

**T.C.
MANİSA CELAL BAYAR ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**



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

Ağırlıklı Graflarda Bazı Zedelenebilirlik Parametreleri ve Algoritmaları

Tarık Köroğlu

**Danışman
Prof. Dr. Ersin ASLAN**

MANİSA-2024

Tarık
KÖROĞLU

Ağırlıklı Graflarda Bazı Zedelenebilirlik Parametreleri ve Algoritmaları

2024

TAAHHÜTNAME

Bu tezin Manisa Celal Bayar Üniversitesi Hasan Ferdi Turgutlu Teknoloji Fakültesi Yazılım Mühendisliği Bölümü'nde, akademik ve etik kurallara uygun olarak yazıldığını ve kullanılan tüm literatür bilgilerinin referans gösterilerek tezde yer aldığını beyan ederim.

Tarık KÖROĞLU



İÇİNDEKİLER

	Sayfa
İÇİNDEKİLER.....	I
SİMGELER VE KISALTMALAR DİZİNİ.....	II
ŞEKİLLER DİZİNİ	III
TABLO DİZİNİ.....	IV
TEŞEKKÜR	V
ÖZET	VI
ABSTRACT	VII
1. GİRİŞ	1
1.1 İletişim Ağı ve Önemi	2
1.2 Çizge Teorisi Tanımı ve Uygulama Alanları	3
1.3 Zedelenebilirlik (Vulnerability).....	4
1.4 Graflarda Kullanılan Algoritmalar ve Etkileri.....	6
2. GENEL TANIMLAR.....	7
3. GRAFLARDA SAÇILMA SAYISI (THE SCATTERING NUMBER OF GRAPHS).....	10
3.1 Graflarda Scattering Sayısı Algoritması	11
3.1.1 Graflarda Scattering Sayısı Örneği ve Sonuç Tabloları	12
3.1.2 Graflarda Scattering Sayısı Kod Yapısı ve Anlatımı	15
4. GRAFLARDA ORTALAMA SAÇILMA SAYISI (THE AVERAGE SCATTERING NUMBER OF GRAPHS).....	17
4.1 Ortalama Saçılma Sayısı için Sınırlar.....	22
4.2. Çizge Yapılarının Ortalama Saçılma Sayıları	24
4.3. Graflarda Ortalama Scattering Sayısı Algoritması.....	25
4.3.1 Graflarda Ortalama Scattering Sayısı Örneği ve Sonuç Tabloları	26
4.3.2 Graflarda Ortalama Scattering Sayısı Kod Yapısı ve Anlatımı	36
5. AĞIRLIKLİ GRAFLARDA ORTALAMA SAÇILMA SAYISI	87
5.1 Tanım ve Motivasyon.....	38
5.2 Ağırlıklı Grafta Ortalama Saçılma Sayısı Örneği	40
5.3 AĞIRLIKLİ GRAFLARDA ORTALAMA SCATTERING SAYISI ALGORİTMASI	42
5.3.1 Ağırlıklı Graflarda Ortalama Saçılma Sayısı Örneği ve Sonuç Tabloları	43
5.3.2 Ağırlıklı Graflarda Ortalama Scattering Sayısı Kod Yapısı ve Anlatımı	52
SONUÇ	56
KAYNAKLAR.....	57

SİMGELER VE KISALTMALAR DİZİNİ

G	Graf (Çizge)
V	Grafın Tepesi
E	Grafın Ayrıtı
V(G)	G Grafının Tepeler Kümesi
E(G)	G Grafının Ayrıtlar Kümesi
$\omega(G-S)$	G-S grafının bileşen sayısı
$deg(v)$	Tepenin Derecesi
$\Delta(G)$	G çizgesine ait maksimum tepe derecesi
$\delta(G)$	G çizgesine ait minimum tepe derecesi
 S 	Çıkarılan tepe kümesi
$\sum_{v \in V(G)} sc_v(G)$	G'nin tüm tepelerin toplamını gösterecektir.
(sc_{av})	Graflarda ortalama saçılma sayısı
$sc(G)$	G grafının saçılma sayısı
$\alpha(G)$	bağımsızlık sayısı
$Sc_w(G)$	Ağırlıklı Graflarda saçılma sayısı
$sc_{av}^w(G)$	Ağırlıklı Graflarda Ortalama Saçılma Sayısı

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 1.1. Basit ve Karmaşık Yapıda Bir İletişim Ağı	2
Şekil 1.2. Bilgisayar Ağı	3
Şekil 2.1 Tam Çizge (K5)	8
Şekil 2.2 Yol Çizge (P4)	8
Şekil 2.3 Döngü Çizge (C5)	8
Şekil 2.4 Yıldız (Star) Çizge	9
Şekil 2.5 Kuyruklu Yıldız Çizgesi.....	9
Şekil 3.1 S7 Yıldız Graf Gösterimi.....	13
Şekil 4.1 G Grafi.....	18
Şekil 4.2 G1 ve G2 Grafları.....	19
Şekil 4.3 G Grafi.....	20
Şekil 4.4 P7 Yol Grafi Gösterimi	27
Şekil 4.5 Uygulamanın Çıktısı	38
Şekil 5.1 P4 Yol Graf Gösterimi	40
Şekil 5.2 Uygulamanın Çıktısı	44
Şekil 5.3 Ağırlıklı Grafta Ortalama Saçılma Sayısı Uygulaması Çıktısı	52

TABLO DİZİNİ

	Sayfa
Tablo 1. G çizgesi için Saçılma Sayısı Hesaplaması (S_0 iterasyonu)	13
Tablo 2. G çizgesi için Ortalama Saçılma Sayısı Hesaplaması (S_1 iterasyonu) .	27
Tablo 3 G çizgesi için Ortalama Saçılma Sayısı Hesaplaması (S_2 iterasyonu) ..	30
Tablo 4. G çizgesi için Ortalama Saçılma Sayısı Hesaplaması (S_3 iterasyonu) .	33
Tablo 5. G çizgesi için Ortalama Saçılma Sayısı Hesaplaması (S_4 iterasyonu) .	34
Tablo 6. G çizgesi için Ortalama Saçılma Sayısı Hesaplaması (S_5 iterasyonu) .	34
Tablo 7. G çizgesi için Ortalama Saçılma Sayısı Hesaplaması (S_6 iterasyonu) .	35
Tablo 8. G çizgesi için Ortalama Saçılma Sayısı Hesaplaması (S_7 iterasyonu) .	35
Tablo 9. G çizgesi için Ortalama Saçılma Sayısı Hesaplaması (Sonuç Tablosu)	35
Tablo 10. G çizgesi için Ağırlıklı Graflarda Ortalama Saçılma Sayısı Hesaplaması (S_1 iterasyonu)	44
Tablo 11. G çizgesi için Ağırlıklı Graflarda Ortalama Saçılma Sayısı Hesaplaması (S_2 iterasyonu)	47
Tablo 12. G çizgesi için Ağırlıklı Graflarda Ortalama Saçılma Sayısı Hesaplaması (S_3 iterasyonu)	49
Tablo 13. G çizgesi için Ağırlıklı Graflarda Ortalama Saçılma Sayısı Hesaplaması (S_4 iterasyonu)	50
Tablo 14. G çizgesi için Ağırlıklı Graflarda Ortalama Saçılma Sayısı Hesaplaması (S_5 iterasyonu)	51
Tablo 15. G çizgesi için Ağırlıklı Graflarda Ortalama Saçılma Sayısı Hesaplaması (S_6 iterasyonu)	51
Tablo 16. G çizgesi için Ağırlıklı Graflarda Ortalama Saçılma Sayısı Sonucu..	52

TEŞEKKÜR

Lisansüstü eğitimim ve çalışmalarım boyunca bilgi birikimleriyle tecrübeleriyle bizlere yol gösteren ve destek olan Danışman hocam Sayın Prof. Dr. Ersin ASLAN'a, lisansüstü eğitim ve öğretim sürecinde desteklerini esirgemeyen Psikoloğum Sayın Göksenin Taşçı'ya, hayatımın her anında bana maddi ve manevi destek veren aileme yürekten teşekkür ederim.

Tarık KÖROĞLU
Manisa, 2024



ÖZET

Yüksek Lisans Tezi

Tarık KÖROĞLU

**Manisa Celal Bayar Üniversitesi
Lisansüstü Eğitim Enstitüsü
Yazılım Mühendisliği Anabilim Dalı**

Danışman: Doç. Dr. Ersi ASLAN

Ağlarda zedelenebilirlik ve ağların güvenilirliği birçok alanda önem arz eden bir konudur. Zedelenebilirlik bir ağda mevcut olan cihazların bağlantılarında oluşabilecek aksaklıkların olması durumunda iletişimlerinin kopana kadar gösterdikleri dayanma gücü olarak ifade edilir. Zedelenebilirlik ölçümleri yapılabilmesi için graflarla modellenmesi ve zedelenebilirlik parametrelerinden faydalanılması gerekmektedir.

Bu tez kapsamında zedelenebilirlik ölçüm parametrelerinden olan saçılma sayısı ele alındı. Ağırlıksız bir graf için saçılma sayısı ve ortalama saçılma sayısı hesabı ve buna bağlı uygulamalı kodu hazırlanmıştır. Modellenen bu yapı için zaman karmaşıklığı hesaplanmıştır. Bu makale kapsamında Ağırlıklı graflarda ortalama saçılma sayısı hesabına algoritmalarına, uygulamalı projesine ve sonuç tablolarına yer verilmiş ve üzerinde çalışılmıştır. Ek olarak hazırlanan algoritmanın karmaşıklık değeri hesaplanmıştır. Ağırlıklı grafta ortalama saçılma sayısı hesabı yapılırken $Sc_{vw}(G) = \max\{c(G - S_v) - w|S_v|, c(G - S_v) \geq 2\}$ tanımlandı. Saçılma sayısı hesabına ek çıkartılan değerlerin ağırlık değerleri hesaba katılmıştır. Sonuç olarak ağırlık eklenen grafta ortalama saçılma sayısı tanımı $sc_{av}w(G) = \frac{\sum_{v \in V(G)} sc_{vw}(G)}{n}$ olacaktır.

Bu çalışmada literatürde öncesinde mevcut olmayan Ağırlıklı Graflarda ortalama saçılma sayısı için genel sonuçlara tanım, teoremlere ve zaman karmaşıklığına yer verildi. Paylaşılan algoritma ve kod ölçütleri ile analiz edilmiştir.

Anahtar Kelimeler: Zedelenebilirlik, Graf Teorisi, Algoritma, Bağlama Sayısı, Bağlantılılık, Ortalama Saçılma Sayısı, Ağırlıklı Graf, Saçılma sayısı

2024, 56 sayfa

ABSTRACT

M.Sc. Thesis

Tarık KÖROĞLU

**Manisa Celal Bayar University
Graduate School of Education
Department of Software Engineering**

Supervisor: Prof. Dr. Ersin ASLAN

Vulnerability in networks and reliability of networks are important issues in many areas. Vulnerability is expressed as the endurance power of the devices in a network until their communication is broken in case of any disruptions in their connections. In order to make vulnerability measurements, it is necessary to model it with graphs and to benefit from vulnerability parameters.

In the scope of this thesis, scattering number, which is one of the vulnerability measurement parameters, was discussed. Scattering number and average scattering number calculations for an unweighted graph and the related applied code were prepared. Time complexity was calculated for this modeled structure. In the scope of this article, average scattering number calculation algorithms in weighted graphs, applied project and result tables were given and worked on. In addition, the complexity value of the prepared algorithm was calculated. While calculating the average scattering number in the weighted graph, $Sc_{vw}(G) = \max\{c(G - S_v) - w|S_v|, c(G - S_v) \geq 2\}$ was defined. The weight values of the values that are removed in addition to the scattering number calculation are taken into account. As a result, the definition of the average scattering number in the weighted graph will be $sc_{av}w(G) = \frac{\sum_{v \in V(G)} sc_vw(G)}{n}$.

In this study, general results, definitions, theorems and time complexity for the average number of scatters in Weighted Graphs, which were not previously available in the literature, were included. It was analyzed with shared algorithm and code criteria.

Keywords: Vulnerability, Graph Theory, Algorithm, Number of Connections, Connectedness, The Average Scattering Number, Weighted Graph, Scattering Number

2024, 56 pages

1. GİRİŞ

Geçmişten günümüze insanlar; kendi talep ve ihtiyaçları doğrultusunda yaşadıkları problemlere çözüm üreten, talep ve isteklerini gerçek hayata yansıtan bir takım arayışlar içine girmişlerdir. Bu durum da beraberinde problemlere ve taleplerine uygun sistemler tasarlayıp bunları geliştirmelerine sebep olmuştur. Günümüzde yaşantımızın standart bir düzende ilerleyebilmesi için birçok sisteme ihtiyaç duyarız. Bu sistemlere örnek olarak iletişim ağları, bilgisayarlar ağlarında iletişim, ulaşım sistemleri, bilgi iletişim sistemleri vb. verilebilir. Bu sistemler günlük hayatta doğrudan ya da dolaylı olarak kullandığımız sistemlerdir. Değerlendirildiğinde, kurulan bu sistemlerin önemi açıktır.

Talep ve isteklerin karşılanabilmesini sağlayan bu sistemler ile ilgili olarak; çözülmesi gereken problemler, ihtiyaçlara bağlı olarak arttığı için karmaşıklık da buna bağlı olarak artmıştır. Bu durum daha karmaşık sistemler ve gereksinimleri beraberinde getirmiştir. Oluşturulan sistemlerin genel amacı ihtiyaçlardan beklenen fonksiyonların tam olarak yerine getirilebilmesidir. Sistemlerin oluşturulması ve beklenen fonksiyonların yerine getirebilmesinin dışında yönetilebilmesi de önem taşır.

Sistemlerin genel olarak sınıflandırıldığında karşımıza 4 farklı türde sınıflandırma çıkmaktadır [1]. Bu sistemler sırasıyla;

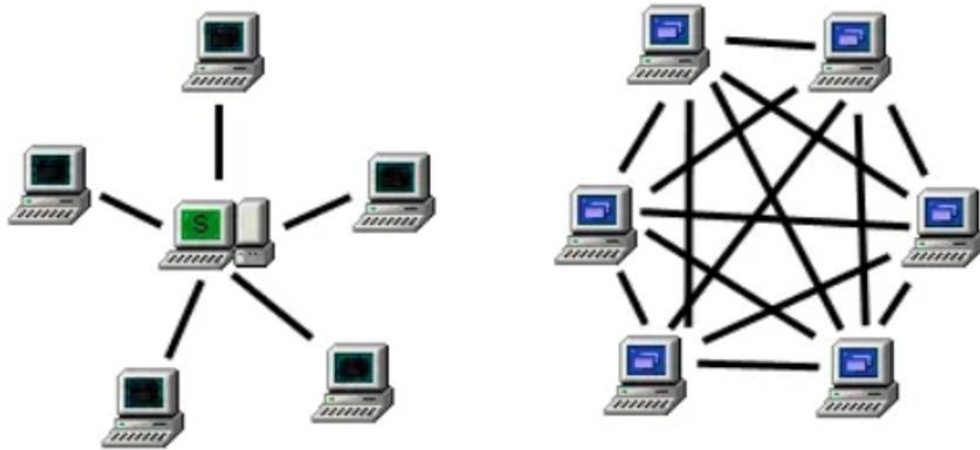
- **Deterministik ve Probalistik Sistemler:** Deterministik sistemlerde girdi için kullanılan verilere bakıldığında çıktı tam olarak hesap edilebilmektedir. Probabilistik sistemlerde ise genel kurallar tam olarak bilinemediği için buna bağlı olan davranış da kestirilememektedir [1].
- **Kapalı ve Açık Sistemler:** Kapalı sistemler çevre (dış etmenler) ile herhangi bir etki ve iletişimi yoktur. Açık sistemlerde çevre ile bilgi ve ürün (malzeme) değişimi yapılır [1].
- **Statik ve Dinamik Sistemler:** Çevrede oluşan değişiklikler kullanılan sistemi hızlı ve etkili bir biçimde etkiliyorsa sistem dinamik sistemdir.
- **Basit ve Karmaşık Sistemler:** Sistemlerde çok az bileşen sayısına sahip ve buna bağlı az ilişki sayısı varsa bu sistem türüne basit sistemler denir [1].

Karmaşık sistemler verimli ve kullanılan algoritmaların optimum kullanılabilmesi için; güvenilir, yönetilebilir, sürdürülebilir, emniyetli vb. niteliklerine sahip olacak biçimde dizayn edilmeli ve tasarlanmalıdır. Kullanılan tasarımlar ve algoritmalar ile karmaşık sistemlere uygun çözümler verebilmek hedeflenmiştir.

Günlük rutinlerde gün geçtikçe daha fazla yer kaplayan karmaşık sistemlerin başında iletişim sistemleri ve işlenmemiş veriler gelmektedir. Bu sistemlere IoT ağları, network'e uzanan tüm sistemler ve alt bileşenleri, bilgisayar ağları ve haberleşme açısından oldukça önemli olan cep telefon şebekeleri örnek verilebilir.

