

**T.C.  
MANİSA CELAL BAYAR ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS TEZİ  
YAZILIM MÜHENDİSLİĞİ ANABİLİM DALI  
YAZILIM MÜHENDİSLİĞİ BİLİM DALI**

**AĞLARDA ORTALAMA BAĞLAMA SAYISI ÜZERİNE**

**Ayşe KAHVECİ ÖZDEMİR**

**Danışman  
Doç. Dr. Ersin ASLAN**



**MANİSA-2021**

## **TEZ ONAYI**

**Ayşe KAHVECİ ÖZDEMİR** tarafından hazırlanan "**Ağlarda Ortalama Bağlama Sayısı Üzerine**" adlı tez çalışması 01/02/2021 tarihinde aşağıdaki jüri üyeleri önünde Celal Bayar Üniversitesi Fen Bilimleri Enstitüsü **Yazılım Mühendisliği Anabilim Dalı**'nda **YÜKSEK LİSANS TEZİ** olarak savunulmuş ve **oybirliği** ile başarılı olarak kabul edilmiştir.

**Danışman**

**Doç. Dr. Ersin ASLAN**

Manisa Celal Bayar Üniversitesi

**Jüri Üyesi**

**Doç.Dr. Vecdi AYTAÇ**

Ege Üniversitesi

**Jüri Üyesi**

**Dr. Öğr. Üyesi Emin BORANDAĞ**

Manisa Celal Bayar Üniversitesi

## **TAAHHÜTNAME**

Bu tezin Celal Bayar Üniversitesi Fen Edebiyat Fakültesi Yazılım Mühendisliği'nde, akademik ve etik kurallara uygun olarak yazıldığını ve kullanılan tüm literatür bilgilerinin referans gösterilerek tezde yer aldığını beyan ederim.

**Ayşe KAHVECİ ÖZDEMİR**

## İÇİNDEKİLER

	<b>Sayfa</b>
İÇİNDEKİLER .....	I
SİMGELER VE KISALTMALAR DİZİNİ.....	II
ŞEKİLLER DİZİNİ.....	III
TABLO DİZİNİ .....	V
TEŞEKKÜR.....	VI
ÖZET.....	VII
ABSTRACT.....	VIII
1. GİRİŞ .....	1
1.1. Çizge Teori ve Uygulama Alanları .....	1
1.2. Zedelenebilirlik.....	3
2. GENEL TANIMLAR.....	5
3. ORTALAMA BAĞLAMA SAYISI .....	9
3.1. Tanım ve Teoremler .....	9
3.2. Bazı Özel Çizgelerin Ortalama Bağlama Sayısı.....	12
4. YAZILIM METRİKLERİ.....	16
5. ALGORİTMALAR.....	20
6. TEST SONUÇLARI VE KARMAŞIKLIK.....	22
7. SONUÇ .....	25
KAYNAKLAR .....	27
ÖZGEÇMİŞ .....	29

## SİMGELER VE KISALTMALAR DİZİNİ

<b>G</b>	Çizge
<b>E(G)</b>	Bir G çizgesinin ayrıtlar kümesi
<b>V(G)</b>	Bir G çizgesinin tepeler kümesi
<b>n</b>	Bir G çizgesinin tepe sayısına bağlı seviyesi
<b>v</b>	Bir çizgenin tepesi
<b>e</b>	Bir çizgenin ayrıtı
<b>deg(V)</b>	v tepesinin derecesi
<b><math>\Delta(G)</math></b>	Bir G çizgesinin en büyük tepe derecesi
<b><math>\delta(G)</math></b>	Bir G çizgesinin en küçük tepe derecesi
<b><math>\alpha(G)</math></b>	Bağımsızlık sayısı
<b>sc(G)</b>	Saçılım sayısı
<b><math>\lambda(G)</math></b>	Ayrıtlar bağlantılılık sayısı
<b><math>P_n</math></b>	n tepeli yol çizge
<b><math>C_n</math></b>	n tepeli çevre çizge
<b><math>K_n</math></b>	n tepeli tam çizge
<b><math>K_{1,n}</math></b>	n+1 tepeli yıldız çizge
<b><math>W_{1,n}</math></b>	n+1 tepeli tekerlek çizge
<b><math>G_n</math></b>	2n+1 tepeli gear çizge
<b><math>G^*</math></b>	Dikenli çizge
<b><math>P^*</math></b>	Dikenli çizge
<b><math>K_{1,n}</math></b>	Yıldız çizge

$C_{n,m}$	Kuyruklu yıldız
$K_{r,m}$	Double Star
$B_{r,m}$	Muz ağacı (Banana Tree)
$\alpha(G)$	Kaplama sayısı (Covering number)
$\beta(G)$	Bağımsızlık sayısı (Independence number)
$\kappa(G)$	Bağlantılılık sayısı (Connectivity number)
$H$	$G$ 'nin alt çizgesi (Spanning subgraph)
$bind_v$	Yerel bağlama sayısı
$bind_{av}$	Ortalama bağlama sayısı

## ŞEKİLLER DİZİNİ

	<b>Sayfa</b>
Şekil 1.1. Bilgisayar Ağı .....	1
Şekil 2.1. Tam Çizge ( $K_4$ ) .....	6
Şekil 2.2. Yol Çizge ( $P_5$ ) .....	6
Şekil 2.3. Dikenli Çizge (Thorn Graph) .....	7
Şekil 2.4. Yıldız Çizge ( $K_{1,5}$ ) .....	7
Şekil 2.5. Kuyruklu Yıldız ( $C_{4,4}$ ) .....	7
Şekil 2.6. Double Star ( $K_{4,5}$ ) .....	8
Şekil 2.7. Muz Ağacı Çizgesi ( $B_{5,4}$ ) .....	8

## TABLO DİZİNİ

	<b>Sayfa</b>
Tablo 1. Risk Değerlendirmesi .....	18
Tablo 2. Bazı Yazılım Metriklerinin Hesaplanması .....	19
Tablo 3. Bazı Özel Metriklerin Sonuçları .....	22
Tablo 4. bind <sub>av</sub> ' ın Ham (Raw) Metrikleri .....	23
Tablo 5. bind <sub>av</sub> ' ın Halstead Metrikleri .....	24
Tablo 6. bind <sub>av</sub> ' ın Sürdürülebilirlik İndeksi .....	24
Tablo 7. bind <sub>av</sub> ' ın Siklomatik Karmaşıklığı .....	25



## TEŐEKKÜR

Çalıőmamın ve lisansüstü eğitimi boyunca bana destek olan, bilgi ve tecrübeleri ile yol gösteren danışman hocam Sayın Doç. Dr. Ersin ASLAN'a, lisansüstü öğrenim hayatımın tüm zorlu aşamalarında yardımcı olan, desteğini hiç esirgemeyen, kendisini tanımaktan onur duyduğum arkadaşım Mehmet Aykut TOSUN'a, hayatımın her safhasında bana destek olan eşim Yusuf ÖZDEMİR'e ve öğrenim hayatımda maddi manevi destek veren ve her zaman yanımda olan aileme yürekten teşekkür ederim.

Ayőe KAHVECİ ÖZDEMİR  
Manisa, 2021

## ÖZET

### Yüksek Lisans Tezi

### Ağlarda Ortalama Bağlama Sayısı Üzerine

Ayşe KAHVECİ ÖZDEMİR

Celal Bayar Üniversitesi Fen Bilimleri Enstitüsü  
Yazılım Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Ersin ASLAN

Bilgi teknolojilerindeki hızlı gelişim ve rekabetin etkileşimi sonucunda bir ağın güvenilirliği ve ne kadar sağlam kaldığı önemlidir. Bir ağdaki bazı merkezler veya merkezler arasındaki ağ bağlantılarının hasar görmesi ile iletişim kesilene kadar ağın dayanıklılığını ölçmeye ağın zedelenebilirliği denir. Bir iletişim ağının merkezlerinde veya bağlantı hatlarında oluşabilecek hasarlara karşı ağın dayanıklılığını inceleyebilmek için; ağın merkezleri çizgenin tepeleri, merkezler arasındaki bağlantıları da çizgenin ayrıtları olacak şekilde çizgeler modellenenbildiğinden, iletişim ağlarının zedelenebilirlik değerlerini araştırmak için çizgelerde çeşitli zedelenebilirlik parametreleri tanımlanmıştır. Bu makalede ortalama bağlama sayısı üzerine çalışılmıştır.  $v \in V(G)$  için,  $v$  tepesi için yerel bağlama sayısı  $bind_v(G)$  olarak gösterilir ve  $bind_v(G) = \min_{S \in F_v(G)} \left\{ \frac{|N(S)|}{|S|} \right\}$ , burada  $F_v(G) = \{S \subseteq V(G) | v \in S, S \neq \emptyset, N(S) \neq V(G)\}$ .  $G$  çizgesinin ortalama bağlama sayısı  $bind_{av}(G)$  şu şekilde tanımlanır,  $bind_{av}(G) = \frac{1}{n} \sum_{v \in V(G)} bind_v(G)$ , burada  $n$ ,  $G$  çizgesindeki tepe sayısıdır.

Bu çalışmada ortalama bağlama sayısı için genel sonuçlar verilmiştir. Ardından bazı özel çizgelerin ortalama bağlama sayıları bulunmuştur. Son olarak, çizgelerin ortalama bağlama sayılarının hesaplanması için algoritma verilmiştir. Algoritma, kod ölçütleriyle analiz edilmiş ve kullanışlılığı gösterilmiştir..

