



EGE ÜNİVERSİTESİ



E. Ü. FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

MINİMUM TEPE ÖRTÜSÜ PROBLEMİ ÜZERİNE

Semiha EMİNOĞLU

Tez Danışmanı : Yrd. Doç. Dr. Elgin KILIÇ

İkinci Danışmanı : Prof. Dr. Alpay KIRLANGIÇ

Matematik Anabilim Dalı

Bilim Dalı Kodu : 619.03.03

Sunuş Tarihi : 03.07.2014

Bornova-İZMİR

2014

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

(YÜKSEK LİSANS TEZİ)

MİNİMUM TEPE ÖRTÜSÜ PROBLEMİ ÜZERİNE

Semiha EMİNOĞLU

Tez Danışmanı : Yrd. Doç. Dr. Elgin KILIÇ

İkinci danışmanı : Prof. Dr. Alpay KIRLANGIÇ

Matematik Anabilim Dalı

Bilim Dalı Kodu : 619.03.03

Sunuş Tarihi : 03.07.2014

Bornova-İZMİR

2014

Semiha EMİNOĞLU tarafından **YÜKSEK LİSANS** tezi olarak sunulan “**MINİMUM TEPE ÖRTÜSÜ PROBLEMİ ÜZERİNE**” başlıklı bu çalışma E.Ü. Lisansüstü Eğitim ve Öğretim Yönetmeliği ile E.Ü. Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi’nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş vetarihinde yapılan tez savunma sınavında aday oybirliği/oyçokluğu ile başarılı bulunmuştur.

Jüri Üyeleri:

İmza

Jüri Başkanı	:	
Raportör Üye	:	
Üye	:	
Üye	:	
Üye	:	

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ**ETİK KURALLARA UYGUNLUK BEYANI**

E.Ü. Lisansüstü Eğitim ve Öğretim Yönetmeliğinin ilgili hükümleri uyarınca Yüksek Lisans Tezi olarak sunduğum “MİNİMUM TEPE ÖRTÜSÜ PROBLEMİ ÜZERİNE ” başlıklı bu tezin kendi çalışmam olduğunu, sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi, tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını, bu tezin herhangi bir bölümünü bu üniversite veya diğer bir üniversitede başka bir tez çalışması içinde sunmadığımı, bu tezin planlanmasından yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

.... / / 20..

İmzası

Adı-Soyadı

ÖZET

MINIMUM TEPE ÖRTÜSÜ PROBLEMİ ÜZERİNE

EMİNOĞLU, Semiha

Yüksek Lisans Tezi, Matematik Anabilim Dalı

Tez Danışmanı: Yrd. Doç. Dr. Elgin KILIÇ, İkinci Danışmanı: Prof. Dr. Alpay

KIRLANGIÇ

Temmuz 2014, 68 sayfa.

Minimum tepe örtüsü problemi (en çok bilinen graf problemlerinden biri) döngüsüz, birleştirilmiş, yönsüz graflar üzerinde çalışılmıştır. $G=(V,E)$ bir graf olsun. G grafının, her bir $(u,v) \in E$ ayrıtının en az bir uç noktasını içeren, $C \subseteq V$ kümesine, G grafının tepe örtüsü (vertex cover) kümesi denir. Minimum tepe örtüsü probleminde bu kümenin en küçük boyutta olanı bulunmak istenir.

Minimum tepe örtüsü problemi Np-Tamdır ve bu problem için polinom zamanlı algoritma yoktur. Minimum tepe örtüsü probleminin Np-Tam sınıftaki problemlerle ilişkisi bu tezde incelenmiştir. Özellikle tepe örtüsü sayısı ile bağımsızlık sayısı arasındaki ilişki üzerinde durulmuştur. Bağımsızlık sayısı için var olan sonuçlar, tepe örtü sayısı için de kullanılmıştır.

Güçlü çarpım, tensör çarpım, lexicografik çarpım, kartezyen çarpım gibi ikili işlemler sonucu oluşan grafların, tepe örtü ve bağımsızlık sayıları arasındaki ilişki incelenmiştir. Özellikle de herhangi iki grafın kartezyen çarpımıyla oluşan grafın tepe örtüsü ve bağımsızlık sayısı arasındaki ilişki bu tez çalışmasının ana kısmını oluşturmuştur. Bağımsızlık sayısına ilişkin var olan kartezyen çarpım sonuçları kullanılarak, ikili ağaçların kartezyen çarpımıyla oluşan grafın tepe örtüsü sayısı incelenmiştir. Bu grafların tepe örtüsü sayısını bulmak için zeki greedy (aç gözlü yaklaşım) algoritması kullanılmıştır. Bu algoritmanın optimal sonucu garanti etmediği halde, $B_2 \times B_3$ den $B_2 \times B_6$ ye kadar yapılan işlemlerde optimal sonuca ulaştığı gözlenmiştir.

Anahtar Sözcükler : Tepe örtüsü problemi, Yaklaşık algoritma, Kartezyen çarpım, İkili ağaç, Greedy algoritması.

ABSTRACT**MİNİMUM TEPE ÖRTÜSÜ PROBLEMİ ÜZERİNE****EMİNOĞLU, Semiha**

MSc. in Mathematics

Supervisor : Asst. Prof. Elgin KILIÇ, Co-supervisor: Prof. Alpay KIRLANGIÇ

July 2014, 68 pages.

Minimum vertex cover problem (one of the most known graph theory problems) has been studied on looples, connected and undirected graps. Let G be a graph $G=(V,E)$. The subset $C \subseteq V$ which contains at least one end of each $(u,v) \in E$ is called vertex cover of graph G . In minimum vertex cover problem it is desired to find vertex cover set which has the smallest cardinality.

Minimum vertex cover problem is Np-Complete problem and no polynomial time algorithm is known. The relation between minimum vertex cover problem and other problems in NP-Complete class are examined. The results previously given for the independence number is also used for vertex covering number.

The relation between vertex covering number and independence number of graphs obtained by binary operations such as strong product, tensor product, lexicographic product, cartesian product are examined. Especially the existing results obtained by cartesian products for vertex covering and independence number is the main part of this thesis. By using the existing results of cartesian product related to independence number, the vertex covering number of resulting graphs obtained by cartesian product of binary trees has been examined. Greedy is used for finding the vertex covering number of these graphs. Even if the optimal result is not guaranteed by greedy algorithms, it has been observed that optimal outcomes are found for the cartesian products of binary trees from $B_2 \times B_3$ to $B_2 \times B_6$.

Keywords: Minimum vertex cover problem, Approximation algorithm, Cartesian product, Binomial trees, Greedy algorithm.

TEŞEKKÜR

Bu çalışma boyunca tüm içtenliğiyle bana destek olan, çalışmamı tamamlamamı sağlayan, paylaştığı bilgilerle, yaptığı yönlendirme ve rehberlikle katkılarını sunan çok değerli sayın Yrd. Doç. Dr. Elgin KILIÇ' a, ayrıca tecrübe ve bilgi birikimini paylaşan ve tezin ortaya çıkmasında çok büyük katkısı bulunan sayın Prof. Dr. Alpay KIRLANGIÇ' a, teşekkürlerimi sunarım.

Ayrıca 2011-2012 döneminde burs aldığım İzmir Ticaret Odasına teşekkür ederim.Son olarak, maddi-manevi tüm desteklerinden dolayı canım aileme ve müstakbel eşim Dr. Gürsel KALKAN'a teşekkürü bir borç bilirim.

İÇİNDEKİLER

Sayfa

ÖZET	vii
ABSTRACT	ix
TEŞEKKÜR	xi
ŞEKİLLER DİZİNİ	xvi
ÇİZELGELER DİZİNİ	xxi
SİMGELER VE KISALTMALAR DİZİNİ.....	xxii
1.GİRİŞ	1
1.1 Temel Tanım ve Teoremler	3
2.NP-TAMLIK VE TEPE ÖRTÜSÜ PROBLEMİ	8
2.1 NP-Tam Problemler	9
2.2 Polinomiyal İndirgenabilirlik ve NP-Tamlık İspatı	14
2.3 Tepe Örtüsü Probleminin İlişkili Olduğu NP-Tam Problemler	16
2.3.1 Küme örtü problemi(Set Cover)	16
2.3.2 Bağımsız küme (Independent Set)	17
2.3.3 Klik(Clique)	17
2.3.4 Ayrit örtü(Edge Cover)	17
3. TEPE ÖRTÜSÜ KÜMESİ	19

İÇİNDEKİLER (devam)

Sayfa

3.1.1 Karar versiyonu.....	19
3.1.2 Optimizasyon versiyonu.....	19
3.1.3 Karar versiyonu ile optimizasyon versiyonunun karşılaştırılması.....	19
3.2 Tepe Örtüsü Problemi Günlük Hayat Uygulaması	21
3.3 Tepe Örtüsü Probleminin Türleri.....	22
3.4 Tepe Örtüsü Problemi İçin Çözüm Yöntemleri ve Algoritmalar	26
3.4.3 Yaklaşık tepe örtüsü algoritması.....	27
3.4.4 Greedy yöntemi.....	30
3.4.5 Alom'un algoritması	37
3.5 Tepe Örtüsü Problemi İçin Bulunan Yaklaşık Sonuçlar	41
4. TEPE ÖRTÜSÜ VE GRAF İŞLEMLERİ	42
4.1 Ayırıt Çıkarma	42
4.2 Ayırıt Büzme	42
4.3 Grafin Tümleyeni.....	43
4.4 Grafin Kuvveti.....	43
4.5 Toplama İşlemi	44
4.6 Lexicographic Çarpım.....	44

İÇİNDEKİLER (devam)

	<u>Sayfa</u>
4.7 Güçlü Çarpım	45
4.8 Tensör Çarpım	47
4.9 Kartezyen Çarpım	49
4.10 Grafların Çarpımlarının Bağımsızlık Sayıları	50
5. BİNOMİAL AĞAÇLARIN KARTEZYEN ÇARPIMI SONUCU OLUŞAN GRAFLARIN TEPE ÖRTÜ SAYISI	51
5.1 $B_2 \times B_3$ Grafına Greedy Algoritmasının Uygulanması	52
6. SONUÇ VE DEĞERLENDİRME	62
KAYNAKLAR DİZİNİ	64
ÖZGEÇMİŞ	68

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
1.1. Basit bir $G=(V,E)$	3
1.2 Yol graflar	3
1.3 Bir C_5 graf.	4
1.4 Bir ağaç graf	4
1.5 Algoritmik notasyonların grafiksel ifadeleri	7
2.1 Basit bir $G=(V,E)$ grafında klik	10
2.2 Np-Tamlık ispatlarının yapısı	14
2.3 Tepe örtüsü ve klik ilişkisi	15
2.4 Bir $G=(V,E)$ grafı ve tümleyeni $\bar{G}$	18
3.1 Bir $G=(V,E)$ grafı.....	20
3.2 Ağırlıklı tepe örtüsü problemi için örnek	22
3.3 Ağırlıksız bir $G=(V,E)$ grafı	23
3.4 Bir $G=(V,E)$ grafı ve onun birleştirilmiş tepe örtüsü	24
3.5 Maksimum kısmi tepe örtüsü için basit bir graf	25

ŞEKİLLER DİZİNİ (Devam)

<u>Şekil</u>	<u>Sayfa</u>
3.6 Yaklaşık oran hesabı için basit bir graf.....	27
3.7 Yaklaşık Tepe örtüsü algoritmasının uygulanışı	28
3.8 Greedy algoritmasının uygulandığı bir G grafi.	30
3.9 $G=(V,E)$ grafının her bir tepesinin dereceleri ile gösterimi	30
3.10 Greedy algoritmasının uygulanışı 1.adım	31
3.11 Greedy algoritmasının uygulanışı 2.adım	31
3.12 Greedy algoritmasının uygulanışı 3.adım	32
3.13 Greedy algoritmasının uygulanışı 4.adım	32
3.14 Algoritma sonunda bulunan tepe örtüsü.....	32
3.15 Greedy algoritmasının optimalden uzaklaştığı örnek	33
3.16 Greedy algoritmasının uygulanışı 1.adım	33
3.17 Greedy algoritmasının uygulanışı 2.adım	34
3.18 Greedy algoritmasının uygulanışı 3.adım	34
3.19 Greedy algoritmasının uygulanışı 4.adım	35

ŞEKİLLER DİZİNİ (Devam)

<u>Şekil</u>	<u>Sayfa</u>
3.20 Greedy algoritmasının uygulanışı 5.adım	35
3.21 Algoritmanın bulduğu optimal tepe örtüsü.....	36
3.22 Alom ‘un Algoritmasının uygulanışı	37
3.23 Alom ‘un Algoritması optimal çözümün bulunamadığı graf	38
3.24 Alom ‘un Algoritmasının uygulanışı 1.adım	39
3.25 Alom ‘un Algoritmasının uygulanışı 2.adım.....	39
3.26 Alom ‘un Algoritmasının uygulanışı 3.adım.....	39
3.27 Greedy algoritmasının uygulanışı 4.adım	40
3.28 Algoritmanın seçimine bağlı durumda 1.adım	40
3.29 Algoritmanın seçimine bağlı durumda 2.adım	40
4.1 Ayrıt büzme işlemi.....	42
4.2 Grafın tümleyeni	43
4.3 Grafın kuvveti.....	43
4.4 Toplama.....	44

ŞEKİLLER DİZİNİ (Devamı)

<u>Şekil</u>	<u>Sayfa</u>
4.5 Lexicografik çarpım	44
4.6 Güçlü çarpım	46
4.7 Uzaklık fonksiyonu örneği için graf	46
4.8 Tensör çarpım	47
4.9 Kartezyen çarpım	49
5.1 Binomial ağaçlar	51
5.2 $B_2 \times B_3$ grafi	52
5.3 $B_2 \times B_3$ grafına greedy algoritmasının uygulanışı 1.adım.....	53
5.4 $B_2 \times B_3$ grafına greedy algoritmasının uygulanışı 2.adım.....	53
5.5 $B_2 \times B_3$ grafına greedy algoritmasının uygulanışı 3.adım.....	54
5.6 $B_2 \times B_3$ grafına greedy algoritmasının uygulanışı 4.adım.....	54
5.7 $B_2 \times B_3$ grafına greedy algoritmasının uygulanışı 5.adım.....	55
5.8 $B_2 \times B_3$ grafına greedy algoritmasının uygulanışı 6.adım.....	55
5.9 $B_2 \times B_3$ grafına greedy algoritmasının uygulanışı 7.adım.....	56

ŞEKİLLER DİZİNİ (Devam)

<u>Şekil</u>	<u>Sayfa</u>
5.10 $B_2 \times B_3$ grafına greedy algoritmasının uygulanışı 8.adım.....	56
5.11 $B_2 \times B_3$ grafına greedy algoritmasının uygulanışı 9.adım.....	57
5.12 $B_2 \times B_3$ grafına greedy algoritmasının uygulanışı 10.adım.....	57
5.13 $B_2 \times B_3$ grafına greedy algoritmasının uygulanışı 11.adım.....	58
5.14 $B_2 \times B_3$ grafına greedy algoritmasının uygulanışı 12.adım.....	58
5.15 $B_2 \times B_3$ grafına greedy algoritmasının uygulanışı kalan adımlar	59

ÇİZELGELER DİZİNİÇizelgeSayfa

1.1	Algoritmanın türüne göre çalışma zamanları	6
3.1	Eleman sayılarına göre tepe örtüsü kümeleri	20
3.2	Tepe örtüsü için bulunan yaklaşık oranlar	41
5.1	Bulunan sonuçlar	61

SİMGELER DİZİNİ

<u>Simgeler</u>	<u>Açıklama</u>
$G=(V,E)$	Basit bir graf
$V(G)$	G nin tepe kümesi
$E(G)$	G nin ayırıt kümesi
P_n	n tepeli yol graf
C_n	n tepeli çevre graf
O	Big O notasyonu
Θ	Teta notasyonu
Ω	Omega notasyonu
$\leq_p$	Polinomiyal indirgenebilirlik
$\bar{G}$	G nin tümleyeni
$\alpha(G)$	G grafının bağımsızlık sayısı
$\beta(G)$	G grafının tepe örtü sayısı
$K_{m,n}$	İki parçalı tam graf
k_v	v tepesinin kapasitesi
w_v	v tepesinin ağırlığı
$\rho(n)$	yaklaşık oran

SİMGELER DİZİNİ(Devam)

<u>Simgeler</u>	<u>Açıklama</u>
C	Örtü kümesi
C^*	Optimal örtü kümesi
G_C	Greedy algoritmasının bulduğu örtü kümesi
A_C	Alom'un algoritmasının bulduğu örtü kümesi
G^n	G grafinin n . kuvveti
$G_1 + G_2$	İki grafin toplamı
$G_1[G_2]$	Lexicografik çarpım
$G_1 \boxtimes G_2$	Güçlü çarpım
$d_G((u,v))$	G grafında herhangi iki tepe arasındaki uzaklık
$G_1 \otimes G_2$	Tensör çarpım
$G_1 \times G_2$	Kartezyen çarpım
B_k	R köklü k binomial ağaç

1. GİRİŞ

Bilgisayar bilimlerinde, graflar yaygın bir veri yapısı modelidir. Birçok hesaplamalı problem graflara göre modellenebilir. Böylelikle birçok problem daha rahat çözülebilir. Bu tezde minimum tepe örtüsü problemi yönsüz, birleştirilmiş, katlı ayrıt içermeyen, graflar üzerinde çalışılmıştır. (Cormen, Leiserson, Rivest and Stein, 2001)

Graflar ayrıt ve tepelerden oluşur. Bir graf $G=(V, E)$ ile gösterilir. V tepeler kümesini, E ise ayrıtlar kümesini temsil etmektedir. Herhangi bir $G=(V,E)$ grafinin tepe örtüsü kümesi ise, her bir ayrıtın en az bir uç noktasını içeren kümedir. Tepe örtüsü kümesinin minimum boyutta olanını bulmak ise minimum tepe örtüsü problemi olarak adlandırılır. Bu problem, çalışma zamanı polinom olan bir algoritmayla çözülememektedir. Problem, Karp' ın 21 Np-Tam problem listesinde yer almaktadır.

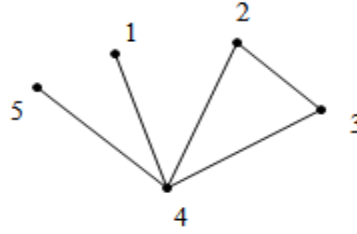
Np-Tam problemleri polinom zamanda çözecek bir algoritma henüz yoktur. Ancak Np-Tam problemlerin herhangi birisi için polinom zamanlı bir algoritma bulunabilirse, diğer Np-Tam problemler için de polinom zamanlı bir algoritma bulunabilecektir. (Garey M. and Johnson D., 1979) Bu durum, Np-Tam problemler üzerindeki çalışmaları arttırmıştır. Minimum tepe örtüsü problemini çözen çalışma zamanı polinom olan bir algoritma bulunamadığından algoritmalarla yaklaşık çözüm bulmak alternatif bir yol olmuştur. Yaklaşık algoritmalar (approximation algorithms), optimale yakın bir çözüm bulmayı amaçlamaktadır. (Cormen, Leiserson, Rivest and Stein, 2001)

Bu çalışmada, birinci bölümde graflara ilişkin temel tanımlar verilmiş, ikinci bölümde karmaşıklık, P ve NP sınıf, Karp'ın 1972 de yayımladığı 21 adet NP-Tam problem, minimum tepe örtüsü probleminin karmaşıklığı gibi konular ele alınmıştır. Üçüncü bölümde ise, tepe örtü probleminin tanımı yapılmış, günlük hayattaki öneminden söz edilmiş, tepe örtüsü probleminin türleri listelenmiş, tepe örtüsü problemi için var olan algoritmalarından bazılarının detaylı incelemesi yapılmıştır. Dördüncü bölümde, graf işlemlerinden bazılarının var olan tanımları verilmiştir. Özellikle de grafların kartezyen çarpım, lexicografik çarpım, güçlü

arpım, tensör arpım gibi işlemlerden sonra bağımsızlık sayısı ve tepe örtü sayısına ilişkin var olan sonuçlar ele alınmıştır. Beşinci bölümde ise $B_2 \times B_3$ kartezyen arpımı yapılmış, oluşan graf üzerinde greedy algoritması uygulanmıştır. Tarafımızdan yapılan $B_2 \times B_3$, $B_2 \times B_4$, $B_2 \times B_5$, $B_2 \times B_6$ arpımları sonucu oluşan graflara greedy algoritması uygulanmış ve ulaşılan sonuçlara yer verilmiştir. En son kısımda genel sonuç ve yapılan değerlendirmelere yer verilmiştir.

1.1 Temel Tanım ve Teoremler

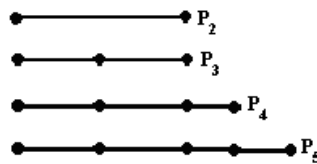
Tanım 1.1.1. (Diestel, 2000) Bir graf (çizge), $G=(V,E)$ ikilisinden oluşmaktadır . $E \subseteq V^2$, yani E kümesinin elemanları, V kümesinin iki elemanlı alt kümeleridir. V kümesinin elemanlarına tepe, düğüm veya nokta denir. E kümesinin elemanlarına ise ayrıt veya çizgi denir. V , tepeler kümesini, E , ayrıtlar kümesini temsil etmektedir. Bir grafın çizimi yapılırken genelde herhangi iki tepeyi birleştiren ayrıt adı verilen bir çizgi çizilir.