1.1 İletişim Ağı ve Önemi

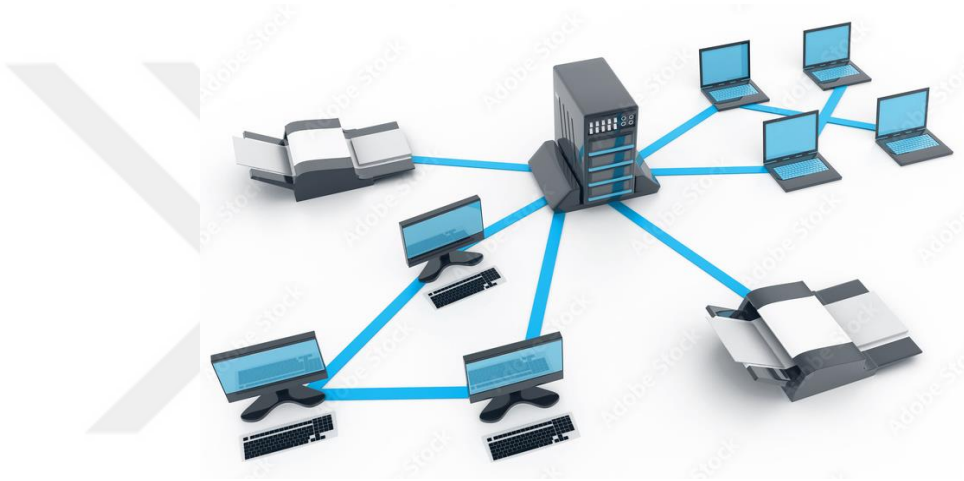
İletişim insanlar arasında, düşünce, bilgi ve haberleşme gibi akla gelebilecek olan birçok çeşit yol ile kişiden kişiye aktarım sağlanmasıdır. Aslında bilgisayar ağları, network connection ve sistemleri temel olarak insan bilgi alışverişlerinden esinlenerek oluşturulmuştur. Ağlarda iletişim uzak mesafeler arası veri iletimi veya haberleşmesi olarak tabir edilir. Bilgisayar ağlarında ise ilk geliştirilen uygulama ve içerikler 1960'larda başlamıştır. Kişisel bilgisayar kullanımının artışı sonrasında ise 1980'lerde yerel ağ içerik ve uygulamalarında oldukça yoğun çalışmalar ve araştırmalar meydana gelmiştir [23]. Bu gelişim sürecinde bilgisayar ağlarının iletişim ağlarına da oldukça büyük etkisi olmuştur. Günümüzden örnek verilecek olursa kişisel kullanım için üretilmiş bir bilgisayar dahi ağ erişimi ve bağlantı sağlayabilecek entegre ve düzeneğe sahiptir.



Şekil 1.1. Basit ve Karmaşık Yapıda Bir İletişim Ağı

1.2 Çizge Teorisi Tanımı ve Uygulama Alanları

İletişim ağlarının öneminde de bahsedildiği gibi bir verinin, bilginin bir noktadan diğer bir noktaya uygun optimizasyon algoritmalarıyla hızlı ve güvenli bir şekilde transfer edilmesinin önemi günden güne artmaktadır. Veri ve bilginin hızlı bir şekilde aktarım sağlarken dikkat edilmesi gereken noktalar vardır. İletişim ve paylaşımın verimli olabilmesi için bir networkte iletişim güvenilir ve kesintisiz olmalıdır. Ağlar, ana iletişim merkezi ve buna bağlı merkezi bağlantı noktalarından meydana gelmektedir. Bilgisayar ağları ve bu alt yapıda oluşturulmuş ağlar belirlenmiş iletim yolları sayesinde bir dizi merkezi oluşturur.



Şekil 1.2. Bilgisayar Ağı

Çizge teorisi, var olan bir problemin tepeler ve ayrıtlar ile modellenmesini bu modelin çizgeler ile gösterilmesini ve ayrıca bu çizgeleri incelemesini yapan bir matematik dalıdır [2]. Çizge teorisinin ana prensibi oluşturulan modelin çözümünü kolaylaştırmaktır. Bu durum da beraberinde gerçek dünyada yaşanan problemin çözümünde zaman ve çaba harcanmasını engeller. Reel sorunlar graflar ile modellenebilir. Çizge teoride modelleme yapmamıza yarasa da bazı zedelenebilirlik, nasıl veya ne kadar dayanıklılık göstereceği önemli bir sorun oluşturmaktadır.

Çizge;

- $G = (V, E)$ şeklinde gösterilir [2].
 - V, Tepeleri ifade etmektedir. (Vertices)
 - E, Ayrıtları ifader etmektedir. (Edge)

Ağ yapıları tepeler (V) ve ayrıtlar (E) ile modellenmektedir [2]. Tepeler ağın merkez noktalarını ifade etmektedir. Ağların birbirleriyle olan bağlantıları ayrıt olan çizgilerle modellenmektedir. Ağ tasarımı sonucunda çıkan çizge modeli üzerinden sayısal hesaplama yapılabilir. Bu durum da beraberinde güvenilir ve düşük maliyetli bir ağ oluşturmamıza yaramaktadır. Düşük maliyet, güvenilir ağ ve kesinti sorunlarını inceleyebilmek ve analiz edebilmek için bazı zedelenebilirlik parametreleri de mevcuttur.

Çizge teorisi ile birlikte gelen önemli çözümlerden bir tanesi kesintilere karşı direnç gösterebilme kabiliyetidir. Kesintilere karşı uyguladığı bu direnç sayesinde diğer oluşturulmuş ağlara kıyasla oldukça avantaj sağlamaktadır. Çizgeler reel hayatta yaşanan problemlerin modellenmesi için kullanılmaktadır. Örnek olarak Casus ağı gibi reel hayatta yaşanabilecek bir sorunun modeli oldukça rahat bir şekilde modellenabilir. Bir ağda oluşabilecek güvenlik açıkları için güvenlik açığı parametresi kullanılmaktadır. Bu parametreler sayesinde daha önce belirlenmiş istasyonlar veya bağlantıların arızalanması sonucunda kesintiye uğraması anındaki direnci ölçülür.

1.3 Zedelenebilirlik (Vulnerability)

İletişim ağında bağlantıların ya da merkezlerinin zarar görmesi ile iletişim bağlantılarının kopması, ağır hasar alması gibi durumlarda iletişim kaybı olana kadar geçen sürede ağın dayanma gücünün ölçüsüne “zedelenebilirlik” denir [3]. Zedelenebilirlik, sistemde herhangi bir konuda işlevini yitirme, kaybolma, azalma gibi sistemin akışında zarar görmesi durumunda oluşan dayanma gücünü temsil eden bir ölçümdür. Bir iletişim ağından temel olarak kesintisiz veri aktarımı ve kesintisiz iletişim beklenmektedir. Ağ tasarımı yapılırken de bunun değerlendirilmesi gerekmektedir. Hangi durumda nasıl bir hasarın olduğu, bu hasarın yenilenmesi veya eski haline yeniden döndürülmesinin maliyeti gibi konuların araştırılması ve analiz edilmesi gerekmektedir. Dayanıklılık analizi, ağın tasarımı yapılırken oluşturulursa ve değerlendirilirse hedefe yönelik bir ağ modeli seçilebilir veya tasarlanabilir [4].

İş yerlerinde kullanılan bilgisayar ağlarında oluşabilecek zedelenmeler verimlilik konusunda oldukça kayba neden olabilmektedir. Örneğin bir iş yerinde

kullanılan herhangi bir networke sızan solucan tipi bir virüs, bulaştığı ağda yalnızca bir bilgisayarın kullanımını etkilemez. Bu durum tespit edilmez ise kontrolsüzce yayılabilir. Ağdaki bu solucan ağa bağlı diğer makineleri, uç birimleri veya sunucuları da etkileyebilir. Bu durum ağın veya bağlı olan cihazlar için zedelenmenin önemini vurgulamaktadır. Zedelenme çeşitli seviyelerde ağda veya cihazda fonksiyon kayıplarına sebep olabilir. Bilgisayar ağı perspektifinden bakıldığında ise “zedelenebilirlik, bir ağın iletişimi sonlanana kadar gösterdiği dayanma ölçümüdür” [5].

Yukarıda belirtilen örnekte bahsediliği üzere, ağa bağlı olan sistemler veya cihazlar dış ve iç bazlı sebeplerle zedelenmeye maruz kalabilirler. Zedelenmenin neticesinde sistem tasarımından kaynaklı sebepler ile zedelenmenin yayılma ihtimali de vardır. Bu durumda ise mevcut sistemde kısmi ya da kalıcı kayıplar meydana gelebilir. Öncesinde kurgulanan ve tasarlanan sistemin işlevselliğini kaybetmesi sistemde ciddi hasarlara yol açabilir. Sistemlerin oluşabilecek kayıpları engellemek ya da minimuma çekebilmesi için sistemin tasarım aşamasındayken zedelenebilirlik ölçümlerinin gerçekleştirilmesi gerekmektedir. Ölçüm üzerine de analizler yapılması gerekmektedir.

Kurulan sistemler zedelenme ölçümü ve devamında gelen hesaplama işlemleri için öncelikli olarak matematiksel bir zemin oluşturulması gerekmektedir. Bu hedef doğrultusunda sistemlerin tasarım yapılarında birbirleriyle uyumlu çalışan, planlı bir şekilde bağlı alt ve üst bileşenlerden oluştuğu ve alt bileşenlerin farklı önlemlere sahip olduğu görülebilmektedir. Bahsedilen bu yapı çizge teorisinden yararlanarak modellenir. Bu şekilde ele alınan problem ya da tasarım aşamasındaki analizler matematiksel temellere dayandırılmış olacaktır.

“Çizge teorisi basitçe, tepe olarak isimlendirilen noktalar ve bu noktaları birbirine bağlayan ayrıtlardan oluşan tepeler arasındaki ilişkiyi somut bir şekilde gösteren çizgiler topluluğudur.” [6]. İletişim ağları çizgenin ayrıları ve tepeleri var olacak şekilde modellenir. Modelleme yapılırken zedelenebilirlik parametrelerinden faydalanılır. Zedelenebilirlik ölçüm parametreleri; saçılma sayısı, bağlanabilirlik, bağlama sayısı ve ortalama bağlama sayısı olarak tanımlanabilir. Bahsedilen ortalama

parametrelerin bazı zaman ve durumlarda en kötü durum seviyesinde daha faydalı olduğu tespit edilmiştir [7].

Bu çalışma kapsamında ağırlıklı graflarda ortalama saçılma sayısı hesabı için bir algoritma oluşturulmuş ve buna bağlı sonuç tabloları verilmiştir. Tez kapsamında ağırlıklı graflarda saçılma sayısı zedelenebilirlik parametresi ölçümleri için bir örnek ve çözümüne yer verilmiştir. Bu sayede optimum çözüme ulaşmaya çalışmıştır. Değerler hesaplandıktan sonra sonuçların analizi yapılmıştır.

1.4 Graflarda Kullanılan Algoritmalar ve Etkileri

Bu bölüm kapsamında ağırlıklı graflarda ortalama saçılma sayısı (sc_{av}) hesaplamalarının yapılabilmesi amacıyla tanımlanan Graflarda Ortalama Saçılma Sayısı parametresinin algoritması, karmaşıklık değerleri, kodu ve sonuç tablolarına yer verilmiştir. Ayrıca temel alınan saçılma sayısı hesabına, algoritmasına, sonuç tablolarına ve örnek bir uygulamaya yer verilmiştir.

Bir grafin saçılma sayısı, grafin tepelerinin belirlenen sınırlarda ne kadar iyi yayılabileceğini farklı bir deyişle dağılabileceğini ölçer [22]. Kablosuz ağlarda, dağıtılmış hesaplamalarda özellikle iletişim ağları uygulamalarında kullanılabilir. Ortalama saçılma sayısında da hedeflenen durum her tepe için hesaplanmış olan saçılma sayılarının ortalamasını elde ederek sonuçta optimum dağılımı sağlamaktır.

Saçılma Sayısı algoritması verilirken daha önce Fengwei Li ve ark'nın makalesindeki algoritmadan faydalanılmıştır [24]. Daha sonra bizim tarafımızdan verilen algoritmalarda da Fengwei Li ve ark'nın algoritmalarından faydalanılmıştır. Ağırlıklı Graflarda Ortalama Saçılma Sayısı parametresi hesabı ve algoritması paylaşılrken temel esas olarak kullanılan saçılma sayısı hesabı algoritması ve Pycharm Ide'de gerçekleştirilmiş olan kod yapısı örneklerle paylaşılmıştır. Paylaşılan kod yapıları saçılma sayısı hesabına dayanmaktadır. Öncelikli olarak saçılma sayısı hesabı üzerinden ortalama saçılma sayısı hesabı, algoritması ve kodu hazırlandı. Hazırlanan bu temele dayanarak da ağırlıklı graflarda ortalama saçılma sayısı hesabı, algoritması ve kodu hazırlanmıştır.

2. GENEL TANIMLAR

Bu bölüm tez içeriğinde kullanılmış ve gerekli olan bazı tanım ve kavramlar belirtilmiştir.

Tanım 2.1. [8] Bir G çizgesinde sayılabilir tepeler kümesini V , ayrıtlar kümesi ise E olarak isimlendirilir. Bu durumda $G=(V,E)$ sıralı ikiliye çizge adı verilir. G çizgesi seçilen u ve v tepeleri bitişik ise yani aralarında herhangi bir ayrıt barındırıyorsa bu ilişki için $e = (u, v)$ şeklinde gösterilir.

Tanım 2.2. [9] Bir G çizgesinin $E(G)$ ayrıtlar kümesi, $V(G)$ tepeler kümesi ise v tepesine bağlı ayrıtların sayısına o **tepenin derecesi** denir ve $deg(v)$ ile gösterilir.

Tanım 2.3. [9] Bir G çizgesinin değer olarak en küçük tepe derecesine **minimum tepe derecesi** denir. $\delta(G)$ ile gösterilir.

Tanım 2.4. [9] Bir G çizgesinin değer olarak en büyük tepe derecesine **maksimum tepe derecesi** denir. $\Delta(G)$ ile gösterilir.

Tanım 2.5. [9] Bir çizgede, çıkarılan ayrıtların kümesi veya çıkarılan tepelere **kesim kümesi** denir. $|S|$ ile gösterilir.

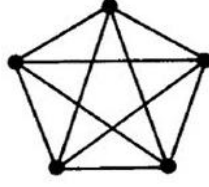
Tanım 2.6. [9] Bir tepenin derecesi ($deg(v)$) sıfır ise o tepeye **izole tepe** denir.

$$deg(v) = 0$$

Tanım 2.7. [10] $V(G)$, G çizgesinin tepeler kümesi olarak değerlendirilir ise, $V(G)$ kümesi içerisinde birbirleriyle komşuluk ilişkisi olmayan maks tepe sayısına **bağımsızlık sayısı** denir. $\alpha(G)$ ile gösterilir.

Tanım 2.8. [9] Bir çizgenin tanımlanmış tüm tepeleri, çizgenin tanımlı diğer tepeleri ile komşuluk ilişkisi var ise bu çizgeye **complete graph (tam çizge)** denir. n tepeli bir tam çizge K_n ile gösterilmektedir.

Şekil 2.1'de K_5 için tanımlı tam çizge örneği verilmiştir.



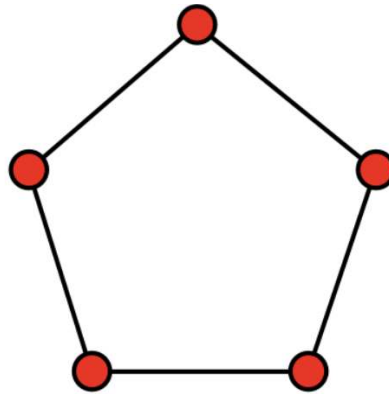
Şekil 2.1 Tam Çizge (K_5)

Tanım 2.9. [11] Bitiş tepeleri ve başlangıç tepeleri tek derecelidir. Diğer tepeleri ise iki dereceli olan çizgelere **yol çizge** (Path Graph) denir. P_n şeklinde gösterilir. n bir tam sayıdır. Şekil 2.2’de P_4 yol çizgesi olarak gösterilmiştir.



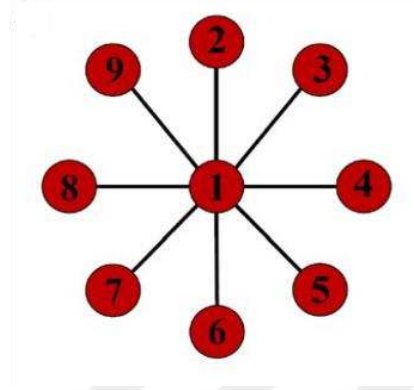
Şekil 2.2 Yol Çizge (P_4)

Tanım 2.10. [12] Yalnızca bir adet döngüden oluşan çizgeye **Döngü Çizge** (Cycle Graph) denir. C_n şeklinde gösterilir. n ile gösterilen ise döngü içerisindeki nokta sayısıdır. Şekil 2.3’de C_5 döngü çizgesi olarak gösterilmiştir.