**Anahtar Kelimeler: Çizge Teorisi, Ağ Tasarımı, Bağlantılılık, Zedelenebilirlik, Ortalama Bağlama Sayısı, Algoritma.**

**2021, 40 sayfa**

## ABSTRACT

M.Sc. Thesis

### On Average Binding Number in Networks

Ayşe KAHVECİ ÖZDEMİR

Celal Bayar University

Graduate School of Applied and Natural Sciences Department of Mathematics

Supervisor: Doç. Dr. Ersin ASLAN

As a result of the interaction of rapid development and competition in information technologies, the reliability of a network and how solid it remains is important. It is called the hat vulnerability of the network to measure the endurance of the network until communication is interrupted by the deterioration of the connection lines between some centers or centers in a network. The centers of the network can be modeled such that the vertex of the network and the connecting lines are the distances of the graph, while investigating the strength of a communication network against disturbances that may occur in the centers or connecting lines. Networks can be modeled with graphs and there are several parameters to measure the vulnerability of these graphs. In this study, the average binding number was studied. For  $v \in V(G)$ , the local binding number of  $v$  is  $bind_v(G) = \min_{S \in F_v(G)} \left\{ \frac{|N(S)|}{|S|} \right\}$ , where  $F_v(G) = \{S \subseteq V(G) | v \in S, S \neq \emptyset, N(S) \neq V(G)\}$ . Furthermore, the average binding number of  $G$  is defined as  $bind_{av}(G) = \frac{1}{n} \sum_{v \in V(G)} bind_v(G)$ , where  $n$  is the number of vertices in graph  $G$ .

In this paper, some bounds of the average binding number are given. The average binding number of some special graphs are obtained. Finally, the algorithm for calculation of average binding numbers of graphs is given. The algorithms of these parameters are developed that calculates for any graph and the algorithms are explained. The algorithms are analyzed by code metrics and their usefulness is shown.

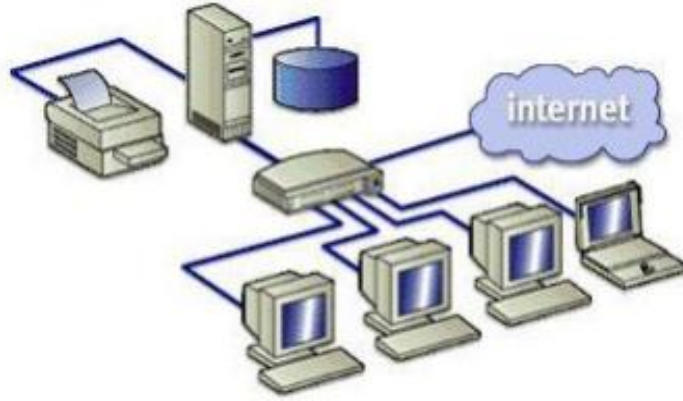
**Keywords: Graph Theory, Network Design, Connectivity, Vulnerability, Average binding number, Algorithm.**

**2021, 40 pages**

# 1. GİRİŞ

## 1.1 Çizge Teorisi ve Uygulama Alanları

Günümüz şartlarında herhangi bir bilginin ya da verinin bir yerden başka bir yere hızlı bir şekilde aktarma isteğinin önemi giderek artmaktadır. Bununla beraber iletişimin güvenilir, kesintisiz ve hızlı olması beklenir. Bununla birlikte iletişim ağlarının kurulum öncesi ve kurulum aşamasındaki önemi de artmaktadır. Ağlar, iletişim merkezlerinin ve bu merkezlerin birbirleriyle bağlantı sağlayan hatlardan meydana gelir.



Şekil 1.1. Bilgisayar ağı

Bilgisayar ağları gibi herhangi bir ağ, iletim yollarıyla bağlanan bir dizi merkezdir. Çizge teorisi yardımıyla çözülen en önemli sorunlardan biri, kesintilere karşı direnci diğer ağlara göre daha fazla olan bir ağ modeli tasarlamaktır. Çizgeler genellikle bir iletişim, bilgisayar veya casus ağı gibi gerçek dünya sorunlarını modellemek için kullanılır. Bir ağda, güvenlik açığı parametreleri, ağın, belirli istasyonların veya bağlantıların arızalanmasından sonra operasyonun kesintiye uğramasındaki direncini ölçer.

Bu ağlar, merkezleri tepelere karşılık gelen ve ayrıtlara olan bağlantıları ile çizgelerle modellenmiştir. Ağlar tasarlanırken, güvenlik açıkları ölçülebilir ve çizgelerle modellendiğinde maliyet hesaplanabilir. Ağ tasarımı aşamasında güvenlik zafiyetlerini tespit etmek ve önlemek, ayrıca ağ maliyetini de dikkate alarak daha

ekonomik ve güvenli ağlar tasarlamak amacıyla geliştirilmiş bazı zedelenebilirlik parametreleri bulunmaktadır. Bir ağın bağlantı hatları hasar gördüğünde iletişim sorunları ortaya çıkar.

Çizge teorisi, bir problemin çözülmesi için tepe ve ayrıt ile model oluşturulup, bu modellemenin çizge olarak gösterilmesi prensibine dayanmaktadır. Çizge teoride tanımlı olan özellikler, modelin çözümünü kolaylaştır ve dolayısıyla gerçek problemin çözümüne de daha kolay gidilmesini sağlar. Açıkçası, gerçek dünyadan bir problem çizge olarak modellenir, bu model çözülür ve gerçek dünyaya uygulanmasıyla sağlanır. Ancak bu şekilde bir modelleme çizge teorisinin işe yaradığını gösterir. Çizge teoride modelleme yapılırken bazı bağlantı hatlarına hasar verecek koşullara karşı nasıl dayandığı ve ne kadar dayanıklı kalacağı önemli bir sorun teşkil etmektedir.

Bir çizgenin  $deg(v)$  ile gösterilen  $v$  derecesi, açık komşuluğunun esas niteliğidir. Bir  $G$  çizgesindeki 1. dereceden bir tepe noktası o çizgenin yaprağı, sarkıtı veya ayrıtı olarak adlandırılır ve bu noktanın komşulukları onun destek tepeleridir. Döngüsü olmayan bir çizge döngüsel değildir. Bir  $G$  çizgesinde yaprağa bağlı ayrıtı sarkık ayrıtı denir. Bir ağa, bağlantılı döngüsel olmayan bir çizgedir.

Bir  $G=(V,E)$  çizgesinde; boş olmayan tepeler ( $V$ ) ve ayrıtlar ( $E$ ) kümesinden doğrusal olmayan tepe ve ayrıtları birleştirerek oluşur. Bir  $G$  çizgesi için açık komşuluk değeri ( $u$ ) ;  $N(u) = \{v \in V(G) | (u,v) \in E(G)\}$  ve kapalı komşuluk değeri ( $u$ );  $N[u]=\{u\} \cup N(u)$  dur. Benzer şekilde; herhangi bir  $S \subseteq V(G)$  alt kümesi için, açık komşuluk değeri  $N(S)= \bigcup_{u \in S} N(u)$  ve kapalı komşuluk değeri  $N[S] = \bigcup_{u \in S} N[u]$  dur. Minimum tepe sayısı, belirli bir çizge için mümkün olan en az sayıda tepeye sahip bir tepe sayısıdır. Bir  $G$  çizgesindeki en küçük tepe sayısının değeri tepe sayısı olarak adlandırılır ve  $\alpha(G)$  olarak gösterilir. Genellikle basitçe "bağımsızlık sayısı" olarak adlandırılan bir çizgenin tepe bağımsızlık sayısı, en büyük bağımsız tepe kümesinin temelidir, yani maksimum bağımsız tepe kümesinin boyutudur.

Bağımsızlık sayısı en yaygın olarak  $\beta(G)$  ile gösterilir.  $S$  seçilen tepe sayısını,  $N(S)$  ise seçilen tepelerin komşu sayılarını tanımlar.

## 1.2. Zedelenebilirlik

Bir ağ merkezler tepelere ve bağlantılar ayrıtlara denk gelecek şekilde *G* çizgesi ile modellenebilir. Bir iletişim ağında belli merkezlerin ya da bağlantıların zarar görmesi ile iletişim bağlantılarının kopması ve bağlantıların kesilmesine kadar ağların bu süre içerisindeki dayanma gücünün ölçüsüne zedelenebilirlik denir.

İletişim ağında zedelenebilirlik, ağ merkezleri veya merkezler arasındaki hatların bağlantısını sağlayan ve hatlarda meydana gelen herhangi bir zarar, hasar ya da kopma neticesinde ağların dayanma gücünü temsil eden ölçümlerdir. Bir iletişim ağında kesintisiz iletişim ve veri aktarımının devamlılığı söz konusu olduğu düşünülürse, ağ tasarlarırken herhangi bir hasardan sonra ağın ne durumda olacağı, ne kadarının hasar alacağı ya da yeniden eski haline getirilmesinin maliyetinin ne olacağı gibi araştırmalar yapılmalıdır. Örnek olarak bir ülkenin şehirlerarası ulaşım ağını düşünelim. Bu durumda, şehir merkezlerindeki duraklar (otogarlar) ağın merkezlerini, duraklar arasındaki yollar da merkezler arasındaki hatları oluşturur. Herhangi bir sebepten oluşacak duraklar ve/veya bazı yollar zarar gördüğünde, oluşan hasarı araştırırken; “Zarar gören durakların sayısı nedir?”, “Zarar gören yolların sayısı nedir?”, “Zarar gören duraklardan ve yollardan sonra aralarından ulaşım devam eden durakların sayısı nedir?”, “En fazla kopukluk olmadan devam eden durakların sayısı nedir?” gibi birçok soru akla gelebilir. Bu ancak ağın gördüğü zararın büyüklüğünün ölçümü veya tasarım aşamasındaki ağın ne kadar dayanıklı olduğunun analizi ile sağlanmalıdır. Dayanıklılık analizleri, ağ tasarımı yapılırken yapılırsa, amaca en uygun ağ modeli, tasarım aşamasındayken seçilebilir (Boruzanlı, 2010).

