantısındaki Lineer Ağ Kodunun Yerel Tanımı) F bir sonlu cisim ve w pozitif bir tamsayı olsun. Devirsiz bir ağ bağlantısındaki F değerli w boyutlu bir ağ bağlantı kodu; her bir (d, e) bitişik ikilisi için yerel kodlama çekirdeği olarak adlandırılan $k_{d,e}$ skalerlerinden oluşur. Yani; T düğümündeki yerel kodlama çekirdeği $|in(T)| \times |out(T)|$ boyutlu $K_T = [k_{d,e}]_{d \in in(T), e \in out(T)}$ matrisidir.

Tanım 3.5 (Devirsiz Ağ Bağlantısındaki Lineer Ağ Kodunun Global Tanımı) F bir sonlu cisim ve w pozitif bir tamsayı olsun. Devirsiz bir ağ bağlantısındaki F değerli w boyutlu bir lineer ağ bağlantı kodu; her bir (d, e) bitişik ikilisi için $k_{d,e}$ yerel kodlama çekirdeğinden ve e kanali için f_e sütun vektöründen oluşur öyle ki;

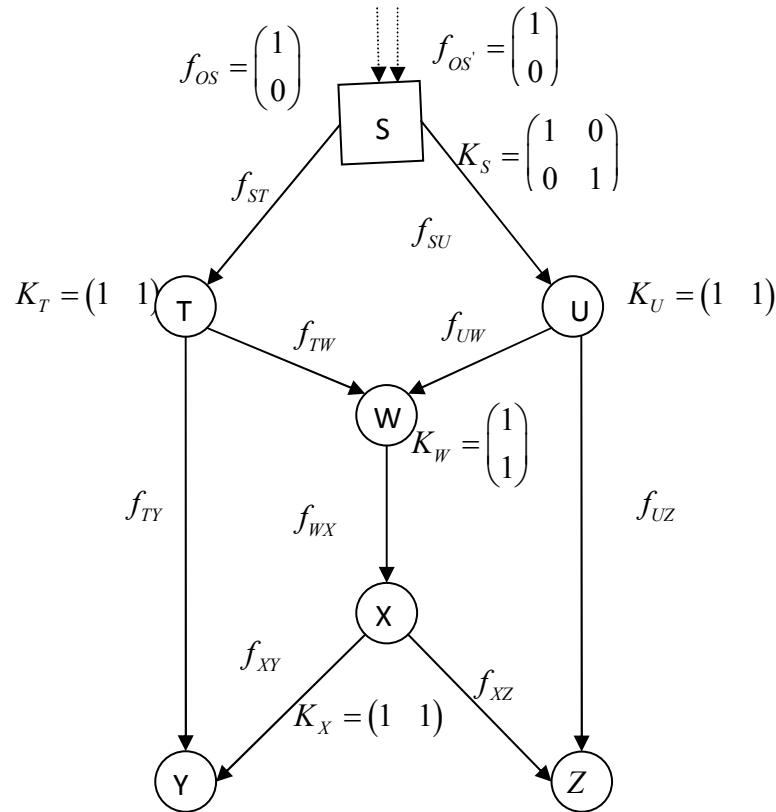
- $f_e = \sum_{d \in in(T)} k_{d,e} \cdot f_d$ burada $e \in out(T)$ (3.2)

- $e \in in(S)$ w hayali kanalları için f_e vektörü F^w vektör uzayını oluşturur. (3.3)

f_e vektörü e kanalı için global kodlama çekirdeği olarak adlandırılır. Kabul edelim ki kaynak w boyutlu sütun vektörü formunda bir mesaj üretsin. T düğümü mesajı $d \in in(T)$ kanallarından xf_d formunda alır ve xf_e hesaplayarak her bir $e \in out(T)$ kanalına iletir. $xf_e = x \cdot \sum_{d \in in(T)} k_{d,e} \cdot f_d = \sum_{d \in in(T)} k_{d,e} \cdot (xf_d)$ olur.

- Bir devirsiz lineer ağ bağlantısı için tüm yerel kodlama çekirdekleri verilmiş ise global kodlama çekirdekleri hesaplanabilir.

Örnek 3.2 [6]



Şekil 3. 2 Yerel kodlama çekirdekleri verilen ağ bağlantısı

Yukarıdaki yerel kodlama çekirdekleri ile verilen 2 boyutu lineer ağ bağlantısının global kodlama çekirdekleri;

$$f_{ST} = \sum_{d \in in(S)} k_{d,e} \cdot f_d = \begin{pmatrix} 1 \\ 0 \end{pmatrix} k_{1,1} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} k_{2,1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot 1 + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot 0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$f_{SU} = \sum_{d \in in(S)} k_{d,e} \cdot f_d = \begin{pmatrix} 1 \\ 0 \end{pmatrix} k_{1,2} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} k_{2,2} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot 0 + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot 1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

$$f_{TY} = \sum_{d \in in(T)} k_{d,e} \cdot f_d = f_{ST} \cdot k_{1,1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} k_{1,1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot 1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

$$f_{TW} = \sum_{d \in in(T)} k_{d,e} \cdot f_d = \begin{pmatrix} 1 \\ 0 \end{pmatrix} k_{1,2} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot 1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad f_{UW} = \sum_{d \in in(U)} k_{d,e} \cdot f_d = \begin{pmatrix} 0 \\ 1 \end{pmatrix} k_{1,1} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot 1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

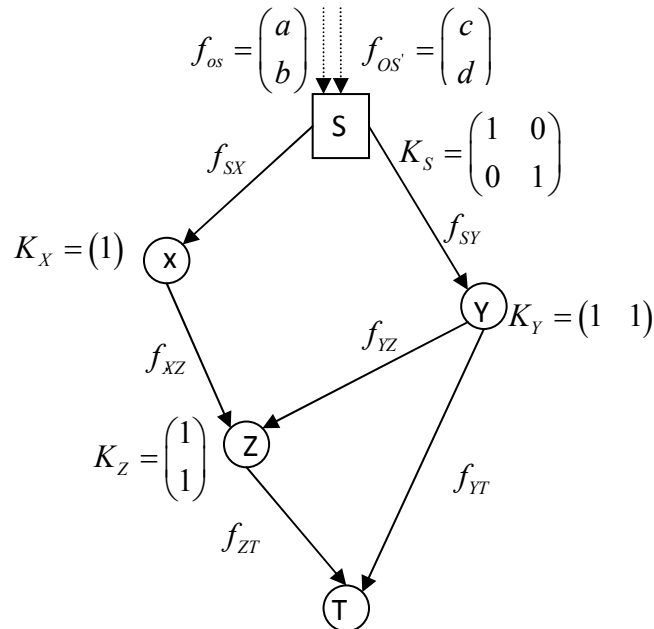
$$f_{UZ} = \sum_{d \in in(U)} k_{d,e} \cdot f_d = \begin{pmatrix} 0 \\ 1 \end{pmatrix} k_{1,2} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot 1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

$$f_{WX} = \sum_{d \in in(W)} k_{d,e} \cdot f_d = \begin{pmatrix} 1 \\ 0 \end{pmatrix} k_{1,1} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} k_{2,1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot 1 + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot 1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

$$f_{XY} = \sum_{d \in in(X)} k_{d,e} \cdot f_d = \begin{pmatrix} 1 \\ 1 \end{pmatrix} k_{1,1} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \cdot 1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad f_{XZ} = \sum_{d \in in(X)} k_{d,e} \cdot f_d = \begin{pmatrix} 1 \\ 1 \end{pmatrix} k_{1,1} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \cdot 1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

şeklinde hesaplanır.

Örnek 3.3



Şekil 3. 3 Ağ kodlaması

Yukarıdaki yerel kodlama çekirdekleri ile verilen 2 boyutlu $F = GF(2)$ değerli lineer ağ bağlantısının global kodlama çekirdekleri;

$$f_{SX} = \sum_{d \in in(S)} k_{d,e} \cdot f_d = \begin{pmatrix} a \\ b \end{pmatrix} k_{1,1} + \begin{pmatrix} c \\ d \end{pmatrix} k_{2,1} = \begin{pmatrix} a \\ b \end{pmatrix} \cdot 1 + \begin{pmatrix} c \\ d \end{pmatrix} \cdot 0 = \begin{pmatrix} a \\ b \end{pmatrix},$$

$$f_{SY} = \sum_{d \in in(S)} k_{d,e} \cdot f_d = \begin{pmatrix} a \\ b \end{pmatrix} k_{1,2} + \begin{pmatrix} c \\ d \end{pmatrix} k_{2,2} = \begin{pmatrix} a \\ b \end{pmatrix} \cdot 0 + \begin{pmatrix} c \\ d \end{pmatrix} \cdot 1 = \begin{pmatrix} c \\ d \end{pmatrix},$$

$$f_{XZ} = \sum_{d \in in(X)} k_{d,e} \cdot f_d = \begin{pmatrix} a \\ b \end{pmatrix} k_{1,1} = \begin{pmatrix} a \\ b \end{pmatrix} \cdot 1 = \begin{pmatrix} a \\ b \end{pmatrix}, f_{YZ} = \sum_{d \in in(Y)} k_{d,e} \cdot f_d = \begin{pmatrix} c \\ d \end{pmatrix} k_{1,1} = \begin{pmatrix} c \\ d \end{pmatrix} \cdot 1 = \begin{pmatrix} c \\ d \end{pmatrix},$$

$$f_{YT} = \sum_{d \in in(Y)} k_{d,e} \cdot f_d = \begin{pmatrix} c \\ d \end{pmatrix} k_{1,2} = \begin{pmatrix} c \\ d \end{pmatrix} \cdot 1 = \begin{pmatrix} c \\ d \end{pmatrix},$$

$$f_{ZT} = \sum_{d \in in(Z)} k_{d,e} \cdot f_d = \begin{pmatrix} a \\ b \end{pmatrix} k_{1,1} + \begin{pmatrix} c \\ d \end{pmatrix} k_{2,1} = \begin{pmatrix} a \\ b \end{pmatrix} \cdot 1 + \begin{pmatrix} c \\ d \end{pmatrix} \cdot 1 = \begin{pmatrix} a+c \\ b+d \end{pmatrix} \text{ şeklinde elde edilir.}$$

Örnek 3.4 [6] Şekil 3.2 deki ağ bağlantısı üzerinde 2 boyutlu genel bir lineer ağ bağlantı kodu için yerel kodlama çekirdekleri; $n, p, q, \dots, z$ değişkenler olmak üzere;

$$K_S = \begin{pmatrix} n & q \\ p & r \end{pmatrix}, K_T = \begin{pmatrix} s & t \end{pmatrix}, K_U = \begin{pmatrix} u & v \end{pmatrix}, K_W = \begin{pmatrix} w \\ x \end{pmatrix}, K_X = \begin{pmatrix} y & z \end{pmatrix} \text{ gibi verilmiş ise}$$

$$f_{OS} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, f_{OS'} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \text{ ile başlayarak global kodlama çekirdekleri yukarıdan aşağıya}$$

doğru;

$$f_{ST} = \begin{pmatrix} n \\ p \end{pmatrix}, f_{SU} = \begin{pmatrix} q \\ r \end{pmatrix}, f_{TW} = \begin{pmatrix} ns \\ ps \end{pmatrix}, f_{TX} = \begin{pmatrix} nt \\ pt \end{pmatrix}, f_{UW} = \begin{pmatrix} qu \\ ru \end{pmatrix}, f_{UZ} = \begin{pmatrix} qv \\ rv \end{pmatrix}, f_{WX} = \begin{pmatrix} nsw + qux \\ psw + rux \end{pmatrix},$$

$$f_{XY} = \begin{pmatrix} nswy + quxy \\ pswy + ruxy \end{pmatrix}, f_{XZ} = \begin{pmatrix} nswz + quxz \\ pswz + ruxz \end{pmatrix} \text{ şeklinde hesaplanır.}$$

• **Lineer Ağ Bağlantı Kodlarının Arzulanan Özellikleri:** Bilgi korunması kanunları ile ara düğümlerdeki herhangi bir kodlama planı ile veri akışı şu şekilde olur: Herhangi bir kaynak olmayan düğümden gönderilen bilginin içeriği dışarıdaki herhangi bir düğüm grubunda birikilen bilgiden alınır. Özellikle kaynak olmayan bir düğümde meydana gelen herhangi bir bilginin içeriği, o düğüm tarafından alınan bilgilerin birikiminden

elde edilir. S , kaynak düğüm ve T alıcı düğüm olmak üzere S' den T' ye maksimum akış $\max flow(T)$ ile gösterilir. Max flow-Min cut teoreminden dolayı T tarafından alınan bilgi $\max flow(T)$ yi aşamaz. Benzer şekilde, S kaynak düğümünden kaynak olmayan düğümlerin $\emptyset$ koleksiyonuna olan maksimum akışta $\max flow(\emptyset)$ ' yi aşamaz. Bu üst sınır ne ağ bağlantısının yapısına ne de kodlama planına bağlı değildir.

Tanım 3.6 f_e , devirsiz bir ağ bağlantısındaki F değerli w boyutlu bir lineer ağ bağlantı kodundaki global kodlama çekirdeği olsun. $V_T = \langle \{f_e : e \in in(T)\} \rangle$ kümesi tanımlansın. Sırası ile aşağıdakiler sağlandığında;

- $\max flow(T) \geq w$ olan her T ara düğümü için $\dim(V_T) = w$ ise;
- Her ara düğüm T için $\dim(V_T) = \min\{w, \max flow(T)\}$ ise;
- Ara düğümlerin her $\emptyset$ Koleksiyonu için $\dim(\langle \bigcup_{T \in \emptyset} V_T \rangle) = \min\{w, \max flow(\emptyset)\}$ ise;

lineer çoğa gönderim, lineer yayılım, lineer saçılma tanımlanır.

Tanımlarından da anlaşılacağı gibi her lineer saçılma bir lineer yayılım ve her lineer yayılım bir lineer çoğa gönderimdir. Bunun tersi her zaman doğru değildir.

Bir S kaynak düğümü ağ bağlantısı üzerindeki w boyutlu veri mesajlarını gönderdiğinde T alıcı düğümü mesajı dekodlayabilmesi için yeterli bilgiyi ancak ve ancak $\dim(V_T) = w$ olduğunda alır. Böylece w boyutlu lineer çoğa gönderim, w boyutlu veri mesajını tüm T ara düğümlere göndermede kullanışlıdır.

Lemma 3.1 [7] F cismi üzerindeki katsayılar ile tanımlı sıfırdan farklı $g(z_1, z_2, \dots, z_n)$ polinomu olsun. Eğer $|F|$, her bir z_j için g nin derecesinden büyük ise $g(a_1, a_2, \dots, a_n) \neq 0$ olacak şekilde $a_1, a_2, \dots, a_n \in F$ vardır.

İspat: İspat n üzerinden tümevarım ile yapılır. $n=0$ için istenilen sağlanır. Kabul edelim ki $n \geq 1$ iken $n-1$ için istenilen sağlansın. $g(z_1, z_2, \dots, z_n)$ yi katsayıları $F[z_1, z_2, \dots, z_n]$ polinomlar halkasından alınan z_n nin polinomu olarak ifade edelim. Yani; $g(z_1, z_2, \dots, z_n) = h(z_1, z_2, \dots, z_{n-1}) \cdot z_n^k + \dots$ olsun. Burada k , g' de z_n 'nin derecesi ve

$h(z_1, z_2, \dots, z_{n-1}), F[z_1, z_2, \dots, z_{n-1}]$ polinom halkasında sıfırdan farklı bir polinomdur. Tümevarım hipotezinden; $h(a_1, a_2, \dots, a_{n-1}) \neq 0$ olacak şekilde $a_1, a_2, \dots, a_{n-1} \in F$ vardır. Böylece $g(a_1, a_2, \dots, a_n, z)$ derecesi $k < |F|$ olan sıfırdan farklı bir polinomdur. Bu polinom F de k dan fazla köke sahip olamayacağı ve $k < |F|$ olduğu için $g(a_1, a_2, \dots, a_{n-1}, a_n) \neq 0$ olacak şekilde $a_n \in F$ vardır.

• w boyutlu F değerli devirli bir ağ bağlantısı üzerinde lineer ağ bağlantı kodunun. lineer çoğa gönderim, lineer yayılım, lineer saçılma olması aşağıdaki üç faktörden bağımsızdır:

- w değeri,
- F cisminin seçimi,
- Ağ bağlantı topolojisi.