Şekil 1.1 Bir $G=(V,E)$ grafı. $V=\{1,2,3,4,5\}$,
 $E=\{(1,4),(5,4),(2,4),(3,4),(2,3)\}$.

Ayrıca bu tezde $V(G)$ sembolü G grafının tepeler kümesini, $E(G)$ sembolü ise G grafının ayrıtlar kümesini göstermektedir.

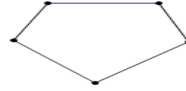
Tanım 1.1.2. (Diestel, 2000) Yol graf, başlangıç ve bitiş tepesinin derecesi 1, diğer bütün tepelerinin derecesi 2 olan graftır. n tepeli bir yol graf P_n ile gösterilir.



Şekil 1.2 Yol grafları.

Tanım 1.1.3. (Diestel, 2000) Çevre, başlangıç ve bitiş tepesi aynı olan en az 3 uzunluğunda bir yoldur, başlangıç ve bitiş tepesi hariç diğer tepeler tekrar etmez. Diğer bir deyişle bir graftaki çevre(cycle) aşağıdaki formdadır.

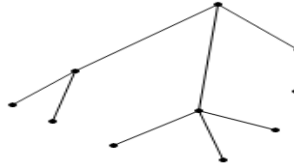
$G=(V,E)$ ve $V=\{v_1, v_2, \dots, v_n\}$, $E=\{v_1v_2, v_2v_3, \dots, v_{n-1}v_n, v_nv_1\}$, n tepeli bir devre C_n ile gösterilir.(Bkz. Şkl 1.3).



Şekil 1.3 Bir C_5 graf.

Tanım 1.1.4. (Diestel, 2000) Bir yönsüz $G=(V,E)$ grafında, her tepe çifti arasında en az bir yol varsa bu graf birleştirilmiştir.

Tanım 1.1.5. (Diestel, 2000) Devre içermeyen birleştirilmiş graflara ağaç denir. Aşağıda bir ağaç graf gösterilmiştir.



Şekil 1.4 Bir ağaç graf.

Teorem 1.1.1.

$G=(V,E)$ bir birleştirilmiş graf ise aşağıdakiler denktir.

- G bir ağaçtır.
- G , devre içermez.
- G deki her bir $u \neq v$ tepe çifti için, u dan v ye sadece bir yol vardır.
- G den herhangi bir ayrıt silindiğinde, birleştirilmiş olmayan bir graf elde edilir.
- G nin tepelerinin sayısı, G nin ayrıtlarının sayısından bir fazladır (Scheinerman, 2012).

Tanım 1.1.6. (Nabiyev Vagıfoğlu, 2009) Algoritma, iyi tanımlı hesaplamalı bir prosedür olup bir değer veya değerler kümesini girdi olarak alıp, bir değer veya değerler kümesini işleyip çıktı olarak sunmaktadır. Algoritma, iyi belirlenmiş hesaplamalı bir problemi çözmek için bir araçtır. Bir algoritma böylece girdileri, sıralı hesaplamalı adımlarla çıktıya dönüştürür. İyi belirlenmiş problemler, çözümün doğruluğunun algoritmik yaklaşımla gösterilebileceği problemlerdir. Algoritmik yaklaşımda ise çözüm uzayının her bir elemanının çözüm şartlarına uyup uymadığı birer birer kontrol edilmektedir.

Bir algoritma genel olarak, problemler için

1. Birçok aday çözüm vardır ki bunların çoğu istenilen çözüm değildir. Bunlardan istenileni oldukça iyi bir şekilde bulabilir.
2. Pratik uygulamalarda, basit örnekleri en kısa yoldan sağlar (Cormen, Leiserson, Rivest and Stein, 2001).

Tanım 1.1.7. (Nabiyev Vagıfoğlu, 2009) Bir problemin algoritmik çözümlerinin sınıflandırılması, bunların yürütülmesi için gerekli işlemlerin sayısı temel alınarak gerçekleştirilebilir, bu ölçüye algoritmik karmaşıklık denir. Karmaşıklık, algoritmanın uygulanacağı bilgisayarın işlemcisinden bağımsız olarak hesaplanır. Karmaşıklık, zaman karmaşıklığı ve alan karmaşıklığı olarak ikiye ayrılmaktadır. Zaman karmaşıklığı, algoritmanın problemi ne kadar zamanda çözeceğine ilişkin bir ölçüyken, alan karmaşıklığı problemin çözümünün bellekte ne kadar yer kaplayacağına ilişkin ölçüdür. Günümüzde, gelişmiş bilgisayar teknolojisini düşünülürse, alan karmaşıklığı çok da önemli değildir.

Tanım 1.1.8. Bir algoritmanın en kötü durumda sergileyeceği performansın üst sınırı Big O notasyonu ile gösterilir. Bu fonksiyon şöyle tanımlanır. $f(n)=O(g(n))$ olması için tüm $n \geq k$ için $0 \leq f(n) \leq cg(n)$ olacak şekilde pozitif c ve k , sabitleri tanımlıysa bu f fonksiyonu büyük O notasyonu ile gösterilebilir. Burada c ve k , f fonksiyonuna bağlı, n den bağımsız sabitler olmalıdır.

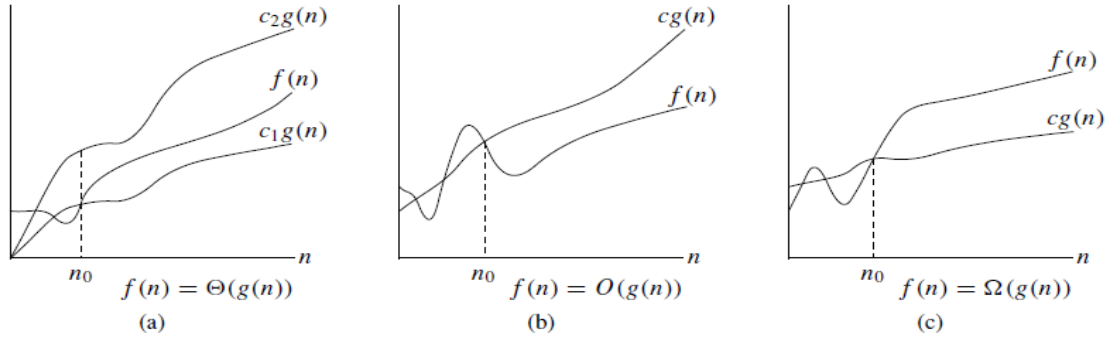
Örneğin, n^3+n^2+4n+5 fonksiyonunda, n nin çok büyük değerleri göz önünde tutulursa, n^3 ün büyümesine karşılık diğer terimlerin büyümesi önemsenmez ve algoritmanın karmaşıklığı böyle bir fonksiyon için $c \cdot O(n^3)$, $c=1$ şeklinde yazılacaktır. Aşağıdaki tabloda algoritmaların türüne göre çalışma zamanları listelenmiştir.

Çizelge 1.1 Algoritmaların türüne göre çalışma zamanları

Notasyon	İsim
$O(1)$	Sabit
$O(\log n)$	Logaritmik
$O([\log n]^c)$	Polilogoritmik
$O(n)$	Lineer
$O(n \log n)$	Loglineer
$O(n^2)$	Karesel
$O(n^c, c>1)$	Polinomial veya cebirsel
$O(c^{cn})$	Üstel veya geometrik
$O(n!)$	Faktöriyel veya kombinatoryal
$O(n^n)$	geokombinator

Tanım 1.1.9. (Nabiyev Vagıfoğlu, 2009 ; Cormen at al, 2001) Teta Notasyonu, ortalama çalışma zamanı olup algoritmanın karakteristiğine ve çözülen probleme göre değişmektedir. Örneğin sıralama algoritmalarında genellikle rastgele dağılımlı değerler ortalama zamanı belirlemektedir. $f(n)=\Theta(g(n))$ olması için, tüm $n \geq k$ için $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ olacak şekilde pozitif c_1, c_2 ve k sabitleri tanımlıysa bu f fonksiyonu Θ notasyonu ile gösterilebilir. Burada c_1, c_2 ve k, f fonksiyonuna bağlı, n den bağımsız sabitler olmalıdır.

Tanım 1.1.10. Omega notasyonu, algoritmaların en iyi durumdaki çalışma zamanını ifade etmek için kullanılır. Giriş değerleri analiz edildikten sonra, en iyi giriş için algoritmanın O notasyonu ile ifadesi Ω notasyonu ile adlandırılır.



Şekil 1.5 Algoritmik notasyonların grafiksel ifadeleri.

Tanım 1.1.11. (Gass,2000) Optimizasyon, bir sistemde var olan kaynakların (iş gücü, zaman, anapara, süreçler, ham maddeler, kapasite ekipman gibi) en verimli şekilde kullanarak belirli amaçlara (maliyet en azaltılması, en çoklanması, kapasite kullanımının en yükseltilmesi ve verimliliğin en çoklanması gibi) ulaşmayı sağlayan bir teknoloji olarak tanımlanmaktadır.

Optimizasyonda, modelleme ve çözümleme iki önemli bileşendir.

Modelleme gerçek yaşamda karşılaşılan problemlerin matematiksel olarak ifade edilmesi; çözümleme ise bu modeli sağlayan en iyi çözümün elde edilmesini kapsamaktadır.

Tanım 1.1.12. (Polinomiyal İngirgenebilirlik) Bir X problemi, Y problemini çözen bir algoritmadan yararlanılarak elde edilen bir algoritma ile çözülebiliyorsa, X problemi Y problemine indirgenebilir denir ve $X \leq_p Y$ ile gösterilir.

Burada, X ya da Y için polinom zamanlı bir algoritmanın varlığı önemli değildir. Burada yalnızca birini çözen iyi bir algoritma varsa diğer için de bulunabileceği söylenmektedir.

2. NP-TAMLIK VE TEPE ÖRTÜSÜ PROBLEMİ

Tanım 2.1.1. P sınıf, çalışma zamanı polinomial bir fonksiyonla sınırlı algoritmalarla çözülebilen problemlerin sınıfıdır. Diğer bir deyişle, bu sınıftaki problemler, iyi belirlenmiş problemlerdir ve deterministik Turing makinesinde polinomial zamanda çözülebilirler.

Örneğin: n basamaklı bir sayının asal olup olmadığını kontrol etmek için çalışma süresi n^6 mertebesinde bir polinomla hesaplanabilen bir algoritma vardır. Bu problem ve birçok karar problemi (Decision problems) P sınıfına girmektedir.

Tanım 2.1.2. NP sınıf, polinomial zamanda deterministik olmayan makinelerde çözülen problemlerin sınıfını göstermektedir. Bu problemlerin çözümü için polinomial zamanda çalıştığı bilinen bir algoritma henüz yoktur.

Deterministik olmayan makineler, bir problem için olası bütün durumları dener ve bu durumları denerken sezgisel olarak çalışmaktadır.

$P \subseteq NP$ olduğu açıktır. Örneğin, n basamaklı bir sayının asal çarpanları bulunmak istendiğinde bu problemi çözecek algoritmanın çalışma zamanı üstel bir fonksiyonla gösterilmektedir.

Tanım 2.1.3. Bir P problem, NP bir problem kadar zorsa bu tür probleme NP-karmaşık problem denir.

Tanım 2.1.4. (Garey M. and Johnson D., 1979 ; Nabyev Vagıfoğlu, 2009)

NP-Tam sınıf, her biri en az NP sınıfındaki problemler kadar zor olan problemlerin sınıfıdır. Yani NP deki her problem NP-karmaşıktır.

NP-Tam problemler, algoritmalar aracılığıyla yeterince hızlı olarak çözülemez. Bu problemler için daha önceden bulunmuş polinom zamanda çözüm verebilen bir algoritma yoktur. NP-Tam problemler, eğer içlerinden herhangi bir problem için polinom zamanda etkili bir çözüm veren bir algoritma varsa diğer NP-Tam problemler için de böyle bir algoritmanın var olduğu gibi bir özelliğe sahiptir (Garey M. and Johnson D., 1979) .

2.1 NP-Tam Problemler

NP-Tam sınıftan problemlere ilişkin ilk çalışma 1971 yılında Stephen Cook's tarafından Bool Tatmin edilebilirlik probleminin NP-tam olduğunun gösterilmesiyle başlamıştır. Bu çalışmadan sonra NP-Tamlık üzerine yapılan çalışmalar hız kazanmış ve 1972 yılında Richard Karp tarafından Kombinatoryal Problemler Arasında İndirgenebilirlik adlı makale yayımlanmış ve 21 adet NP-tam problem listelemiştir. (Karp, 1972)

1. Tatmin edilebilirlik (Satisfiability)

Girdi: Değişkenlerin bir U kümesi, $U = \{u_i \mid u_i \in \{0,1\}, i=1..n\}$ ve $f(u_1, u_2, \dots, u_n)$ fonksiyonu.

Soru: $\exists u_i, i=1..n$, için $f(u_1, u_2, \dots, u_n)=1$ ataması var mı ?

Bu probleme örnek olarak $f=(x \vee y \vee z) \wedge \bar{x}$ fonksiyonunu ele alalım. Bu ifadenin 1 değerini alması için mutlaka $(x \vee y \vee z)$ ve $\bar{x}$ değerlerinin 1 e eşit olması gerekir.

$\bar{x}=1$ ise $x=0$ dır. Böylece problem, $x=0, y=0, z=1$, $x=0, y=1, z=0$ ve $x=0, y=1, z=1$ olmak üzere üç adet durumda tatmin edilebilir.

2. 0-1 Tamsayılı Programlama (0-1 Integer Programming)

Girdi: X sonlu bir küme ve $(\bar{x}, b)$ ikilisi, burada $\bar{x}$, m - li bir tam sayı, b bir tam sayı, c m - li bir tam sayı ve B bir tamsayı olsun.

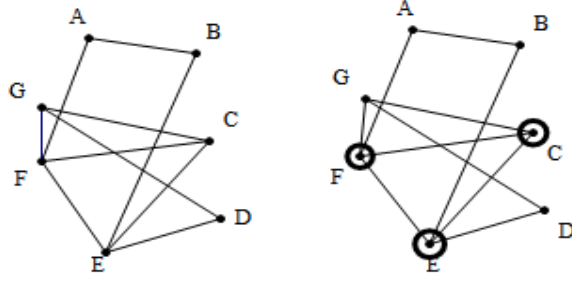
Soru: Tüm $(\bar{x}, b) \in X$ için $\bar{x} \cdot \bar{y} \leq b$ ve $C \cdot \bar{y} \geq B$ olacak şekilde $y \in \{0,1\}$, m -li $\bar{y}$ tam sayısı var mıdır ?

3. Klik (Clique)

Girdi : $G=(V,E)$ grafi ve pozitif K sayısı verildiğinde $K \leq |V|$.

Soru : $V' \subseteq V$ ve $|V'| \geq K$ olacak şekilde V' de her iki tepesi E de bir ayrıt tarafından birleştirilen en az K boyutunda klik içeren G grafi var mıdır ?

(Bkz. Şekil 2.1)



Şekil 2.1 Basit bir $G=(V,E)$ grafi verilmiştir ve klik örneği.
Bu graf için C, E,F klik formundadır.

4.Küme Paketleme (Set packing)

Girdi: C sonlu kümeler yığını, $K \leq |C|$ pozitif tam sayısı.

Soru: C en az K karşılıklı olarak ayrık küme içeriyor mu ?

5. Tepe Örtüsü (Vertex Cover)

Girdi : $G=(V,E)$ grafi ve pozitif K sayısı verildiğinde $K \leq |V|$.

Soru:G için $V' \subseteq V$ ve $K \leq |V|$ olacak şekilde ve her bir $\{u,v\} \in E$ için u ya da v den en az biri V' ne ait olmak üzere bir V' var mı ?

6.Küme Örtme (Set Covering)

Girdi: $U= \{u_1,u_2,\dots,u_n\}$ ve $S_1,S_2,\dots,S_k \subseteq U$.

Soru: $I \subseteq \{1,2,\dots,m\}$ ve $\bigcup_{i \in I} S_i = U$ var mı ?

7.Geri Besleme Düğüm Kümesi (Feedback Node Set)

Girdi: yönlü bir $G=(V,A)$ grafi ve pozitif K sayısı verildiğinde $K \leq |V|$.

Soru: $V' \subseteq V$ ve $V' \leq |K|$ olmak üzere, G deki her yönlü devreden en az bir tepe içeren V' var mı ?

8. Geri Besleme Yay Kümesi (Feedback Arc Set)

Girdi: yönlü bir $G=(V,A)$ grafi ve pozitif K sayısı verildiğinde $K \leq |V|$.

Soru: $A' \subseteq A$ ve $K \leq |A'|$ olmak üzere, G deki her yönlü devreden en az bir arc içeren A' var mı ?

9. Yönlü Hamilton Devresi (Directed Hamilton Circuit)

Girdi: Yönlü $G=(V,A)$ grafi.

Soru: G yönlü bir Hamilton devresi içeriyor mu ?

10. Yönsüz Hamilton Devresi (Undirected Hamilton Circuit)

Girdi: $G=(V,E)$ grafi.

Soru: G Hamilton devresi içeriyor mu ?

11. Madde başına en fazla 3 sabitli Tatmin edilebilirlik (Satisfiability with at most 3 literals per clause)

Girdi: Sonlu bir U üzerindeki cümleciklerin $C=\{c_1, c_2, \dots, c_m\}$ koleksiyonu, öyle ki $1 \leq i \leq m$ için $|c_i| = 3$.

Soru: C deki tüm cümlecikleri tatmin eden U için doğru bir atama var mıdır ?

12. Kromatik Sayı (Chromatic Number)

Girdi: $G=(V,E)$ grafi ve pozitif K sayısı verildiğinde $K \leq |V|$.

Soru : G , K -boyanabilir midir yani, $\{u,v\} \in E$ olduğunda, $f(u) \neq f(v)$ olacak şekilde $f:V \rightarrow \{1,2,\dots,K\}$ var mıdır?

13.Klik Örtü (Clique Cover)

Girdi: $G=(V,E)$ grafi ve pozitif K sayısı verildiğinde $K \leq |E|$.

Soru: V nin, $V_1, V_2, V_3, \dots, V_k$ ve $k \leq K$ olacak şekilde alt kümeleri öyle ki G nin iki parçalı tam altgrafını oluşturan ve her bir $\{u,v\} \in E$ ayrıtı için, hem u hem de v yi içeren bir V_i var mıdır ?

14.Tam Örtü (Exact Cover)

Girdi : X kümesi, $|X| = q$ ve X in alt kümelerinin bir C koleksiyonu,

Soru: Bir $C' \subseteq C$ alt koleksiyonu ve öyle ki, X in her bir elemanı C' nin tam olarak bir üyesini meydana getirir. C, X için bir tam örtü içeriyor mu ?

Örneğin $X=\{A, B, C, D\}$ kümelerinin bir koleksiyonu olsun. $C=\{1,2,3,4\}$ ve $A=\{1\}$, $B=\{1,3\}$, $C=\{2,4\}$, $D=\{2,3\}$ olsun.

$\{B,C\}$ kümelerinin birleşimi X kümesini verir ve bir tam örtüdür.Yine $\{A,B,C\}$ kümelerinin birleşimi de X i meydana getirir bu da tam örtüdür.

15.İsabet Küme (Hitting Set)

Girdi: Bir sonlu S kümesinin alt kümelerinin C koleksiyonu, ve pozitif $K \leq |S|$ sayısı.

Soru: $S' \subseteq S$, $|S'| \leq K$ öyle ki C deki her bir elemanın en az bir tanesini içeren S' alt kümesi var mı?

Bu problemin dualı küme örtü problemidir.

16.Steiner Ağacı (Steiner Tree)

Girdi: $G=(V,E)$ grafi ve her bir $e \in E$ için

$w(e) \in \mathbb{Z}_0^+$ ağırlığı, $R \subseteq V$ ve bir pozitif tamsayı B sınırı.

Soru: R deki tüm tepeleri içeren ve alt ağaçtaki ayrıtların ağırlıkları toplamı B den daha fazla olmayan bir G nin alt ağacı var mıdır ?

17.3-Boyutlu Eşleme (3-Dimensional Matching)

Girdi: $M \subseteq W \times X \times Y$ burada W, X, Y aynı sayıda q elemana sahip ayrık kümeler olsun.

Soru: $M' \subseteq M$ olacak şekilde, $|M'| = q$ ve M' nin herhangi bir koordinatta uyuşan iki elemanı olmayan bir alt küme var mı yani M bir eşleme içeriyor mu ?

18.Sırt çantası (Knapsack)

Girdi: Sonlu bir U kümesi her bir $u \in U$ için $s(u) \in \mathbb{Z}^+$ boyutu ve bir $v(u) \in \mathbb{Z}^+$ değeri, pozitif B ve K sayıları.

Soru: $\sum_{u \in U'} s(u) \leq B$ olacak şekilde ve $\sum_{u \in U'} v(u) \geq K$ olacak şekilde $U' \subseteq U$ var mıdır ?

19.İş Sıralama (Job Sequencing)

Girdi: n adet iş ve m adet makine .

Soru: Her bir iş her makineye en az zamanda yapılacak biçimde atanabilir mi ?

20.Parçalama (Partition)

Girdi: Sonlu bir A kümesi ve, her bir $a \in A$ için $s(a) \in \mathbb{Z}^+$ boyutu

Soru : $A' \subseteq A$ olacak şekilde $\sum_{a \in A'} s(a) = \sum_{a \in A-A'} s(a)$, bir A' alt kümesi var mı ?

