

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BELİRLEME KODLARI VE KÜME KAPLAMA PROBLEMİ

FERHAT KÜRÜZ

YÜKSEK LİSANS TEZİ
MATEMATİK ANABİLİM DALI

DANIŞMAN
DOÇ. DR. BAYRAM ALİ ERSOY

İSTANBUL, 2012

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BELİRLEME KODLARI VE KÜME KAPLAMA PROBLEMİ

FERHAT KÜRÜZ

YÜKSEK LİSANS TEZİ
MATEMATİK ANABİLİM DALI

DANIŞMAN
DOÇ. DR. BAYRAM ALİ ERSOY

İSTANBUL, 2012

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BELİRLEME KODLARI VE KÜME KAPLAMA PROBLEMİ

Ferhat Kürüz tarafından hazırlanan tez çalışması 05.06.2012 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Doç. Dr. Bayram Ali ERSOY
Yıldız Teknik Üniversitesi

Jüri Üyeleri

Prof. Dr. İrfan ŞİAP
Yıldız Teknik Üniversitesi

Doç. Dr. Bayram Ali ERSOY
Yıldız Teknik Üniversitesi

Doç. Dr. Ünsal TEKİR
Marmara Üniversitesi

ÖNSÖZ

Bu tez çalışması yakın zamanda çalışılmaya başlanmış olan Belirleme Kodları ile ilgili temel bilgiler verip bu alandaki çalışmaların gelişmesine vesile olmak amaçlı hazırlanmıştır.

Bu çalışmam esnasında benden desteklerini esirgemeyen değerli hocalarım Sayın Prof. Dr. İrfan Şiap, Sayın Doç. Dr. Bayram Ali Ersoy ve Sayın Yrd. Doç. Dr. Kürşat Hakan Oral'a sonsuz teşekkürlerimi bildiririm. Ayrıca doğumumdan bu yana maddi manevi tüm ihtiyaçlarımı karşılayan ve beni hiç yalnız bırakmayan canım anneme, canım babama ve canım kardeşime şükranlarını sunarım. Son olarak, zor zamanlarımda sıkıntılarımı paylaştığım ve artık kardeşlerim olarak gördüğüm dostlarım Ali Dağdeviren, Murat Aras, Serdar Cellat, Hidayet Beyhan, Fadıl Saha, Çarıyar Çarıyev ve İbrahim Özbek'e sevgilerimi iletirim.

Mayıs, 2012

Ferhat KÜRÜZ

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ.....	vi
ŞEKİL LİSTESİ.....	vii
ÇİZELGE LİSTESİ	viii
ÖZET	ix
ABSTRACT.....	x
BÖLÜM 1	
GİRİŞ	1
1.1 Literatür Özeti	1
1.2 Tezin Amacı.....	2
1.3 Hipotez	3
BÖLÜM 2	
KÜMELER TEORİSİ.....	4
BÖLÜM 3	
KARMAŞIKLIK TEORİSİ	5
3.1 Zaman Karmaşıklığı.....	5
3.2 Algoritmaların karşılaştırılması	6
3.3 Turing makinesi	7
3.4 P karmaşıklık sınıfı	8
3.5 NP karmaşıklık sınıfı	8
BÖLÜM 4	
GRAF TEORİSİ	9

BÖLÜM 5

TANIMLAR VE İLGİLİ ÇALIŞMALAR	16
5.1 Belirleme Kodları.....	16
5.2 Belirlenebilen Graflar	16
5.3 Belirleme Kodları ile ilgili Alt Sınır, Üst Sınır Tahmini	17
5.4 Küme Kaplama Problemi.....	17
5.5 Çoklu Küme Kaplama Problemi	17

BÖLÜM 6

BELİRLEME KODLARI ve KÜME KAPLAMA PROBLEMİ	18
6.1 ID-KOD $\leftarrow$ KÜME KAPLAMA	18
6.2 KÜME KAPLAMA $\leftarrow$ ID-KOD	23
6.3 Minimum Küme Kaplamadan Minimum Belirleme Koduna Geçiş Sağlayan Bir Yapı.....	29

BÖLÜM 7

BELİRLEME KODLARININ UYGULAMALARI	30
7.1 Çok işlemcili sistemlerde Hata tespiti.....	35
7.2 Konum Belirleme	36
7.2.1 Davutpaşa’da Konum Belirleme Uygulaması	38
7.3 Çevresel denetleme	48

BÖLÜM 8

SONUÇ VE ÖNERİLER.....	49
KAYNAKLAR	50
ÖZGEÇMİŞ	52

SİMGE LİSTESİ

$\mathbb{Z}$	Tamsayılar kümesi
$\lfloor x \rfloor$	x'in alt sınırı
$\lceil x \rceil$	x'in üst sınırı
$\subseteq$	Alt kümesi veya eşit
$\emptyset$	Boş küme
$\cap$	Kesişim
$\cup$	Birleşim
$\in$	Elemanı
$\times$	Kartezyen çarpım
$-$	Küme farkı
Δ	Simetrik fark
$P(X)$	X kümesinin kuvvet kümesi
$T_A(n)$	A algoritmasının n. adımındaki işlem sayısı
O	Zaman karmaşıklığı
$\leq$	Küçük veya eşit
$\geq$	Büyük veya eşit
$\neq$	Farklı
$G(V,E)$	G grafi
V_G	G grafinin düğümler kümesi
E_G	G grafinin kenarlar kümesi
$B(v)$	v düğümünün komşuluk kümesi
$B^+(v)$	v düğümüne gelen düğümler kümesi
$B^-(v)$	v düğümünden giden düğümler kümesi
B^k	k-küp graf
$d_G(u,v)$	u ile v düğümleri arasındaki en yakın uzaklık
C	Belirleme kodu
$\log$	Logaritma
$D(u,v)$	u ile v düğümlerinin farklılık kümesi
δ_v	v düğümünün ayırım kümesi
$I_C(v)$	v düğümünün C koduna göre belirleme kümesi
U	Bir graftaki tüm farklı düğüm çiftlerinin kümesi
$M(U,S)$	(U,S) küme kaplama problemine karşılık gelen grafin komşuluk matrisi

ŞEKİL LİSTESİ

Sayfa

Şekil 1.1	Bir belirleme kodu örneği. İçi dolu düğümler $\{0,1,2\}$ belirleme kodunu gösteriyor. Parantezlerin içindekiler düğümlerle belirleme kodunun kesişimini (belirlemekümeleri)gösteriyor.....	2
Şekil 3.1	Problemlerin karmaşıklık sınıflarının Venn şeması ile gösterimi.....	9
Şekil 4.1	Pregel nehri[16]	10
Şekil 4.2	Königsberg nehrinin haritası ve graf modeli	11
Şekil 4.3	Örnek 4.1'deki grafın diyagramı.....	12
Şekil 4.4	Tam grafa bir örnek	12
Şekil 4.5	3-küp grafi.....	13
Şekil 4.6	Ağırlıklı grafa bir örnek	14
Şekil 6.1	Örnek 6.1'in grafi.....	20
Şekil 6.2	Teorem 6.1'i desteklemek için incelenen örnek graf.....	22
Şekil 6.3	İndirgememize bir örnek.....	24
Şekil 6.4	Yapı 6.1'e verilen örneğin belirleme kodunun grafa yerleştirilmiş hâli. Burada içi dolu olanlar kodsözlerdir	26
Şekil 6.5	Yapı 6.2'nin örneğindeki (U,S) küme kaplama problemine karşılık gelen graf	33
Şekil 7.1	Örnek konum belirleme alanımız.....	37
Şekil 7.2	Belirleme kodlarının örneğimize uygulanması	38
Şekil 7.3	Belirleme kodlarını konum belirleme uyg'da kullanacağımız bölge.....	39
Şekil 7.4	Örnek alanımızın basit bir haritası ve graf modeli. İçi dolu olanlar bu graf için bulunan belirleme kodunun kodsözleridir	47
Şekil 7.5	Belirleme kodlarının çevresel denetlemede kullanımına bir örnek	48

ÇİZELGE LİSTESİ

	Sayfa
Çizelge 6.1	Küme Kaplama ve Belirleme Kodu Problemleri arasındaki ilişki..... 18
Çizelge 6.2	Şekil 6.2'deki grafin düğümlerinin belirleme ve ayrım kümeleri..... 22

BELİRLEME KODLARI VE KÜME KAPLAMA PROBLEMİ

FERHAT KÜRÜZ

Matematik Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Doç. Dr. Bayram Ali ERSOY

Graf teorisi günlük hayatta birçok uygulama alanı olan araştırma konularından birisidir. Bu bağlamda çoğu optimizasyon problemi graf teorisine uyarlandıktan sonra belirleme kodları vasıtasıyla çözüme kavuşturulabilir.

Diğer taraftan küme kaplama problemi de gerçek hayata uyarlanabilecek problemlerden birisidir. Bu çalışmada belirleme kodları ve küme kaplama problemi arasındaki bir indirgeme incelendi. Bu indirgeme aracılığıyla bu problemlerden birisine uyarladığımız bir problemi diğerine de aktarabiliyoruz.

Çalışma planımızı şu şekilde özetleyebiliriz: Bölüm 2-3-4-5’de kümeler teorisi, graf teorisi, karmaşıklık sınıfları, belirleme kodları ve kaplama problemi ile ilgili temel tanımlar verilmiştir. Ana problemimiz olan belirleme kodları ile kaplama problemi arasındaki indirgemeye ise Bölüm 6’da geçildi. Bu bölümde bu iki problem arasında geçişi sağlayacak iki yapı verildi. Bir sonraki bölümde de belirleme kodları ile ilgili uygulamalar incelendi.

Anahtar kelimeler: Belirleme kodları, Kaplama problemi, Küme kaplama, Graf teorisi

ABSTRACT

IDENTIFYING CODES AND SET COVERING PROBLEM

Ferhat KÜRÜZ

Department of Mathematics

MSc. Thesis

Advisor: Assoc. Prof. Dr. Bayram Ali ERSOY

Graph theory is a research area that one of the most find practise. In this manner, after an optimization problem adapt to the graph theory, it can be solved via the identifying codes.

In addition to this, the set cover problem is one of the problems can be applied to real life. A reduction between the covering problems and identifying codes have been studied in this thesis. We can transfer a problem that have been adapted one of these problems from one to another.

We can summarize our work such following: Definitions about set theory, graph theory, complexity classes, identifying codes and set cover problems were given in sections 2-3-4-5. The reduction between the covering problems and identifying codes have been studied in 6. chapter. There are two structures that supply to pass from one to another problem in this chapter. In the next section applications about identifying code has been invastigated.

Keywords:Identifying codes, Covering problem, Set cover, Graph theory

YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE

BÖLÜM 1

GİRİŞ

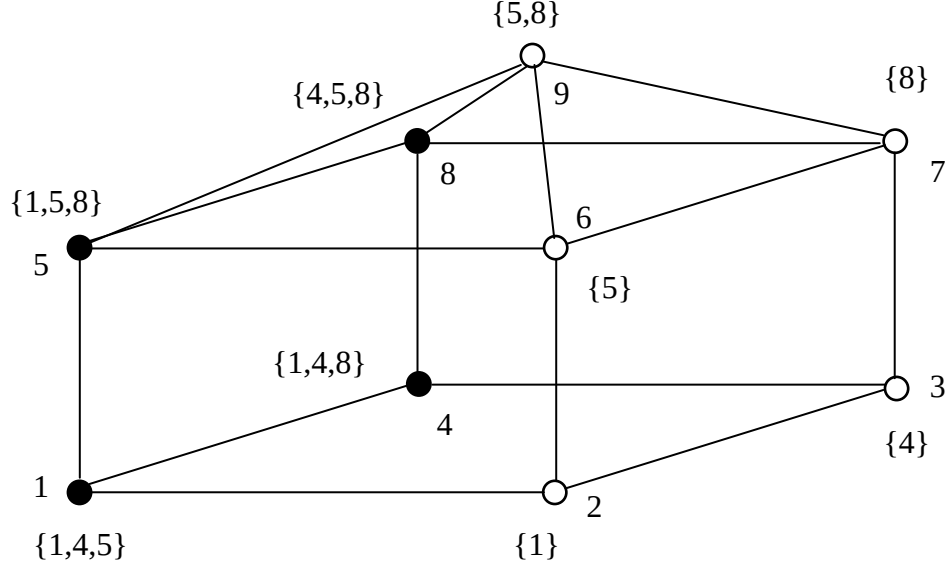
1.1 Literatür Özeti

Belirleme kodları 1998’de Karpovsky vd. tarafından [1]’de çalışılmaya başlandı. Bu çalışmanın ardından belirleme kodları çok geniş bir uygulama alanı buldu. Teorik ve pratikteki uygulamalar açısından birçok alanda temel haline geldi.

Belirleme kodlarının uygulama alanlarına kısaca bir göz atacak olursak, ilk uygulaması çoklu işlemcili sistemlerde hata tespitinde kullanıldığını görürüz[1]. Bu çalışmada öncelikle sistemin graf modeli çıkarılır ve test cihazları belirleme koduna göre yerleştirilir. Böylelikle her bir test cihazı komşulukları içinde hangi test cihazının hataları belirleyeceğini düşünerek hataları belirler[1].

Bu çalışmadan sonra belirleme kodlarının kullanım alanları oldukça genişledi. Bazı sistemlerde enerji düzenleme [2], konum belirleme[3], çevresel denetleme(izleme) [4],kablosuzsensör ağlarında arama yapma [5]gibi birçok alanda uygulanmaya başlandı. Sistemlerde enerji düzenlemedeki uygulamasında öncelikle sistemin kullanıldığı alan sonlu sayıda bölgeye ayrılır ve her bir bölge birer düğümle temsil edilir. Daha sonra bu alanların birbirleriyle olan iletişimlerine bakılarak aralarına kenarlar yerleştirilerek sistemin graf modeli oluşturulmuş olur[2]. Bu sistemde belirleme kodunun bir ömrü vardır, çünkü düğümlerin temsil ettiği cihazlar bataryalıdır(pilli). Burada sistemin ömrünü olabildiğince uzatmak maksadıyla sistem için farklı belirleme kodları üretilir, yani sistemin enerji dengesini ayarlamak için oluşturulabilecek tüm belirleme kodları bulunur. İkinci uygulamada ise(konum belirleme)sinyaller stratejik pozisyonlardan her bir bölge ayırt edilebilecek şekilde yayın yapar. Böyle bir sistemde bir kullanıcı bulunduğu konumda hangi sinyal kümesinden sinyal alıyorsa bu kümeye göre yerini belirler. Burada da sinyal konulacak bölgelere belirleme kodlarına göre karar verilir ve kullanıcılar konumlarını belirleme kümelerine göre belirlemiş olur[3].

Belirleme kodlarının bazı çeşitleri vardır. Yarıçapı $\rho > 1$ olan belirleme kodları, v 'nin komşuluğu, v 'den ρ ve daha az uzaklıktaki düğümleri içerecek şekilde tanımlanmıştır[1]. $(1, \leq l)$ -belirleme kodları [6] aynı anda, en fazla l düğüme sahip



Şekil 1.1 Bir belirleme kodu örneği. İçerik dolu düğümler $\{0,1,2\}$ belirleme kodunu gösteriyor. Parantezlerin içindekiler düğümlerle belirleme kodunun kesişimini(belirleme kümeleri) gösteriyor.

herhangi bir alt kümesini tanımlar. Güçlü belirleme kodları [2]'te, düğümlerin ve kenarların kopmasının kolay olduğu sert çevrelerde konum bulma uygulamaları için çalışıldı. Bir r -güçlü belirleme kodu, tüm belirleme kodlarından r farklı düğüme kadar ekleme veya çıkarma durumunda belirleme özelliğini devam ettiren koddur. r -güçlü belirleme kodlarının, minimum Hamming uzaklığı $2r+1$ olan hata düzelten kodlar olduğu [7]'te gözlemlenmiştir.

1.2 Tezin Amacı

Bir graf için bir belirleme kodu, düğümlerin öyle bir alt kümesidir ki, bu alt kümenin, düğümlerin gelen düğümler kümeleri ile kesişimleri birbirlerinden farklıdır[8]. Örneğin Şekil 1.1'deki graf bir 4 düğümlü belirleme koduna sahiptir(1,4,5,8 ile etiketli olanlar).

Graftaki her bir düğümün gelen düğümler kümelerinin(buradaki graf yönsüz olduğu için düğümlerin gelen düğümler kümeleri, komşuluklarına eşittir) bu kod ile kesişimi farklıdır ve her bir kesişim bir belirleme kümesi olarak isimlendirilir. Herhangi bir belirleme kümesi kendisini üreten düğümü tanımlar. Belirleme kod probleminin hedefi verilen herhangi bir graf için minimum kardinaliteli belirleme kodunu bulmaktır.

Bu alıřmada belirleme kodları ve kme kaplama arasındaki iliřki incelenecek ve zgn rneklerle somutlařtırılmaya alıřılacaktır. Son kısımda ise belirleme kodlarının uygulamaları incelenecek ve yine aıklayıcı rneklerle anlatılanlar pekiřtirilecektir.

1.3 Hipotez

Bu alıřmada belirleme kodları ve kme kaplama ile ilgili bilgiler verilip bu iki problem arasındaki iliřki incelenecek. Tezimizin ana konusu belirleme kodları ile kme kaplama problemi arasında geiřleri saėlayan yapıları gstermektir.

BÖLÜM 2

KÜMELER TEORİSİ

Bu bölümde ileriki bölümlerde kümeler teorisinden gerekli olacak bazı gösterim ve tanımlar vereceğiz.

- Sonlu bir X kümesi için $|X|$, X 'in eleman sayısını gösterir[9].
- 1'den n 'ye kadar olan tam sayıların kümesi $[1,n]$ ile gösterilir, yani $[1,n]=\{1,2,\dots,n\}$ 'dir[9].
Benzer şekilde $i \leq n$ için i 'den n 'ye kadar olan tam sayıların kümesi $[i,n]$ ile gösterilir, yani $[i,n]=\{i,i+1,\dots,n\}$ 'dir[9].
- Bir reel x sayısı için $\lfloor x \rfloor = \max\{k \in \mathbb{Z} \mid k \leq x\}$ tam sayısına x 'in *alt sınırı* denir. Ayrıca yine bir reel x sayısı için $\lceil x \rceil = \min\{k \in \mathbb{Z} \mid x \leq k\}$ tam sayısına da x 'in *üst sınırı* denir[9].
- Bir X kümesi için $X_i \subseteq X$ olacak şekilde, $\{X_1, X_2, \dots, X_k\}$ kümesi tüm farklı i ve j 'ler için $X = \bigcup_{i \in [1,k]} X_i$ ve $X_i \cap X_j = \emptyset$ özelliklerini sağlıyorsa bu $\{X_1, X_2, \dots, X_k\}$ alt küme ailesine X 'in bir *parçalanışı* denir[9].
- X ve Y kümeleri için $X \times Y = \{(x,y) \mid x \in X, y \in Y\}$ kümesi X ve Y 'nin *Kartezyen çarpımı* olarak adlandırılır[9].
- X ve Y kümeleri için $X - Y = \{x \mid x \in X, x \notin Y\}$ kümesine X 'in Y 'den *farkı* denir[9].
- X ve Y kümeleri için $X \Delta Y = (X - Y) \cup (Y - X)$ kümesine X ve Y 'nin *simetrik farkı* denir[9].
- Bir X kümesinin tüm alt kümelerinin ailesine bu X kümesinin *kuvvet kümesi* denir ve $P(X)$ ile gösterilir[10].