Yaşadığımız teknoloji çağında, bir iletişim ağı, bağlantı hatlarını veya merkezlerini kaybetmeye başlarsa bazı aksaklıkların ya da gecikmelerin yaşanması, hatta büyük zaman kayıplarının ortaya çıkması kaçınılmaz olur. Örneğin bir havaalanındaki kulenin, iniş yapmak isteyen uçakların bazıları ile bağlantısı kesilirse, kule ile iletişim kurmadan inmeye çalışan bir uçak pistte bazı kazalara neden olabilir. Bu durumda, pistte oluşan zararın ve bu kazanın neden olabileceği başka zararların (diğer seferlerin aksaması gibi) araştırılması için kaza sonrasında geriye kalan yapının incelenmesi gereklidir. Bu incelemenin sonucunda uğranılan zararın ne boyutta olduğu söylenebilir. Örneğin havaalanındaki pist tümüyle bozulmuş olabilir ya da pistin bir kısmı sağlam olup, diğer seferleri sağlam pist üzerinden yapabiliriz. Bu durum ise

uğradığımız zararı azaltabilir. Bu nedenle, iletişim ağında oluşabilecek zararı araştırmak için, iletişim ağındaki bozulmanın ardından geriye kalan ağın durumu incelenmelidir. Günümüzde iletişim ve iletişimi sağlayan ağların güvenilirliği ve zedelenebilirliği yukarıda belirtilenlere benzer nedenlerden dolayı büyük öneme sahiptir. İletişim ağları; merkezleri çizgenin tepelerini, merkezler arasındaki bağlantı hatları da çizgenin ayrıtları olacak şekilde çizgeler modellenbildiği için, iletişim ağlarının zedelenebilirlik değerlerini araştırmak için çizgelerde birçok zedelenebilirlik parametrelerin tanımları bulunmaktadır (Boruzanlı, 2010). İletişim ağlarının güvenilirliği ve zedelenebilirliği konusunda çizgeler bir model olarak ele alınır. İletişim ağını oluşturan her bir merkez çizgenin bir tepesi ve bu merkezler arasındaki bağlantı hatları da çizgenin ayrıtları olarak adlandırılır.

Zedelenebilirlik ölçümleri, bir ağda iletişim kesilene kadar ağın ne kadar ve nasıl süreceği sorusuna cevap vermek için yapılır. Kısacası, zedelenebilirlik, bir ağın iletişimi sona erene kadar gösterdiği direncin bir ölçüsüdür. Ağın gördüğü zararın büyüklüğünün hesaplanması veya tasarım aşamasındaki ağın dayanıklılığının analizi mümkün olmaktadır. Bu ölçümler ile tasarlanan çizge yapısının ne kadar dayanıklı, hassas ve dağınık olduğunu belirleyebiliriz. Çizge modelinin ağ tasarımı yapılırken, en dayanıklı ağ modeli seçilerek analiz yapılmalıdır. İletişim ağları; çizgenin tepelerini, merkezleri; çizgenin ayrıtlarını da merkezler arasındaki bağlantı hatlarını gösterir. İletişim ağları çizgenin tepeleri ve ayrıtları olacak şekilde modellenir ve zedelenebilirlik parametrelerinden yararlanır. Bu zedelenebilirlik ölçüm parametrelerinden bazıları aşağıdaki gibidir; bağlanabilirlik [24], saçılma sayısı [21], bağlama sayısı [33] ve ortalama bağlama sayısı [1]. Ortalama parametrelerin bazı durumlarda karşılık gelen en kötü duruma dayalı ölçümlerden daha faydalı olduğu bulunmuştur [4].

Bu çalışmamızda amacımız, ortalama bağlama sayısı için algoritma oluşturmaktır ve ortalama bağlama sayısı için aynı güvenlik açığı parametreleri dikkate alındı. Bu nedenle, bu güvenlik açığı parametrelerinin algoritması geliştirilerek, ortalama bağlama sayısı algoritmasını kod ölçümleri ile analiz edildi. Algoritmayı geliştirmek için Python dili kullanılmıştır. Bu şekilde, kod ölçümlerini analiz etmek için Radon [27] kütüphanesi kullanıldı. Radon [27] siklomatik karmaşıklığı, sürdürülebilirlik indeksini, ham ölçümleri ve Halstead ölçümlerini

hesaplayabilen bir kütüphanedir.

Siklomatik Karmaşıklık, bir kod bloğunun içerdiği seçim sayısı artı 1'e karşılık gelir. Bu sayı (ayrıca McCabe sayısı olarak da bilinen), kod aracılığıyla doğrusal olarak tarafsız yolların sayısı ile aynıdır. Bu sayı, bloklarda koşullu iyi muhakemeyi kontrol ederken bir kılavuz olarak kullanılabilir.

Sürdürülebilirlik İndeksi, farklı metriklerin bazı değerlerini kullanarak ölçen bir yazılım metriğidir. Ham metrikler, kod ölçümlerini hesaplamak için kod numaralarının satırını, yorum satırlarının sayısını, çok satırlı dizeleri ve boş satır sayısını kullanır. Halstead'in amacı, kodun ölçülebilir özelliklerini ve bunların ilişkilerini öğrenmektir. Kaynak koddan hesaplanan statik sayılar; farklı operand ve operatörlerin sayısı ve bunların toplam sayısı [11,13,22,25]. Bölüm 3'te, ortalama bağlama sayısı tanımı, temel teoremler ve bazı özel çizgeler elde edilmiştir. 4. bölümde Yazılım Metrikleri elde edilmiştir. Bölüm 5'te Algoritmalar elde edilmiştir. 6. bölümde Test sonuçları ve Karmaşıklık elde edilmiştir.

## 2. GENEL TANIMLAR

Bu bölümde, tezde kullanılmış olan gerekli tanımlar verilmiştir.

**Tanım 2.1.** [17] Herhangi bir  $G$  çizgesinin bir  $v$  tepesine bağlı ayrıtlarının sayısına, o tepenin derecesi denir ve  $\deg(v)$  ile gösterilir. Bir tepenin derecesi sıfır ise o tepeye izole tepe denir.

**Tanım 2.2.** [17] Bir  $G$  çizgesinin tepe dereceleri içinde en küçüğüne minimum tepe derecesi denir ve  $\delta(G)$  ile gösterilir.

**Tanım 2.3.** [17] Bir  $G$  çizgesinin tepe dereceleri içinde en büyüğüne maximum tepe derecesi denir ve  $\Delta(G)$  ile gösterilir.

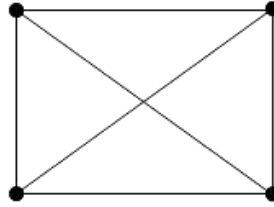
**Tanım 2.4.** [11]  $V(G)$ ,  $G$  çizgesinin tepeler kümesi olmak üzere,  $V(G)$  kümesinde birbirine komşu olmayan maksimum tepe sayısına bağımsızlık sayısı denir ve  $\alpha(G)$  ile gösterilir.



**Tanım 2.5.** [17] Bir  $G$  çizgesinin bazı tepelerini veya ayrıtlarını çizgeden çıkardığımızda bileşen sayısı artıyor veya sabit kalıyorsa atılan tepelerin veya ayrıtların kümesine kesim kümesi denir ve  $|S|$  ile gösterilir.

**Tanım 2.6.** [17] Bir  $G$  çizgesinin, her bir tepesinden çizgenin diğer tüm tepelerine ulaşılabilirse bu çizgeye bağlantılı (connected) çizge denir.

**Tanım 2.7.** [17] Bir çizgenin her bir tepesi, çizgenin diğer tepelerine komşu ise bu çizgeye tam çizge (complete graph) denir ve  $n$  tepeli bir tam çizge  $K_n$  ile gösterilir. Şekil 2.1'de  $K_4$  tam çizgesi gösterilmiştir.



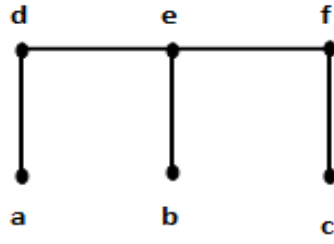
**Şekil 2.1.** Tam Çizge ( $K_4$ )

**Tanım 2.8.** [34] Başlangıç ve bitiş tepeleri tek dereceli olup diğer tepeleri iki dereceli olan çizgeye yol çizge (path graph) denir ve  $n$  bir tam sayı olmak üzere  $P_n$  şeklinde gösterilir. Şekil 2.2'de  $P_5$  yol çizgesi gösterilmiştir.