Örnek 3.5 [6] Şekil 3.2 deki ağ bağlantısı üzerinde 2 boyutlu bir lineer ağ bağlantı

kodu için yerel kodlama çekirdekleri; $n, p, q, \dots, z$ değişkenler olmak üzere;

$$K_S = \begin{pmatrix} n & q \\ p & r \end{pmatrix}, K_T = \begin{pmatrix} s & t \end{pmatrix}, K_U = \begin{pmatrix} u & v \end{pmatrix}, K_W = \begin{pmatrix} w \\ x \end{pmatrix}, K_X = \begin{pmatrix} y & z \end{pmatrix} \text{ ile verilmiş ise } f_{TY}$$

ve f_{UW} yi 2×2 boyutlu L_w matrisinde yan yana yazalım: $L_w = \begin{pmatrix} ns & qu \\ pt & ru \end{pmatrix}$, f_{UZ} ve f_{XZ}

yi 2×2 boyutlu L_y matrisinde yan yana yazalım: $L_y = \begin{pmatrix} nt & nswy + quxy \\ pt & pswy + ruxy \end{pmatrix}$ ve f_{UZ} ve

f_{XZ} yi 2×2 boyutlu L_z matrisinde yan yana yazalım: $L_z = \begin{pmatrix} nswz + quxz & qv \\ pswz + ruwz & rv \end{pmatrix}$ olur.

Açıktır ki; $F[n, p, q, \dots, z]$ de $\det(L_w) \cdot \det(L_y) \cdot \det(L_z) \neq 0$ dir. F cismi yeterince büyük olduğunda $F[n, p, q, \dots, z]$ ye yukarıdaki lemmayı uygulayarak $\det(L_w) \cdot \det(L_y) \cdot \det(L_z) \neq 0$ olsun diye 12 değişken için uygun değerler bulunabilir. Gerçekten; her $\max flow(T) \geq w$ yi sağlayan T ler için $\dim(V_T) = w$ dir ve $p = q = 0, n = r = s = t = \dots = z = 1$ olduğunda $\det(L_w) \cdot \det(L_y) \cdot \det(L_z) = 1$ elde edilir.

Böylece Şekil 3.2 deki ağ bağlantısı üzerinde 2 boyutlu lineer ağ bağlantısı kodu, lineer çoğa gönderimdir ve bu durum F cisminin seçiminden bağımsızdır.

Tanım 3.7 (Jenerik Ağ Bağlantı Kodu) F bir sonlu cisim ve w pozitif bir tamsayı olsun. Devirsiz bir ağ bağlantısındaki F değerli w boyutlu bir lineer ağ bağlantı kodu; $e_j \in out(T_j)$ olacak şekilde $\{e_1, e_2, \dots, e_m\}$ kanalların keyfi bir kümesi için $f_{e_1}, f_{e_2}, \dots, f_{e_m}$ vektörleri lineer bağımsız yani $1 \leq j \leq m$ için $\langle \{f_d : d \in in(T)\} \rangle \not\subset \langle \{f_{e_k} : k \neq j\} \rangle$ şartını sağlıyor ise bu koda jenerik denir.

Burada $f_{e_1}, f_{e_2}, \dots, f_{e_m}$ arasındaki lineer bağımsızlık tüm j ler için $f_{e_i} \notin \langle \{f_{e_k} : k \neq j\} \rangle$ olmasına eşittir. O halde jenerik ağ bağlantı kodlarında global kodlama çekirdekleri lineer bağımsızdır.

- $in(S)$ de w hayali kanalı olan bir ağ bağlantısı kodu düşünelim. Ağ bağlantısındaki düğümlerin $\wp$ kümesi için $cut(\wp)$ ile $\wp$ kümesindeki düğümlerde biten fakat bu düğümlerde başlamayan kanallar gösterilir. Özellikle; $S \in \wp$ ise, $cut(\wp)$ tüm hayali kanalları içerir.

Örnek 3.6 Şekil 3.2 deki ağ bağlantısı için $cut(\{U, X\}) = \{SU, WX\}$ ve $cut(\{S, U, X, Y, Z\}) = \{OS, OS', WX, TY\}$ dir. Burada OS ve OS' hayali kanalları temsil eder.

Teorem 3.1 [6] Devirsiz bir ağ bağlantısı üzerinde w boyutlu lineer ağ bağlantısı kodu için f_e , e kanalı üzerindeki global kodlama çekirdeğini gösterebilir. O halde ara düğümlerin her bir $\wp$ kümesi için; $\langle \{f_e : e \in cut(\wp)\} \rangle \subset \langle \bigcup_{T \in \wp} V_T \rangle$ dir. Burada $V_T = \langle \{f_e : e \in in(T)\} \rangle$ dir.

İspat: $\langle \bigcup_{T \in \wp} V_T \rangle = \langle \{f_e : e, \wp \text{ deki düğümlerde biter}\} \rangle$ dir. Göstermemiz gereken; $\Psi = \{c : f_c \notin \langle \{f_e : e \in cut(\wp)\} \rangle \text{ ve } c, \wp \text{ deki düğümlerde biter}\}$ kümesinin boş olduğudur.

Aksine kabul edelim ki; Ψ kümesi boştan farklı olsun. c , Ψ' de bir kanal olsun. Yani; c , Ψ' de ki herhangi farklı kanalın aşağı akışı olmasın. $c \in out(U)$ olsun. Lineer ağ bağlantısı tanımından; f_c, f_d lerin lineer kombinasyonudur. O halde,

$f_e \notin \langle f_e : e \in \text{cut}(\wp) \rangle$ olduğu için $f_d \notin \langle f_e : e \in \text{cut}(\wp) \rangle$ olacak şekilde $d \in \text{in}(U)$ vardır. d , c' 'nin yukarı olan akışı olduğu için d , Ψ' de biten düğümlerin içinde olamaz. Böylece, d , $\wp$ düğümleri dışında sonlanır. d' nin son olarak bittiği U noktası c' yi üretir. Bu ise c' yi $\text{cut}(\wp)'$ nin bir elemanı yapar. Sonuç olarak; $f_e \notin \langle \{f_e : e \in \text{cut}(\wp)\} \rangle$ olması ile çelişir.

- Bu teorem bize $\bigcup_{T \in \wp} V_T$ yerine sadece $e \in \text{cut}(\wp)$ olacak şekildeki f_e leri bulmamızın yeterli olduğunu gösterir.

- Her jenerik ağ bağlantısı kodu bir lineer saçılma dolayısıyla bir lineer yayılım ve bir lineer çoğa gönderimdir.

3.2 Devirli Ağ Bağlantıları

3.2.1 Devirli Ağ Bağlantıları ve Lineer Ağ Bağlantı Kodları

Tanım 3.8 Bir iletişim ağ bağlantısında en az bir tane yönlendirilmiş devir varsa bu ağ bağlantısına devirli ağ bağlantısı denir.

İletişim kanalı devirli olduğu zaman tüm düğümlerdeki işlemler her bir mesajı tek başına kodlayabilir ve yukarıdan aşağıya doğru aktarabilir. Yani; her bir mesajın işlemi ayrıştırdığı mesaj dizisinden bağımsızdır.

Devirsiz ağ bağlantısında yerel ve global kodların tanımı aynıdır. Çünkü, yerel kodlama çekirdekleri verildiğinde global kodlama çekirdekleri hesaplanabilir. Diğer bir ifade ile (3.2) eşitliğinin çözümündeki yerel kodlama çekirdeklerinin terimleri ile global kodlama çekirdeklerinin terimleri eşitlenir. Eğer aynı tanımlamalar devirli ağ bağlantısı içinde yapılırsa bazı problemler meydana gelir. İlk olarak devirsiz ağ bağlantılarında bir d kanalı için f_d global kodlama çekirdeğini gösterebilir. Daha sonra ara düğümlerin her bir $\wp$ koleksiyonu için; $\langle \{f_d : d \in \text{in}(T), T \in \wp\} \rangle = \langle \{f_e : e \in \text{cut}(\wp)\} \rangle$ dir. Fakat bu devirli ağ bağlantıları için sağlanmaz. Ayrıca devirsiz ağ bağlantılarında tüm yerel kodlama çekirdekleri verildiklerinde global kodlama çekirdekleri hesaplanabiliyordu. Bu devirli ağ bağlantıları için her zaman doğru değildir. Yerel kodlama çekirdekleri verildiğinde global kodlama çekirdekleri bir veya birden fazla olabilir ya da hiç olmayabilir.

Örnek 3.7 [6] Aşağıdaki 2 boyutlu devirli ağ bağlantısı için başlangıçta yerel kodlama

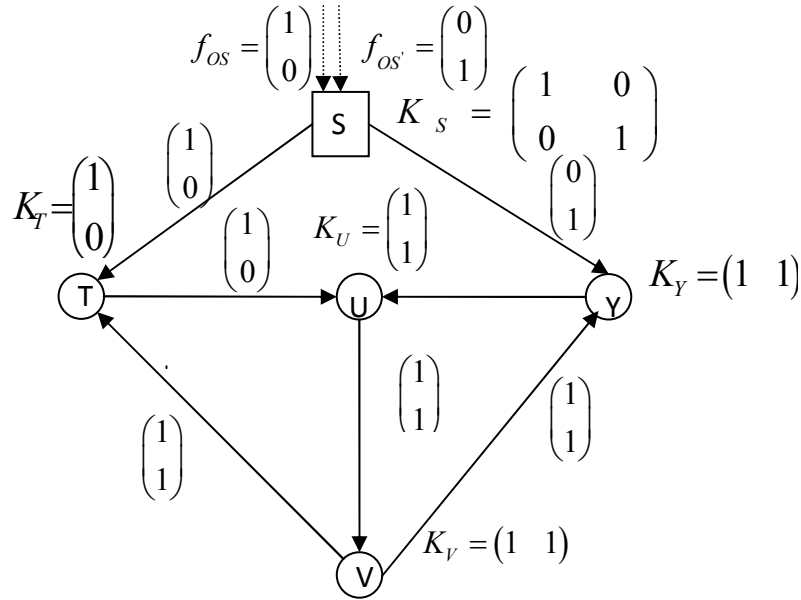
çekirdekleri; $K_S = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, K_T = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, K_U = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, K_V = \begin{pmatrix} 1 & 1 \end{pmatrix}, K_Y = \begin{pmatrix} 1 & 1 \end{pmatrix}$ şeklinde

verilsin.

O halde; global kodlama çekirdeklerini;

$$f_{ST} = \sum_{d \in in(S)} k_{d,e} \cdot f_d = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot k_{1,1} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot k_{2,1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot 1 + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot 0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

$$f_{SY} = \sum_{d \in in(S)} k_{d,e} \cdot f_d = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot k_{1,2} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot k_{2,2} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot 0 + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot 1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$



Şekil 3. 4 Devirli ağ bağlantısında kodlama

$$f_{YU} = \sum_{d \in in(Y)} k_{d,e} \cdot f_d = f_{SY} \cdot k_{1,1} + f_{VY} \cdot k_{2,1} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot k_{1,1} + f_{VY} \cdot 1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot k_{1,1} + f_{VY}$$

$$f_{VY} = \sum_{d \in in(V)} k_{d,e} \cdot f_d = f_{UV} \cdot k_{1,2} = f_{UV} \cdot 1 = f_{UV}$$

$$f_{UV} = \sum_{d \in in(U)} k_{d,e} \cdot f_d = f_{TU} \cdot k_{1,1} + f_{YT} \cdot k_{2,1} = f_{TU} \cdot 1 + f_{YT} \cdot 1 = f_{TU} + f_{YT},$$

$$f_{VT} = \sum_{d \in in(V)} k_{d,e} \cdot f_d = f_{UV} \cdot k_{1,1} = f_{UV} \cdot 1 = f_{UV},$$

$$f_{TU} = \sum_{d \in in(T)} k_{d,e} \cdot f_d = f_{ST} \cdot k_{1,1} + f_{VT} \cdot k_{2,1} = f_{ST} \cdot 1 + f_{VT} \cdot 0 = f_{ST},$$

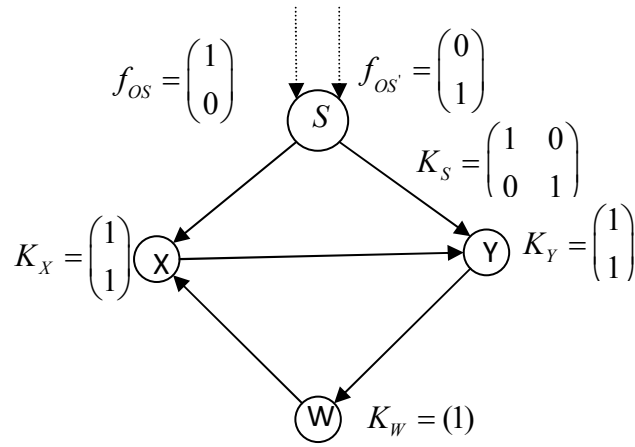
Matris denklem sistemlerinin çözümü ile global kodlama çekirdekleri; $f_{YU} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$,

$f_{VY} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $f_{UV} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $f_{VT} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $f_{TU} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ şeklinde elde edilir. Bu şekilde inşa edilen

ağ bağlantısında global kodlama çekirdekleri için tek çözüm vardır.

Örnek 3.8 [6] Yerel kodlama çekirdekleri $K_S = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $K_X = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $K_Y = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $K_W = (1)$,

biçiminde verilen Şekil 3.5 deki ağ bağlantısı için global kodlama çekirdeklerini hesaplayalım:



Şekil 3. 5 Çözumsuz devirli ağ bağlantısı

$$f_{SX} = \sum_{d \in in(S)} k_{d,e} \cdot f_d = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot k_{1,1} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot k_{2,1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot 1 + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot 0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

$$f_{SY} = \sum_{d \in in(S)} k_{d,e} \cdot f_d = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot k_{1,2} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot k_{2,2} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot 0 + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot 1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

$$f_{XY} = \sum_{d \in in(X)} k_{d,e} \cdot f_d = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot k_{1,1} + f_{WX} \cdot k_{2,1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot 1 + f_{WX} \cdot 1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + f_{WX}, (*)$$

$$f_{YW} = \sum_{d \in in(Y)} k_{d,e} \cdot f_d = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot k_{1,1} + f_{XY} \cdot k_{2,1} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot 1 + f_{XY} \cdot 1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} + f_{XY}, (**)$$

$$f_{WX} = \sum_{d \in in(W)} k_{d,e} \cdot f_d = f_{YW} \cdot k_{1,1} = f_{YW} \cdot 1 = f_{YW} \text{ dir. (***)}$$

(*) , (**) ve (***) eşitliğinden;

$$f_{XY} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + f_{WX} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + f_{YW} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} + f_{XY} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + f_{XY} \text{ elde edilir.}$$

O halde uygun global kodlama çekirdekleri yoktur. Bu durumu mesaj aktarımı için şöyle yorumlayabiliriz: Uygun global kodlama çekirdeklerinin olmaması S nin ürettiği $x = (a, b) \in F^2$ mesajının iletilmesinde XY kanalı üzerinde aktarılan mesaj $p = x.f_{XY}$,

YW kanalı üzerinde aktarılan mesaj $q = x.f_{YW}$ ve WX kanalı üzerinde aktarılan mesaj $r = x.f_{WX}$ olmak üzere; $p = a + r, q = b + p, r = q$ olduğu için $a + b = 0$ elde edilir. Dolayısıyla a ve b lineer bağımlı olur. Bu ise çelişkidir. Yani bu ağ bağlantısında mesaj iletilemez.