KARMAŞIKLIK TEORİSİ

Karmaşıklık teorisi bilgisayar bilimlerinde hesaplama teorisinin bir dalıdır. Bu teori algoritmaların uygulanması için gerekli olan kaynak, girdi, olanak vb. nesnelerin miktarını (uygulanma süresi vb.) ve özel hesaplama problemleri için etkili algoritmalar sağlamadaki zorlukları inceler. Algoritmadaki girdi sayısı arttıkça bu algoritmanın çalışma süresi ve hafıza gerekliliği nasıl değişir ve bu değişikliğin sebepleri nelerdir? sorusu bu teorisin ana sorusudur.

3.1 Zaman Karmaşıklığı

Problemleri çözülebilen ve çözilemeyen olarak iki sınıfa ayırabiliriz. Pratik olarak çözülebilen problemleri makul bir süre içerisinde çözilemeyen problemlerden ayırabiliriz. Bu ayırımı bir bilgisayar algoritması oluştururken depolama ve süre açısından ne kadar masraflı olacağını hesaplamak için çok önemlidir.

Algoritma karmaşıklık analizi bir problemi çözmek için ne kadar süre alacağını gösterir. Bir problem aşamasının büyüklüğü bu aşamadaki girdilerin sayısı ile ölçülür. Algoritma karmaşıklık analizi aritmetik, mantık, bilgi depolama gibi basit hesaplama işlemlerine dayanır. Bir A algoritmasının zaman karmaşıklığı $T_A(n)$ ile gösterilir ve algoritmanın n. adımındaki işlem sayısını belirtir. Herhangi bir algoritmanın tam işlem sayısını hesaplamaktan daha ziyade $T_A(n)$ 'in büyüme oranı incelenir. Bir A algoritmasının uygulamasının pratik veya uygulanabilir olduğunu kararlaştırmak için $T(n)$ 'in çok büyük olmayan bir üst sınırı gereklidir[11].

Algoritmaların zaman karmaşıklığı O ile gösterilir. Bir $f(n)$ fonksiyonu diğer bir $g(n)$ fonksiyonu ile tüm $n > 0$ 'lar ve bazı c sabitleri için $|f(n)| \leq c \cdot |g(n)|$ eşitsizliğini sağlıyorsa bu $f(n)$ fonksiyonunun karmaşıklığı $O(g(n))$ 'dir. Örneğin bir algoritmanın karmaşıklığı

$O(n)$ ile gösteriliyorsa algoritmanın hızı girdinin büyüklüğü ile doğru orantılı demektir. Başka bir örnek vermek gerekirse bir algoritmanın karmaşıklığı $O(n^2)$ ile verildiyse bu algoritmanın hızı girdinin büyüklüğünün karesi ile orantılıdır[12].

Örnek 3.1 $2x^3+5x+1=O(x^3)$ ’tür çünkü her $x>$ için $|2x^3+5x+1|\leq 9|x^3|$ yazabiliriz.

O gösteriminin yanında Ω ve Θ olmak üzere iki ayrı gösterimimiz daha var. Bunları özetlememiz gerekirse, $f(n)=O(g(n))$ f ile g ’nin aynı oranda arttığını veya f ’nin daha yavaş arttığını, $f(n)=\Omega(g(n))$ f ’nin g ’den daha hızlı arttığını, $f(n)=\Theta(g(n))$ de f ile g ’nin aynı oranda arttığını gösterir[12].

Herhangi bir algoritmanın zaman karmaşıklık fonksiyonu bazı $k\geq 0$ için $O(n^k)$ ile veriliyorsa bu algoritmaya *polinomsal zamanlı*(polynomial time) denir. Buradan bir A algoritması için $T_A(n)\leq f(n)$ ve f polinom ise yani $a_1n^k+a_2n^{k-1}+...+a_m$ şeklinde ise bu algoritmaya polinomsal zamanlı denir[12].

3.2 Algoritmaların karşılaştırılması

A ve B bir X probleminin çözümleri olsun ve T_A , T_B sırasıyla A ve B ’nin zaman karmaşıklıkları olsun. Eğer $T_A=T_B$ ise A ve B aynı karmaşıklığa sahiptir denir. Ancak, $T_A=O(T_B)$ fakat $T_B\neq O(T_A)$ ise A algoritması B algoritmasından daha iyidir[13].

Örnek 3.2. A ve B algoritmalarının zaman karmaşıklıkları $T_A(n)$ ve $T_B(n)$ olmak üzere,

$$T_A = \begin{cases} 5 & n \leq 5 \\ n & n > 5 \end{cases},$$

$$T_B = \begin{cases} n & n \leq 5 \\ n^2 & n > 5 \end{cases},$$

şeklinde verilsin. Şimdi bu algoritmaları karşılaştıralım. $T_A(n)=O(T_B(n))$ fakat $T_B(n)\neq O(T_A(n))$ ’dır. Buradan A ’nın B ’den daha iyi bir algoritma olduğu görülür.

Herhangi bir X problemi için bilinen polinomsal zamanlı algoritma varsa bu probleme *uysal problem* denir, eğer bu X problemi çözülebilir bir problem ancak çözüm için bilinen bir polinomsal zamanlı bir algoritma yoksa bu probleme *zorlu problem* denir[14].

Bir A algoritması bir x katar girdisini kabul ederse evet çıktısı verir. Herhangi bir problemin çözümünün algoritması tarafından kabul edilen katarların bir L kümesi varsa

bu problem *karar problemidir*. Burada bir karar problemi için L harfi de kullanılır çünkü katarların kümesi bir *dil(language)* olarak adlandırılır. Bu dil tabanlı bakış açısını, bir A algoritmasına giren x girdisi eğer L'nin elemanı ise 'evet' çıktısı verir, değilse 'hayır' çıktısı verir, şeklinde geliştirebiliriz[14].

Karmaşıklık teorisinde problemler çözülebildikleri en hızlı karmaşıklığa göre karmaşıklık sınıflarına (computational complexity) ayrılırlar. Bunlardan en önemlileri P (deterministically polynomial: deterministik polinomsal) ve NP (non-deterministically polynomial: deterministik olmayan polinomsal)' dir.

3.3 Turing makinesi

Bir kafadan ve bir teyp bandından oluşan bir makinedir. Bu makinede 'yazma', 'okuma', 'bandı ileri sarma' ve 'bandı geri sarma' işlemleri yapılır. Bu işlemleri kullanarak bir işin yapılıp yapılamayacağı veya bir dilin bu basit 4 işleme indirgenip indirgenemeyeceğini inceleyerek dilleri ve işlemleri sınıflandırırız. Turing makinesi $M = (Q, \Sigma, \Gamma, \delta, q_0, \diamond, F)$ ile gösterilir[15]. Burada, M makinesinin parçalarını kısaca şöyle tanımlayabiliriz:

- Q, sonlu sayıdaki makinenin işlem sırasındaki durumlarının kümesidir.
- Σ , makinede işlenecek olan girdilerin kümesidir.
- Γ , dildeki tüm harfleri içeren kümedir.
- δ , makinenin çalışması esnasında kullanacağı geçişler ve ilişkileri gösterir.
- q_0 , makinenin başlangıç durumunu gösterir.
- $\diamond$, teyp bandının üzerindeki boşlukları gösterir.
- F, makinenin son durumunu gösterir.

Örnek 3.3.1 Makinemiz $\{x, y\}$ ifadeleri ile çalışsın, x ve y harflerini ard arda gördüğünde bantın sonuna gitsin. Bu durumda makine $M = (\{q_0, q_1, q_2\}, \{x, y\}, \{x, y, f\}, \{q_0 x \rightarrow x R q_1, q_0 y \rightarrow y R q_0, q_0 f \rightarrow f R q_0, q_1 y \rightarrow y R q_2, q_1 x \rightarrow x R q_1, q_1 f \rightarrow f R q_0\}, q_0, f, q_2)$ şeklinde tanımlanır.

- Q değeri $\{q_0, q_1, q_2\}$ ile gösterilmiş yani makinenin 3 durumu var.
- Σ ifadesi $\{x, y\}$ ile verilmiş, yani makine x ve y girdilerini alıyor.
- Γ ifadesi ile makinedeki kullanılan tüm semboller $\{x, y, f\}$ ile belirtilmiş.

- δ ifadesi ile 6 geiş verilmiř., $q_0 x \rightarrow x R q_1$ geiři x grdüğünde x 'in üzerine x 'i tekrar yazıp bir ileri yapıp y 'yi bekliyor. $q_0 y \rightarrow y R q_0$ geiři ilk bařta y grdüğümüzde y 'nin üzerine y yazıp bir ilerleyip x 'i bekliyor. $q_0 f \rightarrow f R q_0$ geiři ilk bařta boşluk grdüğümüzde boşluęa hibir řey yazmadan bir ilerleyip x 'i bekliyor. $q_1 y \rightarrow y R q_2$ geiři x 'i grdükten sonra y 'yi grnce y 'nin üzerine y yazıp bir saęa sarıp iřlemi bitiriyor. $q_1 x \rightarrow x R q_1$ geiři x 'den sonra x grnce bu x 'in yerine tekrar x yazıp y 'yi beklemeyi saęlıyor. $q_1 f \rightarrow f R q_0$ geiři x 'den sonra boşluk grlnce hibir řey yazmadan bir saęa ilerleyip ilk duruma gelmeyi saęlar.
- q_0 ile makinenin ilk hali belirtilmiřtir.
- $\diamond$ deęeri f ile belirtilmiřtir, yani boşluklar f ile gsterilmiřtir.
- q_2 ile makinanın son hali belirtilmiřtir ve xy 'yi grnce iřlemi bitirir.

3.4 P karmařıklık sınıfı

En kt ihtimalle bir deterministic Turing makinesinde polinomsal zamanlı bir algoritma ile czlebilen tm L karar problemlerinin kmesine *P karmařıklık sınıfı* denir. Dięer bir ifadeyle, bir A algoritmasında $x \in L$ iin n x 'in byklę olmak zere 'evet' ıktısı $p(n)$ (polinom) zamanda ıkıyorsa bu problem P karmařıklık sınıfındadır. P 'nin tanımında kabul edilmeyen yani 'hayır' ıktısı veren girdilerin alıřma sresi ile ilgili bir řey sylenmedięine dikkat edelim. Byle durumlar L dilinin iinde olmayan tm ikili katarları kapsayan *tamamlayıcısı* ile ilgilidir[14].

3.5 NP karmařıklık sınıfı

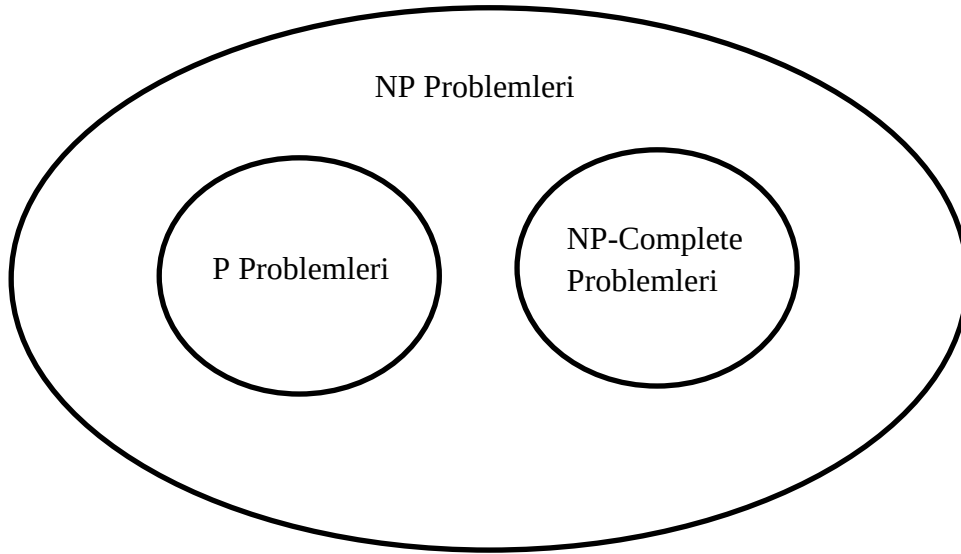
NP karmařıklık sınıfı P karmařıklık sınıfını da kapsar ancak bu sınıf ekstra olarak P 'de olmayan bazı dilleri kapsamaya da izin verir. zel olarak, NP problemleri ile algoritmalarda *seim*(g) iřlemini yapabiliriz ki bu iřlem g iin nondeterministic bir yolla 0 veya 1 deęerini belirler ve atar. Bir A algoritması *seim* iřlemini kullanırsa bu algoritmaya *nondeterministic* deriz. Bu algoritmada *seim* iřlemi mmkn olan tm sonuları arıyor ve eęer bu tr sonuların bir kmesi varsa bunları yalnızca kabul iin kullanıyor. NP karmařıklık sınıfınınondeterministic bir algoritma ile polinomsal zamanda czlebilen L karar problemleri(veya dilleri)'nin kmesidir. Yani burada $x \in L$ ise x 'in byklę n ve $p(n)$ polinom olmak zere x 'in zerinde $p(n)$ zamanda 'evet' ıktısı veren sonuların kmesini *seim* ile arayan nondeterministic bir A algoritması vardır.

Burada NP'nin tanımının herhangi bir ret için çalışma süresi ile alakalı olmadığına dikkat edelim. Gerçekten, bir A algoritmasında bir L dilinin, A algoritması 'hayır' çıktısı verince $p(n)$ polinomsal zamanda $p(n)$ 'den daha fazla adım atmasını sağlıyor[14].

NP sınıfına aşağıdaki problemleri örnek olarak verebiliriz:

- *Vertex-cover problemi*: Bir grafta belirli bir sayıda düğüm içeren ve içerdiği tüm düğümler birbirine bağlı olan alt graflar bulunabilir mi?
- *Hamilton döngü problemi*: Bir graftaki düğümlerden sadece bir kere geçerek başlanılan düğüme geri dönen bir yol bulunabilir mi?
- *Graf boyama*: Bir grafta bir yol ile bağlı olan düğümlerin renkleri farklı olacak şekilde en az kaç renkle boyayabiliriz?

NP sınıfındaki bazı problemler için polinomsal zamanlı algoritmalar bulmak çok uzun çabalar sonunda sonuçsuz kalmıştır. Bu tür problemlerin sınıfına *NP-Complete* adı verilir. Yani, bir probleme NP-Complete denmişse bu problem için polinomsal zamanlı bir algoritmanın bulunmasının çok zor olduğu anlaşılmalıdır[14].



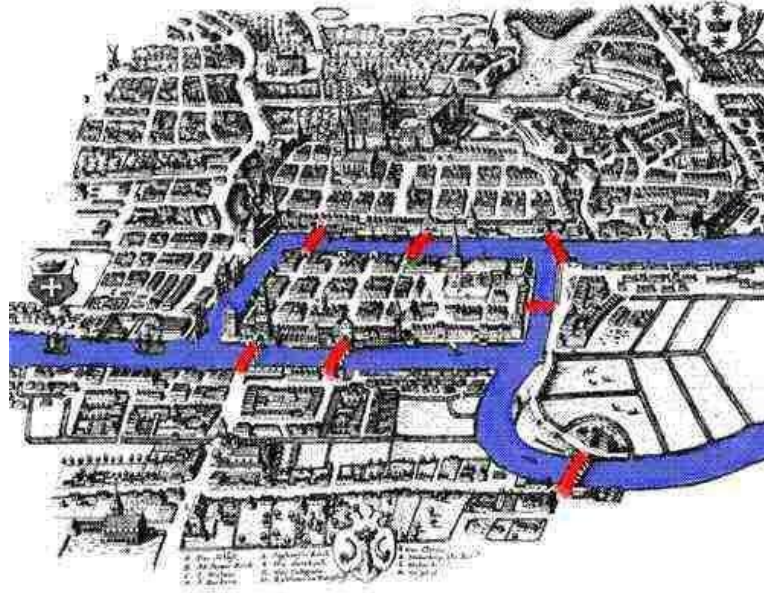
Şekil 3.1 Problemlerin karmaşıklık sınıflarının Venn şeması ile gösterimi

BÖLÜM 4

GRAF TEORİSİ

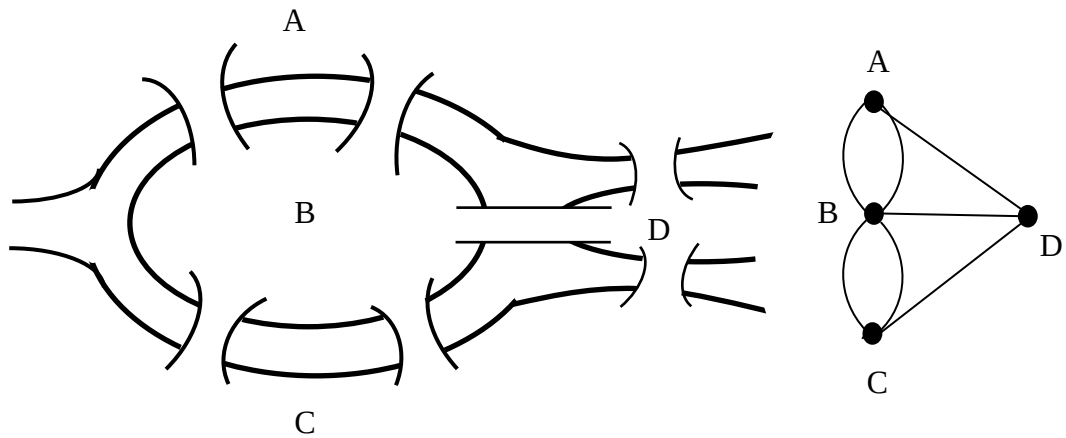
Bu bölümde Graf teorisi ile ilgili temel bilgiler ve ileride kullanacağımız bazı tanım ve örnekler vereceğiz.

Graf teori 1736'da Euler'in Königsberg köprü problemini çözmesi ile başladı diyebiliriz. Bu problemde Şekil 4.1'de görüldüğü gibi bir nehir(Pregel nehri), ortasındaki iki ada ve adaları birbirlerine ve nehrin kenarlarına bağlayan 7 köprüden oluşur. Problem, bu şekildeki herhangi bir kara parçasından başlayarak bütün köprülerden geçerek ama hiçbir köprüden iki kere geçmeden başlangıç noktasına varılabilir mi? sorusudur. Euler bu şekil için kara parçalarını düğümle, köprüleri de kenarlarla göstererek Şekil 4.2'deki gibi bir graf modeli çıkararak problemde aranan yolun olmadığını ispatlamıştır[9].



Şekil 4.1 Pregel nehri[16]

Tanım 4.1 V bir sonlu küme olsun. V 'nin farklı elemanlarının ikililerinin kümesi olan $E(V)=\{(u,v)|u,v \in V, u \neq v\}$ kümesini verelim. $E \subseteq E(V)$ olmak üzere $G=(V,E)$ ikilisine V üzerinde bir *basit graf* denir. V 'nin elemanlarına G 'nin *düğümüleri*, E 'nin elemanlarına da G 'nin *kenarları* denir. G 'nin düğümlerinin kümesi V_G ile kenarlarının kümesi ise E_G ile gösterilir. $u, v \in V_G$ olmak üzere $(u,v) \in E_G$ ise u ve v düğümleri birbirine *komşudur* denir. Ayrıca bir v düğümünün *komşuluk kümesini* $B(v)=\{u|(u,v) \in E_G\}$ gösterebiliriz[9].



Tanım 4.2 $|V_G|$ 'ye G grafinin *derecesi*, $|E_G|$ 'ye de G grafinin *boyutu* denir[9].

Tanım 4.4 Bir gratta herhangi bir düğüm kendisine giderse yani $(v,v) \in E_G(döngü)$ olursa ve herhangi iki düğüm arasında iki kenar olursa(*paralel kenarlar*) bir $G=(V,E)$ *çoklagraf*’ı elde etmiş oluruz[9].

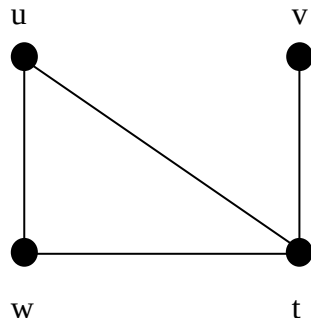
11

Bir yönlü G grafında (u,v) kenarı ile (v,u) kenarı birbirinden farklıdır.