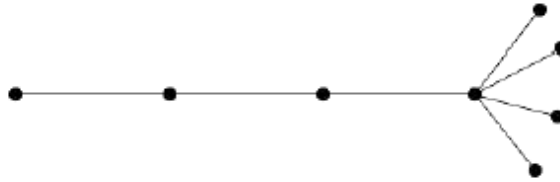
Şekil 2.3 Döngü Çizge (C_5)

Tanım 2.11. [9] $n+1$ tepeye sahip olan bir çizgenin totalde n sayıda tepesinin derecesi bir olduğu durumda merkez tepe derecesi n ise $n+1$ tepeli bir çizgeye yıldız çizge adı verilir. Şekil 2.4’de Yıldız çizgesi(Star Graph) örneği gösterilmiştir.



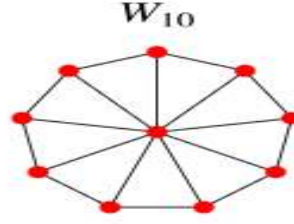
Şekil 2.4 Yıldız (Star) Çizge

Tanım 2.12. [14] Yıldız çizgesi ile kısmi yol çizgesi birleşmesi ile oluşturulur. Yıldız çizgesinin merkez tepesine $n-1$ tepeli P_{n-1} yol çizgesi eklenmesi ile oluşturulur. Oluşturulan bu çizgeye **kuyruklu yıldız çizgesi** (Comet Graph) denir. Kuyruklu yıldız çizgesi Şekil 2.5’de örneğiyle gösterilmiştir.



Şekil 2.5 Kuyruklu Yıldız Çizgesi

Tanım 2.13. [13] Tekerlek çizgesi, bir döngüden yeni bir tepe noktası ve buna bağlı olarak döngünün tüm tepelerini birleştiren kenarlar eklenerek elde edilen bir çizgedir. Yeni eklenen kenarlara tekerleğin jant telleri de denebilir. Döngü çizge C_{n-1} bu çizgeye yeni bir tepe eklenmesiyle C_{n-1} ’deki çizgelerin tüm tepelerinin bu eklenen tepeyle birleştirilmesiyle oluşmaktadır. Buna Tekerlek çizge (Wheel Graph) denir. W_n ile gösterilmektedir. Şekil 2.6’da Wheel Graph örneği gösterilmiştir.



Şekil 2.5 Tekerlek Çizge

3. GRAFLARDA SAÇILMA SAYISI (THE SCATTERING NUMBER OF GRAPHS)

Bu kısımda, saçılma sayısının genel tanımlanması ve buna bağlı parametrelerin tanımı yapılacaktır.

Saçılma sayısı kavramı ilk olarak Jung tarafından ortaya atılmıştır [16]. Ouyang ve ark. ise zedelenebilirlik ölçümleri için ilk kez saçılma sayısı grafını öne sürmüştür [17]. Harary graflarının saçılma sayısının analizi ve ağaçlarda saçılma sayısı hakkında bazı temel sonuçlarda çıkarmışlardır. Zhang, S ve arkadaşları saçılma sayısı ile grafın diğer hesaplanabilir bazı parametreleri arasındaki ilişkiyi göstermişlerdir.

Zedelenebilirlik ölçümlerinde alınan diğer önlemlerden farklı olarak, saçılma sayısı sadece ağın bozulmasının zorluklarını incelemez. Ayrıca hasarın boyutu ve hasarın kendisi hakkında yorum yapabilme imkanı sunmaktadır. G grafının saçılma sayısı “ $sc(G)$ ” olarak isimlendirilir. Tanımlanması ise;

$$sc(G) = \max\{\omega(G - S) - |S| : S \subset V(G), \omega(G - S) \geq 2\} \quad [17]$$

Buradaki $sc(G)$ yukarıda da belirtildiği gibi G grafının saçılma sayısı, $\omega(G - S)$ G-S'deki bileşenlerin sayısını belirtmektedir. $\omega(G - S) \neq 1$ olamaz. $\omega(G - S) - |S|$ olacak şekilde bir S kümesi = $sc(G)$ karşılık gelir. G grafının saçılma kümesi olarakta adlandırılabilir.

3.1 Graflarda Scattering Sayısı Algoritması

Graflarda saçılma sayısı hesabı için bir algoritma hazırlandı. Paylaşılan algoritma üzerinde bir takım metodlar kullanıldı. Saçılma sayısı hesabı için Networkx Kütüphanesi kullanıldı. Networkx kütüphanesi ile scattering number olarak tanımlanan classta removes_from_node ve number_connected_component gibi fonksiyonlar sayesinde graftaki max saçılma sayısı hesaplandı. Uygulama üzerinde saçılma sayısı hesaplanacak olan grafin bir kopyası, bağlı olan komponentlerin listelenmesi için list gibi bazı yardımcı yöntemlere başvuruldu. Paylaşılan bu algoritma sayesinde ağırlıklı olmayan bir grafin saçılma sayısı hesaplanabilir.

Uygulama ve algoritma oluşumunda Fengwei Li ve ark'nın interval graflarda ağırlıklı saçılma sayısı hesabı makalesinden faydalanılmıştır [24].

The Scattering Number Calculation (scattering_number)

Method: scattering_number(G)

1. Input: Star Graph
2. Atanan star graph(G) number_of_nodes() fonksiyonu ile tepe sayısı alınır. (len(self._node)).
3. Yıldız grafin atanmış olan değeri (n) 0 eşitlik kontrolü yap. Eğer 0 ise saçılma sayısı hesaplanmaz.
4. Döngü n defa yürütüleceği ve k'nın 0 ile n-1 arasında değerler alacağı bir iterasyon oluşturun.

4.1 Her Si iterasyon elemanı için;

- 4.1.1 Atanan G grafinin bir kopyasını al. Select edilmiş olan S tepelerini Graftan çıkar. (remove_nodes_from(S)).
- 4.1.2 Grafta kalan tepelerin bağlı olan componentlerini c_w değişkenine ata.
- 4.1.3 Seçilen yaprakların boyutunu (len(S)) negatif kontrolü yapılır. len(S) seçilen_s_yaprak değişkenine atanır.
- 4.1.4 Saçılma sayısı hesabı (c_w – seçilen_s_yaprak) yapılır.
- 4.1.5 Hesaplanan saçılma sayısı değerleri max(max_scatter, scatter_value) karşılaştırılır.

5. Karşılaştırma sonucunda saçılma sayısı için hesaplanan max değeri işlem sonucunda return eder ve saçılma sayısı olarak atanır.
6. Çıktı: scattering_number(G) fonksiyonu sonucu yıldız grafin saçılma sayısı return edilir.

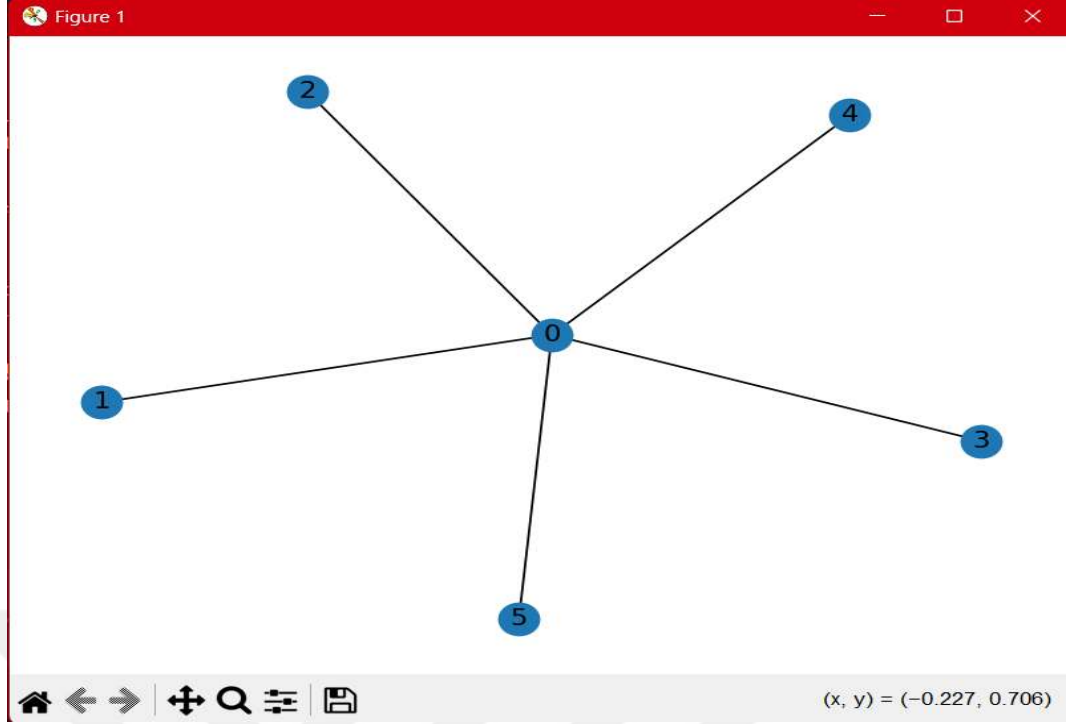
Saçılma sayısı için hazırlanan uygulama ile irtibatlı algoritma paylaşıldı. Saçılma sayısı için hazırlanan bu algoritma ve kod yapısının zaman karmaşıklığı da analiz ve sonuçlar için oldukça önem arz etmektedir. The Scattering Number Calculation başlıklı algoritma $O(n^2)$ olarak hesaplanmış olur. Saçılma sayısı için hazırlanan uygulama iç içe 2 for döngüsüne sahiptir. Buna bağlı olarakta n^2 değeri hesaplanır. Hesaplanan bu değer algoritmanın toplam maliyeti hakkında bilgi vermektedir.

3.1.1 Graflarda Scattering Sayısı Örneği ve Sonuç Tabloları

Saçılma sayısının hesabı için örnek bir uygulama gerçekleştirildi. Uygulamada yıldız graf ele alındı. S_6 grafinin Pycharm Idesi üzerinde oluşturulmuş görseli Şekil 8.1 gösterilmiştir.

Uygulamada belirtilen tepeler ve isimleri görülmektedir. Yıldız grafinin scattering sayısı hesabı yapıldı. Hesaplanan bu değerler tablo haline getirildi. Koşul olarak teoremlerde belirtilmiş olan $w > 1$ değerler tabloda yer almaktadır. Hesaplanmış olan saçılma sayısı değerleri arasındaki maksimum saçılma sayısı yıldız grafin saçılma sayısı olacaktır.

Tablo üzerinde belirtilen S_0 olarak ifade edilen 0. Tepe ile ilişkili olan değerlerin graf üzerinde çıkartılması anlamına gelmektedir. Çıkartılan değerler sonrasında kalan komponentler de tabloda belirtildi. Neticesinde saçılma sayıları her durum için analiz edildi. Saçılma sayıları için maksimum değer içeren durum grafin saçılma sayısı olarak ifade edilmektedir.



Şekil 3.1 S_7 Yıldız Graf Gösterimi

Tablo 1. G çizgesi için Saçılma Sayısı Hesaplaması (S_0 iterasyonu)

S_0 Çıkarılan Değerler	Kalan Komponentler	$W(G-S)$	S	Sc
{0}	{1} {2} {3} {4} {5}	5	1	4
{0, 1}	{2} {3} {4} {5}	4	2	2
{0, 2}	{1} {3} {4} {5}	4	2	2
{0, 3}	{1} {2} {4} {5}	4	2	2

{0, 4}	{1}	4	2	2
	{2}			
	{3}			
	{5}			
{0, 5}	{1}	4	2	2
	{2}			
	{3}			
	{4}			
{0, 1, 2}	{3}	3	3	0
	"{4}			
	{5}			
{0, 1, 3}	{2}	3	3	0
	{4}			
	{5}			
{0, 1, 4}	{2}	3	3	0
	{3}			
	{5}			
{0, 1, 5}	{2}	3	3	0
	{3}			
	{4}			
{0, 2, 3}	{1}	3	3	0
	{4}			
	{5}			
{0, 2, 4}	{1}	3	3	0
	{3}			
	{5}			
{0, 2, 5}	{1}	3	3	0
	{3}			
	{4}			
{0, 3, 4}	{1}	3	3	0
	{2}			
	{5}			
{0, 3, 5}	{1}	3	3	0
	{2}			
	{4}			
{0, 4, 5}	{1}	3	3	0
	{2}			
	{3}			
{0, 1, 2, 3}	{4}	2	4	-2
	{5}			
{0, 1, 2, 4}	{3}	2	4	-2
	{5}			

{0, 1, 2, 5}	{3} {4}	2	4	-2
{0, 1, 3, 4}	{2} {5}	2	4	-2
{0, 1, 3, 5}	{2} {4}	2	4	-2
{0, 1, 4, 5}	{2} {3}	2	4	-2
{0, 2, 3, 4}	{1} {5}	2	4	-2
{0, 2, 3, 5}	{1} {4}	2	4	-2
{0, 2, 4, 5}	{1} {3}	2	4	-2
{0, 3, 4, 5}	{1} {2}	2	4	-2
				Sc_{Max} = 4

3.1.2 Graflarda Scattering Sayısı Kod Yapısı ve Anlatımı

```

import networkx as nx
from itertools import combinations
from matplotlib import pyplot as plt

def scattering_number(G):
    n = G.number_of_nodes()
    if n == 0:
        return 0
    max_scatter = float('-inf')

    for r in range(n):
        for S in combinations(G.nodes(), r):

```

```

G_eksi_S = G.copy()
G_eksi_S.remove_nodes_from(S)
c_w = nx.number_connected_components(G_eksi_S)

components = list(nx.connected_components(G_eksi_S))

if c_w >= 2:
    for i, component in enumerate(components, start=1):
        print(f"Kalan Componentler {i}: {component}")

    global secilen_s_yaprak

    if (len(S) < 0):
        secilen_s_yaprak = -1 * len(S)
    else:
        secilen_s_yaprak = len(S)

    scatter_value = c_w - secilen_s_yaprak
    max_scatter = max(max_scatter, scatter_value)

    print(f"Sacilma sayisi değeri --> {scatter_value}")

    return max_scatter

#Yildiz Grafin yaprak sayisinin belirlenmesi.
n_yaprak_sayisi = 5 # Örnek olarak verilen yaprak sayisi
star_graph = nx.star_graph(n_yaprak_sayisi)

secilen_s_yaprak = 0

# Sacilma Sayisi Hesabi
sacilma_sayisi = scattering_number(star_graph)

print(f"{n_yaprak_sayisi } Yaprakli Yıldız Grafta sacilma sayisi hesabi:

```

```
{sacilma_sayisi }")  
nx.draw_spring(star_graph, with_labels = True)  
plt.show()
```

Saçılma sayısı hesabı için Pycharm ide üzerinde Python programlama dili ile uygulama gerçekleştirildi. Saçılma sayısı hesabı için networkx kütüphanesi kullanıldı. Matplot ve pyplot kütüphaneleri sayesinde oluşturulan grafin modellenmesi yapılmıştır. scattering_number fonksiyonu içeriğinde hesaplamalar gerçekleştirilmiştir. if n == 0: ile tepe sayısı kontrol edilir. Eğer ki tepe sayısı 0 ise saçılma sayısı hesabı gerçekleşmeyecektir. Oluşturulan graftan yaprak çıkartılacağı için grafin bir kopyası alınır. Bu kopyadan remove_nodes_from fonksiyonu ile sırasıyla tepeler belirtilen şartlara uygun graftan ayrılır. Bağlı olan diğer komponentlerde bu noktada listelenecektir. Graflarda saçılma sayısı hesaplama koşulları içerisinde $|S|$ olduğu durumunu yazılan programa yansıtabilmek adına $\text{len}(S)$ 'in negatif durumundan -1 ile çarparak sağlandı. scatter_value değeri her bir durumda w-s için gelen sonucu çıkarır. Bu hesaplanan saçılma sayısının maksimum değeri oluşturulan grafin saçılma sayısı olacağı için bulunan her değer maksimum saçılma sayısı olarak atanan değer ile kıyaslanır. Sonuç olarak tüm değerler analiz edildikten sonra max_scatter olarak tanımlanan değişken return edilir. Return edilen bu değişken grafin saçılma sayısı olarak kabul edilmektedir.

4. GRAFLARDA ORTALAMA SAÇILMA SAYISI (THE AVERAGE SCATTERING NUMBER OF GRAPHS)

Literatürde kullanılan saçılma sayısını da içeren bu çalışma kapsamında graflar içerisinde kullanılabilecek ortalama saçılma sayısı kavramı işlenecektir. Bağlantılı bir G grafında, v bir tepe noktası ise;

$$sc_v(G) = \max\{\omega(G - S_v) - |S_v|, \omega(G - S_v) \geq 2\} \quad [20]$$

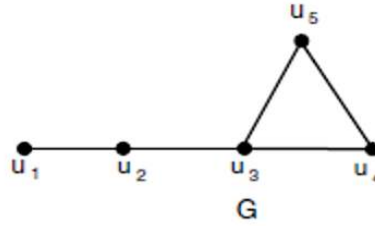
Burada $\omega(G - S_v)$, $G - S_v$ grafinin bileşen sayısını belirtir. Ek olarak maksimum v'yi içeren G'nin tüm bağlantı kesme kümeleri S_v üzerinden alınır. N dereceli (mertebe) bir G grafi için $sc(G) = -n$ ancak G'nin K_n 'nin tamamına izomorf olması durumunda dikkate alınır [21]. Özellikle bakıldığında tam bir K_n grafinin yerel

saçılma sayısı literatürde $-n$ olarak tanımlanmaktadır. “ $Sc_{av}(G)$ ” ile gösterilen G grafinin ortalama saçılma sayısını ifade eder. Tanımlama olarak ise Ortalama saçılma sayısı; [20]

$$sc_{av}(G) = \frac{\sum_{v \in V(G)} sc_v(G)}{n}$$

Burada $\sum_{v \in V(G)} sc_v(G)$, G 'nin tüm tepelerinin toplamını gösterecektir. Herhangi bir G grafi için $sc(G) = \max\{sc_v(G) : v \in V(G)\}$. G 'nin yerel saçılma kümesi, $V(G)$ 'nin kurallara bağlı olarak her bir S_v alt kümesi olarak ifade edilebilir. Bunun için $\omega(G - S_v) - |S_v| = sc_v(G)$ işlemi yapılmalıdır [20]. Son olarak da tam bir K_n grafiğinin ortalama saçılma sayısı $-n$ olarak tanımlanmaktadır.