Örnek 3.9 [6] $F, GF(2)$ nin cisim genişlemesi olsun. Şekil 3.5 deki ağ bağlantısının

yerel kodlama çekirdekleri verilsin. Fakat $K_s = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$ olsun. Bu durumda

$$f_{SX} = \sum_{d \in in(S)} k_{d,e} \cdot f_d = \begin{pmatrix} 1 \\ 0 \end{pmatrix} k_{1,1} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} k_{2,1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot 1 + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot 0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

$$f_{SY} = \sum_{d \in in(S)} k_{d,e} \cdot f_d = \begin{pmatrix} 1 \\ 0 \end{pmatrix} k_{1,2} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} k_{2,2} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot 1 + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot 0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

$$f_{XY} = \sum_{d \in in(X)} k_{d,e} \cdot f_d = \begin{pmatrix} 1 \\ 0 \end{pmatrix} k_{1,1} + f_{WX} \cdot k_{2,1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot 1 + f_{WX} \cdot 1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + f_{WX},$$

$$f_{YW} = \sum_{d \in in(Y)} k_{d,e} \cdot f_d = \begin{pmatrix} 1 \\ 0 \end{pmatrix} k_{1,1} + f_{XY} \cdot k_{2,1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot 1 + f_{XY} \cdot 1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + f_{XY}, \text{ elde edilir.}$$

Dolayısıyla $f_{WX} = f_{YW} \cdot k_{1,1} = f_{YW} \cdot 1 = f_{YW} \Rightarrow f_{YW} = f_{WX}$ çözümü için üç durum vardır.

$$\bullet f_{SX} = f_{SY} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, f_{XY} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \Rightarrow f_{WX} = f_{YW} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\bullet f_{SX} = f_{SY} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, f_{XY} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \Rightarrow f_{WX} = f_{YW} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\bullet f_{SX} = f_{SY} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, f_{XY} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \Rightarrow f_{WX} = f_{YW} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \text{ yani, global kodlama çekirdekleri}$$

için üç farklı durum elde edilir.

3.3 Ağ Bağlantı Kodlamasına Cebirsel Bir Yaklaşım

Bir ağ bağlantısı; gönderici, anahtar, yönlendirici ve alıcıları ilişiklendiren yönlendirilmiş bağlantıların bir koleksiyonudur. Bu bölümdeki amaç ağ bağlantıları kodlamasına cebirsel bir yaklaşım sunmaktır. Bir ağ bağlantısı $G = (V, E)$, V düğümlerin ve E ayrıtların kümesi olmak üzere yönlendirilmiş graflar ile temsil edilir. Ağ bağlantılarında çoklu ayrıtların düğümler arasında olmasına izin verilebilir. Böylece $E = V \times V \times Z_i$ olarak sunulabilir. Burada sonuncu ifade iki düğüm arasındaki ayrıtları sayar. O halde önceki bölümlerdeki ifadelere ek olarak; bir ayrıtı $e = (v', v, i)$ olarak ifade edebiliriz. Bu arada $v = giriş(e)$ ve $v' = çıkış(e)$ olarak gösterilir. Bir T düğümünün girdi ve çıktı kanalları sırası ile şu şekilde tanımlanabilir: $in(T) = \{e \in E : giriş(e) = T\}$ ve $out(T) = \{e \in E : çıkış(e) = T\}$; O halde $\delta_i = |in(T)|$ ve $\delta_o = |out(T)|$ ile de sırası ile girdi ve çıktıların derecesi gösterilsin. Bir ağ bağlantısında her bir e kanalı bir tamsayı ile ilişkilendirilmiştir. Bu sayıya kanalın kapasitesi denir ve $C(e)$ ile gösterilir. Ağ bağlantısında gözlenen işlemler başlangıçta v kaynak düğümde hayali kanallardan kaynaklanan $\mu(v)$ tane işlem; $\chi(v) = \{X(v, 1), X(v, 2), \dots, X(v, \mu(v))\}$ olsun. Bir ağ bağlantısında seçilmiş düğümler arasında iletişim sağlanmak istenir yani; bazı $X(v)$ işlemlerini ağ bağlantısındaki farklı v' düğümlerine de iletmek istenir. Bir c bağlantısı $c = (v, v', \chi(v, v')) \in V \times V \times \zeta_{\chi(v)}$ ile gösterilir. Burada $\zeta_{\chi(v)}$ ile $\chi(v)$ kümesinin eleman sayısı gösterilir. Bir $c = (v, v', \chi(v, v'))$ bağlantısı verildiğinde v düğümü kaynak ve v' düğümü alıcı düğüm olarak adlandırılır ve $v = kaynak(c)$ ve $v' = alıcı(c)$ ile gösterilir. Tanım gereği $kaynak(c) \neq alıcı(c)$ dir. Bir v düğümü birim zamanda $e = (v, u)$ kanalından v tarafından üretilen mesajı en fazla $C(e)$ oranında gönderebilir. e kanalı üzerinde gönderilen rastgele işlemler $Y(e)$ ile gösterilsin. $X(v)$ işlemlerine ilaveten v düğümü $v' \in in(v)$ kanalları için $Y(e')$ işlemlerini de gözlemleyebilir. Genel olarak $Y(e)$, $e = (v, u) \in out(v)$ kanalı üzerindeki rastgele işlemi hem $X(v)$ hem de tüm $e' \in in(v)$ ler için $Y(e')$ nin fonksiyonlarıdır. Eğer v herhangi bir c bağlantısının alıcısı ise $Z(v) = \{Z(v, 1), Z(v, 2), \dots, Z(v, \nu(v))\}$ $\nu(v)$ tane rastgele işlemlerinin koleksiyonu

$v = giriş(c)$ nin çıktılarını gösterir. Eğer $X(v, v')$ nin bir kopyası $Z(v')$ nün bir alt kümesi ise $c = (v, v', \chi(v, v'))$ bağlantısına başarıyla kurulmuştur denir.

c bağlantılarından oluşan G ağına bakalım. Bazı varsayımlar:

- G 'deki herhangi bir bağlantının kapasitesi sabittir. Örneğin bir zaman birimindeki (sn, dk, sa) bit. Bu geliş güzel bir kesinlik derecesiyle belirlenebilecek bir varsayımdır. Kapasite zaman birimindeki bit'i geçtiği takdirde bunu birim kapasiteli paralel kenarlar olarak modellendiririz. Önemsenmeyecek kadar küçük kapasiteler için, zaman birimi büyütülerek bir yaklaşık değer bulunabilir.

- Ağdaki her bir bağlantı aynı gecikmeye sahiptir. Bu durum için gecikmeyi sıfır kabul ederek ağ, gecikmesiz ağ olarak düşüneceğiz. Gecikmesiz ağların istikrar sorununun oluşmaması için her zaman döngüsüz kabul edilir.

Değişik v 'ler için $X(v, l)$ 'ler bağımsızdır. Bu varsayım ağın doğasını yansıtır. Özelliklerde farklı yerlerde ağa girilen bilgilerin bağımsız olduğu varsayılır.

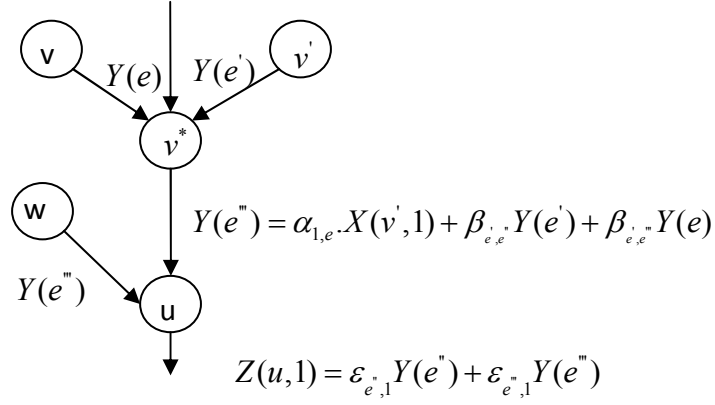
Yukarıdaki kabullere ek olarak ağ bağlantısındaki iletişimin bitlerin vektörlere aktarılması ile gerçekleştirileceğini kabul edilecektir. Bu aktarmalardaki vektörlerin uzunlukları eşittir.

m uzunluğundaki ikili vektörün F_{2^m} nin elemanı olduğunu kabul edeceğiz. F_{2^m} , 2^m elemanlı sonlu cisimdir. Böylece $X(v, l), Y(e), Z(v, l)$ rastgele işlemleri $X(v, l) = \{X_0(v, l), X_1(v, 2), \dots\}$, $Y(e) = \{Y_0(e), Y_1(e), \dots\}$ ve $Z(v, l) = \{Z_0(v, l), Z_1(v, l), \dots\}$ F_{2^m} deki sembollerin dizilerinden oluşur.

Tanım 3.9 $G = (V, E)$ gecikmesiz ağ bağlantısı olsun. Eğer herhangi bir bağlantı üzerindeki bir $e = (v, u, i) \in E$ ayrıtındaki $Y(e)$ rastgele işlemi;

$$Y(e) = \sum_{l=1}^{\mu(v)} \alpha_{l,e} \cdot X(v, l) + \sum_{e': giriş(e') = çıkış(e)} \beta_{e',e} \cdot Y(e')$$

biçiminde tanımlanır ve $\alpha_{l,e}$ ve $\beta_{e',e}$, F_{2^m} nin elemanları ise G ye F_{2^m} üzerinde lineer ağ bağlantı kodu denir. v düğümündeki $Z(v, l)$ çıktıları $e \in in(v)$ için $Y(e)$ rastgele işlemlerinden oluşur. Yani; $Z(v, j) = \sum_{e': giriş(e') = v} \varepsilon_{e',j} Y(e')$ dir. Burada $\varepsilon_{e',j}$, F_{2^m} nin elemanıdır. m ve F_{2^m} yi $\alpha_{l,e}$, $\beta_{e',e}$, $\varepsilon_{e',j}$ yi içerecek şekilde büyük seçebiliriz.

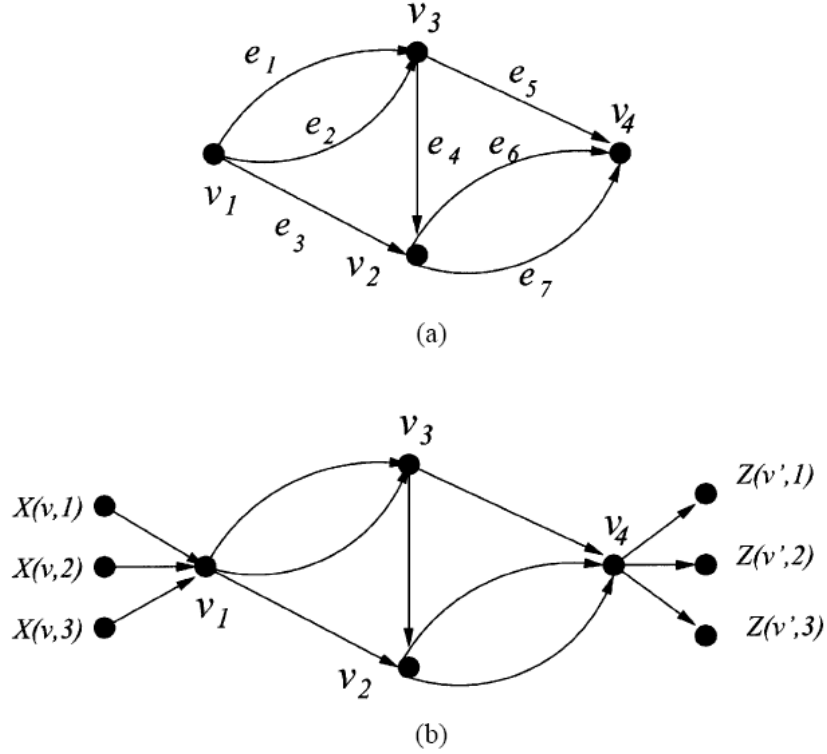


Şekil 3. 6 Cebirsel ağ kodlaması

•Bir $G = (V, E)$ ağ bağlantısı ve $\mathfrak{T}$ bağlantı kümesi verildiğinde $(G, \mathfrak{T})$ ikilisi olarak ağ bağlantı kodlama problemini tanımlayabiliriz. Bu problemin hangi cebirsel şartlar altında çözülebilir olduğu incelenmesi gereken bir problemidir. Yani; tüm istenilen bağlantıların olması için F_{2^m} de uygun $\alpha_{l,e}$, $\beta_{e',e}$, $\varepsilon_{e',j}$ elemanlarının bulunabilmesi ağ bağlantısı için ağ bağlantı kodlama problemidir. Bu $\alpha_{l,e}$, $\beta_{e',e}$, $\varepsilon_{e',j}$ elemanlarının kümesine ağ bağlantı kodlama probleminin çözümü denir. Eğer böyle bir çözüm varsa ağ bağlantı problemine çözülebilir denir. Eğer $\alpha_{l,e}$, $\beta_{e',e}$, $\varepsilon_{e',j}$ zamandan bağımsız (bağımlı) ise zaman- değişmez (zaman değişmeli) denir. Bağlantı hatası gerçekleşen ağ örnekleri de göz önüne alınmaktadır. Bağlantı hataları ergodik süreçler olarak kabul edilmezler ve bir bağlantının ya düzgün bir şekilde çalışacağı ya da ağdan gereken biçimde kaldırılacağı düşünülür. Bir bağlantı hata modeli $|E|$ uzunlukta z ikili vektörleri ile tanımlanabilir ve bu şekilde z' de ki her konum G' de bir kenarla ilişkilendirilir. Eğer bir bağlantı hata verirse z' de karşılık gelen konum 1'e eşittir, aksi halde z' de ki girdinin bağlantıdaki karşılığı sıfıra eşittir. Bir ağda z' nin karşılığı olan bağlantılar kaldırıldığında ağ çözülebilir ise bu ağın, bağlantı hata örneklerine göre çözülebilir olduğu söylenebilir. Belirli bir hata örneğinin çözülebilirliğini araştırmak basit olsa da hata örneği sınıfları için ortak çözümler bulmak çok daha ilgi çekici bir görevdir. Eğer ağ için herhangi bir hata örneği durumunda aynı elemanlarla çözüm bulunabiliyorsa bu ağ çözümünün bağlantı hata örneği dizisinde statik olduğu söylenebilir. Statik çözümler özellikle tercih edilir. Çünkü:

- 1) Eğer $z \in F$ ise yeni bir çözüm bulunması ve ağda dağıtılması gerekmez.

2) İç ağdaki bireysel düğümler hata örneğinde önemsiz olabilir. Yani ağdaki bir düğümden gerçekleşen basit bir uygulama belirli hata örneklerinden bağımsızdır.



Şekil 3. 7 Girdi ve çıktıları verilen ağ bağlantısı

3.3.1 Cebirsel Modelleme

v düğümünün ağdaki tek kaynak olduğunu varsayalım. $\underline{x} = (X(v,1), X(v,2), \dots, X(v, \mu(v)))$, v' de gözlemlenen girdi verilerinin vektörü olarak belirtilir. Aynı şekilde v' tek alıcı düğüm olsun. $\underline{z} = (Z(v',1), Z(v',2), \dots, Z(v', \nu(v')))$ da çıktı verilerinin vektörü olur. F_{2^m} lineer ağın ele almanın en önemli sonucu, bir x girdi vektörü ve bir z çıktı vektörü arasındaki ilişkiyi tanımlayan transfer matrisini ortaya koyabilmektir. M , x girdili ve y çıktılı bir ağın sistem transfer matrisi olsun. Yani $\underline{y} = \underline{x}M$ olur. Sabit bir $\alpha_{l,e}$, $\beta_{e',e}$ ve $\varepsilon_{e',j}$ çarpan dizisi için M , elemanları F_{2^m} sonlu cisminden olan bir matristir. Bu durumu bir adım daha ilerletirsek, çarpanları belirsiz değişkenler olarak kabul ederiz. Bu şekilde M matrisinin $\alpha_{l,e}$, $\beta_{e',e}$ ve $\varepsilon_{e',j}$ elemanlarının $F_2[\dots, \alpha_{l,e}, \dots, \beta_{e',e}, \dots, \varepsilon_{e',j}, \dots]$ halkasındaki çok terimliler olduğunu düşünebiliriz.

