

T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KARMAŞIK AĞLARDA HAKİM DÜĞÜMLERİN BELİRLENMESİ İÇİN YENİ
BİR YÖNTEM

DOKTORA TEZİ

Furkan ÖZTEMİZ

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Prof. Dr. Ali KARCI

EYLÜL 2021

T.C
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KARMAŞIK AĞLARDA HAKİM DÜĞÜMLERİN BELİRLENMESİ İÇİN YENİ
BİR YÖNTEM

DOKTORA TEZİ

Furkan ÖZTEMİZ
(36173619010)

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Prof. Dr. Ali KARCI

EYLÜL 2021

TEŐEKKÜR VE ÖNSÖZ

Doktora eęitimimde ve hayatımın her anında bana destek olan akademik bilgi ve tecrübelerinin yanı sıra hayat tecrübelerini ve bilgi birikimini sabır ve cömertlikle benimle paylaşan kıymetli danışman hocam Prof. Dr. Ali KARCI'ya,

Beni yetiřtiren, bugünlere gelmemi saęlayan, tez çalışmam süresince benden desteklerini esirgemeyen ve üzerimde büyük hakları olan anneme ve babama,

Tez çalışması süresince ihmal ettięim, sıkıntılarımı, mutluluklarımı ve hayatımı paylařtığım sevgili Eřim Ayře ÖZTEMİZ'e ve kızım Aysima'ya

teőekkür ederim.



ONUR SÖZÜ

Doktora tezi olarak sunduđum “Karmařık Ađlarda Hakim Dügümlerin Belirlenmesi için Yeni Bir Yöntem” başlıklı bu çalışmanın bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın tarafımdan yazıldığına ve yararlandığıın bütün kaynakların hem metin içinde hem de kaynakçada yöntemine uygun biçimde gösterilenlerden oluştuđunu belirtir, bunu onurumla doğrularım.

Furkan ÖZTEMİZ



İÇİNDEKİLER

TEŞEKKÜR VE ÖNSÖZ	i
ONUR SÖZÜ	ii
İÇİNDEKİLER.....	iii
ÇİZELGELER DİZİNİ.....	iv
ŞEKİLLER DİZİNİ.....	v
SEMBOLLER VE KISALTMALAR DİZİNİ.....	vii
ÖZET.....	viii
ABSTRACT	x
1. GİRİŞ	1
1.1 Tezin Amacı ve Kapsamı.....	2
1.2 Literatür	2
1.3 Tezin Yenilikçi Yönü	6
1.4 Tezin Organizasyonu	6
2. GRAF TEORİSİ VE BASKIN KÜMELER.....	8
2.1 Çizge Türleri.....	9
2.2 Kapsayan Ağaçlar	14
2.3 NP Complete Problem	15
2.4 Baskın Düğüm ve Hakim Küme Kavramları.....	15
3. SOSYAL AĞLAR.....	17
3.1 Merkezlilik Algoritmaları	17
3.1.1 Sayfa değeri algoritması (Pagerank)	17
3.1.2 Arasındalık algoritması (Betweenness).....	18
3.1.3 Özvektör algoritması (Eigenvector)	18
3.1.4 Yakınlık algoritması (Closeness)	19
3.1.5 Merkezlilik algoritma uygulamaları	19
3.2 Topluluk Tespiti Algoritmaları.....	37
3.3 Benzerlik Algoritmaları	40
3.3.1 Jaccard algoritması	41
3.3.2 Cosine algoritması	41
3.3.3 Euclidean algoritması	42
4. GELİŞTİRİLEN YÖNTEMLER.....	44
4.1 Düğüm Dereceleri.....	44
4.2 Karcı Maksimum Kapsayan Ağaç (Karcı Maximum Spanning Tree)	45
4.3 Temel Kesmeler.....	48
4.4 Düğüm Baskınlık Değeri(Karcı Centrality Value)	50
4.5 Hakim Küme Üyelerinin Tespiti (Dominating Set)	50
4.6 Sözde Kodlar	52
5. UYGULAMA SONUÇLARI	56
5.1 Özel Çizgelerde Hakim Küme Tespiti.....	56
5.2 Karma Çizgelerde Hakim Küme Tespiti	65
5.3 Izgara Çizgelerde Uygulanması.....	66
5.4 Dairesel Çizgelerde Uygulanması	69
5.5 Ağaç Çizgelerde Uygulanması	70
5.6 İki parçalı Çizgelerde Uygulanması	72
6. SONUÇ VE ÖNERİLER	74
KAYNAKLAR.....	74
ÖZGEÇMİŞ	80

ÇİZELGELER DİZİNİ

Çizelge 3.1 : Sayfa değeri algoritması merkezlilik değerleri.....	22
Çizelge 3.2 : Arasındalık algoritması merkezlilik değerleri	23
Çizelge 3.3 : Yakınlık algoritması merkezlilik değerleri.....	24
Çizelge 3.4 : Özvektör algoritması merkezlilik değerleri	25
Çizelge 3.5 : Merkezlilik ve hibrit algoritma sonuçları	26
Çizelge 3.6 : Özvektör merkezliliğe göre en etkin 10 kavşak noktası.....	30
Çizelge 3.7 : Sayfa değeri merkezliliğe göre en etkin 10 kavşak noktası.....	32
Çizelge 3.8 : En yoğun ve en baskın 10 kavşak noktası	33
Çizelge 3.9 : Tüm merkezlilik ölçütlerine ait en etkili 8 düğüm	35
Çizelge 3.10 : Tüm merkezlilik ölçütlerine ait en etkisiz 8 düğüm	35
Çizelge 3.11 : IDAP örnek veri seti	38
Çizelge 3.12 : Jaccard benzerlik sonuçları.....	41
Çizelge 3.13 : Cosine benzerlik sonuçları.....	42
Çizelge 3.14 : Euclidean benzerlik sonuçları.....	43
Çizelge 4.1 : Örnek çizgeye ait düğüm dereceleri	45
Çizelge 4.2 : Komşu düğümlerin güncel düğüm dereceleri.....	46
Çizelge 4.3 : 1.iterasyon komşu düğümlerin kuyrukdaki durumu	47
Çizelge 4.4 : 2. iterasyon komşu düğümlerin kuyrukdaki durumu	47
Çizelge 4.5 : Kmax ağacı düğüm dereceleri	47
Çizelge 4.6 : Tüm temel kesme düğüm dereceleri.....	49
Çizelge 4.7 : Düğüm baskınlık değeri (Γ).....	50
Çizelge 5.1 : A çizgesi düğüm baskınlık değerleri (Γ)	58
Çizelge 5.2 : A çizgesi 3.iterasyon düğüm baskınlık değerleri (Γ).....	60
Çizelge 5.3 : C çizgesi hakim küme üye tespit yöntemi	66
Çizelge 5.4 : Izgara (Grid) çizgeye ait baskınlık değerleri (Γ)	67
Çizelge 5.5 : Dairesel (Circle) çizgeye ait baskınlık değerleri (Γ)	69
Çizelge 5.6 : İki parçalı (Bipartite) çizgeye ait baskınlık değerleri (Γ)	72

ŞEKİLLER DİZİNİ

Şekil 2.1 : Çizge yapısı	8
Şekil 2.2 : Königsberg 7 köprü problemi	9
Şekil 2.3 : Yönlü çizge	9
Şekil 2.4 : Yönsüz çizge	9
Şekil 2.5 : Karışık çizge	10
Şekil 2.6 : Düzenli çizge	10
Şekil 2.7 : Yol çizgesi	11
Şekil 2.8 : İki parçalı çizge	11
Şekil 2.9 : Ağaç çizge	11
Şekil 2.10 : Yıldız çizge	12
Şekil 2.11 : Tam çizge	12
Şekil 2.12 : Tamamlayıcı çizge	12
Şekil 2.13 : Çevre çizge	13
Şekil 2.14 : Tekerlek çizge	13
Şekil 2.15 : Ağırlıklı çizge	14
Şekil 2.16 : Kapsayan ağaçlar	14
Şekil 2.17 : Karmaşıklık sınıfları	15
Şekil 2.18 : Hâkim ve Minimum hâkim kümeler	16
Şekil 3.1 : Malatya kent merkezindeki konutların dağılımı	20
Şekil 3.2 : Hane halkı nüfus dağılımı ve çizge yapısı	20
Şekil 3.3 : Kent merkezi cadde ve bulvarlar çizge yapısı	21
Şekil 3.4 : Ulaşım ağı çizge yapısı	21
Şekil 3.5 : Sayfa değeri merkezliliğe göre Battalgazi kavşağının çizge üzerindeki konumu	22
Şekil 3.6 : Arasındalık merkezliliğe göre Beydağı kavşağının çizge üzerindeki konumu	23
Şekil 3.7 : Yakınlık merkezliliğe göre İnönü_Sıtmapınarı kavşağının çizgedeki konumu	24
Şekil 3.8 : Özvektör merkezliliğe göre SivasYolu_Topsogut kavşağının çizgedeki konumu	25
Şekil 3.9 : Hibrit merkezliliğe göre Sanayi kavşağı konum	27
Şekil 3.10 : Hibrit merkezliliğe göre çizge yapısı	27
Şekil 3.11 : Yandex GPS veri izine göre yoğunluk haritası	28
Şekil 3.12 : Malatya kenti akıllı kavşak noktaları	29
Şekil 3.13 : Akıllı kavşak ulaşım ağı çizgesi	29
Şekil 3.14 : Ulaşım ağı çizgesi yakınlaştırılmış hâli	30
Şekil 3.15 : Özvektör algoritmasına göre ulaşım ağı kavşak noktaları	31
Şekil 3.16 : Özvektör algoritmasına göre ulaşım ağı kavşak noktaları yakınlaştırılmış hâli	31
Şekil 3.17 : Sayfa değeri algoritmasına göre ulaşım ağı kavşak noktaları	32
Şekil 3.18 : Sayfa değeri algoritmasına göre ulaşım ağı kavşak noktaları yakınlaştırılmış hâli	33
Şekil 3.19 : 70 düğümlü çizge	34
Şekil 3.20 : Karci(Γ) merkezliliğin örnek çizge üzerindeki etkinlik(baskınlık) görünümü	36
Şekil 3.21 : IDAP Verilerine ait çizge yapısı	37
Şekil 3.22 : IDAP çizgesi yakınlaştırılmış hâli	38
Şekil 3.23 : Çizge üzerindeki örnek bir topluluk tespiti kesiti	39

Şekil 3.24 : Çizge üzerinde İBM akademisyenlerinden bazılarına ait bağlantılar.....	40
Şekil 4.1 : Örnek çizge	44
Şekil 4.2 : Kmax ağacı aşamaları	46
Şekil 4.3 : Örnek çizgeye ait Kmax ağacı ve tüm kesmeler.....	48
Şekil 4.4 : Hakim küme iterasyonları.....	51
Şekil 4.5 : Hakim küme (DS) üyeleri.....	52
Şekil 5.1 : A çizgesi	56
Şekil 5.2 : A çizgesine ait Kmax ağacı	57
Şekil 5.3 : A çizgesine ait tüm temel kesmeler	57
Şekil 5.4 : A çizgesi hakim küme algoritması 1.iterasyon.....	59
Şekil 5.5 : A çizgesi hakim küme algoritması 2.iterasyon.....	59
Şekil 5.6 : 3. iterasyon için çizgenin güncel hâli.....	60
Şekil 5.7 : Hakim küme 3. iterasyon.....	60
Şekil 5.8 : A çizgesi hakim küme (DS) algoritması 3. iterasyon	61
Şekil 5.9 : Hakim küme iterasyonları.....	61
Şekil 5.10 : B çizgesi.....	62
Şekil 5.11 : B çizgesine ait Kmax ağacı.....	62
Şekil 5.12 : B çizgesi DS algoritması 1. iterasyon.....	63
Şekil 5.13 : B çizgesi DS algoritması 2. iterasyon.....	63
Şekil 5.14 : B çizgesi DS algoritması 3. iterasyon.....	64
Şekil 5.15 : B çizgesi DS iterasyonlar.....	64
Şekil 5.16 : C çizgesinin baskın ve komşu düğümleri	65
Şekil 5.17 : Izgara çizge	66
Şekil 5.18 : Izgara çizge DS algoritması 1. iterasyon	67
Şekil 5.19 : Izgara çizge hakim küme iterasyonları	68
Şekil 5.20 : Izgara çizge hakim küme üyeleri	68
Şekil 5.21 : Dairesel çizge.....	69
Şekil 5.22 : Dairesel çizge hakim küme iterasyonları.....	70
Şekil 5.23 : Dairesel çizge hakim küme üyeleri.....	70
Şekil 5.24 : Ağaç çizge.....	71
Şekil 5.25 : Ağaç çizge hakim küme üyeleri.....	71
Şekil 5.26 : İki parçalı çizge.....	72
Şekil 5.27 : İki parçalı çizge hakim küme iterasyonları.....	73
Şekil 5.28 : Ağaç çizge hakim küme üyeleri.....	73

SEMBOLLER VE KISALTMALAR

P	: Polinomal zaman
NP	: Deterministik olmayan polinom zamanı
DS	: Hâkim küme
MDS	: Minimum hâkim küme
KMax	: Karcı maksimum
GPS	: Küresel konumlandırma sistemi
IDAP	: Uluslararası yapay zeka ve veri işleme sempozyumu
İBM	: İnönü üniversitesi bilgisayar mühendisliği
C_n	: n. kesme
Γ	: Karcı merkezlilik(baskınlık) değeri



ÖZET

Doktora Tezi

KARMAŞIK AĞLARDA HAKİM DÜĞÜMLERİN BELİRLENMESİ İÇİN YENİ BİR YÖNTEM

FURKAN ÖZTEMİZ

İnönü Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

x + 80 sayfa

2021

Danışman: Prof. Dr. Ali KARCI

Dijital ortamlarda sosyalleşmenin yaygınlaşması devasa miktarlarda sayısal verilerin üretilmesine neden olmuştur. Bu verilerden faydalı örüntüler çıkarılması için birçok sosyal ağ analiz yöntemi geliştirilmiştir. Bu analiz yöntemleri sosyal ağ yapısında modellenen bütün problem türleri için çözümler sunmaktadır. Sosyal ağlarda çözümü karmaşık olan birçok problem türü vardır. NP-hard olarak ifade edilen bu problem türleri polinomsal zamanda çözülemeyen zor problemlerdir. Herhangi bir ağ veya çizge üzerindeki minimum hakim kümenin belirlenmesi problemi popüler NP-hard problemlerden birisidir. Minimum hakim kümenin (Minimum dominating set) belirlenmesi için literatürde etkili bir algoritma bulunmamaktadır. Literatürde minimum hakim kümenin belirlenmesi için çözümler uzun zamanlar alan açgözlü (greedy) yaklaşıma sahip ve yaklaşık çözümler sunan algoritmalar bulunmaktadır.

Bu tez çalışmasında literatürde NP-hard problem olarak tanımlanmış minimum dominating set üyelerinin belirlenmesi için optimuma yakın sonuçlar üreten etkili bir algoritma önerilmiştir. Önerilen hakim küme algoritması 2 önemli aşamadan oluşmaktadır. İlk aşamada hakim küme üyelerinin belirlenmesinde seçim önceliği veren Karcı merkezlilik algoritması geliştirilmiştir. İkinci aşamada hakim küme üyelerini tespit eden seçim algoritması geliştirilmiştir. Karcı merkezlilik algoritması herhangi çizgedeki düğümlerin baskın olma değerlerini hesaplamak için kullanılır. Karcı merkezlilik algoritması 3 alt algoritmadan oluşmaktadır. İlk algoritma bir kapsayan ağaç olan Karcı maksimum ağacını (Kmax Tree) oluşturmak için kullanılır. İkinci algoritma Kmax ağacını göz önünde bulundurarak kesme derecelerinin hesaplanmasında kullanılmaktadır. Bu kesme işlemleri neticesinde çizgeden koparılan düğümlerin ağı ne kadar etkilediği sonuçları tespit edilmektedir. Üçüncü algoritma çizge düğüm derecesi, Kmax düğüm derecesi ve kesme derecelerinin birleşiminden oluşan Karcı merkezlilik (baskınlık) değerini üretir.

Çalışmada ayrıca literatürde popüler olarak bilinen sayfa değeri, özvektör, arasındalık ve yakınlık merkezlilik algoritmaları gerçek dünya problemlerine uygulanmış ve başarıları karşılaştırmalı sonuçlar ile incelenmiştir. Diğer bir uygulamada özgün olarak geliştirilen Karcı merkezlilik algoritması ile sayfa değeri, özvektör, yakınlık, derece merkezlilik algoritmaları karşılaştırılmıştır. Karcı merkezlilik algoritmasının literatürdeki

diğer popöler algoritmalar ile kısmi benzerlikler gösterdiği sonuçlarına ulaşmıştır. Önerilen algoritmaların bütün aşamaları ve sözde kodları tez çalışmasında ayrıntılı olarak verilmiştir.

Anahtar Kelimeler: Çizge teorisi, Baskın düğümler, Karcı merkezlik, Hakim küme



ABSTRACT

Phd. Thesis

A NEW METHOD FOR DETERMINING DOMINANT NODES IN COMPLEX NETWORKS

Furkan ÖZTEMİZ

Inonu University
Graduate School of Nature and Applied Sciences
Department of Computer Engineering

x + 80 pages

2021

Supervisor: Prof. Dr. Ali KARCI

The spread of socialization in digital environments has led to the production of huge amounts of digital data. Many social network analysis methods have been developed to extract useful patterns from these data. These analysis methods offer solutions for all types of problems that can be modeled in the social network structure. There are many types of problems in social networks that are complex to solve. These types of problems, expressed as NP-hard, are difficult problems that cannot be solved in polynomial time. The problem of determining the minimum dominant set on any network or graph is one of the popular NP-hard problems. There is no effective algorithm in the literature for determining the minimum dominant set. In the literature, there are algorithms with a greedy approach that provide approximate solutions and take a long time to solve to determine the minimum dominant set.

In this thesis, an effective algorithm that produces near-optimal results is proposed to determine the minimum dominating set members defined as NP-hard problems in the literature. The proposed dominating set algorithm consists of two important stages. In the first stage, Karci centrality algorithm which gives priority to selection in determining the dominant set members was developed. In the second stage, the selection algorithm that detects the dominating set members was developed. Karci centrality algorithm is used to calculate the dominance values of nodes in any graph. Karci centrality algorithm consists of 3 sub-algorithms. The first algorithm is used to construct the Karci maximum tree (Kmax Tree), which is a spanning tree. The second algorithm is used to calculate the cut-set degrees by considering the Kmax tree. As a result of these cut-set operations, the results of how much the nodes removed from the graph affect the network are determined. The third algorithm produces the Karci centrality (dominance) value, which is a combination of graph node degrees, Kmax node degrees, and cut-set degrees.

Besides pagerank, eigenvector, betweenness and closeness centrality algorithms, which are popularly known in the literature, were applied to real world problems and their successes were examined with comparative results in the study. In another application, the Karci centrality algorithm, which was originally developed, and the pagerank, eigenvector, closeness, degree centrality algorithms were compared. It has been concluded that the Karci centrality algorithm shows partial similarities with other popular algorithms in the literature. All stages and pseudo-codes of the proposed algorithms are given in detail in the thesis study.