Örnek olarak Şekil 1'deki G grafi ele alınırsa; Burada $|V(G)| = 5$ ve $|E(G)| = 5$ $sc_{u1} = sc_{u4} = sc_{u5} = 0$ ve $sc_{u2} = sc_{u3} = 1$ olduğu görülebilmektedir. Bundan $sc_{av}(G) = (0 + 1 + 1 + 0 + 0)/5 = 2/5$ sonucu çıkar.



Şekil 4.1 G Grafi

Örnek olarak iki grafin ele alındığını düşünelim. Yaklaşım olarak öneride bulunan ortalama saçılma sayısı, kırılgenlik açısından uygun bir parametre mi yoksa değil mi bunu sorgulayalım.

Örnek 1.1 P_n yolunun ($n > 4$) ve $K_{1,n-1}$ ($n \geq 4$) yıldızının bağlantılılığının eşit olduğu görülebilmektedir. $\kappa(P_n) = \kappa(K_{1,n-1}) = 1$. P_n yolunun ($n \geq 4$) ve $K_{1,n-1}$ ($n \geq 4$) yıldızının ortalama saçılma sayıları farklılık göstermektedir.

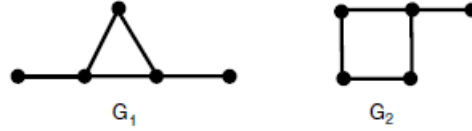
$$sc_{av}(K_{1,n-1}) = \frac{n^2 - 4n + 2}{n} \text{ ve } sc_{av}(P_n) = \frac{n-2}{n}$$

Örnek 1.2 P_7 yolu ile tam iki parçalı $K_{3,4}$ 'ün saçılma sayısının eşit olduğu kolaylıkla görülebilir. Bu durumda P_7 'nin saçılma sayısı (scattering number) $sc(P_7) = sc(K_{3,4}) = 1$ olur. Ek olarak bakıldığında, P_7 yolunun ortalama saçılma sayısı (average scattering number) ve tam bir iki parçalı $K_{3,4}$ farklıdır. $sc_{av}(K_{3,4}) = \frac{-1}{7}$ ve $sc_{av}(P_7) = \frac{5}{7}$. $sc_{av}(P_7)$ $n=7$ olduğu durumda $sc_{av}(P_n) = \frac{n-2}{n}$ yapısına göre neticesini alır. Burada $sc_{av}(K_{3,4})$ ve $sc_{av}(P_7)$ farklı sonuçlar aldığı görülmektedir.

Örnek 1.3 Şekil 3.2'de paylaşılmış olan 2 adet graf vardır. Bunlar G_1 ve G_2 grafları olmak üzere bu grafların saçılma sayısı (scattering number) ve bağlantılılık sayısı (connectivity number) $\kappa(G_1) = \kappa(G_2) = sc(G_1) = sc(G_2)$ bu eşitliğin yanı sıra paylaşılan her bir değer de 1'e eşittir. Fakat bu sonuçları baz alıp ortalama saçılma sayılarının da eşit olduğu kanısına varamayız. Ortalama saçılma sayıları;

$$sc_{av}(G_2) = 1/5$$

$$sc_{av}(G_1) = 2/5$$



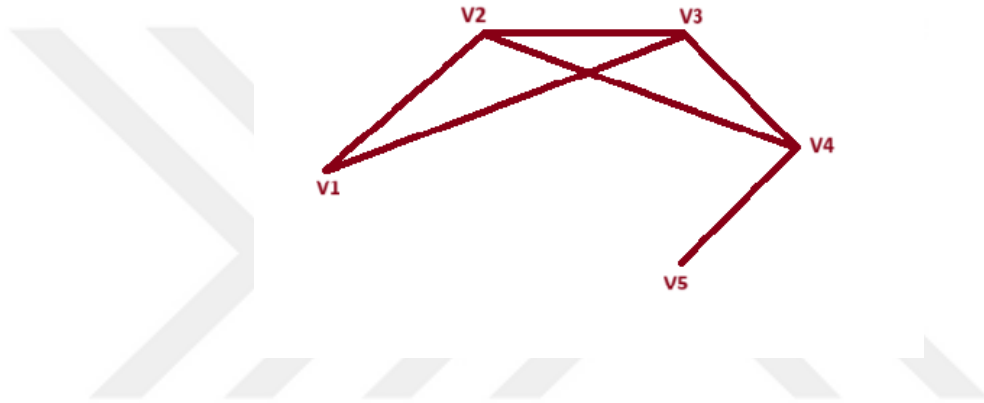
Şekil 4.2 G_1 ve G_2 Grafları

Hesaplamaların neticesinde elde edilen sonuçlar analiz edilmeye açıktır. Genel bir denklem olarak değerlendirildiğinde saçılma sayısı P_n ($sc(P_n) = 1$) 1'e eşit olduğu senaryoda ortalama saçılma sayısı $sc_{av}(P_n) = \frac{n-2}{n}$ denklemine göre P_n sabit değerde olduğu görülmektedir. P_n 'in sabit değerde olmasına karşın ortalama saçılma sayısı her durumda farklı değerler almaktadır. Bu durum da beraberinde ortalama saçılma sayısının muadil diğer güvenlik açığı parametrelerine kıyasla daha verimli sonuçlar elde edilebileceği kanısına varılabilmektedir.

Aynı büyüklükte ve aynı düzende olan graflar arasından daha stabil olan bir grafın seçilmesi istenirse bunun bir yolu, ortalama saçılma sayısı değeri minimum olan

grafın seçilmesidir. Ortalama saçılma sayısı büyük olan graflar daha savunmasızdır ve zedelenmeye oldukça açıktır. Hasar gören bir networkün (ağ) kolayca yeniden yapılandırılması için, tepeler silindikten sonra oluşan bağlı bileşenlerin sayısının daha az olması gerekmektedir.

Şekil 3.3'te G grafi üzerinden graflar üzerinde scattering sayısının hesabına dair bir örnek verilecektir. Hesaplanan bu scattering sayısı üzerinden grafın karmaşıklığı hakkında bilgi verilecektir.



Şekil 4.3 G Grafi

Öncelikle Tepeler kümesi S olarak kabul edilecektir. S denilen Graftan atılan tepelerin kümesidir. Grafın parçalarına ayrılabilmesi için S kümesinin eleman sayısı en az 1, örnekte hedeflenen scattering sayısının şartında $V(G) \supset S$ olduğundan dolayı maksimum 4 olmalıdır. Totalde 5 tepeli bir grafın tüm alt kümelerinin sayısı $2^n - 2$ olarak hesaplanır. Bu örnek için $2^5 = 32$ olacaktır. Şartlara uyum sağlanması için 0 elemanlı boş küme ve 5 elemanlı kümeler çıkartılırsa $32-2=30$ olacaktır. Bu da toplam S kümesi değeridir. Bu alt kümeler özetle değerlendirilecek olunursa; [25]

1. $|S|=1$ olduğu durumda S kümesi şu 5 şekilde seçilebilir:
 - $S=\{v_1\}$, $S=\{v_2\}$, $S=\{v_3\}$, $S=\{v_5\}$ saçılma sayısının genel şartlarında $\omega(G-S)>1$ olduğu için bu kümeler değerlendirmenin dışında kalacaktır.
 - $S=\{v_4\}$ iken $\omega(G-S)=2$ ise saçılma sayısı $\Rightarrow sc(G) = 2 - 1 = 1$ olacaktır.
2. $|S|=2$ olduğu durumda S kümesi şu 10 şekilde seçilebilir:

- $S=\{v_1,v_2\}, \{v_1,v_3\}, \{v_1,v_5\}, \{v_2,v_5\}, \{v_3,v_5\}, \{v_4,v_5\}$ $\omega(G-S)>1$ şartından dolayı bu kümeler değerlendirilmenin dışında kalacaktır.
- $S=\{v_1,v_4\}, \{v_2,v_3\}, \{v_2,v_4\}, \{v_3,v_4\}$ olduğu senaryoda $\omega(G-S) = 2$ olacaktır. Scattering hesabıda $sc(G) = 2-2= 0$

3. $|S|=3$ olduğu durumda S kümesi şu 10 şekilde seçilebilir:

- $\omega(G-S) = 1$ olan değerler saçılma sayısında ki $\omega(G-S) >1$ şartından dolayı değerlendirme dışı kalacağı için yalnızca şartı sağlayan kümeler değerlendirilir.
- $S=\{v_1,v_2,v_4\}, \{v_2,v_3,v_4\}, \{v_2,v_3,v_5\}, \{v_1,v_3,v_4\}$ olduğu durumda $sc(G) = 2-3= -1$ olacaktır.

4. $|S|=4$ olduğu durumda S kümesi şu 5 şekilde seçilebilir:

- $\omega(G-S) = 1$ olan değerler saçılma sayısında ki $\omega(G-S) >1$ şartından dolayı değerlendirme dışı kalacağı için yalnızca şartı sağlayan kümeler değerlendirilir.

Sonuçlar genel olarak yorumlandığında saçılma sayısının en büyük değeri ilk adımda hesaplanan $\{v_4\}$ seçildiği senaryodur. Bu seçimin sonucunda diğer hesaplamalar ile kıyasla saçılma sayısı (scattering number) $sc(G) = 1$ en büyük değeri alır.

Saçılma sayısı hesabında ya da analizinde alt kümeler ve analizleri oldukça önemlidir. G grafi n tepeli bir graf olarak düşünülecek olunursa tüm alt kümeleri 2^n olacaktır. Bu durumdan hiç tepe atılmayan durumu ve tüm tepelerin atıldığı senaryo çıkarılacak olunursa S'in toplam alabileceği 2^n-2 farklı küme mevcut olacaktır.

Hesaplanacak olan her bir kümeyi bir işlem olarak düşünecek olursak n sayısının büyük olduğu durumlarda işlem hacimleri oldukça yükselecek ve karmaşıklık meydana gelecektir. Örnek olarak $n=10$ değerini aldığı durumda $2^{10}-2$ 'den 1022 adet S kümesi olacaktır.

4.1 Ortalama Saçılma Sayısı için Sınırlar

Bu bölüm kapsamında ortalama saçılma sayısı ve buna bağlı grafların parametereleri ile ilişkin teoramlar yer almaktadır. Ortalama saçılma sayısı hesabındaki bazı sınır değerlere ve şartlara yer verilecektir.

Lemma 4.1.1 [20] $G = (V, E)$ bağlantılı bir graf olarak kabul edilirse; Tüm $v \in V$ için $scv(G) \leq k$ olduğu bir durumda saçılma sayısındaki durumda $sc_{av}(G) \leq k$ olmalıdır. Öte taraftan bakılacak olursa yine tüm $v \in V$ durumları için $scv(G) \geq k$ ise $sc_{av}(G) \geq k$ olur.

Teorem 4.1.2 [20] G , n mertebesinde tamamlanmamış bir graf ise, bu durumda

$$sc_{av}(G) \leq sc(G)$$

Teorem 4.1.3 [20] Eğer G grafi, saçılma seti olarak belirlenen değerler benzersiz ise ve n mertebesinden tamamlanmamış bir graf ise bu durum için

$$sc_{av}(G) \leq sc(G)$$

Teorem 4.1.4 [20] Eğer G grafi, β sayısını ve bağımsızlık sayısının α 'yı kapsayan, n mertebesinden tamamlanmamış bir graf olduğu durumda

$$sc_{av}(G) \geq \alpha - \beta - 2$$

Teorem 4.1.5 [20] Eğer G grafi, n mertebesinden, β sayısını ve bağımsızlık sayısını α ve $\beta = \alpha$ 'yı kapsayan, tamamlanmamış bir graf ise, o halde

$$sc_{av}(G) \geq -1$$

Öneri 4.1.6 [20] G , n mertebesinden ve β sayısını kapsayan tamamlanmamış bir graf ise, o durumda;

$$sc_{av}(G) \geq 1 - \beta$$

Teorem 4.1.7 [20] Eğer G , n dereceli, bağımsızlık sayısı α ve sağlamlığı t olan tamamlanmamış bir graf olduğu durumda;

$$2(1 - t), \text{ eğer } t > 1$$

$$sc_{av}(G) \leq \alpha(1 - t), \text{ eğer } t \leq 1$$

Sonuç 4.1.8 [20] G olarak adlandırılan bir grafın, n dereceli ve bağımsızlık sayısında α olan tamamlanmamış bir graf için ortalama saçılma sayısı α değerinden küçük olmalıdır.

$$sc_{av}(G) \leq \alpha$$

Öneri 4.1.9 [20] G olarak adlandırılan bir graf, tamamlanmamış bir r -düzenli graf olduğu durumda ise bu senaryoda,

$$sc_{av}(G) \geq -r,$$

Lemma 4.1.10 [20] H , tamamlanmamış bağlantılı bir G grafının yayılan bir alt grafiği olsun bu durumda ise,

$$sc_{av}(H) \geq sc_{av}(G).$$

Teorem 4.1.11 [22] G olarak adlandırılan bir bağlantılı graf için mertebesi ($n \geq 3$) büyük olduğu durumlar için;

- a) En uzun yol p olarak isimlendirildiğinde $sc(G) \leq n - p$.
- b) $sc(G) \leq n - 2\kappa$.
- c) $sc(G) \leq \alpha - \kappa$.

Teorem 4.1.12 [20] G grafının $n \geq 4$ 'ten büyük olduğu durumda Teorem 5.11'e göre paylaşılan en uzun yol yapısı ile ortalama saçılma sayısı için bir sınır değer paylaşımı;

$$sc_{av}(G) \leq n - p.$$

Teorem 4.1.13 [20] G grafinın, n mertebesinde tamamlanmamış bir graf olduğu durumda,

$$sc_{av}(G) \leq n - 2\kappa.$$

Teorem 4.1.14 [20] G , n dereceli, bağlantılılık κ ve bağımsızlık numarası α 'dan oluşan tamamlanmamış bir graf olduğu durumda ;

$$Sc_{av}(G) \leq \alpha - \kappa.$$

Paylaşılan teoremlerden faydalanılarak hem ortalama saçılma sayısı hesabı hem de bu konu kapsamında algoritmalar ve ortalama saçılma sayısı hesabı için bir yazılım paylaşılacaktır.

4.2. Çizge Yapılarının Ortalama Saçılma Sayıları

Bu bölüm içerisinde bazı grafların ortalama saçılma sayısı aşağıda paylaşılmıştır.

Teorem 4.2.1 [20] T , $n \geq 4$ mertebesinde bir tree (ağaç) olduğu varsayılın. Bu ağacın yaprak sayısında k olduğu durumda;

$$scav(T) \geq \frac{n - k}{n}$$

Teorem 4.2.2 [20] P_n , $n \geq 4$ düzeyinde bir yol grafi olduğu durumda

$$scav(P_n) \geq \frac{n - 2}{n}$$

Teorem 4.2.1'den faydalanılarak P_n yolunun $n \geq 4$ düzeyindeki tüm ağaçlar (trees) arasında minumum ortalama saçılma sayısına sahip olduğu görülebilmektedir.

Teorem 4.2.3 C_n $n \geq 4$ düzeyinde bir döngü grafi ise böyle bir durumda;

$$sc_{av}(C_n) = 0.$$

Teorem 4.2.4 Eğer $K_{a,b}$ $a + b$ mertebesinde tam iki parçalı bir grafik ise ($2 \leq a \leq b$), o zaman

$$sc_{av}(K_{a,b}) = \begin{cases} \frac{b^2 - a^2 - 2b}{a+b}, & \text{SINIR değerleri } a < b \\ 0, & \text{SINIR değerleri } a = b \end{cases}$$

Teorem 4.2.5 [20] $K_{1,n-1}$ ($n \geq 4$) yıldızının ortalama saçılımı $\frac{n^2 - 4n + 2}{n}$ olarak hesaplanabilir.

Teorem 4.2.6 [20] ile ortalama saçılma sayısı hesabı için istikrarlı sonuç elde edilmesinde fayda sağlayacaktır.