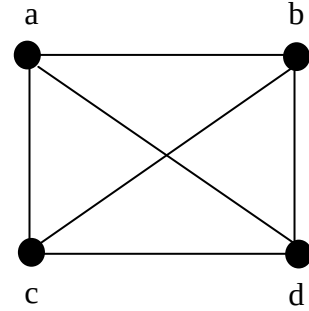
Bir G grafi iki boyutlu bir düzlem üzerinde düğümleri noktalarla ve kenarları E 'nin elemanı olan ikililerin arasına çizilecek çizgilerle belirterek bir diyagram ile gösterebiliriz.

Tanım 4.6 Bir yönlü G grafında $v \in V_G$ düğümü için bu düğüme doğru gelen düğümler kümesine v 'ye gelen düğümler kümesi denir ve $B^+(v)$ ile gösterilir. Diğer taraftan bu v düğümünün gittiği düğümler kümesine de v 'den giden düğümler kümesi denir ve $B^-(v)$ ile gösterilir. Yönsüz graflarda herhangi bir düğüm için gelen düğümler kümesi ile giden düğümler kümesinin birbirine eşit ve komşuluk kümesi olduğu açıktır, yani $B^+(v) = B^-(v) = B(v)$ 'dir[8].

Örnek 4.1 $V = \{u,v,w,t\}$ olsun. Bu durumda $E(V) = \{(u,v), (u,w), (u,t), (v,w), (v,t), (w,t)\}$ olur. $E \subseteq E(V)$ olacak şekilde $E = \{(u,w), (v,t), (u,t), (w,t)\}$ ile V , bir $G = (V, E)$ grafi oluştururlar. Bu grafi Şekil 4.3'deki gibi bir diyagram ile gösterebiliriz.



Şekil 4.3 Örnek 4.1'deki grafin diyagramı



Şekil 4.4 Tam grafa bir örnek

Tanım 4.7 Bir $G(V, E)$ grafında $E = E(V)$ ise bu grafa *tam graf* denir. Tam grafta her iki düğüm arasında kenar vardır[9]. Şekil 4.4'deki graf 4 düğümlü bir tam graf örneğidir.

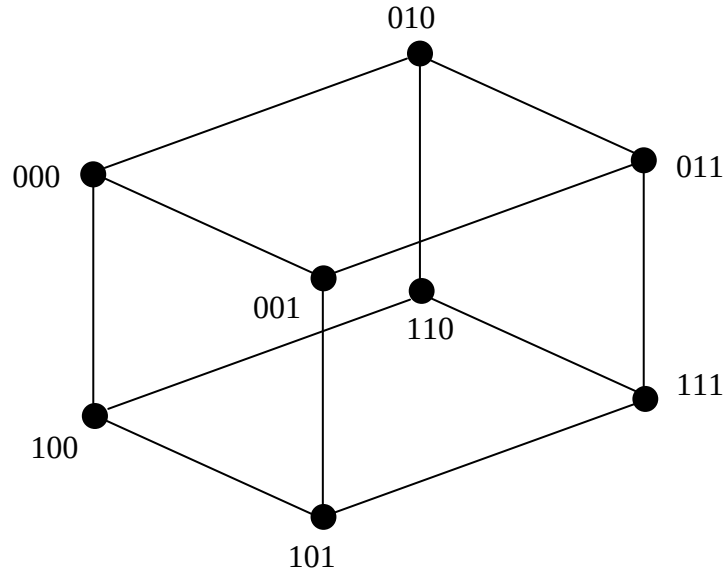
Tanım 4.8 Bir $G(V, E)$ grafının için düğümlerini $V_G = \{v_1, v_2, \dots, v_n\}$ şeklinde sıralı olarak verelim. $n \times n$ 'lik bir M matrisini düşünelim. Bu M matrisinin elemanlarını, eğer v_i ve v_j komşu ise $M_{ij} = 1$, değilse $M_{ij} = 0$ şeklinde oluşturalım. Oluşturulan bu M matrisi G 'nin *komşuluk matrisi* olarak adlandırılır. Bu matrisin simetrik olduğu aşikârdır[9].

Örnek 4.1'deki grafin komşuluk matrisi, $M = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$ 'dir.

Tanım 4.9 Bir $G(V,E)$ grafi için $V_H \subseteq V_G$ ve $E_H \subseteq E_G$ olacak şekilde bir H grafi varsa bu H grafına G 'nin *alt grafi* denir ve $H \subseteq G$ şeklinde gösterilir. Özel olarak, $V_H = V_G$ ise H alt grafi G 'yi *gerer* denir. Ayrıca bir G grafindan bazı düğümleri ve bu düğümlere bağlı tüm kenarları çıkartarak oluşturduğumuz grafa *indirgenmiş graf* denir[9].

Tanım 4.10 Bir $G(V,E)$ grafinin tüm düğümlerinin dereceleri aynı ise bu grafa *düzenli graf* denir. Düğümlerin derecesi r ise bu grafa *r-düzenli graf* ta denilebilir [10].

Tanım 4.11 $k \geq 1$ olmak üzere k uzunluklu ikili katarların kümesini B^k ile gösterelim. Örneğin $B^3 = \{000, 001, 010, 100, 011, 101, 110, 111\}$ 'dir. Düğümleri B^k 'nin elemanları ile gösterilen ve birbirine komşu olan düğümler arasında sadece bir katar farklı ise bu grafa *k-küp graf* denir ve Q_k ile gösterilir. Burada $V_{Q_k} = B^k$ 'dir[9]. Şekil 4.5'de 3-küp grafını görebilirsiniz.



Şekil 4.5 3-küp grafi

Tanım 4.12 $e_i = (v_i, v_{i+1}) \in E_G$ G grafinin bir kenarı olmak üzere için $W = e_1 e_2 \dots e_k$ 'ya bir *k uzunluklu bir yürüyüş(walk)* denir. Burada yürüyüş $W = v_1 v_2 \dots v_{k+1}$ ile de gösterilebilir.

Bir W yürüyüşünde $v_1=v_{k+1}$ ise bu yürüyüşe *kapalı yürüyüş(closedwalk)* denir[9].

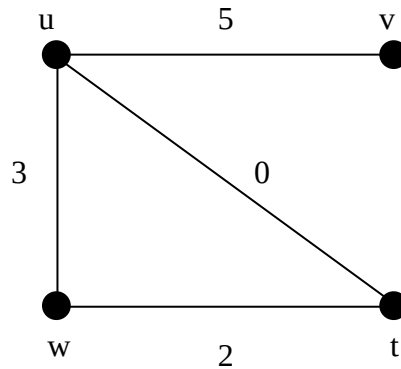
Bir yürüyüşte her $i \neq j$ için $v_i \neq v_j$ ise bu yürüyüşe bir *yol(path)* denir. Bir yolun düğüm sayısı tek ise bu yola *tek dereceli yol*, çift ise *çift dereceli yol* denir[9].

Bir kapalı yürüyüşte $v_1=v_{k+1}$ hariç her $i \neq j$ için $v_i \neq v_j$ ise bu kapalı yürüyüşe *döngü(cycle)* denir. Bir döngünün düğüm sayısı tek ise bu döngüye *tek dereceli döngü*, çift ise *çift dereceli döngü* denir[9].

Tanım 4.13 $u, v \in G$ olmak üzere u 'dan v 'ye bir yol varsa, $d_G(u, v) = \min\{k | u \xrightarrow{k} v\}$ 'ye u ile v arasındaki *uzaklık* denir. Eğer u ile v arasında herhangi bir yol yoksa $d_G(u, v) = \infty$ 'dur denir[9].

Bir G grafında, her $u, v \in G$ için $d_G(u, v) < \infty$ ise bu G grafına *bağlı graf(connected)* denir[9].

Tanım 4.14 Kenarlarına $\alpha: E_G \rightarrow \mathbb{R}$ gibi bir fonksiyonla sayısal değerler verilmiş G grafına *ağırlıklı graf(weighted)* denir. Ayrıca tüm kenarların ağırlıkları toplamına *grafın ağırlığı* denir[9]. Şekil 4.6 ağırlıklı grafa bir örnektir. Bu grafın ağırlığı 10'dur.



Şekil 4.6 Ağırlıklı grafa bir örnek.

Tanım 4.15 Bir G grafında tüm kenarlar V_G 'nin X ve Y gibi iki alt kümesini bağlıyorsa, yani X 'teki düğümlerden herhangi ikisini birbirine ve Y 'deki düğümlerden herhangi ikisini birbirine bağlayan kenar yoksa bu G grafına *iki taraflı graf(bipartite)* denir. Burada X ve Y G grafının birer tarafı olurlar ve G 'ye (X, Y) -iki taraflı denebilir[9].

Tanım 4.16 Bir G grafi kenarları birbirini kesmeyecek şekilde çizilebiliyorsa bu grafa *düzlem graf(planar)* denir[9].

Bir graftaki düğümlerin herhangi F alt kümesini düşünelim. Bu F 'deki herhangi iki düğüm bir döngü sayesinde birbirlerine ulaşabiliyorsa bu F 'ye graf üzerinde bir *yüzey* denir.

Şimdi Euler'in, bir grafın düzlem graf olup olmaması ile ilgili bir teoremini vereceğiz.

Teorem 4.1(Euler'in formülü) G bir bağlı düzlem graf olsun. Bu grafın düğüm sayısı v , kenar sayısı e , bu graf üzerindeki yüzey sayısı t olmak üzere $v-e+t=2$ 'dir[9].

Bizim konumuz ile alakalı olmadığından dolayı burada bu teoremin ispatını vermiyoruz.

TANIMLAR VE İLGİLİ ÇALIŞMALAR

Bu kısımda çalışmamızda kullanacağımız terimler ile ilgili genel bilgiler verip çalışmamızla alakalı araştırmalar hakkında kısaca bahsedeceğiz.

5.1 Belirleme Kodları

Bir G grafinin düğümlerinin bir alt kümesi C olsun. Bu C kümesinin G grafindaki tüm gelen düğümler kümeleri ile kesişimleri birbirinden farklı ise bu C alt kümesine bu G grafi için bir *belirleme kodudur(identifyngcode)* denir. Bir C belirleme kodunun herhangi bir düğümün gelen düğümler kümesi ile kesişimine bu v düğümünün *belirleme kümesi(identifyng set)* denir, yani $I_C(v)=B^+(v) \cap C$ 'dir. Bu durumda $\emptyset \neq C \subseteq V_G$ olacak şekilde bir C kümesi için $\forall u,v \in V_G$ için, $u=v \leftrightarrow I_C(u)=I_C(v)$ olur. Burada C 'ye kod C 'nin elemanı olan düğümlere de kodsözler denebilir[8].

5.2 Belirlenebilen Graflar

Şu ana kadar belirleme kodunu bulacağımız graflarla ilgili herhangi bir şart söylemedik. Ancak buradan her graf için belirleme kodu bulabileceğimiz çıkarılmamalı. Çünkü her graf için belirleme kodu yoktur. Bir grafin belirleme kodu varsa bu grafa *belirlenebilir graf(distinduishable graph)* denir. Aksine belirleme kodu bulunamıyorsa bu grafa *belirlenemez graf(indistinguishable graph)* denir. Bir grafin tüm gelen düğüm kümeleri birbirinden farklı ise bu graf belirlenebilirdir. Fakat en az iki gelen düğüm kümesi birbirine eşit ise belirleme kodlarının tanımından bu grafin belirleme kodunun bulunamayacağı açıktır. Çünkü bu durumda gelen düğüm kümeleri aynı olan grafların belirleme kümeleri de aynı olacaktır. Böylece bu tür bir grafin belirleme kodu bulunamayacağı anlaşılır[2].

Bundan sonraki kısımlarda aksi söylenmedikçe graflarımız belirlenebilir varsayılacaktır.

5.3 Belirleme Kodları ile ilgili Alt Sınır, Üst Sınır Tahmini

Bir polinomsal zamanlı tahmin geliştirmek için ilk çalışmalardan bazıları [2,17]'te yapıldı. Çok yakın zamanlarda, birkaç grup belirleme kodları ve dominating-locating kümelerin tahmin edilebilirliği sorusunu araştırdı [18,19], tahmin sonuçlarının zorluğunu ve bir $O(\log|V|)$ zaman orantısındaki optimal belirleme kodu tahmin eden polinomsal zamanlı algoritmaları verdiler.

5.4 Küme Kaplama Problemi

U bir m elemanlı taban küme ve S de U 'nun altkümelerinin bir ailesi olsun. Bir $C \subseteq S$ kaplaması, birleşimleri U olan altkümelerin bir ailesidir. Bu C alt küme ailesine U için bir *kaplama(cover)* denir. Küme kaplama problemi(covering problem) en küçük elemanlı bir C kaplamasını bulmaya yarar. Küme kaplama problemi uzun zamandır çalışılan problemlerden birisidir. Bu problemin NP-Complete olması bize şu hırslı(greedy) algoritmayı uygulamamıza izin verir: U kaplanıncaya kadar her bir adımda S 'deki, kaplanmamış en çok elemanı kaplayan küme seçilir[8].

Hırslı küme kaplama algoritmasının performans oranı çalışıldı. Lovazs'ın çalışması [20], $s_{\min}$ =minimum kaplama; $s_{\text{hırslı}}$ =hırslı kaplama; m de taban kümenin eleman sayısı olmak üzere, $s_{\text{hırslı}}/s_{\min} = \Theta(\ln m)$ olduğunu gösterdi. Daha sonraları Slavik bu oranı geliştirerek [21], performans oranında alt sınır ve üst sınır arasında 1.1'den daha küçük bir farka ulaştı.

5.5 Çoklu Küme Kaplama Problemi

Minimum k -çoklu küme kaplama problemi(minimum k -set multicover problem), verilen bir (U,S) çifti için U 'daki her bir elemanı en az k kere kaplayan S 'nin en küçük alt kümesini arama olan bu problem minimum küme kaplama probleminin bir genelleştirmesidir[8].

Çoklu küme kaplama problemi, küme kaplama problemi için benzer bir hırslı algoritmaya izin verir: her bir adımda k -çoklu kaplamanın elemanı olmayan en fazla elemanı kaplayan kümeyi seçer. α en büyük kümenin eleman sayısı olmak üzere, bu algoritmanın performans garantisinin üst sınırı $1+\log \alpha$ ile sınırlandırılır[8].

BELİRLEME KODLARI VE KÜME KAPLAMA PROBLEMİ

Bu bölümde belirleme kodlarından küme kaplamaya, küme kaplamadan da belirleme kodlarına birer indirgeme kuracağız.

Küme kaplama problemi ile Belirleme kodu problemi arasındaki benzerliği aşağıdaki şekilde verebiliriz.

Çizelge 6.1 Küme Kaplama ve Belirleme Kodu Problemleri arasındaki ilişki

	<u>Küme Kaplama</u>	<u>Belirleme Kodları</u>
Çalışılan Bölge	Bir U taban kümesinin altkümelerinin bir S ailesi	Bir $G(V,E)$ grafi
Aranan	$\bigcup_{s \in S'} s = U$ olacak şekilde bir $S' \subseteq S$ kümesi.	G 'nin belirleme kodu olan bir $C \subseteq V$ kümesi.
Ölçü	Kaplamanın eleman sayısı: $ S' $	Belirleme Kodunun eleman sayısı: $ C $

6.1 ID-KOD $\leftarrow$ KÜME KAPLAMA

Öncelikle herhangi bir graf için bir belirleme kodu bulma işleminden küme kaplama bulmaya bir geçiş vereceğiz. Bu geçişi göstermek için ilk olarak ana teoremi verip birkaç tanım ve yardımcı teorem yardımıyla bu teoremin ispatını yapacağız.

Teorem 6.1n düğümlü bir G grafi için bir belirleme kodu bulma, n uzunluklu ikili vektörlerde $O(n^3)$ işlemleri ile (skaler çarpımlar, toplamalar veya karşılaştırmalar)

$\frac{n(n-1)}{2}$ elemanlı bir küme üzerinde bir küme kaplama çözümünden daha fazla işlem gerektirmez[8].

Burada bir graftan küme kaplama problemine indirgeme yapmak için, U taban kümesini farklı düğümlerin tüm çiftlerinden oluşan küme ve S alt küme ailesini de v herhangi bir düğüm olmak üzere v'yi gelen düğümler kümesinde bulunduran tüm düğümlerin çiftlerinin oluşturduğu küme S_v olacak şekilde $S=\{S_{v \in V}\}$ kümesi oluşturulur. Böylelikle bir U taban kümemizi ve S alt küme ailemizi oluşturmuş olduk. Şimdi indirgemeye bazı notasyon ve tanımlarla başlayalım.

Tanım 6.1 $D(u,v)$ *farklılık kümesi*, $u,v \in V$ düğümlerinin gelen düğümler kümeleri arasındaki simetrik fark olarak tanımlanır:

$$\begin{aligned} D(u,v) &\doteq B^+(u) \Delta B^+(v) \\ &= [B^+(u) - B^+(v)] \cup [B^+(v) - B^+(u)] \end{aligned}$$

Ayrıca C kodu ile $D(u,v)$ 'nin kesişimini $D_C(u,v)$ ile göstereceğiz, yani $D_C(u,v) = D(u,v) \cap C$ [8].

$D_C(u,v)$, $u,v \in V$ 'nin belirleme kümelerinin simetrik farkıdır diyebiliriz. Çünkü belirleme kodlarının tanımında C özel olarak verilmediyse tüm düğümlerin kümesi olarak tanımlanabilir demiştik. O halde $I_C(u) = B^+(u) \cap C$ idi, buradan C yerine V alırsak $I_C(u) = B^+(u) \cap V = B^+(u)$ olur, yani, $D_C(u,v) = I_C(u) \Delta I_C(v)$ alabiliriz.

Buna ek olarak U 'yu tüm farklı düğümlerin çiftlerinin kümesi olarak göstereceğiz, $U = \{(u,v) | u \neq v \in V\}$ [8].

Son olarak bir $v \in V$ düğümünün *ayrım kümesi*, $v \in D(u,z)$ olacak şekilde (u,z) 'lerden oluşur;

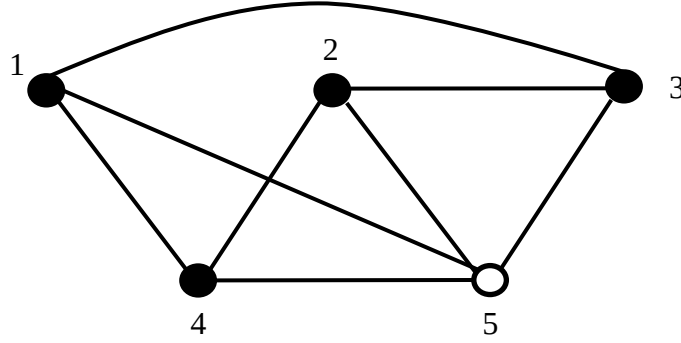
$$\delta_v = \{(u,z) \in U | v \in D(u,z)\} \text{ [8].}$$

Örnek 6.1 Şekil 6.1'deki graf için $C = \{1, 2, 3, 4\}$ bir belirleme kodudur. Bu graf yönsüz bir graf olduğu için $B^+(u) = B^-(u) = B(u)$ 'dur ve

$B^+(1) = \{1,3,4,5\}$, $B^+(2) = \{2,3,4,5\}$, $B^+(3) = \{1,2,3,5\}$, $B^+(4) = \{1,2,4,5\}$, $B^+(5) = \{1,2,3,4,5\}$ elde edilir. Buradan bu graf için farklılık kümeleri,

$D(1,2)=\{1,2\}$, $D(1,3)=\{2,4\}$, $D(1,4)=\{2,3\}$, $D(1,5)=\{2\}$, $D(2,3)=\{1,4\}$, $D(2,4)=\{1,3\}$, $D(2,5)=\{1\}$, $D(3,4)=\{3,4\}$, $D(3,5)=\{4\}$, $D(4,5)=\{3\}$ olarak bulunur. Ayrıca ayırım kümelerini bulmak istersek,

$\delta_1=\{(1,2),(2,3),(2,4),(2,5)\}$, $\delta_2=\{(1,2),(1,3),(1,4),(1,5)\}$, $\delta_3=\{(1,4),(2,4),(3,4),(3,5)\}$, $\delta_4=\{(1,3),(2,3),(3,4),(3,5)\}$, $\delta_5=\{\}$ şeklinde elde edilir.