21.Maksimum Kesim (Max Cut)

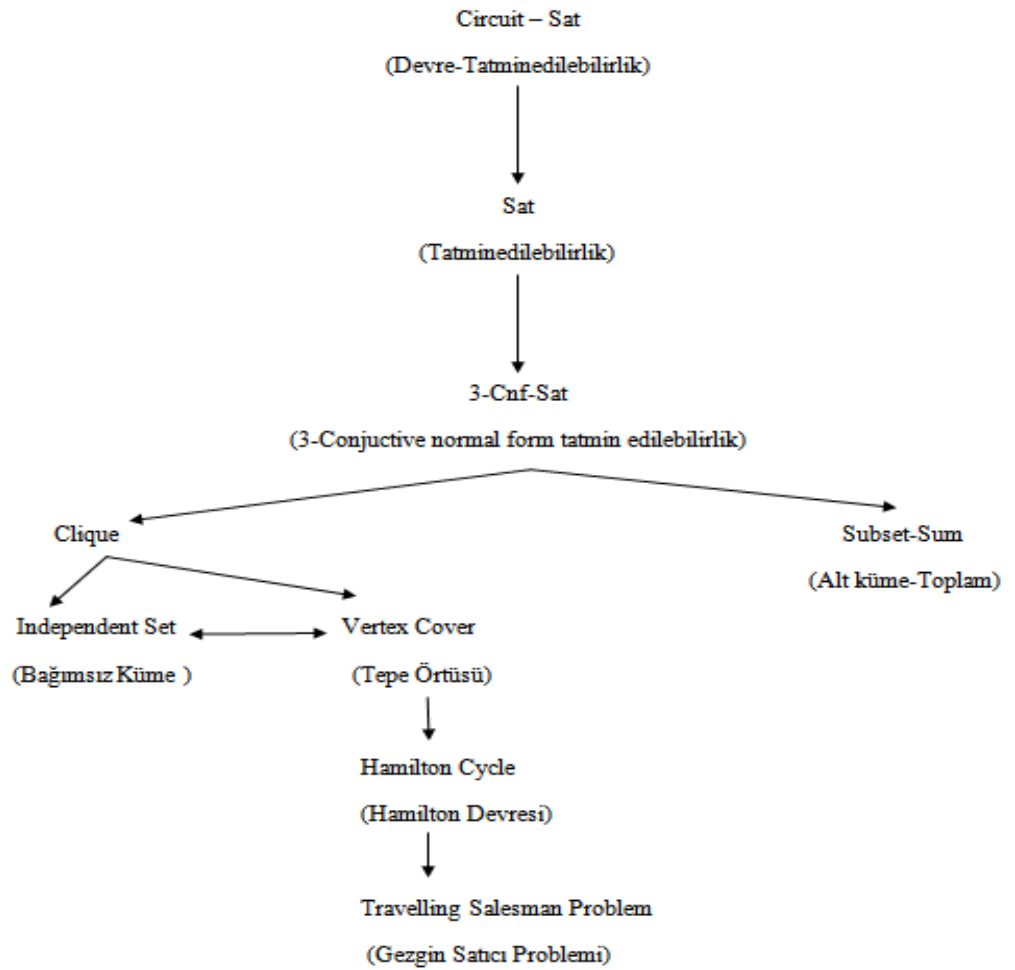
Girdi: her bir $e \in E$ için $w(e) \in \mathbb{Z}^+$ ağırlığı, pozitif K tamsayısı.

Soru: V_1, V_2 ayrık kümeler öyle ki bir bitiş noktası V_1 de olan diğer bitiş noktası V_2 de olan ve E nin ayrıtlarının ağırlıklarının toplamı en az K olan, V nin bir parçalanması var mı ?

2.2 Polinomiyal İndirgenbilirlik ve NP-Tamlık İspatı

Minimum tepe örtüsü problemi Karp' ın 21 Np-Tam probleminden biridir.(Karp,1972) Bir problemin Np-Tam olduğunu ispat etmek için ise, ilk olarak problemin NP sınıfı olduğunun gösterilmesi gerekir. Daha sonra NP-Tam olduğu ispat edilmek istenen problem, NP-tam olduğu bilinen başka bir probleme indirgenir. Eğer bu indirgeme polinom zamanda yapılabilirse, problemin NP-tam olduğu ispat edilmiş olur.

Tepe örtüsü probleminin diğer NP-tam problemlerle bu çerçevede ilişkisi mevcuttur. Yukarıda bahsi geçen indirgenebilirliği aşağıdaki şema göstermektedir.

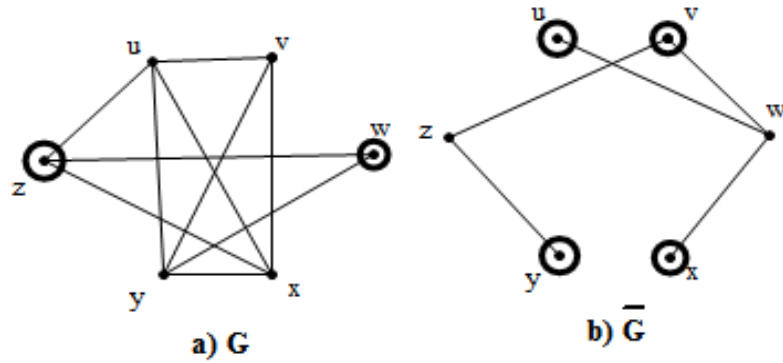


Şekil 2.2 Np-Tamlık ispatlarının yapısı

Teorem 2.2.1. .(Cormen et all, 2001; Karp,1972).Tepe Örtüsü Np-Tamdır.

İspat : İlk olarak Tepe Örtüsü $\in NP$ olduğu ispatlanır. $G=(V,E)$ herhangi bir graf ve k de bir tamsayı olsun . $V' \subseteq V$ tepe örtü kümesi olsun. Doğrulama algoritması $|V'| = k$ olduğunu sağlar ve her $(u,v) \in E$ ayrıtı için ya $u \in V'$ ya da $v \in V'$ yi kontrol eder.Açıkça, bu doğrulama polinom zamanda yapılabilir.

Şimdi tepe örtüsü probleminin $\in NP$ -Zor olduğunu, $KLİK \leq_p TEPE\text{-}ÖRTÜ$ (Yani tepe örtüsü probleminin klik problemine polinom zamanında indirgendliğini) göstererek ispat edelim. $G=(V,E)$ klik problemi için bir graf olsun. Tepe Örtüsü problemi için yeni bir $\bar{G}$ tümleyen graf oluşturalım. Bir grafın tümleyeni $\bar{G}$, G grafında olmayan ayrıtlardan oluşmaktadır.Yani $\bar{G}(V,\bar{E})$, $\bar{E}=\{(u,v): u, v \in V, u \neq v, (u,v) \notin E\}$.



Şekil 2.3 a) da $G=(V,E)$ grafi verilmiştir. $|V'|=\{u, v, x, y\}$ bir kliktir. b) indirgeme algoritmasının ürettiği $\bar{G}$ grafıdır, $V-V'=\{w, z\}$ tepe örtüsüdür.

İndirgeme algoritması girdi olarak, klik probleminin $\langle G, k \rangle$ örneğini alır.Çıktı olarak da tepe örtüsü probleminin bir $\langle \bar{G}, |V|-k \rangle$ örneğini verir.

$|V|=n$, $k'=n-k$ olsun .

G nin bir k -klik içerdiğini göstermek zorundayız ancak ve ancak $\bar{G}$, k' boyutta bir tepe örtüsüne sahipse.

Kabul edelim ki G bir k -klik içeren bir graf ve $V' \subseteq V$ olsun. İddia ediyoruz ki $V-V', \bar{G}$ de bir tepe örtüsüdür. Bunu görmek için, $(u,v) \in E'$ ayrıtına bakalım. Açıkça, $(u,v) \in E$. Böylelikle ya u ya da v , V' ne ait değildir. O zaman u ya da v , $V-V'$ ye aittir. Böylelikle $\bar{G}$ nin bir tepe örtüsü $V-V'$ dir. Boyutu

$$|V-V'| = n-k = k'.$$

Aksine, kabul edelim ki $\bar{G}$, $V' \subseteq V$ olacak şekilde bir tepe örtüsüne sahip olsun. Burada

$|V'| = n-k = k'$, o zaman herhangi bir $u, v \in V$ için, eğer $(u,v) \in E'$ ise, $u \in V'$ veya $v \in V'$ ya da her ikisi de. Bu eğer, $u \in V'$ ve $v \in V'$ ise o zaman $(u,v) \in E$ veya $u, v \in E$ olmasını gerektirir. Böylece G nin bir kliği $V-V'$ dür, boyutu $|V-V'| = n-k = k$.

Böylece, $KLİK \leq_p TEPE-ÖRTÜ$ olduğunu ispatlamış oluruz.

Sonuç olarak $TEPE-ÖRTÜ \in NP\text{-Tam}$. (Cormen, Leiserson, Rivest and Stein, 2001)

2.3 Tepe Örtüsü Probleminin İlişkili Olduğu NP-Tam Problemler

2.3.1 Küme örtü problemi (Set Cover)

(Cormen et al., 2002; Garey M. and Johnson D., 1979).

Tepe örtüsü problemi küme örtü probleminin özel bir türüdür. Küme örtü problemi ise aşağıda ifade edilmiştir.

Küme Örtü probleminin bir örneği (X, F) olmak üzere, X , sonlu küme ve F , X in altkümelerinin bir ailesidir, öyle ki X in her elemanı F deki altkümenin en az birine aittir.

$$X = \bigcup_{S \in F} S$$

Diğer bir tanımı ise,

Girdi: $U = \{u_1, u_2, \dots, u_n\}$ ve $S_1, S_2, \dots, S_k \subseteq U$ olmak üzere,

Soru: $I \subseteq \{1, 2, \dots, m\}$ ve $\bigcup_{i \in I} S_i = U$ var mı ?

2.3.2 Bağımsız küme (Independent Set)

Girdi: $G=(V,E)$ grafi ve pozitif K sayısı, $K \leq |V|$.

Soru: G , K boyutta ya da daha fazla bir bağımsız küme içeriyor mu öyle ki $V' \geq K$ ve $V' \subseteq V$, V' ise E den bir ayrıtla birleştirilmiş iki tepe içermeyen bir alt kümedir.

Bağımsızlık sayısı ise en büyük boyuttaki bağımsız kümenin boyutudur. Genel olarak bir G grafinin bağımsızlık sayısı $\alpha(G)$ ile gösterilir.

2.3.3 Klik (Clique)

Girdi: $G=(V,E)$ grafi ve pozitif J sayısı, $J \leq |V|$

Soru: G , J boyutta ya da daha fazla bir klik içeriyor mu öyle ki $V' \geq J$ ve $V' \subseteq V$, V' deki her tepe E den bir ayrıtla birleştirilmiştir.

2.3.4. Ayrıt örtüsü (Edge Cover)

Girdi: $G=(V,E)$ grafi ve pozitif K sayısı.

Soru : $E' \leq K$ ve $E' \subseteq E$ olacak şekilde bir E' var mıdır öyle ki her bir $v \in V$ için $v \in e$ olan bir $e \in E'$ vardır.

Tepe örtüsü probleminin özellikle bağımsız küme ve klik problemiyle ilişkisi Lemma 2.3.1.de daha detaylı olarak incelenmiştir.

Lemma 2.3.1.

Herhangi bir $G=(V,E)$ ve $V' \subseteq V$ için aşağıdakiler denktir .

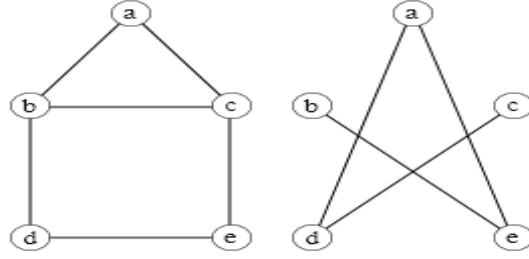
(a) V' , G de bir tepe örtüsüdür.

(b) $V-V'$, G nin tümleyeninde bir kliktir. G nin tümleyeni $\bar{G}=(V,\bar{E})$ olup,

$\bar{E} = \{(u,v): u,v \in V \text{ ve } (u,v) \notin E\}$, yani G nin tümleyeninin ayrıtları G de olmayan ayrıtlardır.

(c) $V-V'$, G için bir bağımsız kümedir(Garey M. and Johnson D., 1979).

Örnek olarak aşağıdaki graflar ele alınmıştır.



Şekil 2.4 Bir $G=(V,E)$ grafi ve tümleyeni $\bar{G}$.

$V' \subseteq V$ olmak üzere, $V' = \{b,c,e\}$ bir tepe örtü kümesidir.

$V-V' = \{a,d\}$ $\bar{G}$ de bir kliktir.

$V-V' = \{a,d\}$ G de bir bağımsız kümedir.

3. TEPE ÖRTÜSÜ KÜMESİ

Tanım 3.1. $G=(V,E)$ bir graf, V tepeler kümesi, E ayrıtlar kümesi olsun. G grafının, her bir $(u,v) \in E$ ayrıtının en az bir uç noktasını içeren, $u \in C$ ya da $v \in C$ ya da $(u,v) \in C$, $C \subseteq V$ kümesine, G grafının tepe örtüsü (vertex cover) kümesi denir. Bu kümenin boyutu, tepe örtüsü sayısı olarak adlandırılır. Bir G grafının tepe örtüsü sayısı, $\beta(G)$ ile gösterilir.

Tepe örtüsü problemi aşağıdaki gibi iki türde tanımlanabilir.

3.1.1 Karar Versiyonu

Girdi : Bir $G=(V,E)$ grafi ve pozitif K sayısı, $K \leq |V|$.

Soru: G için, $|C| \leq K$ olacak şekilde, her bir $(u,v) \in E$ için u ya da v den en az biri C kümesine ait olmak üzere bir $C \subseteq V$ var mı ? (Karp, 1972)

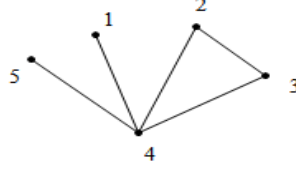
3.1.2 Optimizasyon Versiyonu

Bir $G=(V,E)$ yönsüz graf olmak üzere, G grafindaki her bir $(u,v) \in E$ ayrıtının en az bir uç noktası C kümesinde olmak üzere $C \subseteq V$ olacak şekildeki en küçük boyuttaki C kümesi tepe örtüsü kümesidir.

3.1.3 Karar Versiyonu ve Optimizasyon Versiyonun Karşılaştırılması

Karar versiyonunda amaç, bir G grafi için özel olarak verilen bir boyutta veya ondan daha küçük bir tepe örtüsünün var olup olmadığını doğrulamaktır. Optimizasyon versiyonunda ise amaç, bir G grafi için minimum boyuttaki tepe örtüsünü elde etmektir. Dolayısıyla minimum tepe örtüsü tepe örtü probleminin optimizasyon versiyonudur.

Örnek olarak şekil 3.1 deki $G=(V,E)$ grafinı ele alalım.



Şekil 3.1 Bir $G=(V,E)$ grafi

$V=\{1,2,3,4,5\}$ kümesi G grafinin tepe kümesidir.

$C=\{1,2,3,5\}$, G grafinin bir tepe örtüsü kümesidir.

Öte yandan problem karar versiyonu ve optimizasyon versiyonu açısından ele alınırsa,

Karar Versiyonu:

Girdi: Bir $G=(V,E)$ grafi ve pozitif sayısı, $K=3$ olsun.

Soru : G grafinin her bir ayrıtının en az bir uç noktası C kümesinde olmak üzere $|C| \leq 3$ olacak şekilde $C \subseteq V$ var mıdır ?

Örnekte verilen yönsüz G grafinin 3 veya 3 den daha az elemanlı alt kümelerinin sayısı $\binom{5}{3} + \binom{5}{2} + \binom{5}{1} = 25$ olup tepe örtüsü kümesi için 25 aday küme vardır.

Çizelge 3.1 Eleman sayılarına göre tepe örtüsü kümeleri.

3 elemanlı tepe örtüsü kümeleri	$C_1=\{2,3,4\}$
2 elemanlı tepe örtüsü kümeleri	$C_2=\{2,4\}$ $C_3=\{3,4\}$
1 elemanlı tepe örtüsü kümeleri	----

Optimizasyon versiyonu:

Verilen yönsüz G grafinin, en küçük boyuttaki tepe örtü kümesi $C_2=\{2,4\}$ veya $C_3=\{3,4\}$ olup optimal yani en küçük boyuttaki tepe örtüsü $|C^*|=2$ dir.

3.2 Tepe Örtüsü Problemi Günlük Hayat Uygulaması

Tepe örtüsü probleminin gerçek hayattaki birçok alanda uygulaması mevcuttur. İletişim ağları, elektrik devre tasarımı, istatistiksel mekanik, hesaplamalı biyokimya, güvenlik sistemleri gibi.

Örneğin güvenlik sorunu yaşayan bir şehir merkezini ve buranın cadde ve sokaklarını düşünelim. Bu cadde ve sokakların tümünü görecektir. Bu cadde ve sokakların bitiş noktalarına güvenlik kamerası yerleştirmek isteyelim. Ekonomik açıdan düşünüldüğünde olası en az sayıda kamerayla iş halledilmek istenir. Cadde ve sokakları bir G grafinin ayrıtları, güvenlik kamerası yerleştirilecek noktaları da G grafinin tepeleri olarak düşünersek, problem minimum tepe örtüsü problemine dönüşecektir.

Cadde ve sokakların sayısına yani ayrıtların sayısına n ve maliyeti karşılanabilecek kamera sayısına da k ve $k < n$ dersek, güvenlik kamerası koymak için $\binom{n}{k}$ adet durum vardır. Ancak böyle bir durumda görülmeyen cadde ve sokakların bulunması da olasıdır. Daha iyi bir çözüm için yani daha az kamerayla bu işin yapılabilmesi konusu araştırılmak istendiğinde ise örneğin, 1 adet kamera koymak için $\binom{n}{1}$ adet durum, 2 adet kamera koymak için $\binom{n}{2}$ adet durum vardır. Bu şekilde tüm durumları denemek için ise

$$\binom{n}{1} + \binom{n}{2} + \binom{n}{3} + \dots + \binom{n}{n} = 2^n - 1$$

Adet durum var olup zaman n ye bağlı olarak üstel olarak büyüyecektir.

3.3 Tepe Örtüsü Probleminin Türleri

Tanım 3.3.1. Minimum Ağırlıklı Tepe Örtüsü (Minimum weighted vertex cover)

$G=(V,E)$ bir graf ve $W:V \rightarrow \mathbb{R}$ ağırlık fonksiyonu olsun.

G nin her bir $(u,v) \in E$ ayrıtının en az bir uç noktasını içeren, $u \in A$ ya da $v \in A$ ya da $(u,v) \in A$, en küçük ağırlığa sahip $A \subseteq V$ kümesine, minimum ağırlıklı tepe örtüsü kümesi denir.

Ağırlıklı tepe örtüsü problemi Lineer program modeliyle aşağıdaki gibi ifade edilebilir.

$$\min \sum_{i \in V} w_i x_i$$

$$x_i + x_j \geq 1 \quad \text{her bir } (i, j) \in E \text{ ayrıtı için}$$

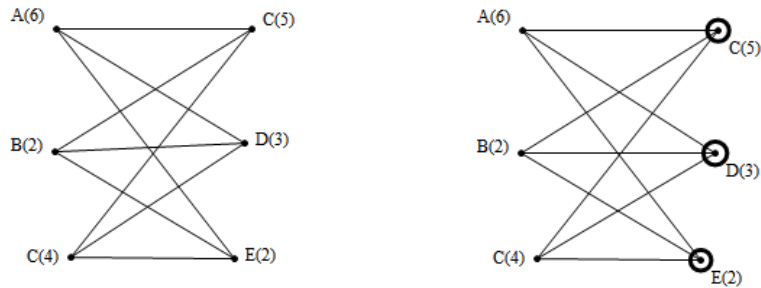
$$x_i \in \{0,1\} \quad \text{her bir } i \in V \text{ tepesi için}$$

$x_i + x_j \geq 1$, her bir $(i, j) \in E$ ayrıtı için, ayrıtın en az bir uç noktası alınması gerektiğini gösterir.

$x_i \in \{0,1\}$, her bir $i \in V$ tepesi için, ilgili tepe, örtü kümesine alındıysa 1 değerini, alınmadıysa 0 değerini alır.

Burada w_i ler ise $i \in V$ için ilgili tepenin ağırlığını temsil etmektedir.

Aşağıdaki graf ağırlıklı tepe örtüsü problemi için bir örnek teşkil etmektedir (Balaji S., Swaminathan V. and Kannan K., 2010).

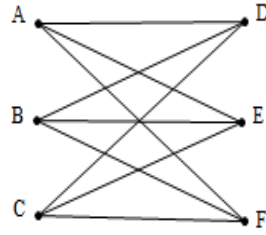


Şekil 3.2 Ağırlıklı tepe örtüsü problemi için verilen bir graf ve bu grafın tepe örtüsü.

Bu örnekte, minimum ağırlıklı tepe örtüsünün içerdği tepeler daire içine alınmıştır.