Şekil 3.7'deki ağı ele alalım. Aşağıdaki eşitlik dizisi $\alpha_{l,e}$, $\beta_{e',e}$ ve $\varepsilon_{e',j}$ parametrelerini ve ağıdaki verileri yönetir.

$$\begin{aligned}
Y(e_1) &= \alpha_{1,e_1} X(v,1) + \alpha_{2,e_1} X(v,2) + \alpha_{3,e_1} X(v,3) \\
Y(e_2) &= \alpha_{1,e_2} X(v,1) + \alpha_{2,e_2} X(v,2) + \alpha_{3,e_2} X(v,3) \\
Y(e_3) &= \alpha_{1,e_3} X(v,1) + \alpha_{2,e_3} X(v,2) + \alpha_{3,e_3} X(v,3) \\
Y(e_4) &= \beta_{e_1,e_4} Y(e_1) + \beta_{e_2,e_4} Y(e_2) \\
Y(e_5) &= \beta_{e_1,e_5} Y(e_1) + \beta_{e_2,e_5} Y(e_2) \\
Y(e_6) &= \beta_{e_3,e_6} Y(e_3) + \beta_{e_4,e_6} Y(e_4) \\
Y(e_7) &= \beta_{e_3,e_7} Y(e_3) + \beta_{e_4,e_7} Y(e_4) \\
Z(v',1) &= \varepsilon_{e_5,1} Y(e_5) + \varepsilon_{e_6,1} Y(e_6) + \varepsilon_{e_7,1} Y(e_7) \\
Z(v',2) &= \varepsilon_{e_5,2} Y(e_5) + \varepsilon_{e_6,2} Y(e_6) + \varepsilon_{e_7,2} Y(e_7)
\end{aligned}$$

x ve y arasındaki ilişkiyi tanımlayan transfer matrisini hesaplamak kolaydır. Özellikle A ve B matrisleri şu şekilde tanımlanabilir.

$$A = \begin{pmatrix} \alpha_{1,e_1} & \alpha_{1,e_2} & \alpha_{1,e_3} \\ \alpha_{2,e_1} & \alpha_{2,e_2} & \alpha_{2,e_3} \\ \alpha_{3,e_1} & \alpha_{3,e_2} & \alpha_{3,e_3} \end{pmatrix}, B = \begin{pmatrix} \varepsilon_{e_5,1} & \varepsilon_{e_5,2} & \varepsilon_{e_5,3} \\ \varepsilon_{e_6,1} & \varepsilon_{e_6,2} & \varepsilon_{e_6,3} \\ \varepsilon_{e_7,1} & \varepsilon_{e_7,2} & \varepsilon_{e_7,3} \end{pmatrix} \text{ olmak üzere } M \text{ sistem}$$

matrisinin aşağıdakine eşit olduğu bulunur.

$$M = A \begin{pmatrix} \beta_{e_1,e_5} & \beta_{e_1,e_4} \beta_{e_4,e_6} & \beta_{e_1,e_4} \beta_{e_4,e_7} \\ \beta_{e_2,e_5} & \beta_{e_2,e_4} \beta_{e_4,e_6} & \beta_{e_2,e_4} \beta_{e_4,e_7} \\ 0 & \beta_{e_3,e_6} & \beta_{e_3,e_7} \end{pmatrix} B^T$$

$$\det(M) = \det(A)(\beta_{e_1,e_5} \beta_{e_2,e_4} - \beta_{e_2,e_5} \beta_{e_1,e_4})(\beta_{e_4,e_6} \beta_{e_3,e_7} - \beta_{e_4,e_7} \beta_{e_3,e_6}) \det(B)$$

M determinantının F_{2^m} 'de sıfırdan farklı olması için parametreler F_2^m den seçilebilir.

Böylece genel matris M 'yi de dönüşüm matrisi olarak kurabilmek için A 'yı birim matris ve B 'yi seçeriz. Böyle bir çözüm $\beta_{e',e}$ tipinin bütün diğer parametreleri sıfıra eşit seçilirken $\beta_{e_1,e_5} = \beta_{e_2,e_4} = \beta_{e_4,e_6} = \beta_{e_3,e_7} = 1$ olacaktır. F cisminin tüm genişlemelerinin içerildiği $\overline{F}$ ile F' nin kapanışı gösterilmek üzere F' nin cebirsel kapanışı üzerinde, ortaya çıkan ağ kodlama problemi için sonsuz sayıda çözüm bulunur.

Bunlar, M transfer matrisinde sıfırdan farklı bir determinanı oluşturan $\beta_{e',e}$ için tüm seçimleridir denilebilir. Örneğe bakacak olursak ağın en önemli özelliği, $(\beta_{e_1,e_5}\beta_{e_2,e_4} - \beta_{e_2,e_5}\beta_{e_1,e_4})(\beta_{e_4,e_6}\beta_{e_3,e_7} - \beta_{e_4,e_7}\beta_{e_3,e_6})$ eşitliğinin sıfıra eşit olmasını diye değişkenler arasından seçim yapabilmesidir.

Teorem 3.2 (Max-Flow Min-Cut) [7] Tek kaynaklı ve alıcılı bir $G = (V, E)$ ağ bağlantısı verilsin. Yani istenen tek bağlantı $c = (v, v', X(v, v'))$ olsun. Ağ kodlama problemi ancak ve ancak $R(c)$ bağlantı hızı v ve v' arasındaki tüm kesitlerin minimum eşit değerine eşit ya da daha az olduğu takdirde çözülebilir.

Aşağıda gecikmesiz F_{2^m} lineer iletişim ağ bağlantısının tanımını vereceğiz.

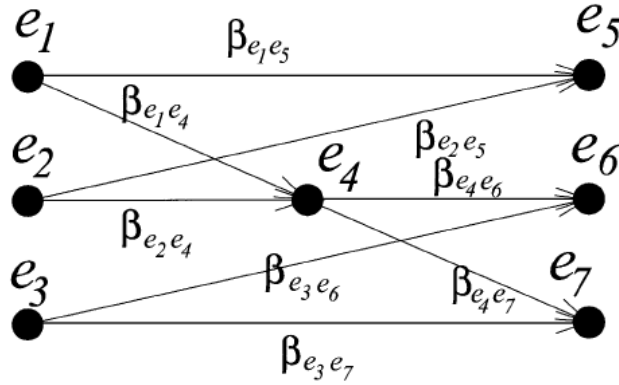
Teorem 3.3 [7] Bir lineer ağın v kaynak düğüm, v' alıcı düğüm ve $R(c)$ hızında istenen bir $c = (v, v', X(v, v'))$ bağlantısı ile verildiğini varsayalım. Aşağıdaki 3 durum eşittir.

- 1) Noktadan noktaya (p2p) bir bağlantı $c = (v, v', X(v, v'))$ olasıdır.
- 2) $R(c)$ hızı için v ve v' arasında Min-Cut Max-Flow koşulu (Teorem 3.2) karşılanır.
- 3) $R(c) \times R(c)$ boyutlu ve M transfer matrisinin determinanı $F_2[\dots, \alpha_{l,e}, \dots, \beta_{e',e}, \dots, \varepsilon_{e',j}, \dots]$ halkası üzerinde sıfırdan farklıdır.

İSPAT: Teoremin büyük bir kısmı Min-Cut Max-Flow Teoreminin doğrudan sonucudur. Özellikle, 1 ve 2 maddeler Teorem 3.2' e denktir. Aslına bakılırsa, bu teorem, tam sayılı akışa sahip ağlar için yalnızca tek kaynaklı tek alıcılı olarak işler. Ford–Fulkerson algoritması [8] böylece kaynak ve alıcı düğüm noktaları arasında $R(c)$ hızları farklı yolları verir. 1. ve 3. Maddelerin eşitliğini gösterelim. Bu, bu şekilde 2 ve 3 ün eşitliğini de gösterecektir. Ford–Fulkerson algoritması lineer ağ kodlama problemi için bir çözümün var olduğunu ortaya koyar. Lineer ağ kodlama probleminin parametreleri için bu çözümü seçmek, M 'nin tanım matrisi olduğunu ve böylece $F_2[\dots, \alpha_{l,e}, \dots, \beta_{e',e}, \dots, \varepsilon_{e',j}, \dots]$ de determinantının benzer şekilde sıfırlanmadığı sonucuna götürür. Aksine M 'nin determinanı $F_2[\dots, \alpha_{l,e}, \dots, \beta_{e',e}, \dots, \varepsilon_{e',j}, \dots]$ 'nin üzerinde sıfırdan farklı ise benzer şekilde parametreler seçerek ters matrisi oluşturabiliriz. Lemma 1'de gösterildiği gibi bu determinanı sıfırdan farklı yapmak için

parametreler istenen şekilde seçilebilir. Böylece 3. madde 1. maddeyi doğrular ve eşitlik gösterilmiş olur.

Tanım 3.10 (Dönüşüm Matrisi) Lineer iletişim ağ bağlantısındaki herhangi bir v_i düğümünün çıkan ayrıtlarının üzerindeki işlemler giren ayrıtlarının üzerindeki işlemlerin lineer kombinasyonudur. Lineer iletişim ağ bağlantısı üzerindeki ayrıtların arasındaki bu ilişki önemli bir problemdir. Eğer $giriş(e) = çıkış(e')$ ise herhangi bir $e = (u, v)$ ayrıtı $e' = (v, u')$ ayrıtlarının içine beslenir denir. $G = (V, E)$ grafının “yönlendirilmiş işaretli çizgi grafi” $\aleph(v, \varepsilon)$ olarak gösterilir. Burada $v = E$ düğümlerin kümesini ve $\varepsilon = \{(e, e') : giriş(e) = çıkış(e')\}$ ayrıtların kümesini gösterir. Tüm (e, e') ayrıtları $\beta_{e, e'}$ ile işaretlensin. Şekil 3.7 (a) daki ağ bağlantısını işaretlenmiş çizgi grafi;



Şekil 3. 8 İşaretlenmiş çizgi grafi

Üst üçgensel matrislerin kümesi bir gruptur. Bu gruplara Borel alt grupları denir. Bu matrislerin terslenebilir olması demek köşegen üzerinde hiç sıfır olmaması demektir. Bir ağ bağlantısının çok sayıda kaynak ve alıcılara sahip olduğu durumu düşünersek $\underline{x} = (x_1, x_2, \dots, x_\mu) = \{X(v_1, 1), X(v_1, 2), \dots, X(v_1, \mu(v_1)), X(v_2, 1), \dots, X(v_{|v|}, \mu(v_{|v|}))\}$, V deki tüm girdi işlemlerinin vektörü olsun. Eğer bir ağ bağlantısında v kaynak düğüm değil ise uygun $\mu(v)$ parametresi sıfıra eşit olur. $\underline{x} = (x_1, x_2, \dots, x_\mu)$, $\mu = \sum_i \mu(v_i)$ uzunluğunda bir vektördür. A matrisi $\mu \times |E|$ boyutunda elemanları,

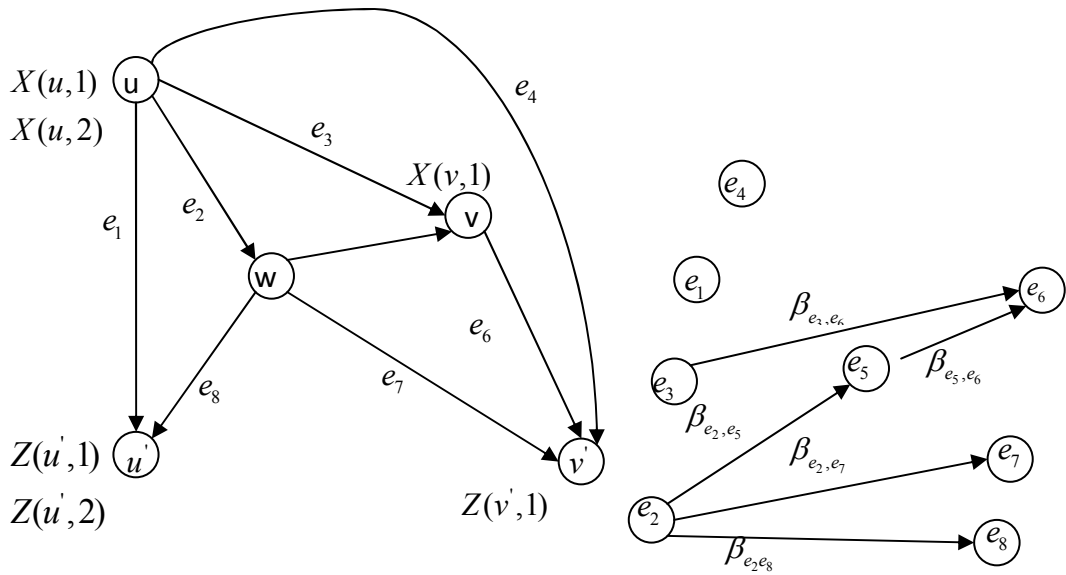
$$A_{i,j} = \begin{cases} \alpha_{i,j}, & x_i = X(çıkış(e_j), l) \\ 0, & diğer \end{cases} \quad \text{şeklindedir.} \quad \text{Benzer} \quad \text{biçimde}$$

$$\underline{z} = (z_1, z_2, \dots, z_\mu) = \{Z(v_1, 1), Z(v_1, 2), \dots, Z(v_1, \mu(v_1)), Z(v_2, 1), \dots, Z(v_{|v|}, \mu(v_{|v|}))\} \quad \text{çıkış}$$

işlemlerinin vektörü olsun. Eğer ağ bağlantısındaki v_j düğümü bir çıkış değil ise $\mathcal{G}(v_j)$ sıfıra eşit alınır. $\underline{z}, \mathcal{G} = \sum_i \mathcal{G}|v_i|$ uzunluğunda bir vektördür. $|\mathcal{G}| \times |E|$ boyutundaki bir B matrisi; $B_{i,j} = \begin{cases} \varepsilon_{e_{j,l}}, & z_i = Z(\text{giriş}(e_j), l) \\ 0, & \text{diğer} \end{cases}$, biçiminde tanımlanır. Eğer bu koşullar sağlanabilirse ağ kodlama yöntemiyle veri iletimi sağlanabilir denir.

Aşağıdaki teoremi vermeden önce çizge teori ve bitişiklik matrisi ile ilgili bazı özellikleri ifade edelim.

Örnek 3.10 [7]



Şekil 3. 9 Graf ve grafa ait işaretlenmiş çizgi grafi

Şekil 3.9 daki ağ bağlantısı ve ağ bağlantısına uygun işaretlenmiş çizgi grafi verilmiş olsun. Şekildeki ağ bağlantısı iki $c_1 = (u, u', \{X(u,1), X(u,2)\})$ ve $c_2 = (v, v', \{X(v,1)\})$ bağlantılarını içersin ve ayrıtları $e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8$ şeklinde sıralansın. Bu ayrıtlar için işaretlenmiş çizgi grafi için ((b) den) F bitişiklik matrisi yazılırsa;

$$F = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \beta_{e_2, e_5} & 0 & \beta_{e_2, e_6} & \beta_{e_2, e_7} \\ 0 & 0 & 0 & 0 & 0 & \beta_{e_3, e_6} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \beta_{e_5, e_6} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{ayrıca } A \text{ ve } B \text{ matrisleri;}$$

$$A = \begin{pmatrix} \alpha_{1, e_1} & \alpha_{1, e_2} & \alpha_{1, e_3} & 0 & 0 & 0 & 0 & 0 \\ \alpha_{2, e_1} & \alpha_{2, e_2} & \alpha_{2, e_3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha_{3, e_6} & 0 & 0 \end{pmatrix},$$

$$B = \begin{pmatrix} \varepsilon_{e_1, 1} & 0 & 0 & 0 & 0 & 0 & 0 & \varepsilon_{e_8, 1} \\ \varepsilon_{e_1, 2} & 0 & 0 & 0 & 0 & 0 & 0 & \varepsilon_{e_8, 2} \\ 0 & 0 & 0 & \varepsilon_{e_4, 3} & 0 & \varepsilon_{e_6, 3} & \varepsilon_{e_7, 3} & 0 \end{pmatrix} \quad \text{dir. } F, A \text{ ve } B \text{ matrislerinin}$$

tanımından tüm ağ bağlantısının dönüşüm matrisi kolayca verilebilir.