Şekil 6.1 Örnek 6.1'in grafi

Ayırım kümesinin C kodundan bağımsız olduğuna dikkat edelim.

Sıradaki yardımcı teorem bir belirleme kodunun tanımından basit bir şekilde çıkarılabilir.

Lemma 6.1 Bir C kodunun belirleme kodu olabilmesi için gerek ve yeter koşul $\emptyset \notin \{D_C(u,z) | (u,z) \in U\}$ olmasıdır[8].

İspat:

$\Rightarrow$: Tanımdan yola çıkarak, C bir belirleme kodu ise $I_C(v_i)$ 'lerin hepsi birbirinden farklı olmalı. O halde $I_C(v_i) \Delta I_C(v_j)$ 'ler boştan farklıdır. Çünkü $I_C(v_i)$ 'lerin hepsi birbirinden farklı ise $I_C(v_i) - I_C(v_j)$ ve $I_C(v_j) - I_C(v_i)$ kümeleri boştan farklı olmak zorundadır. Bu durumda $\emptyset \notin I_C(v_i) \Delta I_C(v_j)$. $D_C(v_i, v_j) = I_C(v_i) \Delta I_C(v_j)$ ise $\emptyset \notin D_C(v_i, v_j)$ diyebiliriz. Buradan da $\emptyset \notin \{D_C(u,z) | (u,z) \in U\}$ olur.

$\Leftarrow$: $\emptyset \notin \{D_C(u,z) | (u,z) \in U\}$ olsun. Böylece $\emptyset \neq D_C(u,v)$ olur, yani $\emptyset \neq I_C(u) \oplus I_C(v)$ 'dir. Bu durumda $I_C(v_i)$ 'ler birbirlerinden farklıdır ve bütün $I_C(v_i)$ 'ler birbirinden farklı ise C bir belirleme kodudur.

Lemma 6.1'i bir örnek üzerinde somut bir şekilde görelim.

Örnek 6.2 Bir önceki örnek için $D_C(u,v)$ 'leri bulalım.

$D_C(1,2)=\{1,2\}$, $D_C(1,3)=\{2,4\}$, $D_C(1,4)=\{2,3\}$, $D_C(1,5)=\{2\}$, $D_C(2,3)=\{1,4\}$, $D_C(2,4)=\{1,3\}$, $D_C(2,5)=\{1\}$, $D_C(3,4)=\{3,4\}$, $D_C(3,5)=\{4\}$, $D_C(4,5)=\{3\}$. Görüldüğü gibi $D_C(u,v)$ 'lerin hiçbirisi $\emptyset$ değil. Diğer taraftan belirleme kodu olmadığını bilmediğimiz $C=\{1,2,3\}$ 'ü alıp buna göre $D_C(u,v)$ 'leri bulalım.

$D_C(1,2)=\{1,2\}$, $D_C(1,3)=\{2\}$, $D_C(1,4)=\{2,3\}$, $D_C(1,5)=\{2\}$, $D_C(2,3)=\{1\}$, $D_C(2,4)=\{1,3\}$, $D_C(2,5)=\{1\}$, $D_C(3,4)=\{3\}$, $D_C(3,5)=\{\}$, $D_C(4,5)=\{3\}$. Burada da $D_C(3,5)=\{\}$ 'dir. Lemma.1'den $C=\{1,2,3\}$ 'nin belirleme kodu olmadığı anlaşılıyor.

Anlattıklarımıza ek olarak bir belirleme kodunu ayrım kümelerinin açısından tanımlayabiliriz.

Lemma 6.2 C 'nin belirleme kodu olması için gerek ve yeter koşul C 'deki düğümlerin ayrım kümelerinin ailesinin $U=\{(u,z) \in V^2 | u \neq z\}$ kümesini kaplamasıdır[8].

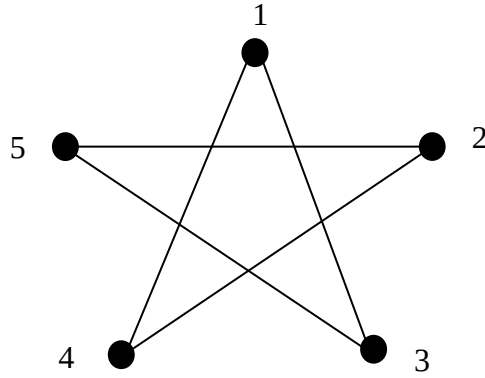
İspat: Lemma 6.1'den C için bir belirleme kodunun tüm farklılık kümelerinin en az bir elemanı olmalı. Ayrım kümelerinin tanımından $(u,v) \in \delta_c$ olacak şekilde herhangi $(u,v) \in U$ için bazı $c \in C$ olduğu çıkarılabilir. Bu yüzden $\bigcup_{c \in C} \delta_c = U$ olur.

Diğer taraf benzer şekilde gösterilebilir.

Teorem 6.1'in İspatı: Bir belirleme kodunun aşağıdaki yapısını düşünelim;

$ID(G) \rightarrow C$

- 1) Gelen düğüm kümelerini bul; $\{B^+(u) | u \in V\}$
- 2) Ayrım kümelerini bul; $\partial = \{\delta_u | u \in V\}$
- 3) (U, ∂) 'nin minimum küme kaplamasını bul; C
- 4) Sonuç olarak $\{u \in V | \delta_u \in C\} \rightarrow C$, yani minimum küme kaplamadaki ayrım kümelerine karşılık gelen düğümleri elde ederiz.



Şekil 6.2 Teorem 6.1’i desteklemek için incelenen örnek graf

Sonuç kodu C ’nin Lemma 6.2 tarafından belirleme kodu olduğu garantilendi ve 3. adım küme kaplamanın optimalliği yani daha küçük bir belirleme kodu bulunamayacağı garantilenir. İspatın tamamlanması için belirleme kümelerinin hesaplanması amacıyla ikili vektörlerin $\Theta(n^2)$ toplamalarının gerektiği ve ∂ ’nın hesaplanması için $|U|$ ’da her $\frac{n(n-1)}{2}$ eleman için n tane işlem gerektiğini gözlemledik[8].

Teorem 6.1’i desteklemek için basit bir örnek olarak Şekil 6.2’deki grafi düşünelim. Düğümün belirleme kümeleri ve ayrım kümeleri:

Çizelge 6.2 Şekil 6.2’deki grafin düğümlerinin belirleme ve ayrım kümeleri

$v \in V$ $B^+(v)$ δ_v

1	{1,3,4}	{(1,2),(1,5),(2,3),(2,4),(3,5),(4,5)}
2	{2,4,5}	{(1,2),(1,4),(1,5),(2,3),(3,4),(3,5)}
3	{1,3,5}	{(1,2),(1,4),(2,3),(2,5),(3,4),(4,5)}
4	{1,2,4}	{(1,3),(1,5),(2,3),(2,5),(3,4),(4,5)}
5	{2,3,5}	{(1,2),(1,3),(1,5),(2,4),(3,4),(4,5)}

Buna karşılık gelen küme kaplama problemi bir $\{(u,z)|1 \leq u \neq z \leq 5\}$ taban kümesinin yerine konabilirdi, bu durumda altküme ailesi tablodaki δ_v 'nin tamamını oluştururdu[8].

6.2 KÜME KAPLAMA ← ID-KOD

Bu kısımda da Küme kaplama probleminden Belirleme kodu bulmaya bir geçiş vereceğiz. Burada da öncelikle ana teoremi verip, özel bir yapı ve bu yapının özellikleri üzerinden bu teoremin ispatını yapacağız.

Teorem 6.2 Verilen bir m elemanlı U taban kümesi ve $s=|S|$ kardinaliteli altkümelerin bir S ailesi için, optimal küme kaplama bulmanın, toplamsal $O(m+s)$ işlemleriyle $n \leq 2\max(s,m+1)+\lg(m+1)$ düğümlü bir yönlü graf üzerinde bir belirleme kodu bulmaktan daha fazla işlem gerektirmez[8].

Sonuç 6.1 En az 4 elemanlı bir taban küme ile bir optimal küme kaplama bulma, toplamsal $O(ms^2)$ işlemleriyle $2m+\max(s,m+1) \leq n \leq 3m+\max(s,m+1)$ düğümlü yönsüz bir grafın üzerinde bir belirleme kodu bulmaktan daha fazla işlem gerektirmez[8].

Teorem.2'yi ispatlamak için, küme kaplama probleminin bir örneğinden özel bir yönlü grafı (indirgemedede kullanılacak olan) sağlayıp analiz edeceğiz.

Yapı 6.1 $U=\{u_1, u_2, \dots, u_m\}$ bir taban küme ve $S=\{S_1, S_2, \dots, S_{m+1}\}$ U'nun altkümelerinin bir ailesi olsun. $n=2m+1$ olmak üzere $V=\{v_1, v_2, \dots, v_n\}$ düğümleri ve E kenarları ile öyle bir $G(V,E)$ grafı inşa ediyoruz ki;

1) $f:U \rightarrow V$ 1-1 ve her $i \leq m$ için $f(u_i)=v_i$ ve $f(S)=\bigcup_{u \in S} f(u)$ ($S \subset V$) olduğunda her bir

$v_1, v_2, \dots, v_{m+1}$ düğümlerinin giden düğümler kümeleri,

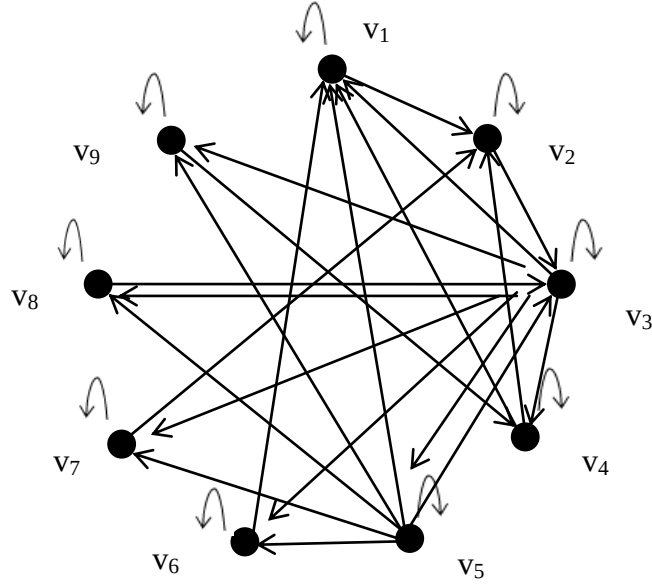
$$B^-(v_i) = \begin{cases} f(S_i) & , u_i \in S_i \\ V - f(S_i), u_i \notin S_i \end{cases}$$

şeklinde oluşturulur.

2) Geriye kalan $v_{m+2}, v_{m+3}, \dots, v_n$ düğümleri,

$$B^-(v_i) = \{v_i, v_{i-m-1}\} \text{ şeklinde oluşturulur[8].}$$

$$U=\{u_1, u_2, u_3, u_4\} \quad S=\{S_1=\{u_1, u_2\}, S_2=\{u_2, u_3\}, S_3=\{u_2\}, S_4=\{u_1, u_2, u_4\}, S_5=\{u_2, u_4\}\}$$



Şekil 6.3 İndirgememize bir örnek.

Şimdi $G(V,E)$ grafi için keyfi bir C belirleme kodunun birkaç özelliğini vereceğiz; bu özellikleri okurken basit bir örnek üzerinde gösterdiğimiz yapıyı Şekil 6.3'den takip etmeniz sizin için faydalı olacaktır. $D(v_i, v_j)$ gösteriminin $(v_i, v_j) \in V$ çiftinin farklılık kümesi, δ_v gösteriminin de v düğümünün ayırım kümesini belirttiğini hatırlayalım. Yeni olarak $\vec{U} = \{(v_i, v_{i+m+1}) | i \leq m\}$ gösterimini vereceğiz ve buna karşılık $\vec{\delta} = \delta \cap \vec{U}$ operatörünü vereceğiz[8].

Farklılık kümeleri ve ayırım kümeleri yardımıyla Şekil 6.3'deki örneğin belirleme kodlarını bulalım. Burada kolaylık olması için düğümleri sadece numaraları ile belirteceğiz.

$$D(1,2)=\{2,3,5,6,7\}$$

$$D(1,3)=\{1,2,4,6,8\} \quad D(2,3)=\{1,3,4,5,7,8\}$$

$$D(1,4)=\{1,5,6,9\} \quad D(2,4)=\{1,2,3,7,9\} \quad D(3,4)=\{2,4,5,8,9\}$$

$$D(1,5)=\{1,4,6\} \quad D(2,5)=\{1,2,3,4,5,7\} \quad D(3,5)=\{2,8\} \quad D(4,5)=\{4,5,9\}$$

$$D(1,6)=\{1,4\} \quad D(2,6)=\{1,2,3,4,5,6,7\} \quad D(3,6)=\{2,6,8\} \quad D(4,6)=\{4,5,6,9\}$$

$$D(1,7)=\{1,4,6,7\} \quad D(2,7)=\{1,2,3,4,5\} \quad D(3,7)=\{2,7,8\} \quad D(4,7)=\{4,5,7,9\}$$

$$D(1,8)=\{1,4,6,8\} \quad D(2,8)=\{1,2,3,4,5,7,8\} \quad D(3,8)=\{2\} \quad D(4,8)=\{4,5,8,9\}$$

$$D(1,9)=\{1,4,6,9\} \quad D(2,9)=\{1,2,3,4,5,7,9\} \quad D(3,9)=\{2,8,9\} \quad D(4,9)=\{4,5\}$$

$$D(5,6)=\{6\}$$

$$D(5,7)=\{7\} \quad D(6,7)=\{6,7\}$$

$$D(5,8)=\{8\} \quad D(6,8)=\{6,8\} \quad D(7,8)=\{7,8\}$$

$$D(5,9)=\{9\} \quad D(6,9)=\{6,9\} \quad D(7,9)=\{7,9\} \quad D(8,9)=\{8,9\}$$

$$\delta_1=\{(1,3),(1,4),(1,5),(1,6),(1,7),(1,8),(1,9),(2,3),(2,4),(2,5),(2,6),(2,7),(2,8),(2,9)\}$$

$$\delta_2=\{(1,2),(1,3),(2,4),(2,5),(2,6),(2,7),(2,8),(2,9),(3,4),(3,5),(3,6),(3,7),(3,8),(3,9)\}$$

$$\delta_3=\{(1,2),(2,3),(2,4),(2,5),(2,6),(2,7),(2,8),(2,9)\}$$

$$\delta_4=\{(1,3),(1,5),(1,6),(1,7),(1,8),(1,9),(2,3),(2,5),(2,6),(2,7),(2,8),(2,9),(3,4),(4,5),(4,6),(4,7),(4,8),(4,9)\}$$

$$\delta_5=\{(1,2),(1,4),(2,3),(2,5),(2,6),(2,7),(2,8),(2,9),(3,4),(4,5),(4,6),(4,7),(4,8),(4,9)\}$$

$$\delta_6=\{(1,2),(1,3),(1,4),(1,5),(1,7),(1,8),(1,9),(2,6),(3,6),(4,6),(5,6),(6,7),(6,8),(6,9)\}$$

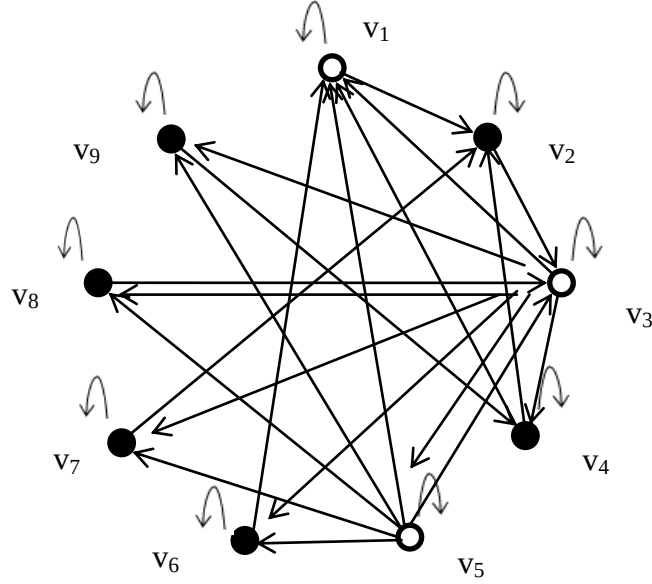
$$\delta_7=\{(1,2),(1,7),(2,3),(2,4),(2,5),(2,6),(2,8),(2,9),(3,7),(4,7),(5,7),(6,7),(7,8),(7,9)\}$$

$$\delta_8=\{(1,3),(1,8),(2,3),(2,8),(3,4),(3,5),(3,6),(3,7),(3,9),(4,8),(5,8),(6,8),(7,8),(8,9)\}$$

$$\delta_9=\{(1,4),(1,9),(2,4),(2,9),(3,4),(3,9),(4,5),(4,6),(4,7),(4,8),(5,9),(6,9),(7,9),(8,9)\}$$

(5,6), (5,7), (5,8), (5,9) ikililerini kaplamak için 6,7,8 ve 9'u kodun içine koymamız gerektiği açıktır. 6,7,8 ve 9'un kaplayamadığı sadece (1,6), (2,7), (3,8) ve (4,9)

ikilileridir. Bu ikilileri de tek seferde kaplayabilen düğüm yoktur. Bu yüzden iki düğümler geri kalan bu ikilileri kaplamaya çalışalım. 2 ve 4 düğümleri bu görevi yerine getirebiliyor. O halde minimal belirleme kodumuz $\{2,4,6,7,8,9\}$ 'dur(Şekil 6.4).



Şekil 6.4 Yapı 6.1'e verilen örneğin belirleme kodunun grafa yerleştirilmiş hâli. Burada içi dolu olanlar kodsözlerdir.

Özellik 6.1 $G(V,E)$ 'nin herhangi bir belirleme kodu tüm $v_{m+2}, \dots, v_n$ düğümlerini içermek zorunda[8].

İspat: Yapıdan dolayı $B^+(v_{m+1})$ $m+1$ 'den daha büyük indisli düğüm içermez, yani $B^+(v_{m+1}) \cap \{v_{m+2}, \dots, v_n\} = \emptyset$. Çünkü $B^+(v_{m+1})$ 'de $m+1$ 'den büyük eleman olması için v_{m+1} 'in $B^-(m+2), \dots, B^-(n)$ kümelerinden birisinde olması gerekiyor. Ancak bu yapıyı oluştururken bu kümeleri $B^-(i) = \{v_i, v_{i-m-1}\}$ ($m+2 \leq i \leq n$) olarak belirlemiştik. v_{m+1} 'in ilk bileşen olamayacağı açık; ikinci bileşen olması için $i-m-1=m+1$ olması gerekiyor. Buradan da $i=2m+2$ olması gerekir fakat i böyle bir değer alamayacağı için v_{m+1} ikinci bileşen de olamaz. Dolayısıyla v_{m+1} $B^-(m+2), \dots, B^-(n)$ kümelerinin hiçbirisinde bulunamaz, yani $v_{m+2}, \dots, v_n$ $B^+(m+1)$ 'de bulunamaz.