Örneklenen graf $K_{3,3}$ iki parçalı tam graftır. Burada minimum ağırlıklı örtü kümesi, C(5), D(3), E(2) tepelerinden oluşur.

Aynı graf, ağırlıksız bir graf olarak düşünülüp tepe örtü kümesi bulunacak olursa,



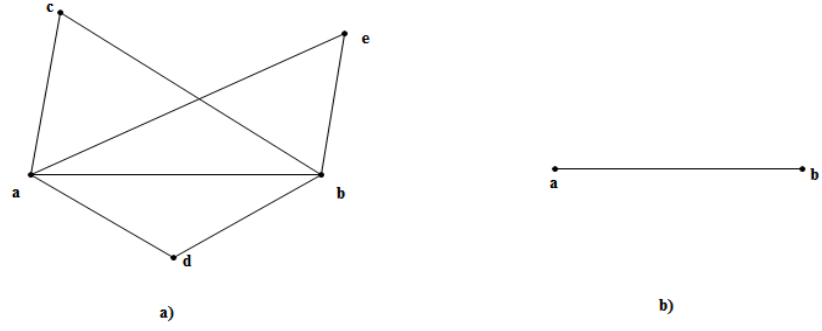
Şekil 3.3 $G=(V,E)$ grafi.

şekilde görüldüğü gibi A, B, C tepeleri ya da D, E, F tepeleri alınırsa tepe örtü kümesi bulunur. Ancak ağırlıklı versiyonunda, böyle olasılıksal bir seçim yapılamamıştır. Ağırlıklı versiyonunda amaç, tepe örtüsüne alınan tepelerin sayısının az olması değil, tepe örtüsüne alınan tepelerin ağırlıklarının toplamının en az olmasıdır.

Tanım 3.3.2. Birleştirilmiş Tepe Örtüsü (Connected vertex cover)

Bir $G=(V,E)$ grafi ve bir $k \geq 0$ tam sayısı verildiğinde, G için en fazla k tepe içeren bir C tepe örtüsünün var olup olmadığını belirlemek amaçlanır. Ancak G nin alt grafi olan C , tepe örtü kümesi birleştirilmiştir. ((Guo J., Niedermeier R., Wernicke S., 2007).

Aşağıdaki grafa bulunan tepe örtü kümesi tarafından oluşturulan alt graf birleştirilmiştir.



Şekil 3.4 Bir $G=(V,E)$ grafi ve onun birleştirilmiş tepe örtüsü

Tepe örtü kümesi $C=\{a,b\}$. Tepe örtü kümesinde bulunan a ve b ayrıtları tarafından oluşan alt grafi birleştirilmiştir.(Bkz Şkl 3.4 b))

Tepe örtüsü probleminin bu türünde en az sayıda tepeyle bir tepe örtüsünün istenmesinin yanı sıra bulunan tepe örtü kümesinin oluşturduğu alt grafin birleştirilmiş olmasının istenmesi problemi çok daha zor yapmaktadır. Bu problem için bulunan her bir örtü kümesinin birleştirilmiş olup olmadığı da test edilmek zorundadır. Bu problem için 2-yaklaşık algoritma vardır.Ancak, n , grafin tepe sayısı, m , grafin ayrıt sayısı ve k de birleştirilmiş tepe örtüsünün boyutu olmak üzere, bu problem için çalışma zamanı $O(6^k n + 4^k n^2 + n^2 \log n + nm)$ olan algoritma geliştirilmiştir. (Guo J., Niedermeier R. and Wernicke S., 2007)

Tanım 3.3.3. Kapasitelendirilmiş Tepe Örtüsü (Capacitated vertex cover)

Bu problem türünde kapasiteli bir $G=(V,E)$ grafi ve bir ağırlık fonksiyonu vardır.

Kapasiteli grafin anlamı, her bir $v \in V$ tepesine ayrıtların toplam sayısı ile sınırlı olan bir $c(v) \geq 1$ kapasite atanmasıdır ki burada tepeler kısmi olarak örtülebilir.

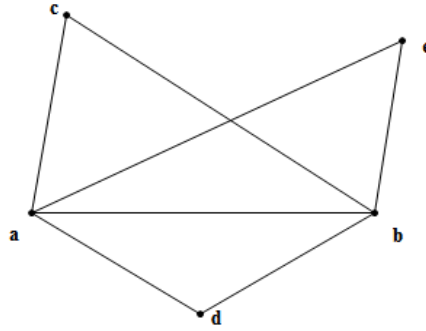
Ağırlıklı ve kapasiteli bir $G=(V,E)$ grafi verilsin. w_v , v tepesinin ağırlığını, k_v ise v tepesinin kapasitesini gösterecek şekilde k_v bir tamsayı olsun .)

Bir kapasitelendirilmiş tepe örtüsü fonksiyonu $x:V \rightarrow \mathbb{N}_0$, öyle ki G nin ayrıtlarının bir yöneltmesi (orientation) vardır ve $v \in V$ tepesine doğru olan ayrıtların sayısı en fazla $k_v w_v$ dir. Örtünün ağırlığı $\sum_{v \in V} k_v w_v$ dir.Burada her bir tepe için $k_v = |V| - 1$ alınırsa, problem minimum ağırlıklı tepe örtüsüne dönüşür.

Bu problem türünde her bir tepenin kapasitesi olduğundan, kendisine bitişik bütün ayrıtları örtemeyebilir. Bu problem de 2-yaklaşık algoritmayla çözülebilir. Ancak enumerative yaklaşım kullanılarak $O(1.2^{k^2+n^2})$, çalışma zamanlı algoritmayla çözümlenebilmiştir. Hard ve soft kapasiteli olmak üzere iki türü vardır.(Guha S., Hassin R., Khuller S., Or E., 2002; Guo J., Niedermeier R. and Wernicke S., 2007)

Tanım 3.3.4. Maksimum Kısmi Tepe Örtüsü (Maximum partial vertex cover)

$G=(V,E)$ yönsüz bir grafi verildiğinde ve $k \geq 0$ ve $t \geq 0$ tam sayıları verildiğinde, en az t ayrıtı örten ve en fazla k boyutunda olan bir tepe alt kümesi $C \subseteq V$, C nin var olup olmadığını araştırılır.



Şekil 3.5 Basit bir $G=(V,E)$ grafi

Şekilde verilen graf için en az 3 ayrıtı örten ve en fazla 2 boyutlu bir tepe örtüsü bulmak isteyelim. $C=\{a,b\}$ kümesi bu koşulu sağlamaktadır. Ancak, daha karmaşık durumlarda, ayrıt sayısının ve boyut sayısının sınırlandırılması, problemi zorlaştırmaktadır. (Guo J., Niedermeier R. and Wernicke S., 2007)

3.4 Tepe Örtüsü Problemi İçin Çözüm Yöntemleri ve Algoritmalar

Tepe örtüsü gibi polinom zamanda çözilemeyen problemler NP–Tam sınıfındadır. (Garey M. and Johnson D., 1979)

Bir problemin NP-tam olması çerçevesinde üç tür yaklaşım vardır.

1. Eğer problemin çözümü için önerilen algoritma küçük girdi sınıfları üzerinde çalışıyorsa, algoritmanın çalışma zamanı üstel olsa bile bu durum kabul edilebilirdir
2. Bazı özel durumları probleminden çıkartabilirsek polinom zamanda çözülebilir.
3. Optimal sonucu bulmaya çalışmak yerine optimale yakın sonuçlar bulmak da tatmin edici olabilir. (Cormen, Leiserson, Rivest and Stein, 2001)

Tepe örtüsü probleminin karar versiyonunda da benzer şekilde tam bir sonuç bulunamamaktadır. Bu versiyonda, ancak problem küçük k sabit parametreleri için polinom zamanda çözülebilir.

Karar problemlerini, sabit parametrenin küçük değerleri için etkili olarak çözen algoritmalara fixed-parameter-tractable algoritma denmektedir. Bu yöntem parametrelendirilmiş karmaşıklık teorisi kapsamındadır. İlk sistematik çalışmayı Downey ve Fellows 1999 da yapmıştır.

Bir G grafi, pozitif bir tamsayı k parametresi, n , G grafinin tepe sayısı, m de G grafinin ayrıt sayısı olmak üzere bu yaklaşımla tepe örtü problemi $O(2^k (n+m))$ zamanda çözülebilir.

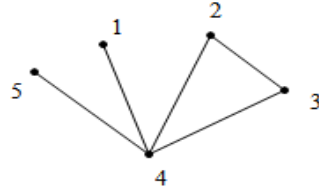
Tepe örtüsü problemi için sınırlı bir ağaç arama algoritması tekniği vardır ki rekürsif olarak çalışır. Rekürsif olarak bir tepe seçer ve her bir adımda iki duruma dallanmaktadır, ya bu tepenin olduğu örtü kümesine ya da tüm komşularının olduğu örtü kümesine dallanır. (Chen et al., 2006)

Diğer bir yöntem ise optimale yakın sonuçlar bulmayı amaçlayan yaklaşık çözüm yöntemleridir.

Bir problem için tasarlanan bir algoritma $\rho(n)$ yaklaşık oranına sahiptir eğer, n boyutlu herhangi bir girdi için algoritma tarafından üretilen çözümün maliyeti C , optimal çözüm olan C^* ile $\rho(n)$ nin çarpımıdır.

$$\text{Yani } \max\left(\frac{C}{C^*}, \frac{C^*}{C}\right) \leq \rho(n)$$

Bu algoritmalar $\rho(n)$ yaklaşık algoritma olarak adlandırılmaktadır (Cormen, Leiserson, Rivest and Stein, 2001).



Şekil 3.6 Bir G grafi

Yukarıdaki grafin optimal tepe örtü kümesi $C^* = \{4, 2\}$ ya da $C^* = \{4, 3\}$ olup $|C^*| = 2$ dir.

Herhangi bir algoritmanın bulduğu, rastgele bir tepe örtü kümesi $C = \{1, 2, 3, 5\}$ olsun. Bu kümenin boyutu ise $|C| = 4$ tür.

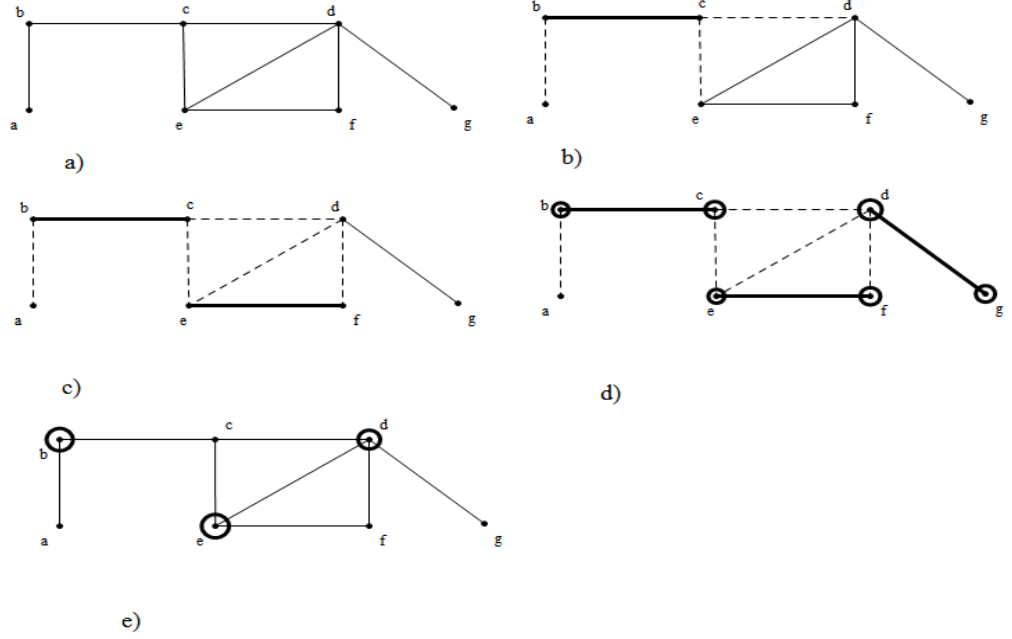
$\max\left(\frac{C}{C^*}, \frac{C^*}{C}\right) = \left(\frac{4}{2}, \frac{2}{4}\right) = 2 \leq \rho(n)$.Buna göre yaklaşık oran $\rho(n) = 2$, yani algoritma optimalden en fazla iki katı büyüklüğünde sonuç verebilir denir.

3.4.3 Yaklaşık tepe örtüsü algoritması

- 1 $C \leftarrow \emptyset$
- 2 $E' \leftarrow E(G)$
- 3 **while** $E' \neq \emptyset$
- 4 E' den keyfi bir (u, v) ayrıtı al
- 5 $C \leftarrow C \cup (u, v)$
- 6 E' de, u veya v ye bitişik olan her bir ayrıtı sil
- 7 **return** C

(Monien B. and Speckenmeyer E., 1985).

Algoritmanın uygulanışı.



Şekil 3.7 Yaklaşık Tepe örtüsü algoritmasının uygulanışı.

Herhangi bir ayrıt olarak (b, c) ayrıtı alınır. (b, c) ayrıtını oluşturan b ve c tepelerine bitişik olan ayrıtlar graftan silinir. $C = \{b, c\}$. Bütün ayrıtlar silinmemiş olduğundan devam edilir.

Yine rastgele bir ayrıt alınır, (e, f) ayrıtını alalım.(e,f) ayrıtını oluşturan e ve f tepelerine bitişik olan ayrıtlar graftan silinir. $C=\{b, c, e, f\}$. Bütün ayrıtlar silinmemiş olduğundan devam edilir.

Yine rastgele bir ayrıt alınır, (d, g) ayrıtını alalım.(d, g) ayrıtını oluşturan d ve g tepelerine bitişik olan ayrıtlar graftan silinir. $C=\{b, c, e, f, d, g\}$.Alınacak başka ayrıt kalmadığından algoritma durur.

Bulunan tepe örtü kümesi $C=\{b, c, e, f, d, g\}$. Ancak bu örtü kümesi optimal yani en az sayıda tepeye sahip olan küme değildir. Optimal, tepe örtü kümesi ise $C^*=\{b, e, d\}$ dir.

Algoritmanın bulduğu örtü kümesinin eleman sayısı 5 iken, optimal tepe örtü kümesinin eleman sayısı 3 tür.

Bu algoritmanın karmaşıklık analizini yapacak olursak,

Algoritmada 3. deki döngüden, (3-6) satırları üzerinde tekrarlı olarak E' den keyfi bir (u,v) ayrıtı alınır, onun bitiş noktaları u ve v , C ye eklenir ve u ve v den biri tarafından örtülen tüm ayrıtlar E' den silinir. Bu algoritmanın çalışma zamanı $O(E)$ dir.

Teorem 3.4.3.1. Tepe Örtüsü Yaklaşık Algoritması 2-yaklaşık bir algoritmadır. (Cormen, Leiserson, Rivest and Stein, 2001).

İspat: c kümesi yaklaşık tepe örtüsü algoritmasının ürettiği küme, C^* kümesi de optimal tepe örtü algoritmasının ürettiği küme olsun. Ayrıca A , satır 4 'te şekil 3 de seçilen ayrıtların kümesi olsun. Çünkü, bir ayrıt alındığında her iki tepe de alınış olur ve $C=2|A|$ elde edilir. Ama optimal tepe örtüsü algoritmasında bu ayrıtın uç noktalarından biri eklenmiş olurdu. $\Rightarrow \frac{C}{C^*} \leq \rho(n)=2$. Biçimsel olarak, (Böylece bir ayrıt satır 4 te bir kez alınır ve onun bitiş noktalarına bitişik tüm ayrıtlar satır 6 da E' den silinir.) C^* dan olan, A da aynı ayrıtı örten iki tepe yoktur. Ve böylece sahip olunan alt sınır $C^* \geq |A| \dots (1)$

Şekil 1 de satır 4 te, aldığımız her iki bitiş noktası Tepe Örtüsünün boyutunun üzerinde bir üst sınıra ulaşır. $|C|=2|A|$ olduğundan bu durumda üst sınır tamdır.

$$|C|=2|A| \dots (2)$$

İlk eşitlikte $|C|/2=|A|$ alınır.

$$|C^*| \geq |C|/2$$

$$|C^*|/|C| \geq 1/2$$

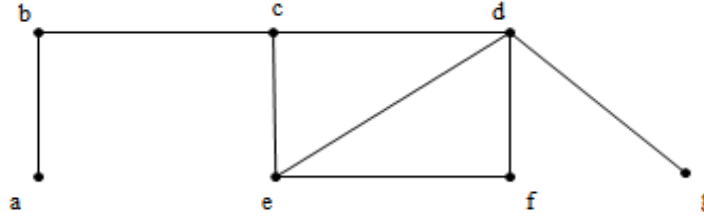
$$\frac{|C|}{|C^*|} \leq 2 = \rho(n)$$

3.4.4 Greedy (Açgözlü yaklaşım) yöntemi

Optimizasyon problemlerinin çözümünde kullanılan bir tekniktir. Aç gözlü yaklaşım tekniği her bir adımda çalışır, bir girdiyi bir kere göz önünde tutar (özellikle zeki olanı). Her bir adımda, bu adımdaki en iyisi olup olmadığına bağlı olarak bir karar verir. Her adımda yerel olarak bulunan optimal çözümün genel bir çözüm olabilme ihtimali vardır.

Zeki Greedy Algoritması

- 1 $C \leftarrow \emptyset$
- 3 **while** $E \neq \emptyset$
- 4 İlgili graftan maksimum dereceli bir $v \in V$ tepesi al
- 5 $C \leftarrow C \cup \{v\}$
- 6 $E \leftarrow E - \{e \in E : v \in e\}$
- 7 **return** C .

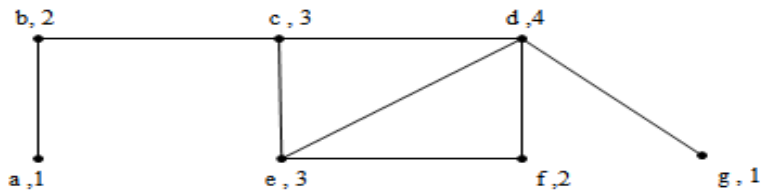


Şekil 3.8 Bir $G=(V,E)$ grafi.

Yukarıdaki şekilde verilen bir $G=(V,E)$ grafinin tepe örtüsünü aç gözlü yaklaşım algoritmasıyla bulunması :

G_C = Greedy (Aç gözlü yaklaşım) algoritmasının bulduğu tepe örtüsü kümesi.

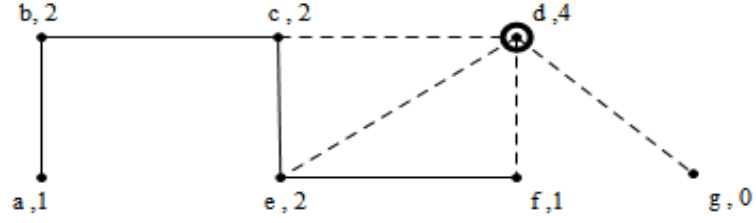
C^* = Optimal tepe örtüsü kümesi



Şekil 3.9 $G=(V,E)$ grafinin her bir tepesinin dereceleri ile gösterimi.

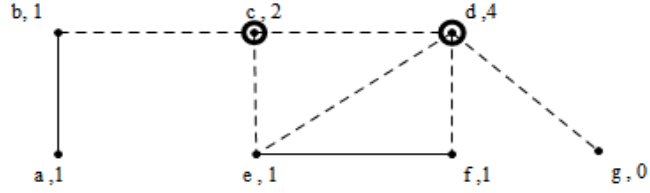
1.Adım: Öncelikle her bir tepenin derecesi bulunur. Derecesi en büyük olan tepe d tepesidir, derecesi 4 tür. d tepesi alınırsa,

d tepesi ve ona bitişik olan ayrıtlar graftan silinir. $G_C = \{d\}$.



Şekil 3.10 Algoritmanın grafa uygulandığı 1.adım.

2.Adım: Derecesi en büyük olan birden fazla tepe vardır. Bunlar b, c, e tepeleri olup her birinin derecesi 2 dir. Herhangi biri alınabilir, c tepesini alalım.

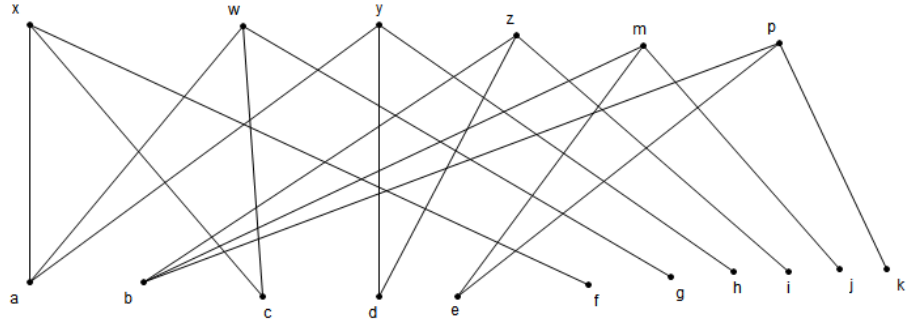


Şekil 3.11 Algoritmanın grafa uygulandığı 2.adım.

c tepesi örtü kümesine alınır ve c tepesi ve ona bitişik olan ayrıtlar graftan silinir. $G_C = \{d, c\}$.