Keywords: Graph theory, Dominant node, Karci centrality, Dominating set

1. GİRİŞ

İnsanların yaşadığı toplumun bir üyesi olma süreci sosyalleşme ile olmaktadır. Teknolojinin hızlı bir gelişim sağlaması insan hayatını her aşamada etkilemiştir. Gelişen teknolojinin önemli çıktılarında olan internet ve elektronik cihazların kabiliyetleri sosyalleşmenin hatırı sayılır bir miktarda dijital ortama kaymasına neden olmuştur. Kişinin bulunduğu toplulukdaki insanlar ile olan ilişkisinin belirlenmesi ise sosyal ağ kavramını ortaya çıkarmıştır. Normal sosyal yaşamda kullanılan jest ve mimiklerin, semboller ile duygu ve düşüncelere dönüştürerek iletişim kurulan ortamlara sosyal ağ denilmektedir [1]. Bu sosyalleşme sürecinin sayısal ortama yönelmesi beraberinde devasa verilerin ortaya çıkmasına sebep olmuştur. Bu veriler insanların ağdaki diğer insanlar üzerindeki etkisini, diğer insanlar ile arasındaki bağları, doğrudan bağlantısı olmayan kişilerin birbirleri ile bağlantı ihtimallerinin tespitinde, bireylerin ağ içerisinde hangi topluluk yakın veya dahil olduğunu ve bireylerin birbirleri arasındaki benzerliklerinin tespit edilmesinde vb birçok karmaşık örüntünün tespit edilmesinde kullanılabilir [2,3]. Bu örüntülerin keşfi normal istatistiksel yöntemler ile tespit edilemeyecek kadar karmaşık ve uzun zaman alan analiz işlemlerdir. Bu tarz ağ problemlerinin çözümü için çok sayıda sosyal ağ analiz yöntemi geliştirilmiştir. Bu yöntemler sadece bireylerin bir birleri ile etkileşim içerisinde olduğu sosyal ağlarda değil, sosyal ağ mantığında modellenen her türlü problemin çözümünde kullanılmaktadır. Sosyal ağ analiz yöntemleri graf (çizge) temelli yaklaşımlardan oluştuğundan kendisine çok geniş bir kullanım alanı bulmaktadır. Örnek olarak sosyal ağ yapısına göre modellenen bir kent içi ulaşım ağında kavşak noktalarının sosyal ağdaki bireyleri ifade ettiği düğünoldüğünde ulaşım ağı içerisindeki en etkin, yoğun kavşak noktalarının hangileri olduğu, kavşak noktalarının konumsal üstünlükleri, toplu taşıma sisteminde durak noktalarını bir birey olarak baz alındığında en etkin durakları, bir hat için en baskın durağın hangisi olduğunun belirlenmesi gibi bir çok özel çözüm sunabilmektedir [4,5]. Daha özet bir ifade ile anlatılacak olursa, sosyal ağ analiz algoritmaları çizge teorisi ile modellenen birçok problem türüne çözüm üretmektedir. Bu popüler kullanım alanlarından bir kaçını ifade edilecek olursa, ulaşım, telekomünikasyon, sağlık, otomasyon sistemleri, network sistemleri, lojistik, üretim sistemleri, harita sistemleri vb.. birçok alandan oluşmaktadır [6-8].

1.1 Tezin Amacı ve Kapsamı

Başta sosyal ağlar olmak üzere çizge teorisi yapısı içerisinde modellenen her türlü ağdaki bireyler veya nesnelere düğümler olarak tanımlanırken, bu düğümler arasındaki bağlantılar ise kenarlar olarak tanımlanmaktadır. Bu tez çalışmasında herhangi bir ağdaki etkin düğümlerin tespiti ve ağda seçilen minimuma yakın sayıda düğüm ile tüm ağı baskın altında tutabilecek düğümleri tespit eden etkili bir algoritma geliştirilmesi amaçlanmıştır. Ağdaki etkin olan düğümlerin belirlenmesi, sosyal ağlar üzerinde kişiler üstünde nüfusu yüksek olan bireylerin tespitinde, ulaşım ağına uygulanması durumunda araç sayımı veya konumsal durumları göz önüne alındığında kavşak noktalarının önemlerinin belirlenmesinde, üretim tesislerine uygulanmasında otomasyon sisteminde kilit konumda bulunan cihazların tespit edilmesi vb bir çok kullanım alanına sahiptir. Düğümlerin baskınlık değerinin yani merkezilik (centrality) değerlerinin belirlenmesi için özgün bir algoritma önerilmiştir. Bu özgün algoritmanın bir diğer önemli işlevi ağ üzerindeki baskın kümenin belirlenmesi için seçim önceliği sağlayacak parametre olmasıdır. Ağdaki baskın kümelerin belirlenmesi için sunulan algoritma yüksek performanslı, aç gözlü (greedy) yaklaşımından uzak, minimum hakim kümeye yakın sonuçlar vermektedir. Sunulan algoritmanın kullanım alanları incelendiğinde ağdaki hakim kümenin belirlenmesi sosyal ağlarda bir düşüncenin yayılmasının istenmesi durumunda bütün ağa en az sayıda kişi üzerinden yayılabilir, ulaşım ağlarında uygulanması durumunda ulaşım ağının denetleyen kolluk kuvvetlerinin minimum hangi kavşaklarda konumlanması gerekir gibi sorunlara çözüm sunabilmektedir. Önerilen yeni algoritma, 2 popüler çizge problemi olan baskın düğüm ve hakim küme kavramlarını özgün bir yaklaşım ile çözmekle birlikte, kolay uygulanabilirlik ve anlaşılabilirliğinden dolayı literatüre önemli katkılar sağlayacaktır. Ayrıca önerilen baskın düğüm ve hakim küme algoritmaları geliştirilmeye uygun bir yapıdadır ve sonraki süreçlerde minimum dominating set, max independent set, max clique gibi NP-complate problemlerin çözümünde önemli faydalar sağlayacağı düşünülmektedir.

1.2 Literatür

Literatürde farklı yaklaşımlara sahip birçok baskın düğüm ve hakim küme algoritmaları mevcuttur. Örneğin çizge üzerindeki hakim kümenin keşfi için sunulan algoritmalar bazıları büyük zaman karmaşıklığı ile kesin çözümler sunarken bazı algoritmalar küçük zaman karmaşıklığı ile yaklaşık çözümler sunmaktadır. Literatürde NP-Complate problem türü olarak tanımlanan (MDS) minimum hakim kümenin belirlenmesi için etkili ve kesin

sonuç veren bir algoritma bulunmamaktadır. Kesin sonuç veren algoritmalar genellikle greedy yaklaşımı ile çözümü uzun süre alan algoritmalarıdır. Yaklaşık sonuç veren algoritmalar ise kısa sürede sunduğu çözüm önerisinden dolayı cazip konumda bulunmaktadır. Baskın düğümlerin belirlenmesi için geliştirilen algoritmalara ise literatürde merkezlilik algoritmaları denilmektedir. Merkezlilik algoritmaları belirli problem türlerine özgü çözüm sunmak için geliştirilirken, yapılan geliştirmeler neticesinde birçok farklı alandaki graf problemlerinde başarılı sonuçlar alınabilmektedir.

Merkezlilik çalışmaları literatür özeti;

W.Xing ve A. Ghorboni web sayfalarının popülerliğinin tespit edilmesi için geliştirilen sayfa değeri algoritması ile ağırlıklı sayfa değeri algoritmasını karşılaştırmışlardır. Sayfa değeri algoritması diğer web sitelerinden kendine gelen bağlantı sayısına göre hesaplama yaparken, ağırlıklı sayfa değeri algoritması hem kendine gelen hem de giden bağlantıları hesaba katarak web sitesinin popülerliğini hesaplamaktadır. Yaptıkları çalışmanın sonucunda ağırlıklı sayfa değeri algoritmasının daha fazla ilgili sayfayı tespit ettiği sonucuna ulaşmışlardır [9].

Z.Yuanyuan kablosuz (wireless) sensör ağlarındaki sensörlerin enerji maliyetini azaltmak için çalışma yapmıştır. Yaptığı çalışmada sensör ağlarının iletişim yükünü azaltmak için ağ içerisindeki etkili konumdaki sensörleri belirlemiş ve en az sayıda sensör ile ağ içerisindeki iletişimi gerçekleştirmeyi amaçlamıştır [10].

T.Alahakoon ve diğerleri k-path isimli yeni bir ölçü sisteminin geliştirilmesi sürecinde baskın düğümlerin belirlenmesi için arasıdalık merkezlilik yöntemini kullanmışlardır. Gerçek ve yapay sosyal ağlar üzerinde yapılan deney sonuçları literatürde tanımlı ,başka algoritmalar ile karşılaştırılmıştır. Karşılaştırma neticesinde önerilen ölçü sisteminin etkili olan düğümlerin tespit edilmesinde doğruluk değeri ve çalışma süresi açısından kazanç sağladığı görülmüştür [11].

Y.Ding ve diğerleri sayfa değeri algoritmasına ait sönümlenme (damping) faktörlerinin yazarların baskın olma değerleri üzerine nasıl bir etkisi olduğunu belirlemek için bir çalışma yapmışlardır. Çalışmada kullanılan veriler 1970 ile 2008 yılları arasında Information retrieval (IR) alanında en çok atıf alan 108 yazara aittir. 0,05 ile 0,95 aralığında değişen sönümlenme katsayısı ile 108 yazarın baskın olma değerleri hesaplanmıştır. Yapılan çalışmanın çıktıları incelendiğinde sayfa değeri algoritmasının sonuçları ile atıf dereceleri

ve h- indekslerinin baskın olma deęerleri üzerinde önemli ölçüde iliřkisi olmadığı sonucuna ulařmıřlardır [12].

V. Jagadishwari ve S. Chakrabarty yakınlık merkezlilik yaklařımlı yeni bir algoritma önererek sosyal ağlarda baęlantı tahmini gerçekleřtirmişlerdir . Online sosyal ağlar üzerinde test edilen algoritma aynı zamanda temel baęlantı tahmin algoritmalarından Common neighbor (CN), Jaccard ve Adamic Adar ile karşılaştırılmıştır. Önerilen algoritmanın temel algoritmalarından daha iyi performans verdiği sonucuna ulařılmıştır [13].

Taras Agryzkov ve arkadaşları kentsel ağlar üzerinde ağın topolojisine ve ağdan çıkarılan coęrafik verilerin etkisini ölçen yeni bir merkezlilik modeli önerilmiştir. Yeni önerilen model özvektör merkezlilik yaklařımına dayanmaktadır. Yeni model belirlenen kentsel alandaki perakende ve hizmet kuruluşlarının daęılımını ve iliřkilerini ortaya çıkardığı sonuçlarına çalışmalarında yer vermiştir [14].

Maharani W. Twitter dan alınan sosyal ağ verileri üzerine özvektör ve derece merkezlilik algoritmalarını uygulayarak bireyler arasındaki etkili olan kişileri tespit etmeyi amaçlamışlardır. Tespit edilen sonuçlar neticesinde çalışmada kullanılan sosyal ağ içerisindeki en etkili 10 bireyin farklılıkları üzerinde yoğunlaşmışlardır [15].

Zhao S. ve arkadaşları kavşak, yol ve topluluk gibi farklı modlara göre kentsel trafik akışını analiz etmek için bir yol ağının hem topolojik özelliklerini hemde geometrik özelliklerini dikkate alan gelişmiş bir ağ merkezlięi ölçüm çevçevesi önerilmiştir. Analiz ölçüleri olarak derece, arasındalık ve sayfa deęeri merkezlilik algoritmaları ile şehir içi trafik akışını deęerlendirebilmek için GPS destekli taksi yörünge verileri kullanılmıştır [16].

Mededovic E. çalışmasında Milan ve Trento kentleri için telekom verileri kullanılarak yüksek iletişim trafięinin olduęu yerler olan hotspot noktalarını belirlemek ve aralarındaki iliřkiyi ölçmeyi hedeflemiştir. Sayfa deęeri ve özvektör merkezlilik algoritmalarını telekom veri setine uygulamış ve elde ettięi sonuçları karşılařtırmalı olarak vermiştir. Sonuçların varyansı, Toronto kentinin Milano kentinden önemli ölçüde daha küçük olduęunu göstermiştir [17].

Yan E. ve dięerleri sundukları çalışmada, akademik yayınlarda bulunan ortak yazarlar ve yayınlara ait atıfların kullanılması ile oluşturulan ağ üzerinde önerdikleri aęırlıklandırılmış sayfa deęeri algoritmasını uygulamışlardır. Elde edilen sonuçları ISSI konferanslarının h- indeksi, atıf ve program komite üyelikleri ile karşılařtırmışlardır. Bulgular önerilen aęırlıklı

sayfa değeri algoritmasının yazar etkisini ölçmede güvenilir sonuçlar sağladığını göstermektedir [18].

Hakim küme ve minimum hakim küme çalışmaları literatür özeti; Wang N. ve diğerleri bir kablosuz ağın arama yollarının verimliliğini belirlemek için bir yaklaşık sonuçlar üreten algoritma geliştirmişlerdir. Çalışmanın temel hedefi ağda bozulan düğümlerin bakımını ve yönlendirilmesini daha etkili olarak sağlamaktır [19].

Ruan L. ve diğerleri yaptıkları çalışmada aç gözlü (greedy) yaklaşımı ile minimum hakim küme algoritması önermişlerdir. Önerdikleri algoritmanın performans değeri $\ln\delta + 2$ olarak tespit edilmiştir. Buradaki δ sembolü, giriş çizgesinin minimum düğüm derecesini göstermektedir [20].

Xu X. ve diğerleri minimum hakim küme problemini Hopfield ağlarına uygulayarak yapay sinir ağlarına özgü başarılı sonuçlar elde etmişlerdir. Çalışmalarında ayrıca geliştirilen algoritmanın maksimum hizip (max clique), maksimum bağımsız küme (max independent set) problemlerine çözüm saylayacağı ifade edilmiştir [21].

Ho K.C. ve diğerleri metasezgisel yöntemler kullanılarak minimum hakim kümenin tespiti için özgün bir algoritma geliştirilmiştir. Gerçekleştirilen çalışmada genetik algoritma yapısında turnuva seçimi adı verilen bir algoritma sunulmuştur. Turnuva seçiminin kullanımı, karınca kolonisi algoritması kullanılarak en iyi feromon güncellemesi ile birleştirilmiştir. Çalışmada arama alanının daha yüksek bir seviyede keşfedilmesi amaçlanmış ve karınca kolonisi algoritmasının genetik algoritmaya göre daha başarılı sonuçlar verdiği belirtilmiştir [22].

Jovanovic R. ve arkadaşları minimum hakim küme probleminin çözümü için iki aşamalı karınca kolonisi (ACO) algoritmasını kullanmışlardır. Önerilen yeni yaklaşımın yardımıyla feromon düzeltme stratejisi eklenmiştir. Bu sayede açgözlü yaklaşıma dayalı olarak tek adımlı ACO'yu yerel optimumda yakalama sorununu çözümlüşlerdir [23].

Chalupa D. klasik açgözlü yaklaşım algoritmaları ve karma sezgisel tabanlı algoritmalarından daha başarılı olduğu iddia edilen sıra tabanlı rastgele yerel arama (RLS) algoritması geliştirilmiştir. Bu algoritma, Barabasi-Albert Model birim disk çizgesi, ölçeksiz çizge ve iki sosyal ağ dahil olmak üzere gerçek dünya problemlerinde MDS tespiti için başarılı sonuçlar vermiştir [24].

Zhou Y. ve diğerleri tarafından minimum hakim küme için yaklaşık sonuçlar üreten ve zaman karmaşıklığı $O(n^2)$ olan bir algoritma geliştirilmiştir. Önerilen bu algoritma,

çalışmanın yayınlandığı tarih itibari ile literatürde tanımlı olan en hızlı algortimalardan biri ile karşılaştırılarak sonuçlarına yer verilmiştir [25].

Grandoni F. gerçekleştirdiği çalışmada asimtotik olarak $\Omega(2^n)$ zaman karmaşıklığına sahip çalışmanın yayınlandığı tarih itibari ile en hızlı MDS algoritmasından daha hızlı bir algoritma sunmuştur. Önerdiği yeni algoritmanın zaman karmaşıklığı $O(1,81^n)$ olarak ifade edilmiştir [26]. (n ifadesi düğüm sayısını ifade etmektedir)

Grinstead D.L. ve Slater P.J. minimum hakim kümenin belirlenmesi için geliştirdikleri T ağacına ait kesişme sayısı dikkate alınarak bir lineer algoritma önermişlerdir. Çalışmada rastgele bipartite çizgeler için iki ayrı MDS olup olmadığını belirleme probleminin NP-Hard bir problem olduğunu belirtmişlerdir [27].

1.3 Tezin Yenilikçi Yönü

Tezin temel amacı herhangi bir çizge üzerindeki baskın düğümleri ve çizge üzerindeki hakim kümenin belirlenmesi için etkili bir algoritma önerisidir. Önerilen algoritma 2 önemli çizge problemine daha önce literatürde olmayan özgün bir yaklaşım ile başarılı çözüm sunmaktadır. Tez çalışmasında sunulan algoritmalar bir birlerini tamamlayıcı durumdadır. Çizge üzerindeki hakim kümenin tespiti için çizge içerisindeki baskın kümelerin tespit edilmesi gereklidir. Yine baskın kümelerin belirlenmesi için 3 önemli aşamadan oluşan alt algoritma önerilmiştir. Bu algoritmalarda en karmaşık olanlarından birisi kesme algoritması diğeri ise bir kapsayan ağac olan K-Max (Karci-Maximum) tree dir. Bu iki alt algoritmada özgündür ve literatürde bulunmamaktadır. Literatüre önemli katkı sağlayacak olan kesme algoritmasını kısaca bahsetmek gerekirse belirli kriterlere göre çizgeden koparılan düğümlerin çizgeyi nasıl etkilediklerinin tespit edilmesine dayanmaktadır. Bu etkiler çizgedeki düğümün baskınlık değerinin belirlenmesinde en önemli kriterdir. Özgün olarak geliştirilen Kmax algoritması ise kesmelerin gerçekleştirileceği kenarların belirlendiği özel bir kapsayan ağaçtır.

1.4 Tezin Organizasyonu

Bu tez çalışması altı bölümden oluşmaktadır. Bölümler aşağıda belirtildiği şekilde düzenlenmiştir.

Birinci bölümde sosyal ağ kavramından kısaca bahsedilmiş ve uygulama alanlarına değinilmiştir. Tezin ana yapısını oluşturan çizge içerisindeki baskın düğüm ve hakim küme

kavramlarından bahsedilmiş ve literatürde bu alanda gerçekleştirilen çalışmalara yer verilmiştir. Ayrıca tezin amacına, kapsamına ve yenilikçi yönüne bu bölümde değinilmiştir.

İkinci bölümde çizge teorisinin tanımı yapılarak çizge türlerine örneklendirilerek yer verilmiştir. Hakim küme ve baskın düğüm kavramları ayrıntılı olarak açıklanmıştır.

Üçüncü bölümde sosyal ağ kavramı ayrıntılı olarak açıklanmıştır. Sosyal ağlar üzerinde uygulanan önemli yöntemlerden bahsedilmiştir. Bu tez çalışmasının önemli çıktılarından olan baskın düğüm kavramı üzerinde durularak literatürde kabul görmüş popüler merkezlilik algoritmalarının açıklamaları yapılmıştır. Sosyal ağ algoritmalarının karşılaştırmalarına ve gerçek dünya problemlerine yönelik sundukları çözümler, uygulamalı olarak açıklanmıştır.

Dördüncü bölümde tezin ana konusu olarak geliştirilen algoritmalar hakkında bilgiler verilmiştir. Önerilen baskın düğüm ve hakim küme algoritmaları ayrıntılı olarak örnekler üzerinde açıklanmıştır. Algoritmanın tüm aşamalarını içeren sözde kodlar bu bölümde verilmiştir.

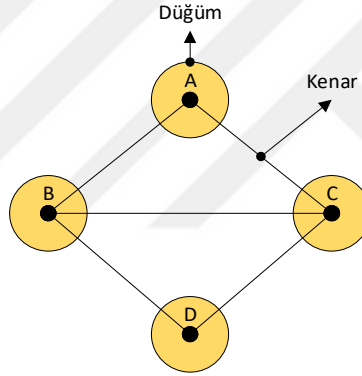
Beşinci bölümde algoritmaların farklı çizge türlerindeki deneysel sonuçlarına yer verilmiştir.

Altıncı bölümde önerilen algoritmaya ait sonuç ve öneriler verilmiştir.