Diğerleri, yani $j \leq m+1$ için v_j 'nin $B^+(v_{m+1})$ içinde olması için gerek ve yeter koşul tüm $i \geq m+2$ için farklılık kümelerinin $D(v_{m+1}, v_i) = \{v_i\}$ olduğuna ulaşırız. Bu özelliğin ispatı için Lemma 6.2'den $\{v_i \mid i \geq m+2\}$ 'nin herhangi bir C belirleme kodu tarafından kapsanması gerektiğini kullanabiliriz[8].

Yapı 6.1 için vermiş olduğumuz örnekte belirleme kodunun 6,7,8 ve 9'u içerdiğini görebiliyoruz.

Özellik 6.2 Tüm $k \geq m+2$ 'ler için ayırım kümesi $\overrightarrow{\delta_{v_k}}$ boştur[8].

İspat: $\overrightarrow{\delta_{v_k}} = \delta_{v_k} \cap \overrightarrow{U}$ olduğundan dolayı $\overrightarrow{\delta_{v_k}} = \{(v_i, v_{i+m+1}) | i \leq m \text{ ve } v_k \in D(v_i, v_{i+m+1})\}$ yazabiliriz. Yapı gereği $k \geq m+2$ için $v_k \in B^+(v_k)$ 'dir. Ayrıca yapının tanımından $m+2 \leq i \leq n$ için $B^-(v_i) = \{v_i, v_{i-m-1}\}$ olduğunu biliyoruz. Buradan $B^-(v_k) = \{v_k, v_{k-m-1}\}$ olur. O halde $v_k \in B^+(v_{k-m-1})$ diyebiliriz. Burada $B^+(v_k)$ ile $B^+(v_{k-m-1})$ 'e dikkat edelim. $(v_{k-m-1}, v_k) \in \overrightarrow{U}$ 'dur. Bu durumda $v_k \in D(v_{k-m-1}, v_k)$ olacak şekilde bir (v_{k-m-1}, v_k) ya da başka bir ifadeyle (v_i, v_{i+m+1}) yoktur. Yani $\overrightarrow{\delta_{v_k}} = \emptyset$ 'dir.

Özellik 6.3 C'nin bir belirleme kodu olması için gerek ve yeter koşul $\{v_{m+2}, \dots, v_n\} \subseteq C$ ve $\{\overrightarrow{\delta_{v_i}} | i \leq m+1 \text{ ve } v_i \in C\}$ 'nin $\overrightarrow{U}$ için bir kaplama olmasıdır[8].

İspat:

$\Rightarrow$: Bu özelliğin ispatı için Lemma6.2'den $\overrightarrow{\delta_{v \in C}}$ 'nin $\overrightarrow{U}$ 'yu kapladığını hatırlayalım (U için δ ne ifade ediyorsa, $\overrightarrow{U}$ için de $\overrightarrow{\delta}$ kümesi onu ifade eder. Bu yüzden Lemma6.2 burada uygulanabilir). Ayrıca, Özellik6.2 tüm $k > m+1$ 'ler için $\overrightarrow{\delta_{v_k}} = \emptyset$ olduğunu verir ve kalanlar için Özellik 6.1 ile ispat tamamlanır.

$\Leftarrow$: Bu tarafın ispatı için, tüm farklılık kümelerinin boştan farklı olması özelliğindeki sondan iki durumu göstereceğiz, bu yüzden Lemma6.1 C'nin bir belirleme kodu olduğunu göstermeyi sağlar. İlk olarak $(v_i, v_j) \in \overrightarrow{U}$ olmak üzere $D_C(v_i, v_j)$ 'yi düşünelim; $\overrightarrow{U}$ 'daki öyle çiftleri düşünelim ki, bizim yapımız v_i 'nin ya $I_C(v_i)$ belirleme kümesinde ya da $I_C(v_j)$ 'de olmasını sağlar, bu yüzden tüm $(v_i, v_j) \in \overrightarrow{U}$ 'ler için $D_C(v_i, v_j) \neq \emptyset$ 'dir. $(v_i, v_j) \notin \overrightarrow{U}$ çiftleri ve tüm C'ler için, i için (istisnalar hariç) iki olasılığı düşünerek;

$$D(v_i, v_j) \cap \{v_{m+2}, \dots, v_n\} \neq \emptyset$$

olduğunu gözlemleyelim:

i) $i \leq m$, $I_C(v_i)$ v_{i+m+1} 'i içerir ve $I_C(v_j)$ içeremez veya

ii) $j > m+1$, $I_C(v_j)$ v_j 'i içerir ve $I_C(v_i)$ içeremez[8].

Özellik6.3'ün, ayırım kümelerini kullanarak belirleme kodları ve küme kaplamaları arasında birebir uyum olduğunu gösterir, buradan bir minimum belirleme kodu bir minimum küme kaplaması üretir. Şimdi bu özelliğin belirleme kodları ile orijinal S altkümeleri arasında ilişki kuran bir kullanımını vereceğiz.

$\vec{U}$ taban kümesi üzerindeki ayırım kümelerinin $\{\vec{\delta}_{v_i} \mid i \leq m+1\}$ ailesi U taban kümesi üzerindeki S altkümelerinin orijinal ailesine denktir. Gerçekten de $U = \{u_1, u_2, u_3, u_4\}$ kümesine $\vec{U} = \{(v_1, v_6), (v_2, v_7), (v_3, v_8), (v_4, v_9)\}$ kümesi karşılık gelir. Burada u_1 'e (v_1, v_6) , u_2 'e (v_2, v_7) , u_3 'e (v_3, v_8) , u_4 'e (v_4, v_9) karşılık geldiği görülür. Böylece $\vec{\delta}_{v_1} = \{(v_1, v_6), (v_2, v_7)\}$ kümesine $S_1 = \{u_1, u_2\}$, $\vec{\delta}_{v_2} = \{(v_2, v_7), (v_3, v_8)\}$ kümesine $S_2 = \{u_2, u_3\}$, $\vec{\delta}_{v_3} = \{(v_1, v_6), (v_3, v_8), (v_4, v_9)\}$ kümesine $S_3 = \{u_1, u_3, u_4\}$, $\vec{\delta}_{v_4} = \{(v_1, v_6), (v_2, v_7), (v_4, v_9)\}$ kümesine $S_4 = \{u_1, u_2, u_4\}$ ve $\vec{\delta}_{v_5} = \{(v_2, v_7), (v_4, v_9)\}$ kümesine $S_5 = \{u_2, u_4\}$ karşılık geldiğini görerek doğrulamış oluyoruz. Bunu özellik aşağıdaki lemmayı geliştirmek için kullanılacaktır.

Lemma 6.3 C 'nin, $G(V, E)$ 'nin bir belirleme kodu olması için gerek ve yeter koşul $\{v_{m+2}, \dots, v_n\} \subseteq C$ olması ve $\{S_i \mid v_i \in C, i \leq m+1\}$ 'in (U, S) 'nin bir küme kaplaması olmasıdır[8].

İspat: Özellik6.3'e dayanarak, $\vec{U}$ destek kümesi üzerinde $\{\vec{\delta}_{v_i} \mid i \leq m+1\}$ ayırım kümelerinin ailesi ile U destek kümesi üzerinde S altkümelerinin orijinal ailesi arasında birebir örtüşme olduğunu göstermemiz gerekiyor. i, j indisleri için $i \leq m+1$ ve $j \leq m$ olsun. Yapıdan dolayı eğer $u_j \notin S_i$ ise v_j ve v_{j+m+1} düğümlerinin ya ikisi birden $B^-(v_i)$ 'nin içindedir ya da ikisi birden $B^-(v_i)$ 'nin içinde değildir. Diğer taraftan eğer $u_j \in S_i$ ise ya v_j ya da v_{j+m+1} $B^-(v_i)$ 'nin içindedir. Bu da gösteriyor ki $(v_j, v_{j+m+1}) \in \vec{\delta}_{v_i}$ olması için gerek ve yeter koşul $u_j \in S_i$ olmasıdır ve ispat tamamlanmış oldu[8].

Teorem.2'nin ispatı: Verilen m boyutlu bir U taban kümesi ve s boyutlu bir S altkümelerinin ailesi için, aşağıdakileri sağlayan U' ve S' kümeleri üretilmiş oldu. Bunlar Yapı6.1 ile tam uyumludur;

i) Eğer $s < m+1$ ise, x , $m+x+1 \leq s+2^x$ eşitsizliğini sağlayan en küçük tamsayı olmak üzere, U' 'ya x adet yeni eleman ekleyerek U' üretilir. Bu yeni x adet

elemanın($x < 1 + \lg m + 1$ olduğunu hatırla) $x + 1 + m - s$ adet farklı altkümelerinin eklenmesi ile S' 'den S' türetilir.

ii) Eğer $s \geq m + 1$ ise, U' 'ya $s - m - 1$ adet eleman eklenerek U' türetilir ve bu $s - 1 - m$ adet eleman her bir S' 'ye eklenerek S'' 'ler türetilir.

Ayrıca Lemma 6.3, üretilen grafın, (U, S) 'nin bir minimum küme kaplaması için bir minimum belirleme kodu olduğunu sağlar[8].

6.3 Minimum Küme Kaplamadan Minimum Belirleme Koduna Geçiş Sağlayan Bir Yapı

Bu kısımda bir (U, S) küme kaplama probleminde minimum kaplamaya, bir $G = (V, E)$ grafi için minimum belirleme kodunu karşılık getiren bir yapı vereceğiz. Yapımızı inşa ederken işimize yarayacak bir Lemma'yı vererek başlayalım.

Lemma 6.4 Boş kümeden farklı m kardinaliteli bir topluluğu düşünelim, $M = [M_1, \dots, M_m]$, $\lg m + 2 \leq k \leq m$ olacak şekilde bir $U = \{u_1, \dots, u_k\}$ taban kümesi üzerinde olsun. $2m$ tane birbirinden farklı altkümelerin öyle bir $I = \{I_1, \dots, I_{2m}\}$ ailesi vardır ki;

- $\forall i \leq k$ için $u_i \in I_i$,
- $\forall i \leq m$ için $M_i = I_i \Delta I_{i+m}$ 'dir[8].

İspat: İspatımızda I' 'yı üreteceğiz. Öncelikle M' 'deki ilk kümeyi (M_1) düşünelim ve simetrik farkları $I_1 \Delta I_{1+m} = M_1$ olan I_1 ve I_{1+m} çiftlerini I' 'ya keyfî bir şekilde ekleyelim. En az $2^{\lg(m)+1} \geq 2m$ adet çift olduğuna dikkat edelim (çünkü I_1 için $2m$ tane seçenek vardır), çiftteki ilk küme u_1 'i kapsayan taban kümenin herhangi bir altkümesi olabileceğinden çiftteki ikinci küme istenen simetrik fark tarafından tek olarak saptanır. Tam olarak, $I_{2i} \Delta I_{2i+1} = M_i$ özelliği ile I' 'ya m tane farklı küme çiftini güvenli bir şekilde ekleyebiliriz. Yapımızda I' 'nın elemanlarını $\forall i \leq m$ için $M_i = I_i \Delta I_{i+m}$ olarak düzenleyeceğiz[8].

Yapı 6.2 $(U = \{u_1, \dots, u_m\}, S = \{S_1, \dots, S_{2m-k}\})$ bir küme kaplama problemi olsun. $S_{\min}$ minimum küme kaplama, $S_{\text{hırslı}}$ hırslı küme kaplama (hırslı algoritmayla bulunan küme kaplama) ve bunların kardinaliteleri $s_{\min}$ ve $s_{\text{hırslı}}$ olsun. Ayrıca $m = 2^k$ ($\lg(m) = k$) ve $s_{\min} \geq k + 2$ olduğunu varsayalım. İstisnalar hariç, $S_{\min}$ altkümesi S' 'de ilk olarak görülsün (yani S' 'nin ilk $s_{\min}$ tane elemanı $S_{\min}$ 'i oluştursun), $S_{\min} = \{S_1, \dots, S_{s_{\min}}\}$.

Şimdi (U, S) 'den, az sonra yazacağım özellikleri sağlayan bir G grafini üretelim. $\forall i \leq m$ için S_i kümesine karşılık gelen v_i düğümleriyle, bu graf $n=2m$ düğümlüdür. Bu grafin kenarlarını kararlaştırmak için iki topluluktan yararlanacağız:

- $M=[M_i]$, $M_i=\{v_j | S_j \in S_{\min} \text{ ve } u_i \in S_j\}$ olacak şekilde m tane kümenin topluluğudur (M_i kümesindeki v_j 'nin S_j 'ye karşılık gelen düğüm olduğuna dikkat edin). Yani, M_i , u_i 'yi kapsayan ve $S_{\min}$ 'de olan S_j 'lere karşılık gelen düğümlerden oluşuyor.
- $\underline{M}=[\underline{M}_i]$, $\underline{M}_i=\{v_j | S_j \notin S_{\min} \text{ ve } u_i \in S_j\}$ olacak şekilde m tane kümenin topluluğudur. Yani, M_i , u_i 'yi kapsayan ve $S_{\min}$ 'de olmayan S_j 'lere karşılık gelen düğümlerden oluşuyor.

$k > 1$ olmak üzere, Lemma 6.4 şu kümenin varlığını sağlar:

- $I=\{I_i\}$ kümesinde, $\{v_1, \dots, v_{s_{\min}}\}$ üzerinde (Lemma 6.4'teki k yerine $s_{\min}$; U kümesi yerine de $\{v_1, \dots, v_{s_{\min}}\}$ gelmiş oldu) $2m$ tane farklı küme vardır öyle ki;
 - $\forall i \leq m$ için $M_i = I_i \Delta I_{i+m}$,
 - $S_j \in S_{\min}$ 'i sağlayan $\forall j$ için $v_j \in I_j$ 'dir.

Ayrıca aşağıdaki listeyi kolay bir şekilde üretebiliriz:

- $\underline{I}=\{\underline{I}_i\}$ kümesinde, $\{v_{s_{\min}+1}, \dots, v_{2m-k}\}$ üzerinde (Lemma 6.4'teki k yerine $(2m-k-s_{\min})$; U kümesi yerine de $\{v_{s_{\min}+1}, \dots, v_{2m-k}\}$ gelmiş oldu) $2m$ tane farklı küme vardır öyle ki;
 - $\forall i \leq m$ için $\underline{I}_i \Delta \underline{I}_{i+m} = \underline{M}_i$,
 - $S_j \notin S_{\min}$ 'i sağlayan $\forall j$ için $v_j \in \underline{I}_j$ 'dir.

$\forall i \leq m$ için $\underline{I}_i = \emptyset$ ve $\underline{I}_{i+m} = \underline{M}_i$ olduğu takdirde bu şartlar sağlanır ve sonra $\underline{I}_i$ ve $\underline{I}_{i+m}$ kümelerindeki u_i ve u_{i+m} 'lerin varlığı belirtilen özellikleri sağlaması için değişir.

Böylece G 'nin kenarları, düğümlerin gelen düğümler kümeleri tarafından;

$$B^+(v_i) = I_i \cup \underline{I}_i \cup p_{i-1} \pmod{m} \quad [8] \quad (6.1)$$

şeklinde tanımlanır. Burada p_i , tüm $j \geq 2m-k+1$ için $v_j \in B(v_j)$ olacak şekilde $\{v_{2m-k+1}, \dots, v_{2m}\}$ üzerindeki altkümelerden tek olarak seçilir. Bu $P(\{v_{2m-k+1}, \dots, v_{2m}\})$ kuvvet kümesindeki $(i+1)$. kümeyi seçerek tamamlanabilir. Burada kuvvet kümesinin elemanları, $\forall j \leq k$ için $(m-j)$. küme v_{2m-j} 'yi kapsayacak şekilde sıralanmıştır.

Örnek 6.3 $U=\{u_1, u_2, u_3, u_4\}$ kümesini ve $S=\{S_1=\{u_1\}, S_2=\{u_2\}, S_3=\{u_3\}, S_4=\{u_4\}, S_5=\{u_1\}, S_6=\emptyset\}$ kümelerini düşünelim. Yapı 6.2'deki terimlerden, $k=2$ ve $m=4$ olduğu

ve S 'nin $|S|=2m-k=6$ 'yı sağladığı görülür. Ayrıca (U,S) için minimum küme kaplamanın $s_{\min}=4$ 'dür, çünkü u_1, u_2, u_3, u_4 elemanlarını kaplayan en küçük alt küme ailesi $S_{\min}=\{\{u_1\}, \{u_2\}, \{u_3\}, \{u_4\}\}$ 'dir.

Sonra $v_1, \dots, v_{2m}$ olarak $2m$ düğümlü, az önceki küme kaplama problemine karşılık gelen bir $G=(V,E)$ grafi üretilir. Yapımızdan M ve $\underline{M}$ topluluklarını üretelim:

$$M=[\{v_1\}, \{v_2\}, \{v_3\}, \{v_4\}]$$

$$\underline{M}=[\{v_5\}, \emptyset, \emptyset, \emptyset]$$

M 'deki i . küme, minimum küme kaplamadaki u_i elemanını kaplayan kümeleri gösterir, diğer taraftan da $\underline{M}$ 'deki i . küme, u_i 'yi kaplayan fakat $S_{\min}$ 'de olmayan kümeleri gösterir.

Şimdi de Lemma 6.4'ü kullanarak I ve $\underline{I}$ kümesini inşa edelim:

I 'yı oluştururken ilk 4 elemanı keyfî seçip, $I_i \Delta I_{i+m} = M_i$ olacak şekilde I_{i+m} 'leri oluştururuz. Aynı şekilde $\underline{I}$ 'yı oluştururken ilk 4 elemanı keyfî seçip, $\underline{I}_i \Delta \underline{I}_{i+m} = \underline{M}_i$ olacak şekilde $\underline{I}_{i+m}$ 'leri oluştururuz.

$$I = \{\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}, \{v_1, v_3, v_4\}, \{v_2\}, \{v_3\}, \{v_4\}, \{v_1, v_3\}\}$$

$$\underline{I} = [\emptyset, \{v_6\}, \{v_5, v_6\}, \{v_6\}, \{v_5\}, \{v_6\}, \{v_5, v_6\}, \{v_6\}]$$

Yapı ile uyumlu olacak şekilde I ve $\underline{I}$ 'nin tek olmadığına dikkat ediniz.