Bu durum her zaman geçerliliğini korumamaktadır, 2. Adımda tepe dereceleri eşit olan b, c, e den herhangi birinin alınabileceği söylenmiş ve c alınmıştır. Bunun yerine b ya da e tepesi alınsaydı, aç gözlü yaklaşım algoritması optimal çözümlerden birine ulaşacaktı.

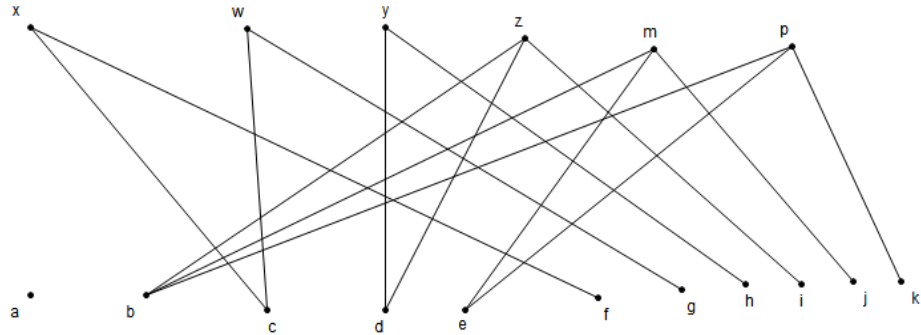
Aç gözlü yaklaşım algoritması optimal çözüme ulaşabilir, ancak bunu garanti etmemektedir. Aşağıdaki grafta aç gözlü yaklaşım algoritması optimalden çok fazla uzaklaşmaktadır.



Şekil 3.15 Bir $G=(V,E)$ grafi

1.Adım: Derecesi en büyük olan x, w, y, z, m, p, a, b tepeleri vardır. Alt satırda yer alan, derecesi en büyük olan a veya b tepelerinden biri örtü kümesine eklenir. a, alınırsa, a tepesi ve ona bitişik olan ayrıtlar graftan silinir.

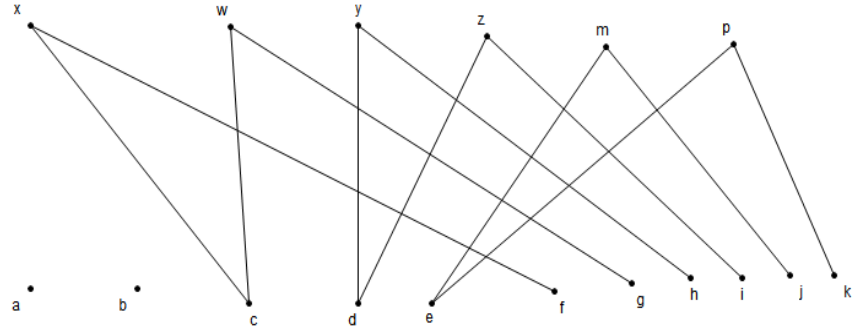
$G_C = \{a\}$.



Şekil 3.16 Algoritmanın grafa uygulanışı 1.adım.

2.Adım: z, m, p ve b tepelerinin derecesi 3 olup en büyüktür. Yine bu tepelerden herhangi biri örtü kümesine eklenebilir. Alt satırdan b tepesinin eklendiğini varsayarsak, b tepesi ve ona bitişik olan ayrıtlar graftan silinir.

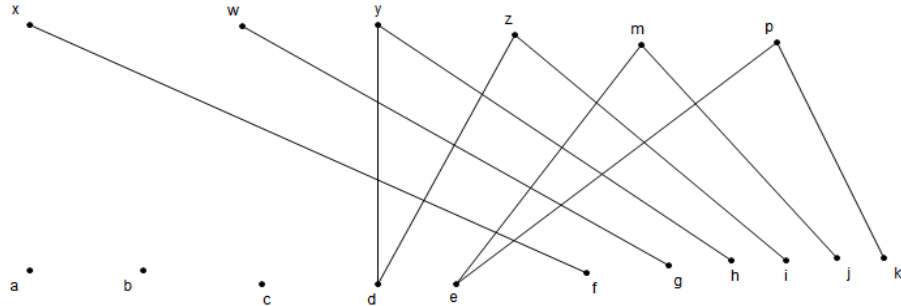
$$G_C = \{a, b\}.$$



Şekil 3.17 Algoritmanın grafa uygulandığı 2.adım.

3.Adım: Bu grafta yine en büyük dereceli tepelerden x, w, y, z, m, p, c, d, e tepelerinden biri alınır. Algoritmanın alt satırdan c tepesini aldığını varsayalım .c tepesi ve ona bitişik olan ayrıtlar graftan silinir.

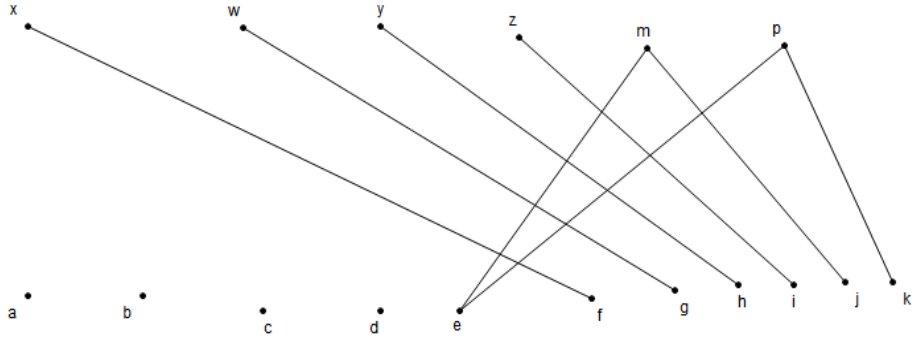
$$G_C = \{a, b, c\}.$$



Şekil 3.18 Algoritmanın grafa uygulandığı 3.adım.

4.Adım: En büyük dereceli tepelerden, y, z, m, p, d, e den biri örtü kümesine alınır. Algoritmanın alt satırdan d tepesini aldığı varsayarsak, d tepesi ve ona bitişik olan ayrıtlar graftan silinir.

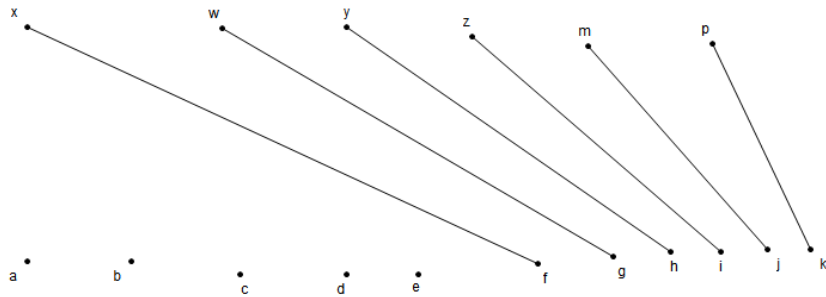
$G_C = \{a, b, c, d\}$.



Şekil 3.19 Algoritmanın grafa uygulandığı 4.adım.

5.Adım: En büyük dereceli tepelerden m, p veya e örtüye alınabilir. Algoritmanın alt satırdaki e tepesini örtüye aldığı varsayarsak, e tepesi ve ona bitişik olan ayrıtlar graftan silinir.

$G_C = \{a, b, c, d, e\}$.

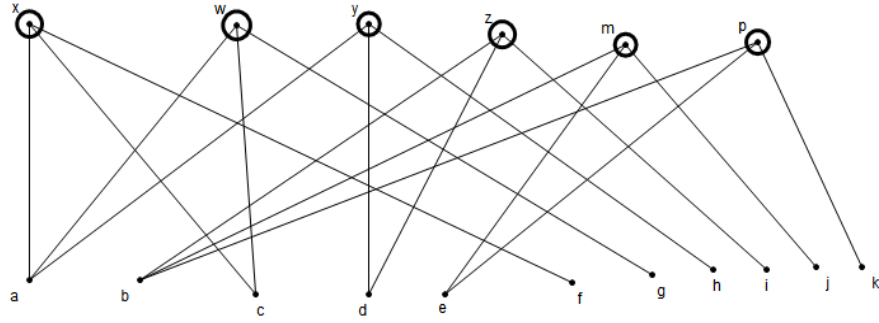


Şekil 3.20 Algoritmanın grafa uygulandığı 5.adım.

6.Adım: Yukarıdaki graftan derecesi en büyük $x, w, y, z, m, p, f, g, h, i, j, k$ tepelerinden herhangi biri örtüye alınabilir. Algoritmanın çalışması sonucunda f, g, h, i, j, k tepelerinin, örtü kümesine alındığı kabul edilirse, greedy algoritmasının bulduğu örtü kümesi $G_C = \{a, b, c, d, e, f, g, h, i, j, k\}$.

Ancak optimal çözüm, x, w, y, z, m, p tepelerinin alınmasıyla oluşmaktadır.

$$C^* = \{x, w, y, z, m, p\}.$$



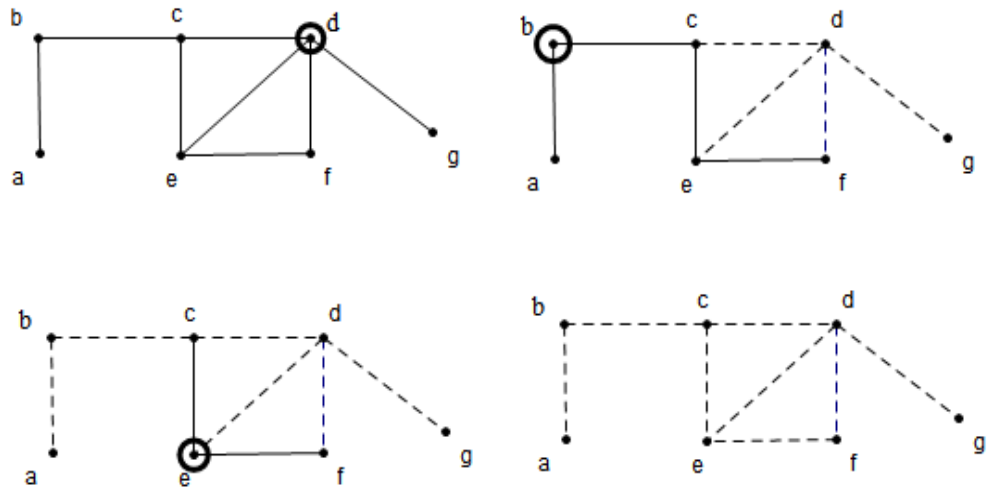
Şekil 3.21 Optimal tepe örtüsü.

Aç gözlü yaklaşım algoritması böyle bir grafta optimalden oldukça uzaklaşmış olup $|C^*| = 6$ iken $|G_C| = 11$ dir.

3.4.5 Alom' un algoritması

Bu algoritma yaklaşık tepe örtüsü algoritması ve aç gözlü yaklaşım algoritmasına göre daha iyi sonuç vermektedir. Algoritma, maksimum sayıda ayrıta bitişik olan bir tepe seçer. Bu tepeye bitişik olan tüm ayrıtlar atılır. Aynı maksimum sayıda ayrıta sahip birden fazla tepe varsa, algoritma diğer tepeler tarafından örtülmeyen, maksimum ayrıtlı en az bir ayrıta sahip olan tepe seçer. Bu işlem graftaki tüm tepeler örtülene kadar devam eder. (Alom, B.M. and Kumar K. V. R., 2009)

- 1 OPTIMAL_VT_COVER (E,V) { // E ayrıt kümesi, V tepe kümesi }
- 2 $V' \leftarrow \emptyset$
- 3 $E' \leftarrow E(G)$
- 4 **while** ($E' \neq \emptyset$) {
- 5 $M \leftarrow$ maksimum sayıda ayrıta bitişik tepeyi al
- 6 **If** (maksimum sayıda ayrıta sahip birden fazla tepe varsa) **then**
- 7 $M \leftarrow$ maksimum sayıda ayrıta sahip olan ve diğerleri tarafından örtülmeyen en az bir ayrıta sahip olan tepeyi seç
- 8 $V' \leftarrow V' \cup M$
- 9 M deki tepeye bitişik olan tüm ayrıtları sil.
- 10 Yeni graftaki bitişik ayrıtları say }
- 11 **return** V' }.



Şekil 3.22 Alom 'un Algoritmasının uygulanışı.

A_C : Alom algoritmasının bulduğu örtü kümesi

C^* : Optimal örtü kümesi

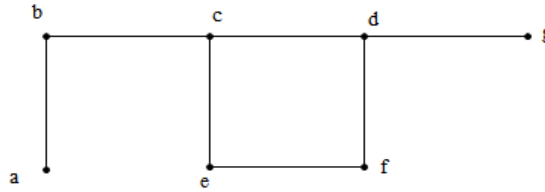
Algoritma ilk olarak en çok ayrıta bitişik olan d tepesini alır, d tepesinin bitişik olduğu ayrıt sayısı 4 tür. 4 adet ayrıta sahip başka tepe yoktur. d, tepesi $A_C=\{d\}$ örtü kümesine alınır. d tepesine bitişik olan ayrıtlar silinir.

Şimdiki iterasyonda b, e ve c tepelerinin hepsi 2 ayrıta sahiptir, ancak b ve e tepelerinin c tarafından örtülmeyen 1 ayrıtı, c tepesinin ise b ve e tarafından örtülmeyen hiç ayrıtı yoktur. Algoritma diğerleri tarafından örtülmeyen maksimum sayıda ayrıta sahip b ya da e yi seçer. Burada b seçilirse, b ye bitişik tüm ayrıtları silersek sadece c, e ve f kalır. $A_C=\{d, b\}$

Son aşamada ise maksimum sayıda ayrıta sahip olan e, örtü kümesine alınır ve e ye bitişik ayrıtlar silinir. $A_C=\{d, b, e\}$.

Graftan silinebilecek başka ayrıt kalmadığından algoritma sonlanır.

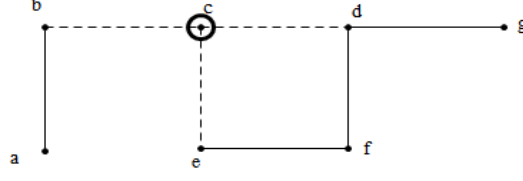
Aşağıdaki grafta ise Alom 'un algoritmasının optimal çözümü garanti etmediği örneklenmiştir.



Şekil 3.23 Basit bir $G=(V,E)$ grafi.

1.Adım: c ve d tepelerinin ikisi de maksimum dereceli ve ikisi de birbiri tarafından örtülmeyen 1 ayrıta sahiptir.Örtü kümesine rastgele c tepesi alınır, c tepesi ve ona bitişik ayrıtlar graftan silinir

$$A_C=\{c\}.$$

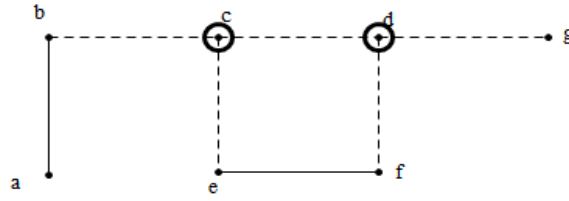


Şekil 3.24 Algoritmanın uygulanışı 1. Adım.

2. Adım: d ve f tepelerinin ikisi de maksimum derecelidir ve birbirini tarafından örtülmeyen 1 ayrıta sahiptir.

İkisinden biri rastgele alınabilir. d alınırsa, d tepesi ve ona bitişik olan ayrıtlar graftan silinir.

$$A_C = \{c, d\}.$$

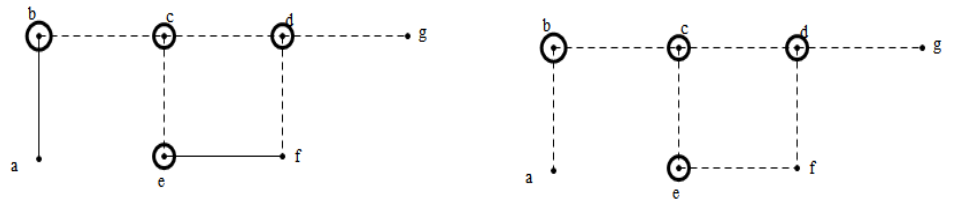


Şekil 3.25 Algoritmanın uygulanışı 2. Adım.

3. Adım: b, a, e, f tepelerinden herhangi biri örtü kümesine alınır. Öncelikle e ve sonrasında b alınırsa,

$$A_C = \{c, d, e, f\}. \text{ Olur.}$$

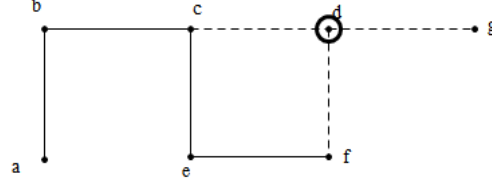
$$|A_C| = 4 \text{ tür.}$$



Şekil 3.26 Algoritmanın uygulanışı 3. Adım.

1.Adım: Yine aynı grafta ilk adımda rastgele d tepesi alınsaydı,

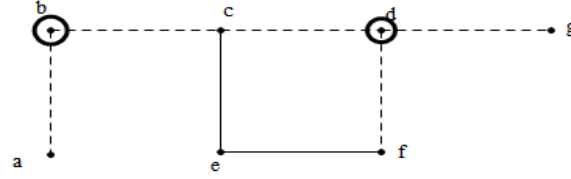
$$A_C = \{d\}.$$



Şekil 3.27 Algoritmanın rastgele d tepesi alınması durumunda işlevişi 1.Adım.

2.Adım: b,c,e tepelerinin derecesi maksimumdur ancak b ve e tepelerinin diğerleri tarafından örtülmeyen en az bir ayrıtı vardır. b tepesi rastgele alınırsa,

$$A_C = \{d, b\}.$$

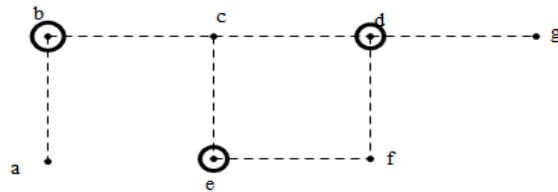


Şekil 3.28 Algoritmanın rastgele d tepesi alınması durumunda işlevişi 2.Adım.

3.Adım: e tepesi maksimum dereceye sahiptir. Örtü kümesine alınır. Algoritma örtülmemiş tepe kalmadığından sonlanır.

$$A_C = \{d, b, e\}.$$

$$|A_C| = 3.$$



Şekil 3.29 Algoritmanın rastgele d tepesi alınması durumunda işlevişi 3.Adım.

İlk iterasyonda algoritma c tepesini alarak başladığında 4 tepeli bir örtü kümesi bulmuşken, ikinci iterasyonda d tepesini alarak başladığında 3 tepeli bir örtü kümesi bulmuştur. Görüldüğü gibi Alom 'un algoritması da optimumu garanti etmemektedir.

3.5 Tepe Örtüsü Problemi İçin Bulunan Yaklaşık Sonuçlar

Çizelge 3.2 Tepe örtüsü için bulunan yaklaşık oranlar.

Referanslar	Yaklaşık oran
Gavril, F	2
Hochbaum, D.S,1982	2
Bar Yehuda R. and S.Even, 1982	2
Hochbaum, D.S,1983	$2 - \frac{2}{\Delta}$
Bar Yehuda R. and S.Even, 1985	$2 - \frac{2}{\sqrt{V}}$
Bar Yehuda R. and S.Even, 1985	$2 - \frac{\log \log V}{2 \log V}$
Düzlemsel graflar için (planar)	
Referanslar	Yaklaşık Oran
Hochbaum, D.S,1983	1.6
Hochbaum, D.S,1983	1.5
Bar Yehuda R. and S.Even, 1985	1.5
Bar Yehuda R. and S.Even, 1982	$1 + \frac{2}{3}$
Chiba S., T.Nishizeki and N.Saito, 1981	$1 + \varepsilon$

(Bar-Yehuda R., Flysher G. and Mestre J., 2007)

4. TEPE ÖRTÜSÜ VE GRAF İŞLEMLERİ

Graf işlemleri, iki temel gruba ayrılır. Bunlar tekli (unary) ve ikili (binary) graf işlemleridir. Bu işlemlerden bazıları aşağıda listelenmiştir.

Tekli İşlemler : Ayrıt çıkarma, ayrıt büzme, tümleyen ve kuvvet işlemleri

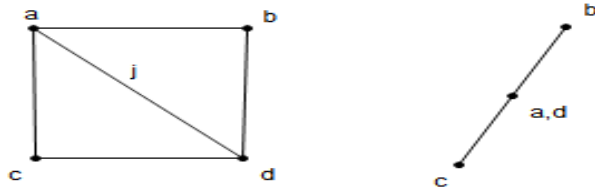
İkili İşlemler : Toplama, kartezyen çarpım, lexicographic çarpım, güçlü çarpım, tensör çarpım (ayrıca doğrudan, kategorik, kardinal ve kronecker çarpım olarak da adlandırılır.)

4.1 Ayrıt Çıkarma

Tanım 4.1.1. (Chatrand and Lesniak, 1996) Bir G grafında $G-e$ işlemi, herhangi bir e ayrıtının silinmesidir.

4.2 Ayrıt Büzme

Tanım 4.2.1. (Chatrand and Lesniak, 1996) Bir G grafına uygulanan ayrıt büzme işleminde, birbirine komşu olan iki tepe silinir ve bunların yerine başka bir tepe eklenir. Sonuç olarak daha önce silinen tepelere komşu olan tepeler, silinen tepelerin yerine eklenen tepeye komşu olur. Oluşan yeni graf ise $G.e$ ile gösterilir.