4.3 Graflarda Ortalama Scattering Sayısı Algoritması

The Avarage Scattering Number Calculation (scattering_number)

Method: avg_scattering_number(G)

1. Input: Yol Grafi
2. Atanan path graph(G) number_of_nodes() fonksiyonu ile tepe sayısı alınır.
3. Yol grafin atanmış olan değeri (n) 0 eşitlik kontrolü yap. Eğer 0 ise ortalama saçılma sayısı hesaplanmaz.
4. Her bir tepe için hesaplanan ortalama saçılma sayıları için bir list tanımla (max_use_list[]). Listenin index hesabı için tepe sayısını kullanılır.
5. Döngü n defa yürütüleceği ve k'nın 0 ile n-1 arasında değerler alacağı bir iterasyon oluşturulur.
 - 5.1 Her Si iterasyon elemanı için;

- 5.1.1 Atanan G grafinin bir kopyasını al. Select edilmiş olan S tepelerini Graftan çıkartılır. (remove_nodes_from(S)).
- 5.1.2 Grafta kalan tepelerin bağlı olan componentlerini w değişkenine atanır.
- 5.1.3 Seçilen yaprakların boyutunu (len(S)) negatif kontrolü yapılır. len(S) seçilen_s_yaprak değişkenine atanır.
- 5.1.4 Saçılma sayısı hesabı (w – seçilen_s_yaprak) yapılır.
- 5.1.5 Her Si iterasyon elamanı için tanımlı index değeri tanımlanır.
- 5.2.2 Her tepe için tanımlı index değerlerine o tepe için hesaplanmış maksimum saçılma sayısı değeri atanır.
6. Graftaki tüm tepeler için hesaplanan maksimum saçılma sayılarından atama yapılmayan indexte saçılma sayısı değeri olarak 0 değeri atanır.
- 6.1 Graftaki tüm tepeler için hesaplanan maksimum saçılma sayıları toplanır.
7. Tüm tepeler için hesaplanan maksimum saçılma sayılarının toplamını graftaki tepe sayısına bölünür. (sc_toplam / G.number_of_nodes())
8. Tanımlı fonksiyon sonucunda ortalama saçılma sayısı için hesaplanan değeri işlem sonucunda return edilir ve ortalama saçılma sayısı olarak atanır.
9. Çıktı: avg_scattering_number(G) fonksiyonu sonucu Yol (Path) grafinin ortalama saçılma sayısını return eder.

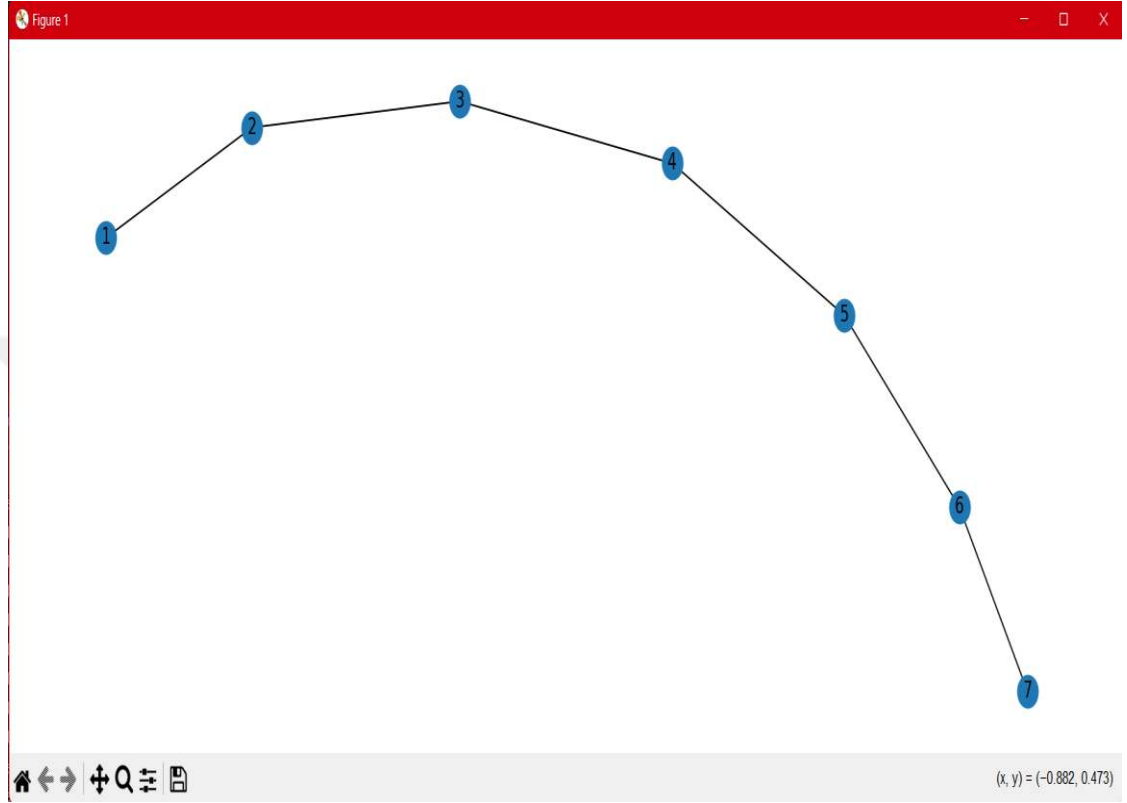
Algoritma hazırlığında Fengwei Li ve ark'nın interval graflarda ağırlıklı saçılma sayısı hesabı için hazırladığı algorithmadan faydalanılmıştır [24].

Ortalama Saçılma sayısı için hazırlanan uygulama ile irtibatlı algoritma yukarıda paylaşılmıştır. 4.3 Graflarda Ortalama Scattering Sayısı Algoritması başlıklı algoritma $O(n^3)$ olarak hesaplanmış olur. Ortalama saçılma sayısı için hazırlanan uygulama iç içe 3 for döngüsüne sahiptir. Buna bağlı olarak da n^3 değeri hesaplanır. Hesaplanan bu değer algoritmanın toplam maliyeti hakkında bilgi vermektedir.

4.3.1 Graflarda Ortalama Scattering Sayısı Örneği ve Sonuç Tabloları

Ortalama saçılma sayısının hesabı için örnek bir uygulama gerçekleştirildi. Uygulamada yol graf ele alındı. P₇ grafinin Pycharm Idesi üzerinde oluşturulmuş görseli Şekil 8.2 gösterilmiştir. Uygulamada belirtilen tepeler ve isimleri görülmektedir.

Yol grafinin scattering sayısı ve ortalama scattering sayısı hesabı yapıldı. Hesaplanan bu değerler tablo haline getirildi. Koşul olarak teoremlerde belirtilmiş olan $w>1$ değerlerine uymayan değerlerde tabloda sunulmaktadır. Fakat elde edilmiş olan bu sonuçlar **işleme alınmamıştır.**



Şekil 4.4 P_7 Yol Grafi Gösterimi

Tablo 2. G çizgesi için Ortalama Saçılma Sayısı Hesaplaması (S_1 iterasyonu)

S_1 Çıkarılan Değerler	Kalan Komponentler	$W(G-S)$	S	Sc
{1}	{2, 3, 4, 5, 6, 7}	1	1	0
{1, 2}	{3, 4, 5, 6, 7}	1	2	-1
{1, 3}	{2} {4, 5, 6, 7}	2	2	0
{1, 4}	{2, 3} {5, 6, 7}	2	2	0
{1, 5}	{2, 3, 4} {6, 7}	2	2	0

{1, 6}	{2, 3, 4, 5} {7}	2	2	0
{1, 7}	{2, 3, 4, 5, 6}	1	2	-1
{1, 2, 3}	{4, 5, 6, 7}	1	3	-2
{1, 2, 4}	{3} {5, 6, 7}	2	3	-1
{1, 2, 5}	{3, 4} {6, 7}	2	3	-1
{1, 2, 6}	{3, 4, 5} {7}	2	3	-1
{1, 2, 7}	{3, 4, 5, 6}	1	3	-2
{1, 3, 4}	{2} {5, 6, 7}	2	3	-1
{1, 3, 5}	{2} {4} {6, 7}	3	3	-1
{1, 3, 6}	{2} {4, 5} {7}	3	3	0
{1, 3, 7}	{2} {4, 5, 6}	2	3	-1
{1, 4, 5}	{2, 3} {6, 7}	2	3	-1
{1, 4, 6}	{2, 3} {5} {7}	3	3	0
{1, 4, 7}	{2, 3} {5, 6}	2	3	-1
{1, 5, 6}	{2, 3, 4} {7}	2	3	-1
{1, 5, 7}	{2, 3, 4} {6}	2	3	-1
{1, 6, 7}	{2, 3, 4, 5}	1	3	-2
{1, 2, 3, 4}	{5, 6, 7}	1	4	-3
{1, 2, 3, 5}	{4} {6, 7}	2	4	-2
{1, 2, 3, 6}	{4, 5} {7}	2	4	-2
{1, 2, 3, 7}	{4, 5, 6}	1	4	-3

{1, 2, 4, 5}	{3} {6, 7}	2	4	−2
{1, 2, 4, 6}	{3} {5} {7}	3	4	−1
{1, 2, 4, 7}	{3} {5, 6}	2	4	−2
{1, 2, 5, 6}	{3, 4} {7}	2	4	−2
{1, 2, 5, 7}	{3, 4} {6}	2	4	−2
{1, 2, 6, 7}	{3, 4, 5}	1	4	−3
{1, 3, 4, 5}	{2} {6, 7}	2	4	−2
{1, 3, 4, 6}	{2} {5} {7}	3	4	−1
{1, 3, 4, 7}	{2} {5, 6}	2	4	−2
{1, 3, 5, 6}	{2} {4} {7}	3	4	−1
{1, 3, 5, 7}	{2} {4} {6}	3	4	−1
{1, 3, 6, 7}	{2} {4, 5}	2	4	−2
{1, 4, 5, 6}	{2, 3} {7}	2	4	−2
{1, 4, 5, 7}	{2, 3} {6}	2	4	−2
{1, 5, 6, 7}	{2, 3, 4}	1	4	−3
{1, 2, 3, 4, 5}	{6, 7}	1	5	−4
{1, 2, 3, 4, 6}	{5} {7}	2	5	−3
{1, 2, 3, 4, 7}	{5, 6}	2	5	−3
{1, 2, 3, 5, 6}	{4} {7}	2	5	−3

{1, 2, 3, 5, 7}	{4} {6}	2	5	-3
{1, 2, 3, 6, 7}	{4, 5}	1	5	-4
{1, 2, 4, 5, 6}	{3} {7}	2	5	-3
{1, 2, 4, 5, 7}	{3} {6}	2	5	-3
{1, 2, 4, 6, 7}	{3} {5}	2	5	-3
{1, 2, 5, 6, 7}	{3, 4}	1	5	-4
{1, 3, 4, 5, 6}	{2} {7}	2	5	-3
{1, 3, 4, 5, 7}	{2} {6}	2	5	-3
{1, 3, 4, 6, 7}	{2} {5}	2	5	-3
{1, 3, 5, 6, 7}	{2} {4}	2	5	-3
{1, 4, 5, 6, 7}	{2, 3}	1	5	-4
{1, 2, 3, 4, 5, 6}	{7}	1	6	-5
{1, 2, 3, 4, 5, 7}	{6}	1	6	-5
{1, 2, 3, 4, 6, 7}	{5}	1	6	-5
{1, 2, 3, 5, 6, 7}	{4}	1	6	-5
{1, 2, 4, 5, 6, 7}	{3}	1	6	-5
{1, 3, 4, 5, 6, 7}	{2}	1	6	-5
				Sc_{Max} = 0

Tablo 3. G çizgesi için Ortalama Saçılma Sayısı Hesaplaması (S₂ iterasyonu)

S ₂ Çıkarılan Değerler	Kalan Komponentler	W(G-S)	S	Sc
{2}	{1} {3, 4, 5, 6, 7}	2	1	1

{2, 3}	{1} {4, 5, 6, 7}	2	2	0
{2, 4}	{1} {3} {5, 6, 7}	3	2	1
{2, 5}	{1} {3, 4} {6, 7}	3	2	1
{2, 6}	{1} {3, 4, 5} {7}	3	2	1
{2, 3, 4}	{1} {5, 6, 7}	2	3	0
{2, 3, 5}	{1} {4} {6, 7}	3	3	0
{2, 3, 6}	{1} {4, 5} {7}	3	3	0
{2, 3, 7}	{1} {4, 5, 6}	2	3	-1
{2, 4, 5}	{1} {3} {6, 7}	3	3	0
{2, 4, 6}	{1} {3} {5} {7}	4	3	1
{2, 4, 7}	{1} {3} {5, 6}	3	3	0
{2, 5, 6}	{1} {3, 4} {7}	3	3	0
{2, 5, 7}	{1} {3, 4} {6}	3	3	0
{2, 6, 7}	{1} {3, 4, 5}	2	3	-1
{2, 3, 4, 5}	{1} {6, 7}	2	4	-2
{2, 3, 4, 6}	{1} {5} {7}	3	4	-1

{2, 3, 4, 7}	{1}	2	4	−2
	{5, 6}			
{2, 3, 5, 6}	{1}	3	4	−1
	{4}			
	{7}			
{2, 3, 5, 7}	{1}	3	4	−1
	{4}			
	{6}			
{2, 3, 6, 7}	{1}	2	4	−2
	{4, 5}			
{2, 4, 5, 6}	{1}	3	4	−1
	{3}			
	{7}			
{2, 4, 5, 7}	{1}	3	4	−1
	{3}			
	{6}			
{2, 4, 6, 7}	{1}	3	4	−1
	{3}			
	{5}			
{2, 5, 6, 7}	{1}	2	4	−2
	{3, 4}			
{2, 3, 4, 5, 6}	{1}	2	5	−3
	{7}			
{2, 3, 4, 5, 7}	{1}	2	5	−3
	{6}			
{2, 3, 4, 6, 7}	{1}	2	5	−3
	{5}			
{2, 3, 5, 6, 7}	{1}	2	5	−3
	{4}			
{2, 4, 5, 6, 7}	{1}	2	5	−3
	{3}			
{2, 3, 4, 5, 6, 7}	{1}	1	6	−5
				SC_{Max} = 1

Tablo 4. G çizgesi için Ortalama Saçılma Sayısı Hesaplaması (S_3 iterasyonu)

S_3 Çıkartılan Değerler	Kalan Komponentler	W(G-S)	S	Sc
{3}	{1, 2} {4, 5, 6, 7}	2	1	1
{3, 4}	{1, 2} {5, 6, 7}	2	2	0
{3, 5}	{1, 2} {4} {6, 7}	3	2	1
{3, 6}	{1, 2} {4, 5} {7}	3	2	1
{3, 7}	{1, 2} {4, 5, 6}	2	2	0
{3, 4, 5}	{1, 2} {6, 7}	2	3	-1
{3, 4, 6}	{1, 2} {5} {7}	3	3	0
{3, 4, 7}	{1, 2} {5, 6}	2	3	-1
{3, 5, 6}	{1, 2} {4} {7}	3	3	0
{3, 5, 7}	{1, 2} {4} {6}	3	3	0
{3, 6, 7}	{1, 2} {4, 5}	2	3	-1
{3, 4, 5, 6}	{1, 2} {7}	2	4	-2
{3, 4, 5, 7}	{1, 2} {6}	2	4	-2
{3, 4, 6, 7}	{1, 2} {5}	2	4	-2
{3, 5, 6, 7}	{1, 2} {4}	2	4	-2
{3, 4, 5, 6, 7}	{1, 2}	1	5	-4
				Sc_{Max} = 1

Tablo 5. G çizgesi için Ortalama Saçılma Sayısı Hesaplaması (S₄ iterasyonu)

S ₄ Çıkarılan Değerler	Kalan Komponentler	W(G-S)	S	Sc
{4}	{1, 2, 3} {5, 6, 7}	2	1	1
{4, 5}	{1, 2, 3} {6, 7}	2	2	0
{4, 6}	{1, 2, 3} {5} {7}	3	2	1
{4, 7}	{1, 2, 3} {5, 6}	2	2	0
{4, 5, 6}	{1, 2, 3} {7}	2	3	-1
{4, 5, 7}	{1, 2, 3} {6}	2	3	-1
{4, 6, 7}	{1, 2, 3} {5}	2	3	-1
{4, 5, 6, 7}	{1, 2, 3}	1	4	-3
				Sc_{Max} = 1

Tablo 6. G çizgesi için Ortalama Saçılma Sayısı Hesaplaması (S₅ iterasyonu)

S ₅ Çıkarılan Değerler	Kalan Komponentler	W(G-S)	S	Sc
{5}	{1, 2, 3, 4} {6, 7}	2	1	1
{5, 6}	{1, 2, 3, 4} {7}	2	2	0
{5, 7}	{1, 2, 3, 4} {6}	2	2	0
{5, 6, 7}	{1, 2, 3, 4}	1	3	-2
				Sc_{Max} = 1

Tablo 7. G çizgesi için Ortalama Saçılma Sayısı Hesaplaması (S_6 iterasyonu)

S_6 Çıkartılan Değerler	Kalan Komponentler	$W(G-S)$	S	Sc
{6}	{1, 2, 3, 4, 5} {7}	2	1	1
{6, 7}	{1, 2, 3, 4, 5}	1	2	-1
$Sc_{Max} = 1$				

Tablo 8. G çizgesi için Ortalama Saçılma Sayısı Hesaplaması (S_7 iterasyonu)

S_7 Çıkartılan Değerler	Kalan Komponentler	$W(G-S)$	S	Sc
{7}	{1, 2, 3, 4, 5, 6}	1	1	0
$Sc_{Max} = 0$				

Yukarıda P_7 yol grafi için scattering sayısı hesabı yapılmıştır. Tablolarda $S_1, S_2, S_3, S_4, S_5, S_6, S_7$ değerleri Şekil 8.2 örneğindeki graf örneği referans alınarak elde edilmiştir. Matplotlib kütüphanesi yardımıyla Python programlama dili kapsamında sayısal bir matematik uzantısı olan Numpy için bir çizim kütüphanesidir. Grafiklerin oluşturulmasında ve yerleştirilmesi için nesne yönelimli bir API sağlamaktadır.