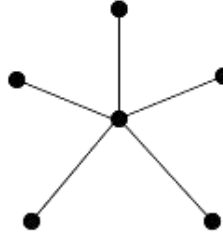
**Şekil 2.2.** Yol Çizge ( $P_5$ )

**Tanım 2.9.** Dikenli Çizge (Thorn Graph);  $p_1, p_2, \dots, p_n$  negatif olmayan tamsayılar ve  $G, V(G) = n$  olan bir çizge olsun.  $p_1, p_2, \dots, p_n$  olan diken çizgesi 1. dereceden tepe noktalarının  $G$  çizgesinin  $u_i$  tepe noktasına,  $i = 1, 2, \dots, n$ , eklenmesi ile elde edilir.  $G$  çizgesinin diken çizgesi  $G^*$  veya sıralı parametlerinin tanımlanması gerektiğinde  $G^*(p_1, p_2, \dots, p_n)$  şeklinde ifade edilecektir.



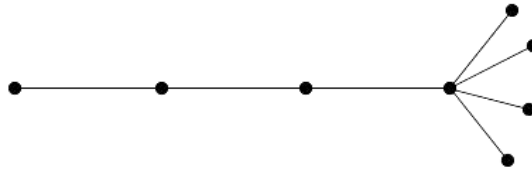
Şekil 2.3. Dikenli Çizge (Thorn Graph)

**Tanım 2.10.** [17]  $n+1$  tepeli bir çizgenin  $n$  tane tepesinin derecesi bir ve merkezindeki tepesinin derecesi  $n$  olmak üzere  $n+1$  tepeli çizgeye yıldız (star) çizge denir ve  $K_{1,n}$  ile gösterilir. Şekil 2.4’de  $K_{1,5}$  yıldız çizgesi gösterilmiştir.



Şekil 2.4. Yıldız Çizge ( $K_{1,5}$ )

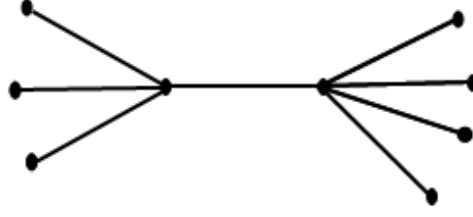
**Tanım 2.11.** [9]  $K_{1,r}$  yıldız çizgenin merkez tepesine,  $n-1$  tepeli  $P_{n-1}$  yol çizgesinin eklenmesi ile oluşan çizgeye kuyruklu yıldız (Comet Graph) çizgesi denir ve  $C_{n,m}$  ile gösterilir. Şekil 2.5’de  $C_{4,4}$  kuyruklu yıldız çizgesi gösterilmiştir.



Şekil 2.5. Kuyruklu Yıldız ( $C_{4,4}$ )

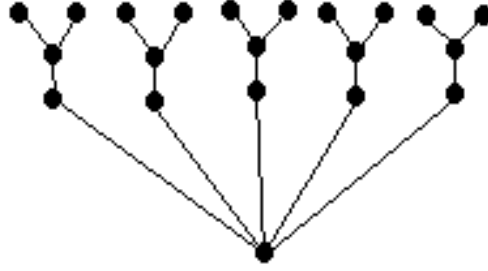
**Tanım 2.12.** Çift Yıldız (Double star)  $K_{r,m}$  ; çapı 3 olan ve sırasıyla  $r$  ve  $m$  dereceye sahip merkezi tepeleri olan bir ağaçtır, burada çizgenin çapı, en uzak tepeler arasındaki

en kısa yolun uzunluğudur.



Şekil 2.6. Double Star ( $K_{4,5}$ )

**Tanım 2.13.**  $B(r,m)$  ile tanımlanan Muz ağacı (Banana Tree) çizgesi, yıldız çizgesinin  $r$  kopyasının her birinin bir yaprağını tüm yıldızlardan farklı olan tek bir kök tepe noktasına yıldızın en yakın düğümden bağlanarak elde edilen bir çizgedir.



Şekil 2.7. Muz Ağacı Çizgesi ( $B_{5,4}$ )

**Tanım 2.14.** [1]  $\alpha(G)$  ile tanımlanan Kaplama sayısı (Covering number),  $G$  çizgesinin minimum tepe kaplamasının boyutu tepe kaplama sayısı olarak bilinir ve  $\alpha(G)$  gösterilir.

**Tanım 2.15.** [1]  $\beta(G)$  ile tanımlanan Bağımsızlık sayısı (Independence number),  $G$  çizgesinin maksimum tepe sayısı olarak bilinir ve  $\beta(G)$  ile gösterilir.

**Tanım 2.16.** [30] Bağlantılık sayısı  $K(G)$ , bağlantısız bir çizge elde etmek için veya tek bir izole tepe kalana kadar çıkarılması gereken minimum tepe sayısı olarak tanımlanır.

**Tanım 2.17.** [19] Bir  $G$  çizgesinin saçılım sayısı (scattering)  $sc(G)$  ile gösterili ve,  $sc(G) = \max\{ \omega(G-S) - |S| : S \subseteq V(G) \text{ ve } \omega(G-S) > 1 \}$  burada  $S$  tepe kümesi,  $\omega(G-S)$ ;  $G-S$ ,  $G$  çizgesinde geriye kalan bileşenlerinin sayısını gösterir.

### 3. ORTALAMA BAĞLAMA SAYISI (AVERAGE BINDING NUMBER)

Bu bölümde, ortalama bağlama sayısının tanımı, bilinen ve hesaplanan bazı teoremler verilmiştir. Daha sonra bazı özel çizgelerin ve çizge işlemlerinin ortalama bağlama sayısı hesaplanmıştır.

#### 3.1. Tanım ve Teoremler

Aşağıda ortalama bağlama sayısını vermiştir.

**Tanım 3.1.1.** [1]  $v \in V(G)$  için,  $v$  tepesi için yerel bağlama sayısı  $bind_v(G)$  olarak gösterilir, ve  $bind_v(G) = \min_{S \in F_v(G)} \left\{ \frac{|N(S)|}{|S|} \right\}$ , burada  $F_v(G) = \{S \subseteq V(G) | v \in S, S \neq \emptyset, N(S) \neq V(G)\}$ .

$G$  çizgesinin ortalama bağlama sayısı  $bind_{av}(G)$  şu şekilde tanımlanır,  $bind_{av}(G) = \frac{1}{n} \sum_{v \in V(G)} bind_v(G)$ , burada  $n$ ,  $G$  çizgesindeki tepe sayısıdır. Ortalama bağlama sayısı ile ilgili Aslan [1] tarafından verilen bazı sonuçlara yer verilmiştir.

**Teorem 3.1.2.** [1]  $G$ , minimum derece (minimum degree)  $\delta(G)$  ye sahip  $n$  tepeli bir çizge ise;

$$bind_{av}(G) \geq \frac{\delta(G)}{n - \delta(G)}.$$

**Teorem 3.1.3.** [1]  $G$ , kaplama sayısı (covering number)  $\alpha(G)$  ve bağımsızlık sayısı (independence number)  $\beta(G)$  ile  $n$  tepeli bir çizge ise;

$$bind_{av}(G) \leq \frac{\alpha(G)(\beta(G) + n - 1)}{n\beta(G)}.$$

**Teorem 3.1.4.** [1]  $G$ , minimum dereceli  $\delta(G)$  ve maksimum dereceye  $\Delta(G)$  sahip  $n$  tepeli bir çizge ise;

$$bind_{av}(G) \geq \frac{\delta(G)}{\Delta(G) + 1}.$$

**Teorem 3.1.5.** [1]  $H$ ,  $G$ 'nin bir alt çizgesi (spanning subgraph) ise;

$$bind_{av}(H) \leq bind_{av}(G).$$

**Teorem 3.1.6.** [1]  $G$ ,  $n$  tepeli bir çizge ise;

$$bind(G) \leq bind_{av}(G).$$

Bu tez kapsamında elde edilen, Ortalama bağlama sayısının ve diğer çizge parametrelerinin ilgili teoremleri aşağıda verilmiştir.

**Teorem 3.1.7.**  $G$ ,  $n$  derecesinin bir çizgesi ise,

$$bind_{av}(G) \leq n - 1.$$

*İspat.*  $v \in V(G)$  ve  $S_v$ ,  $v$ 'de yerel bağlama sayısı olsun. Kesinlikle  $|S_v| \geq 1$  ve  $|N(S_v)| \leq n - 1$  olur.

Böylece,

$S_v^*$  olduğundan dolayı

$$bind_{av}(G) = \frac{1}{n} \sum_{v \in V(G)} bind_v(G) \leq n - 1.$$

olarak ifade edilir.

**Teorem 3.1.8.**  $G$ ,  $n$  dereceli bir çizge ise,

$$bind_{av}(G) \leq bind_{av}(G + e).$$

*İspat.*  $v \in V(G)$  ve  $S_v$  ve  $S_v^*$ ,  $V(G)$ 'de yerel bağlama sayısı olsun ve  $V(G+e)$ , sırasıyla.

Açıkça görülüyor ki,

$$|S_v| = |S_v^*| \text{ ve } |N(S_v)| \leq |N(S_v^*)|.$$

Buradan,

$$bind_v(G) = \frac{|N(S_v)|}{|S_v|} \leq \frac{|N(S_v^*)|}{|S_v^*|} = bind_v(G + e)$$

Böylece,

$$bind_{av}(G) \leq bind_{av}(G + e).$$

olarak tanımlanır.