2. ÇİZGE TEORİSİ VE BASKIN KÜMELER

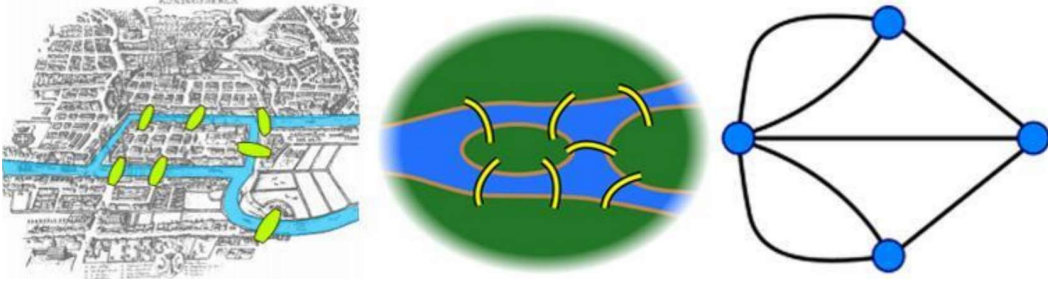
Çizgeler kenar ve düğüm olmak üzere iki önemli parametreye sahiptir. Çizgeler $\mathcal{C}=(D, K)$ olarak ifade edilirler. D (düğümler) nesne ve olayları modellemek için kullanılırken, K (kenarlar) bu nesne veya olaylar arasındaki ilişkileri ifade etmek için kullanılmaktadır [28]. Çizgeler farklı problem türlerinde kolaylıkla kullanılmaktadır. Örnek verilecek olursa, ulaşım ağlarında kavşak noktaları düğümleri ifade ederken, kenarlar bu kavşak noktalarını birbirine bağlayan yolları ifade etmektedir [29]. Sosyal ağlarda bireyler düğüm olarak modellenirken, bireyler arasındaki ilişkiler kenar bağlantıları ile gösterilmektedir [30].

$\mathcal{C} = (D,K)$ ifadesi birçok kaynakta sıralı olarak kabul edilmiştir yani öncelikle düğümler ardından kenarlar gösterilmiştir [31]. Örneğin $\mathcal{C} = (\{A,B,C,D\}, \{(A,B), (A,C), (C,D), (A,D), (B,D)\})$ şeklinde ifade edilmektedir. Şekil 2.1’de örnekte belirtilen çizge yapısına ait görsel verilmiştir.



Şekil 2.1. Çizge yapısı

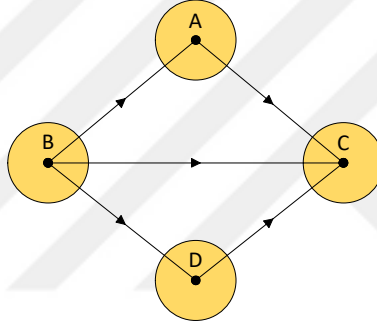
Çizge teorisi Euler tarafından 1736 senesinde sunulan bir makale sonrası doğmuştur. Makalenin içeriği Königsberg Brige problemlerini çözebilen bir teoriyi içermektedir. Pregel nehri Königsberg kasabasının içinden akmaktadır. Nehrin ortasında Şekil 2.2’deki gibi nehrin kıyılarına ve birbirlerine köprüler ile bağlı iki ada bulunmaktadır [32]. Königsberg kasabasının vatandaşları için problem kıyıların veya adaların birbirinden başlayıp tüm köprülerden sadece bir kez geçerek başlanılan yere yürünebilmesidir [31,33] Bu problemin çözüm süreci çizge teorisinin doğmasına neden olmuştur.



Şekil 2.2. Königsberg 7 köprü problemi [32]

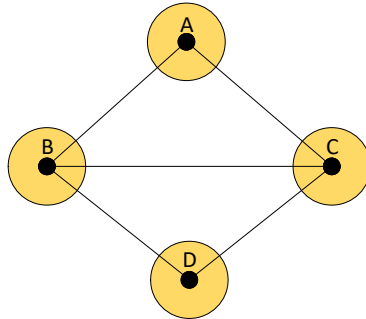
2.1 Çizge Türleri

Çizgeler yönlerine ve yapısal özellikleri bakımından çeşitlilik göstermektedir. Çizgeler yönlerine göre yönlü, yönsüz, karışık olmak üzere 3'e ayrılmaktadır. Yönlü çizgeler Şekil 2.3'de belirtildiği üzere bütün kenarlar arasında tek yönlü bağlantının olmasıdır [28].



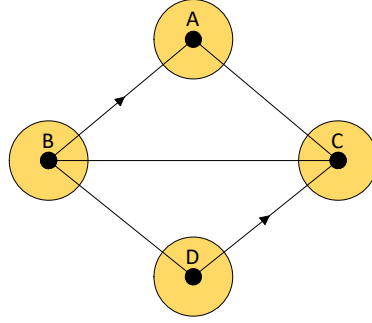
Şekil 2.3. Yönlü çizge

Yönsüz çizge Şekil 2.4'de gösterildiği üzere çizgeyi oluşturan bütün kenarlar yönsüz olma durumudur.



Şekil 2.4. Yönsüz çizge

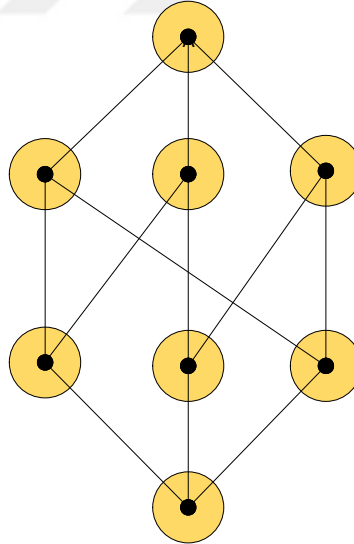
Eğer çizge içerisinde yönlü ve yönsüz kenarlar bulunuyorsa bu çizgelere karışık çizgeler denir. Şekil 2.5’ de karışık çizgelere ait bir örnek verilmiştir.



Şekil 2.5. Karışık çizge

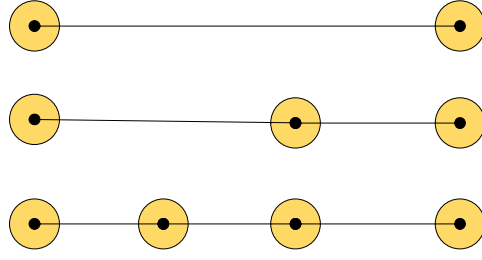
Yapısal bakımdan bazı özel çizge çeşitleri vardır. Bu çizgelerden bazıları tam, tamamlayıcı, düzgün, çevre, yol, iki parçalı, ağaç, yıldız ve tekerlek çizgelerdir.[34]

Bir çizgede tüm düğümler aynı dereceye sahip ise bu çizgelere düzenli çizgeler denilmektedir. Şekil 2.6’ da düzenli çizgeye ait örnek verilmiştir.



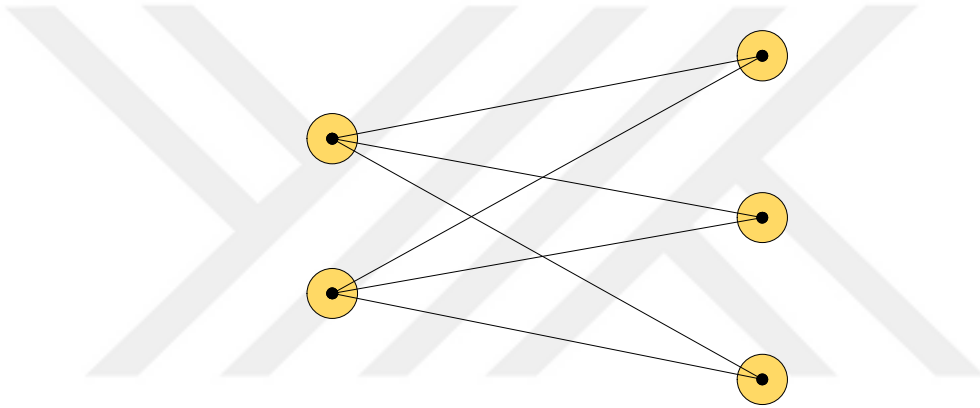
Şekil 2.6. Düzenli çizge

İlk ve son düğümlerinin dereceleri 1 olan ve diğer düğüm dereceleri 2 olan çizgelere yol çizgesi denmektedir. Örnek çizge Şekil 2.7’ de verilmiştir.



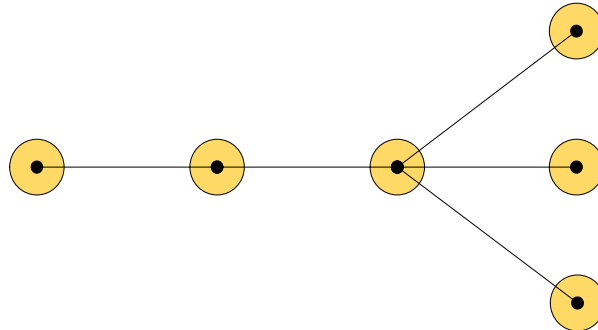
Şekil 2.7. Yol çizgesi

Bir çizgedeki düğüm kümesi V_1 ve V_2 olarak iki kümeye ayrılmış olsun. Çizge V_1 deki düğümler ile çizge V_2 deki düğümlerin birleşiminden oluşuyor ise, bu çizgelere iki parçalı (bipartite graph) çizgeler denmektedir. Şekil 2.8' de iki parçalı çizge yapısı verilmiştir.



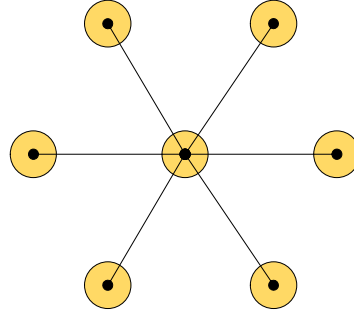
Şekil 2.8. İki parçalı çizge

Bağlı çizge içerisinde hiç devre bulunmuyorsa buna ağaç denir. Ağaç çizgeleri T ifadesi ile gösterilmektedir. Şekil 2.9' da örnek bir ağaç çizge yapısı verilmiştir.



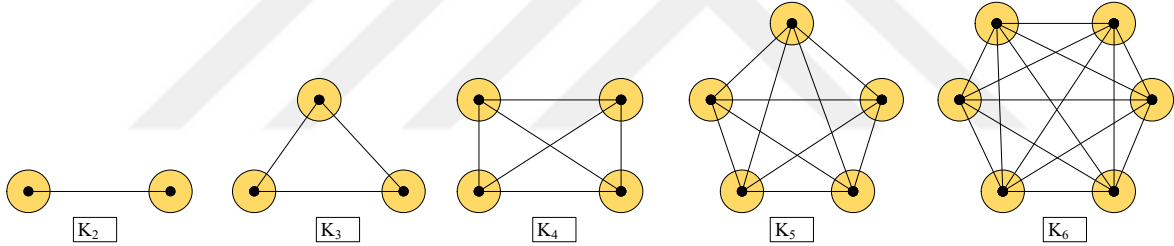
Şekil 2.9. Ağaç çizge

N düğümlü bir ağaçta bir noktasının derecesi $n-1$ ve diğer düğümlerin derecesi 1 ise bu çizgelere yıldız çizge denmektedir. Şekil 2.10'da yıldız çizgeye ait örnek verilmiştir.



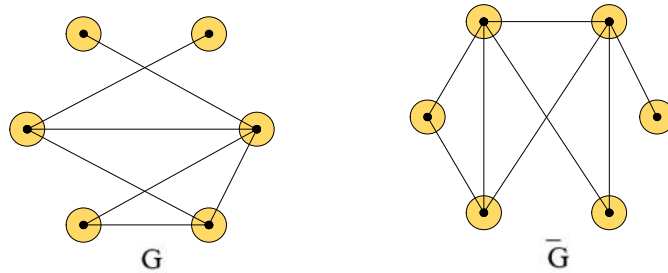
Şekil 2.10. Yıldız çizge

Bir çizge üzerindeki herbir düğüm çifti birbiri ile komşu ise bu çizgeye tam çizge denilmektedir. N noktalı bir tam çizge K_n biçiminde gösterilmektedir. Şekil 2.11'de örnek bir tam çizge verilmiştir.



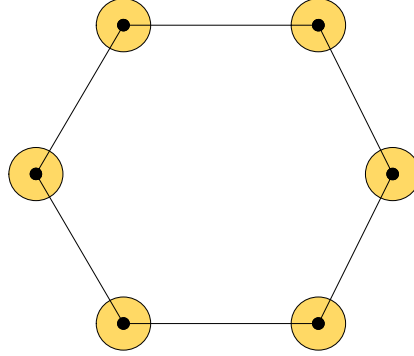
Şekil 2.11. Tam çizge

Düğüm kümesi G nin düğüm kümesi ile aynı olan, kenar kümesi ise G de olmayan kenarlardan oluşan ve komşu olmayan düğümleri birbirine komşu yapan çizgeye tamamlayıcı çizge denilmektedir. Bir G çizgesinin tamamlayıcısı \bar{G} ile ifade edilir.



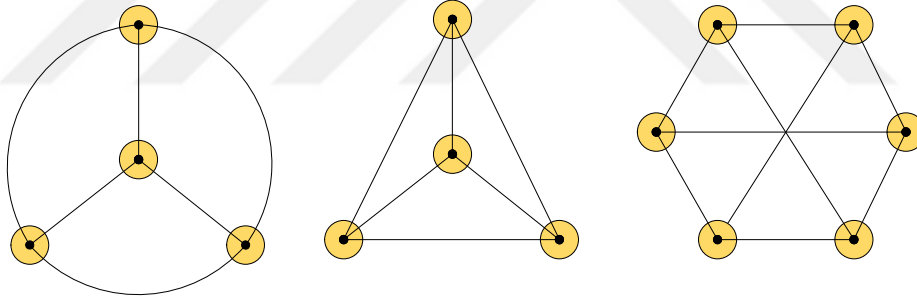
Şekil 2.12. Tamamlayıcı çizge

Başlangıç ve bitiş düğümleri aynı olan ve bütün düğümlerin derecesi 2 olan çizgelere çevre çizge denilmektedir. n düğümlü bir çevre çizge C_n ile ifade edilmektedir. Şekil 2.13’ de örnek bir çevre çizge verilmiştir.



Şekil 2.13. Çevre çizge

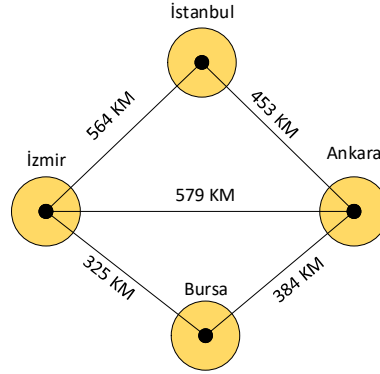
N düğümlü bir C_n çevre çizgesinin tüm düğümlerine tek bir kenarla komşu olan yeni bir düğüm eklenmesiyle elde edilen çizgeye tekerlek çizge denilmektedir.



Şekil 2.14. Tekerlek çizge

Çizge üzerinde düğümler arasındaki bağlantıların yoğunluğunu, sağlamlığını ve uzunluğunu belirten sayısal ifadeler olabilir. Bu çizge türlerine ağırlıklı ve ağırlıklandırılmış çizge denilmektedir. Çizge içerisinde bulunan kenarların tamamının maliyeti 1 veya eşit ise bu çizge türüne ağırlıksız çizge denir [28]. Çizge üzerindeki zengin modelleme yapısından dolayı çok çeşitli kullanım alanları bulunmaktadır. Bunlardan bazıları; nesne ve konumlar arasındaki uzaklıkların belirtilmesinde, ağ sistemlerinde bant genişliklerinin gösteriminde, kişi ve nesnel arasındaki ilişki kuvvetlerinin gösteriminde vb.. olarak sayılabilir. Şekil

2.15’de şehirler arası uzaklık değerlerinin verildiği örnek bir ağırlıklı çizge yapısı verilmiştir [35].

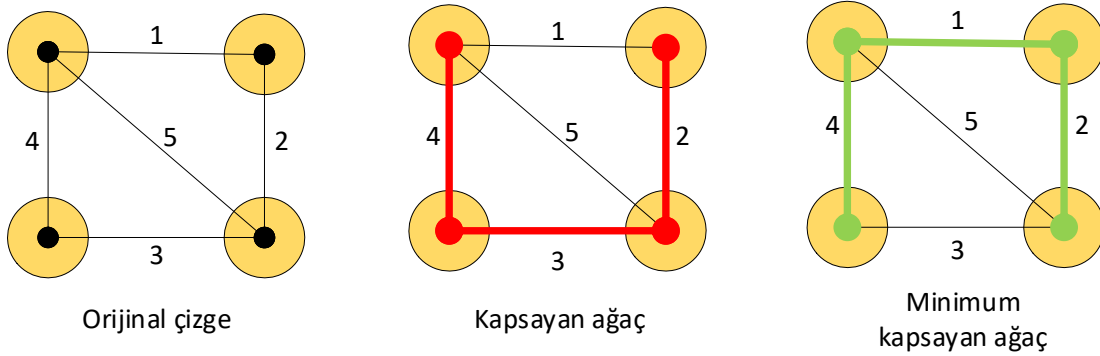


Şekil 2.15. Ağırlıklı çizge

2.2 Kapsayan Ağaçlar

Bir ağaç bağlı yönsüz bir çizgedir. Kapsayan ağacı bir G çizgesinin alt çizgesi ise ve G çizgesinin bütün düğümlerini içeriyorsa buna G çizgesinin kapsayan ağacı denilmektedir. Bağlı bir G çizgesinin kapsayan ağacı, G çizgesinin döngü içermeyen maksimum kenar kümesi veya tüm düğümleri birleştiren minimum kenar kümesi olarak da tanımlanabilir [36].

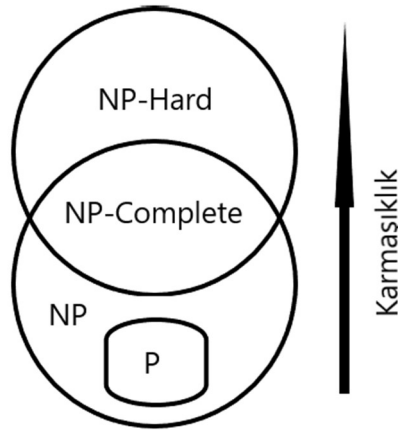
Literatürdeki popüler algoritmalarından bir kaçından bahsedilirse; Prim, Bellman Ford, Floyd Warshall, Kruskal minimum kapsayan (minimum spanning tree) ağaçlarına örnek olarak verilebilir. Her iki algoritmada çizge üzerindeki en az maliyetli yolları tespit ederek çizge üzerindeki bütün düğümlere ulaşmayı hedeflemektedir [37]. Şekil 2.16’da örnek bir çizgede kapsayan ve minimum kapsayan ağaçlara ait görseller verilmiştir.



Şekil 2.16. Kapsayan ağaçlar

2.3 NP Complete Problem

Bir problem evet veya hayır olarak sonuç üretiyorsa buna karar problemi denir. Bu karar problemleri polinomal zamanda evet veya hayır şeklinde bir çözüm üretiyor ise P (Polynomial Time) problem denilmektedir. NP (Non-deterministic-polynomial-Time) polinomsal zamanda çözülemeyen karar problemlerinin karmaşıklık sınıfıdır. NP-Complete her adımdaki çözümlene zamanının bir önceki çözümlene zamanlarından daha fazla olduğu problem türleridir. Bu problem türleri polinomal zamanlarda çözülmesi mümkün olmamakla birlikte belirsiz (Non-Deterministic) Turing makineler ile belirli zamanlarda çözülebilmektedir [38]. NP-Hard problemler ise polinomsal zamanda bir çözümü olduğu ispatlanamayan problemlerin karmaşıklık sınıfıdır. Minimum Dominatin Set, 3SAT ve Halting problemleri NP-Hard problem türlerine örnek olarak verilebilir [39,40]. Şekil 2.17' de karmaşık sınıflarına ait görsel verilmiştir.