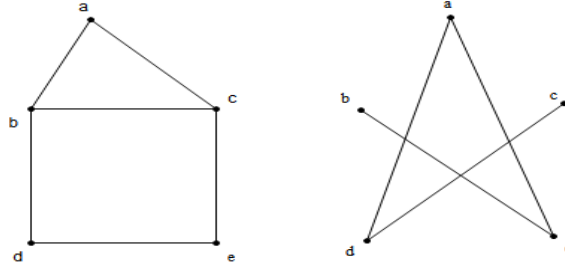


Şekil 4.1 Bir $G=(V,E)$ grafına ayrıt büzme işleminin uygulanması.

Yukarıda herhangi bir G grafi verilmiş olup j ayrıtı büzülmüştür.

4.3 Grafın Tümleyeni

Tanım 4.3.1. (Chatrand and Lesniak, 1996) Bir G grafindaki tepeler ile grafa olmayan ayrıtların oluşturduğu grafa, grafın tümleyeni denir. Bir G grafının tümleyeni $\bar{G}$ ile gösterilir.

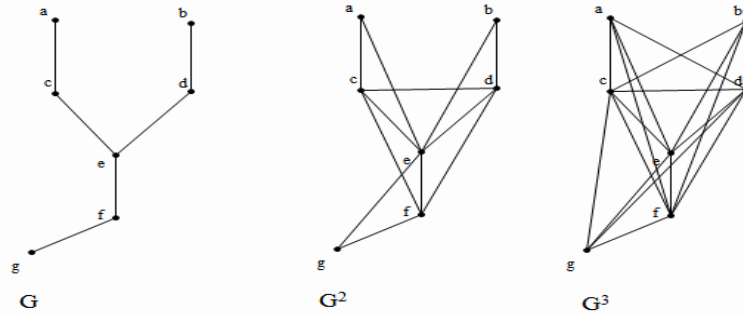


Şekil 4.2 Bir $G=(V,E)$ grafi ve onun tümleyeni.

4.4 Grafın Kuvveti

Tanım 4.4.1. (Chatrand and Lesniak, 1996) G , v tepeli ve e ayrıtlı birleştirilmiş bir graf olsun. G nin n .ci gücü G^n ile gösterilir. $V(G^n) = V(G)$ dir. $E(G^n)$ ise şöyle belirlenir. G nin herhangi iki tepesi arasındaki en kısa yolun ayrıt sayısı en çok n ise G^n de bu iki tepe doğrudan bir ayrıt ile birleştirilir.

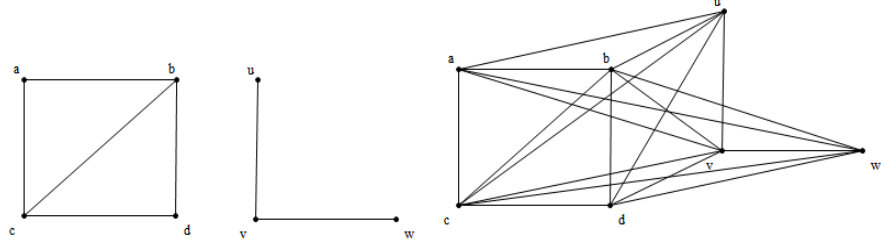
Aşağıdaki şekilde grafın kuvveti işlemi örneklenmiştir.



Şekil 4.3 Bir $G(V,E)$ grafına kuvvet işleminin uygulanışı.

4.5 Toplama İşlemi

Tanım 4.5.1. (Chatrand and Lesniak, 1996) G_1 ve G_2 iki graf olmak üzere, bu grafların toplamı $G_1 + G_2$, G_1 in her tepesi ile G_2 nin her bir tepesinin bir ayrıtla birleşmesi sonucu oluşur.



Şekil 4.4 G_1 ve G_2 iki graf olmak üzere , bu grafların toplamı $G_1 + G_2$.

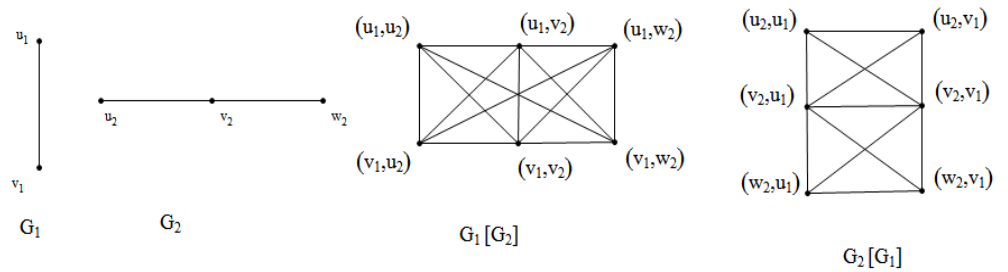
4.6 Lexicographic Çarpım

Tanım 4.6.1. (Chatrand and Lesniak, 1996) G_1 ve G_2 iki graf olmak üzere bu iki grafın lexicographic çarpımı $G_1[G_2]$ ile gösterilir. $G_1[G_2]$ nin tepe kümesi

$V(G_1[G_2]) = V(G_1) \times V(G_2)$ dir.

$E(G_1[G_2])$ ayrıtlar kümesi ise şu şekilde tanımlanır :

$u = (u_1, u_2)$ ve $v = (v_1, v_2)$ tepeleri komşudur ancak, ya u_1, v_1 ile komşudur veya $u_1 = v_1$ ve u_2 ile v_2 komşudur.



Şekil 4.5 G_1 ve G_2 iki graf olmak üzere , bu grafların Lexicographic çarpımı $G_1[G_2]$ ve $G_2[G_1]$.

Lexicographic çarpım ilk olarak Felix Hausdorff (1914) tarafından çalışıldı. Lexicographic çarpım genelde değişmeli değildir. Geller ve Stahl 1975 yılında bağımsızlık sayısına ilişkin aşağıdaki eşitliği vermiştir.

$\alpha(G)$ bir G grafinin bağımsızlık sayısını gösterebilir.

$$\alpha(G[H]) = \alpha(G) \cdot \alpha(H)$$

Tepe örtü sayısı için ise, n , G nin tepe sayısı, m , H nin tepe sayısı olmak üzere,

$$\beta(G[H]) = n \cdot \beta(H) + m \cdot \beta(G) - \beta(G) \cdot \beta(H)$$

sonucuna ulaşılmıştır. (Geller D., Stahl S., 1975)

4.7 Güçlü Çarpım

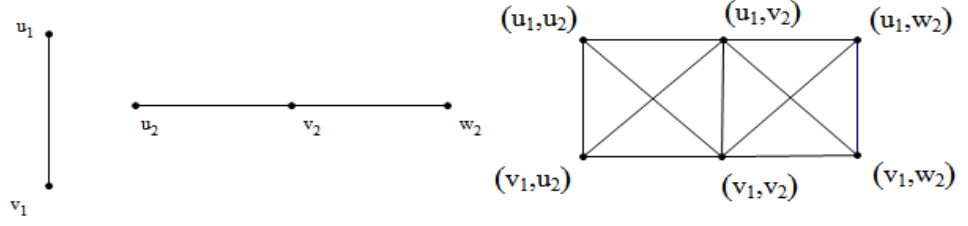
Tanım 4.7.1. (Chatrand and Lesniak, 1996) G_1 ve G_2 iki graf olmak üzere, bu iki grafin güçlü çarpımı $G_1 \boxtimes G_2$ ile gösterilir. $G_1 \boxtimes G_2$ nin tepe kümesi $V(G_1 \boxtimes G_2) = V(G_1) \times V(G_2)$ dir. Ayrıca normal çarpım veya AND çarpım olarak da adlandırılır.

$E(G_1 \boxtimes G_2)$ ayrıtlar kümesi ise şu şekilde tanımlanır :

$u = (u_1, u_2)$ ve $v = (v_1, v_2)$ tepeleri komşudur ancak, aşağıdaki koşullar sağlanmalıdır.

1. $(u_1, v_1) \in E(G_1)$ ve $u_2 = v_2$
2. $(u_2, v_2) \in E(G_2)$ ve $u_1 = v_1$
3. $(u_1, v_1) \in E(G_1)$ ve $(u_2, v_2) \in E(G_2)$

Şekil 4.6 da iki grafin güçlü çarpımı görülmektedir.



Şekil 4.6 G_1 ve G_2 iki graf olmak üzere , bu grafların güçlü çarpımı $G_1 \boxtimes G_2$.

Güçlü çarpıma ilişkin bazı özellikler aşağıda gösterilmiştir.

Önerme 4.7.1.

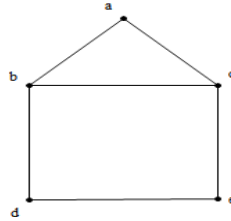
$H = G_1 \boxtimes G_2 = (V(H), E(H))$ o zaman,

- i) $|V(H)| = |V(G_1)| \cdot |V(G_2)|$
- ii) $|E(H)| = 2|E(G_1)| \cdot |E(G_2)| + |V(G_1)| \cdot |E(G_2)| + |V(G_2)| \cdot |E(G_1)|$

- iii) Her $(u, v) \in V(H)$ için $d_H((u, v)) = d_{G_1}(u) \cdot d_{G_2}(v) + d_{G_1}(u) + d_{G_2}(v)$

Burada $d_H((u, v))$, H grafindaki herhangi bir u, v tepesi arasındaki uzaklığı göstermekte olup, $u-v$ arasındaki en kısa yolun uzunluğunu göstermektedir. Örneğin, aşağıdaki $G = (V, E)$ grafında,

$d_G((a, e)) = 2$ dir.



Şekil 4.7 Bir $G=(V, E)$ grafı.

Lemma 4.7.1. (Barnes,. Mackey,1978 ; West, 2001) G , n .ci dereceden basit bir graf olsun . O zaman $\alpha(G)$, G grafının bağımsızlık sayısı ve $\beta(G)$, G grafının tepe örtü sayısı olmak üzere,

$$\alpha(G)+\beta(G)=n \text{ dir.}$$

Teorem 4.7.1: (Baitiang and Sitthiwiratham, 2012) G , m .ci dereceden (m tepeli) birleştirilmiş bir graf olsun .O zaman $\beta(K_n \boxtimes G)=mn-\alpha(G)$.

İspat: $\alpha(K_n \boxtimes G)+ \beta(K_n \boxtimes G)=nm$.

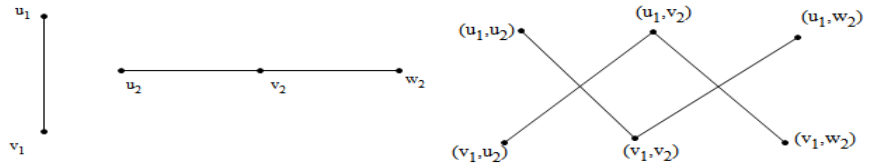
$$\alpha(G)+\beta(K_n \boxtimes G)=nm.$$

$$\beta(K_n \boxtimes G)=nm-\alpha(G).$$

4.8 Tensör Çarpım

Tanım 4.8.1. (Chatrand and Lesniak, 1996) G_1 ve G_2 iki graf olmak üzere , bu iki grafın tensör çarpımı $G_1 \otimes G_2$ ile gösterilir. $G_1 \otimes G_2$ nin tepe kümesi $V(G_1 \otimes G_2)=V(G_1) \times V(G_2)$ dir.Tensör çarpım ayrıca direkt, kategorik, kardinal, ilişkisel, Kronecker, zayıf direkt çarpım veya bağlaç çarpım olarak da adlandırılır.

Eğer, $(u_1, v_1) \in E(G_1)$ ve $(u_2, v_2) \in E(G_2)$ olmak üzere, $u=(u_1, u_2)$ ve $v=(v_1, v_2)$ tepeleri $E(G_1 \otimes G_2)$ deki bir ayrıtla birleştirilir ancak ve ancak u_1 ile v_1 ve u_2 ile v_2 birbirleriyle komşudur.



Şekil 4.8 Sırasıyla, G_1 ve G_2 iki graf olmak üzere , bu grafların tensör çarpımı $G_1 \otimes G_2$

Daha önce belirtildiği gibi $\alpha(G)$, G grafinin bağımsızlık sayısı olmak üzere,

$$\underline{\alpha}(G \otimes H) = \max \{ \alpha(G) \cdot |H|, \alpha(H) \cdot |G| \}.$$

Ayrıca G veya H graflarından birisi yol veya devre ise,

$$\alpha(G \otimes H) = \underline{\alpha}(G \otimes H).$$

Üstelik bu eşitlik, G iki parçalı bir graf ve mükemmel eşleme olduğunda ve H izlenebilir(Traceable) graf olduğunda da sağlanır.

Ancak en az bir ayrıtlı herhangi bir G grafı için ve herhangi bir $i \in \mathbb{N}$ için bir H grafı vardır ki,

$$\alpha(G \otimes H_i) > \alpha(G \otimes H_i) + i.$$

Tam grafların kronecker çarpımının bağımsızlık sayısı ve tepe örtme sayısı ile ilgili aşağıdakiler ispatlanmıştır (Sitthiwirattam T.,2012). G, p.dereceden basit bir graf,

$$\alpha(K_{m,n} \otimes G) = \max \{ (m+n)\alpha(G), p \max \{m,n\} \}$$

$$\beta(K_{m,n} \otimes G) = \min \{ (m+n)\beta(G), p \min \{m,n\} \}.$$

4.9 Kartezyen Çarpım

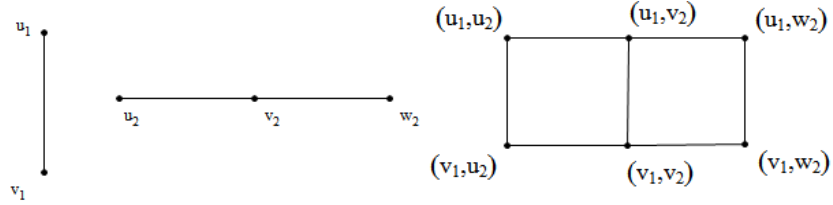
Tanım 4.9.1. (Chatrand and Lesniak, 1996; Harary, 1994) G_1 ve G_2 iki graf olmak üzere bu iki grafın kartezyen çarpımı $G_1 \times G_2$ ile gösterilir.

$G_1 \times G_2$ grafının tepe kümesi

$V(G_1 \times G_2) = V(G_1) \times V(G_2)$ dir.

$E(G_1 \times G_2)$ ayrıtlar kümesi ise şu şekilde tanımlanır :

$u = (u_1, u_2)$ ve $v = (v_1, v_2)$ tepeleri komşudur ancak, ya $u_1 = v_1$ ve u_2 ile v_2 komşudur ya da $u_2 = v_2$ ve u_1 ile v_1 komşudur. (Harary, 1994).



Şekil 4.9 G_1 ve G_2 iki graf olmak üzere , bu grafların kartezyen çarpımı $G_1 \times G_2$

Kartezyen çarpım, ayrık birleşim üzerine dağılmalıdır.

$$G_1 \times (G_2 + G_3) = G_1 \times G_2 + G_1 \times G_3$$

Sonuç 4.9.1. (Vizing, 1963)

Herhangi bir G ve H grafı için, $\alpha(G \times H) \leq \min \{ \alpha(G) \cdot |V(H)|, \alpha(H) \cdot |V(G)| \}$

Sonuç 4.9.2. (Vizing, 1963)

Herhangi bir G ve H grafı için,

$$\alpha(G \times H) \geq \alpha(G) \cdot \alpha(H) + \min \{ |V(G)| - \alpha(G), |V(H)| - \alpha(H) \}.$$

(Klavzar, 2005).

Teorem 4.9.1. G ve H iki parçalı graf , $V(G) = V_1 + V_2$ ve $V(H) = W_1 + W_2$

Eğer $\alpha(G) = |V_1|$ ve $\alpha(H) = |W_1|$ ise o zaman,

$$\alpha(G \times H) = |V_1| \cdot |W_1| + |V_2| \cdot |W_2|$$

Sonuç 4.9.3. G ve H birleştirilmiş iki parçalı graf olsun.

O zaman, $\alpha(G \times H) = \min \{ \alpha(G) \cdot |V(H)|, \alpha(H) \cdot |V(G)| \}$ ancak ve ancak $|V(G)| = 2 \cdot \alpha(G)$ veya $|V(H)| = 2 \cdot \alpha(H)$.

Sonuç 4.9.4. G, birleştirilmiş iki parçalı graf olsun .Herhangi bir $n \geq 1$ için, $\alpha(G^n) = \alpha(G) \cdot |V(G)|^{n-1}$, ancak ve ancak $|V(G)| = 2 \cdot \alpha(G)$.

4.10 Grafların Çarpımının Bağımsızlık Sayıları

(Jha, P.K. and Slutzki, G., 1994)

$\times$: Kartezyen Çarpım, $\otimes$: Tensör Çarpım, $\boxtimes$: Güçlü Çarpım olmak üzere,

Teorem 4.10.1. $\alpha(G) \cdot \alpha(H) \leq \alpha(G \boxtimes H) \leq \alpha(G \times H) \leq \alpha(G \otimes H)$

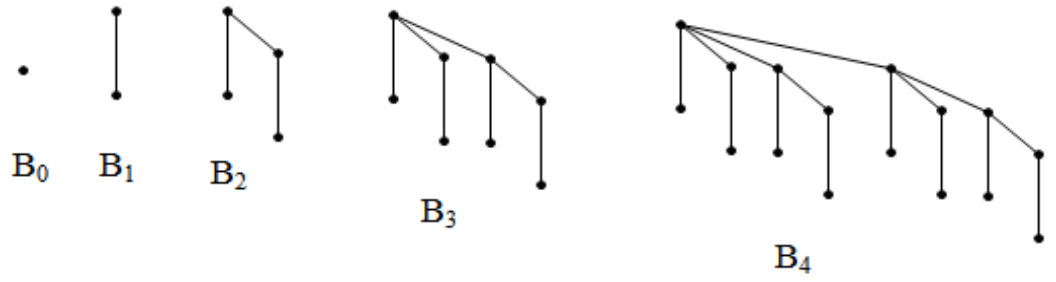
5. BINOMİAL AĞAÇLARIN KARTEZYEN ÇARPIMI SONUCU OLUŞAN GRAFLARIN TEPE ÖRTÜ SAYISI

Tanım 5.1. (Chartrand and Lesniak,1996) $n \geq 0$ için B_n binomial ağacında R ağacın kökü olmak üzere,

Binomial ağaç, $k \geq 0$ mertebeden R köklü bir ağaçtır . B_k aşağıdaki gibi tanımlanır.

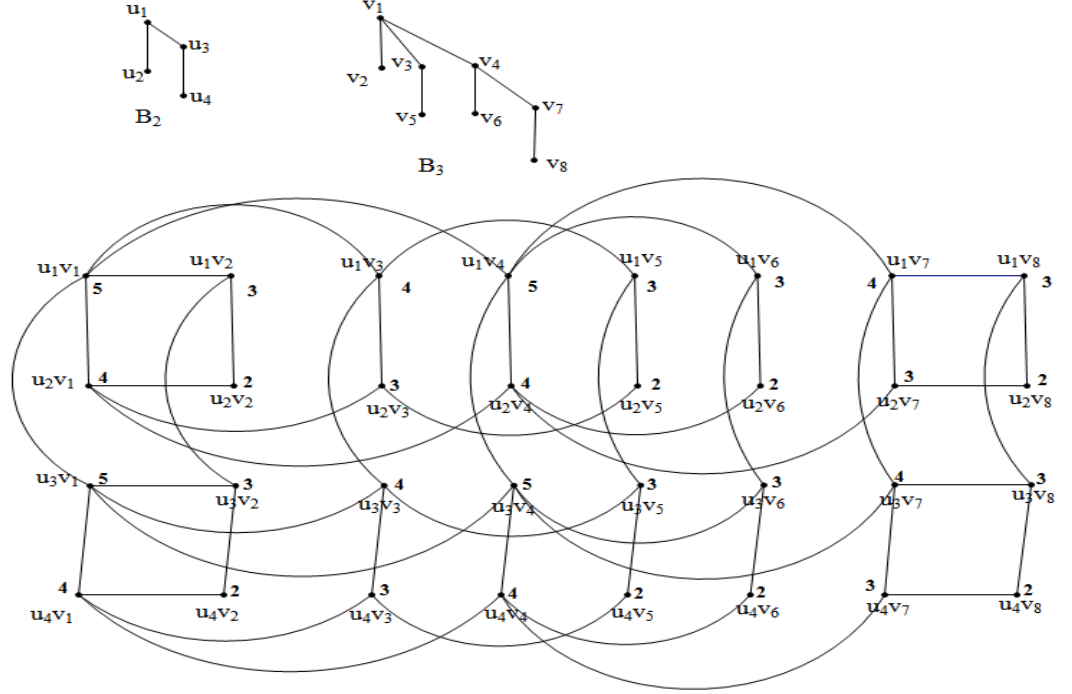
- 1.Eğer $k=0$ ise $B_k=B_0=R$, tek bir düğüm içerir.
- 2.Eğer $k>0$ ise, $B_k=\{R,B_0,B_1,\dots,B_{k-1}\}$. R köklü k binomial alt ağaçlarından oluşur.

Aşağıda bazı binomial ağaçlar gösterilmiştir.



Şekil 5.1 Binomial Ağaçlar.