• $G = (V, E)$ ağ bağlantısında kaynak düğümü ve alıcı düğümü gösteren bitişiklik matrisi F olmak üzere $F^2, F^3, F^4 \dots$ kuvvetleri ağ bağlantısındaki tüm iletişimi analiz etmeyi sağlar. F , matrisinde $F_{i,k}$ nin i düğümünden k düğümüne ve $F_{k,j}$ nin k düğümünden j düğümüne direk yol sayısını verdiğine dikkat edilirse $F_{i,k} \cdot F_{k,j}$ i düğümünden j düğümüne 2 uzunluğundaki yolların sayısını vermelidir. Sonuç olarak

$F^2 = F \cdot F$ deki (i, j) bileşen $(F)_{i,j}^2 = \sum_{k=1}^n F_{i,k} \cdot F_{k,j}$ i düğümünden j düğümüne 2

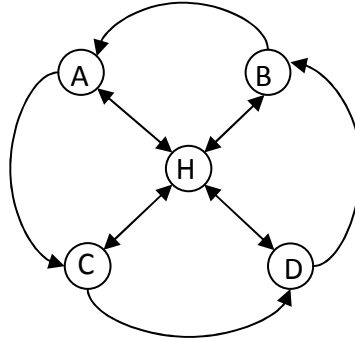
uzunluğundaki yolların sayısını, $(F)_{i,j}^3 = \sum_{k=1}^n F_{i,k} \cdot F_{k,j}$ i düğümünden j düğümüne 3

uzunluğundaki yolların sayısını, genellenirse; $(F)_{i,j}^t = \sum_{k=1}^n F_{i,k_1} \cdot F_{k_1, k_2} \dots F_{k_{t-2}, k_{t-1}} \cdot F_{k_{t-1}, j}$ i

düğümünden j düğümüne t uzunluğundaki yolların sayısını verir. Böylece i düğümünden j düğümüne t uzunluğundan fazla olmayan yolların sayısı;

$$(F)_{i,j} + (F)_{i,j}^2 + (F)_{i,j}^3 + \dots + (F)_{i,j}^t = (F + F^2 + F^3 + \dots + F^t)_{i,j} \quad \text{dir.}$$

Örnek 3.11



Şekil 3. 10 Ağ bağlantılarında n uzunluğundaki yolların sayısı

A dan B ye gitmek istenirse en az 2 yoldan geçilmelidir. ($A \rightarrow H$ ve $H \rightarrow B$). Bu en kısa yoldur. Eğer en az 3 yol kullanılarak gidilmek istenirse matris yardımıyla bu yolların sayısını bulunabilir.

$F_{i,j} = \begin{cases} 1, & \text{eğer } i \text{ den } j \text{ ye ayrıt var ise} \\ 0, & \text{diğer durumlarda} \end{cases}$ olacak şekilde $F = [f_{i,j}]$ bitişiklik matrisi

tanımlansın. Sırası ile A, B, C, H düğümleri satır ve sütun olmak üzere F matrisi

$$F = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix} \text{ dir. Böylece } A \text{ düğümünden } B \text{ düğümüne en fazla 4}$$

uzunluğundaki yoldan geçerek kaç yolla gidilebilir sorusuna cevap: $(F^2)_{1,2} = 1$ olduğu için A düğümünden B düğümüne 2 uzunluğundaki yol sayısı 1, $(F^3)_{1,2} = 3$ olduğu için A düğümünden B düğümüne 3 uzunluğundaki yol sayısı 3, $(F^4)_{1,2} = 7$ olduğu için A düğümünden B düğümüne 4 uzunluğundaki yol sayısı 7, böylece $(F + F^2 + F^3 + F^4)_{1,2} = 0 + 1 + 3 + 7 = 11$ dir. A düğümünden B düğümüne en fazla 4 uzunluğundaki yoldan geçerek 11 yolla gidilebilir

F matrisi kare matris olsun. Eğer $\lim_{n \rightarrow \infty} F^n = 0$ ise $I - F$ terslenebilir ve

$$(I - F)^{-1} = I + F + F^2 + \dots = \sum_{k=0}^{\infty} F^k \text{ dir. Gerçekten;}$$

$(I - F).(I + F + \dots + F^k) = I(I + F + \dots + F^k) - F.(I + F + \dots + F^k) = I - F^{k+1}$ olduğu için her iki taraf $(I - F)$ nin tersi ile çarpılırsa

$$(I + F + \dots + F^k) = (I - F)^{-1} . (I - F^{k+1}) = (I - F)^{-1} . I - (I - F)^{-1} . F^{k+1}$$

$(I + F + \dots + F^k) - (I - F)^{-1} = -(I - F)^{-1} . F^{k+1}$ elde edilir. Buradan $\lim_{n \rightarrow \infty} F^n = 0$ olduğu için

$$(I - F)^{-1} = I + F + F^2 + \dots = \sum_{k=0}^{\infty} F^k \text{ dir.}$$

Teorem 3.4 [7] Bir ağ bağlantısı A, B ve F matrisleri ile verilsin. Ağ bağlantısının dönüşüm matrisi $M = A(I - F)^{-1} B^T$ dir. Burada $I, |E| \times |E|$ boyutlu birim matristir.

İspat: A ve B matrisleri aslında dönüşüm matrisine pek katkıda bulunmazlar. Çünkü girdi ve çıktı rastgele işlemlerinin lineer olarak işleme katılmasına yardımcı olurlar. $X(v, i)$ girdi işlemleri ve $Z(v', j)$ çıktı işlemlerinin arasındaki bağlantının “iletişim cevabını” bulabilmek için $X(v, i)$ rastgele işlemlerinin $Z(v', j)$ ye katkıda bulunabileceği tüm kazanımları tüm yollar boyunca eklemeliyiz. Ağ bağlantısındaki düğümler arasındaki tüm yolların etkisi yukarıda bahsedildiği gibi $I + F + F^2 + \dots$ serisi ile izah edilir. F nilpotent olduğu için $F^N = 0$ olacak şekilde bir N tamsayısı vardır. Böylece, $(I - F)^{-1} = I + F + F^2 + \dots$ olur. Sonuç olarak istenilen sağlanır.

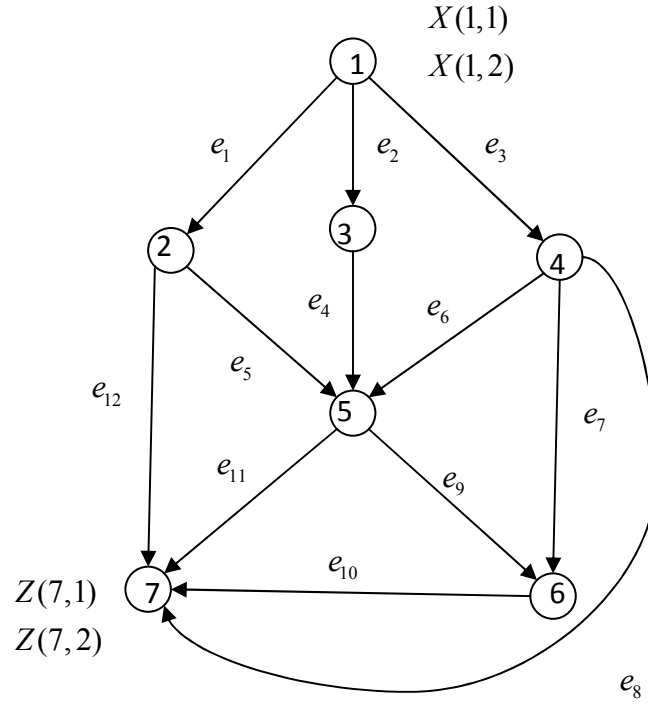
M dönüşüm matrisi $[F_2(\dots, \alpha_{l,e}, \dots, \beta_{e',e}, \dots, \varepsilon_{e',j})]$ polinom halkası üzerinde bir matris olarak düşünülür.

Örnek 3.12: Örnek 3.8 deki devirli ağ bağlantı kodu için bitişiklik matrisi;

$$K = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}, I - K = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} \text{ olup } I - K \text{ matrisinin satırları lineer}$$

bağımlı olduğu için $(I - K)^{-1}$ yoktur. Çünkü; $\det(I - K) = 0$ dir. O halde bu ağ bağlantı kodu için dönüşüm matrisi tanımlanamaz ve çözüm yoktur.

Örnek 3.13



Şekil 3.11 Dönüşüm matrisi verilen ağ bağlantısı

Şekil 3.11 da verilen ağ bağlantısı için dönüşüm matrisini inceleyip. $x = (1,1)$ mesajı gönderilsin. Ağ bağlantısı için

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{ ve}$$

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

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

bitişiklik matrisidir. O halde

$$(I - F)^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

olur. Burada I , 12×12 boyutlu birim matristir. O halde M dönüşüm matrisi

$M = A(I - F)^{-1} B^t$ olmak üzere $M = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ dir. Sonuç olarak $x = (1,1)$ mesajı

$(1,1) \cdot \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = (1,1)$ olarak alınır.

3.3.2 Genel Ağ Bağlantı Kodlama Problemi

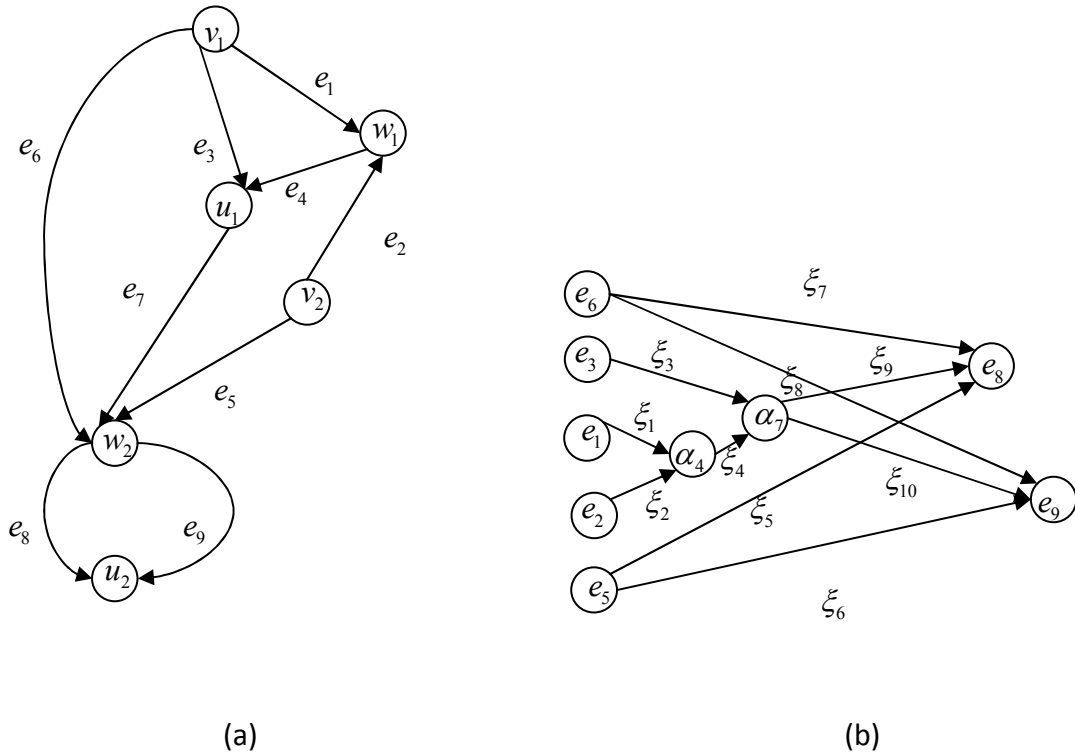
Bir G ağ bağlantısı ve $\mathfrak{S}$ bağlantıların keyfi kümesi verildiğinde tüm bağlantıların sağlanabilmesi için ağ bağlantısına çözüm bulma problemi tek kaynaklı ve tek alıcı

düğümlü ağ bağlantıları için çözüm bulmaktan daha zordur. Bu kısımda özellikle ağ bağlantısının her bir bağlantı için Min-Cut Max-Flow sınırını sağlaması ve bağlantıların birbirini rahatsız etmemesi arzulanmaktadır. Ayrıca dönüşüm matrisinin özellikle istenilen bağlantılar için ifade edilmesinin yeterli olacağı gösterilmektedir. Aşağıdaki teorem genel ağ bağlantısı kodlama problemi için gerekenleri ifade etmek için verilmiştir.

Teorem 3.5 [7] G gecikmesiz bir ağ bağlantısı olsun ve

$\mathfrak{T} = \{(v, u_1, X(v)), (v, u_2, X(v)), \dots, (v, u_N, X(v))\}$ istenilen bağlantı olsun. $(G, \mathfrak{T})$ ağ bağlantı problemi çözülebilirdir ancak ve ancak Min-Cut Max-Flow sınırı tüm $\mathfrak{T}$ bağlantıları için sağlanmalıdır.

Örnek 3.14 [7]



Şekil 3. 12 İki kaynaklı iki alıcı düğümlü ağ bağlantısı, uygun işaretlenmiş çizgi graf

Bu ağ bağlantısı üzerindeki iki farklı

$$\mathfrak{T} = \{(v_1, u_1, \{X(v_1, 1), X(v_1, 2)\}), (v_2, u_2, \{X(v_2, 1), X(v_2, 2)\})\}$$

bağlantıları inşa edilmek istensin. Burada $\underline{x} = \{X(v_1,1), X(v_1,2), X(v_2,1), X(v_2,2)\}$ ve $\underline{z} = \{Z(u_1,1), Z(u_1,2), Z(u_2,1), Z(u_2,2)\}$ vektörleri kaynak ve alıcı düğümlerdeki işlemleri temsil eden vektörler olsun. Böylece sistem matrisi M ;

$$M = \begin{pmatrix} \xi_{11} & \xi_{12} & \xi_{13} & 0 & 0 \\ \xi_{14} & \xi_{15} & \xi_{16} & 0 & 0 \\ 0 & 0 & 0 & \xi_{17} & \xi_{18} \\ 0 & 0 & 0 & \xi_{19} & \xi_{20} \end{pmatrix} \cdot \begin{pmatrix} 0 & \xi_1 & \xi_1 \cdot \xi_4 \cdot \xi_9 & \xi_1 \cdot \xi_4 \cdot \xi_{10} \\ 1 & 0 & \xi_3 \cdot \xi_9 & \xi_3 \cdot \xi_{10} \\ 0 & 0 & \xi_7 & \xi_8 \\ 0 & \xi_2 & \xi_2 \xi_4 \xi_9 & \xi_2 \xi_4 \xi_{10} \\ 0 & 0 & \xi_5 & \xi_6 \end{pmatrix} \cdot \begin{pmatrix} \xi_{21} & \xi_{22} & 0 & 0 \\ \xi_{23} & \xi_{24} & 0 & 0 \\ 0 & 0 & \xi_{25} & \xi_{26} \\ 0 & 0 & \xi_{27} & \xi_{28} \end{pmatrix}$$

şeklinde olduğu kolayca görülebilir. Bu matris $M = \begin{pmatrix} M_{1,1} & M_{1,2} \\ M_{2,1} & M_{2,2} \end{pmatrix}$ şeklinde blok

matrisler olarak ifade edilebilir. Burada $M_{1,1}, (X(v_1,1), X(v_1,2))$ ve $((Z(u_1,1), Z(u_1,2)))$ arasındaki $M_{1,2}, (X(v_1,1), X(v_1,2))$ ve $(Z(u_2,1), Z(u_2,2))$, $M_{2,1}, (X(v_2,1), X(v_2,2))$ ve $(Z(u_1,1), Z(u_1,2))$ son olarak $M_{2,2}, (X(v_2,1), X(v_2,2))$ ve $(Z(u_2,1), Z(u_2,2))$ arasındaki dönüşümü temsil eder. Bu ağ bağlantı kodlamasının çözümü olabilmesi için $\det(M) \neq 0$ olmalıdır. Özellikle ağ bağlantısında verilen bağlantıların sağlanabilmesi için $\det(M_{1,1}) \neq 0$ ve $\det(M_{2,2}) \neq 0$ olmalıdır. Öte yandan bağlantıların birbirine rahatsız etmemesi için $M_{1,2}$ ve $M_{2,1}$ nün tüm elemanlarının sıfır seçilmesi gerekir. Gerçekten $\det(M_{1,1}) = (\xi_{11}\xi_{15} - \xi_{12}\xi_{14})\xi_1(\xi_{21}\xi_{24} - \xi_{22}\xi_{23})$

ve $\det(M_{2,2}) = \xi_2\xi_4(\xi_{17}\xi_{20} - \xi_{18}\xi_{19})(\xi_9\xi_6 - \xi_5\xi_{10})(\xi_{25}\xi_{28} - \xi_{26}\xi_{27})$ dir. Öte yandan $M_{2,1}$ in sıfır olması için $\xi_2 = 0$ olmalıdır bu ise $\det(M_{2,2}) = 0$ olmasını gerektirir. O halde bu şekilde istenilen bağlantılar için çözüm bulunamaz. Yani istenilen ağ bağlantısı $(G, \mathfrak{I})$ ağ bağlantı kodlama problemi için bir çözüm sunmaz.

Teorem 3.6 [7] $(G, \mathfrak{I})$ devirsiz, gecikmesiz bir lineer ağ bağlantı kodlama problemi ve $M = (M_{i,j})$ girdi düğümlerinin kümesi ile alıcı düğümlerinin kümesi arasındaki dönüşüm matrisi olsun. Ağ bağlantı problemi çözülebilirdir ancak ve ancak $F_2(\dots, \alpha_{l,e}, \dots, \beta_{e',e}, \dots, \varepsilon_{e',j})$ üzerindeki değişkenler şu iki şartı sağlamalıdır:

1. $(v_i, v_j, X(v_i, v_j)) \notin \mathfrak{I}$ olan tüm (v_i, v_j) düğüm ikilileri için $M_{i,j} = 0$ dir.