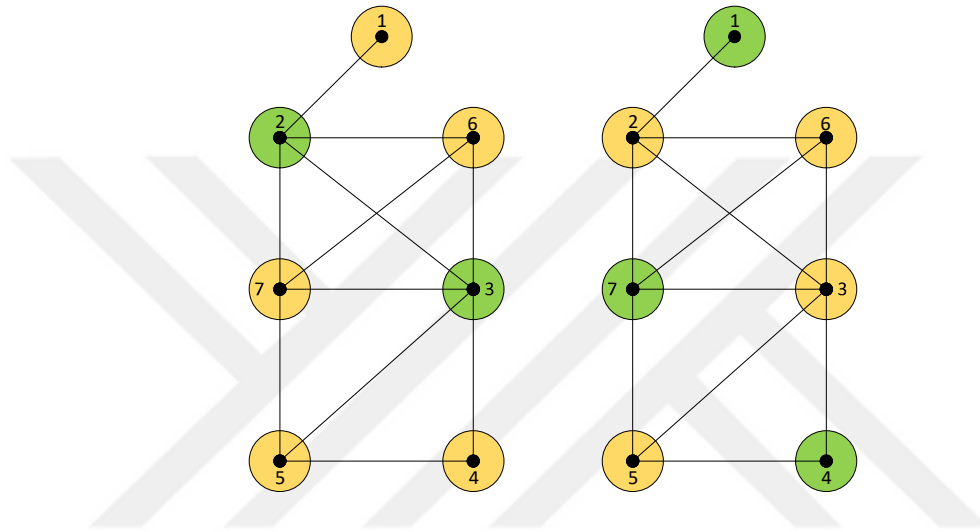
Şekil 2.17. Karmaşıklık sınıfları

2.4 Baskın Düğüm ve Hakim Küme Kavramları

Çizge üzerinde giriş düğümünden n düğümüne kadar olan her yol d düğümünden geçmek zorunda olursa, d düğümü n düğümünü domine ediyor denir [41]. Günlük hayattan örnekler ile açıklanacak olursa, bireylerin sosyal ağları üzerinde kişiler üzerindeki etkileri, ulaşım ağlarındaki kavşak noktalarının kendisine bağlı kavşak noktaları üzerindeki etkisi ilgili kişi veya nesnenin baskınlık değerleri olmaktadır.

Yönsüz olarak verilen $G = (V,E)$ çizgesi V düğümlerinden ve E kenarlarından oluşmaktadır. Bir hakim kümesinde olan her düğüm $D \subseteq V$ şeklinde ifade edilir. $v \in V$, D 'nin bir üyesidir ya da D 'ye ait bir düğümün komşusudur. D 'nin üyelerine hakim küme denilmektedir. Eğer

mümkün olan minimum sayıda düğüm ile baskın küme oluşturulmuş ise o baskın kümeye minimum hakim küme (Minimum Dominating Set) denir [42]. Şekil 2.18’ de hakim küme ve minimum hakim kümeye ait örnek bir çizge verilmiştir. Minimum hakim kümenin belirlenmesi çözümü oldukça zor olan NP-Hard problem türü olarak literatüre girmiştir. Literatürde MDS’nin belirlenmesi için deterministik etkili bir algoritma bulunmamaktadır. Kesin çözüm sunan algoritmalar genellikle greedy yaklaşımını kullanmaktadır ve bu yaklaşım çözüm süresini uzatmaktadır. Performansın ön planda olduğu işlemler için yaklaşım çözüm sunan hakim küme algoritmaları geliştirilmiştir [43].



Şekil 2.18. Hâkim ve Minimum hâkim kümeler

Şekil 2.18’de ki çizgede düğüm 2 ve düğüm 3 minimum hakim küme üyelerine örnek iken düğüm 1, düğüm 7 ve düğüm 4 hakim küme üyelerine örnek olarak verilebilir.

3. SOSYAL AĞLAR

Sosyal ağlar bireyler arasındaki kişisel veya profesyonel ilişkilerin oluşturduğu ağ yapılarıdır. Son dönemlerde bu ilişkilerin çevrim içi (online) sistemler üzerinde gerçekleşmesi ile sosyal ağ yapıları sayısal bir forma dönüşmüştür [44]. Bu sayısal formdaki sosyal ağ verileri zamanla içerisinde gizli örüntüler barındıran devasa veri yapılarına dönüşmüştür. Çizge (Graph Theory) teorisinin gelişmesi ile sosyal ağlar içerisindeki örüntülerin keşfi için birçok analiz algoritması geliştirilmiştir. Geliştirilen sosyal ağ analiz yöntemleri sadece bireyler üzerindeki problemlerin çözümünde değil, Çizge teorisi ile modellenen birçok problem türüne çözüm sunmuştur [45]. Merkezlilik algoritmaları (Centrality) modellenen bir çizgedeki baskın düğümlerin belirlenmesinde, diğer bir ifade ile topluluk içerisindeki baskın olan bireylerin belirlenmesinde kullanılır. Topluluk tespit algoritmaları (Community detection) düğüm gruplarının nasıl kümelendiğini veya bölündüğünü tespit etmek için kullanılır. Benzerlik algoritmaları (Similarity) farklı vektör tabanlı ölçümler kullanarak düğüm çiftlerinin benzerliklerini hesaplar. Bağlantı tahmini algoritmaları (Link prediction), bir çift düğümün yakınlığını belirlemeye yardımcı olur. Hesaplanan puanlar daha sonra aralarındaki yeni ilişkileri tahmin etmek için kullanılabilir. Yol bulma algoritmaları (Path finding) iki veya daha fazla düğüm arasındaki en kısa yolların bulunması ve yolların kalitesinin değerlendirilmesinde kullanılır [46].

3.1 Merkezlilik Algoritmaları

Herhangi bir çizge üzerinde etkili veya baskın olan düğümlerin tespit edilmesi için kullanılan yöntemlerdir [47]. Bu baskın düğümler sosyal ağlarda diğer insanları etkisi altına alan baskın bireyleri, ulaşım ağlarında yolların kesişim noktaları olan kavşak noktalarını, ağ sistemleri üzerindeki hub noktalarını ifade etmektedir. Literatürde popüler olarak kullanılan merkezlilik algoritmalarından sayfa değeri (pagerank), yakınlık (closeness), arasındalık (betweenness) ve özvektör (eigenvector) merkezlilik algoritmalarına ait açıklamalara aşağıda yer verilmiştir.

3.1.1 Sayfa değeri algoritması (Pagerank)

Lary Page ve Sergey Brin tarafından 1998 senesinde geliştirilen algoritma web sayfalarının bağlantılarını tespit etme ve arama motorlarında aranan içeriklerin daha başarılı bir şekilde

sunulmasını amaçlamıştır. Web sayfaları birer düğüm olarak düşünülürse gelen bağlantı sayısı ve kaynak düğümün baskınlığına göre tüm düğümlere ait puanları yinelemeli olarak komşulara aktararak hesaplanır [48]. Bu durum çizgenin tamamının incelenmesi anlamına gelmektedir. Bir düğümün önemi, kendisine bağlı düğümlerin önemi ile doğrudan ilgilidir. Denklem (1)'de algoritmaya ait formül verilmiştir.

$$PR(A) = (1 - d) + d \left(\frac{PR(t1)}{C(t1)} + \dots + \frac{PR(tn)}{C(tn)} \right) \quad (1)$$

Denklem (1) de verilen $t1 \dots tn$ ifadesi baskınlık değeri tespit edilmesi istenen düğümlere bağlantı veren düğümleri, $PR(tn)$ diğer düğümlere bağlantı veren düğümlere ait baskınlık değerlerini, $C(tn)$ her düğümün diğer düğümlere verdiği bağlantı sayısını, d (damping factor) baskınlık değerlerinin 1'i geçmemesi için kullanılan katsayıyı ifade etmektedir.

3.1.2 Arasındalık algoritması (Betweenness)

Arasındalık algoritması düğümlerin birbirlerine olan en kısa mesafeleri hesaplanırken en kısa mesafe ile bağlantısı olan düğümlerin daha önemli olduğunu belirtir. Bu durumda çizgenin ortasına yakın olan düğümlerin daha etkin olduğu sonucunu vermektedir [49]. Denklem (2) de algoritmaya ait formül verilmiştir.

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \quad (2)$$

σ_{st} , s düğümü ile t düğümü arasındaki en kısa yolların toplamıdır. $\sigma_{st}(v)$ ise v 'den geçen ayrıtların sayısıdır.

3.1.3 Özvektör algoritması (Eigenvector)

1987 yılında Phillip Bonacich tarafından önerilmiştir. Özvektör algoritması düğümlerin geçiş etkisini ölçer ve puanlandırır. Çizgedeki bir düğüm yüksek puanlı düğümlerden, düşük puanlı düğümlere göre daha fazla etkilenmektedir. Bu durum yüksek puanlı düğümün yine başka bir yüksek puanlı düğüm ile bağlantısı olduğunu göstermektedir [50]. Denklem (3) ve

(4) de algoritmaya ait formül verilmiştir. A ifadesi bitişiklik matrisi, λ A matrisinin en büyük özdeğer vektörünü ifade etmektedir.

$$Ax = \lambda x, \quad \lambda x_i = \sum_{j=1}^n a_{ij} x_j, \quad i = 1, \dots, n \quad (3)$$

$$c(\beta) = \sum_{k=1}^{\infty} \beta^{k-1} A^k \mathbf{1}, \quad |\beta| < 1/\lambda \quad (4)$$

3.1.4 Yakınlık algoritması (Closeness)

Yakınlık merkezlilik algoritması her bir düğümün diğer tüm düğümlere olan uzaklıkları hesaplanarak bütün düğüm çiftleri arasındaki en kısa yolların hesaplanması ile gerçekleştirilir. Elde edilen sonuçlar ters çevrilir ve yüksek puanlı düğümler diğer tüm düğümlere en kısa mesafe toplamına sahiptir [51]. Algoritmaya ait formüller denklem (5) ve (6)' da verilmiştir.

$$mesafe[u] = \sum_{\substack{u \in V \\ d_G(u,v) \neq \infty}} d_G(u, v) \quad (5)$$

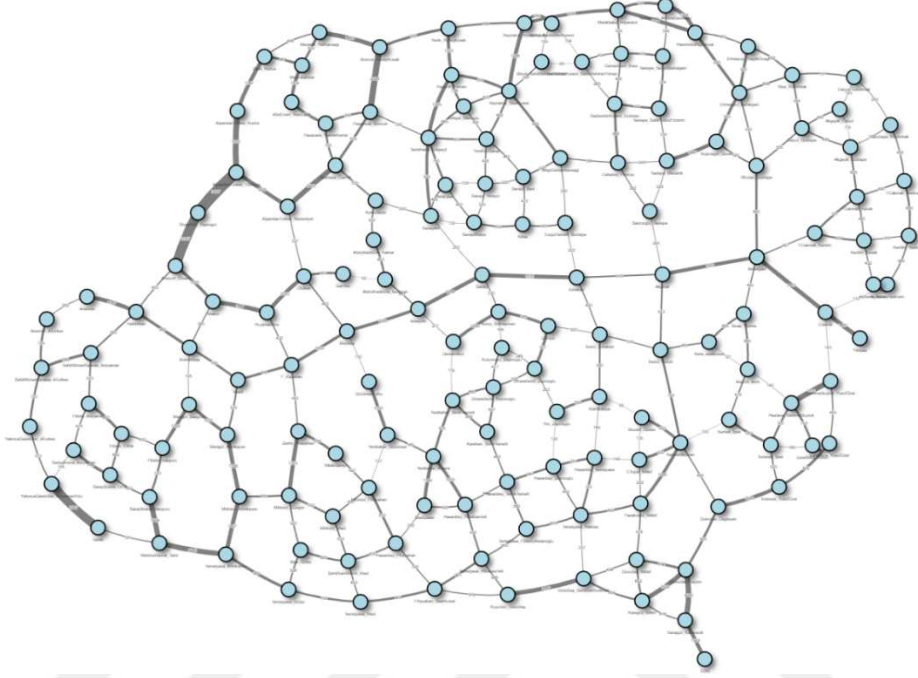
$G = (V, E)$ için $u \in V$ olmak üzere $mesafe[u]$ ilgili düğümün diğer düğümler ile arasındaki mesafe toplamını ifade etmektedir. Denklem (6) deki $cc[u]$ ise yakınlık merkezlilik değerini belirtmektedir.

$$cc[u] = \frac{1}{mesafe[u]} \quad (6)$$

3.1.5 Merkezlilik algoritma uygulamaları

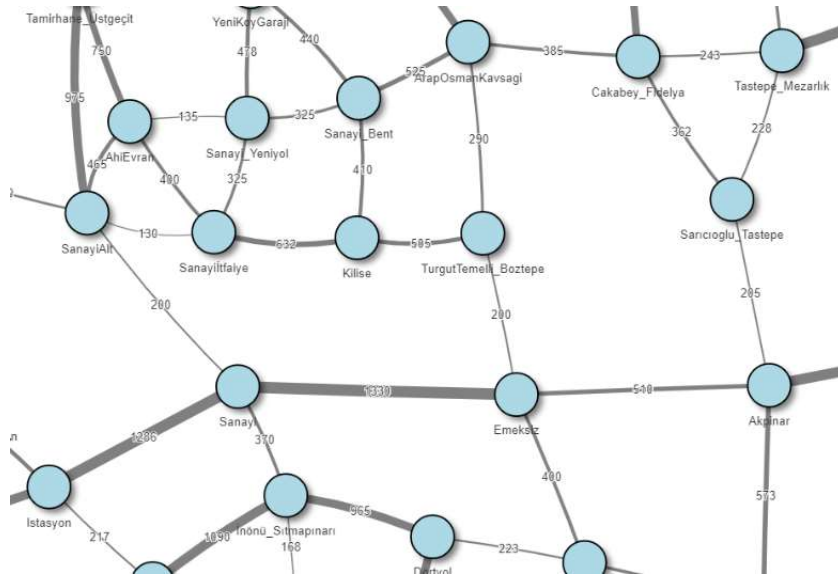
Bu tez çalışmasında önerilen hakim küme algoritmasının en önemli aşaması hakim küme üyelerinin seçiminde öncelik vermek için kullanılan özgün bir merkezlilik algoritmasıdır. Bu bölümde literatürde popüler olarak kullanılan merkezlilik algoritmaları kullanılarak gerçek dünya problemlerine çözüm sunan çalışmalara yer verilmiştir. Ayrıca diğer bir uygulamada ise önerilen merkezlilik algoritmasının diğer merkezlilik algoritmaları ile karşılaştırılmalı sonuçlarına yer verilmiştir. İlk uygulamada hibrit bir yöntem uygulayarak Malatya ili kavşak noktalarının merkezlilik analizi sonuçları ile kent için kavşak noktalarının yük yoğunluklarının tespit edilmesi amaçlanmıştır. Temel amaç sadece kavşak konumlarının birbirlerine olan mesafelerini ve kent içerisindeki konumlarını kullanarak ulaşım ağındaki kavşak noktalarına ait yük analizi yapılmasıdır. İlk aşamada farklı merkezlilik

Şekil 3.3’de kent’e ait kavşak noktaları ve yollara ait çizge yapısı verilmiştir. Kavşak noktaları arasındaki yolların uzunluğu çizge üzerinde orantılı olarak kalın kenarlar ile gösterilmiştir. Ulaşım ağı yollara ait mesafe bilgileri sayesinde ağırlıklandırılmış çizgeye dönüştürülmüştür.



Şekil 3.3. Kent merkezi cadde ve bulvarlar çizge yapısı

Şekil 3.4’de ulaşım ağı çizgesinin yakınlılaştırılmış hâli verilmiştir.



Şekil 3.4. Ulaşım ağı çizge yapısı

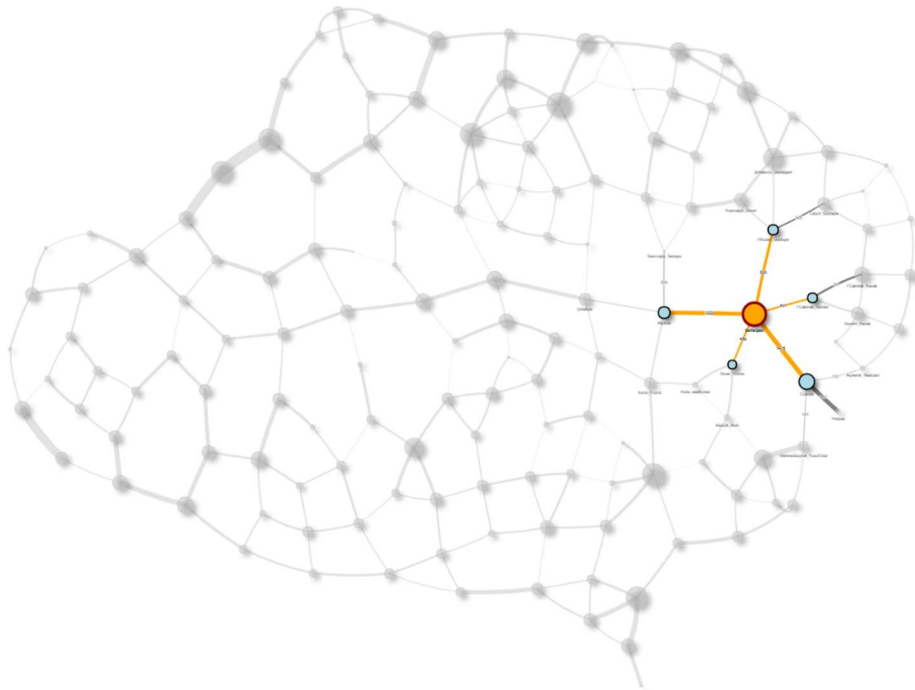
Ağırlıklandırılmış çizge üzerinde sayfadeğeri, arasındalık, yakınlık ve özvektör algoritmaları ayrı ayrı uygulanmıştır.

Sayfa değeri algoritmasına göre en baskın ilk 15 kavşak noktasına ait sonuçlar Çizelge 3.1’de verilmiştir.

Çizelge 3.1. Sayfa değeri algoritması merkezlilik değerleri

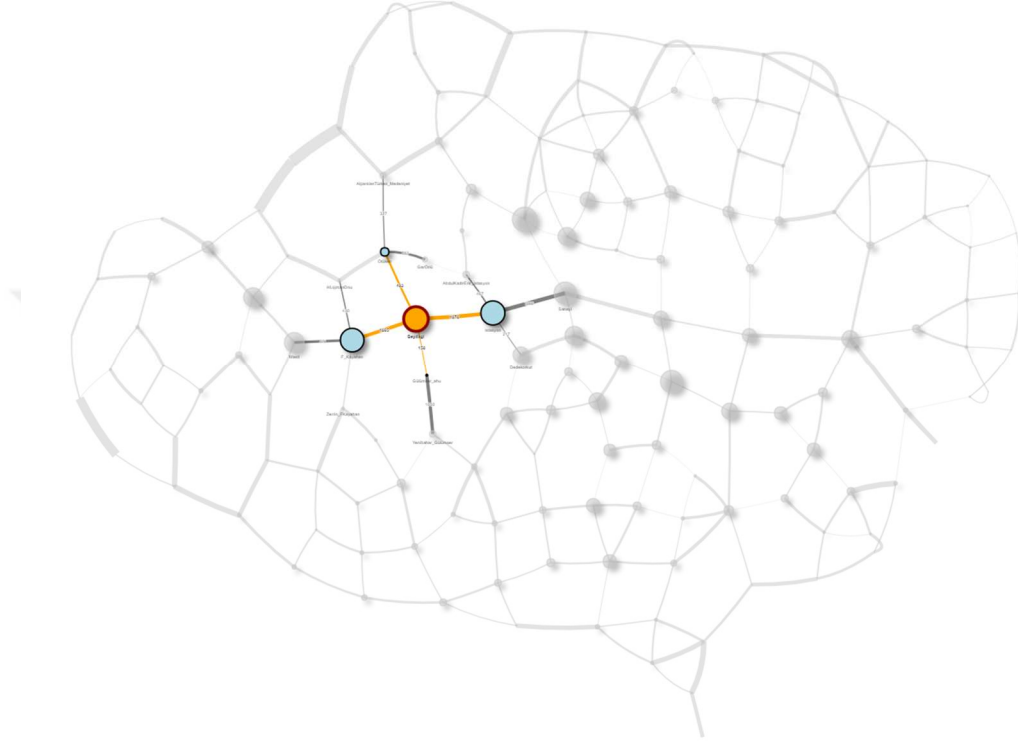
Kavşak sırası	Kavşak Adı	Sayfa değeri merkezlilik
1	Battalgazi	0,012159244
2	Kaynarca_SaatKulesi	0,012127767
3	KernekMeydan	0,011764611
4	Gonultas_Fayton	0,011331811
5	Tamirhane_Üstgeçit	0,011291699
6	AlparslanTürkes_19Mayıs	0,011233167
7	SivasYolu_Topsogut	0,010985098
8	Elmasuyu_Battalgazi	0,0104836
9	Yenibahar_Alyans	0,01027985
10	Kaynarca_KuzeyKusak	0,009922362
11	HanımınciftligiSinyal	0,009812878
12	İsmetpasa_Altınkayısı	0,009354261
13	Melekbaba_İmparator	0,00931265
14	Kayacık_Yaman	0,00927627
15	Hastane_MehmetBuyruk	0,009271501

Sayfa değeri algoritması sonuçlarına göre çizge üzerindeki kavşak noktalarının baskınlık değerleri Şekil 3.5’deki gibi görselleştirilmiştir. Şekilde turuncu olarak ifade edilen kavşak noktası sayfa değeri değeri en yüksek olan Battalgazi kavşağına aittir. Mavi renkli düğümler ise Battalgazi kavşağı ile bağlantısı olan komşu kavşak noktalarını ifade etmektedir. Görseldeki düğüm büyüklükleri sayfa değeri değerleri ile doğru orantılı olarak yansıtılmıştır.