5.1 $B_2 \times B_3$ Grafına Aç Gözlü Yaklaşım Algoritmasının Uygulanması



Şekil 5.2 $B_2 \times B_3$ grafi

$B_2 \times B_3$ grafi şöyle oluşturulmuştur. Öncelikle tepe kümesi oluşturuldu.

$$V(B_2 \times B_3) = \{u_1v_1, u_1v_2, u_1v_3, u_1v_4, u_1v_5, u_1v_6, u_1v_7, u_1v_8, u_2v_1, u_2v_2, u_2v_3, u_2v_4, u_2v_5,$$

$$u_2v_6, u_2v_7, u_2v_8, u_3v_1, u_3v_2, u_3v_3, u_3v_4, u_3v_5, u_3v_6, u_3v_7, u_3v_8,$$

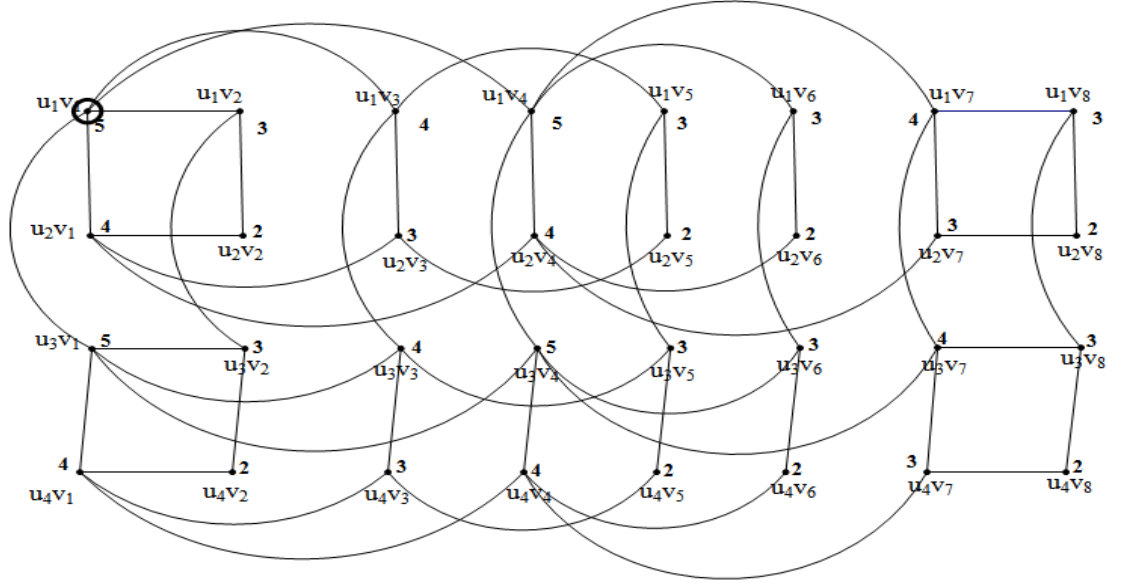
$$u_4v_1, u_4v_2, u_4v_3, u_4v_4, u_4v_5, u_4v_6, u_4v_7, u_4v_8 \}$$

Herhangi iki tepe arasındaki ayrıt şöyle çizilir.

Çarpım kümesinden herhangi iki tepe alınır, başlangıç tepeleri aynı ve bitiş tepeleri komşuysa ya da bitiş tepeleri aynı başlangıç tepeleri komşuysa bu iki tepe arasına bir ayrıt çizilir. u_1v_1, u_1v_2 tepeleri ele alındığında, her ikisinin de başlangıç tepesi aynıdır ve bitiş tepeleri olan v_1 ile v_2 komşu olduğundan u_1v_1, u_1v_2 tepeleri arasına bir ayrıt çizilmiştir.

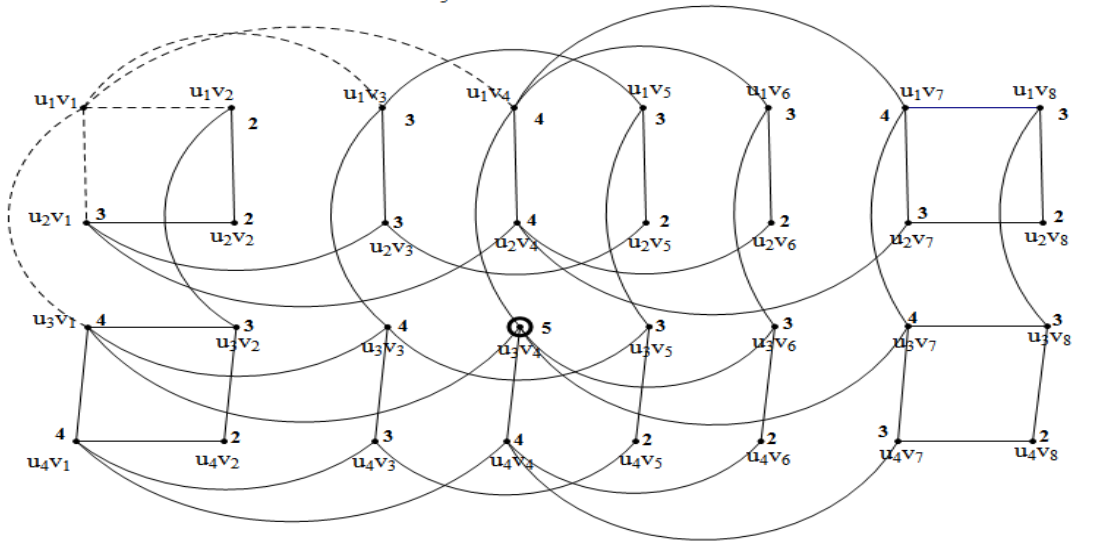
1.Adım : Derecesi en büyük olan tepeler, u_1v_1 , u_3v_1 , u_3v_4 bunlardan herhangi biri alınabilir. İlk olarak u_1v_1 tepesi alınır ve ona bitişik olan ayrıtlar graftan silinir.

$$C=\{u_1v_1\}.$$



Şekil 5.3 Algoritmanın uygulanışı 1. Adım.

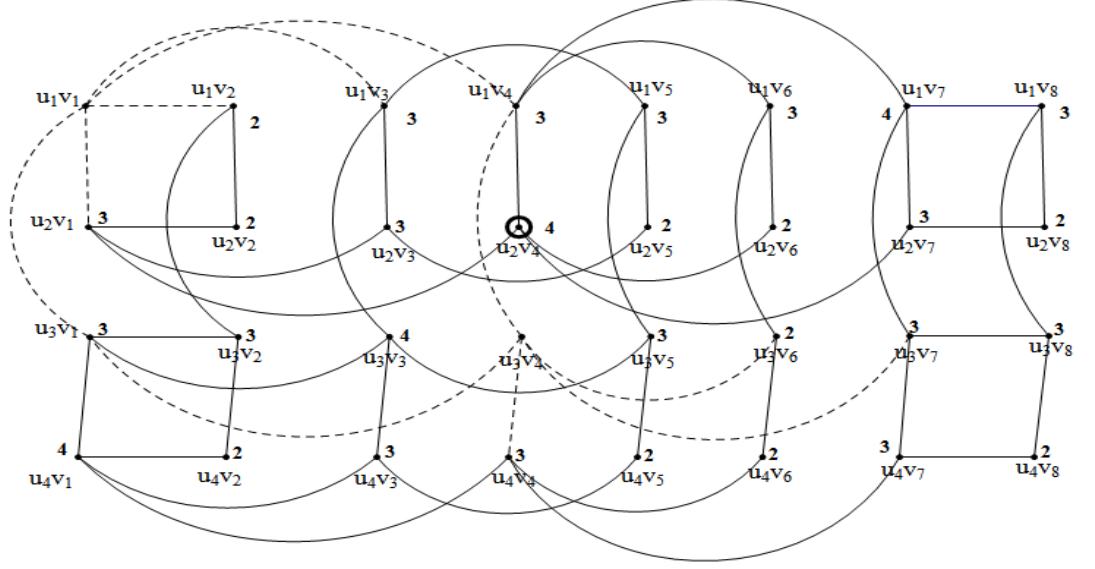
2.Adım : Derecesi en büyük olan tepe u_3v_4 tepesidir. u_3v_4 alınır ve ona bitişik olan ayrıtlar graftan silinir. $C=\{u_1v_1, u_3v_4\}$.



Şekil 5.4 Algoritmanın uygulanışı 2. Adım.

3.Adım: u_1v_7 , u_2v_4 , u_3v_3 , u_4v_1 tepelerinin derecesi en büyüktür, herhangi biri alınabilir. u_2v_4 alınır ve ona bitişik olan ayrıtlar graftan silinir.

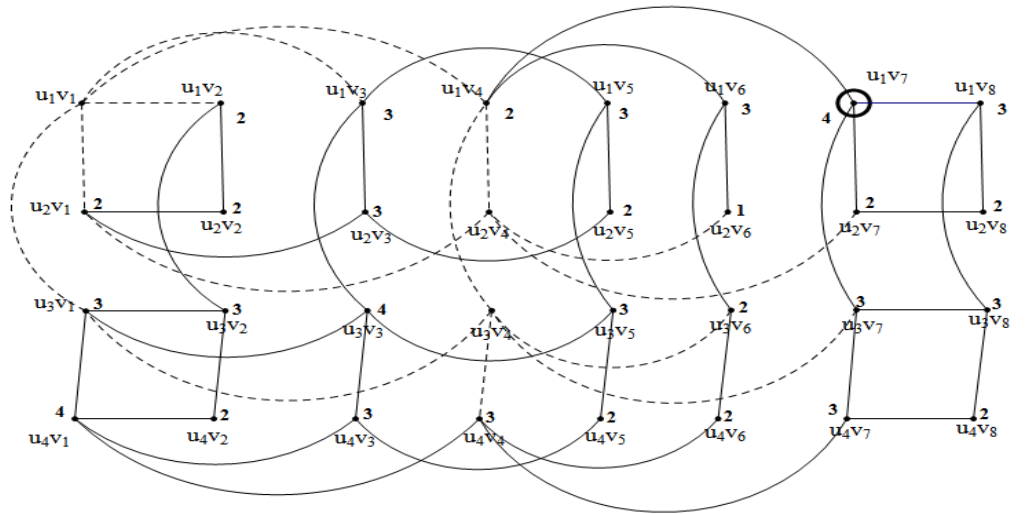
$$C=\{u_1v_1, u_3v_4, u_2v_4\}.$$



Şekil 5.5 Algoritmanın uygulanışı 3.Adım.

4.Adım: u_1v_7 , u_3v_3 , u_4v_1 tepelerinin derecesi en büyüktür, herhangi biri alınabilir. u_1v_7 alınır ve ona bitişik olan ayrıtlar graftan silinir.

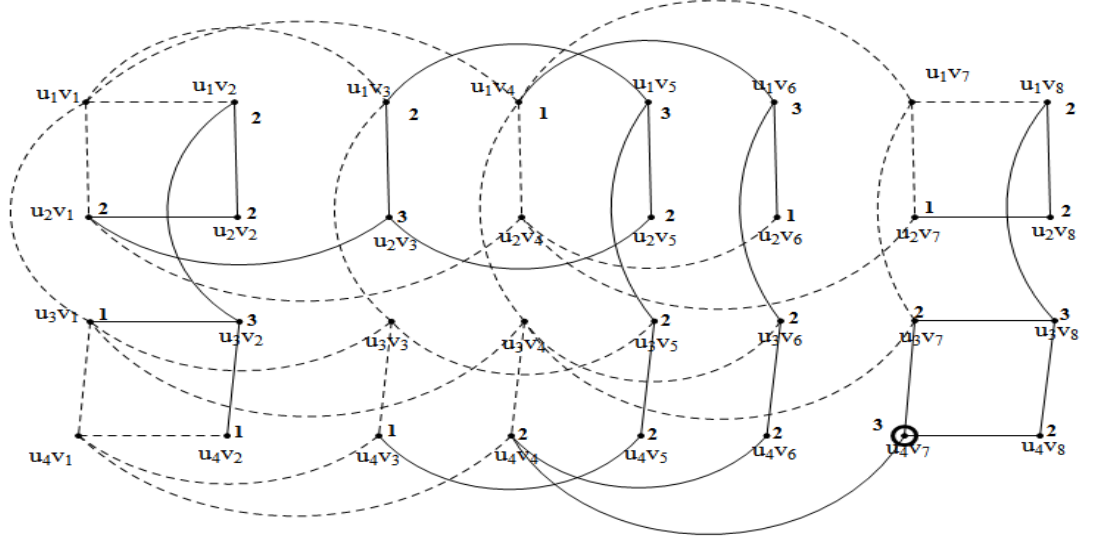
$$C=\{u_1v_1, u_3v_4, u_2v_4, u_1v_7\}.$$



Şekil 5.6 Algoritmanın uygulanışı 4.Adım.

7.Adım : $u_1v_5, u_1v_6, u_2v_3, u_3v_2, u_3v_8, u_4v_7$, tepelerinin dereceleri en büyüktür, rastgele u_4v_7 alınır ve ona bitişik olan ayrıtlar graftan silinir.

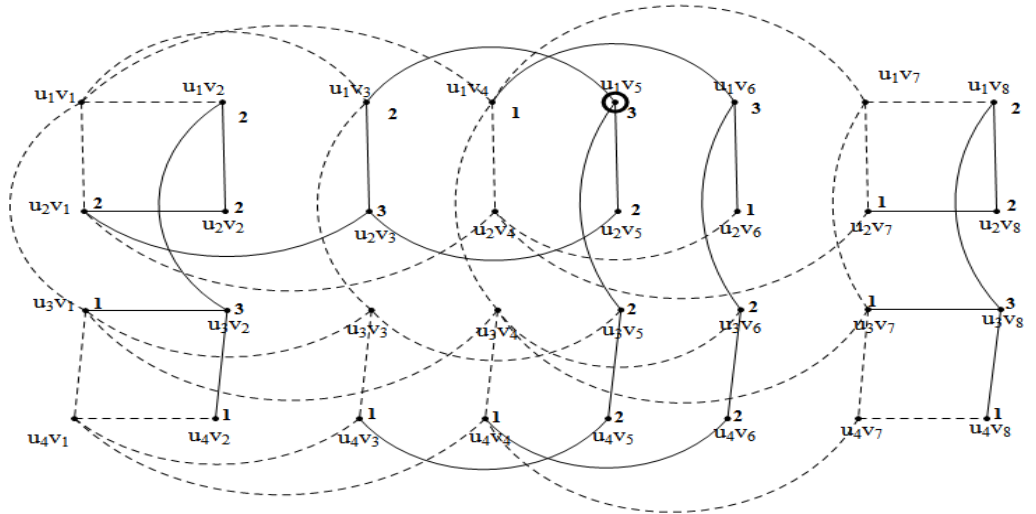
$$C=\{u_1v_1, u_3v_4, u_2v_4, u_1v_7, u_4v_1, u_3v_3, u_4v_7\}.$$



Şekil 5.9 Algoritmanın uygulanışı 7.Adım

8.Adım : $u_1v_5, u_1v_6, u_2v_3, u_3v_2, u_3v_8$, tepelerinin dereceleri en büyüktür, rastgele u_1v_5 alınır ve ona bitişik olan ayrıtlar graftan silinir.

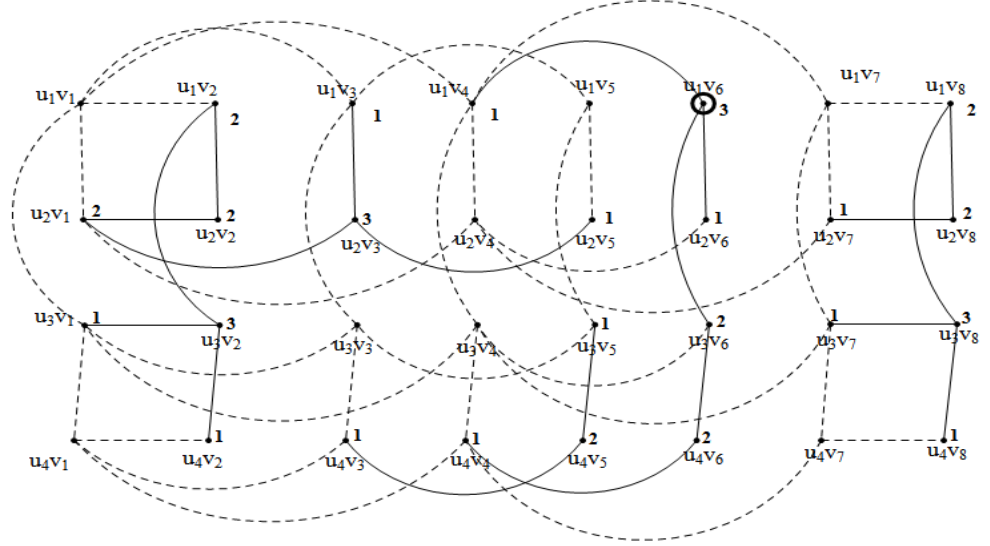
$$C=\{u_1v_1, u_3v_4, u_2v_4, u_1v_7, u_4v_1, u_3v_3, u_4v_7, u_1v_5\}.$$



Şekil 5.10 Algoritmanın uygulanışı 8.Adım

9.Adım: $u_1v_6, u_2v_3, u_3v_2, u_3v_8$, tepelerinin dereceleri en büyüktür, rastgele u_1v_6 alınır ve ona bitişik olan ayrıtlar graftan silinir.

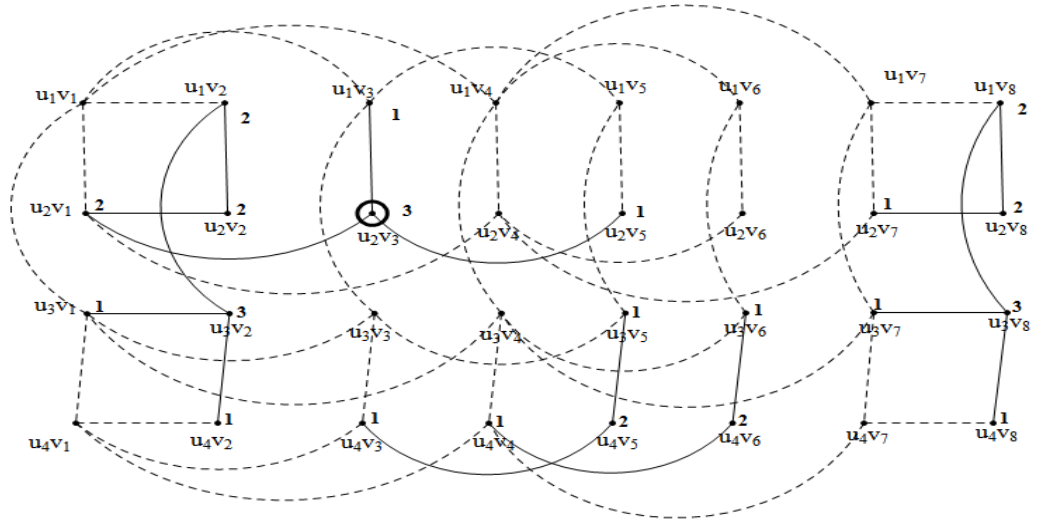
$$C=\{u_1v_1, u_3v_4, u_2v_4, u_1v_7, u_4v_1, u_3v_3, u_4v_7, u_1v_5, u_1v_6\}.$$



Şekil 5.11 Algoritmanın uygulanışı 9.Adım

10.Adım : u_2v_3, u_3v_2, u_3v_8 , tepelerinin dereceleri en büyüktür, rastgele u_2v_3 alınır ve ona bitişik olan ayrıtlar graftan silinir.

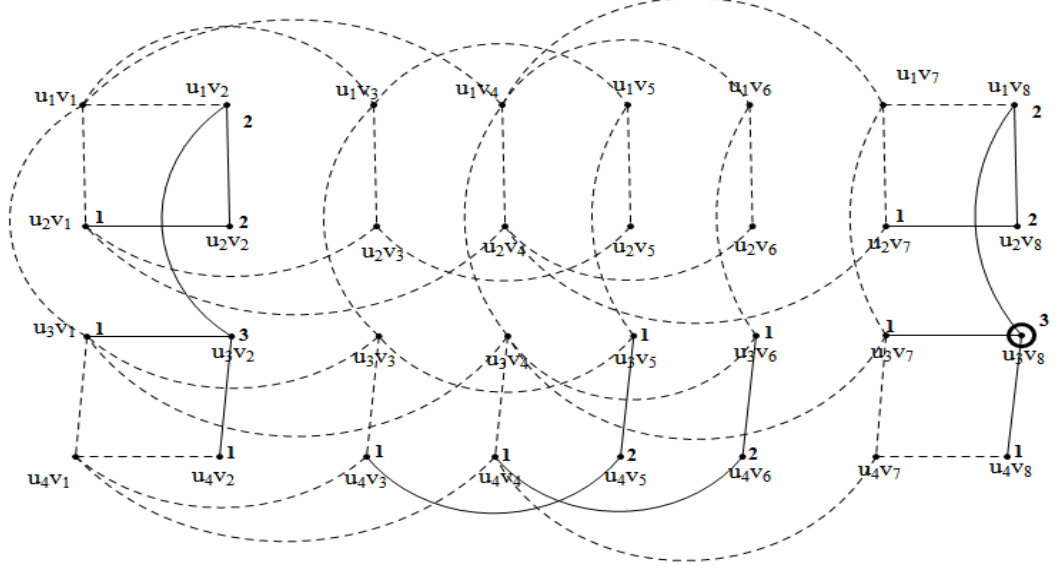
$$C=\{u_1v_1, u_3v_4, u_2v_4, u_1v_7, u_4v_1, u_3v_3, u_4v_7, u_1v_5, u_1v_6, u_2v_3\}.$$



Şekil 5.12 Algoritmanın uygulanışı 10.Adım

11.Adım : u_3v_2 , u_3v_8 , tepelerinin dereceleri en büyüktür, rastgele u_3v_8 alınır ve ona bitişik olan ayrıtlar graftan silinir.