2.Eğer $\mathfrak{S}$ kümesi $\{((v_{i_1}, v_{j_1}, X(v_{i_1}, v_{j_1})), ((v_{i_2}, v_{j_2}, X(v_{i_2}, v_{j_2}))), \dots, ((v_{i_l}, v_{j_l}, X(v_{i_l}, v_{j_l})))\}$ bağlantılarını içeriyor ise $[M'_{i_1, j_1} M'_{i_2, j_2} \dots M'_{i_l, j_l}]$ alt matrisi $\mathcal{G}(v_j) \times \mathcal{G}(v_j)$ boyutlu terslenebilir matris olmalıdır.

İspat: Kabul edelim ki ağ bağlantı problemi çözülebilir olsun. 1. Şart bağlantıların birbirini rahatsız etmemesini ve değişkenlere atanan değerler bağlantılara engel olmamasını garanti eder. Ayrıca 2. şart herhangi bir düğümün bilgiyi doğru tanımasını garanti eder. Aksine kabul edelim ki iki şartta sağlanmasın. Eğer 1. şart sağlanmazsa rastgele işlemleri alıcı düğümlerde üst üste aktarılır ve bilgi doğru aktarılamaz. Öte yandan 2. şart Min-Cut Max-Flow şartını sağlar. Doğru olmazsa çözüm olamaz.

Örnek 3.15 Şekil 3.7 deki ağ bağlantısı için;

$$F_8 = F_2[x] / \langle x^3 + x + 1 \rangle = \{a\alpha^2 + b\alpha + c : a, b, c \in F_2, \alpha^3 + \alpha + 1 = 0\} = \{0 + \langle x^3 + x + 1 \rangle, 1 + \langle x^3 + x + 1 \rangle, x + \langle x^3 + x + 1 \rangle, x + 1 + \langle x^3 + x + 1 \rangle, x^2 + \langle x^3 + x + 1 \rangle, x^2 + 1 + \langle x^3 + x + 1 \rangle, x^2 + x + \langle x^3 + x + 1 \rangle, x^2 + x + 1 + \langle x^3 + x + 1 \rangle\}$$

sonlu cismi üzerinde bir çözüm bulalım. $x^3 + x + 1$, α' 'yı kök kabul etsin. Bu sonlu cisim üzerinde

$$M = \begin{pmatrix} \alpha & 1 & 0 \\ 0 & 1 & \alpha \\ \alpha + 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \beta_{e_1, e_5} & \beta_{e_1, e_4} \beta_{e_4, e_6} & \beta_{e_1, e_4} \beta_{e_4, e_7} \\ \beta_{e_2, e_5} & \beta_{e_2, e_4} \beta_{e_4, e_6} & \beta_{e_2, e_4} \beta_{e_4, e_7} \\ 0 & \beta_{e_3, e_6} & \beta_{e_3, e_7} \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ biçiminde } A \text{ ve } B$$

matrislerine değerler verildiğinde ve bitişiklik matrisinin elemanları sıra ile

$$\begin{aligned} \beta_{e_1, e_4} &= \alpha^2, \beta_{e_1, e_5} = \alpha, \beta_{e_2, e_4} = \alpha + 1, \beta_{e_2, e_5} = \alpha^2 + \alpha \\ \beta_{e_3, e_6} &= \alpha^2 + \alpha + 1, \beta_{e_3, e_7} = \alpha, \beta_{e_4, e_6} = \alpha + 1, \beta_{e_4, e_7} = \alpha^2 + \alpha, \end{aligned} \quad \text{diğer değerler } 0 \text{ olacak}$$

biçimde seçildiğinde

$$\begin{aligned} \beta_{e_1, e_4} \beta_{e_4, e_6} &= \alpha^2 \cdot (\alpha + 1) = \alpha^3 + \alpha^2 = \alpha^2 + \alpha + 1, \quad \beta_{e_1, e_4} \beta_{e_4, e_7} = \alpha^2 \cdot (\alpha^2 + \alpha) = \alpha^4 + \alpha^3 = \alpha^2 + 1, \\ \beta_{e_2, e_4} \beta_{e_4, e_6} &= (\alpha + 1) \cdot (\alpha + 1) = \alpha^2 + 1, \quad \beta_{e_2, e_4} \beta_{e_4, e_7} = (\alpha + 1) \cdot (\alpha^2 + \alpha) = \alpha + \alpha + 1 = 1, \end{aligned}$$

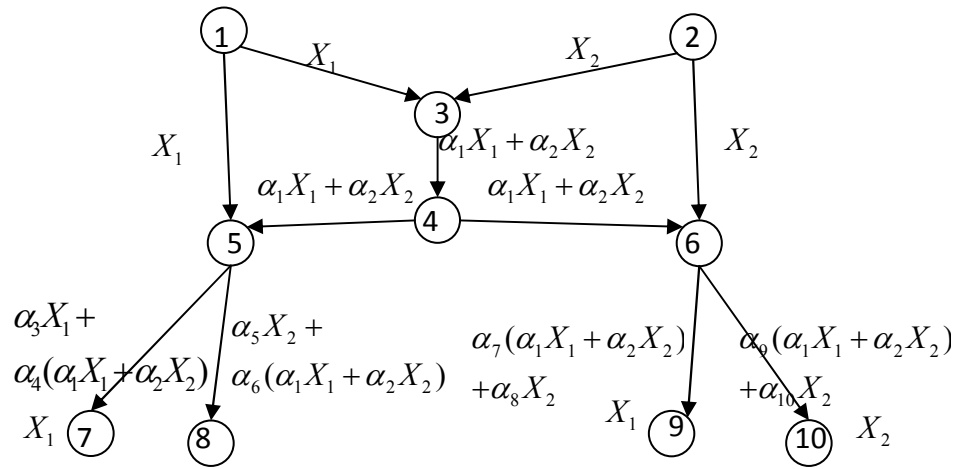
$$M = \begin{pmatrix} \alpha & 1 & 0 \\ 0 & 1 & \alpha \\ \alpha + 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \alpha & \alpha^2 + \alpha + 1 & \alpha^2 + 1 \\ \alpha^2 + \alpha & \alpha^2 + 1 & 1 \\ 0 & \alpha^2 + \alpha + 1 & \alpha \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \alpha & 0 & 0 \\ 0 & 0 & \alpha^2 + 1 \\ \alpha^2 + \alpha & \alpha & \alpha^2 \end{pmatrix}$$

ve $\det(M) = \alpha \neq 0$ olduğu için M dönüşüm matrisi ile iletişim sağlanır ve alıcı düğümler hangi mesajın gönderilmiş olduğu belirleyebilir.

Bir ağ bağlantısının her zaman lineer çözümü olmak zorunda değildir. Bir ağ bağlantısında alıcılar almak istedikleri verileri karşılayabiliyor ise ağ bağlantısı işlevini yerine getirmiş olur. Kurulan ağ bağlantısı alıcılar talep ettikleri verileri alacak şekilde çözüme sahip olduğunda çözülebilir denir.

Aşağıda verilen ağ bağlantısının $GF(4) = F_2[x] / \langle x^2 + x + 1 \rangle = \{0, 1, \alpha, \alpha^2 = \alpha + 1\}$ cismi üzerinde bir çözüme sahip olup olmadığını inceleyelim. Bu ağ bağlantısında iki kaynak ve dört alıcı düğüm vardır. Kaynak düğümler şekilde gösterildiği gibi 1 ve 2 olmak üzere sırası ile X_1, X_2 mesajlarını gönderirken alıcı düğümler şekilde gösterildiği gibi 7, 8, 9 ve 10 düğümleridir ve sıra ile X_1, X_2, X_1 ve X_2 mesajlarını talep ederler.

Örnek 3.16



Şekil 3. 13 Çözüme sahip ağ bağlantısı

Eğer ağ bağlantısı lineer ise ayrıtlar üzerindeki işlemler şekilde gösterildiği gibi tanımlanmalıdır. $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \in GF(4)$ olacak şekilde bu elemanları belirlemeye çalışalım. Sırası ile ayrıtlar üzerindeki işlemlerden 57 ayrıtı üzerindeki işlemde 7 düğümü X_1 mesajını talep ettiği için $\alpha_3 + \alpha_4 \alpha_1 = 1$ ve $\alpha_2 \alpha_4 = 0$ dir. Benzer şekilde sırası ile 8, 9 ve 10 düğümlerindeki taleplerden dolayı $\alpha_5 + \alpha_1 \alpha_6 = 0$, $\alpha_2 \alpha_6 = 1$,

$\alpha_2\alpha_7 + \alpha_8 = 0$, $\alpha_2\alpha_7 = 1$, $\alpha_1\alpha_9 = 0$, $\alpha_{10} + \alpha_2\alpha_9 = 1$ olur. $GF(4)$ cismi üzerinde

$$\alpha_1 = \alpha_4 = \alpha_5 = \alpha_9 = 0, \quad \alpha_2 = \alpha_3 = \alpha_6 = \alpha_7 = \alpha_8 = \alpha_{10} = 1$$

ya da $\alpha_1 = \alpha_2 = \alpha_5 = \alpha_6 = \alpha_7 = \alpha_{10} = 1, \alpha_3 = \alpha, \alpha_4 = \alpha + 1, \alpha_8 = 1, \alpha_9 = 0$, bu ağ bağlantısı için bir çözümdür ve bu şekilde pek çok çözüm bulunabilir.

HATA DÜZELTEBİLEN AĞ BAĞLANTI KODLARI

Aşağıdaki kısımda ağ bağlantı kodları üzerinde hata belirleme ve düzeltme ile ilgili genel tanım ve teoremleri vereceğiz. Bu bölümdeki genel tanım ve teoremleri ifade etmek için “Weight Properties of Network Codes” [9] isimli makaleden faydalanılmıştır.

4.1 Genel Hata Düzeltilebilen Ağ Bağlantıları

Tanım 4.1 Genelliği bozmaksızın ağ bağlantılarında S kaynak düğümü için $in(S) = \emptyset$ dir. $n_S = |out(S)|$ olsun. S kaynak düğümü rastgele girdi işlemlerini $x = [x_e; e \in out(S)] \in F^{n_S}$ i satır vektörlerine enkodlar. Bu satır vektörlerine kodsözler denir. Tüm kodsözlerin kümesine kod denir ve C ile gösterilir. S kaynak düğümü kodsözlerin n_S bileşenini $out(S)$ deki ayrıtlar ile dönüştürülerek iletir.

Tanım 4.2 Bir z hata vektörü F cismi üzerindeki $|E|$ boyutlu satır vektördür öyle ki i . bileşeni E ayrıt kümesindeki i . ayrıt üzerindeki hatayı temsil eder. Bir ağ bağlantısı üzerindeki tüm hata vektörlerinin kümesi Σ ile gösterilsin.

Tanım 4.3 Bir hata şekli; E ayrıt kümesin herhangi alt kümesidir ve ρ ile gösterilir.

Tanım 4.4 Eğer tüm hatalar bir hata şeklindeki ayrıtlar üzerinde oluyorsa hata vektörüne hata şekli ile eşleşir denir. Bir ρ hata şekli ile eşleşen tüm hata vektörlerinin kümesi ρ^* ile gösterilir. ρ_z ile bir z hata vektörünün sıfırdan farklı bileşen yerleri gösterilsin.

x kodsözleri ve z hata vektörleri düşünölsün. Herhangi $x \in C$ kodsözü ve $z \in \Sigma$ hata vektörü için $F(x, z)$ alınan vektördür. Burada $F: C \times \Sigma \rightarrow \Phi$ dönüşümüne transfer

dönüşümü denir. F iletişim kanallarını modeller. Ayrıca bir hata vektörü için ağırlık ölçüsü; $w: \Sigma \rightarrow Z^*$ dönüşümü ile tanımlanır. Burada Z^* kümesi negatif olmayan tamsayıların kümesidir. F_e ve $\overline{F_e}$ sırası ile e ayrıtının girdi ve çıktı işlemleri ve ayrıt üzerindeki hata vektörü z_e olmak üzere; $\overline{F_e} = F_e + z_e$ eşitliği sağlanır. Herhangi ρ ayrıtlarının kümesi için iki satır vektörünü $F_\rho = [F_e : e \in \rho]$ ve $\overline{F_\rho} = [\overline{F_e} : e \in \rho]$ şeklinde tanımlayalım. Diğer bir ifade ile bir ağ bağlantı kodu; G ağ bağlantısı üzerindeki $C \subseteq F^{n_s}$ kodu ve $\{\overline{\beta_e} : e \in E \setminus out(S)\}$ yerel enkodlama fonksiyonlarının bir kümesidir. Burada $\overline{F_e} = \overline{\beta_e}(F_{in(çıkış(e))})$ olacak şekildeki $\overline{\beta_e} : F^{|in(çıkış(e))|} \rightarrow F$ dir. Ağ bağlantısı üzerindeki iletişim ayrıtların belli kısmi sıralı kümesi üzerinde yukarıdan aşağıya doğru gerçekleştirilir. Bir x kodsözü ve z hata vektörü için bir ağ bağlantı kodu bir kez $\overline{F_e} = F_e + z_e$ ve $\overline{F_e} = \overline{\beta_e}(F_{in(çıkış(e))})$ dönüşümleri sağlanırsa tüm ayrıtlar için $\overline{F_e}$ $\overline{F_{out(S)}} = x$ sınırını sağlayacak şekilde sırası ile hesaplanabilir.

4.2 Lineer Ağ Bağlantı Kodlarının Hata Düzeltme Kapasitesi

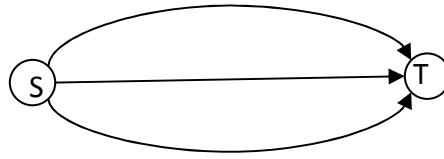
Bölüm 3 te lineer ağ bağlantı kodlarının tanımını vermiştik. Bu bölümde alfabe F sonlu cismidir. Eğer $\overline{\beta_e}$ tüm $e \in E - out(S)$ için lineer fonksiyon ise yani $\overline{F_e} = \sum_{e' \in E} \beta_{e',e} F_{e'}$ ise ağ bağlantı koduna lineerdir denir. Burada $\beta_{e',e}$, e' den e ye yerel kodlama dönüşümünü ifade eder. Sadece $e' \in in(çıkış(e))$ ise $\beta_{e',e}$ yerel kodlama çekirdeği sıfırdan farklı olur. Bölüm 3 deki bir bitişiklik matrisini $K = [K_{i,j}]$ notasyonu ile ifade edelim. Buradaki tanımdan $K_{i,j} = \beta_{e_i,e_j}$ ve $F = I + K + K^2 + \dots$ ağ bağlantısı devirsiz olduğunda bir pozitif N tamsayısı için $K^N = 0$ olduğu için $F = (I - K)^{-1}$ olduğunu biliyoruz. Ayrıtların herhangi bir ρ kümesi için $|\rho| \times E$ boyutlu $A_\rho = [A_{i,j}]$ matrisi $A_{i,j} = \begin{cases} 1, & e_j, \rho \text{ deki } i. \text{ ayrıt ise} \\ 0, & \text{diğer} \end{cases}$ olacak şekilde tanımlansın. E' de ki sırayı ρ' ye uygulayınca $A_{\rho'}$ nin $|\rho|$ sütunu birim matrisi oluşturur. Notasyonu basitleştirmek için $F_{\rho,\rho} = A_\rho F A_\rho^t$ olarak alındığında x kodsözleri ve z hata vektörleri için çıktı

$F_\rho(x, z) = (x.A_{out(S)} + z)FA_\rho^t = xF_{out(S),\rho} + zFA_\rho^t$ olarak hesaplanır. Herhangi bir T düğümü için $F_{in(T)} = F_T$ yazarak $F_T(x, z) = xF_{S,T} + zF_T$ dir. Burada $F_{S,T} = F_{out(S),in(T)}$ ve $F_T = FA_{in(T)}^t$ dir. Bu matrisler sırası ile mesaj ve hata transfer matrisleridir.

Tanım 4.5 w ağırlıklı x kodsözü ve c tamsayısı için;

$\Phi_w(x, c) = \{F(x, z) : z \in \Sigma, w(z) \leq c\}$ kümesini tanımlayalım. Bu küme c ya da az hata ile oluşabilecek hatalar için alınan vektörlerin kümesini ifade eder.