Şekil 3.5. Sayfa değeri merkezliliğe göre Battalgazi kavşağının çizge üzerindeki konumu

Şekil 3.6’da arasındalık merkezlilik sonuçlarına göre kavşak noktalarının görsel yapısı verilmiştir. Turuncu düğüm arasındalık algoritmasına göre en baskın düğüm olan beydağı kavşağına aittir. Çizge üzerinde uygulanan arasındalık merkezliliğe göre en etkin 15 kavşak noktasına ait değerler Çizge 3.2’de verilmiştir.

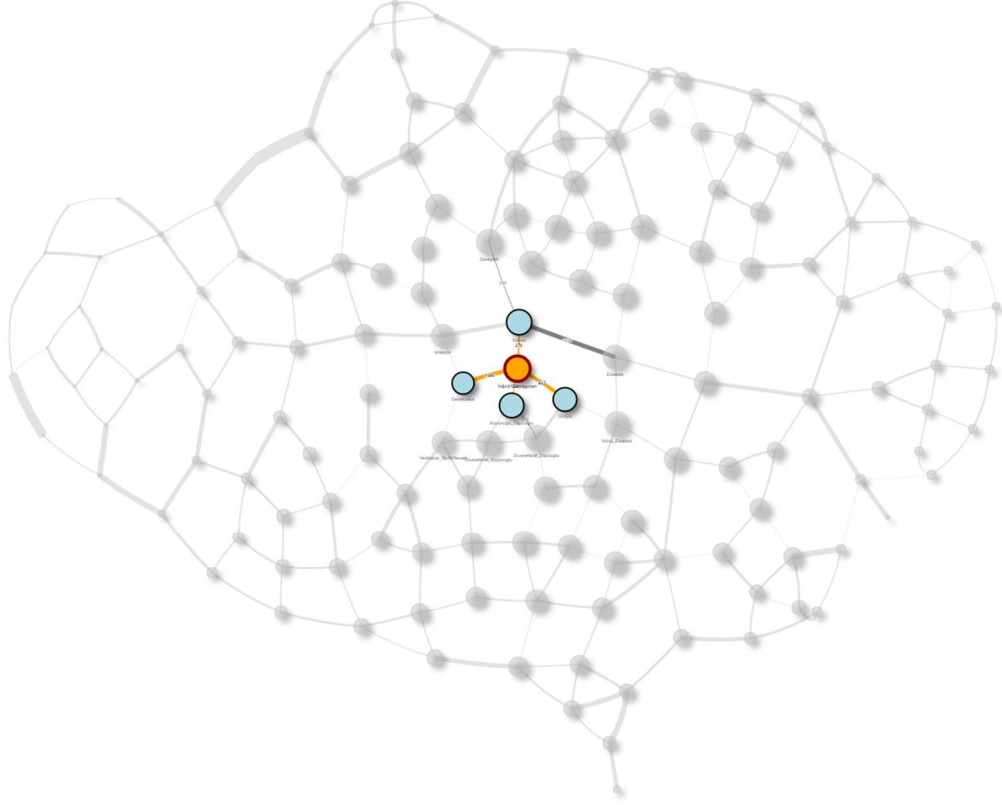


Şekil 3.6. Arasındalık merkezliliğe göre Beydağı kavşağının çizge üzerindeki konumu

Çizelge 3.2. Arasındalık algoritması merkezlilik değerleri

Kavşak sırası	Kavşak Adı	Arasındalık merkezlilik
1	Beydagi	2367
2	SanayiAlt	2353
3	Istasyon	2314
4	F_Kayahan	2268
5	Sanayi	2135
6	İnönü_Emeksiz	2131
7	Masti	1890
8	Bostanbaşı	1813
9	İnönü_Sıtmapınarı	1774
10	Sanayiİtfaiye	1714
11	Battalgazi	1602
12	Divanefendi_Zapcioglu	1553
13	Dedekorkut	1540
14	İnönü_Fuzuli	1537
15	Sanayi_Yeniyol	1427

Şekil 3.7’de yakınlık merkezlilik sonuçlarına göre ulaşım ağındaki en etkin kavşak noktası turuncu ile vurgulanan İnönü Sıtmapınarı kavşağıdır. Ulaşım ağındaki en etkin 15 kavşak noktasına ait yakınlık merkezlilik sonuçları Çizelge 3.3’de verilmiştir.

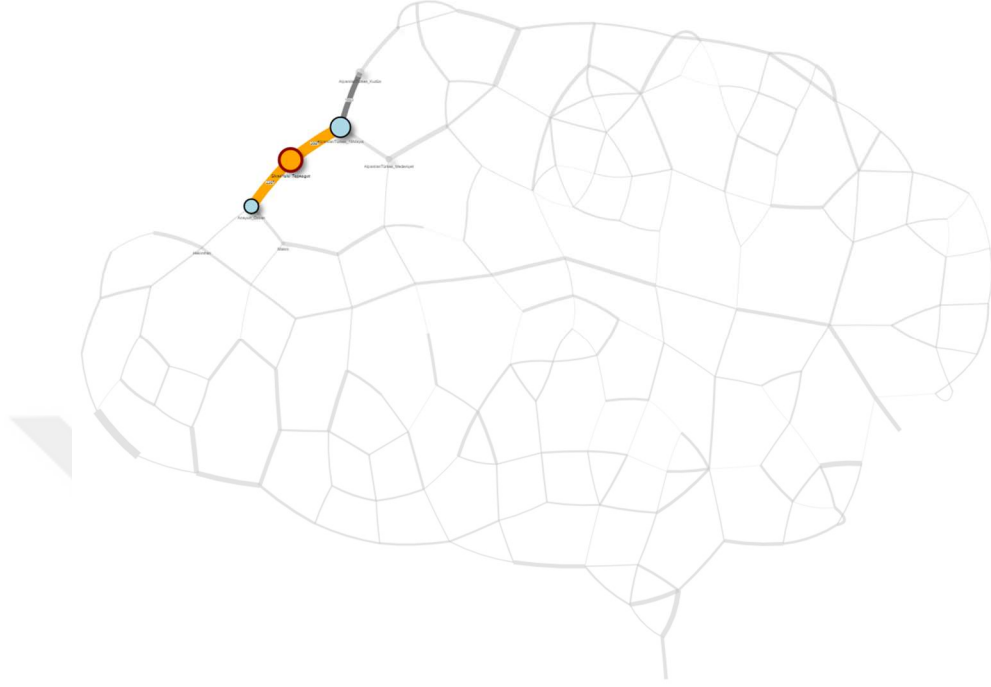


Şekil 3.7. Yakınlık merkezliliğe göre İnönü_Sıtmapınarı kavşağının çizgedeki konumu

Çizelge 3.3. Yakınlık algoritması merkezlilik değerleri

Kavşak sırası	Kavşak Adı	Yakınlık merkezlilik
1	İnönü_Sıtmapınarı	0,0000019797
2	Sanayi	0,0000019690
3	Divanefendi_Zapcioglu	0,0000019499
4	SanayiAlt	0,0000019422
5	Koyunoglu_Zapcioglu	0,0000019341
6	Sanayiİtfaiye	0,0000019234
7	İnönü_Emeksiz	0,0000019108
8	Dört Yol	0,0000018954
9	Emeksiz	0,0000018916
10	Divanefendi_Koyunoglu	0,0000018849
11	Hız_Zapcioglu	0,0000018778
12	Sanayi_Yeniyol	0,0000018672
13	Komuttevzi	0,0000018561
14	Hasanbey_Zapcioglu	0,0000018560
15	TurgutTemelli_Boztepe	0,0000018556

Şekil 3.8’de verildiği üzere özvektör merkezlilik sonuçlarına göre ulaşım ağındaki en etkin kavşak noktası SivasYolu_Topsogut kavşağıdır. Çizelge 3.4’de özvektör merkezlilik algoritmasına göre en etkin 15 kavşak noktasına ait sonuçlar verilmiştir.



Şekil 3.8. Özvektör merkezliliğe göre SivasYolu_Topsogut kavşağının çizgedeki konumu

Çizelge 3.4. Özvektör algoritması merkezlilik değerleri

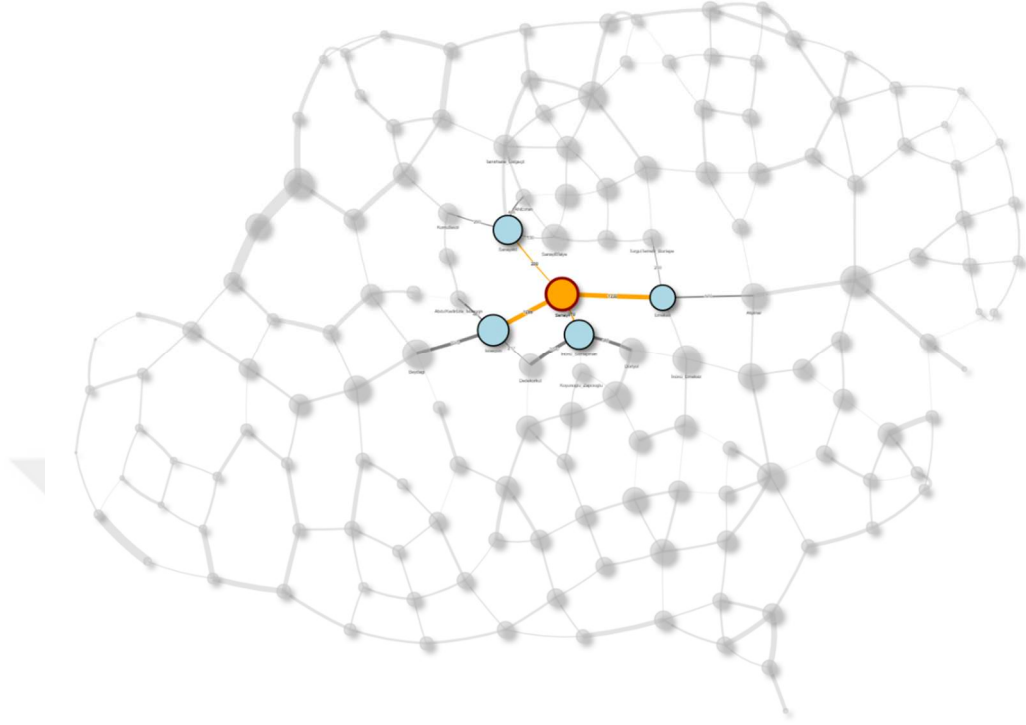
Kavşak sırası	Kavşak Adı	Özvektör merkezlilik
1	SivasYolu_Topsogut	1,000000000
2	AlparslanTürkes_19Mayıs	0,878734100
3	Anayurt_Özsan	0,634145600
4	AlparslanTurkes_Kudüs	0,340808300
5	AlparslanTürkes_Medeniyet	0,231448100
6	Makro	0,128102100
7	Dilek_Kudüs	0,087821430
8	Yesiltepe_Camlık	0,078547030
9	Hekimhan	0,060386470
10	HLojmanOnu	0,040242510
11	Ötüken	0,024440170
12	Bostanbaşı	0,019626010
13	Pasazade_Boncuk	0,018738870
14	Mahrukatcılar	0,017035820
15	Anemon	0,014869380

Oluşturulan özel ulaşım ağı çizge üzerinde dört farklı merkezilik algoritması ve bu algoritmaların hepsine ait özgün yönleri içeren hibrit yönteme ait sonuçlar Çizelge 3.5’de verilmiştir. Çizelge 3.5’de hibrit yönteme göre en etkili 15 kavşak noktasına ait veriler gösterilmektedir. Çizge üzerinde uygulanan merkezilik algoritmaları sonuçlar incelendiğinde sayfa değeri, özvektör, yakınlık ve arasındalık algoritmalarının tek başlarına kavşak noktalarının yük durumunu ifade etmekte yetersiz kaldığı sonucuna ulaşılmıştır. Algoritmalar tek başlarına incelendiğinde sayfa değeri algoritması kavşak noktalarına farklı kavşak noktalarından yapılan bağlantıyı göz önüne bulundururken, özvektör komşu kavşak noktası ile arasındaki bağlantının gücüne odaklanır, yakınlık algoritması ulaşım ağı içerisindeki kavşak noktaları arasında bütün kavşaklara en yakın mesafede olan kavşağın baskın olduğunu ifade eder, arasındalık ise bütün düğümlere en yakın konumda bulunan kenar bağlantısına yakın olan düğümlerin baskın olduğunu mantığıyla çalışmaktadır. Uygulanan hibrit yöntem ile bu algoritmalara ait bütün özgün yapının tek algoritmada olması sağlanmıştır. Önerilen hibrit yöntem özetle sayfa değeri, özvektör, yakınlık ve arasındalık algoritmalarının her birinin %25 oranında etkisinin dahil edilmesi ile geliştirilmiştir. Çizelge 3.5’deki veriler incelendiğinde hibrit yönteme göre Sanayi kavşak noktası tasarlanan ulaşım ağı içerisinde kendisine bağlı yolların sayısı ve mesafesi, diğer kavşak noktalarına olan yakınlığı ve kavşaklar arasındaki en kısa yollar üzerinde bulunmasından dolayı en baskın kavşak olarak tespit edilmiştir. Sanayi kavşağının diğer merkezilik algoritmalarındaki baskın olma sıralaması sayfa değerinde 18., özvektör’de 44., yakınlık’da 2., arasındalık’da 5. dir.

Çizelge 3.5. Merkezilik ve hibrit algoritma sonuçları

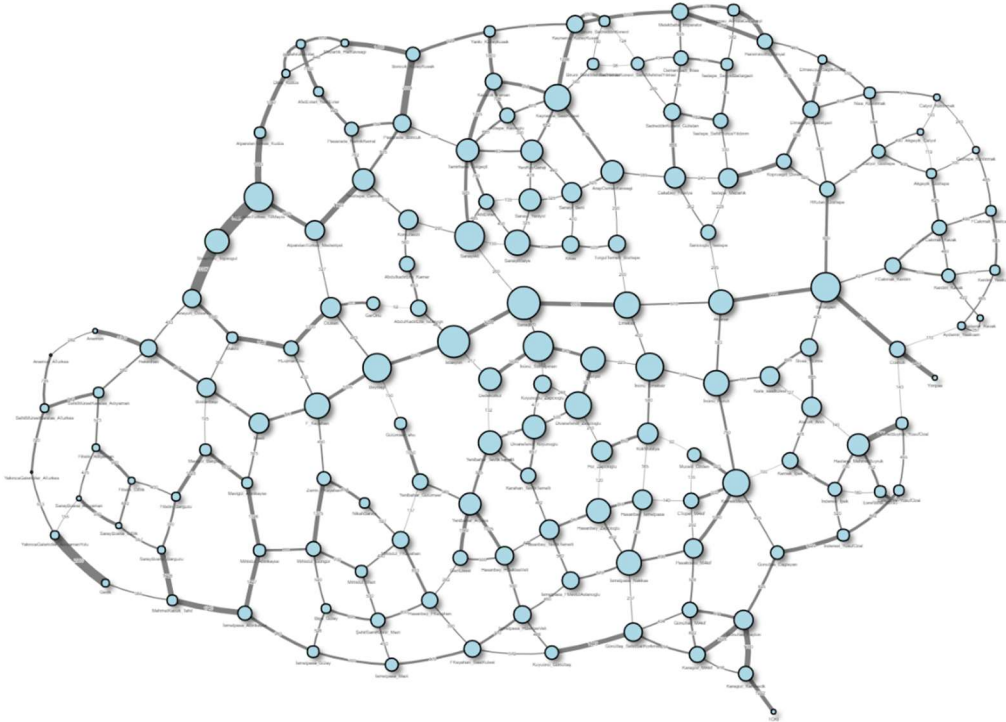
Kavşak Adı	Sayfa değeri	Özvektör	Yakınlık	Arasındalık	Hibrit
Sanayi	0,66452919	0,0001978	0,9911573	0,9019856	2,557869955
Istasyon	0,632371889	0,0006686	0,8660133	0,9776088	2,476662533
İnönü_Sıtmapınarı	0,585844395	0,0000202	1,0000000	0,7494719	2,335336525
SanayiAlt	0,342558443	0,0007375	0,9691243	0,9940853	2,30650558
Battalgazi	1	0,0000014	0,6033344	0,6768061	2,280141854
AlparslanTürkes_19Mayıs	0,902633155	0,8787341	0,3466006	0,1212505	2,249218425
Beydagi	0,495533326	0,0030901	0,7039216	1,0000000	2,202545027
İnönü_Emeksiz	0,292661243	0,0000039	0,9432920	0,9002957	2,136252799
KernekMeydan	0,958508666	0,0000001	0,7379453	0,3831855	2,079639551
Divanefendi_Zapcioglu	0,436121568	0,0000007	0,9754736	0,6561048	2,067700723
Kaynarca_SaatKulesi	0,996690538	0,0003218	0,7229657	0,3354457	2,055423714
Emeksiz	0,48766083	0,0000491	0,9274140	0,6020279	2,017151826
F_Kayahan	0,470431065	0,0043171	0,5379673	0,9581749	1,970890414
İnönü_Fuzuli	0,437948043	0,0000008	0,8766107	0,6493452	1,963904701
Sanayiİltfaiye	0,277440054	0,0000419	0,9536514	0,7241234	1,955256702

Şekil 3.9’da Sanayi kavşağının (turuncu düğüm) ve komşu kavşak noktalarının (Mavi düğümler) ulaşım ağı içerisindeki konumu verilmiştir.