**Teorem 3.1.9.**[19]  $G$ ,  $n$  dereceli bir çizge olsun.

$$2\alpha(G) - n \leq sc(G) \leq \alpha(G) - K(G).$$

**Teorem 3.1.10.**  $G$ , saçılım sayısı (scattering)  $sc(G)$  ve bağımsızlık sayısı olan  $n$  dereceli bir çizge ise;

$$bind_{av}(G) \leq sc(G) + 2\beta(G) - 1.$$

*İspat.* Teorem 3.1.6'ya göre,

$$bind_{av}(G) \leq n - 1.$$

Biliyoruz ki,  $n = \alpha(G) + \beta(G)$ . Bu nedenle,

$$bind_{av}(G) \leq \alpha(G) + \beta(G) - 1.$$

$$\text{bind}_{av}(G) - 2\beta(G) + 1 \leq \alpha(G) - \beta(G)$$

*İspat.* Teorem 3.1.8' e göre,

$$\text{bind}_{av}(G) - 2\beta(G) + 1 \leq \alpha(G) - \beta(G) \leq \text{sc}(G)$$

$$\text{bind}_{av}(G) \leq \text{sc}(G) + 2\beta(G) - 1.$$

**Teorem 3.1.11.**  $G$ , Bağlantı sayısı (connectivity number)  $\kappa(G)$ , kaplama sayısı (covering number)  $\alpha(G)$ , bağımsızlık sayısı (independence number)  $\beta(G)$  olan  $n$  dereceli bir çizge ise;

$$\text{bind}_{av}(G) \leq \alpha(G) - K(G) + 2\beta(G) - 1.$$

*İspat.* Teorem 3.1.8 ve 3.1.9'a göre

$$\text{bind}_{av}(G) \leq \alpha(G) - K(G) + 2\beta(G) - 1.$$

### 3.2.Bazı Özel Çizgelerin Ortalama Bağlama Sayısı

Bu kısımda, bazı özel çizgelerin ortalama bağlama sayısı verilmiştir. Aşağıda Muz Ağacı, Kuyruklu Yıldız, Çift Yıldız ve Dikenli Çizge gibi özel çizgeler için teoremler verilecektir.

**Teorem 3.2.1.** Eğer  $B_{r,m}$   $r \geq 2$  ve  $m \geq 2$  olan bir Muz ağacı çizgesi ise,

$$\text{bind}_{av}(B_{r,m}) = \left( \frac{r * m + (r + 1)^2}{(r * m + 1) * (r * m - r + 1)} \right) * \frac{1}{(r * m) - 1}$$

*İspat.*  $v \in V(B_{r,m})$  ise,  $S_v \in F_v(B_{r,m})$  olur. Yıldızın merkez tepe hariç tüm tepelerin yerel bağlama sayısı  $|S_v| = r * m - r$  dir.

Muz ağacında bulunan yıldızların merkez tepe hariç tüm tepelerinin ( $S_v$ ) çizgedeki tüm tepelere olan komşulukları  $|N(S_v)| = r * m + 1$  olur. Bu durumda

$$\text{bind}_v(B_{r,m}) = \frac{r*m+1}{r*m-r} \text{ olur.}$$

Eğer  $\deg(v)=r$  ise  $V(B_{r,m})$ 'nin  $S_v$  olan yerel bağlama sayısı vardır. Öyle ki  $|S_v| = r*m-r+1$  ve  $|N(S_v)|=r+1$  olur. Böylece  $\text{bind}_v(K_{r,m}) = \frac{r+m}{r*m-r+1}$ .

Eğer  $\deg(v)=m$  ise  $(B_{r,m})$ 'nin  $S_v$  olan yerel bağlama sayısı vardır. Öyle ki  $|S_v| = r*m-r+1$  ve  $|N(S_v)|=2r+1$ . Böylece  $\text{bind}_v(K_{r,m}) = \frac{2r+1}{r*m-r+1}$ .

Tanımdan yola çıkarak;

$$\text{bind}_{av}(B_{r,m}) = \frac{1}{r*m+1} \sum_{v \in V(B_{r,m})} \text{bind}_v(B_{r,m}) = \frac{\frac{r*m+1}{r*m-r} * r + \frac{r+m}{r*m-r+1} * r + \frac{2r+1}{r*m-r+1}}{r*m+1}$$

$$\text{bind}_{av}(B_{r,m}) = \left( \frac{r*m+(r+1)^2}{(r*m+1)*(r*m-r+1)} \right) * \frac{1}{(r*m)-1}$$

**Teorem 3.2.2.** Eğer  $C_{t,r}$   $r \geq 2$  ve  $t \geq 2$  olan bir kuyruklu yıldız çizge ise, o zaman

$$\text{bind}_{av}(C_{t,r}) = \frac{2t + 2r + 1}{(r + t)(r + 1)}$$

olur.

*İspat.*  $P_t$ 'nin tepeleri yol boyunca  $p_1, p_2, \dots, p_t$ ,  $\deg(p_1)=1$  ve  $\deg(p_t)=r+1$  olsun.  $v \in V(C_{t,r})$ ,  $S_v \in F_v(C_{t,r})$  iken iki duruma ayrılır.

*Durum 1:*  $V(C_{t,r})$ 'nin  $|S_v| = r$  olan bir yerel bağlama kümesi vardır.  $S_v, C_{t,r}$ 'nin tüm ayrılmış tepe noktaları kümesini kapsadığında  $|N(S_v)|=1$  olur.

*Durum 2:*  $v \in V(P_t)$  olduğunu varsayalım.

•  $v = \{p_1\}$  veya  $v = \{p_t\}$  olduğunda;  $|S_v| = r+1$  ve  $|N(S_v)|=2$  elde edilir.

Böylece iki tepe noktası için,  $\text{bind}_v(C_{t,r}) = \frac{2}{r+1}$  olur.



•  $v=\{p_t\}$  olduğunda;  $|S_v| = r+1$  ve  $|N(S_v)| = r+2$  elde edilir. Böylece;  $bind_v(C_{t,r}) = \frac{r+2}{r+1}$  olur.

•  $v \in \{p_2, p_3, \dots, p_{t-2}\}$  olduğunda,  $V(C_{t,r})$  'nin  $|S_v| = r+1$  ve  $|N(S_v)|=3$  olan bir yerel bağlama kümesi vardır ve  $bind_v(C_{t,r}) = \frac{3}{r+1}$  elde edilir.

Tanımdan yola çıkarak;

$$bind_{av}(C_{t,r}) = \frac{1}{r+t} \sum_{v \in V(C_{t,r})} bind_v(C_{t,r}) = \frac{\frac{1}{r}(r) + \frac{2}{r+1}(2) + \frac{r+2}{r+1} + \frac{2}{r+1}(t-3)}{r+t}$$

$$bind_{av}(C_{t,r}) = \frac{2t + 2r + 1}{(r+t)(r+1)}$$

**Teorem 3.2.3.** *Double Star*  $(K_{r,m})$   $r \geq 2$  ve  $m \geq 1$ , ise;

$$bind_{av}(K_{r,m}) = \frac{3}{r+m-1}$$

*İspat.*  $v \in V(K_{r,m})$ ,  $S_v \in F_v(K_{r,m})$  olsun.  $V(K_{r,m})$ 'nin  $|S_v| = r+m-2$  olan bir yerel bağlama kümesi vardır.  $S_v$ ,  $K_{r,m}$ ' nin tüm ayrılmış tepe noktaları kümesini kapsadığında  $|N(S_v)|=2$  olur.

Böylece  $K_{r,m}$ 'nin  $r+m-2$  tepeleri için;

$$bind_v(K_{r,m}) = \frac{2}{r+m-2}$$

olur.

Eğer  $deg(v)=r$  ise;  $V(K_{r,m})$ 'nin  $|S_v| = r+m-1$  olan bir yerel bağlama kümesi vardır ve  $|N(S_v)|=r+1$ 'dir. Böylece,  $bind_v(K_{r,m}) = \frac{r+1}{r+m-1}$ .

Eğer  $\deg(v)=m$  ise;  $V(K_{r,m})$ 'nin  $|S_v| = r+m-1$  olan bir yerel bağlama kümesi vardır ve  $|N(S_v)|=m+1$ 'dir. Böylece,  $\text{bind}_v(K_{r,m}) = \frac{m+1}{r+m-1}$ .

Tanımdan yola çıkarak;

$$\text{bind}_{av}(K_{r,m}) = \frac{1}{r+m} \sum_{v \in V(K_{r,m})} \text{bind}_v(K_{r,m}) = \frac{\frac{2}{r+m-2}(r+m-2) + \frac{r+1}{r+m-1} + \frac{m+1}{r+m-1}}{r+m}$$

$$\text{bind}_{av}(K_{r,m}) = \frac{3}{r+m-1}$$

elde edilir.

**Teorem 3.2.4.**  $P_n^*$  her  $p_i=1$  olan bir dikenli çizgedir ( $P_n$ ). O zaman

$$\text{bind}_{av}P_n^* = 1.$$

*İspat.* Yerel bağlayıcı kümesinin kardinalitesi  $\text{bind}_v(P_n^*)$  olarak tanımlanır,  $P_n^*$  'nin her tepe noktası için her zaman aynıdır ve 1'e eşittir. O zaman;

$$\text{bind}_{av}P_n^* = 1.$$

**Teorem 3.2.5.**  $P_n^*$ ;  $P_n$  yol çizgesinin  $p_i=k$  ve  $k > 1$  olan bir dikenli çizge olsun. O zaman;

$$\text{bind}_{av}(P_n^*) = \frac{(k+1)n^2 + n + kn}{(nk+k)(nk+1)}.$$

*İspat.*  $v \in V(P_n^*)$ ,  $S_v \in F_v(P_n^*)$  olsun.

Eğer  $\deg(v)=1$  ise  $V(P_n^*)$ 'nin yerel bağlama sayısı vardır. Öyle ki,  $|S_v| = nk$  ve  $|N(S_v)|=n$  olur. Böylece  $P_n^*$  'nin  $nk$  tepe noktaları için  $\text{bind}_v(P_n^*) = \frac{n}{nk} = \frac{1}{k}$  bulunur.