Örnek 4.1



Şekil 4. 1 Hatalı ağ bağlantısında kodlama

$$A_{out(S)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, A_{in(T)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ ve } K = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \text{ olduğu için } F = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

elde edilir. Buradan $x = (x_1, x_2, x_3), z = (z_1, z_2, z_3) \in F_2^3$ olmak üzere

$$\begin{aligned} F_T(x, z) &= (xA_{out(S)} + z)FA_{in(T)} \\ &= ((x_1, x_2, x_3) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + (z_1, z_2, z_3) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= ((x_1, x_2, x_3) + (z_1, z_2, z_3)) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (x_1, x_2, x_3) + (z_1, z_2, z_3) \end{aligned}$$

olarak elde edilir. Böylece ağ bağlantı kodlama problemi klasik kodlama problemine dönüşür. Sonuç olarak klasik hata düzeltme problemi lineer ağ bağlantısı hata düzeltme probleminin özel bir halidir. Mesela $C = \{(0, 0, 0), (1, 1, 1)\}$ kodu ve $z = \{(0, 1, 0)\}$ olmak üzere alınan vektörler

$$F_T((0, 0, 0), (0, 1, 0)) = ((0, 0, 0) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + (0, 1, 0) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (0, 1, 0)$$

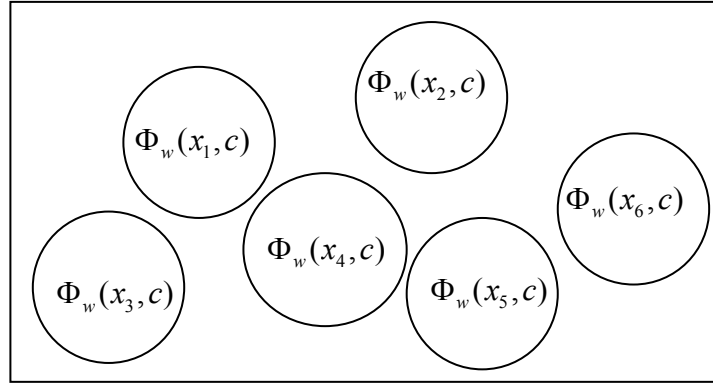
$$F_T((1,1,1),(0,1,0)) = ((1,1,1) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + (0,1,0) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (1,0,1) \text{ olur. Bir}$$

ağ bağlantısı üzerindeki kodun bir kodsözü seçilir ve ağ bağlantısı üzerindeki kanallardan iletilir.

Tanım 4.6 Bir koddaki kodsöz için Φ' nin bir alt kümesi kodsöz için dekodlama bölgesi olarak adlandırılır. Bir ağ bağlantı kodunun dekodlanabilmesi için tüm kodsözleri için dekodlama bölgelerinin ayrık olması gerekir. Eğer alınan bir vektör bir kodsözün dekodlama bölgesinde ise o vektör dekodlanabilir yani; hata düzeltilebilir. Eğer alınan vektör hiçbir kodsözün dekodlama bölgesinde değil ise alınan vektörün hangi mesajdan elde edildiği belirlenemez yani; hata düzeltilemez. Ayrıca eğer alınan herhangi vektör herhangi bir kodsözle aynı değil ise hata belirlenmiş olur.

Tanım 4.7 w ile ilgili minimum dekodlayıcı I , MWD_w^I ile gösterilir ve alınan bir y vektörünü şu şekilde dekodlar: öncelikle $F(x,z) = y$ olacak biçimdeki tüm $x \in C$ ve $z \in \Sigma$ değişkenleri belirlenir. Eğer (x,z) ikilisine $F(x,z) = y$ eşitliğini sağlıyorsa x' e mesaj kısmı ve z' ye hata kısmı olmak üzere çözüm denir. Ayrıca $w(z)$ tüm çözümleri sağlayacak şekilde minimum ağırlıklı hata ise bu çözüme minimum ağırlıklı çözüm denir. Eğer tüm minimum ağırlıklı çözümlerin mesaj kısımları aynı ise alınan mesajlar dekodlanabilir. Aksi olursa hata belirlenir ama düzeltilemez. Yukarıda bahsedildiği gibi $\{\Phi_w(x,c) : x \in C\}$ ayrık ise minimum dekodlayıcı bölgesi olarak adlandırılır. Ayrıca $\Phi_w(x,c)$ dekodlama yuvarı olarak da adlandırılabilir.

Tanım 4.8 Tüm $x \in C$ için, $\Phi_w(x,c)$ kümeleri boş değil ve ayrık ise w ile ilgili minimum ağırlıklı dekodlayıcı II , MWD_w^{II} ile gösterilir ve alınan bir y vektörünü şu şekilde dekodlar; $F(x,z) = y$ olacak biçimdeki tüm $x \in C$ değişkenleri belirlenir ve bu değişkenlere dekodlanan mesajlar denir. Eğer y herhangi bir yuvarın içinde değil ise dekodlanamaz denir. Eğer bir kodsözü boş bir kodlama bölgesi ile iletilmişse alınan vektör hiçbir zaman dekodlanamaz.



Şekil 4. 2 c ağırlıklı hata yuvarları

Tanım 4.9 Herhangi $x_1, x_2 \in C$ kodsözü için x_1 ve x_2 arasındaki uzaklık $D_w^0(x_1, x_2) = \min\{c_1 + c_2 : |c_1 - c_2| \leq 1, \Phi(x_1, c_1) \cap \Phi(x_2, c_2) \neq \emptyset\}$ olarak ifade edilir. Her zaman iyi tanımlıdır. Dikkat edilirse bu uzaklık metrik değil fakat simetriktir.

Tanım 4.10 Herhangi $x_1, x_2 \in C$ kodsözü için x_1 ve x_2 arasındaki minimum uzaklık $d_{\min, w}^0 = \min\{D_w^0(x_1, x_2) : x_1 \neq x_2 \in C\}$ dir ve minimum uzaklık olarak adlandırılır.

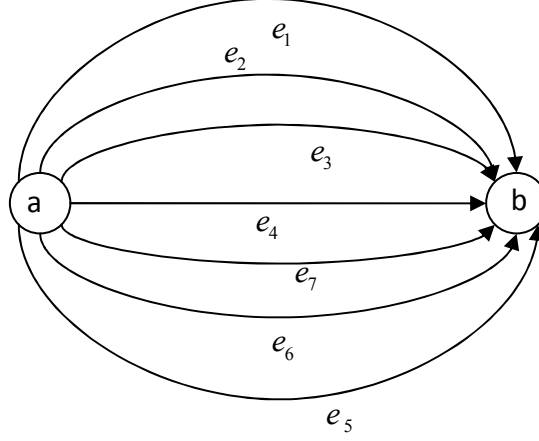
Tanım 4.11 Bir kodda $w(z) \leq c$ olacak şekildeki tüm z hata vektörleri MWD_w^{II} ile düzeltilebilir ise bu koda c hata düzeltilebilir denir.

Lemma 4.1 [9] Bir kod c hata düzeltilebilir ancak ve ancak $MWD_w^{II}(c)$ vardır.

İspat: Kabul edelim ki herhangi bir kod için $MWD_w^{II}(c)$ var olsun. Yani; $\Phi_w(x, c) \neq \emptyset$ ve ayrık olsunlar. x gönderilen mesaj, z hata vektörü için $w(z) \leq c$ ve alınan vektör $y = F(x, z)$ olsun. $y = F(x, z)$ için minimum ağırlıklı çözüm (x^*, z^*) olsun. $\Phi_w(x, c) \neq \emptyset$ olduğu için böyle bir çözüm vardır. $MWD_w^{II}(c)$ yuvarları ayrık olduğu için herhangi iki $x \neq x'$ kodsözü ve $w(z') \leq c$ için $F(x', z') \neq y$ dir. $w(z^*) \leq w(z) \leq c$ ve $F(x^*, z^*) = y$, $x^* = x$ olur. Sonuç olarak $y = F(x, z)$ nin minimum ağırlıklı çözümünün mesaj kısmı x dir. Bu ise $MWD_w^I(c)$ nin tüm doğru mesajlarını sunar. Sonuç olarak tam c hata düzeltilebilir. Aksine $MWD_w^{II}(c)$ var olmasın. O halde bazı x_0 için ya $\Phi_w(x_0, c) = \emptyset$ ya da $\{\Phi_w(x, c) : x \in C\}$ ayrık değildir. Birinci durumda $MWD_w^I(c)$ x_0 ' ı doğru dekodlayamaz. İkinci durumda ise $x_1 \neq x_2 \in C$ ve $w(z_1) \leq c$, $w(z_2) \leq c$ olacak şekilde z_1 ve z_2 bulunabilir öyle ki $F(x_1, z_1) = F(x_2, z_2)$ olur. Eğer

$F(x_1, z_1) = F(x_2, z_2) = y$ ise $MWD_w^I(c)$ her zaman doğru dekodlama yapamaz yani; c hata düzeltilemez. Bu ise kabul ile çelişir.

Örnek 4.2



Şekil 4. 3 $[7, 4, 3]_2$ Hamming kod için ağ bağlantısı

$C = \{0000000, 1000110, 0100101, 1100011, 0010011, 1010101, 0110110, 1110000, 0001111, 1001001, 01010101, 1101100, 0011100, 1011010, 0111001, 1111111\}$

$[7, 4, 3]$ Hamming kod olsun. Tüm kodsözler için $MDW^II(1)$ var ve ayrıktır.

Gerçekten; gönderilen mesajlar için 1 hata ağırlığı ile oluşan yuvarlarda;

$x = (x_1, x_2, x_3, x_4, x_5, x_6, x_7) \in C$ ve $z = (z_1, z_2, z_3, z_4, z_5, z_6, z_7) \in Z_2^7$ olmak üzere

$$A_{out(a)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad A_{in(b)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$K = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{ ve } F = (I - K)^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \text{ sistem}$$

dönüşüm matrisi için alınan sözler;

$$\begin{aligned} F_T(x, z) &= (xA_{out(S)} + z)FA_{in(T)} \\ &= ((x_1, x_2, x_3, x_4, x_5, x_6, x_7) \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} + \\ &\quad (z_1, z_2, z_3, z_4, z_5, z_6, z_7) \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \\ &= ((x_1, x_2, x_3, x_4, x_5, x_6, x_7) + (z_1, z_2, z_3, z_4, z_5, z_6, z_7)) \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

$= (x_1, x_2, x_3, x_4, x_5, x_6, x_7) + (z_1, z_2, z_3, z_4, z_5, z_6, z_7)$ şeklinde hesaplanır ve yuvarlar ayrık

olur.

$$\Phi(0000000, 1) = \{0000000, 1000000, 0100000, 0010000, 0001000, 0000100, 0000010, 0000001\}$$

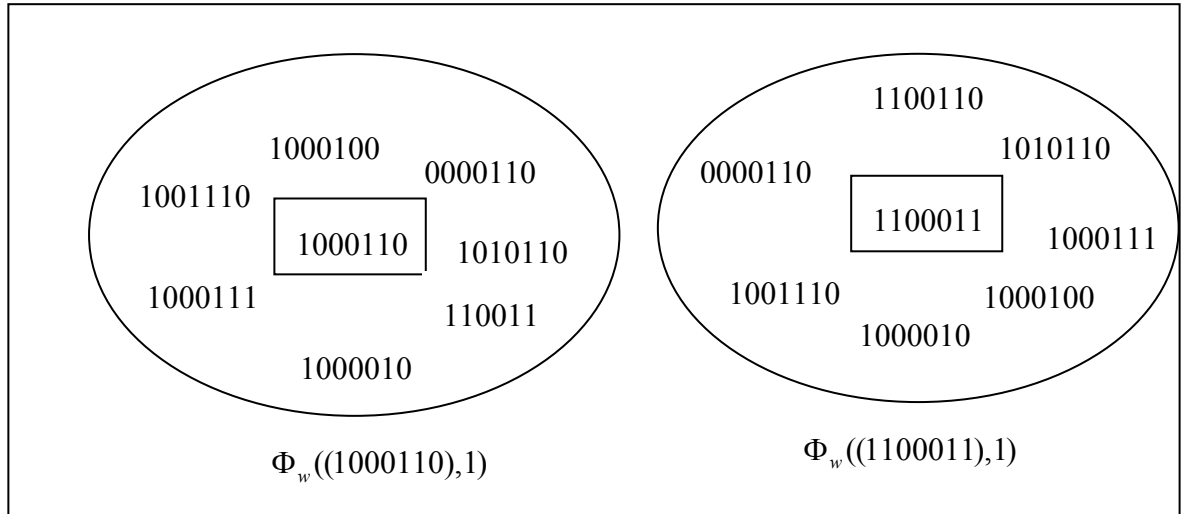
$$\Phi(1000110, 1) = \{1000110, 0000110, 1100110, 1010110, 1001110, 1000010, 1000100, 1000111\}$$

$$\Phi(0100101, 1) = \{0100101, 1100101, 0000101, 0110101, 0101101, 0100001, 0100111, 0100100\}$$

$$\Phi(0010011, 1) = \{0010011, 1010011, 0110011, 0000011, 0011011, 0010111, 0010001, 0010010\}$$

$\Phi(0001111,1) = \{0001111, 1001111, 0101111, 0011111, 0000111, 0001011, 0001101, 0001110\}$
 $\Phi(1100011,1) = \{1100011, 0100011, 1000011, 1110011, 1101011, 1100111, 1100001, 1100010\}$
 $\Phi(1010101,1) = \{1010101, 0010101, 1110101, 1000101, 1011101, 1010001, 1010111, 1010100\}$
 $\Phi(1001001,1) = \{1001001, 0010101, 1110101, 1000101, 1011101, 1010001, 1010111, 1010100\}$
 $\Phi(0110110,1) = \{0110110, 1110110, 0010110, 0100110, 0111110, 0110010, 0110100, 0110111\}$
 $\Phi(0101010,1) = \{0101010, 1101010, 0001010, 0111010, 0100010, 0101110, 0101000, 0101011\}$
 $\Phi(0011100,1) = \{0011100, 1011100, 0111100, 0001100, 0010100, 0011000, 0011110, 0011101\}$
 $\Phi(1110101,1) = \{1110101, 0110101, 1010101, 1100101, 1111101, 1110001, 1110111, 1110100\}$
 $\Phi(1101100,1) = \{1101100, 0101100, 1001100, 1111100, 1100100, 1101000, 1101110, 1101101\}$
 $\Phi(1011010,1) = \{1011010, 0011010, 1111010, 1001010, 1010010, 1011110, 1011000, 1011011\}$
 $\Phi(0111001,1) = \{0111001, 1111001, 0011001, 0101001, 0110001, 0111101, 0111011, 0111000\}$
 $\Phi(1111111,1) = \{1111111, 0111111, 1011111, 1101111, 1110111, 1111011, 1111101, 1111110\}$

olmak üzere her $x, x' \in C$ için $\Phi(x,1) \cap \Phi(x',1) = \emptyset$ dir. Örneğin; $x = 1100011$ ve $x = 1000110$ kodsözleri ve 1 ağırlıklı hatalar için yuvarlar;



Şekil 4. 4 Ayrık yuvarlar

şeklinde elde edilir. Kabul edelim ki alınan vektör $F(x,z) = 0111000$ olsun. O halde bu kodsözü dekodlayabilmek için $F(x,z) = x + z$ sisteminin tüm çözümlerini bulmalıyız.

$MDW''(1) \neq \emptyset$ ve ayrık olduğu (x,z) ikilisi için tek çözüm vardır. $x = 0111001$ ve $z = 0000001$ dir. Yani hata e_7 ayırıtında yapılmıştır. Dolayısıyla; 0111000 alınan vektör 0111001 olarak dekodlanabilir.

Teorem 4.1 [9] Bir kod c hata düzeltebilir ancak ve ancak $d_{\min,w}^0 \geq 2c + 1$ dir.

İspat: Teoremin ispatını aşağıda verilen ifadelerin eşitliğini göstererek yapacağız.

1. $MWD_w''(c)$ var olsun. Yani; $x_1 \neq x_2 \in C$ için $\Phi(x_1, c_1) \cap \Phi(x_2, c_2) = \emptyset$ dir.

2. Herhangi $x_1 \neq x_2 \in C$ için ve $\Phi(x_1, c_1) \cap \Phi(x_2, c_2) \neq \emptyset$ olacak biçimdeki c_1 ve c_2 için ya $c_1 > c$ ya da $c_2 > c$ dir.