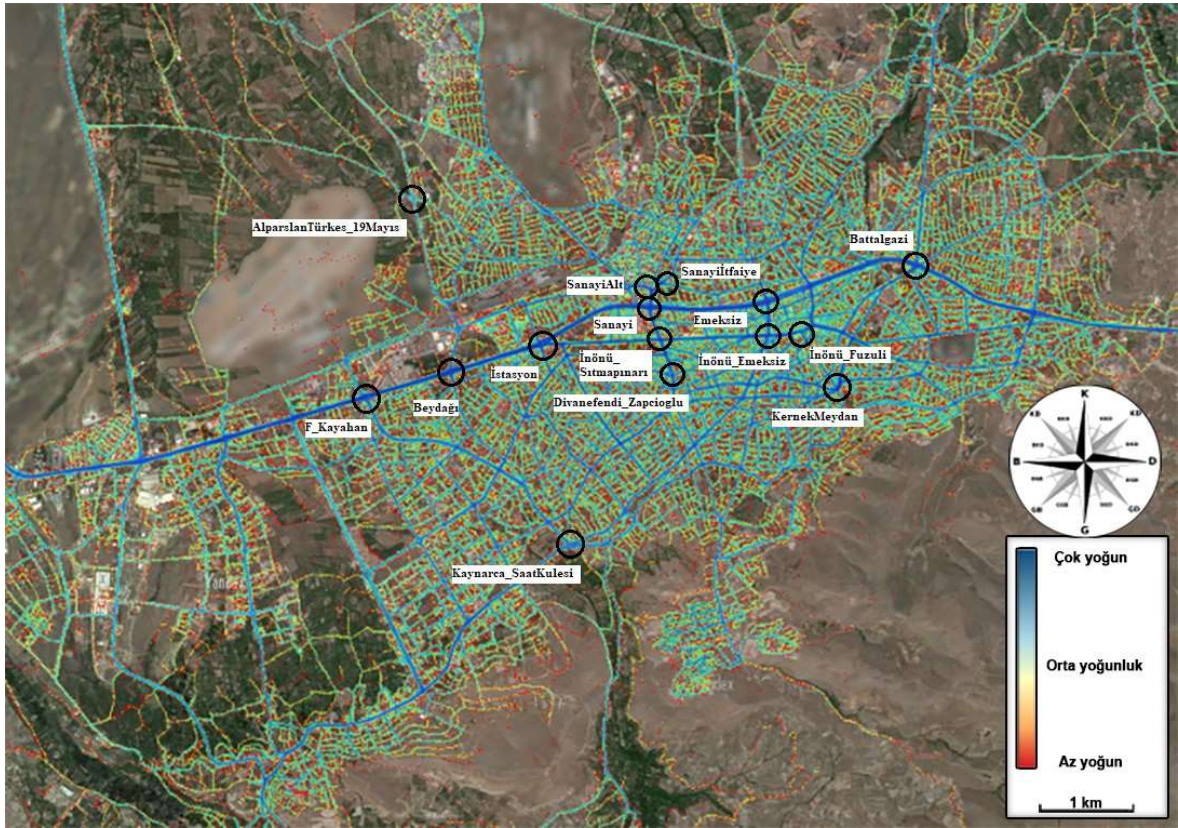
Şekil 3.9. Hibritleştirilme göre Sanayi kavşağı konum

Şekil 3.10’da hibritleştirme sonuçlarına göre ulaşım ağı yapısı görselleştirilmiştir.



Şekil 3.10. Hibritleştirilme göre çizge yapısı

Şekil 3.10’da sonuçlarının verildiği önerilen yöntemin çıktıları Yandex GPS izi üzerinde doğrulanarak en baskın olan diğer bir ifade ile yük altında olan 15 kavşak noktası Şekil 3.11’de gösterilmiştir. Bu veriler incelendiğinde çalışmada önerilen yöntemin çıktısı Çizelge 3.5’deki en kritik kavşak noktalarının GPS verilerine göre mavi renk ile belirtilen en yoğun kavşak noktaları olarak ifade edildiği görülmektedir. Şekil 3.11 üzerinde ayrıca Çizelge 3.5’de verilen kavşak noktalarının konumları harita üzerinde siyah çember ile belirtilmiştir. Verilen 15 kavşak noktasının tamamı Yandex GPS verilerine göre yüksek yoğunluklu bölge olan mavi renk ile ifade edilmiştir.

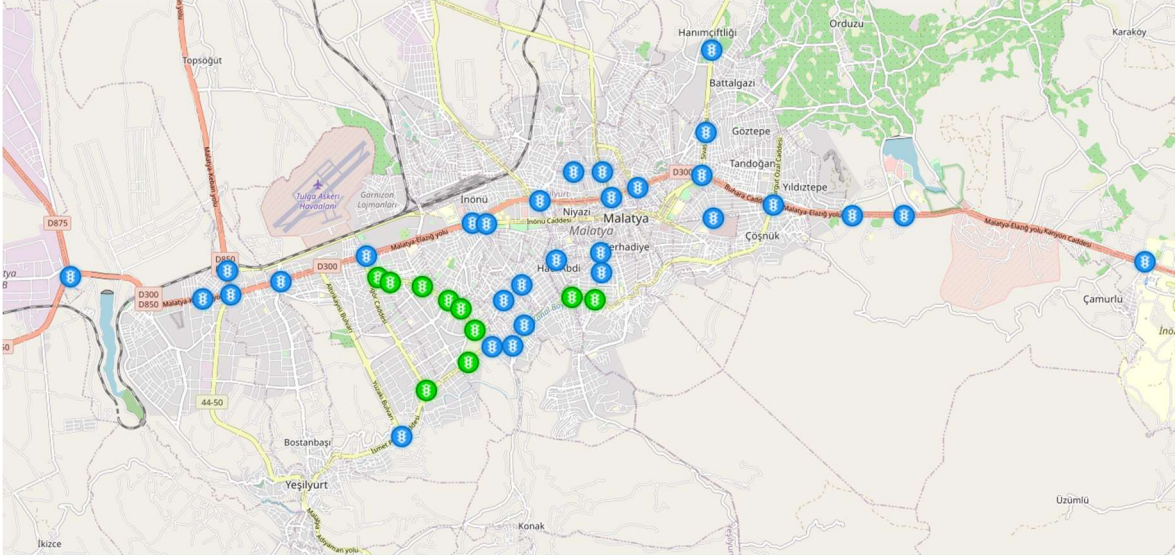


Şekil 3.11. Yandex GPS veri izine göre yoğunluk haritası[52]

Önerilen Hibrit merkezlilik yöntemi sayesinde araç sayımı veya herhangi bir cihaz ile sayısal ölçüme gerek duymadan sadece kavşak noktalarının birbirlerine olan mesafesi ve ulaşım ağı içerisindeki konumu dikkate alınarak kent içi kavşak noktalarının baskınlık değerleri hesaplanmıştır. Hesaplanan bu baskınlık verilerinin gerçek GPS verileri ile büyük oranda benzerlik gösterdiği tespit edilmiştir.

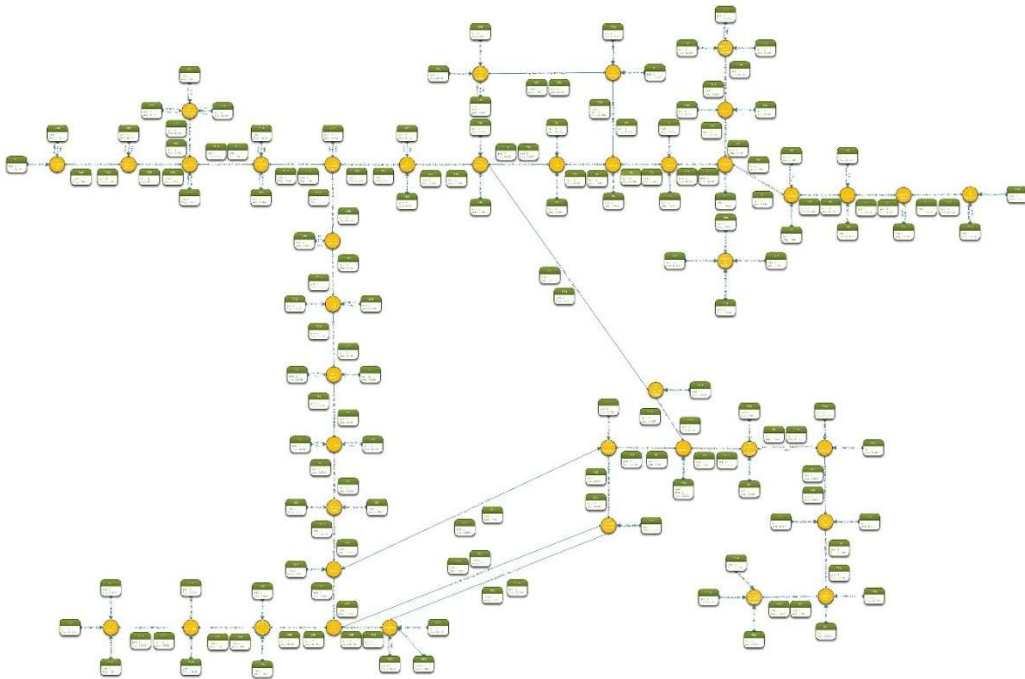
Tez çalışması sürecinde merkezlilik algoritmalarının gerçek dünya problemleri üzerine uygulanmasına yönelik yapılan bir diğer çalışma ise sinyalizasyon verileri kullanılarak kent merkezindeki kavşak noktalarının analizidir. Bu analiz çalışmasında hedef kavşak

noktalarının yük durumunun belirlenmesi ve sinyalizasyon sistemlerinin verimliliklerinin tespit edilmesidir. Bu çalışmada Malatya kent merkezinde 2019 yılına ait sinyalizasyon ve araç sayım verilerinin senkronize olarak alınabildiği 40 adet akıllı kavşak noktasına ait veriler kullanılmıştır. Bu kavşak noktalarına bağlı 158 yoldan yaklaşık 351 milyon araç geçmiştir. Şekil 3.12’de kent içerisindeki akıllı kavşak noktalarının konumları verilmiştir.



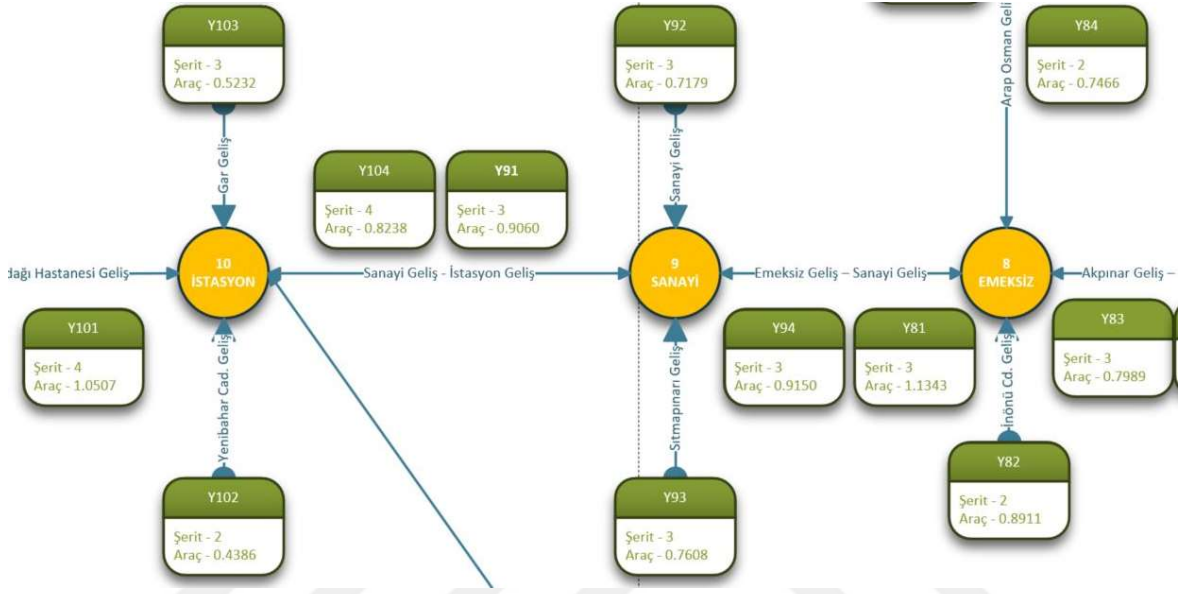
Şekil 3.12. Malatya kenti akıllı kavşak noktaları

Şekil 3.13’de akıllı kavşak noktaları ve bu kavşak noktalarına bağlı yolların oluşturduğu çizge yapısı verilmiştir.



Şekil 3.13. Akıllı kavşak ulaşım ağı çizgesi

Şekil 3.14’de akıllı kavşak ulaşım ağına ait çizgenin yakınlaştırılmış hâli verilmiştir. Çizge üzerinde her kavşak noktasını besleyen yolların kaç şerit olduğu ve 1 yeşil saniyesi süresince kaç araç geçtiği bilgileri verilmiştir. Ağırlıklandırılmış çizgede örneğin “12 Beydağı” kavşağına ait durum incelendiğinde Y121 yolundan 1 yeşil saniyesinde 0,8762, Y122 yolundan 0,5238, Y123 yolundan 0,54, Y124 yolundan 0,4945 araç kavşaktan geçmiştir.



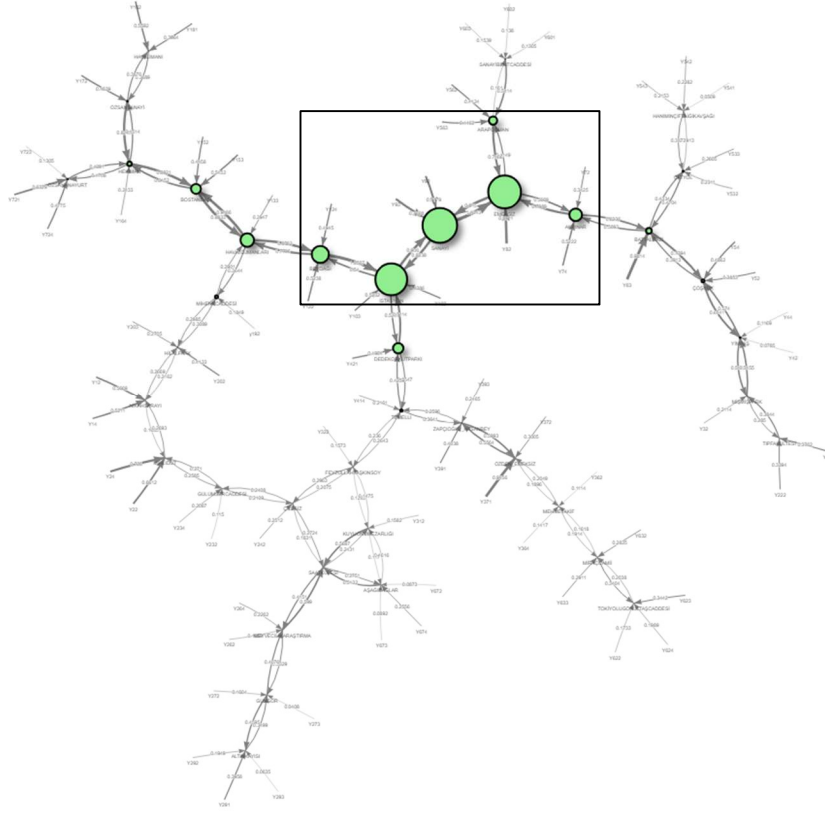
Şekil 3.14. Ulaşım ağı çizgesi yakınlaştırılmış hâli

Ağırlıklandırılmış çizge üzerinde özvektör ve sayfa değeri merkezlilik algoritması uygulanmıştır. Özvektör merkezliliğe göre en etkin 10 kavşağına ait sonuçlar Çizelge 3.6’da verilmiştir.

Çizelge 3.6. Özvektör merkezliliğe göre en etkin 10 kavşak noktası

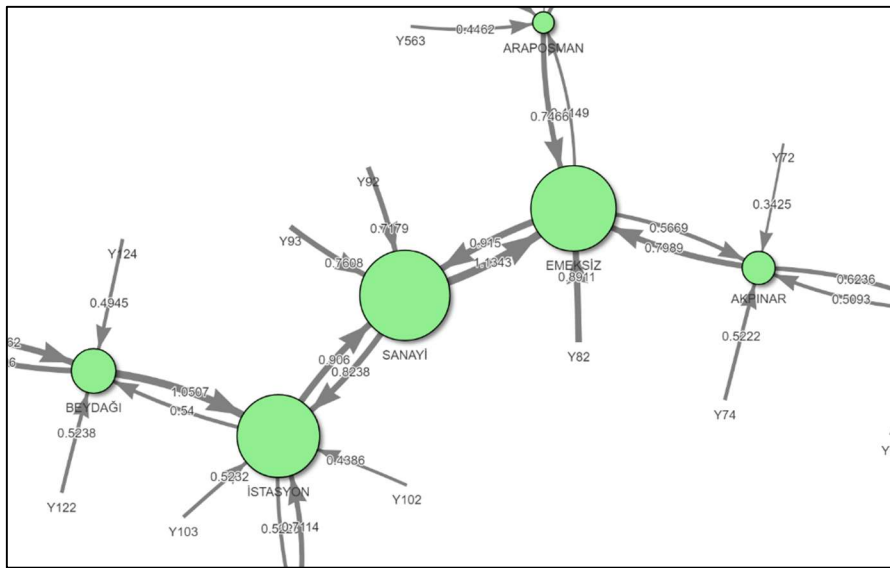
Etkin düğüm sıralaması	Kavşak Adı	Özvektör
1	SANAYİ	1
2	EMEKSİZ	0,946
3	İSTASYON	0,917078
4	BEYDAĞI	0,49412
5	HAVALOJMANLARI	0,391501
6	AKPINAR	0,366035
7	DEDEKORKUTPARKI	0,299151
8	BOSTANBAŞI	0,287254
9	ARAPOSMAN	0,238587
10	BATTALGAZİ	0,166261

Özvektör merkezlilik değerlerine göre oluşturulan çizgeye ait görsel Şekil 3.15’de verilmiştir.



Şekil 3.15. Özvektör algoritmasına göre ulaşım ağı kavşak noktaları

Şekil 3.16’da Şekil 3.15’de dikdörtgen içerisinde belirtilen özvektör merkezlilik değeri en yüksek olan Sanayi kavşağı ve komşu kavşak noktalarının çizge üzerindeki konumu verilmiştir. Şekilden de anlaşılacağı üzere Sanayi, Emeksiz ve İstasyon kavşak noktaları özvektör merkezlilik değerleri en yüksek olan kavşak noktalardır.



Şekil 3.16. Özvektör algoritmasına göre ulaşım ağı kavşak noktaları yakınlaştırılmış hâli

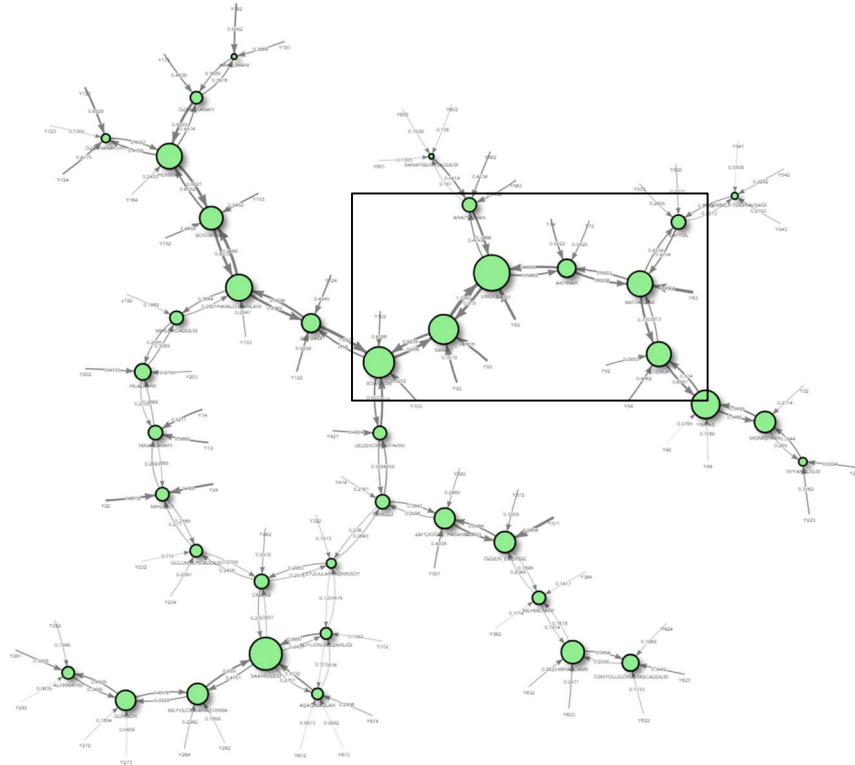
En yüksek merkezlilik değerine sahip olan Sanayi kavşağı incelendiğinde 1 yeşil saniyesi süresince ortalama olarak Y92 yolundan 0,7179 araç, Y93 yolundan 0,7608 araç, Emeksiz kavşağından 0,915 araç, İstasyon kavşağından 0,906 araç sanayi kavşağına gelmektedir.