Son olarak kuvvet kümesine 6.1 numaralı denklemi uygulayarak;

$$\begin{aligned} P(\{v_7, v_8\}) &= \{p_0, \dots, p_3\} \\ &= \{\emptyset, \{v_8\}, \{v_7\}, \{v_7, v_8\}\} \end{aligned}$$

bulunan kuvvet kümesi I ve $\underline{I}$ 'nin elemanları ile yapıda belirtildiği gibi birleştirilerek grafin kenarları oluşturulmuş olur

$$B^+(v_1) = \{v_1, v_2\}$$

$$B^+(v_5) = \{v_2, v_5\}$$

$$B^+(v_2) = \{v_2, v_3, v_6, v_8\}$$

$$B^+(v_6) = \{v_3, v_6, v_7\}$$

$$B^+(v_3) = \{v_3, v_4, v_5, v_6, v_8\}$$

$$B^+(v_7) = \{v_4, v_5, v_6, v_8\}$$

$$B^+(v_4) = \{v_1, v_3, v_4, v_6, v_7, v_8\}$$

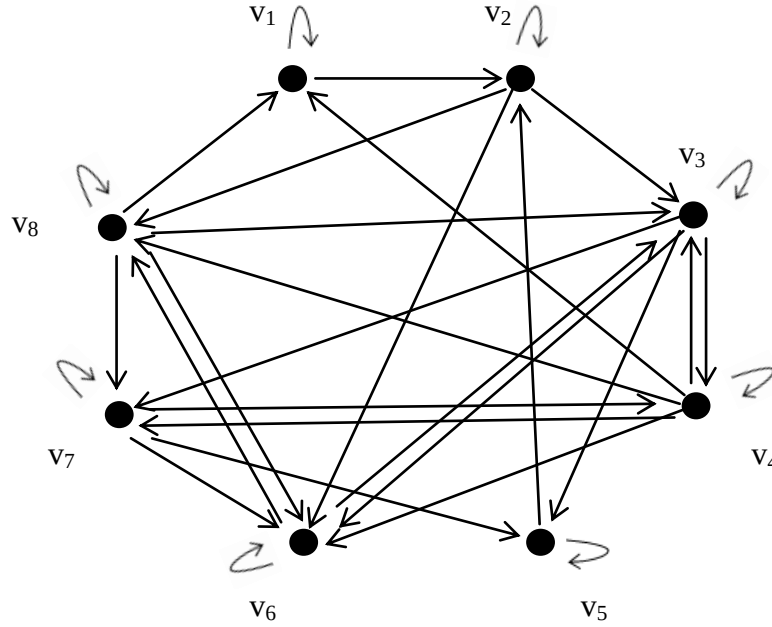
$$B^+(v_8) = \{v_1, v_3, v_6, v_7, v_8\}$$

Ürettiğimiz grafın komşuluk matrisi aşağıdaki gibidir.

$$M(U, S) = \begin{bmatrix} & \overbrace{v_1 \ v_2 \ v_3 \ v_4}^I & \overbrace{v_5 \ v_6}^I & \overbrace{v_7 \ v_8}^P \\ v_1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ v_2 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ v_3 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ v_4 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ v_5 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ v_6 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ v_7 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ v_8 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Bu grafın diyagramını çizerek daha somut bir şekilde görelim (Şekil 6.5).

Yapımızdaki (U,S) küme kaplama problemindeki minimum küme kaplama ile elde ettiğimiz grafın minimum belirleme kodu arasında ilişkiyi gösteren özelliği verelim.



Şekil 6.5 Yapı 6.2'nin örneğindeki (U,S) küme kaplama problemine karşılık gelen graf

Özellik 6.4 $|U|=2^k=m$, $|S|=2m-k$ ve $s_{\min} \geq k+2$ olmak üzere bir (U,S) küme kaplama problemi ile Yapı 6.2'nin ürettiği G grafının minimum belirleme kodları arasında doğrudan bir bağ vardır. $S_{\min}$ ile ilişkili olan $V_{\min}=\{v_1, \dots, v_{s_{\min}}\}$ düğümleri, $G(V,E)$ 'nin bir belirleme kodunu oluşturur. Yani (U,S) için küme kaplaması olan bir $S=\{S_1, \dots, S_t\}$ alt küme ailesine karşılık gelen $C=\{v_1, \dots, v_t\}$ düğüm kümesi G grafi için bir belirleme kodudur. Dahası, eğer bu $S=\{S_1, \dots, S_t\}$ kaplaması minimum ise buna karşılık gelen $C=\{v_1, \dots, v_t\}$ kodu da G için minimumdur[8].

İspat: Yapıyı oluştururken I 'daki kümelerin hepsinin birbirinden farklı olması şartı koyulmuştu. Buradan oluşacak bütün gelen düğüm kümelerinin birbirinden farklı olduğunu anlarız. Böylelikle $B^+(v_i) \cap V_{\min}$ 'lerin birbirinden farklı olduğu ortaya çıkar. Bu da $V_{\min}$ 'in belirleme kodu olduğunu gösterir[8].

Bu özelliği somut olarak görmek için Örnek 6.3'teki (U,S) 'in minimum küme kaplamasına veya kaplamalarına bakalım. $\{S_1, S_2, S_3, S_4\}$ ve $\{S_2, S_3, S_4, S_5\}$ küme aileleri (U,S) için birer minimum küme kaplamadır. Bunlara karşılık gelen $\{v_1, v_2, v_3, v_4\}$ ve $\{v_2, v_3, v_4, v_5\}$ kümeleri ürettiğimiz grafın minimum belirleme kodlarıdır.

Ayrıca (U, S) için küme kaplaması olmayan küme ailelerine karşılık gelen düğüm kümeleri belirleme kodu değildir. Örneğin $\{S_1, S_2, S_3, S_5\}$ küme ailesine karşılık gelen $A = \{v_1, v_2, v_3, v_5\}$ kümesi belirleme kodu değildir çünkü bu küme için $I_A(v_4) = I_A(v_7)$ 'dir, belirleme kodunun tanımından A 'nın belirleme kodu olmadığı anlaşılır.

BELİRLEME KODLARININ UYGULAMALARI

Grafların bilgi bilimlerindeki ve mühendisliklerdeki birçok alanda çok çeşitli uygulamaları vardır. Bir graf herhangi bir fiziksel durumu ve çeşitli nesnelerin birbirleri ile olan ilişkilerini göstermede kullanılabilir. Ayrıca graf modelleri, birçok pratik problemin çözümünde de kullanılabilir[22]. Bu manada, içerisinde birçok girdinin bulunduğu sistemlerde optimizasyon işlemi için öncelikle sistemin graf modeli çıkarılır daha sonra da graflarda optimize işlemini gerçekleştirecek yani en az elemanla sistemi belirlemeyi sağlayacak olan “belirleme kodları” kullanılır.

Bu bölümde belirleme kodlarının bazı uygulamaları ve çalışma prensipleri verilecek.

7.1 Çok işlemcili sistemlerde Hata tespiti

Bu uygulamada hata tespitinin amacı, sistemi denetleyip hatalı işlemciyi veya işlemcileri bulmaktır. Burada çok işlemcili sistemi bir $G=(V,E)$ yönsüz grafi ile temsil edilir. Bu temsil işleminde V işlemcilerin kümesi ve E de bu işlemciler arasındaki bağlantıları göstermektedir. Belli seçilmiş işlemcilerin üzerine özel yazılımlar uygulanarak hatalar tespit edilir. Bu işlemcilerin seçimi de, hatalı işlemcilerin tek bir şekilde belirlmesine izin veren C kodunun üretilmesi ile yapılır. Bu şekilde her bir kodsöz düğüme karşılık gelen işlemciler kendilerini ve komşu işlemcileri test eder. Buradaki düğümlere işlemcileri karşılık getirme işlemi 1 yarıçaplı düğümler kümesi kullanılarak gerçekleştirilmiştir. Böylelikle bir optimal kod sistemde hata belirleme için gerekli minimum işlemci sayısını elde etmeyi sağlar[2].

7.2 Konum Belirleme

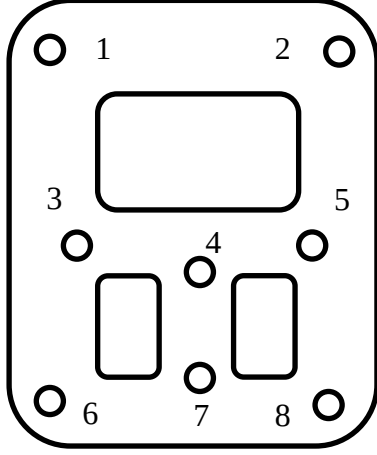
Sensor teknolojilerindeki son gelişmeler, acil durumlarda şebekelerin etkili kullanımı için duman detektörleri veya aydınlatma gibi var olan altyapılardaki küçük cihazların bağlantısını ve kurulumunu mümkün hâle getirdi. Bu ağlar acil kontrol merkezleri için 3 boyutlu görüntü alma veya önemli bölgeleri canlı izleme gibi hayati görevleri yerine getiriyor[3].

İleri seviye önemli görevlere sahip olan acil tepki sistemleri aşağıdaki özellikleri sağlaması gerekir:

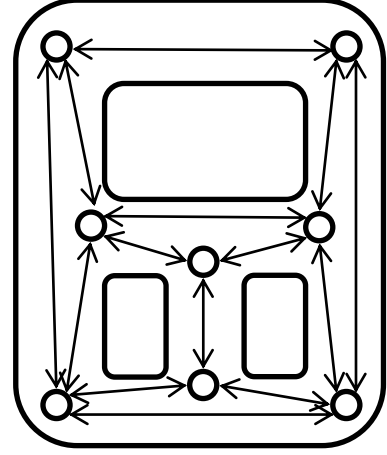
- Ekip elemanları kendilerinin ve diğerlerinin yerlerini belirleyebilmeli.
- Potansiyel tehlikeleri ve acil durum kaynaklarının yerlerini belirleyebilmeli.
- Tuzaktaki kişileri belirleyip kurtarmalı[3].

Var olan sistemlerin genel olarak en büyük eksikleri herhangi bir donanım eksikliğinde veya alanın değişikliğinde sistemin sağlamlığının kaybedilmesidir. Bu tür durumlara karşı belirleme kodlarına dayanan yeni sistemler kullanılarak sistemlerin güvenilirliği ve sağlamlığı arttırılmaya başlanmıştır. Bu yaklaşımda tahmin tabanlı sistemleri belli bir algoritmaya bağlayarak, sensorlar tüm alanı belirleyecek şekilde genelleştirmeye çalışılır. Buradaki ana fikir her bir bölünmüş bölgeyi gözetleyen sensorların kümelerinin birbirinden farklı olmasıdır[3]. Bu alandaki kullanımı bir örnekle daha iyi bir şekilde anlayabiliriz.

Şekil 7.1(a)'deki gibi bir alan planı tanımlayalım ve bunun üzerinde $P=\{1,2,3,4,5,6,7,8\}$ noktaları olsun. Bu noktaların birbiri ile olan bağlantılarını da Şekil 7.1(b)'deki gibi tanımlayalım.



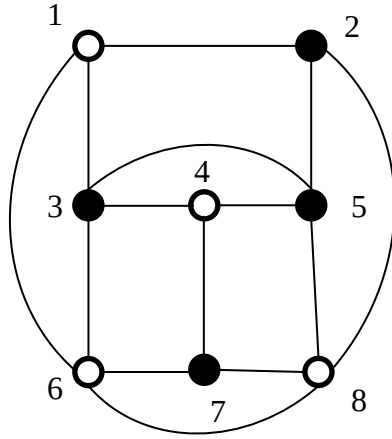
(a): Örnek alanımız ve üzerindeki noktalar.



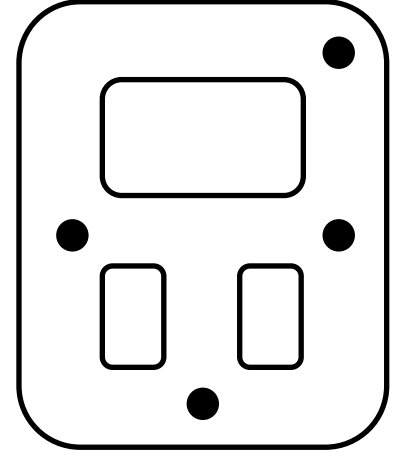
(b): Örnek alanımız üzerindeki noktaların birbirleri ile olan bağlantıları.

Şekil 7.1 Örnek konum belirleme alanımız.

Örneğimizdeki noktaları birer düğüm ve birbirleri ile olan bağlantıları kenar olarak düşünerek bir graf oluşturabiliriz. Bu graf üzerinde bulacağımız belirleme kodu ile örnek alanımızı kaplayabiliriz. Tabii ki amacımız en az noktayı kullanmak olduğu için bir optimal belirleme kodu bulacağız. İstedığımız gibi bir belirleme kodunu Şekil 7.2(a)'deki graf üzerindeki koyu renkli düğümler gösterir. Şekil 7.2(b)'de de bu belirleme kodunu alanımızın üzerinde görüyoruz.



(a): Örneğimizin graf modeli



(b): Belirleme kodumuzun alanımızın üzerine yerleştirilmiş hâli

Şekil 7.2 Belirleme kodlarının örneğimize uygulanması.

4 nokta ile diğer bütün noktaları belirleyebiliyoruz. Bunu açıklamamız gerekirse, bütün düğümlerin sinyal alabildikleri sinyal vericilerin kümelerini(belirleme kümeleri) yazalım.

$$1 \rightarrow \{2,3\}$$

$$2 \rightarrow \{2,5\}$$

$$3 \rightarrow \{3,5\}$$

$$4 \rightarrow \{3,5,7\}$$

$$5 \rightarrow \{2,3,5\}$$

$$6 \rightarrow \{3,7\}$$

$$7 \rightarrow \{7\}$$

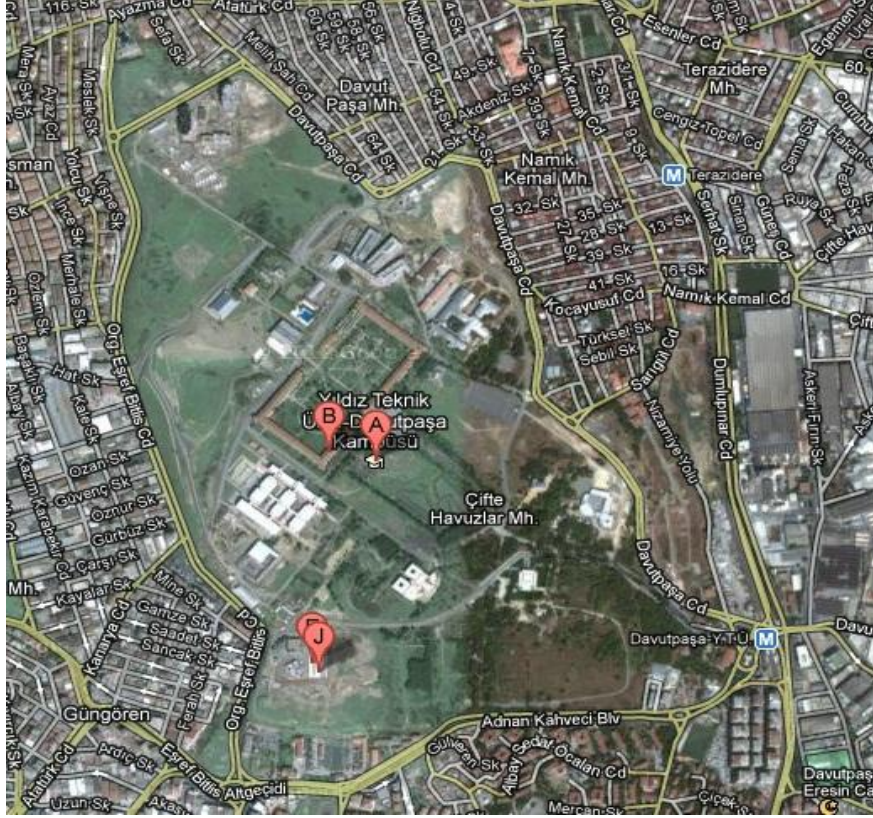
$$8 \rightarrow \{2,5,7\}$$

Görüldüğü gibi bütün kümeler farklı. Zaten belirleme kodunun tanımından belirleme kümelerinin birbirinden farklı olması gerekiyor. Böylelikle herhangi bir düğümdaki sinyal alıcı hangi düğümlerden sinyal aldığını belirttiğinde konumu belirlenmiş olur.

Şimdi belirleme kodlarının belirli bir bölgede konum belirleme ile alakalı uygulamasına gerçek hayattan bir örnek verelim.

7.2.1 Davutpaşa'da Konum Belirleme Uygulaması

Örnek bölgemiz Yıldız Teknik Üniversitesi-Davutpaşa Kampüsü olsun. Örnek alanımızı Şekil 7.3'ten takip edebilirsiniz.



Şekil 7.3 Belirleme kodlarını konum belirleme uygulamasında kullanacağımız bölge

Öncelikle alanımıza noktalar yerleştiriyoruz ve bu noktaların birbirleri ile irtibatlarına göre kenarlar belirleniyor ve graf modelimiz oluşmuş oluyor(Şekil 7.4). Bölgemizi temsil eden grafın gelen düğümler kümelerini yazalım(tüm noktalar kendileri ile irtibatlı olduklarından herhangi bir düğüm kendi gelen düğümler kümesinde olmak zorundadır).

$B^+(1)=\{1,2,4\}$	$B^+(2)=\{1,2,3\}$	$B^+(3)=\{2,3,6\}$	$B^+(4)=\{1,4,5,12\}$
$B^+(5)=\{4,5,6,8\}$	$B^+(6)=\{3,5,6,10\}$	$B^+(7)=\{7,10\}$	$B^+(8)=\{5,8,9,13\}$
$B^+(9)=\{8,9,10,14\}$	$B^+(10)=\{6,7,9,10,11\}$	$B^+(11)=\{10,11,14,15\}$	$B^+(12)=\{4,12,16\}$
$B^+(13)=\{8,13,14,16\}$	$B^+(14)=\{9,11,13,14,17\}$	$B^+(15)=\{11,15,18\}$	$B^+(16)=\{12,13,16,17\}$
$B^+(17)=\{14,16,17,18\}$	$B^+(18)=\{15,17,18,19\}$	$B^+(19)=\{18,19,20\}$	$B^+(20)=\{19,20\}$

Örneğimiz için belirleme kodunu Lemma 6.2'den yararlanarak bulmaya çalışalım. Bunun için farklılık kümelerini ve ayırım kümelerini bulmalıyız.