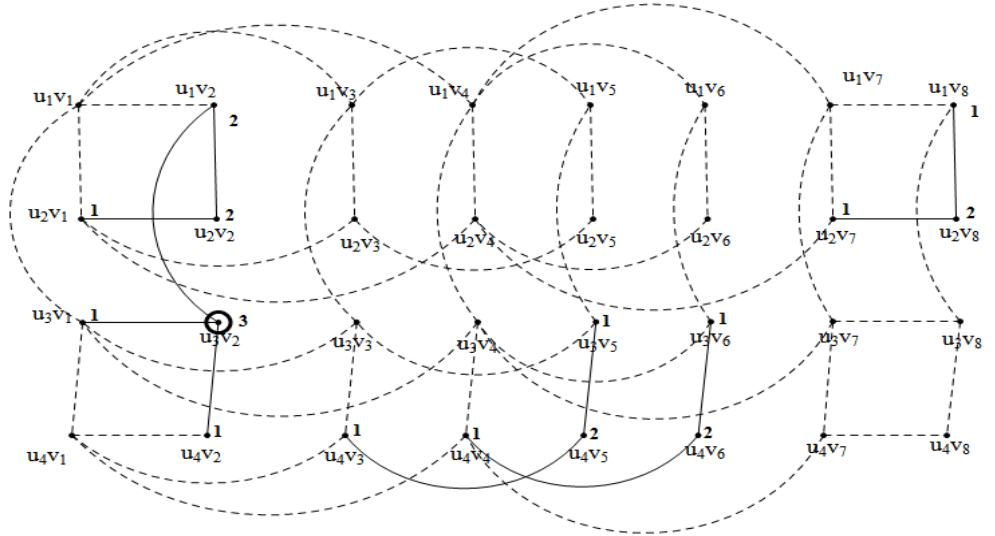
$$C=\{u_1v_1, u_3v_4, u_2v_4, u_1v_7, u_4v_1, u_3v_3, u_4v_7, u_1v_5, u_1v_6, u_2v_3, u_3v_8\}.$$



Şekil 5.13 Algoritmanın uygulanışı 11.Adım

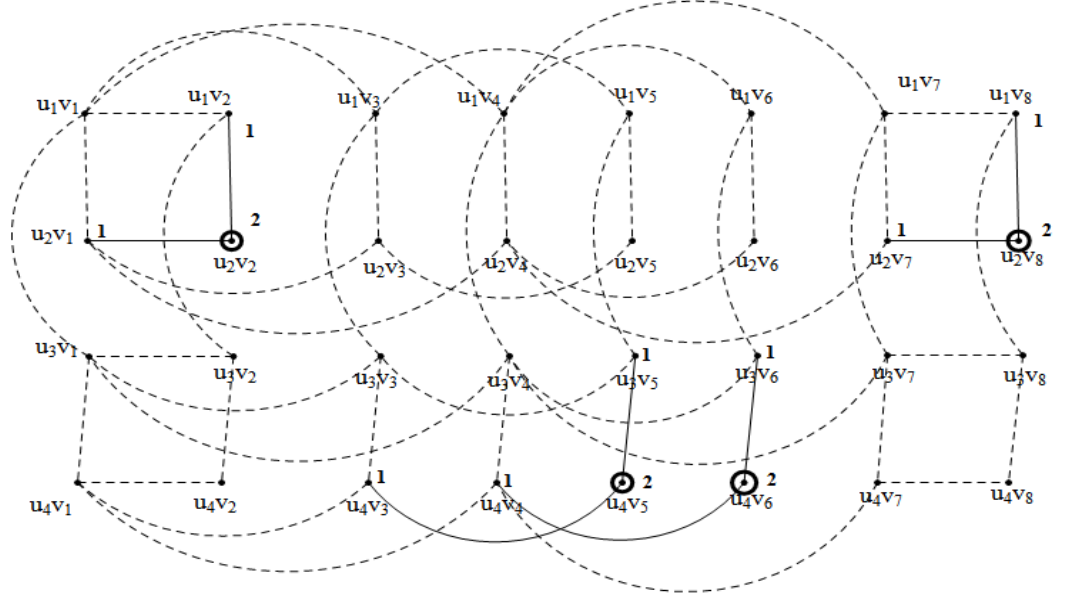
12.Adım : u_3v_2 , tepesinin derecesi en büyüktür, u_3v_2 alınır ve ona bitişik olan ayrıtlar graftan silinir.

$$C=\{u_1v_1, u_3v_4, u_2v_4, u_1v_7, u_4v_1, u_3v_3, u_4v_7, u_1v_5, u_1v_6, u_2v_3, u_3v_8, u_3v_2\}.$$



Şekil 5.14 Algoritmanın uygulanışı 12.Adım

13., 14., 15. ve 16. Adımlar: $u_2v_2, u_2v_8, u_4v_5, u_4v_6$ tepeleri alınır ve algoritma sonlanır.



Şekil 5.15 Algoritmanın uygulanışı 13.,14.,15. Ve 16.Adımlar.

$$C = \{u_1v_1, u_3v_4, u_2v_4, u_1v_7, u_4v_1, u_3v_3, u_4v_7, u_1v_5, u_1v_6, u_2v_3, u_3v_8, u_3v_2, u_2v_2, u_2v_8, u_4v_5, u_4v_6\}$$

$$|C| = 16.$$

Aç gözlü yaklaşım algoritmasıyla, $B_2 \times B_3$, $B_2 \times B_4$, $B_2 \times B_5$ ve $B_2 \times B_6$ için bazı sonuçlara ulaşılmıştır.

Ulaşılan sonuçlar aşağıda incelenmiştir.

Daha önce verilen lemmadan G , n .ci dereceden basit bir graf ise, o zaman $\alpha(G)$, G grafinin bağımsızlık sayısı ve $\beta(G)$, G grafinin tepe örtme sayısı olmak üzere, $\alpha(G) + \beta(G) = n$ idi.

Buna göre

$$\beta(B_2) = 2 \text{ ve } \beta(B_3) = 4 \text{ olmak üzere,}$$

$$\alpha(B_2) + \beta(B_2) = 4 \text{ ve } \alpha(B_3) + \beta(B_3) = 8 .$$

$$\alpha(B_2) = 2 \text{ ve } \alpha(B_3) = 4 .$$

$$|V(B_2)| = 4 \text{ ve } |V(B_3)| = 8 .$$

Vizing 'in teoreminde, herhangi bir G ve H grafi için,

$$\alpha(G \times H) \leq \min \{ \alpha(G) \cdot |V(H)|, \alpha(H) \cdot |V(G)| \} \text{ idi.}$$

$$\alpha(B_2 \times B_3) \leq \min \{ \alpha(B_2) \cdot |V(B_3)|, \alpha(B_3) \cdot |V(B_2)| \}$$

$$\alpha(B_2 \times B_3) \leq \min \{ 2 \cdot 8, 4 \cdot 4 \}$$

$$\alpha(B_2 \times B_3) \leq \min \{ 16, 16 \}$$

$$\alpha(B_2 \times B_3) \leq 16$$

$$\alpha(B_2 \times B_3) + \beta(B_2 \times B_3) = |V(B_2)| \cdot |V(B_3)| \quad (*)$$

$$\alpha(B_2 \times B_3) + \beta(B_2 \times B_3) = 4 \cdot 8 = 32$$

$$(*) \text{ dan } 16 \leq \beta(B_2 \times B_3).$$

Vizing 'in diğer bir teoreminde ise,

$$\alpha(G \times H) \geq \alpha(G) \cdot \alpha(H) + \min \{ |V(G)| - \alpha(G), |V(H)| - \alpha(H) \}.$$

Buna göre,

$$\alpha(B_2 \times B_3) \geq \alpha(B_2) \cdot \alpha(B_3) + \min \{|V(B_2)| - \alpha(B_2), |V(B_3)| - \alpha(B_3)\}$$

$$\alpha(B_2 \times B_3) \geq 2 \cdot 4 + \min \{4 - 2, 8 - 4\}$$

$$\alpha(B_2 \times B_3) \geq 8 + \min \{2, 4\}$$

$$\alpha(B_2 \times B_3) \geq 10$$

$$(*) \text{ dan } \beta(B_2 \times B_3) \leq 22$$

Böylece $16 \leq \beta(B_2 \times B_3) \leq 22$ dir.

Görüldüğü gibi aç gözlü yaklaşım algoritması alt sınır olan 16 ya ulaşmıştır. Bu sonuç, incelenmiş olan $(B_2 \times B_6)$ ya kadar olan çarpımlar için de geçerlidir.

Çizelge 5.1 Bulunan sonuçlar.

$B_2 \times B_3$	$\beta(B_2 \times B_3) = 16$
$B_2 \times B_4$	$\beta(B_2 \times B_4) = 32$
$B_2 \times B_5$	$\beta(B_2 \times B_5) = 64$
$B_2 \times B_6$	$\beta(B_2 \times B_6) = 128$

6. SONUÇ VE DEĞERLENDİRME

Bu tezde minimum tepe örtüsü problemi çeşitli türleriyle ele alınmıştır. Özellikle de minimum tepe örtüsü sayısı üzerine temel tanımlar, teoremler, algoritmalar açıklanmıştır. Minimum tepe örtüsü problemi polinom zamanda çözilemeyen bir problem olup NP-Tam'dır. Problem, ancak küçük boyutlu veri gruplarında tam olarak çözülebilir veya yaklaşık algoritmalarla optimale yakın sonuçlar bulunarak çözümlenmeye çalışılmaktadır.

Minimum tepe örtüsü problemin Karp'ın 21 Np-Tam problem listesinin en önde gelen ve en çok çalışılan problemlerinden biridir. Özellikle de diğer Np-Tam problemlerle ilişkisi düşünüldüğünde, bu çalışmaların önemi büyüktür.

Minimum tepe örtüsü problemi için 2-yaklaşık algoritma, aç gözlü yaklaşım (greedy) algoritması ve Alom'un algoritması incelenmiş, algoritmaların aksak yönleri ve birbirlerinden üstünlükleri olup olmadığı irdelenmiştir.

Graf işlemleri, tekli ve ikili işlemler olarak ele alınmıştır. Özellikle güçlü çarpım, lexicografic çarpım, kartezyen çarpım üzerinde durulmuş ve bu işlemler sonucu oluşan grafların bağımsızlık sayısı ve tepe örtü sayısı arasındaki ilişkiye yer verilmiştir.

Bağımsızlık sayısı ve tepe örtüsü sayısı arasındaki ilişki, bu tezin temel kaynağı olmuştur. Çünkü G , n .ci dereceden basit bir graf ve $\alpha(G)$, G grafının bağımsızlık sayısı ve $\beta(G)$, G grafının tepe örtme sayısı olmak üzere, $\alpha(G)+\beta(G)=n$ dir.

Vizing tarafından 1963 yılında grafların kartezyen çarpımının bağımsızlık sayısı için bazı sınırlar verilmiştir. G ve H herhangi iki graf olmak üzere,

$$\alpha(G \times H) \leq \min \{ \alpha(G) \cdot |V(H)|, \alpha(H) \cdot |V(G)| \}$$

ve

$$\alpha(G \times H) \geq \alpha(G) \cdot \alpha(H) + \min \{ |V(G)| - \alpha(G), |V(H)| - \alpha(H) \}.$$

Binomial ağaçların tanımı verilmiş ve $B_2 \times B_3$, $B_2 \times B_4$, $B_2 \times B_5$ ve $B_2 \times B_6$ çarpımları yapılmış, oluşan graflar üzerinde aç gözlü yaklaşım algoritmasıyla tepe örtü sayıları bulunmuştur.

Bulunan tepe örtüsü sayılarının yukarıda verilen alt sınıra eşit olduğu gözlemlenmiştir. Ancak aç gözlü yaklaşım algoritmasının optimal sonucu garanti etmediği bilinmektedir.

Burada aç gözlü yaklaşım algoritması ile teoremden verilen optimal sonuca ulaşıldığından diğer algoritmalar kullanılmamıştır.

Aç gözlü yaklaşım algoritmasının $B_2 \times B_n$ sonucu oluşan graflarda veya $B_m \times B_n$ sonucu oluşan graflarda optimal çözümü bulabileceği yoruma açıktır.

KAYNAKLAR DİZİNİ

- Alom, B.M., Das, S. and Rouf, M.A.,** 2011, Performans Evaluation of Vertex Cover and Set Cover Problem using Optimal Algorithm, DUET Journal, Vol. 1, Issue 2.
- Avis, D., Imamura, T.,** 2007, A List Heuristic for Vertex Cover, Operations Research Letters, March,, 201-204pp.
- Baitiang, C., Sitthiwirattham, T.,** 2012, Independent and vertex covering number on strong product of complete graphs, International Journal of Pure and Applied Mathematics Volume 81 No. 3 2012, 497-503pp.
- Balaji, S., Swaminathan, V. and Kannan, K.,** 2010, Effective algorithm for minimum weighted vertex cover problem, International Journal of Computational and Mathematical Sciences, 4(1) (2010) 34-38pp.
- Bar-Yehuda, R., Flysher, G., Mestre, J.,** 2007, Rawitz Approximation of Partial Capacitated Vertex Cover ESA'07 Proceedings of the 15th annual European conference on Algorithms.
- Bar-Yehuda, R. and Even, S.,** 1981, A linear time approximation algorithm for the weighted vertex cover problem, Jurnal of Algorithms, 2:198-203pp.
- Bar-Yehuda, R. and Even, S.,** 1985, A local-ratio theorem for approximating the weighted vertex cover problem, Annals of Discrete Mathematics, 25:27-46pp.
- Bar-Yehuda, R., Hermelin, D. and Rawitz, D.,** 2011, Computational Geometry: Theory and Applications - COMGEO, vol. 44, no. 6, 356-364 pp.
- Barnes, B.H., Mackey, K.E.,** 1978, A generalized measure of independence and the strong product of graphs, Networks, 8 (1978), 135-151.

KAYNAKLAR DİZİNİ (Devam)

- Chen, Jianer; Kanj, Iyad A.; Xia, Ge**, 2006, Improved Parameterized Upper Bounds for Vertex Cover, Lecture Notes in Computer Science, **4162**: 238–249pp.
- Clarkson, K.**, 1983, A modification of the greedy algorithm for vertex cover, Information Processing Letters, 16:23-25pp.
- Cook, S.A.**, 1971, The complexity of theorem proving procedures, Proceedings, Third Annual ACM Symposium on the Theory of Computing, ACM, New York, 151–158pp.
- Cormen, H.T., Leiserson, E.C., Rivest and L.R., Stein, C.**, 2002, Introduction to Algorithms, , McGraw-Hill Book Com., 1145p.
- Dharwadkar, A.**, 2011, The Vertex Cover Algorithm, Createspace,.
- Diestel, R.**, 2000, Graph Theory, Electronic Edition, 451p.
- Dinur, I., Safra, S.**, 2005, On the hardness of approximating minimum vertex cover, Annals of Mathematics 162 (1): 439–485pp.
- Dinur, I., Safra, S.**, 2002, The importance of being biased, In Proceedings of the 34th annual Acm Symposium on Theory Of Computing (STOC), 33-42pp.
- Downey, R.G. and Fellows, M.R.**, 1999, Monographs in Computer Science XV, 533 p.
- Garey, M., and Johnson, D.**, 1979, Computers and Intractability: A Guide to the Theory of NP-completeness, Freeman, San Francisco, 339p.
- Geller, D. and Stahl, S.**, 1975, The chromatic number and other functions of the lexicographic product. J. Combin. Theory Ser. B, 19:87–95pp.
- Guha, S., Hassin, R., Khuller, S., Or, E.**, 2002, Capacitated Vertex Covering with Applications, Soda '02 Proceedings of the eleventh annual ACM-SIAM symposium on Discrete Algorithms.
- Guo, J., Niedermeier, R., Wernicke, S.**, 2007, Parameterized complexity of vertex cover variants, Theory Comput. Syst. 41, 501-520pp.

KAYNAKLAR DİZİNİ (Devamı)

- Halperin, E.**, 2000, Improved Approximation Algorithms for the Vertex Cover Problem in Graphs and Hypergraphs, Soda '00 Proceedings of the eleventh annual ACM-SIAM symposium on Discrete.
- Hammack, R., Imrich, W., Klavzar, S.**, 2011, Handbook of product graphs 2.ed, CRC.
- Harary, F.**, 1969, Graph Theory. Addison Wesley, Reading, MA.
- Hochbaum, D.**,1982, Approximation algorithms for the set covering and vertex cover problems, SIAM Journal on Computing, 11(3):555-556pp.
- Jain, S., Sharma, A.**, 2012, Vertex cover using approximating approach, IPAJOURNALS research article.
- Jha, P.K. and Slutzki, G.**, 1994, Applied Mathematics Letters, vol. 7, no. 4, 91-94pp.
- Karakostas, G.**, 2005, A better approzimation ratio for the vertex cover problem.In 32nd International Colloquium on Automata, Languages and Programming, 1043-1050pp.
- Karp, M.R.**, 1972, Reducibility Among Combinatorial Problems,In Complexity of Computer Computation, 85-103pp.
- Khuller, S.**, 2002, Algorithms Column:The Vertex Cover Problem, Sigact News-SIGACT, vol. 33, no. 2, 31-33pp.
- Klavzar, S.**, 2005, Some new bounds and exact results on the independence number of Cartesian product graphs Ars Combin. 74 (2005) 173–186pp.
- Kumar, K. V. R.**, 2009, Choosing the Efficient Algorithm for Vertex Cover Problem, Computer Sicence and Engineering Department Thapar University.

KAYNAKLAR DİZİNİ (Devam)

- Monien, B. and Speckenmeyer, E.,** 1985, Ramsey numbers and an approximation algorithm for the vertex cover problem, Acta Informatica 22(1):115-123pp.
- Nabiyev, Vagıfoğlu, V.,** 2009, Teoriden uygulamalara algoritmalar, Seçkin Yayıncılık, Ankara, 824s.
- Neidermeier R.,** 2006, Invitation to Fixed Parameter Algorithms, Oxford University Press.
- Neidermeier, R. and Rossmanith, P.,** 2003, On efficient fixed-parameter algorithms for weighted vertex cover, Journal of Algorithms, vol. 47, 63-77pp.
- Nemhauser, G. L. and Trotter, L. E.,** 1975, Vertex packings: structural properties and algorithms, Mathematical Programming, 8:232-248pp.
- Savage, C.,** 1982, Depth-First Search and The Vertex Cover Problem, Information Processing Letters, Volume 14, Number 5.
- Scheinerman, R.E,** 2012, Mathematics: A Discrete Introduction, Cengage Learning, 504p.
- Sitthiwiratttham, T.,** 2012, Independent and vertex covering number on kronecker product of $K_{m,n}$, Applied Mathematical Sciences, Vol. 6, 2012, no. 28, 1403-1408pp.
- Sonneman, E. and Krafft, O.,** 1974, Journal of Combinatorial Theory - JCT, vol. 17, no. 2, 133-142pp.
- Vizing, V.G.,** 1963 Cartesian Product of graphs, Vychisl Sistemy (in Russian), page 30-43, English translation Compt . El. Syst. 2 (1966), 352-365.
- Weight, M. and Hartmann, A. K.,** 2000, The number of guards needed by a museum - a phase transition in vertex covering of random graphs, Phys-Rev. Lett., 84p.
- West, D.B.,** 2001, Introduction the Graph Theory, Prentice-Hall.

ÖZGEÇMİŞ

04.10.1988 tarihinde Bulgaristan-Kırcaali ‘de doğdu.2005 yılında Bornova Çimentaş Lisesinden mezun oldu. Aynı yıl Ege Üniversitesi Fen Fakültesi Matematik Bölümünü kazandı. 2005-2006 döneminde İngilizce hazırlık sınıfı okudu. Lisans öğrenimi boyunca İzmir Büyükşehir Belediyesinden karşılıksız burs aldı. 2010 yılında Ege Üniversitesi Matematik Bölümü Bilgisayar Bilimleri Anabilim Dalından mezun oldu. 2010-2011 döneminde Ege Üniversitesi Eğitim Fakültesinden Pedagojik Formasyon sertifikası aldı. 2011-2012 döneminde Ege Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalında Yüksek Lisans eğitimine başladı. Aynı dönem İzmir Ticaret Odasından karşılıksız burs aldı. 2012 yılının eylül ayında Manisa ‘nın Turgutlu ilçesindeki Hasan Ferdi Turgutlu Anadolu Meslek ve Teknik Lisesine Matematik Öğretmeni olarak atandı. Halen aynı yerde görevine devam etmektedir. 2-4 Mayıs 2014 tarihleri arasında Türkiye Kadın Matematikçiler Derneğinin düzenlediği, Kocaeli- Gebze‘ de yapılan 1.Kadın Matematikçiler Derneği Çalıştayına “Minimum Tepe Örtüsü Problemi Üzerine” adlı poster sunumuyla katıldı.