Eğer  $\deg(v)>1$  ise  $V(P_n^*)$  nin  $S_v$ , yerel bağlama sayısı vardır. Öyle ki,  $|S_v|$

$=nk+1$  ve  $|N(S_v)|=n+k$  bulunur. Böylece  $P_n^*$  nin  $n$  tepe noktaları için  $bind_v(P_n^*) = \frac{n+k}{nk+1}$  olur .

Tanımdan yola çıkarak;

$$bind_{av}(P_n^*) = \frac{1}{(n+1)k} \sum_{v \in V(P_n^*)} bind_v(P_n^*) = \frac{\frac{1}{k}(nk) + \frac{n+k}{nk+1}(n)}{(n+1)k}$$

$$bind_{av}(P_n^*) = \frac{(k+1)n^2 + n + kn}{(nk+k)(nk+1)}$$

#### 4.YAZILIM METRİKLERİ (SOFTWARE METRICS)

Alniak (2017)' a göre Yazılım metrikleri, geliştirilen yazılımların ölçülebilmesi ya da yapılan ölçümler sonucu hesaplanan değerleridir. Bilgi elde etmek için yazılım metriklerini hesaplayan araçlar bulunmaktadır. Bu araçlar yazılım metriklerini otomatik olarak hesaplar. Yazılım projeleri milyonlarca kod satırından oluşmaktadır. Bu kapsamlı projeler hakkında detaylı bilgilere yazılım metrikleri araçlarıyla ulaşılabilir. Boyut, çaba, süre, maliyet ve kalite dahil olmak üzere beş temel yazılım ölçütü vardır. Bu ölçütler, karmaşıklık, yeniden kullanılabilirlik, test edilebilirlik, etkinlik / verimlilik, dayanıklılık, anlaşılabilirlik gibi yazılımların kalite seviyelerini ölçmek için kullanılabilir. Kısaca tüm bu ölçütler yazılımların kalitesini ölçmek için kullanılabilir. Yazılım kalite ölçümlerini yapabilmek için birçok metrik ölçüm kümeleri sayılabilmektedir. En çok üzerinde durulan metrik ölçüm kümeleri: Chidamber&Kemerer, Brito e Abreu ve Bansiya & Davis gibi metrik kümeleri verilebilir. Burada yapılan tanımlamalar ve açıklamalar furkanalniak 'ın web sitesinde bulunan kaynaktan yararlanılmıştır [<http://furkanalniak.com/yazilim-muhendisligi-yazilim-kalite-metrikleri/>].

Kabakçı (2015)' ya göre McCabe Çevrimsel Karmaşıklığı(McCabe Cyclomatic Complexity) ise, Fonksiyonel-yazılım karmaşıklık ölçüm yöntemlerinden bir diğeridir. Amacı, güvenilir olmayan ve test edilmesinde zorluk yaşanan program

modüllerini açıklamayı amaçlar. Temelde kontrol akış çizgesi(control flow graph) kullanarak ilgili modülün karmaşıklığını kısaca CC olarak adlandırılan bu yöntem ölçmeye çalışır. Bu ölçümleri çizge üzerinde bulunan tepe ve ayrıtların durumuna göre yapar. Çizge üzerinde bulunan her bir tepe, program kodunda yer alan mantıksal(logic) ifadelerine göre oluşturulur. Mantıksal ifadeler while, for, if, switch ve goto gibi ifadeleri içerir. Her mantıksal deyim, kontrol çizgesinde bulunan tepelere dönüştürerek, mantıksal deyimler ayrıtlar ile birbiri arasında bağlantı sağlanır. Program kodunda bulunan dallanmalar, kontrol çizgesi üzerindeki ayrıtların yönünün belirlemesini sağlar. Tamamlanmış çizgenin, toplam tepe ve ayrıt sayıları hesaplanarak aşağıdaki bağıntıdan yararlanır[16].

$$V(G) = E - N + 2 * P \quad (CC - Cyclomatic Complexity) \quad (1)$$

*E: Toplam ayrıt sayısı (number of edge)*

*N: Toplam tepe sayısı (number of node)*

*P: Modül(parça) sayısı (number of module, equals to 1 as initial)*

Kabakçı (2015)' ya göre Program kodunda hata sayısını gösteren karmaşıklık değerini Siklomatik karmaşıklık modeli (Cyclomatic complexity model) hesaplar. Hata ise, program kodunun ne kadar riskli olduğunu gösterir[16].

$$CommentRate = CommentStatements/TotalStatements \quad (2)$$

Kabakçı (2015)' ya göre Denklem 2' de gösterildiği gibi, Yorum Oranı hesaplayabilmek için, yorum ifadeleri sayısının toplam ifade sayısına bölünmesiyle hesaplanır[16].

Kabakçı (2015)' ya göre Program kodunun ne kadar riskli olduğunu hesaplayabilmek için Denklem 1'de bulunan formülden yararlanır. Kodun riskli olması, program kodunun hataya ne derecede eğilimli olduğu anlamına gelir. Aşağıdaki tabloda verilen değerler, Yazılım Mühendisliği Enstitüsü (Software Engineering Institute - SEI) tarafından kabul edilen risk değerlendirmesi verilmiştir. Bağıntıdan elde edilen değer tablo 1 ile ilişkilendirilerek değerlendirilir [16].

**Tablo 1. Risk Değerlendirmesi**

<b>Siklomatik Karmaşıklık</b>	<b>Risk Değerlendirmesi</b>
1-10	Fazla riskli olmayan basit bir modül
11-20	Orta riskli biraz daha karmaşık modül
21-50	Yüksek riskli karmaşık bir modül
51 -	Çok riskli test edilemez bir modül

Kabakçı (2015)' ya göre Yukarıdaki tablo 1' e göre, McCabe CC değerlerine göre risk değerlendirme gösterilmektedir. 1 ila 10 arasındaki değerler fazla riskli olmayan basit, 51 den yukarısında çıkan CC değeri, bir program kodunun çok riskli test edilemez durumda olduğunu ve alarm verdiğini ifade eder. CC değerinin karmaşıklığının hesaplanması ve test sonuçları aşağıda tablolar ile gösterilmektedir[16].

Kabakçı (2015)' ya göre Sonuç olarak, bir program kod modülü içerisinde if, while, switch gibi dallanma ifadelerini inceleyerek modülün ne kadar güvenilir olduğunu ölçmeye çalışan McCabe karmaşıklığıdır. Çizge üzerinde bulunan tepe ve ayrıtların toplam sayısına bakarak program kod bloğunun kontrol akış çizgesini çıkarır. Sonuç olarak, modülün karmaşıklığı  $V(G) = E - N + 2 * P$  bağıntısı yardımıyla ölçülür[16].

Kabakçı (2015)' ya göre Halstead Karmaşıklığı ise, Fonksiyonel-yazılım karmaşıklık ölçüm yöntemlerinden biridir. Program kod bloğunda ne kadar operatör ve operand ifadeleri varsa, sayısına göre yazılımın karmaşıklığını Halstead karmaşıklığı(complexity) inceler ve bu da bir yazılım metriği olarak bilinir. Operatör ve operandlar, Halstead yönteminde temel elementlerdir. Bildiğimiz gibi, operatörler aritmetiksel ve mantıksal işlemleri gerçekleştirmeyi sağlayan anahtar kelimeler ve semboller olarak tanımlanır. Operandlar ise, aritmetiksel ve mantıksal işlemler harici kullanılan diğer değişken ve sayılara verilen isimdir. Halstead karmaşıklığı(complexity) operatörleri ve operandları aşağıdaki gibi kullanarak bazı

yazılım metriklerini hesaplar[16].

$\mu_1$  = tekil operatörlerin sayısı

$\mu_2$  = tekil operandların sayısı

$N_1$  = toplam operatörlerin sayısı

$N_2$  = toplam operandların sayısı

**Tablo 2. Bazı Yazılım Metriklerinin Hesaplanması**

Program vocabulary	$\mu = \mu_1 + \mu_2$
Program length	$N_1 + N_2$
Calculated program length	$H = \mu_1 \log_2 \mu_1 + \mu_2 \log_2 \mu_2$
Halstead Program Volume	$V = N \times \log_2 \mu$
Halstead Program Difficulty	$D = (\mu_1 / 2) * (N_2 / \mu_2)$
Halstead Program Effort	$E = V * D$

Kabakçı (2015)' ya göre Yukarıdaki program uzunluğu ve sözcük kapasitesi Halstead' ın diğer metriklerin hesaplanmasında yardımcı olabilmesi için öncelikle temel girdi olarak . program uzunluğu ve sözcük kapasitesi hesaplanır. Diğer metriklere aşağıda yer verilmektedir[16].

*Halstead Program Uzunluğu (Length);*  $N$  (Program Length) =  $N_1 + N_2$  en fazla 300 uzunluk değerine sahip olmalıdır. Kod bloğu 300 den büyük uzunluk değerine sahip ise yeniden tasarlanması tavsiye edilir. Uzunluk değeri 500 den büyük ise kod bloğu kesinlikle gözden geçirilip yeniden tasarlanmalıdır[16].

*Halstead Program Hacmi (Volume);*  $V$  (Volume) =  $N \times \log_2 \mu$  hacim değeri; en az 20, en fazla 1000 olmalıdır. Hacim değeri 1000 den büyük ise bu kod bloğu yeniden tasarlanmalıdır[16].