$$D(1,2)=\{3,4\}$$

$$D(1,3)=\{1,3,4,6\}$$

$$D(1,4)=\{2,5,12\}$$

$$D(2,3)=\{1,6\}$$

$$D(2,4)=\{2,3,4,5,12\}$$

$$D(3,4)=\{2,3,4,5,6,12\}$$

$D(1,5)=\{1,2,5,6,8\}$	$D(2,5)=\{1,2,3,4,5,6,8\}$	$D(3,5)=\{2,3,4,5,8\}$
$D(1,6)=\{1,2,3,4,5,6,10\}$	$D(2,6)=\{1,2,5,6,10\}$	$D(3,6)=\{2,5,10\}$
$D(1,7)=\{1,2,4,7,10\}$	$D(2,7)=\{1,2,3,7,10\}$	$D(3,7)=\{2,3,6,7,10\}$
$D(1,8)=\{1,2,4,5,8,9,13\}$	$D(2,8)=\{1,2,3,5,8,9,13\}$	$D(3,8)=\{2,3,5,6,8,9,13\}$
$D(1,9)=\{1,2,4,8,9,13\}$	$D(2,9)=\{1,2,3,8,9,10,14\}$	$D(3,9)=\{2,3,6,8,9,10,14\}$
$D(1,10)=\{1,2,4,6,7,9,10,11\}$	$D(2,10)=\{1,2,3,6,7,9,10,11\}$	$D(3,10)=\{2,3,7,9,10,11\}$
$D(1,11)=\{1,2,4,10,11,14,15\}$	$D(2,11)=\{1,2,3,10,11,14,15\}$	$D(3,11)=\{2,3,6,10,11,14,15\}$
$D(1,12)=\{1,2,12,16\}$	$D(2,12)=\{1,2,3,4,12,16\}$	$D(3,12)=\{2,3,4,6,12,16\}$
$D(1,13)=\{1,2,4,8,13,14,16\}$	$D(2,13)=\{1,2,3,8,13,14,16\}$	$D(3,13)=\{2,3,6,8,13,14,16\}$
$D(1,14)=\{1,2,4,9,11,13,14,17\}$	$D(2,14)=\{1,2,3,9,11,13,14,17\}$	$D(3,14)=\{2,3,6,9,11,13,14,17\}$
$D(1,15)=\{1,2,4,11,15,18\}$	$D(2,15)=\{1,2,3,11,15,18\}$	$D(3,15)=\{2,3,6,11,15,18\}$
$D(1,16)=\{1,2,4,12,13,16,17\}$	$D(2,16)=\{1,2,3,12,13,16,17\}$	$D(3,16)=\{2,3,6,12,13,16,17\}$
$D(1,17)=\{1,2,4,14,16,17,18\}$	$D(2,17)=\{1,2,3,14,16,17,18\}$	$D(3,17)=\{2,3,6,14,16,17,18\}$
$D(1,18)=\{1,2,4,15,17,18,19\}$	$D(2,18)=\{1,2,3,15,17,18,19\}$	$D(3,18)=\{2,3,6,15,17,18,19\}$
$D(1,19)=\{1,2,4,18,19,20\}$	$D(2,19)=\{1,2,3,18,19,20\}$	$D(3,19)=\{2,3,6,18,19,20\}$
$D(1,20)=\{1,2,4,19,20\}$	$D(2,20)=\{1,2,3,19,20\}$	$D(3,20)=\{2,3,6,19,20\}$
$D(4,5)=\{1,6,8,12\}$		
$D(4,6)=\{1,3,4,6,10,12\}$	$D(5,6)=\{3,4,8,10\}$	
$D(4,7)=\{1,4,5,7,10,12\}$	$D(5,7)=\{4,5,6,7,8,10\}$	$D(6,7)=\{3,5,6,7\}$
$D(4,8)=\{1,4,8,9,12,13\}$	$D(5,8)=\{4,6,9,13\}$	$D(6,8)=\{3,6,8,9,10,13\}$
$D(4,9)=\{1,4,5,8,9,10,12,14\}$	$D(5,9)=\{4,5,6,9,10,14\}$	$D(6,9)=\{3,5,6,8,9,14\}$
$D(4,10)=\{1,4,5,6,7,9,10,11,12\}$	$D(5,10)=\{4,5,7,8,9,10,11\}$	$D(6,10)=\{3,5,7,9,11\}$
$D(4,11)=\{1,4,5,10,11,12,14,15\}$	$D(5,11)=\{4,5,6,8,10,11,14,15\}$	$D(6,11)=\{3,5,6,11,14,15\}$
$D(4,12)=\{1,5,16\}$	$D(5,12)=\{5,6,8,12,16\}$	$D(6,12)=\{3,4,5,6,10,12,16\}$
$D(4,13)=\{1,4,5,8,12,13,14,16\}$	$D(5,13)=\{4,5,6,13,14,16\}$	$D(6,13)=\{3,5,6,8,10,13,14,16\}$
$D(4,14)=\{1,4,5,9,11,12,13,14,17\}$	$D(5,14)=\{4,5,6,8,9,11,13,14,17\}$	$D(6,14)=\{3,5,6,9,10,11,13,14,17\}$
$D(4,15)=\{1,4,5,11,12,15,18\}$	$D(5,15)=\{4,5,6,8,11,15,18\}$	$D(6,15)=\{3,5,6,10,11,15,18\}$
$D(4,16)=\{1,4,5,13,16,17\}$	$D(5,16)=\{4,5,6,8,12,13,16,17\}$	$D(6,16)=\{3,5,6,10,12,13,16,17\}$
$D(4,17)=\{1,4,5,12,14,16,17,18\}$	$D(5,17)=\{4,5,6,8,14,16,17,18\}$	$D(6,17)=\{3,5,6,10,14,16,17,18\}$

$D(4,18)=\{1,4,5,12,15,17,18,19\}$	$D(5,18)=\{4,5,6,8,15,17,18,19\}$	$D(6,18)=\{3,5,6,10,15,17,18,19\}$
$D(4,19)=\{1,4,5,12,18,19,20\}$	$D(5,19)=\{4,5,6,8,18,19,20\}$	$D(6,19)=\{3,5,6,10,18,19,20\}$
$D(4,20)=\{1,4,5,12,19,20\}$	$D(5,20)=\{4,5,6,8,19,20\}$	$D(6,20)=\{3,5,6,10,19,20\}$
$D(7,8)=\{5,7,8,9,10,13\}$		
$D(7,9)=\{7,8,9,14\}$	$D(8,9)=\{5,10,13,14\}$	
$D(7,10)=\{6,9,11\}$	$D(8,10)=\{5,6,7,8,10,11,13\}$	$D(9,10)=\{6,7,8,11,14\}$
$D(7,11)=\{7,11,14,15\}$	$D(8,11)=\{5,8,9,10,11,13,14,15\}$	$D(9,11)=\{8,9,11,15\}$
$D(7,12)=\{4,7,10,12,16\}$	$D(8,12)=\{4,5,8,9,12,13,16\}$	$D(9,12)=\{4,8,9,10,12,14,16\}$
$D(7,13)=\{7,8,10,13,14,16\}$	$D(8,13)=\{5,9,14,16\}$	$D(9,13)=\{9,10,13,16\}$
$D(7,14)=\{7,9,10,11,13,14,17\}$	$D(8,14)=\{5,8,11,14,17\}$	$D(9,14)=\{8,10,11,13,17\}$
$D(7,15)=\{7,10,11,15,18\}$	$D(8,15)=\{5,8,9,11,13,15,18\}$	$D(9,15)=\{8,9,10,11,14,15,18\}$
$D(7,16)=\{7,10,12,13,16,17\}$	$D(8,16)=\{5,8,9,12,16,17\}$	$D(9,16)=\{8,9,10,12,13,14,16,17\}$
$D(7,17)=\{7,10,14,16,17,18\}$	$D(8,17)=\{5,8,9,13,14,16,17,18\}$	$D(9,17)=\{8,9,10,16,17,18\}$
$D(7,18)=\{7,10,15,17,18,19\}$	$D(8,18)=\{5,8,9,13,15,17,18,19\}$	$D(9,18)=\{8,9,10,14,15,17,18,19\}$
$D(7,19)=\{7,10,18,19,20\}$	$D(8,19)=\{5,8,9,13,18,19,20\}$	$D(9,19)=\{8,9,10,14,18,19,20\}$
$D(7,20)=\{7,10,19,20\}$	$D(8,20)=\{5,8,9,13,19,20\}$	$D(9,20)=\{8,9,10,14,19,20\}$
$D(10,11)=\{6,7,9,14,15\}$		
$D(10,12)=\{4,6,7,9,10,11,12,16\}$	$D(11,12)=\{4,10,11,12,14,15,16\}$	
$D(10,13)=\{6,7,8,9,10,11,13,14,16\}$	$D(11,13)=\{8,10,11,13,15,16\}$	$D(12,13)=\{4,8,12,13,14\}$
$D(10,14)=\{6,7,10,13,14,17\}$	$D(11,14)=\{9,10,13,15,17\}$	$D(12,14)=\{4,9,11,12,13,14,16,17\}$
$D(10,15)=\{6,7,9,10,15,18\}$	$D(11,15)=\{10,14,18\}$	$D(12,15)=\{4,11,12,15,16,18\}$
$D(10,16)=\{6,7,9,10,11,12,13,16,17\}$	$D(11,16)=\{10,11,12,13,14,15,16,17\}$	$D(12,16)=\{4,13,17\}$
$D(10,17)=\{6,7,9,10,11,14,16,17,18\}$	$D(11,17)=\{10,11,15,16,17,18\}$	$D(12,17)=\{4,12,14,17,18\}$
$D(10,18)=\{6,7,9,10,11,15,17,18,19\}$	$D(11,18)=\{10,11,14,17,18,19\}$	$D(12,18)=\{4,12,15,16,17,18,19\}$
$D(10,19)=\{6,7,9,10,11,18,19,20\}$	$D(11,19)=\{10,11,14,15,18,19,20\}$	$D(12,19)=\{4,12,16,18,19,20\}$
$D(10,20)=\{6,7,9,10,11,19,20\}$	$D(11,20)=\{10,11,14,15,19,20\}$	$D(12,20)=\{4,12,16,19,20\}$

$$D(13,14)=\{8,9,11,16,17\}$$

$$D(13,15)=\{8,11,13,14,15,16,18\} \quad D(14,15)=\{9,13,14,15,17,18\}$$

$$D(13,16)=\{8,12,14,17\} \quad D(14,16)=\{9,11,12,14,16\} \quad D(15,16)=\{11,12,13,15,16,17,18\}$$

$$D(13,17)=\{8,13,17,18\} \quad D(14,17)=\{9,11,13,16,18\} \quad D(15,17)=\{11,14,15,16,17,18\}$$

$$D(13,18)=\{8,13,14,15,16,17,18,19\} \quad D(14,18)=\{9,11,13,14,15,18,19\} \quad D(15,18)=\{11,17,19\}$$

$$D(13,19)=\{8,13,14,16,18,19,20\} \quad D(14,19)=\{9,11,13,14,17,18,19,20\} \quad D(15,19)=\{11,15,19,20\}$$

$$D(13,20)=\{8,13,14,16,18,19,20\} \quad D(14,20)=\{9,11,13,14,17,19,20\} \quad D(15,20)=\{11,15,18,19,20\}$$

$$D(16,17)=\{12,13,14,18\}$$

$$D(16,18)=\{12,13,15,16,18,19\} \quad D(17,18)=\{14,15,16,19\}$$

$$D(16,19)=\{12,13,16,17,18,19,20\} \quad D(17,19)=\{14,16,17,19,20\} \quad D(18,19)=\{15,17,20\}$$

$$D(16,20)=\{12,13,16,17,19,20\} \quad D(17,20)=\{14,16,17,18,19,20\} \quad D(18,20)=\{15,17,18,20\}$$

$$D(19,20)=\{18\}$$

$$\delta_1 = \{(1,3), (1,5), (1,6), (1,7), (1,8), (1,9), (1,10), (1,11), (1,12), (1,13), (1,14), (1,15), (1,16), \\ (1,17), (1,18), (1,19), (1,20), (2,3), (2,5), (2,6), (2,7), (2,8), (2,9), (2,10), (2,11), (2,12), (2,13), \\ (2,14), (2,15), (2,16), (2,17), (2,18), (2,19), (2,20), (4,5), (4,6), (4,7), (4,8), (4,9), (4,10), (4,11), \\ (4,12), (4,13), (4,14), (4,15), (4,16), (4,17), (4,18), (4,19), (4,20)\}$$

$$\delta_2 = \{(1,4), (1,5), (1,6), (1,7), (1,8), (1,9), (1,10), (1,11), (1,12), (1,13), (1,14), (1,15), (1,16), \\ (1,17), (1,18), (1,19), (1,20), (2,4), (2,5), (2,6), (2,7), (2,8), (2,9), (2,10), (2,11), (2,12), (2,13), \\ (2,14), (2,15), (2,16), (2,17), (2,18), (2,19), (2,20), (3,4), (3,5), (3,6), (3,7), (3,8), (3,9), (3,10), \\ (3,11), (3,12), (3,13), (3,14), (3,15), (3,16), (3,17), (3,18), (3,19), (3,20)\}$$

$$\delta_3 = \{(1,2), (1,3), (1,6), (2,4), (2,5), (2,7), (2,8), (2,9), (2,10), (2,11), (2,12), (2,13), (2,14), (2,15), (2,16), \\ (2,17), (2,18), (2,19), (2,20), (3,4), (3,5), (3,7), (3,8), (3,9), (3,10), (3,11), (3,12), (3,13), (3,14), \\ (3,15), (3,16), (3,17), (3,18), (3,19), (3,20), (4,6), (5,6), (6,7), (6,8), (6,9), (6,10), (6,11), (6,12), \\ (6,13), (6,14), (6,15), (6,16), (6,17), (6,18), (6,19), (6,20)\}$$

$$\delta_4 = \{(1,2), (1,3), (1,6), (1,7), (1,8), (1,9), (1,10), (1,11), (1,13), (1,14), (1,15), (1,16), (1,17), (1,18), \\ (1,19), (1,20), (2,4), (2,5), (2,12), (3,4), (3,5), (3,12), (4,6), (4,7), (4,8), (4,9), (4,10), (4,11), (4,13)\}$$

,(4,14),(4,15),(4,16),(4,17),(4,18),(4,19),(4,20),(5,6),(5,7),(5,8),(5,9),(5,10),(5,11),(5,13),(5,14),(5,15),(5,16),(5,17),(5,18),(5,19),(5,20),(6,12),(7,12),(8,12),(9,12),(10,12),(11,12),(12,13),(12,14),(12,15),(12,16),(12,17),(12,18),(12,19),(12,20)}

δ_5
 $=\{(1,4),(1,5),(1,6),(1,8),(2,4),(2,5),(2,6),(2,8),(3,4),(3,5),(3,6),(3,8),(4,7),(4,9),(4,10),(4,11),(4,12),(4,13),(4,14),(4,15),(4,16),(4,17),(4,18),(4,19),(4,20),(5,7),(5,9),(5,10),(5,11),(5,12),(5,13),(5,14),(5,15),(5,16),(5,17),(5,18),(5,19),(5,20),(6,7),(6,9),(6,10),(6,11),(6,12),(6,13),(6,14),(6,15),(6,16),(6,17),(6,18),(6,19),(6,20),(7,8),(7,9),(8,10),(8,11),(8,12),(8,13),(8,14),(8,15),(8,16),(8,17),(8,18),(8,19),(8,20)\}$

δ_6
 $=\{(1,3),(1,5),(1,6),(1,10),(2,3),(2,5),(2,6),(2,10),(3,4),(3,7),(3,8),(3,9),(3,11),(3,12),(3,13),(3,14),(3,15),(3,16),(3,17),(3,18),(3,19),(3,20),(4,5),(4,6),(4,10),(5,7),(5,8),(5,9),(5,11),(5,12),(5,13),(5,14),(5,15),(5,16),(5,17),(5,18),(5,19),(5,20),(6,7),(6,8),(6,9),(6,11),(6,12),(6,13),(6,14),(6,15),(6,16),(6,17),(6,18),(6,19),(6,20),(7,10),(8,10),(9,10),(10,11),(10,12),(10,13),(10,14),(10,15),(10,16),(10,17),(10,18),(10,19),(10,20)\}$

δ_7
 $=\{(1,7),(1,10),(2,7),(2,10),(3,7),(3,10),(4,7),(4,10),(5,7),(5,10),(6,7),(6,10),(7,8),(7,9),(7,11),(7,12),(7,13),(7,14),(7,15),(7,16),(7,17),(7,18),(7,19),(7,20),(8,10),(9,10),(10,11),(10,12),(10,13),(10,14),(10,15),(10,16),(10,17),(10,18),(10,19),(10,20)\}$

δ_8
 $=\{(1,5),(1,8),(1,9),(1,13),(2,5),(2,8),(2,9),(2,13),(3,5),(3,8),(3,9),(3,13),(4,5),(4,8),(4,9),(4,13),(5,6),(5,7),(5,10),(5,11),(5,12),(5,14),(5,15),(5,16),(5,17),(5,18),(5,19),(5,20),(6,8),(6,9),(6,13),(7,8),(7,9),(7,13),(8,10),(8,11),(8,12),(8,14),(8,15),(8,16),(8,17),(8,18),(8,19),(8,20),(9,10),(9,11),(9,12),(9,14),(9,15),(9,16),(9,17),(9,18),(9,19),(9,20),(10,13),(11,13),(12,13),(13,14),(13,15),(13,16),(13,17),(13,18),(13,19),(13,20)\}$

δ_9
 $=\{(1,8),(1,5),(1,10),(1,14),(2,8),(2,5),(2,10),(2,14),(3,8),(3,9),(3,10),(3,14),(4,8),(4,9),(4,10),(4,14),(5,8),(5,9),(5,10),(5,14),(6,8),(6,9),(6,10),(6,14),(7,8),(7,9),(7,10),(7,14),(8,11),(8,12),(8,13),(8,15),(8,16),(8,17),(8,18),(8,19),(8,20),(9,11),(9,12),(9,13),(9,15),(9,16),(9,17),(9,18),(9,19),(9,20),(10,11),(10,12),(10,13),(10,15),(10,16),(10,17),(10,18),(10,19),(10,20),(11,14),(12,14),(13,14),(14,15),(14,16),(14,17),(14,18),(14,19),(14,20)\}$

δ_{10}
 $=\{(1,6),(1,7),(1,10),(1,11),(2,6),(2,7),(2,9),(2,10),(2,11),(3,6),(3,7),(3,9),(3,10),(3,11),(4,$

,6),(4,7),(4,9),(4,10),(4,11),(5,6),(5,7),(5,9),(5,10),(5,11),(6,8),(6,12),(6,13),(6,14),(6,15),
(6,16),(6,17),(6,18),(6,19),(6,20),(7,8),(7,12),(7,13),(7,14),(7,15),(7,16),(7,17),(7,18),(7,
,19),(7,20),(8,9),(8,10),(8,11),(9,12),(9,13),(9,14),(9,15),(9,16),(9,17),(9,18),(9,19),(9,20
) ,(10,12),(10,13),(10,14),(10,15),(10,16),(10,17),(10,18),(10,19),(10,20),(11,12),(11,13),
(11,14),(11,15),(11,16),(11,17),(11,18),(11,19),(11,20)}

δ_{11}
 $=\{(1,10),(1,11),(1,14),(1,15),(2,10),(2,11),(2,14),(2,15),(3,10),(3,11),(3,14),(3,15),(4,10$
 $),(4,11),(4,14),(4,15),(5,10),(5,11),(5,14),(5,15),(6,10),(6,11),(6,14),(6,15),(7,10),(7,11),$
 $(7,14),(7,15),(8,10),(8,11),(8,14),(8,15),(9,10),(9,11),(9,14),(9,15),(10,12),(10,13),(10,1$
 $6),(10,17),(10,18),(10,19),(10,20),(11,12),(11,13),(11,16),(11,17),(11,18),(11,19),(11,20$
 $),(12,14),(12,15),(13,14),(13,15),(14,16),(14,17),(14,18),(14,19),(14,20),(15,16),(15,17),$
 $(15,18),(15,19),(15,20)\}$

δ_{12}
 $=\{(1,4),(1,12),(1,16),(2,4),(2,12),(2,16),(3,4),(3,12),(3,16),(4,5),(4,6),(4,7),(4,8),(4,9),(4$
 $,10),(4,11),(4,13),(4,14),(4,15),(4,17),(4,18),(4,19),(4,20),(5,12),(5,16),(6,12),(6,16),(7,1$
 $2),(7,16),(8,12),(8,16),(9,12),(9,16),(10,12),(10,16),(11,12),(11,16),(12,13),(12,14),(12,$
 $15),(12,17),(12,18),(12,19),(12,20),(13,16),(14,16),(15,16),(16,17),(16,18),(16,19),(16,2$
 $0)\}$

δ_{13}
 $=\{(1,8),(1,13),(1,14),(1,16),(2,8),(2,13),(2,14),(2,16),(3,8),(3,13),(3,14),(3,16),(4,8),(4,1$
 $3),(4,14),(4,16),(5,8),(5,13),(5,14),(5,16),(6,8),(6,13),(6,14),(6,16),(7,8),(7,13),(7,14),(7,$
 $16),(8,9),(8,10),(8,11),(8,12),(8,15),(8,17),(8,18),(8,19),(8,20),(9,13),(9,14),(9,16),(10,1$
 $3),(10,14),(10,16),(11,13),(11,14),(11,16),(12,13),(12,14),(12,16),(13,15),(13,17),(13,18$
 $),(13,19),(13,20),(14,15),(14,17),(14,18),(14,19),(14,20),(15,16),(16,17),(16,18),(16,19),$
 $(16,20)\}$