Tablo üzerinde $W(G-S)$ olarak tanımlanmış olan sütundaki değerlerin teoremlerde belirtildiği üzere $w > 1$ şartını sağlamak zorundadır. Sağlamayan değerler saçılma sayısı hesabına dâhil edilmemiştir. Yalnızca tablo üzerinde çıkartılmış olan değerlerin veya kalan komponentlerin tamamını görüntüleyebilmek için belirtilmiştir.

Ortalama saçılma sayısı örneği sonuç tablosu alt kısımda paylaşılmıştır. Sırayla $S_1, S_2, S_3, S_4, S_5, S_6, S_7$ içeren değerlerin graftan çıkartılmasıyla hesaplanmış olan saçılma sayısı değerlerinin ortalaması alınmıştır.

Tablo 9. G çizgesi için Ortalama Saçılma Sayısı Hesaplaması (Sonuç Tablosu)

ScS_1	ScS_2	ScS_3	ScS_4	ScS_5	ScS_6	ScS_7
0	1	1	1	1	1	0

$$AvgScNum = \frac{5}{7}$$

4.3.2 Graflarda Ortalama Scattering Sayısı Kod Yapısı ve Anlatımı

```
import networkx as nx
from itertools import combinations
from matplotlib import pyplot as plt

def avg_scattering_number(G):
    n = G.number_of_nodes()
    if n == 0:
        return 0
    max_sc = float('-inf')
    global sc_toplam
    global secilen_s_yaprak

    max_use_list = []

    for index in range(0, n):
        max_use_list.append(float('-inf'))

    # Kaldırılacak tüm düğümlerin alt kümelerini yinelenir.
    for k in range(1, n):
        for S in combinations(G.nodes(), k):
            G_eksi_S = G.copy()
            G_eksi_S.remove_nodes_from(S)

            w = nx.number_connected_components(G_eksi_S)
            components = list(nx.connected_components(G_eksi_S))

            if w >= 2:
                for i, component in enumerate(components, start=1):
                    print(f"Kalan Component {i}: {component}")
                    secilen_s_yaprak = len(S)
                    if secilen_s_yaprak < 0:
                        secilen_s_yaprak = -1 * secilen_s_yaprak
                    sc = w - secilen_s_yaprak
                    print("secilen s yaprak -->", secilen_s_yaprak)
                    print("Scattering sayisini ekrana bastir -->", sc)

    # Standart scattering number için kullanılan yapı.
    if sc > max_sc:
        max_sc = sc

    using_node = S[0]

    if max_use_list[using_node - 1] < sc:
        max_use_list[using_node - 1] = sc

    for num in max_use_list:
```

```

if num == float('-inf'):
    num = 0

sc_toplam += num

return sc_toplam / G.number_of_nodes()

# Ortalama Scattering Number Hesabı İçin Bir Örnek.

G = nx.Graph()
edges = [(1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7)]
G.add_edges_from(edges)

sc_toplam = 0
secilen_s_yaprak = 0

print("Ortalama Scattering number degeri:", avg_scattering_number(G))
print("Scattering number index bazli toplam :", sc_toplam)
print("Graf node sayisi :", G.number_of_nodes())

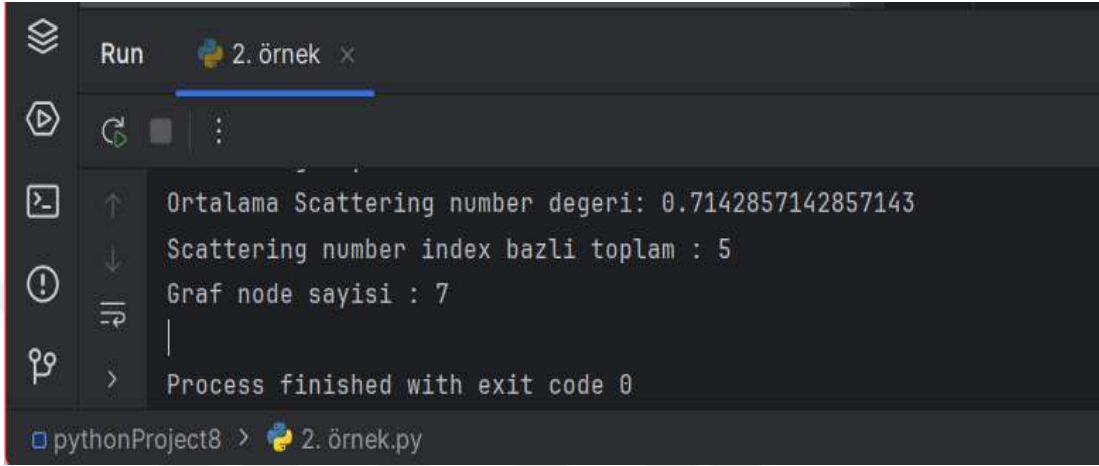
G = nx.from_edgelist(edges)
nx.draw_spring(G, with_labels=True)
plt.show()

```

Kod bloğunda P_7 yol grafinin ortalama saçılma sayısını hesaplayan program yer almaktadır. Oluşturulan programın yapısında saçılma sayısı hesabı için networkx kütüphanesi kullanılmıştır. Graflarda zedelenebilirlik hesapları için oldukça faydalı olan bu kütüphane sayesinde belirtilen tepeler ile saçılma sayısı hesabı yapılmıştır. Matplotlib ve pyplot kütüphaneleri sayesinde de graf modellemesi yapılmıştır. Bu sayede öncesinde belirlenen noktaların birbirlerine olan bağları ve sonuç olarak çıkan graf görüntülenebilmektedir.

Scattering_number fonksiyonu öncesinde noktaları belirlenen G grafi gönderilir. Tepe kontrolü yapılır. Tepe sayısı 0'a eşit olmamalıdır. Gerekli tanımlamalar yapıldıktan sonra max_use_list.append dizisi tanımlanır. Bu tanımlama sayesinde index bazlı hesaplanan o tepeye özgü maksimum saçılma sayıları ele alınır. Networkx kütüphanesi sayesinde kullanılabilen remove_nodes_from fonksiyonu ile öncesinde kopyası oluşturulmuş P_7 grafindan tepe silme işlemini gerçekleştirilecektir. Graftan tepe çıkarıldıktan sonra connected_components fonksiyonu sayesinde kalan tepelerin birbiriyle olan bağları ve sonuç olarak w değeri elde edilmiş olacaktır. Teoremden yola çıkılarak w değerinin 2'den büyük olması gerekmektedir. Teoreme

göre hareket edildiği için seçilen yaprakların sayısı 0'dan küçük olmaması gerektiği için de if (secilen_s_yaprak < 0) sorgusu eklenmiştir. Son olarak ortalama saçılma sayısı hesabı için Indexe göre analiz edilen max saçılma değerleri her bir yaprak için toplanır ve number_of_nodes'a (toplam tepe sayısı) bölünür. Bu sayede ortalama saçılma sayısı hesabı gerçekleştirilmiş olacaktır.



```
Run 2. örnek x
Ortalama Scattering number degeri: 0.7142857142857143
Scattering number index bazli toplam : 5
Graf node sayisi : 7
Process finished with exit code 0
pythonProject8 > 2. örnek.py
```

Şekil 4.5 Uygulamanın Çıktısı

5 AĞIRLIKLİ GRAFLARDA ORTALAMA SAÇILMA SAYISI

Bu bölümde ağırlıklı graflarda ortalama saçılma sayısı tanımları sunulacaktır. Ağırlıklı ortalama saçılma sayısı parametresinin tanımı verilecektir. Graflarda bu parametre üzerine oluşturulan algoritma ve zaman karmaşıkları sunulacaktır.

5.1 Tanım ve Motivasyon

Graflarda ortalama saçılma sayısı tanımı ve hesabından yola çıkılarak, buna ek bir sistematik ağırlıklı graflarda ortalama saçılma sayısı tanımı ve teoremlerine bu kısımda yer verilecektir.

Tanım 5.1.1 [24] Bir tam olmayan graf için ağırlıklı saçılma sayısı;

$$sw(G) = \max \{c(G - X) - w(X) : X \subset V(G), c(G - X) \geq 2\}$$

Bu durumda $w : V \rightarrow R +$ pozitif tepe ağırlıkları fonksiyonu ve $R +$ pozitif rasyonel sayıların kümesi olarak ifade edilir. Burada $w(x) = \sum_{v \in X} w(v)$ ve G grafi için en büyük kesim kümeleridir ve $c(G-X)$ geriye kalan graftaki bileşen sayısıdır.

Burada daha önce tanımlanan ağırlıklı saçılma sayısı ve ortalama saçılma sayısı birleştirilerek ağırlıklı ortalama saçılma sayısı verilecektir.

Öncelikle lokal ağırlıklı saçılma sayısının tanımını verelim. v tepesi bağlantılı ağırlıklı G grafinin bir tepesi olmak üzere,

$$Sc_v w(G) = \max\{c(G - S_v) - w|S_v|, c(G - S_v) \geq 2\}$$

$w|S_v| = \sum_{v \in V(G)} w(S_v)$, G grafindan çıkartılan tepelerin toplam ağırlığını ifade etmektedir. Burada S_v , v tepesinin dahil olduğu kesim kümeyi ve $c(G - S_v)$, geriye kalan graftaki bileşen sayısıdır.

Ağırlıklı bir G grafinda ağırlıklı graflarda ortalama saçılma sayısı $sc_{av} w(G)$ ile gösterilir ve tanımı,

$$sc_{av} w(G) = \frac{\sum_{v \in V(G)} sc_v w(G)}{n}$$

Burada $Sc_v w(G)$, lokal ağırlıklı saçılma sayısı ve n , tepe derecesidir. Ağırlıklı grafta tüm tepeler için hesaplanan lokal maksimum saçılma sayıları toplanır. Toplanan saçılma sayıları ağırlıklı grafin tepe sayısına bölünür. Sonuç olarak Ağırlıklı Grafta Ortalama Saçılma Sayısı hesaplanmış olacaktır.

Ağırlıklı grafta ortalama saçılma sayısı hesabı yapılırken ağırlıklı olmadığı duruma ek olarak çıkartılan tepe sayısı (S_v)'nin yerine çıkartılan tepenin ağırlıklarının toplamı kullanılmaktadır.

Ortalama parametrelerin bazı durum veya olaylarda en kötü senaryoya karşın daha büyük bir direnç gösterdiği ve optimizasyon algoritmalarında da verisel olarak hesaplanıp parametresel olarak değerlendirilmektedir [19]. Bu netice kapsamında

saçılma sayısı kavramı ile birlikte literatüre yeni bir parametre olarak ortalama saçılma sayısı adı verilen yeni bir graf parametresi önerilmiştir. Bu zedelenebilirlik parametresi literatürde E Aslan ve arkadaşları tarafından da incelenmiş ve analizlerde bulunulmuştur [20]. Bunun ile birlikte yeni bir tanım ve analizlere ek olarak ağırlıklı graflar için ortalama saçılma sayısı yukarıda tanımı ve teoremleri paylaşılmıştır.

5.2 Ağırlıklı Grafta Ortalama Saçılma Sayısı Örneği

Ağırlıklı graflarda ortalama saçılma sayısı hesabı için örnek bir hesaplama yapılacaktır. Örnek olarak ele alınan yol graf olacaktır. Bu örnek kapsamında yol grafin tepe sayısı 4 olarak belirlenmiştir.



Şekil 5.1 P₄ Yol Graf Gösterimi

Ağırlıklı grafta ortalama saçılma sayısı hesabı yapılırken öncelikle genel ağırlık değeri belirlenir. Bu her tepe noktası için ayrı ayrı da belirtilebilir. Bu örnek kapsamında her tepe noktası için 5 değeri ağırlık olarak atanır. Bu örnekteki asıl amaç teoremler ışığında sözel bir şekilde hesaplamayı göstermektir. Detaylı örneğe uygulama içerisinde yer verilmiştir.

Numaralandırma soldan sağa 1-2-3-4 olarak tepe noktaları isimlendirilir. Saçılma sayısı hesaplanırken her bir tepe noktasının graftan tek tek çıkartılması gerekmektedir. Teoremlerde belirtilen şartlara uygun değerler hesaplamaya dahil edilecektir.

Ağırlıklı graflarda ortalama saçılma sayısı hesabında belirtilen örnek kapsamında teoremler ele alınarak yalnızca hesaba katılacak durumlar ele alınmıştır. Hesaplanmış olan maksimum saçılma sayıları üzerinden işlemler yapılmıştır.

1- $|S_1| =$ olduğu durumda S kümesi şu şekilde seçilebilir:

- $S=\{S_1\}$, Kalan Komponentler $\rightarrow \{S_2, S_3, S_4\}$ ağırlıklı grafta ortalama saçılma sayısının hesaplanabilmesi için $\omega(G-S) > 1$ olduğu için bu kümeler değerlendirmenin dışında kalacaktır.
- $S=\{S_1, S_3\}$ Kalan Komponentler $\rightarrow \{2\}, \{4\}$ bu durumda $\omega(G-S) = 2$ olacaktır.
- Ağırlık değeri 5 olarak atandığı için bu durumda saçılma sayısı $sc = 2 - 10 = -8$ olacaktır.

2- $|S_2|$ olduğu durumda S kümesi şu şekilde seçilebilir:

- $S=\{S_2\}$, Kalan Componentler $\rightarrow \{1\}, \{3, 4\}$ bu durumda $\omega(G-S) = 2$ olacaktır.
- Ağırlık değeri 5 olarak atandığı için bu durumda saçılma sayısı $sc = 2 - 5 = -3$ olacaktır.

3- $|S_3|$ olduğu durumda S kümesi şu şekilde seçilebilir:

- $S=\{S_3\}$, Kalan Componentler $\rightarrow \{1,2\}, \{4\}$ bu durumda $\omega(G-S) = 2$ olacaktır.
- Ağırlık değeri 5 olarak atandığı için bu durumda saçılma sayısı $sc = 2 - 5 = -3$ olacaktır.

4- $|S_4|$ olduğu durumda S kümesi şu şekilde seçilebilir:

- $S=\{S_4\}$, Kalan Komponentler $\rightarrow \{S_1, S_2, S_3\}$ ağırlıklı grafta ortalama saçılma sayısının hesaplanabilmesi için $\omega(G-S) > 1$ olduğu için bu kümeler değerlendirmenin dışında kalacaktır.

Ağırlıklı graflarda ortalama saçılma sayısı hesabı için hesaplanmış olan bu saçılma sayılarının toplanması gerekmektedir. $|S_1|$, $|S_2|$, $|S_3|$, $|S_4|$ sırasıyla -8, -3, -3 olarak hesaplanmıştır. Bu saçılma sayılarının toplamı;

$$Sc = (-8) + (-3) + (-3) = -14$$

Ağırlıklı graflarda ortalama saçılma sayısını hesaplayabilmek için toplanmış olan saçılma sayılarını tepe noktasına bölmektir.

Ağırlıklı Graflarda Ortalama Scattering Number Degeri: $\frac{-14}{4} = -3.5$

5.3 AĞIRLIKLI GRAFLARDA ORTALAMA SCATTERING SAYISI ALGORİTMASI

Bu bölüm kapsamında ağırlıklı graflarda ortalama saçılma sayısı (sc_{av}) algoritmasına yer verilmiştir. Scattering number algoritması ve ortalama scattering number algoritması temel olarak alındı. Algoritma içeriğinde Ağırlık ataması yapılmış bir grafin ortalama saçılma sayısının hesaplanmasına yaramaktadır. Algoritma içeriğinde Python kendi kütüphanesi olan networkx kütüphanesinden faydalanılmıştır. Ağırlık değeri graf içerisinde sabit tutuldu. Sabit tutulan bu değer kullanıcı tarafından değiştirilebilmektedir. Saçılma sayısı hesabı ya da algoritmasına ek olarak kullanılan ağırlıklı katsayı değeri yapıda kısmi değişikliklere sebep olmuştur.

Teorem açısından bakıldığında çıkartılan yaprakların sayısı olarak değerlendirilen yapı ağırlıklı katsayı sistemi devreye girdiğinde index bazlı dinamik bir yapıya evrildi. Ek olarak bazı index değerlerinden teoremler kapsamında herhangi bir atama yapılamamaktadır. Bu sebeple algoritma içeriğinde de belirtildiği gibi index toplamında etkisiz bir eleman olması açısından 0 değeri tahsis edildi. Uygulama kodlama üzerinde ve devamında detaylı bir şekilde açıklandı.