Aynı ulaşım ağı üzerinde sayfa değeri algoritması uygulanmış ve sonuçları Çizelge 3.7’de verilmiştir.

Çizelge 3.7. Sayfa değeri merkezliliğe göre en etkin 10 kavşak noktası

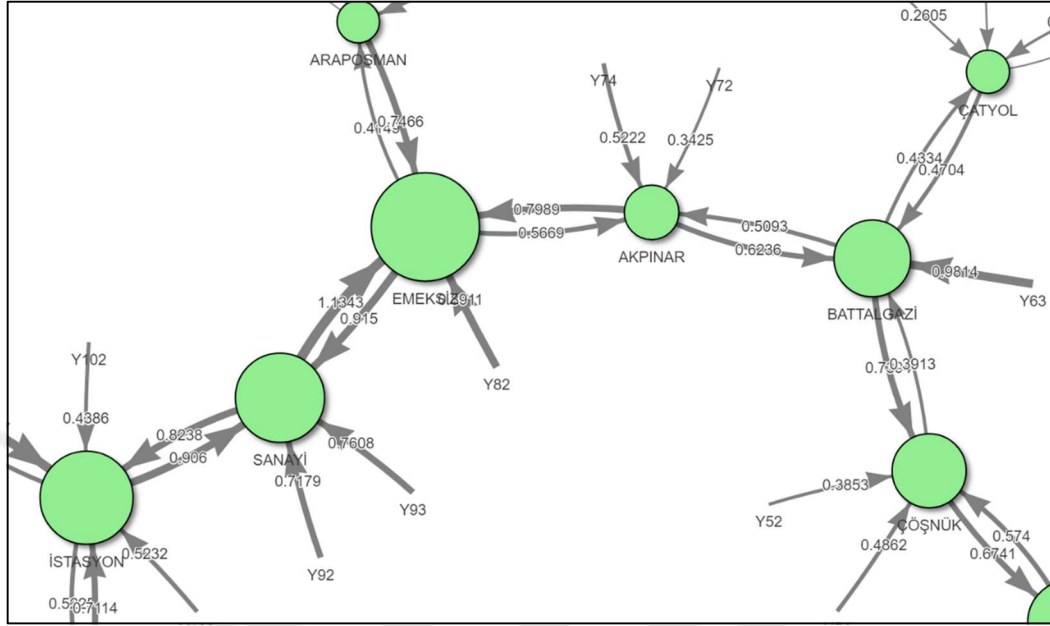
Etkin düğüm sıralaması	Kavşak Adı	Sayfa değeri
1	EMEKSİZ	0,043940534
2	SAAT KULESİ	0,03989051
3	İSTASYON	0,038007032
4	SANAYİ	0,03646169
5	YİMPAŞ	0,034739874
6	HAVALOJMANLARI	0,032686023
7	HEKİMAN	0,031565594
8	BATTALGAZİ	0,031549173
9	ÇÖŞNÜK	0,030495898
10	BOSTANBAŞI	0,028792932

Sayfa değeri merkezlilik değerlerine göre en baskın kavşak noktaları Emeksiz, Saat Kulesi, İstasyon noktalarıdır. Şekil 3.17’de sayfa değeri merkezlilik değerlerine göre kavşak noktalarının ulaşım çizgesi üzerindeki durumu verilmiştir.



Şekil 3.17. Sayfa değeri algoritmasına göre ulaşım ağı kavşak noktaları

Şekil 3.18’de şekil 3.17’de dikdörtgen içerisinde belirtilen sayfa değeri merkezliliğe göre en baskın olan kavşak noktası olan Emeksiz ve komşu kavşak noktalarının çizge üzerindeki konumu verilmiştir.



Şekil 3.18. Sayfa değeri algoritmasına göre ulaşım ağı kavşak noktaları yakınlaştırılmış hâli

En yüksek sayfa değeri merkezlilik değerine sahip Emeksiz kavşağı incelendiğinde, 1 yeşil saniyesi süresince ortalama olarak Y82 yolundan 0,8911 araç, Sanayi kavşağından 1,1343, Arap Osman kavşağından 0,7466 araç, Akpınar kavşağından 0,7989 emeksiz kavşağından geçmiştir.

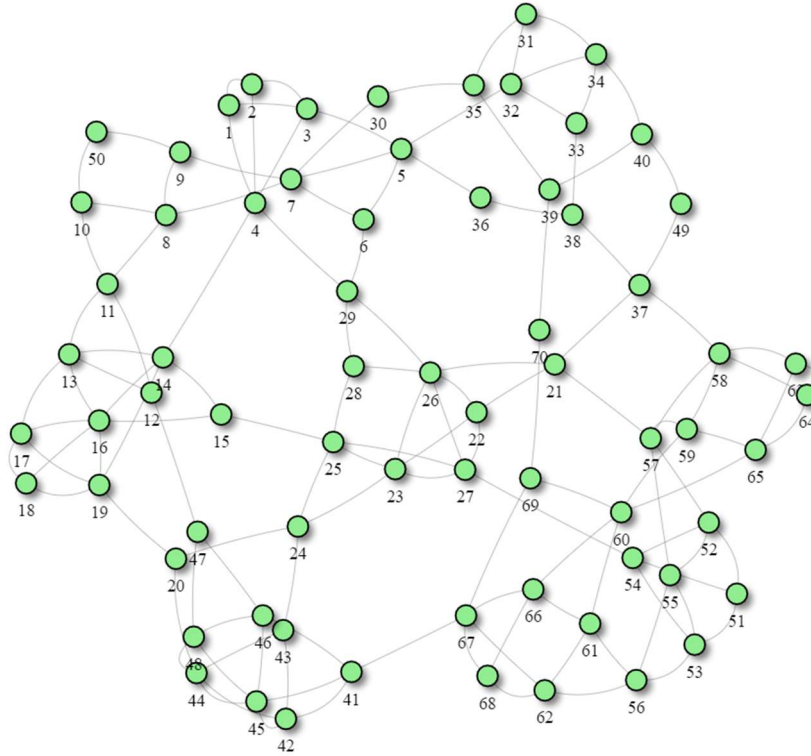
Kavşak noktalarından geçen araç sayıları ve her iki algoritmaya ait merkezlilik değerlerine göre ilk 10 kavşak noktası Çizelge 3.8’de verilmiştir.

Çizelge 3.8. En yoğun ve en baskın 10 kavşak noktası

Sıra No	Araç Sayısı	Sayfa değeri	Özvektör
1	İSTASYON	SANAYİ	EMEKSİZ
2	BOSTANBAŞI	EMEKSİZ	SAAT KULESİ
3	EMEKSİZ	İSTASYON	İSTASYON
4	HAVALOJMANLARI	BEYDAĞI	SANAYİ
5	BEYDAĞI HASTANESİ	HAVALOJMANLARI	YİMPAŞ
6	SANAYİ	AKPINAR	HAVALOJMANLARI
7	HEKİMHAN	DEDEKORKUTPARKI	HEKİMHAN
8	ÇÖŞNÜK	BOSTANBAŞI	BATTALGAZİ
9	AKPINAR	ARAPSMAN	ÇÖŞNÜK
10	BATTALGAZİ	BATTALGAZİ	BOSTANBAŞI

Çizelge 3.8 incelendiğinde Emeksiz, Sanayi ve İstasyon kavşaklarının hem araç sayıları açısından hemde sayfa değeri ve özvektör merkezlilik değerleri açısından üst sıralarda olduğu görülmektedir. Bu durum bu 3 kavşak noktasının çok yoğun olduğunu ve mevcut çalışan sinyalizasyon programına göre aşırı yük altında olduğunu göstermektedir. Bu kavşak noktalarındaki yükün azaltılması için ulaşım ağında alternatif kavşak noktaları oluşturularak bu kavşakları kullanan araç sayılarının azaltılması yada mevcut sinyalizasyon planlarının optimize edilerek 1 yeşil sn süresince daha fazla aracın kavşaktan geçmesi sağlanmalıdır. En çok aracın geçtiği 2. Kavşak olan Bostanbaşı kavşağı incelendiğinde her iki merkezlilik algoritması sonuçlarında da en baskın kavşak noktaları arasında bulunmamaktadır. Bu durum Bostanbaşı kavşağının araç trafik yükünü Emeksiz, Sanayi, İstasyon vb.. kavşak noktalarına göre daha başarılı olarak kaldırdığını göstermektedir.

Literatürdeki popüler merkezlilik algoritmaları ile 2 farklı gerçek dünya probleminin çözümü için uygulamalar yapılmıştır. Bu aşamada tez çalışmasının iki önemli çıktısından birisi olan Karcı merkezlilik algoritmasının diğer merkezlilik algoritmaları ile karşılaştırılmasını içeren bir çalışma yapılmıştır. Şekil 3.19'da 70 düğüm ve 133 kenar bağıntısı içeren örnek bir çizge üzerinde sayfa değeri, yakınlık, özvektör, derece ve önerilen karcı merkezlilik algoritması uygulanmıştır. Tez çalışmasının sonraki bölümlerinde önerilen Karcı(Γ) merkezlilik algoritmasının tüm aşamaları ayrıntılı olarak açıklanmıştır.



Şekil 3.19. 70 düğümlü çizge

Merkezlilik algoritmalarının örnek çizge üzerinde uygulanması neticesinde Çizelge 3.9’da en etkili 8 düğüm, Çizelge 3.10’da ise en etkisiz 8 düğüme ait bilgiler verilmiştir. Çizelge 3.9 incelendiğinde 60. düğüm önerilen Karcı merkezlilik algoritmasına göre en etkili düğüm olarak tespit edilmiştir. 60. Düğüm aynı zamanda sayfa değeri algoritmasına göre en etkili düğüm sıralamasında 7. sırada bulunmaktadır. Sayfa değeri ölçütüne göre çizgedeki en etkili düğüm 7. düğüm olarak tespit edilmiştir. 7. düğüm derece algoritmasına göre 5. sırada bulunmaktadır. 26. düğüm yakınlık algoritması için 1. sırada, sayfa değeri algoritması için 3. sırada, özvektör ve derece algoritmaları için 2. sırada bulunmaktadır. 16. düğüm ise özvektör ve derece algoritmaları için en etkili düğüm iken sayfa değeri algoritması için 2. en etkili düğüm olarak tespit edilmiştir. 16. Düğüm yakınlık ve Karcı merkezlilik değerlerine göre en etkili ilk 8 düğüm içerisinde yer almamıştır.

Çizelge 3.9. Tüm merkezlilik ölçütlerine ait en etkili 8 düğüm

Etkili düğüm sıralaması	Sayfa değeri	Yakınlık	Özvektör	Derece	Karcı (Γ)
1	7	26	16	16	60
2	16	29	26	26	69
3	26	21	23	4	29
4	5	23	27	5	38
5	58	22	14	7	4
6	67	27	19	13	55
7	60	28	25	14	56
8	4	25	17	19	40

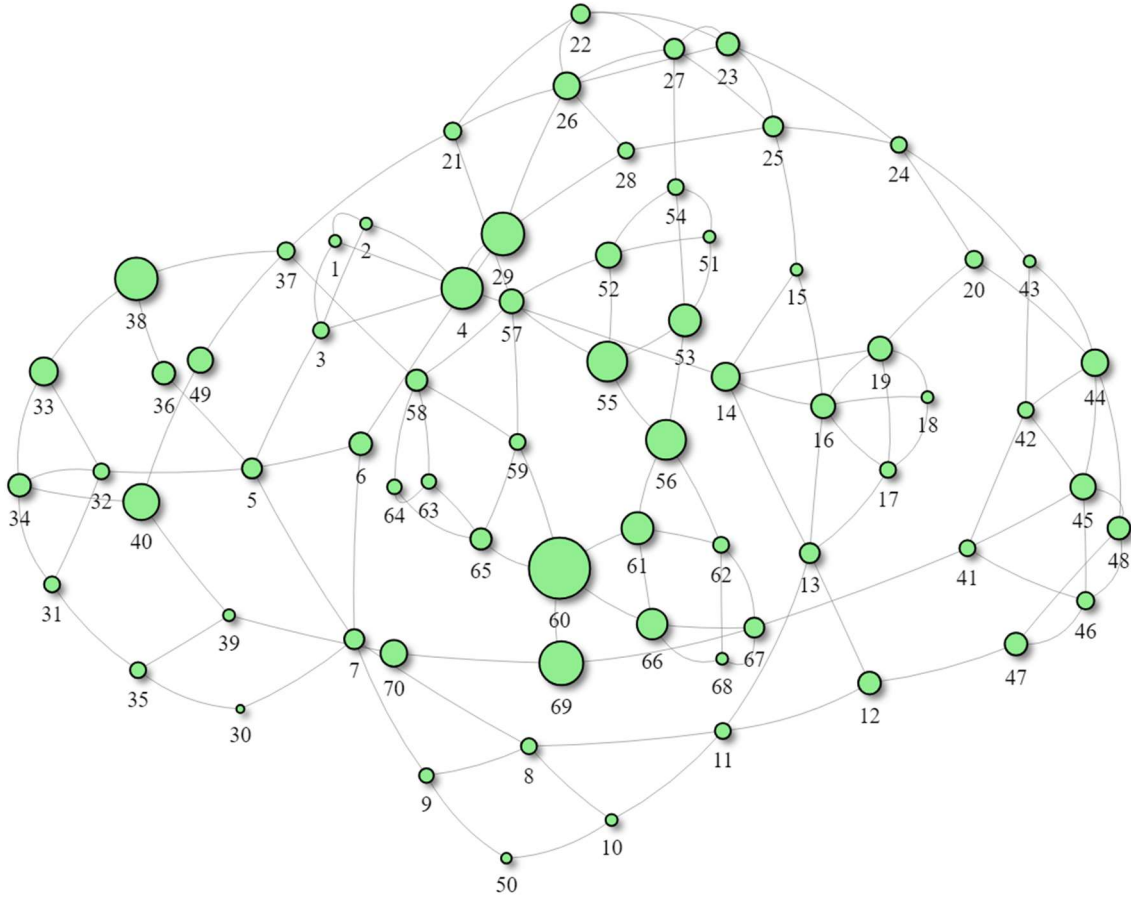
Çizelge 3.10’da verilen merkezlilik algoritmalarının hepsi için çizgedeki en etkisiz 8 düğüme ait bilgiler verilmiştir. Karcı merkezlilik değerine göre çizge içerisindeki en etkisiz düğüm 30. düğümdür. 30. düğüm derece algoritması içinde en etkisiz düğüm olarak tespit edilmiştir. Diğer algoritma sonuçlarına göre 30. düğüm sayfa değeri için 3. sırada, özvektör için 7. sırada bulunan en etkisiz düğümdür. Çizelgeki sonuçlar incelendiğinde en etkisiz düğümler için büyük benzerlikler olduğu görülmektedir.

Çizelge 3.10. Tüm merkezlilik ölçütlerine ait en etkisiz 8 düğüm

Etkisiz düğüm sıralaması	Sayfa değeri	Yakınlık	Özvektör	Derece	Karcı (Γ)
1	36	50	39	30	30
2	49	18	35	36	50
3	30	10	70	49	1
4	50	63	40	50	2
5	70	64	31	70	10

6	15	68	34	1	15
7	18	31	30	2	18
8	28	17	33	6	39

Karcı merkezlilik algoritmasının sonuçlarının çizge üzerindeki görsel hâli Şekil 3.20’de verilmiştir. Düğümün merkezlilik değerlerinin büyüklükleri görseldeki düğümlerin çap büyüklükleri ile doğru orantılıdır.



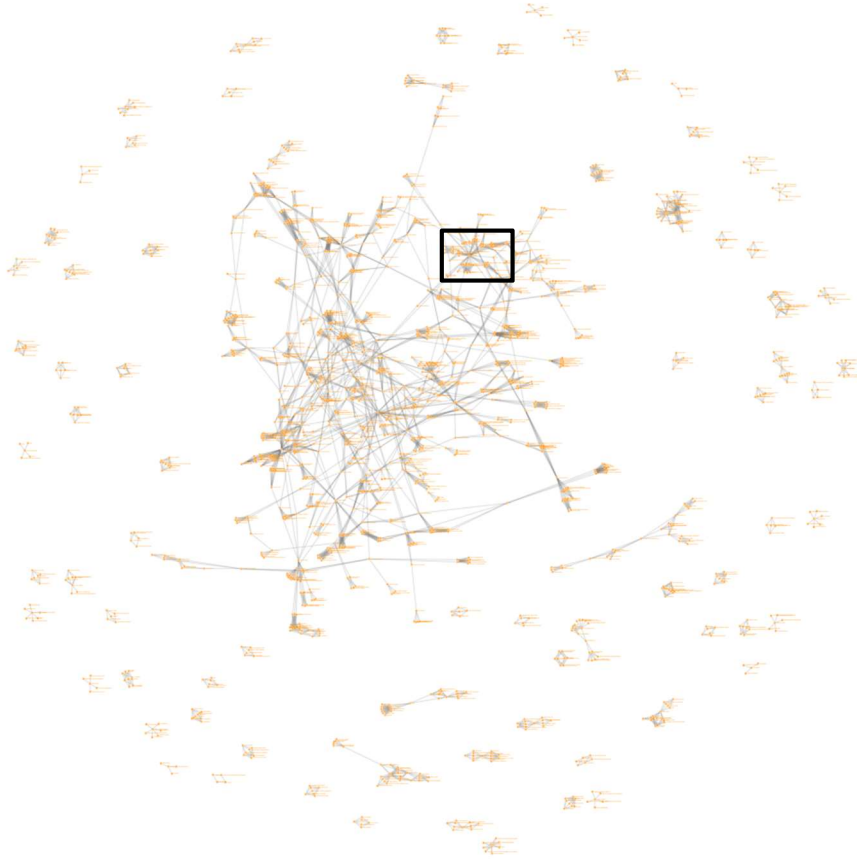
Şekil 3.20. Karcı (Γ) merkezliliğin örnek çizge üzerindeki etkinlik (baskınlık) görünümü

Uygulamada Karcı merkezlilik algoritması ile başarısı ispatlanmış popüler merkezlilik algoritmalarından sayfa değeri, yakınlık, özvektör ve derece merkezlilik algoritmalarının karşılaştırmalı sonuçlarına yer verilmiştir. Her algoritmanın kendine özgü çözüm yaklaşımına göre bazı algoritmalarda benzer sonuçlar olabileceği gibi bazı algoritmalar arasında düşük benzerlik olacağı sonucuna ulaşılmıştır. Önerilen karcı merkezlilik algoritması etkili ve etkisiz düğüm sıralamalarında diğer popüler merkezlilik algoritmaları ile kısmi benzerlik göstererek başarılı sonuçlar vermiştir.

3.2 Topluluk Tespiti Algoritmaları

Tasarlanan sosyal ağ içerisindeki bir düğüm kümesi, kendi içerisinde dışarıya göre daha fazla etkileşim içerinde veya daha çok sayıda bağlantı içeriyor ise bu düğüm kümesi bir topluluk olarak nitelendirilebilir [53]. Diğer bir tanıma göre gerçek ağlar rastgele olmayan heterojen bir yapıdan oluşmaktadır. Ağ içerisindeki belirli bir alanda yoğunlaşan, küme oluşturan ve topluluk olarak adlandırılan yapılar aynı veya benzer rolü bulunan düğümlerdir [54,55].