δ_{14}
 $=\{(1,9),(1,11),(1,13),(1,14),(1,17),(2,9),(2,11),(2,13),(2,14),(2,17),(3,9),(3,11),(3,13),(3,$
 $14),(3,17),(4,9),(4,11),(4,13),(4,14),(4,17),(5,9),(5,11),(5,13),(5,14),(5,17),(6,9),(6,11),($
 $6,13),(6,14),(6,17),(7,9),(7,11),(7,13),(7,14),(7,17),(8,9),(8,11),(8,13),(8,14),(8,17),(9,10$
 $),(9,12),(9,15),(9,16),(9,18),(9,19),(9,20),(10,11),(10,13),(10,14),(10,17),(11,12),(11,15)$
 $),(11,16),(11,18),(11,19),(11,20),(12,13),(12,14),(12,17),(13,15),(13,16),(13,18),(13,19),($
 $13,20),(14,15),(14,16),(14,18),(14,19),(14,20),(15,17),(16,17),(17,18),(17,19),(17,20)\}$

$$\delta_{15} = \{(1,11),(1,15),(1,18),(2,11),(2,15),(2,18),(3,11),(3,15),(3,18),(4,11),(4,15),(4,18),(5,11),(5,15),(5,18),(6,11),(6,15),(6,18),(7,11),(7,15),(7,18),(8,11),(8,15),(8,18),(9,11),(9,15),(9,18),(10,11),(10,15),(10,18),(11,12),(11,13),(11,14),(11,16),(11,17),(11,19),(11,20),(12,15),(12,18),(13,15),(13,18),(14,15),(14,18),(15,16),(15,17),(15,19),(15,20),(16,18),(17,18),(18,19),(18,20)\}$$

$$\delta_{16} = \{(1,12),(1,13),(1,16),(1,17),(2,12),(2,13),(2,16),(2,17),(3,12),(3,13),(3,16),(3,17),(4,12),(4,13),(4,16),(4,17),(5,12),(5,13),(5,16),(5,17),(6,12),(6,13),(6,16),(6,17),(7,12),(7,13),(7,16),(7,17),(8,12),(8,13),(8,16),(8,17),(9,12),(9,13),(9,16),(9,17),(10,12),(10,13),(10,16),(10,17),(11,12),(11,13),(11,16),(11,17),(12,14),(12,15),(12,18),(12,19),(12,20),(13,14),(13,15),(13,18),(13,19),(13,20),(14,16),(14,17),(15,16),(15,17),(16,18),(16,19),(16,20),(17,18),(17,19),(17,20)\}$$

$$\delta_{17} = \{(1,14),(1,16),(1,17),(1,18),(2,14),(2,16),(2,17),(2,18),(3,14),(3,16),(3,17),(3,18),(4,14),(4,16),(4,17),(4,18),(5,14),(5,16),(5,17),(5,18),(6,14),(6,16),(6,17),(6,18),(7,14),(7,16),(7,17),(7,18),(8,14),(8,16),(8,17),(8,18),(9,14),(9,16),(9,17),(9,18),(10,14),(10,16),(10,17),(10,18),(11,14),(11,16),(11,17),(11,18),(12,14),(12,16),(12,17),(12,18),(13,14),(13,16),(13,17),(13,18),(14,15),(14,19),(14,20),(15,16),(15,17),(15,18),(16,19),(16,20),(17,19),(17,20),(18,19),(18,20)\}$$

$$\delta_{18} = \{(1,15),(1,17),(1,18),(1,19),(2,15),(2,17),(2,18),(2,19),(3,15),(3,17),(3,18),(3,19),(4,15),(4,17),(4,18),(4,19),(5,15),(5,17),(5,18),(5,19),(6,15),(6,17),(6,18),(6,19),(7,15),(7,17),(7,18),(7,19),(8,15),(8,17),(8,18),(8,19),(9,15),(9,17),(9,18),(9,19),(10,15),(10,17),(10,18),(10,19),(11,15),(11,17),(11,18),(11,19),(12,15),(12,17),(12,18),(12,19),(13,15),(13,17),(13,18),(13,19),(14,15),(14,17),(14,18),(14,19),(15,16),(15,20),(16,17),(16,18),(16,19),(17,20),(18,20),(19,20)\}$$

$$\delta_{19} = \{(1,18),(1,19),(1,20),(2,18),(2,19),(2,20),(3,18),(3,19),(3,20),(4,18),(4,19),(4,20),(5,18),(5,19),(5,20),(6,18),(6,19),(6,20),(7,18),(7,19),(7,20),(8,18),(8,19),(8,20),(9,18),(9,19),(9,20),(10,18),(10,19),(10,20),(11,18),(11,19),(11,20),(12,18),(12,19),(12,20),(13,18),(13,19),(13,20),(14,18),(14,19),(14,20),(15,18),(15,19),(15,20),(16,18),(16,19),(16,20),(17,18),(17,19),(17,20)\}$$

$$\delta_{20} = \{(1,19), (1,20), (2,19), (2,20), (3,19), (3,20), (4,19), (4,20), (5,19), (5,20), (6,19), (6,20), (7,19), (7,20), (8,19), (8,20), (9,19), (9,20), (10,19), (10,20), (11,19), (11,20), (12,19), (12,20), (13,19), (13,20), (14,19), (14,20), (15,19), (15,20), (16,19), (16,20), (17,19), (17,20), (18,19), (18,20)\}$$

Lemma 6.2’de söylenildiği gibi düğümlerdenayrım kümeleri $U = \{(u,z) \in V^2 | u \neq z\}$ kümesini kaplayanlar belirleme kodudur. O halde $(U, \partial = \{\delta_v | v \in V\})$ için optimal bir kaplama bulalım.

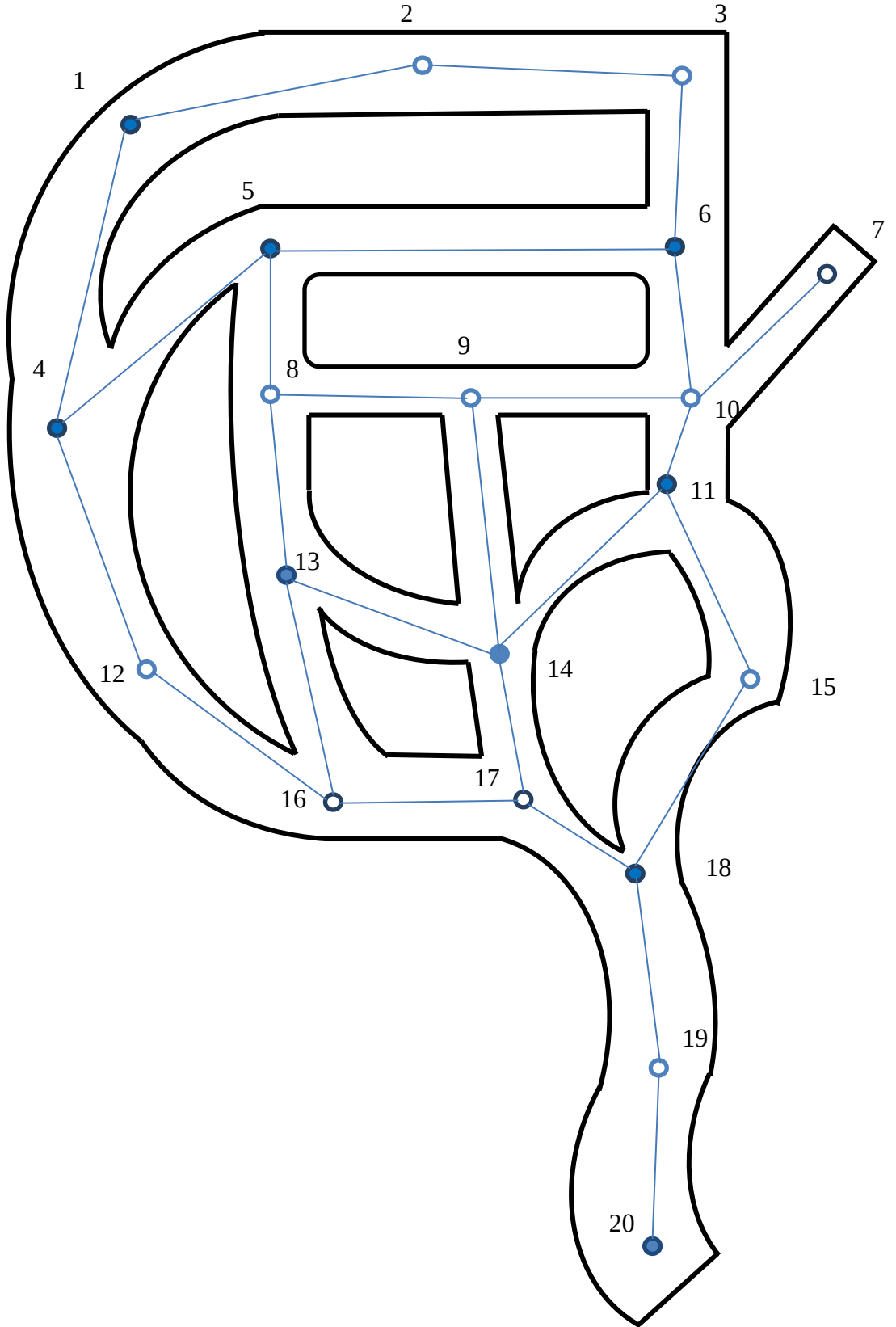
Kaplamamızda 18 düğümünün olması gerektiği açıktır. Çünkü (19,20) ikilisini kaplayan sadece 18 düğümü vardır. Bundan sonraki kaplama işlemi basit ancak oldukça uzun olduğundan burada sadece sonuç verilecek. $\{\delta_1, \delta_4, \delta_5, \delta_6, \delta_{11}, \delta_{13}, \delta_{14}, \delta_{18}, \delta_{20}\}$ alt küme ailesi U için bir kaplamadır. Buradan $C = \{1,4,5,6,11,13,14,18,20\}$ düğüm kümesi de örnek alanımızın graf modeli için bir belirleme kodudur. Gerçekten de,

$$\begin{aligned} I_C(1) &= \{1,4\}, I_C(2) = \{1\}, I_C(3) = \{6\}, I_C(4) = \{1,4,5\}, I_C(5) = \{4,5,6\}, I_C(6) = \{5,6\}, I_C(7) = \{\}, \\ I_C(8) &= \{5,13\}, I_C(9) = \{14\}, I_C(10) = \{6,11\}, I_C(11) = \{11,14\}, I_C(12) = \{4\}, I_C(13) = \{13,14\}, \\ I_C(14) &= \{11,13,14\}, I_C(15) = \{11,18\}, I_C(16) = \{13\}, I_C(17) = \{14,18\}, I_C(18) = \{18\}, \\ I_C(19) &= \{18,20\}, I_C(20) = \{20\} \end{aligned}$$

belirleme kümelerinin hepsi birbirinden farklı.

Şimdi elde ettiğimiz belirleme kodunu konum belirlemede nasıl kullanacağımıza geçelim. Bu kodun düğümlerine sinyal vericiler yerleştirdiğimizi düşünelim. Örneğin 5 düğümündeki sinyal alıcı $\{4,5,6\}$ sinyal vericilerinden sinyal alır. Tüm kümelerin sinyal aldığı düğüm kümesi farklı olacağından (çünkü alınan sinyal kümeleri belirleme kümesine karşılık gelir ve belirleme kümelerinin hepsi birbirinden farklıdır) bu kümenin hangi düğümün belirleme kümesi olduğunu, elimizdeki belirleme kümelerinin listesine bakıp buluruz ve sinyal alıcının nerede olduğunu saptarız.

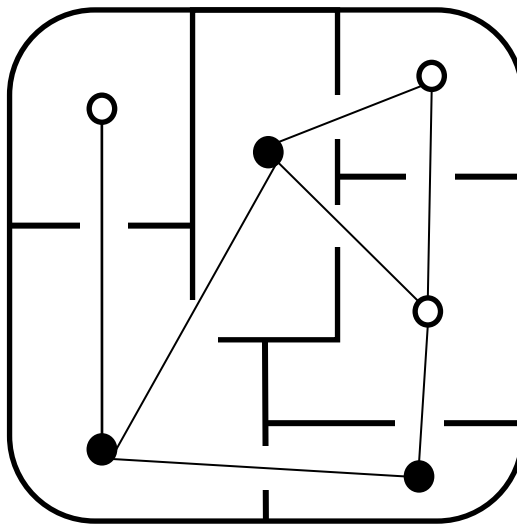
Aksine, her düğümün bulunduğu yerlere sinyal vericiler koyup aynı iş yapılırsa 20 adet sinyal verici kullanılması gerekir. Ancak aynı işi belirleme kodlarını kullanarak 9 adet sinyal verici ile yapmış oluruz ve böylece büyük bir tasarruf elde ederiz.



Şekil 7.4 Örnek alanımızın basit bir haritası ve graf modeli. İçi dolu olanlar bu graf için bulunan belirleme kodunun kodsözleridir.

7.3 Çevresel denetleme

Belirleme kodları kirlilik kaynaklarını tanımlama ve yerlerini belirleme gibi gözleme problemlerinde geniş bir uygulama alanına sahiptir. Bu tür uygulamalarda gözlenecek alan sonlu sayıda bölgelere ayrılır ve bir graf olarak modellenir. Burada her bir düğüm Şekil 7.5’deki gibi farklı birer bölgeyi temsil eder. Bu modelde iki düğüm, temsil ettikleri bölgeler birbirleri ile bağlantılı ise, kenarlarla birbirlerine bağlıdır. Bu graf için bir belirleme kodu gözlem araçlarının bir alt kümesidir. Bu alt kümenin özelliği her bir elemanın iletişiminde olduğu gözlem araçlarının kümesi(bu kümeler belirleme kümesidir) birbirinden farklı olmasıdır. Böylelikle belirleme kodları sayesinde her bir bölgeye gözlem aracı koymaya gerek kalmaksızın tüm bölgeler izlenebiliyor. Böylece hem gerekli araç-gereç sayısı hem de enerji sarfıyatı azaltılmış oluyor. Ayrıca belirleme kodlarına dayalı sistemler yakın alanı gözleyen sistemlerden daha güvenli ve yüksek çözünürlük sağlıyor[4]. Bu uygulamayı daha iyi anlamak için Şekil 7.5’e bakmanız faydalı olacaktır. Bu örnekte bölgeler birbirlerinden duvarlarla ayrılmıştır. Sadece bölgelere yerleştirilen gözlem araçlarının birbirleri ile bağlantı kurabilmesi için küçük açıklıklar vardır. Bölgelerin içindeki düğümler gözlem araçlarını, düğümlerin arasındaki kenarlar da bu gözlem araçlarının birbirleri ile olan bağlantılarını gösteriyor. Buradaki graf için içi dolu düğümler bir belirleme kodudur. Yani bu düğümlerin temsil ettiği bölgelere gözlem araçları yerleştirildiğinde tüm alan gözlemlenebilmiş oluyor.



Şekil 7.5 Belirleme kodlarının çevresel denetlemede kullanımına bir örnek.

BÖLÜM 8

SONUÇ VE ÖNERİLER

Bu çalışmada anlatılanlardan yola çıkılarak küme kaplamada zor olan bir problemi belirleme kodlarına aktararak ya da tersine, yani belirleme kodlarında çözümü zor olan bir problemi küme kaplama problemine aktararak elimizdeki probleme değişik bir açıdan bakabiliriz.

Vermiş olduğumuz belirleme kodları ile ilgili uygulamalar günlük hayattaki optimizasyon problemlerini eğer mümkünse belirleme kodlarına aktararak çözme konusunda bir fikir verebilir.

KAYNAKLAR

- [1] Karpovsky, M. G., Chakrabarty, K. ve Levitin, L. B., (1998) “A new class of codes for identification of vertices in graphs,” IEEE Transactions on Information Theory, 44, 2, 599–611.
- [2] Ray, S., Ungrangsi, R., Pellegrini, F. D., Trachtenberg, A. ve Starobinski, D., (2003). “Robust location detection in emergency sensor networks,” Proceedings INFOCOM, April 2003.
- [3] Ungrangsi, R., Trachtenberg, A. ve Starobinski, D., (2004). “An implementation of indoor location detection systems based on identifying codes,” Proceedings of the IFIP International Conference on Intelligence in Communication Systems (INTELLCOMM 04) 2004.
- [4] Laifenfeld, M., Trachtenberg, A., Cohen, R., ve Starobinski, D., (2007). “Joint monitoring and routing in wireless sensor networks using robust identifying codes,” IEEE Broadnets 2007 September 2007.
- [5] Berger-Wolf, T. Y., Hart, W. E., ve Saia, J., (2005). “Discrete sensor placement problems in distribution networks,” Journal of Mathematical and Computer Modelling, 42, 13, 1385–1396 December 2005.
- [6] Gravier, S. ve Moncel, J., (2005). “Construction of codes identifying sets of vertices,” The Electronic Journal of Combinatorics, 12, 1.
- [7] Laifenfeld, M. ve Trachtenberg, A., (2005). “Disjoint identifying codes for arbitrary graphs,” IEEE International Symposium on Information Theory, Adelaide, Australia 4-9 September 2005.
- [8] Laifenfeld, M. ve Trachtenberg, A., (2008). “Identifying Codes and Covering Problems,” IEEE Transactions on Information Theory, September 2008.
- [9] Laihonon, T., (1994-2011). Lecture Notes on Graph Theory, Department of Mathematics, university of Turku, FIN-20014, Turku.
- [10] Weiss, W. A. R., (2008). An Introduction to Set Theory, University of Toronto, Toronto.
- [11] Şeker, Ş. E., (2010). Karmaşıklık Sınıfları, <http://www.bilgisayarkavramlari.com/2010/06/17/karmasiklik-siniflari-complexity-classes/>, 15 Mart 2012.
- [12] Demir, İ., (2010). Algoritmalar da Karmaşıklık, [http://e-bergi.com/2010/Ocak/Algoritmalar da-Karmasiklik](http://e-bergi.com/2010/Ocak/Algoritmalar-da-Karmasiklik), 14 Mart 2012.

- [13] Wikipedia, Büyük O gösterimi,
http://tr.wikipedia.org/wiki/B%C3%BCy%C3%BCk_O_g%C3%B6sterimi, 12 Mart 2012.
- [14] Goodrich, M. T., Tamassia, R., (2002). Algorithm Design, John Wiley&Sons, Inc., New York.
- [15] Şeker, Ş. E., (2009). Turing Makinesi,
<http://www.bilgisayarkavramlari.com/2009/06/27/turing-makinesi-turing-machine/>, 15 Mart 2012.
- [16] The MAA Mathematical Sciences Digital Library, Loci: Convergence,
mathdl.maa.org/mathDL/46/?pa=content&sa=viewDocument&nodeId=2429, 20 Mart 2012.
- [17] Ray, S., Starobinski, D., Trachtenberg, A. ve Ungrangsi, R., (2004). “Robust location detection with sensor networks,” IEEE Journal on Selected Areas in Communications (Special Issue on Fundamental Performance Limits of Wireless Sensor Networks), 22, 6, August 2004.
- [18] Gravier, S., Klasing, R. ve Moncel, J., (2006). “Hardness results and approximation algorithms for identifying codes and locating-dominating codes in graphs,” LaBRI, Technical Report RR-1417-06, November 2006.
- [19] Suomela, J., (2007) “Approximability of identifying codes and locating-dominating codes,” Information Processing Letters, 103, 1, 28–33.
- [20] Lovasz, L., (1975). “On the ratio of optimal integral and fractional covers,” Discrete Mathematics, 13, 383–390.
- [21] Slavik, P., (1996). “A tight analysis of the greedy algorithm for set cover,” In Proceedings of the 28th Annual ACM Symposium on Theory of Computing (Philadelphia, Pa.), 435–439, May 1996.
- [22] Harary, F., (1969) Graph Theory. Reading, MA: Addison-Wesley, Massachusetts.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı :Ferhat Kürüz
Doğum Tarihi ve Yeri:02/12/1988, Üsküdar/İstanbul
Yabancı Dili:İngilizce
E-posta:ferhatkuruz@hotmail.com

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Lisans	Matematik	Yıldız Teknik Üniversitesi	2010
Lise	Fen-Matematik	Ataşehir Habire Yahşi Lisesi	2005

İŞ TECRÜBESİ

Yıl	Firma/Kurum	Görevi
2007-2010	Yıldız Teknik Üniversitesi/Matematik Bölümü	Asistan öğrenci
2010-2011	Yıldız Teknik Üniversitesi/Fen Bilimleri Enstitüsü	Asistan öğrenci