Method: `weighted_graph_avg_scattering_number(G)`

1. Input: Yol Grafi
2. Atanan `path graph(G) number_of_nodes()` fonksiyonu ile tepe sayısı alınır. (`len(self._node)`).
3. Ağırlık Yol grafin atanmış olan değeri (n) 0 eşitlik kontrolü yap. Eğer 0 ise ortalama saçılma sayısı hesaplanmaz.
4. Her bir tepe için hesaplanan ortalama saçılma sayıları için bir list tanımla (`max_use_list[]`). Listenin index hesabı için tepe sayısını kullanılır.
5. Döngü n defa yürütüleceği ve k 'nın 0 ile $n-1$ arasında değerler alacağı bir iterasyon oluşturulur.

5.1 Her S_i iterasyon elemanı için;

5.1.1 Atanan G ağırlıklı grafinin bir kopyasını al. Select edilmiş olan S tepelerin Graftan çıkartılır. (`remove_nodes_from(S)`).

5.1.2 Ağırlıklı Grafta kalan tepelerin bağlı olan componentlerini w değişkenine atanır.

- 5.1.3 Ağırlıklı Graftan çıkartılan her S elemanı için ağırlığı ile toplama işlemi yapılır (Dinamik yapı). Hesaplanan toplam ağırlık değeri seçilen_s_yaprak değişkenine atanır.
- 5.1.4 Seçilen yaprakların negatif kontrolü yapılır.
- 5.1.5 Saçılma sayısı hesabı ($w - \text{seçilen_s_yaprak}$) yapılır.
- 5.1.6 Her Si iterasyon elemanı için;
 - 5.1.6.1 Tanımlı grafta index değeri tanımlanır.
 - 5.1.6.2 Her tepe için tanımlı index değerlerine o tepe için hesaplanmış maksimum saçılma sayısı değeri atanır.
- 6. Graftaki tüm tepeler için hesaplanan maksimum saçılma sayılarından atama yapılmayan indexe max saçılma sayısı değeri olarak 0 değeri atanır.
- 6.1 Graftaki tüm tepeler için hesaplanan maksimum saçılma sayıları toplanır.
- 7. Graftaki tüm tepeler için hesaplanan maksimum saçılma sayılarını toplanır.
- 8. Tüm tepeler için hesaplanan maksimum saçılma sayılarının toplamını graftaki tepe sayısına bölünür. ($\text{sc_toplam} / \text{G.number_of_nodes}()$)
- 9. Tanımlı fonksiyon sonucunda ortalama saçılma sayısı için hesaplanan değeri işlem sonucunda return edilir ve ortalama saçılma sayısı olarak atanır.
- 10. Çıktı: `weighted_graph_avg_scattering_number(G)` fonksiyonu sonucu bir ağırlıklı grafinin ortalama saçılma sayısını return eder.

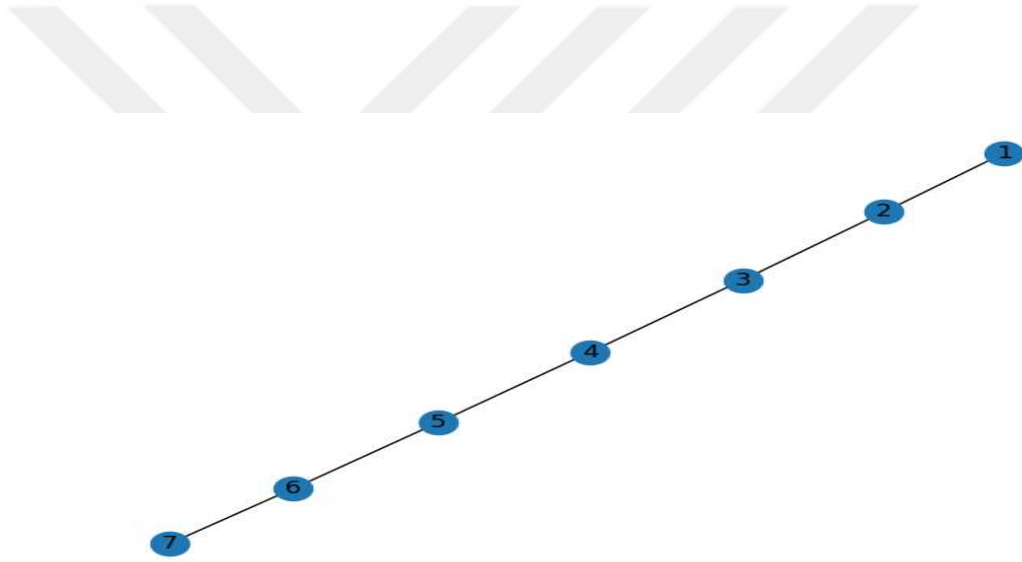
Ağırlıklı Graflarda Ortalama Saçılma sayısı için hazırlanan uygulama ile irtibatlı algoritma yukarıda paylaşılmıştır. 5.3 Ağırlıklı Graflarda Ortalama Scattering Sayısı Algoritması başlıklı algoritma $O(n^3)$ olarak hesaplanmış olur. Ağırlıklı Graflarda Ortalama saçılma sayısı için hazırlanan uygulama iç içe 3 for döngüsüne sahiptir. Buna bağlı olarakta n^3 değeri hesaplanır. Hesaplanan bu değer algoritmanın toplam maliyeti hakkında bilgi vermektedir.

5.3.1 Ağırlıklı Graflarda Ortalama Saçılma Sayısı Örneği ve Sonuç Tabloları

Ağırlıklı graflarda ortalama saçılma sayısı hesabı için örnek bir uygulama gerçekleştirildi. Uygulama Python uygulama dili ile yazıldı. Pycharm idesi kullanıldı. Uygulama P7 yol graf için gerçekleştirildi. Örnek uygulamanın pycharm ide çıktısı şekil 7.4 gösterildi. Uygulama sonucunda çıkan değerler tablo haline getirildi. Tabloların içeriğinde tepe olarak 1. 2. 3 .. vs her tepe için ayrı ayrı ifade edildi.

Çıkartılan değerler neticesinde kalan tepeler kalan komponentler sütununda yer verildi. $W(G-S)$ olarak ifade edilen kalan komponentin sayısıdır. Bu $W(G-S)$ değeri teorem bazlı bakıldığında 2'den büyük veya eşit olmalıdır. Bu sebeple bu uygulama kapsamında $w \geq 2$ değerleri listelenmiştir. Yalnızca 7. tepe değeri için tabloda $w = 1$ değeri yer almaktadır. Sonucun ne çıktığına bakılmaksızın harici bir atama olmadığı için 7. Index değeri 0 olarak set edildi. Son olarak ağırlıklı katsayı sistemi uygulama içerisine entegre edildi.

Bu uygulama üzerinde ağırlıklı katsayı değeri 10 olarak set edildi. Bu değer kullanıcı tarafından değiştirilebilmektedir. Tablo içeriğinde S ağırlıklı toplam olarak yer almaktadır. Bu değer her bir tepe için ayrı ayrı toplanmaktadır.



Şekil 5.2 Uygulamanın Çıktısı

Tablo 10. G çizgesi için Ağırlıklı Graflarda Ortalama Saçılma Sayısı Hesaplaması
(S_1 iterasyonu)

S_1 Çıkartılan Değerler	Kalan Komponentler	$W(G-S)$	S Ağırlıklı Toplam	Sc
{1, 3}	{2} {4, 5, 6, 7}	2	20	-18
{1, 4}	{2, 3} {5, 6, 7}	2	20	-18
{1, 5}	{2, 3, 4} {6, 7}	2	20	-18

{1, 6}	{2, 3, 4, 5} {7}	2	20	−18
{1, 2, 4}	{3} {5, 6, 7}	2	30	−28
{1, 2, 5}	{3, 4} {6, 7}	2	30	−28
{1, 2, 6}	{3, 4, 5} {7}	2	30	−28
{1, 3, 4}	{2} {5, 6, 7}	2	30	−28
{1, 3, 5}	{2} {4} {6, 7}	3	30	−27
{1, 3, 6}	{2} {4, 5} {7}	3	30	−27
{1, 3, 7}	{2} {4, 5, 6}	2	30	−28
{1, 4, 5}	{2, 3} {6, 7}	2	30	−28
{1, 4, 6}	{2, 3} {5} {7}	3	30	−27
{1, 4, 7}	{2, 3} {5, 6}	2	30	−28
{1, 5, 6}	{2, 3, 4} {7}	2	30	−28
{1, 5, 7}	{2, 3, 4} {6}	2	30	−28
{1, 2, 3, 5}	{4} {6, 7}	2	40	−38
{1, 2, 3, 6}	{4, 5} {7}	2	40	−38
{1, 2, 4, 5}	{3} {6, 7}	2	40	−38
{1, 2, 4, 6}	{3} {5} {7}	3	40	−37
{1, 2, 4, 7}	{3} {5, 6}	2	40	−38

{1, 2, 5, 6}	{3, 4} {7}	2	40	−38
{1, 2, 5, 7}	{3, 4} {6}	2	40	−38
{1, 3, 4, 5}	{2} {6, 7}	2	40	−38
{1, 3, 4, 6}	{2} {5} {7}	3	40	−37
{1, 3, 4, 7}	{2} {5, 6}	2	40	−38
{1, 3, 5, 6}	{2} {4} {7}	3	40	−37
{1, 3, 5, 7}	{2} {4} {6}	3	40	−37
{1, 3, 6, 7}	{2} {4, 5}	2	40	−38
{1, 4, 5, 6}	{2, 3} {7}	2	40	−38
{1, 4, 5, 7}	{2, 3} {6}	2	40	−38
{1, 4, 6, 7}	{2, 3} {5}	2	40	−38
{1, 2, 3, 4, 6}	{5} {7}	2	50	−48
{1, 2, 3, 5, 6}	{4} {7}	2	50	−48
{1, 2, 3, 5, 7}	{4} {6}	2	50	−48
{1, 2, 4, 5, 6}	{3} {7}	2	50	−48
{1, 2, 4, 5, 7}	{3} {6}	2	50	−48
{1, 2, 4, 6, 7}	{3} {5}	2	50	−48
{1, 3, 4, 5, 6}	{2} {7}	2	50	−48

{1, 3, 4, 5, 7}	{2} {6}	2	50	-48
{1, 3, 4, 5, 7}	{2} {6}	2	50	-48
{1, 3, 4, 6, 7}	{2} {5}	2	50	-48
{1, 3, 5, 6, 7}	{2} {4}	2	50	-48
				Sc_{Max} = -18

Tablo 11. G çizgesi için Ağırlıklı Graflarda Ortalama Saçılma Sayısı Hesaplaması
(S₂ iterasyonu)

S ₂ Çıkarılan Değerler	Kalan Komponentler	W(G-S)	S Ağırlıklı Toplam	Sc
{2}	{1} {3, 4, 5, 6, 7}	2	10	-8
{2, 3}	{1} {4, 5, 6, 7}	2	20	-18
{2, 4}	{1} {3} {5, 6, 7}	3	20	-17
{2, 5}	{1} {3, 4} {6, 7}	3	20	-17
{2, 6}	{1} {3, 4, 5} {7}	3	20	-17
{2, 7}	{1} {3, 4, 5, 6}	2	20	-18
{2, 3, 4}	{1} {5, 6, 7}	2	30	-28
{2, 3, 5}	{1} {4} {6, 7}	3	30	-27
{2, 3, 6}	{1} {4, 5} {7}	3	30	-27
{2, 3, 7}	{1} {4, 5, 6}	2	30	-28

{2, 4, 5}	{1}	3	30	−27
	{3}			
	{6, 7}			
{2, 4, 6}	{1}	4	30	−26
	{3}			
	{5}			
	{7}			
{2, 4, 7}	{1}	3	30	−27
	{3}			
	{5, 6}			
{2, 5, 6}	{1}	3	30	−27
	{3, 4}			
	{7}			
{2, 5, 7}	{1}	3	30	−27
	{3, 4}			
	{6}			
{2, 6, 7}	{1}	2	30	−28
	{3, 4, 5}			
{2, 3, 4, 5}	{1}	2	40	−38
	{6, 7}			
{2, 3, 4, 6}	{1}	3	40	−37
	{5}			
	{7}			
{2, 3, 4, 7}	{1}	2	40	−38
	{5, 6}			
{2, 3, 5, 6}	{1}	3	40	−37
	{4}			
	{7}			
{2, 3, 5, 7}	{1}	3	40	−37
	{4}			
	{6}			
{2, 3, 6, 7}	{1}	2	40	−38
	{4, 5}			
{2, 4, 5, 6}	{1}	3	40	−37
	{3}			
	{7}			
{2, 4, 5, 7}	{1}	3	40	−37
	{3}			
	{6}			
{2, 4, 6, 7}	{1}	3	40	−37
	{3}			
	{5}			

{2, 5, 6, 7}	{1} {3, 4}	2	40	-38
{2, 3, 4, 5, 6}	{1} {7}	2	50	-48
{2, 3, 4, 5, 7}	{1} {6}	2	50	-48
{2, 3, 4, 6, 7}	{1} {5}	2	50	-48
{2, 3, 5, 6, 7}	{1} {4}	2	50	-48
{2, 4, 5, 6, 7}	{1} {3}	2	50	-48

Sc_{Max} = -8

Tablo 12. G çizgesi için Ağırlıklı Graflarda Ortalama Saçılma Sayısı Hesaplaması
(S₃ iterasyonu)

S ₃ Çıkarılan Değerler	Kalan Komponentler	W(G-S)	S Ağırlıklı Toplam	Sc
{3}	{1, 2} {4, 5, 6, 7}	2	10	-8
{3, 4}	{1, 2} {5, 6, 7}	2	20	-18
{3, 5}	{1, 2} {4} {6, 7}	3	20	-17
{3, 6}	{1, 2} {4, 5} {7}	3	20	-17
{3, 7}	{1, 2} {4, 5, 6}	2	20	-18
{3, 4, 5}	{1, 2} {6, 7}	2	30	-28
{3, 4, 6}	{1, 2} {5} {7}	3	30	-27
{3, 4, 7}	{1, 2} {5, 6}	2	30	-28

{3, 5, 6}	{1, 2} {4} {7}	3	30	-27
{3, 5, 7}	{1, 2} {4} {6}	3	30	-27
{3, 6, 7}	{1, 2} {4, 5}	2	30	-28
{3, 4, 5, 6}	{1, 2} {7}	2	40	-38
{3, 4, 5, 7}	{1, 2} {6}	2	40	-38
{3, 4, 6, 7}	{1, 2} {5}	2	40	-38
{3, 5, 6, 7}	{1, 2} {4}	2	40	-38

Sc_{Max} = -8

Tablo 13. G çizgesi için Ağırlıklı Graflarda Ortalama Saçılma Sayısı Hesaplaması
(S₄ iterasyonu)

S ₄ Çıkarılan Değerler	Kalan Komponentler	W(G-S)	S Ağırlıklı Toplam	Sc
{4}	{1, 2, 3} {5, 6, 7}	2	10	-8
{4, 5}	{1, 2, 3} {6, 7}	2	20	-18
{4, 6}	{1, 2, 3} {5} {7}	3	20	-17
{4, 7}	{1, 2, 3} {5, 6}	2	20	-18
{4, 5, 6}	{1, 2, 3} {7}	2	30	-28
{4, 5, 7}	{1, 2, 3} {6}	2	30	-28
{4, 6, 7}	{1, 2, 3} {5}	2	30	-28

Sc_{Max} = -8

Tablo 14. G çizgesi için Ağırlıklı Graflarda Ortalama Saçılma Sayısı Hesaplaması
(S₅ iterasyonu)

S ₅ Çıkartılan Değerler	Kalan Komponentler	W(G-S)	S Ağırlıklı Toplam	Sc
{5}	{1, 2, 3, 4} {6, 7}	2	10	-8
{5, 6}	{1, 2, 3, 4} {7}	2	20	-18
{5, 7}	{1, 2, 3, 4} {6}	2	20	-18
				Sc_{Max} = -8

Tablo 15. G çizgesi için Ağırlıklı Graflarda Ortalama Saçılma Sayısı Hesaplaması
(S₆ iterasyonu)

S ₆ Çıkartılan Değerler	Kalan Komponentler	W(G-S)	S Ağırlıklı Toplam	Sc
{6}	{1, 2, 3, 4, 5} {7}	2	10	-8
				Sc_{Max} = -8

Tablo 16. G çizgesi için Ağırlıklı Graflarda Ortalama Saçılma Sayısı Hesaplaması
(S₇ iterasyonu)

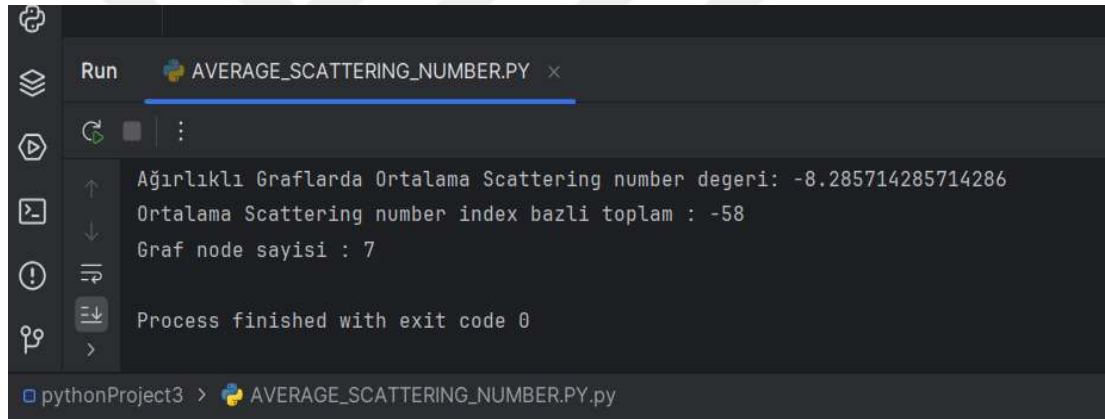
S ₇ Çıkartılan Değerler	Kalan Komponentler	W(G-S)	S	Sc
{7}	{1, 2, 3, 4, 5, 6}	1	10	-9
				Sc_{Max} = 0

Uygulama sonucunda oluşan sonuç tabloları paylaşılmıştır. 7. İndeks değeri için teorem ve koşulları sağlayan herhangi bir saçılma sayısı değeri yoktur. Kodlama içerisinde de belirtildiği gibi $w \geq 2$ olmalıdır. Tek bir durum olarak 7. Yaprığın çıkartılması olacaktır. Bu durum için herhangi bir saçılma sayısı değeri hesaba dahil edilemeyeceği için 7. İndeks için return eden maksimum saçılma sayısı 0 olacaktır.