*Halstead Program Zorluğu (Difficulty);* programın hata toleransını  $D = (\mu_1 / 2) * (N_2 / \mu_2)$  ile gösterir ve sonucu en fazla 50 olmalıdır. Eğer 50' den büyük değer çıkarsa bu kod bloğu yeniden tasarlanmalıdır[16].

*Halstead Program Eforu (Effort)*; programın anlaşılma çabasını  $E = V * D$  ile gösterir. Bir kod bloğu 500.000' den büyük değerlere sahipse az kaliteli bir program olarak değerlendirilebilir ve yeniden tasarlanması gerekebilir[16].

*Halstead Program Zaman Eforu*; programın anlaşılma çabasını saniye cinsinden  $T = E / 18$  ile gösterir. Bir kod bloğunun değeri 5.000' den, yani 1 saat 30 dakikadan büyük değerlere sahip az kaliteli bir kod bloğu olarak değerlendirilebilir ve bu kod bloğunun yeniden tasarlanması gerekebilir[16].

*Halstead Program Hata(Bug) Sayısı*; programda tahmin edilen hata sayısını  $B = (E^{(2/3)}) / 3000$  ile gösterir[16].

"Hesaplanan bu metrikler, kaynak koddan statik olarak hesaplanır. Halstead program uzunluğu (H) aşağıdaki formüle göre hesaplanır"[16].

$$H = \mu_1 \log_2 \mu_1 + \mu_2 \log_2 \mu_2 \quad (3)$$

Halstead yazılım metriklerinin geçerliliği, çok çeşitli yazılım dillerinde istatistiksel olarak pek çok kez onaylanmıştır. Bu Halstead metriklerinden çeşitli yazılım ölçümleri hesaplanabilir[16].

Kabakçı (2015)' ya göre Sonuç olarak Halstead karmaşıklığı, fonksiyonel yazılımlar için yazılım metrikleri kullanılmakta ve yazılımların yapısal karmaşıklığını ölçüp, program kodlarının analizini yaparak sayısal bilgi sağlar. Halstead karmaşıklığının temel çalışma mantığı, program içerisinde bulunan operatör ve operand sayılarının araştırılıp, bu sayılar aritmetik olarak toplanır ve programın karmaşıklığını ölçerek hesaplayabilir[16]. Yazılım metrikleri üzerine açıklanan yazımızda Fatih Kabakçı'nın hazırlamış olduğu web sayfasında yayınlanan yazısından yararlanılmıştır [[https://www.fatihkabakci.com/Makaleler-halstead\\_karmasikligi/](https://www.fatihkabakci.com/Makaleler-halstead_karmasikligi/)].

## 5. ALGORİTMALAR (ALGORITHMS)

Ortalama bağlama sayısı için algoritma üzerinde çalışılmıştır. Görüldüğü üzere, komşu ortalama bağlama sayısı algoritmasında `append_node`, `node` ve `neighbors` gibi bazı methodlar kullanılmıştır. Python için NetworkX [20] 'de bulunan çizge yapısında kullanılan yöntemler ve arayüzlerdir. Ayrıca Python'da `copy`, `clear` gibi yardımcı yöntemler kullandık. Algoritma, çizgeler için ortalama bağlama sayısını hesaplayabilir. Yönlendirilmemiş herhangi bir bağlı çizge için ortalama bağlama sayısını hesaplamak için geliştirilen bir algoritma veriyoruz.

#### The Average Binding Number (Bindav)

*Method:* Bindav(G)

*Input:* Graph G

*Output:* The average binding number of any graph

*Begin*

```

// Set the total Bindv(local vertex binding number) as 0
total = 0
// Take the vertices list of graph G
vertices = G.nodes
foreach v in vertices
    // Set the maximum bind for vertex v as minimum integer
    minBindv = MaxInt
    // Take the copy of graph G as temporary graph
    tempS = copy.deepcopy(S)
    // append the vertex v
    tempS.append_node(v)
    // Take the number of components of temporary graph
    nbs = nx.number_connected_components(tempS)
    // If number of components of temporary graph is not equal vertices,
    // calculate the Bindv
    if v not in tempS      tempS.append(v) end if
    // Calculate the Bind with vertex from temporary graph
    tempBindv1 = len(nbs)
    tempBindv2 = Bind(graph, tempS.nbs,tempS)
    // Compare the Bindvs with the minimum Bindv

```



```

        if      tempBindv1<minBindv      minBindv= tempBindv1
end if

        if      tempBindv2< minBindv      minBindv= tempBindv2
end if

tempS.clear()
// Take the total of Bindv
total = total + minBindv
end foreach
// Return the calculeted total Bindv
return total/(len(vertices))
end

```

Algoritma öncelikle tepe sayısını seçer ve seçilen tepelerin komşu sayısını bulur. Ardından her tepenin bağlama sayısı sonuçları bulunur. Son olarak, çizgenin ortalama bağlama sayısını döndürür.

## 6. TEST SONUÇLARI VE KARMAŞIKLIK

Bir çizgenin bağlama sayısı, bir çizgenin yapısını daha iyi anlamak için kullanılabilen önemli bir karakteristik ölçümdür. Bu, bir çizgenin ayrıtlarının ne kadar "iyi dağılmış" olduğunu ölçme girişimidir. Bir çizgenin bağlama sayısı, polinom zamanında hesaplanabilir[36]. G'nin ortalama bağlama sayısı  $bind_{av}(G) = \frac{1}{n} \sum_{v \in V(G)} bind_v(G)$  olarak tanımlanır, ortalama bağlama sayısı n cinsinden bir polinom olarak sınırlandırılır.

Bazı özel çizgelerin ortalama bağlama sayısı elde edilmiş ve bu sonuçlardan bazıları aşağıda gösterilmiştir.

**Tablo 3: Bazı Özel Çizgelerin Sonuçları**

<i>Graphs</i>	<i>bind<sub>av</sub></i>	<i>Elapsed Time(sec)</i>
P5	0.799	0.018
P7	0.857	0.567
K(3,3)	0.428	11.357
B(2,4)	0.603	67.800
B(3,4)	0.548	96.687
B(2,5)	0.399	82.350
B(3,5)	0.441	106.594

Bu bölümde,  $bindav(G)$ 'nin polinom zamanda hesaplanabileceğini gösteriyoruz. Ortalama bağlama sayısı polinomial karmaşıklık olduğundan, "n" ile tanımlanır, burada n, G çizgesindeki tepe sayısıdır.

Algoritmaların Radon kod metriklerinin karşılaştırmalı analizi aşağıda verilmiştir. Bazı sonuçlar Tablo 4 ile 7 kullanılarak elde edilebilir.

Önerilen algoritmalar Python'da uygulandı. Algoritmaları Radon kod ölçümleri ile analiz ettik. Aşağıda verilen tablolar, ortalama bağlama sayısı algoritmasının kod ölçütlerinin sonucunu içerir.

*Tablo 4. bind av' ın Ham (Raw) Metrikleri.*

<i>Ham Metrikler</i>	<i>bind av</i>
<i>LOC</i>	<i>75</i>
<i>LLOC</i>	<i>46</i>
<i>SLOC</i>	<i>46</i>
<i>Comments</i>	<i>27</i>
<i>Single Comments</i>	<i>27</i>
<i>Multi</i>	<i>0</i>
<i>Blank</i>	<i>2</i>
<i>Comment Stats</i>	
<i>(C%L)</i>	<i>36%</i>
<i>(C%S)</i>	<i>59%</i>
<i>(C+M+L)</i>	<i>36%</i>

Tablo 4'e göre, bu programın yorum satırları ile iyi açıklandığını göstermektedir. Algoritma 75 satırdan oluşmaktadır. Mantıksal satır sayısı, 46 satırdır. Yorum satırı sayısı oranının mantıksal satır sayısı oranına göre % 20' nin üzerinde olduğu görülmektedir. Bu da programın yorum satırları ile iyi anlatıldığını göstermektedir.

Tablo 5. bindav' ın Halstead Metrikleri

Halstead Metrikler	bindav
<i>h 1</i>	6
<i>h 2</i>	23
<i>N1</i>	14
<i>N2</i>	28
<i>Vocabulary</i>	29
<i>Length</i>	42
<i>Calculated Length</i>	119.55
<i>Volume</i>	204.035
<i>Difficulty</i>	3.652
<i>Effort</i>	745.172
<i>Time</i>	41.398
<i>Bugs</i>	0.068

Tablo 5'e göre, bindav algoritması için; Fonksiyonel yazılım karmaşıklığını ölçmek için en çok kullanılan metrikler olan Halstead metrikleri ile program incelendiğinde N1 + N2 program uzunluğunun 42 olduğu görülmektedir. Programın uzunluğu 300 sınırının altında olduğu için programın yeniden tasarlanmasına gerek yoktur. Program hacim değerine bakıldığında; bu değer 29 olarak görünmektedir. Bu değer 1000 ve üzeri sayılara sahip olmadığı için programın karmaşık bir yapıya sahip olmadığı görülmektedir. Bir kalite standardı olan efor değeri de 5000 sınır değerinin altındadır. Bu, programın ölçütleriyle ilgili başka bir artı gibi görünüyor. Benzer şekilde, 90'ın altında bir zaman değerine sahip olmak, programın kalitesini gösteren başka bir değerdir. Tahmin edilen hata sayısını gösteren bug değeri de 0'a çok yakındır, bu da programdaki kodla ilgili bir hata olasılığının çok düşük olacağını gösterir.