3. Herhangi $x_1 \neq x_2 \in C$ için eğer c_1 ve c_2 $|c_1 - c_2| \leq 1$ ve $\Phi(x_1, c_1) \cap \Phi(x_2, c_2) \neq \emptyset$ şartlarını sağlıyor ise $\max(c_1, c_2) > c$ dir.

4. Herhangi $x_1 \neq x_2 \in C$ için eğer c' , $\Phi(x_1, c') \cap \Phi(x_2, c') \neq \emptyset$ ise $c' > c$ dir.

5. $d_{\min, w}^0 \geq 2c + 1$ dir.

İspat şu sıra ile yapılır: $1) \Leftrightarrow 2) \Rightarrow 3) \Rightarrow 4) \Rightarrow 2)$ ve $3) \Leftrightarrow 5)$. $1) \Rightarrow 2)$ olduğunu olmayan ergi ile ispatlayacağız. Eğer 2) doğru değilse $x_1 \neq x_2 \in C$ için $\Phi(x_1, c_1) \cap \Phi(x_2, c_2) \neq \emptyset$ olacak biçimde $c_1 \leq c$ ve $c_2 \leq c$ var olsun. O halde 1) yanlış olur. Tersine de benzer şekilde çelişki ile elde edilir. 3), 2)nin özel bir halidir sadece $|c_1 - c_2| \leq 1$ şartı eklenmelidir. Ayrıca 4), 3) ün $c_1 = c_2$ olduğu halidir. Böylece $2) \Rightarrow 3) \Rightarrow 4)$ olur. $4) \Rightarrow 2)$ olduğunu olmayana ergi ile ispatlayacağız. Kabul edelim ki 2) sağlanmasın. Yani; $x_1 \neq x_2 \in C$ için $\Phi(x_1, c_1) \cap \Phi(x_2, c_2) \neq \emptyset$ olacak biçimde $c_1 \leq c$ ve $c_2 \leq c$ var olsun. O halde $c' = \max\{c_1, c_2\} \leq c$ vardır öyle ki $\Phi(x_1, c') \cap \Phi(x_2, c') \neq \emptyset$ olur. Yani 4) sağlanmaz. (Çelişki!) Son olarak $3) \Leftrightarrow 5)$ olduğu $d_{\min, w}^0$ nin tanımından elde edilir.

Teorem 4.1 $d_{\min, w}^0$ nin C kodunun tüm hata düzeltme kapasitesini karakterize ettiğini ispat eder. $d_{\min, w}^0$ özel durumunda klasik kodlama teorisindeki minimum uzaklığa eşittir. Fakat ağ bağlantı kodlarının minimum uzaklığını belirlemek klasik kodlamadaki minimum uzaklık kavramını belirlemekten daha karmaşıktır.

Örnek 4.3 Örnek 4.2 deki ağ bağlantısı için her $x, x' \in C$ farklı kodsözleri için $\Phi(x, 1) \cap \Phi(x', 1) = \emptyset$ olduğunu incelemiştik. O halde $\Phi(x, 0) \cap \Phi(x', 0) = \emptyset$ ve $\Phi(x, 0) \cap \Phi(x', 1) = \emptyset$ dir. Öte yandan mümkün tüm kodsözler ve farklı ağırlıktaki hatalar için yuvarlar incelendiğinde bazı $x \neq x'$ kodsözleri için $\Phi(x, 1) \cap \Phi(x', 2) \neq \emptyset$ dir. Mesela; $\Phi_w(0000000, 1) \cap \Phi_w(1000110, 2) = \{0000010, 1000000, 1000100\} \neq \emptyset$ ya da

$\Phi_w(0000000, 2) \cap \Phi_w(0010011, 1) = \{0000011, 0010010, 0001001\} \neq \emptyset$ olur. Yani;

$D_w^0(x_1, x_2) = \min\{1 + 2 : |2 - 1| \leq 1, \Phi_w(x_1, 1) \cap \Phi_w(x_2, 2) \neq \emptyset\}$ olacak biçimde minimum

uzaklık değeri elde edilir. Buradan $d_{\min, w}^0 = 3$ olur. Sonuç olarak $d_{\min, w}^0 = 3 \geq 2c + 1$

olduğu için bu kod tam 1 hata düzeltebilir. (Örnek 4.2 de gösterildiği gibi.)

4.3 Lineer Ağ Bağlantı Kodlarının Hata Belirleme Kapasitesi

Herhangi $x_1, x_2 \in C$ kod sözü için farklı bir uzaklık kavramı tanımlayalım.

Tanım 4.12 Herhangi $x_1, x_2 \in C$ kod sözü için x_1 ve x_2 arasındaki uzaklık

$D_w^1(x_1, x_2) = \min\{c : \Phi_w(x_1, 0) \cap \Phi_w(x_2, c) \neq \emptyset\}$ olarak ifade edilir. Bu uzaklık her zaman

simetrik değildir.

$D_w^1(x_1, x_2)$ ile irtibatlı olmak üzere minimum uzaklığı

$d_{\min, w}^1 = \min\{D_w^1(x_1, x_2) : x_1 \neq x_2 \in C\}$ olarak tanımlanır.

Tanım 4.13 Bir C kodu için $MWD_w^H(0)$ var ve $0 \leq w(z) \leq c$ olacak şekildeki tüm z hata vektörleri $MWD_w^H(0)$ ile belirlenebiliyor ise bu koda c hata belirleyebilir denir.

Teorem 4.2 [9] Bir kod c hata belirleyebilir ancak ve ancak $d_{\min, w}^1 \geq c + 1$ dir.

İspat: ispat aşağıdaki ifadelerin eşit olduğunun gösterilmesi yapılır.

1. Herhangi $x_1 \neq x_2 \in C$ için $\Phi_w(x_1, 0) \cap \Phi_w(x_2, c) = \emptyset$ dir.

2. Herhangi $x_1 \neq x_2 \in C$ için eğer $\Phi_w(x_1, 0) \cap \Phi_w(x_2, c') = \emptyset$ ise $c' > c$ dir.

3. $d_{\min, w}^1 \geq c + 1$ dir. $d_{\min, w}^1$ nin tanımından $1) \Leftrightarrow 2) \Leftrightarrow 3)$ olduğu açıktır. Şimdiye kadar

hata düzelten ve hata belirleyen ağ bağlantısı kodları için $d_{\min, w}^0$ ve $d_{\min, w}^1$ olmak üzere

iki farklı uzaklık ifadesi tanımladık. Bunlar her zaman eşit olmak zorunda değildir.

Örnek 4.4 Örnek 4.2 deki ağ bağlantı kodunu ele alalım. Örnek 4.2 den

$\Phi(x, 1) \cap \Phi(x', 1) = \emptyset$ olduğunu biliyoruz. Dolayısı ile $\Phi(x, 0) \cap \Phi(x', 1) = \emptyset$ dir. Öte

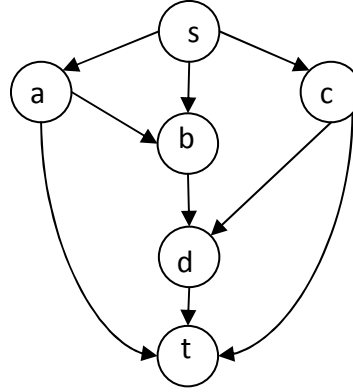
yandan tüm kod sözler istenilen hata ağırlığa göre iletildiğinde daima

$\Phi(x, 0) \cap \Phi(x', 2) = \emptyset$ dir. Fakat;

$\Phi(0000000, 0) \cap \Phi(1000110, 3) = \{0000000\}$,
 $\Phi(1000110, 0) \cap \Phi(1011010, 3) = \{1000110\}$, olacak biçimde kod sözler bulunur.

Buradan $c = 3$ olur. Tanım gereği $d_{\min, w}^1(C) = 3$ olup tam $d_{\min, w}^1(C) - 1 = 3 - 1 = 2$ hata tespit edilebilir.

Örnek 4.5 [9]



Şekil 4. 5 Lineer olmayan ağ bağlantısı

Yukarıdaki ağ bağlantı kodu $F_3 = \{0, 1, 2\}$ cismi üzerinde tanımlansın ve $C = \{x_0 = (0, 0, 0), x_1 = (1, 1, 1)\}$ olsun. Ağ bağlantısı lineer olmasın ve yerel kodlama fonksiyonları: $\bar{F}_{(a,t)} = F_{(s,a)}$, $\bar{F}_{(a,b)} = F_{(s,a)}$, $\bar{F}_{(c,t)} = F_{(s,c)}$, $\bar{F}_{(c,d)} = F_{(s,c)}$, ve $\bar{\beta}_{(b,d)}$ ve $\bar{\beta}_{(d,t)}$ sırası ile aşağıdaki gibi verilsin.

$F_{(s,b)}$	0	0	0	1	1	1	2	2	2
$F_{(a,b)}$	0	1	2	0	1	2	0	1	2
$\bar{F}_{(b,d)}$	0	0	0	0	2	1	0	0	0

$F_{(b,d)}$	0	0	0	1	1	1	2	2	2
$F_{(c,d)}$	0	1	2	0	1	2	0	1	2
$\bar{F}_{(b,t)}$	0	0	0	1	1	0	0	1	0

Tablo 4. 1 Yerel kodlama dönüşümleri

Bu ağ bağlantısı için $\Sigma = F_3^9$ ve uzaklık Hamming uzaklık olarak tanımlasın. O halde lineer olmayan ağ bağlantısı için verilen tablolardan $\Phi(x_0, 1)$ ve $\Phi(x_1, 1)$ yuvarlarını belirleyelim. Yukarıdaki ağ bağlantısı için x_0 mesajı gönderildiğinde

$(s, a), (s, b), (s, c), (b, d), (c, d), (a, t), (c, t)$ ayrıtları üzerinde sırası ile ağırlığı 1 den küçük ya da eşit olan hatalar yapılırsa alınan vektörlerin kümesi;

$\Phi(x_0, 1) = \{(0, 0, 0), (1, 0, 0), (0, 1, 0), (0, 0, 1), (2, 0, 0), (0, 2, 0), (0, 0, 2)\}$ ve x_1 mesajı gönderildiğinde alınabilecek tüm vektörlerin kümesi

$\Phi(x_1, 1) = \{(0, 1, 1), (1, 0, 1), (1, 1, 0), (1, 1, 1), (1, 0, 2), (2, 0, 1), (2, 1, 1), (1, 2, 1), (1, 1, 2)\}$ olur.

Sonuç olarak $\Phi(x_0, 1) \cap \Phi(x_1, 1) = \emptyset$ dir. Buradan $\Phi_w(x, z)$ kümesinin tanımından

$\Phi(x_0, 0) \subset \Phi(x_0, 1)$ ve $\Phi(x_1, 0) \subset \Phi(x_1, 1)$ olduğu için açıktır ki $\Phi(x_0, 0) \cap \Phi(x_1, 1) = \emptyset$

ve $\Phi(x_0, 1) \cap \Phi(x_1, 0) = \emptyset$ dir. Kabul edelim ki z hata vektörü $z_{(s,a)} = z_{(s,c)} = 2$ ve diğer

tüm bileşenleri için 0 olsun. Yani, $z = (0, 2, 2, 0, 0, 0, 0, 0, 0)$ olsun. O halde

$F_t(x_1, z) = F_t(x_0, 0) = (0, 0, 0)$ dir. Sonuç olarak $\Phi(x_0, 0) \cap \Phi(x_1, 2) \neq \emptyset$, $d_{\min, w}^0 = 3$ ve

$d_{\min, w}^1 = 2$ dir. Yani bu C kodu bir hata 1 hata belirleyebilir ve 1 hata düzeltebilir. Yani;

$d_{\min, w}^0 \neq d_{\min, w}^1$ olur. Bu örnek iki uzaklık ifadesinin eşit olmadığını gösteren lineer

olmayan bir ağ bağlantı kodu sunar.

SONUÇ VE ÖNERİLER

Bu tezde ağ bağlantı kodları, inşa şekilleri ve ağ bağlantısında bilgi iletiminin nasıl olduğu incelenmiştir. Ağ bağlantı kodlarına cebirsel bir yaklaşımda bulunulmuştur. Ayrıca ağ bağlantı kodlama probleminin çözümü için sonlu cisimler üzerinde gerekli şartlar verilmiş ve örnekler ile çözüme sahip olmayan ağ bağlantıları verilmiştir.

Ağ bağlantılarında meydana gelebilecek hata ve veri kayıplarına karşı hata belirleme ve hata düzeltme için gerekli tanım ve teoremler ifade edilmiştir. Ağ bağlantı kodlamasının yapısı klasik kodlama teorisine göre daha karmaşıktır. Klasik kodlama teorisinin ağ bağlantı kodlamasının özel bir hali olduğu örneklerle gösterilmiştir.

Ağ bağlantı kodlaması yeni bir çalışma alanı olduğundan bu konuda pek çok cevaplanmayı bekleyen soru vardır. Bu kodlamanın halkalar, modüller ve değişik cebirsel yapılar üzerinde nasıl uygulanacağı, hata tespiti ve düzeltilmesi araştırılabilir. Ayrıca “Application of Automata Theory in Coding” [10] isimli makalede hücrel dönüşümler tanımlanmıştır. Daha sonra “Cellular Neural Network: Theory” [11] isimli makalede hücrel sinir ağ bağlantıları ifade edilmiştir. 1988 ‘de “Fractals, Neural Networks, Cellular Automata, Formal Language and Coding Theory” [12] isimli makalede farktallar, veri iletimi ve sinir ağ bağlantıları arasındaki ilişkiden bahsedilmiştir. Ağ bağlantı kodlaması ve hücrel dönüşümler arasındaki ilişki çalışılması gereken ve merak uyandıran bir konudur.

KAYNAKLAR

-
- [1] Yeung, R.W. ve Zhen, Z., (1999). "Distributed Source Coding for Satellite communications", Information Theory IEEE Transactions on: 1111-1120
 - [2] Rudolf, A., Ning, C., Shuo-Yen, R.L. ve Raymond, W.Y (2000). "Network Information Flow", Information Theory IEEE Transactions on:1204-1216.
 - [3] Tracey H., Ralf K., Muriel M., David R. K. Ve Michelle E., (2003). "The Benefits of Coding over Routing in a Randomized Setting", Information Theory IEEE Transactions on:442
 - [4] Robin J.W. ve John J W., (1990). Graphs: An Introductory Approach : A First Course in Discrete Mathematics, 1, 9780471615545 John Wiley High Education, South Bend, U.S.A
 - [5] Ling S. ve Xing C. (2004). Coding Theory First Course, 1, Cambridge University Press in New York
 - [6] Raymond W.Y., Shuo-Yen R.L., Ning C., Zhen Z. (2006). Network Coding Theory 1, 1933019247, Now Publisher Inc., Boston
 - [7] Koetter R., Medard M., (2003). "An Algebraic Approach to Network Coding" Networking, IEEE/ACM Transactions on: 102
 - [8] Vatter V., (2004). "Graph, Flows, and the Ford-Fulkerson Algorithm", <http://www.math.ufl.edu/~vatter/teaching/summer04/flow.pdf>, 06.07.2011.
 - [9] Yeung, R.W., Zhen Z. ve Yang S (2007). "Weight Properties of Network Codes", European Transaction on Telecommunications:1-14
 - [10] Massey, J.L., (1968). "Application of Automata Theory in Coding" ISI Web of Knowledge: 125-146
 - [11] Leon O.C. ve Yang, L. (1988). "Cellular Neural Network: Theory", IEEE Transactions on Circuits and Systems: 1257-1272
 - [12] Ling, Y., (1992). "Fractals, Neural Networks, Cellular Automata, Formal Language and Coding Theory", Information Theory IEEE Transactions on: 1663-1669

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Ayşegül BAYRAM
Doğum Tarihi ve Yeri : 16.02.1987, Araklı/Trabzon
Yabancı Dili : İngilizce
E-posta : aaysegulbayram@gmail.com

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Y.Lisans	Matematik Müh.	Yıldız Teknik Üniversitesi	2012
Lisans	Matematik	Sakarya Üniversitesi	2009
Lise	Fen Bilimleri	Adapazarı Atatürk Lisesi	2005

Proje

- Proje No:** 110T713
Proje Adı: Cisim ve Halkalar Üzerinde Tanımlı 2-Boyutlu Hücresel Dönüşümlerin Cebirsel Yapıları ve Davranışları (Bursiyer)