Tablo 16. G çizgesi için Ağırlıklı Graflarda Ortalama Saçılma Sayısı Sonucu

W_1ScS_1	W_2ScS_2	W_3ScS_3	W_4ScS_4	W_5ScS_5	W_6ScS_6	W_7ScS_7
-18	-8	-8	-8	-8	-8	0
Weighted Avarage Scattering Number:				$\frac{-58}{7} = -8.28$		

Sonuç itibariyle Ağırlıklı graflarda ortalama saçılma sayısı hesaplanmıştır. Bu hesaba göre index bazlı ağırlık saçılma sayıları toplamı -58'dir. 7. İndeks üzerinden herhangi bir değer alınmadığı için 0 gelmektedir. İndexler arasında maksimum saçılma sayıları da yukarıdaki tabloda verilmiştir. Yol grafın örnekte 7 tepe ile yapıldığı için bu toplam 7'ye bölümüştür. Ağırlıklı katsayı değeri ile ortalama saçılma sayısı hesabı gerçekleştirilmiştir.



```
Run AVERAGE_SCATTERING_NUMBER.PY x
Ağırlıklı Graflarda Ortalama Scattering number degeri: -8.285714285714286
Ortalama Scattering number index bazli toplam : -58
Graf node sayisi : 7
Process finished with exit code 0
pythonProject3 > AVERAGE_SCATTERING_NUMBER.PY.py
```

Şekil 5.3 Ağırlıklı Grafta Ortalama Saçılma Sayısı Uygulaması Çıktısı

5.3.2 Ağırlıklı Graflarda Ortalama Scattering Sayısı Kod Yapısı ve Anlatımı

Python üzerinde gerçekleştirilen uygulama aşağıda yer almaktadır. Hazırlanan kod yapısında networkx kütüphanesi import edilmiştir. Grafların görsel olarak ifade edilebilmesi için de pyplot kütüphanelerinden faydalanılmıştır.

Hesaplama için `weighted_graph_avg_scattering_number` classı oluşturulmuştur. Parametre olarak main fonksiyonda oluşturulan graf ve `ağırlık_katsayi` değeri gönderilmiştir. Tanımlanmış olan grafın tepe sayısı eğer ki 0 değeri atandıysa herhangi bir saçılma sayısı hesabı yapılmadan 0 değeri döndürecektir.

Fonksiyon içerisinde ve main fonksiyonda kullanılmak üzere global değişkenler ve bir dizi tanımlanmıştır. Maksimum saçılma sayısı hesabı için tanımlı olan G grafinin n tepe sayısı kadar bir dizi index ataması yapılır. Graftan tepe çıkartma işlemi ve tüm alt kümelerin incelenmesi için bir döngü tanımlaması yapıldı. Bu döngü içerisinde grafin bir adet kopyası alındı. `Remove_nodes_from` ile graftan sırasıyla düğümler çıkartılmaya başlandı. Remove işlemi tamamlanan grafların bağlı olan component sayıları `number_connected_components` fonksiyonu ile hesaplanacak ve w değişkenine atanacaktır. Teoremlerde belirtildiği gibi bağlı olan komponent sayısı 2 ve üzeri bir sayı olması gerekmektedir. Bu sebeple if koşulu olan $w \geq 2$: eklendi. Ağırlıklı graflarda hesaplama yapılabilmesi adına `agirlik_katsayi` değeri tanımlanmıştır. Bu değer bu uygulama özelinde 10 olarak belirlenmiştir. Çıkartılan her S değeri için bu işlem dinamik olarak gerçekleştirilir. Son durumda hesaplanan `agirlikli_deger` diğer uygulamalarda olduğu gibi `secilen_s_yaprak` değişkenine atanır. Teoremden belirtildiği üzere `secilen_s_yaprak` değeri 0 veya negatif değer olma ihtimali kontrol edildi. Böyle bir durumda pozitif değer döndürülmesi adına -1 ile çarpıldı. İşlem sonrası saçılma sayısı değeri hesaplanır. Standart saçılma sayısı hesabında maksimum saçılma sayısını return edilmesi yeterlidir.

Bu uygulama kapsamında Ağırlıklı graflar için ortalama saçılma sayısı hesabı için çıkartılan değerler bir dizi olarak döndüğü için dizide ilk değer `indexte` değerlendirilecek değer olacaktır. Index alındığı durumda o index için hesaplanmış olan maksimum saçılma sayısı değeri alınacaktır. Hesaplanan `sc` değeri `max_use_list[using_node - 1]` belirtilen `indexte`ki değerden büyük ise maksimum saçılma sayısı bulabilmek için hesaplanan son değer belirtilen `indexe` atanır. Bu sayede o index için hesaplanan maksimum saçılma sayısı değeri hesaplanmış olacaktır. Son olarak index için hesaplanan değerlerde belirtilen `indexe` uygun herhangi bir değer ataması yapılmadıysa genel saçılma sayısı toplamında etkisiz bir eleman olabilmesi için 0 değeri atanır.

Sonuç olarak return eden değer ağırlıklı graf için ortalama saçılma sayılarının toplamının grafin genel tepe sayısına `sc_toplam / G.number_of_nodes()` bölünmesi olacaktır. Ana fonksiyon içerisinde de gerekli printler ile sonuçları görebilmek mümkündür. Son olarak grafin modellenmesi adına `G = nx.from_edgelist(edges)`,

`nx.draw_spring` (`G,with_labels=True`), `plt.show()` fonksiyonlarından faydanlanılmıştır.

```
import networkx as nx
from itertools import combinations
from matplotlib import pyplot as plt

def weighted_graph_avg_scattering_number(G, agirlik_katsayi):
    n = G.number_of_nodes()
    if n == 0:
        return 0
    max_sc = float('-inf')
    global sc_toplam
    global secilen_s_yaprak

    max_use_list = []

    for index in range(0, n):
        max_use_list.append(float('-inf'))

    # Kaldırılacak tüm düğümlerin alt kümelerini yinelenir.
    for k in range(1, n):
        for S in combinations(G.nodes(), k):

            G_eksi_S = G.copy()
            G_eksi_S.remove_nodes_from(S)

            w = nx.number_connected_components(G_eksi_S)
            components = list(nx.connected_components(G_eksi_S))

            if w >= 2:
                for com, component in enumerate(components, start=1):
                    print(f"Kalan Component {com}: {component}")

                agirlikli_deger = 0
                for i in S:
                    agirlikli_deger += agirlik_katsayi

                secilen_s_yaprak = agirlikli_deger

                if secilen_s_yaprak < 0:
                    secilen_s_yaprak = -1 * secilen_s_yaprak

                sc = w - agirlikli_deger

                print("secilen s yaprak -->", secilen_s_yaprak)
                print("Scattering sayisini ekrana bastir -->", sc)

# Standart scattering number için kullanılan yapı.
```

```

        if sc > max_sc:
            max_sc = sc

        using_node = S[0]

        if max_use_list[using_node - 1] < sc:
            max_use_list[using_node - 1] = sc

    for num in max_use_list:
        if num == float('-inf'):
            num = 0

    sc_toplam += num

    return sc_toplam / G.number_of_nodes()

# Ağırlıklı Graflarda Ortalama Scattering Number Hesabı İçin Bir Örnek.
G = nx.Graph()
edges = [(1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7)]
G.add_edges_from(edges)

agirlik_katsayi = 10
sc_toplam = 0
secilen_s_yaprak = 0

print("Ağırlıklı Graflarda Ortalama Scattering number degeri:",
      weighted_graph_avg_scattering_number(G, agirlik_katsayi))
print("Ortalama Scattering number index bazli toplam :", sc_toplam)
print("Graf node sayisi :", G.number_of_nodes())

G = nx.from_edgelist(edges)
nx.draw_spring(G, with_labels=True)
plt.show()

```

Uygulama kapsamında ağırlıklı graflarda ortalama saçılma sayısı hesabı yapılmıştır. Ağırlık değeri olarak örnek kapsamında 10 değeri atanmıştır. Bu durum kullanıcı tarafı için değiştirebilir bir değerdir. Her tepe için ağırlık katsayı değeri sabit bir değere bağlıdır. Farklı örneklerde her tepe için farklı ağırlık değerlerine sahip uygulamalar gerçekleştirilebilir. Bu uygulama kapsamında sabit bir ağırlık değeri tüm tepeler için sağlanmıştır. Sonuç olarak ağırlıklı bir graf modeli için ortalama saçılma sayısı hesabı gerçekleştirilmiştir. Saçılma sayısına ek olarak ortalama saçılma sayısı hesabı gerçekleştirilirken teorem ve şartlara bağlı tepe atanamayan index değerleri 0'a eşitlenmiştir. Ortalama saçılma sayısı hesabında maksimum saçılma sayıları indexe göre toplandığı için 0 etkisiz eleman olacaktır.

SONUÇ

Çalışma kapsamında Ağlarda Zedelenebilirlik Parametreleri konusu için henüz yeni sayılabilecek ortalama saçılma sayısı parametresine dikkate çekilmiştir. Genel olarak bakıldığında ağların zedelenebilirliği ve buna bağlı örnekler sunulmuştur. Ortalama saçılma sayısı hakkında literatür analizi yapıldı ve sonuçlar paylaşıldı. Ortalama saçılma sayısı hesabı için bazı örnekler paylaşıldı ve sonuçlar analiz edildi. Graflar ve graf yapıları incelendi. Makale içeriğinde kullanılan bazı graf türleri tanıtıldı. Bu graf türlerinden aynı sıra ve aynı büyüklükteki graflar arasından daha stabil olan graf seçilmek istenirse bunun bir yolu, ortalama saçılma sayısı minimum olan grafi seçmektir. Ortalama saçılma sayısı minimum değerli grafin seçilmesi ağ zedelenebilirliğinin değerlendirilmesi açısından oldukça yol gösterici bilgiler vermektedir.

Bu tez içeriğinde ağırlıklı graflarda ortalama saçılma sayısı hesabı, algoritması ve uygulamalı kod yapısı gerçekleştirilmiştir. Buna bağlı olarak çıkan zaman karmaşıkları hesaplanmıştır. Ağırlıklı graflar için ortalama saçılma sayısı hesabında hazırlanan algoritma için $O(n^3)$ zaman karmaşıklığı hesaplanmıştır. Ağırlıksız olduğu durumda da ortalama saçılma sayısı hesabında $O(n^3)$ zaman karmaşıklığı hesaplanmıştır. Bunlara ek standart saçılma sayısı hesabı için hazırlanan algoritma da bu karmaşıklık değeri $O(n^2)$ olarak hesaplanmıştır. Saçılma sayısı hesabına kıyasla çıkan sonuçlar değerlendirildiğinde daha hassas bir sonuç alınabilmektedir.

Hazırlanan algorima reel hayata bakıldığında ağlarda zedelenebilirlik konusunda nasıl kullanılabildiği konusunda bilgi vermektedir. Ağlar için ağ modellenmesi oldukça önem arz etmektedir. Ağların tasarımı ağırlıklı graflar ile modellendi. Ağlarda zedelenebilirlik hesabı sonucunda maksimum kararlılık ve verimli olabilmesi beklenmektedir. Bu durum için ağırlıklı graflarda hesaplama için ortalama saçılma sayısı hesabı önerilmiştir. Sistem tasarımcısı ya da yöneticisi ağın güvenilirliği konusunda bir fikir sahibi olabilir. Ancak ölçüm ve algoritmik yapı sonucu değerlendirmeler zaman karmaşıklıkları sonucunda çıkan değerler ağın güvenilirliği ya da kararlılığı hakkında bilgi vermektedir. Ağırlıklı graflarda yapılan ortalama saçılma sayısı hesabı literatürde daha geniş bir kapsamda incelenmesi zedelenebilirlik hesabı için önem arz etmektedir.

KAYNAKLAR

- [1] Wheeler, F. P. (2000). Systems thinking, systems practice: includes a 30-year retrospective. *Journal of the Operational Research Society*, 51(5), 647-647.
- [2] Gross, J. L., Yellen, J., & Anderson, M. (2018). *Graph theory and its applications*. Chapman and Hall/CRC.
- [3] Barefoot, C. A., Entringer, R., & Swart, H. (1987). Vulnerability in graphs-a comparative survey. *J. Combin. Math. Combin. Comput*, 1(38), 13-22.
- [4] Boruzanlı, G. (2010). *Zedelenebilirlik ölçümleri üzerine* (Master's thesis, Ege Üniversitesi).
- [5] TOSUN, M.A. (2019). Ağlarda Güvenlik İçin Komşu Zedelenebilirlik Ölçümleri Üzerine, 1 s.
- [6] POLAT, D. (2016). Graflarda Zedelenebilirlik Ve Komşu Ayırıt Kopma Derecesi, 1 s.
- [7] Aslan, E. The average lower connectivity of graphs. *J Appl Math* 2014; 2014.
- [8] Aldous, J. M., & Wilson, R. J. (2003). *Graphs and applications: an introductory approach*. Springer Science & Business Media.
- [9] Harary, F. *Graph Theory*. Addison-Wesley Publishing, New York, USA, 1969, 274 s.
- [10] Bondy, J. A., & Murty, U. S. R. (1976). *Graph theory with applications* (Vol. 290). London: Macmillan.
- [11] Woodall, D. R. (1973). The binding number of a graph and its Anderson number. *Journal of Combinatorial Theory, Series B*, 15(3), 225-255.
- [12] Brouwer A. E., Haemers W., (2011). *Spectra of Graphs*. Springer. 8,9 s.
- [13] Bondy, J. A., & Murty, U. S. R. (1976). *Graph theory with applications* (Vol. 290). London: Macmillan.
- [14] Bagga, K. S., Beineke, L. W., Lipman, M. J., & Pippert, R. E. (1994). Edge-integrity: a survey. *Discrete mathematics*, 124(1-3), 3-12.
- [15] Chvátal, V. (1973). Tough graphs and Hamiltonian circuits. *Discrete Mathematics*, 5(3), 215-228.

- [16] Jung, H. A. (1978). On a class of posets and the corresponding comparability graphs. *Journal of Combinatorial Theory, Series B*, 24(2), 125-133.
- [17] Ouyang, K. Z., Ouyang, K. Y., & Yu, W. (1993). Relative breaktivity of graphs. *J. Lanzhou University (Natural Science)*, 29(3), 43-49.
- [18] Zhang, S., & Peng, S. (2004). Relationships between scattering number and other vulnerability parameters. *International Journal of Computer Mathematics*, 81(3), 291-298.
- [19] Henning, M. A., & Oellermann, O. R. (2004). The average connectivity of a digraph. *Discrete applied mathematics*, 140(1-3), 143-153.
- [20] Aslan, E., Kiliç, D., Yücalar, F., & Borandağ, E. (2016). The average scattering number of graphs. *RAIRO-Theoretical Informatics and Applications*, 50(3), 263-272.
- [21] Giakoumakis, V., Roussel, F., & Thuillier, H. (1997). Scattering number and modular decomposition. *Discrete Mathematics*, 165, 321-342.
- [22] Zhang, S., & Wang, Z. (2001). Scattering number in graphs. *Networks: An International Journal*, 37(2), 102-106.
- [23] Keskenler, M. F., & Keskenler, E. F. (2017). Geçmişten günümüze yapay sinir ağları ve tarihçesi. *Takvim-i Vekayi*, 5(2), 8-18.
- [24] Li, F., Zhang, X., & Broersma, H. (2019). A polynomial algorithm for weighted scattering number in interval graphs. *Discrete Applied Mathematics*, 264, 118-124.
- [25] Kaval, B. *Graflar ve scattering sayısı* (Master's thesis, Fen Bilimleri Enstitüsü).