Tablo 6. bindav' ın Sürdürülebilirlik Endeksi.

Sürdürülebilirlik Endeksi	
bindav	A

Tablo 6' ya göre algoritma; Sürdürülebilirlik Endeksi, programın güncellenebilirlik özelliğini gösterir. Geliştirilen program, A değerini alarak, en yüksek değeri almıştır. Bu, programın gelecekte yeni talepleri kendi bünyesinde kolaylıkla karşılayabileceğini göstermektedir.

Tablo 7.  $bindav$ ' ın Siklomatik Karmaşıklık.

Siklomatik Karmaşıklık	
$bindav$	$F 5:0 bind - B$
	$F 44:0 bind_{av} - B$

Tablo 7' ye göre, geliştirilen algoritmalarda ortalama bağlama sayısı ( $Bind_{av}$ ), yazılım metriklerinden değerleri 10'un üzerinde olduğu için B değerini almıştır. Bu değer 50 ve üzeri olmadığından programın döngüsel karmaşıklık oluşturmadığı görülmektedir.

Önerilen algoritmalar Python'da uygulandı. Algoritmaları yol(path), tam çizgeler (complete graph) ve birkaç özel çizgeler sınıfında test ettik. Verilen tablolar ortalama bağlama sayısı değerlerini içerir.

## 7. SONUÇ

Bu çalışmada, ortalama bağlama sayısı, bir çizgenin her tepesinin yerel bağlama sayısının ortalaması olarak bir ağın zedelenebilirliğini ölçmek üzerinde çalışılmıştır. Ek olarak, ortalama bağlama sayıları kodla hesaplanmış ve popüler ara bağlantı ağlarının kararlılığı incelenmiştir. Eğer  $bind_{av}(G)$  büyükse, bu, G'nin birçok iyi dağıtılmış ayrıta sahip olması anlamında, bu grafikte G'nin tepelerinin birbirine iyi bağlı olduğu anlamına gelir.

Ağların tasarımı, güvenilirlik ve verimlilik gibi önemli kriterlere sahiptir. Güvenilirlik ve verimlilik, ağ tasarımında önemli kriterlerdir. Bir ağ tasarlamak istediğimizde, mümkün olduğunca dayanıklı olmasını isteriz. Herhangi bir iletişim ağı, bağlantılı bir çizge olarak modellenabilir. Bu çalışmada, ağ zedelenebilirliği için yeni bir çizge parametresi yani ortalama bağlama sayısı sunulmuştur.

Gerçek hayattaki ağlarda güvenlik açığını ölçmek için algoritmaların nasıl kullanılabileceği ve verilen zedelenebilirlik parametresinin nasıl faydalı olabileceği gösterilmiştir.

Çalışmanın amacı; ortalama bağlama sayısının daha belirgin olduğu bir zedelenebilirlik ölçüm parametresi olarak tanımlanması ve aralık çizgelerinin ortalama

bağlama sayısı için bir polinom zaman algoritması oluşturulmasıdır.

Bu çalışmada, ortalama bağlama sayısının genellikle çizgelerin zedelenebilirliğini ölçtüğü düşünüldü. Bu parametreleri hesaplamak için algoritmalar önerdik. Algoritmayı analiz ettik ve Tablo 4 ile 7' de gösterilen Radon yazılım metrikleriyle karşılaştırdık. Sonuçlar, önerilen algoritmaların verilen zedelenebilirlik parametrelerini doğru bir şekilde hesapladığını göstermektedir. Radon yazılım metriklerinden elde edilen sonuçlardan da anlaşılacağı üzere algoritmanın değerleri ideal sınırlar içindedir.

## KAYNAKLAR

- [1] Aslan, E. The Average Binding Number of Graphs. *Scienceasia*, 2019. 45/1,85-91
- [2] Aslan, E., Bacak-Turan, G. Mean rupture degree of graphs, *University Politehnica Of Buch. Sci. Bull. Ser. A-App. Math. and Phy.* 2016. 78/1: 233–242.
- [3] Aslan, E and Kırılancı A., Computing the Scattering Number and the Toughness for Gear Graphs, *Bulletin of Society of Mathematicians Banja Luka*, Vol 18(2011),5-15
- [4] Aslan, E. The average lower connectivity of graphs. *J Appl Math* 2014; 2014.
- [5] Aslan, E. Neighbour isolated scattering number of graphs. *ScienceAsia* 2015. 41(6): 423–31.
- [6] Aslan, E., Kılınç, D., Yücel, F., et al. The average scattering number of graphs. *Rairo - Theor Informatics Appl* 2016. 50(3): 263–72.
- [7] Aytac, V. Average lower domination number in graphs. *Comptes Rendus de L'Academie Bulgare des Sciences*. 2012. 65. 1665-1674.
- [8] Aytac, V., & Berberler, Z. (2017). Binding Number and Wheel Related Graphs. *International Journal of Foundations of Computer Science*. 28. 29-38.
- [9] Bagga, K. S., Beineke, L. W., Goddard, W. D., Lipman, M. J., Pippert, R. E. A Survey of Integrity. *Discrete Applied Mathematics*. 1992, 37/38, 13–28.
- [10] Barefoot, C. A., Entringer, R. and Swart, H. C. (1987). Vulnerability in graphs- a comparative survey. *J. Combin. Math. Combin. Comput.*, 1, 13–22.
- [11] Bondy, J. A., Murty, U.S.R. *Graph Theory with Applications*. Elsevier Science Publishing, New York, USA, 1976, 264 s.
- [12] Coleman D, Ash D, Lowther B, et al. Using Metrics to Evaluate Software System. *IEEE Comput* 1994; 27(8): 44–49.
- [13] Chartrand G, Lesniak I (1996) *Graphs and Digraphs*, 3rd edn, Chapman Hall, Orange, CA.
- [14] Deursen A van. Think Twice Before Using the “Maintainability Index.” <https://avandeursen.com/2014/08/29/think-twice-before-using-the-maintainability-index/> 2014;
- [15] G. Gunther and B. L. Hartnell, Optimal K-secure graphs, *Top Catal* 2(3) (1980).
- [16] Halstead Karmaşıklığı. Makale blog 2015[Online]. Available at: [https://www.fatihkabakci.com/Makaleler-halstead\\_karmasikligi/](https://www.fatihkabakci.com/Makaleler-halstead_karmasikligi/)
- [17] Harary, F. *Graph Theory*. Addison-Wesley Publishing, New York, USA, 1969, 274 s.
- [18] Henning M. A, “Trees with equal average domination and independent domination numbers,” *Ars Combinatoria*, vol. 71, pp. 305–318, 2004.
- [19] Jung H. A, On a Class of Posets and the Corresponding Comparability Graphs, *J. Combin. Theory Ser.B*,1124(1978), 125-133.
- [20] Kırılancı A. A Measure Of Graph Vulnerability: Scattering Number, *Hindawi Publishing Corp., IJMMS*30:1 (2002), 1-8.
- [21] Liu Y, Wei Z, Shi J, et al. A polynomial algorithm of edge-neighbor-scattering number of trees. *Appl Math Comput* 2016; 283: 1–5.

- [22] Maintainability Index Range and Meaning. Code Analysis Team Blog, blogs.msdn 2007[Online] 2007 [cited 2020]. Available at: <https://docs.microsoft.com/en-us/archive/blogs/codeanalysis/maintainability-index-range-and-meaning>.
- [23] NetworkX Official Website [cited 2020]. Available at: <http://networkx.github.io>.
- [24] O. R. Oellermann, “Connectivity and edge-connectivity in graphs: a survey,” *Congressus Numerantium*, vol. 116, pp. 231–252, 1996.
- [25] Oman P, Hagemester J. Metrics for assessing a software system’s maintainability. *Proc - Conf Softw Maintenance, ICSM 1992* 1992; 337–44.
- [26] Piazza, B. L. Roberts, F. S. and Stueckle, S. K. Edge-tenacious networks. *Networks*, 1995. 25, 7–17.
- [27] Radon Documentation [cited 2020]. Available at: <https://radon.readthedocs.io/en/latest/>.
- [28] Schach S. R, *Object-Oriented and Classical Software Engineering*, McGraw-Hill (New York, 2011).
- [29] S.S.Al-Tobaili, On the binding number of lexicographic product of wheels with some graphs, *University of Sharjah Journal of Pure & Applied Sciences* 5(15) (2008) 11–23.
- [30] Thenjipalam, Calicut University P.O.,Malappuram, Kerala - 673 635. *Graph Theory. Study Material B.SC. Mathematics, Elective Course (2011 Admission)*
- [31] Turaci, T., Okten, M. 2015. Vulnerability of mycielski graphs via residual closeness, *Ars Combin.*, 118; 419–427.
- [32] Wang Z and Wang Z, Relationships between rupture degree and other parameters, *Int. J. Comput. Math.* 83(12) (2006) 839–851.
- [33] Wei Z, Yue C, Li Y, et al. A polynomial algorithm for computing the weak rupture degree of trees. *Appl Math Comput Elsevier Inc.* 2019; 361: 730–34.
- [34] Woodall D. R, The binding number of a graph and its Anderson number, *J. Combinatorial Theory Ser. B* 15 (1973) 225–255.
- [35] Wu SSY, Cozzens MB. The minimum size of critically m-neighbour-connected graphs. *Ars Comb* 1990; 29: 149–60.
- [36] Zhang & Guoping, Han, Congying & He, Duan, Hua &, Xuping. (2008). A Binding Number Computation of Graph. *Proceedings - 5th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2008.* 4. 285 - 288.