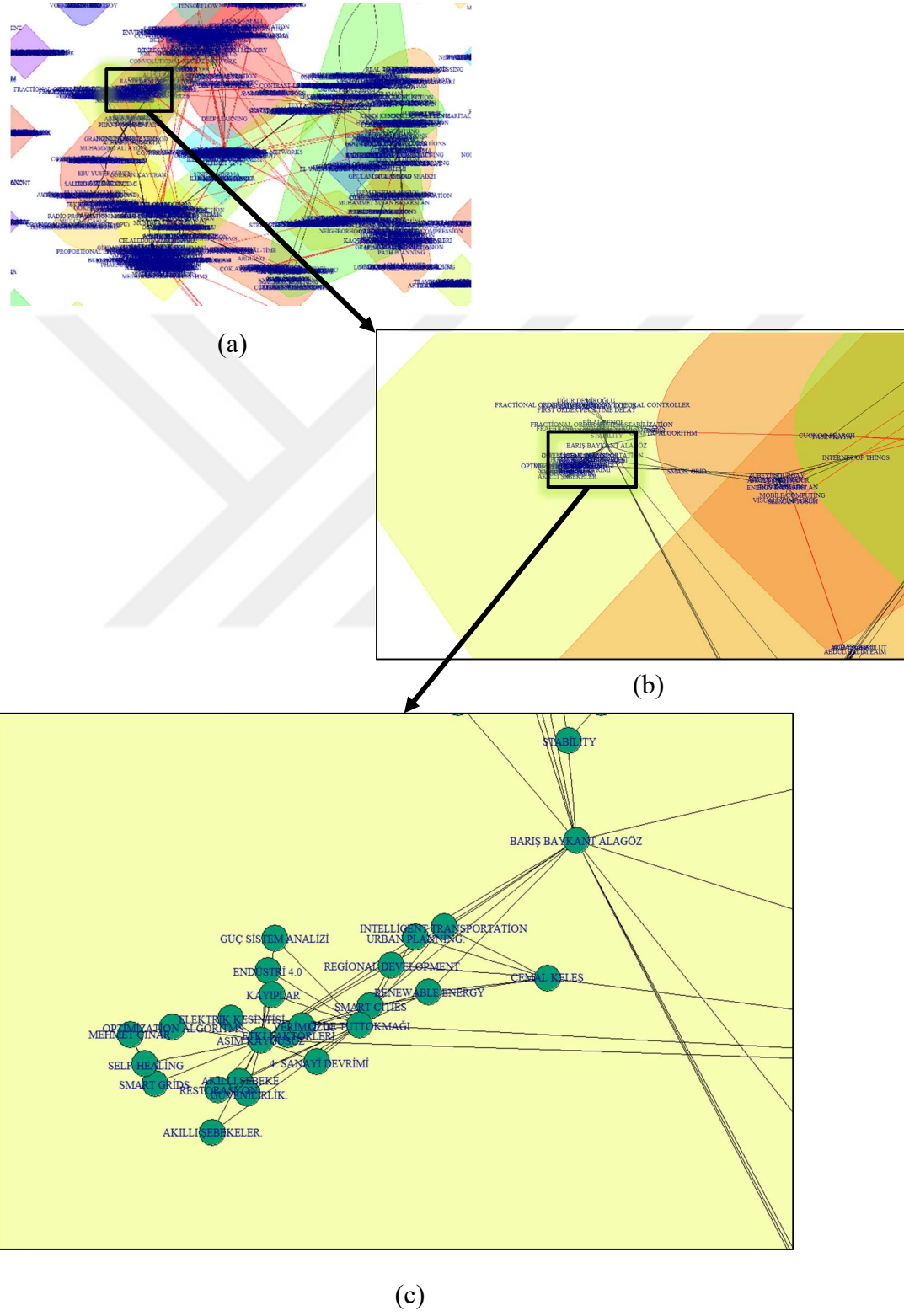
Gerçek veriler üzerinde yapılan bir diğer çalışmada cluster edge betweenness algoritması kullanılarak sosyal ağ üzerinde topluluk tespiti işlemleri yapılmıştır [28]. Şekil 3.21’de uygulamada kullanılan uluslararası yapay zeka ve veri işleme sempozyumunun (IDAP) 2017 yılı verilerine göre oluşturulan çizge yapısı verilmiştir. Çizge akademik çalışma yapan 536 yazar ile bu yazarların yayınlarında kullandıkları 1188 anahtar kelimelerden oluşmaktadır.



Şekil 3.21. IDAP Verilerine ait çizge yapısı

Şekil 3.22’de Şekil 3.21’deki dikdörtgen içerisinde verilen alandaki çizge bölümünün yakınlaştırılmış hâli verilmiştir.

Yazarların yayınlarında belirttikleri anahtar kelimeler kullanılarak yazarların gruplandırılması yapılmıştır. Bu sayede belirli grupların hakimiyeti altındaki araştırma konuları veya belirli akademik konular üzerinde hakimiyeti bulunan yazarların tespiti sağlanmıştır.



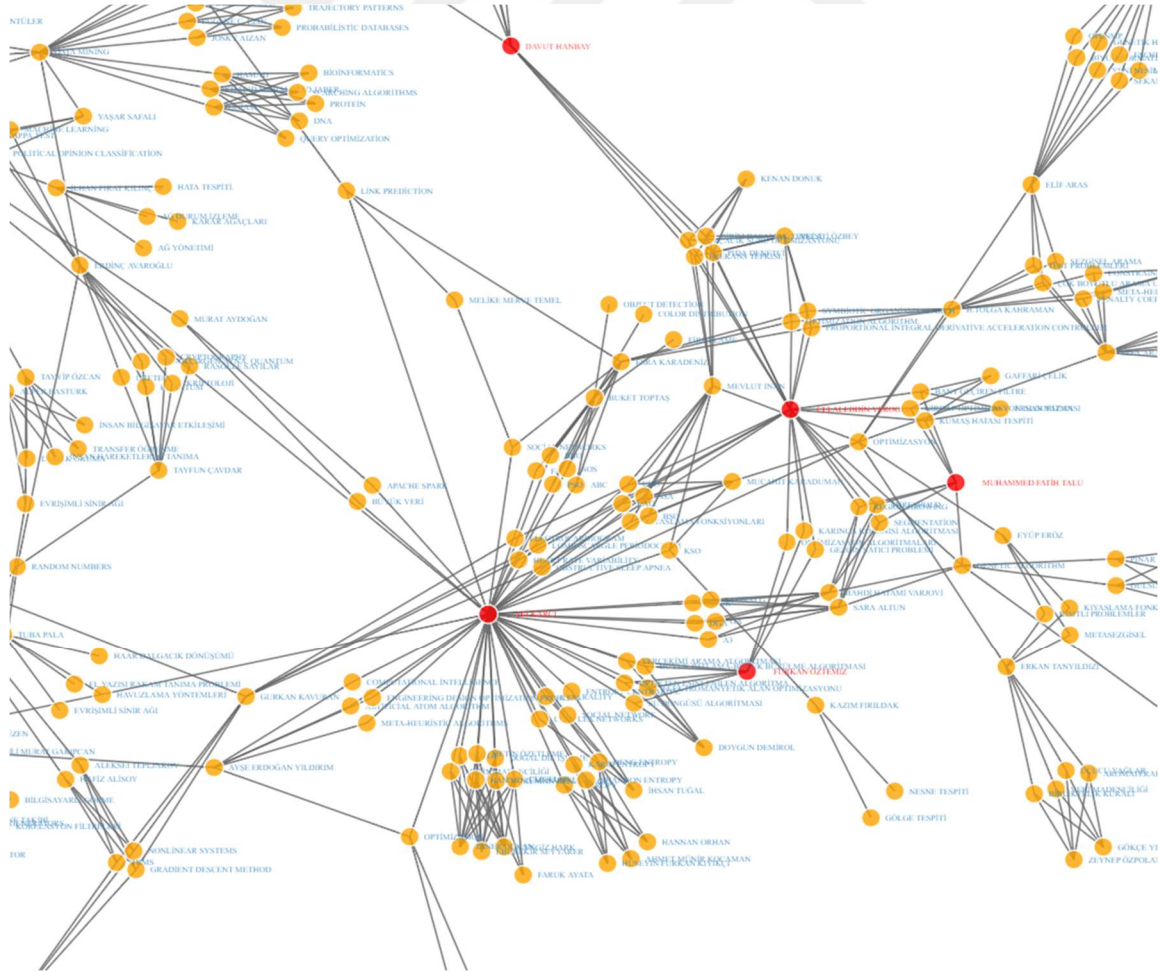
Şekil 3.23. Çizge üzerindeki örnek bir topluluk tespiti kesiti

Şekil 3.23’de topluluk tespitine ait örnek bir kesit verilmiştir. Şekil 3.23’de verilen (b) görüntüsü (a) görüntüsünde, (c) görüntüsü ise (b) görüntüsünde dikdörtgen içerisinde belirtilen alanın yakınlaştırılmış hâlidir. Şekil 3.23 incelendiğinde yazarlardan Cemal KELEŞ ve Barış Baykant ALAGÖZ akademik çalışmalarında kullandıkları anahtar kelimelerin benzerliklerinden dolayı aynı küme içerisinde kalmaktadır.

3.3 Benzerlik Algoritmaları

Metinler arasındaki benzerliklerin çıkarılması için geliştirilen algoritmalar [56]. Bu bölümde IDAP verileri kullanılarak oluşturulan çizge üzerinde Jaccard, Cosine ve Euclidean benzerlik yöntemleri uygulanmıştır. Ayrıca İnönü Üniversitesi Bilgisayar Mühendisliği Bölümü (İBM) Akademisyenlerinin yayınlarında kullandıkları anahtar kelimeler dikkate alınarak birbirleri arasındaki benzerlik sonuçlarına yer verilmiştir.

Şekil 3.24’de verilen çizgede, kırmızı düğümler ile belirtilen İBM akademisyenlerinden bazılarının ait bağlantılar ve konumlar gösterilmiştir.



Şekil 3.24. Çizge üzerinde İBM akademisyenlerinden bazılarının ait bağlantılar

3.3.1 Jaccard algoritması

İki metin arasındaki ortak özelliklerin, toplam özellik sayısına oranı bize Jaccard benzerlik değerini verir [57]. Jaccard benzerlik indeksi 0 ile 1 arasında değer üretir ve bu değer yükseldikçe benzerlik oranının artmaktadır. Benzerlik oranı yükseldiğinde benzerlik mesafesi kısılacağı için hesaplanan Jaccard mesafesi benzerlik indek değeri 1 den çıkarılarak elde edilir. İlgili Jaccard benzerlik formülleri denklem (7) ve (8)'de verilmiştir [58].

$$\text{Benzerlik}(A, B) = |A \cap B| / |A \cup B| \quad (7)$$

$$J\text{delta}(A, B) = 1 - J(A, B) \quad (8)$$

Çizelge 3.12'de İBM'de görev yapan hocaların IDAP konferansına göre Jaccard benzerlik değerleri verilmiştir. Sonuçlar incelendiğinde Adnan Fatih Kocamaz'ın bölümdeki hocalar ile benzer çalışmalar yapmadığı görülmektedir. Benzerlik değeri yayınlarında kullandıkları anahtar kelimelere göre en yüksek olan yazarlar ise Muhammet Fatih Talu ile Celaleddin Yeroğlu ve Muhammet Fatih Talu ile Sara Altun olarak tespit edilmiştir.

Çizelge 3.12. Jaccard benzerlik sonuçları

Jaccard Benzerlik	Adnan Fatih Kocamaz	Ali Karcı	Barış Baykant Alagöz	Bilal Şenol	Celaleddin Yeroğlu	Davut Hanbay	Furkan Öztemiz	Muhammed Fatih Talu	Sara Altun
Adnan Fatih Kocamaz	1	0	0	0	0	0	0	0	0
Ali Karcı	0	1	0	0	0	0	0,1	0	0,1
Barış Baykant Alagöz	0	0	1	0,32	0	0	0	0	0
Bilal Şenol	0	0	0,32	1	0	0	0	0	0
Celaleddin Yeroğlu	0	0	0	0	1	0,15	0,12	0,33	0,15
Davut Hanbay	0	0	0	0	0,15	1	0	0	0
Furkan Öztemiz	0	0,10	0	0	0,12	0	1	0	0
Muhammed Fatih Talu	0	0	0	0	0,33	0	0	1	0,33
Sara Altun	0	0	0	0	0,15	0	0	0,33	1

3.3.2 Cosine algoritması

Trigonometrideki kosinüs fonksiyonu üzerinden iki metin arasındaki mesafeyi belirlemeyi amaçlar [59]. Metinler birer vektör olarak düşünüldüğü bu yaklaşımda iki vektörün birbiri ile ilişkisi bir açı ile ifade edilmektedir. Aynı yönü gösteren iki vektörün kosinüs değeri 1 dir ($\cos(0) = 1$). Tamamen birbiri ile ilişkisiz olan vektörler için ise kosinüs değeri 0 olacaktır

($\cos(90) = 0$). Tamamen birbirine zıt olan dökümanlar için aralarında ilişki yoktur denemez, kosinüs değerleri -1 olacaktır. Bu yapı ile iki metin içerisindeki tüm değerler +1 ve -1 arasında sayısallaştırılacaktır [60]. Cosine benzerlik indeksine ait formül denklem (9)'da verilmiştir [61].

$$\text{Cosine}(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} \quad (9)$$

Çizelge 3.13'de verilen cosine benzerlik değerlerine göre 0,58 değeri ile yaptığı akademik çalışmalar ile aralarında en çok benzerlik bulunan yazarlar Muhammed Fatih Talu ile Celaleddin Yeroğlu'dur. İkinci sırada ise 0,5 değerine sahip Muhammed Fatih Talu ile Sara Altun ve Bilal Şenol ile Barış Baykant Alagöz isimli yazarlar bulunmaktadır.

Çizelge 3.13. Cosine benzerlik sonuçları

Cosine Benzerlik	Adnan Fatih Kocamaz	Ali Karcı	Barış Baykant Alagöz	Bilal Şenol	Celeleddin Yeroğlu	Davut Hanbay	Furkan Öztemiz	Muhammed Fatih Talu	Sara Altun
Adnan Fatih Kocamaz	0	0	0	0	0	0	0	0	0
Ali Karcı	0	0	0	0	0	0	0,26	0	0,24
Barış Baykant Alagöz	0	0	0	0,5	0	0	0	0	0
Bilal Şenol	0	0	0,5	0	0	0	0	0	0
Celeleddin Yeroğlu	0	0	0	0	0	0,23	0,23	0,58	0,29
Davut Hanbay	0	0	0	0	0,29	0	0	0	0
Furkan Öztemiz	0	0,26	0	0	0,23	0	0	0	0
Muhammed Fatih Talu	0	0	0	0	0,58	0	0	0	0,5
Sara Altun	0	0	0	0	0,29	0	0	0,5	0

3.3.3 Euclidean algoritması

En popüler uzaklık ölçütlerinden birisi olmakla birlikte çok boyutlu uzaydaki nesnelerin birbirlerine geometrik uzaklığıdır. Nesnelerin birbirinden ne kadar farklı olduklarını tespit etmek için kullanılır. Birbirlerinden ayrı ve yoğun veri seti içeren kümelerde iyi sonuçlar üretir. İki nesne birbirine ne kadar yakın ise Euclidean (İbn-i Sina) uzaklığıda o ölçüde sıfıra yakın olur [62].

Denklem (10)'da İbn-i Sina benzerlik ölçütüne ait formül verilmiştir [62].

$$\text{Euclidean}(A, B) = \sqrt{\sum_{i=1}^k (A_i - B_i)^2} \quad (10)$$

Euclidean (İbn-i Sina) mesafe yöntemi Jaccard ve Cosine yöntemlerinin aksine değeri küçük olan yazarlar arasındaki benzerlik bağlantısı büyüktür. Çizelge 3.14’de verilen İbn-i Sina benzerlik verileri incelendiğinde 2,8 birim ile Muhammed Fatih Talu ile Sara Altun isimli yazarlar arasındaki benzerlik bağlantısı en güçlüdür. Yaptıkları yayınlar ile birbirlerine en az benzerlik gösteren yazarlar 9,6 değerine sahip olan Ali Karcı ile Adnan Fatih Kocamaz olarak tespit edilmiştir.

Çizelge 3.14. Euclidean benzerlik sonuçları

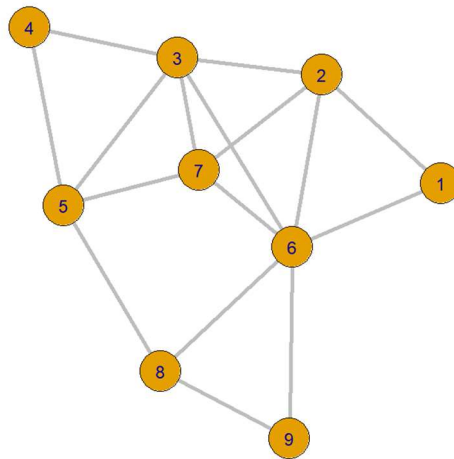
Euclidean Benzerlik	Adnan Fatih Kocamaz	Ali Karcı	Barış Baykant Alagöz	Bilal Şenol	Celaleddin Yeroğlu	Davut Hanbay	Furkan Öztemiz	Muhammed Fatih Talu	Sara Altun
Adnan Fatih Kocamaz	0	9,6	7,6	7,2	8	7,2	7,1	7,1	7,2
Ali Karcı	9,6	0	8	7,5	8,3	7,6	6,8	7,5	6,80
Barış Baykant Alagöz	7,7	8	0	3,6	6	5	4,9	4,8	5
Bilal Şenol	7,2	7,5	3,6	0	5,5	4,3	4,1	4	4,2
Celaleddin Yeroğlu	8	8,3	6	5,5	0	4,7	4,8	3,70	4,70
Davut Hanbay	7,2	7,5	5	4,3	4,7	0	4,1	4	4,2
Furkan Öztemiz	7,1	6,8	4,9	4,1	4,8	4,1	0	3,9	4,1
Muhammed Fatih Talu	7,1	7,4	4,8	4	3,8	4	3,9	0	2,80
Sara Altun	7,2	6,9	5	4,3	4,7	4,3	4,1	2,80	0

4. GELİŞTİRİLEN YÖNTEMLER

Önerilen yöntem herhangi bir çizge türünde minimum hakim kümeye (MDS) yakın sonuçlar veren ve bütün çizge türlerinde hakim kümeyi (DS) veren özgün bir algoritmadır. Sunulan algoritma kendi içerisinde birden fazla algoritma içermektedir. Özetle algoritma özgün olarak tasarlanan bir merkezlilik algoritmasını kullanarak hakim küme üyelerini belirlemeyi amaçlamaktadır. Bu merkezlilik algoritması düğüm dereceleri, Karcı Max tree dereceleri ve temel kesme derecelerinin hesaplanması ile tespit edilmektedir. Bu merkezlilik algoritmasının çıktısı Karcı merkezlilik değeri olarak isimlendirilmiş ve tez içerisindeki farklı bölümlerde düğüm baskınlık değeri olarak ifade edilecektir. Düğüm baskınlık değeri DS üyeleri tespit edilirken gerekli iterasyonlarda kullanılacaktır. Bu gereklilik çizge üzerinde ilgili iterasyon sırasında pendant düğüm olup olmaması ile alakalıdır. Eğer pendant (derecesi 1 olan düğüm) düğüm yoksa düğüm baskınlık değerleri hesaplanır, olmaması durumunda pendant düğümün atası (komşu düğüm) DS üyesi olarak belirlenir. Aşağıda algoritmanın bütün aşamaları örnek üzerinde ayrıntılı olarak verilmiştir.

4.1 Düğüm Dereceleri

Çizge üzerindeki düğümlere ait kenar bağlantıları sayısı o düğümlerin derecelerini belirtmektedir. $K \subseteq \{\{x, y\} | x, y \in D \text{ and } x \neq y\}$ örnek çizgeye ait kenar listesi $K = \{\{1,2\}, \{1,6\}, \{2,6\}, \{2,7\}, \{2,3\}, \{3,4\}, \{3,5\}, \{3,6\}, \{3,7\}, \{4,5\}, \{5,7\}, \{5,8\}, \{6,7\}, \{6,8\}, \{6,9\}, \{8,9\}\}$. Şekil 4.1'de örnek çizgeye ait görsel verilmiştir.



Şekil 4.1. Örnek çizge

Verilen örnek çizgeye ait düğüm dereceleri Çizelge 4.1. belirtilmiştir.

Çizelge 4.1. Örnek çizgeye ait düğüm dereceleri

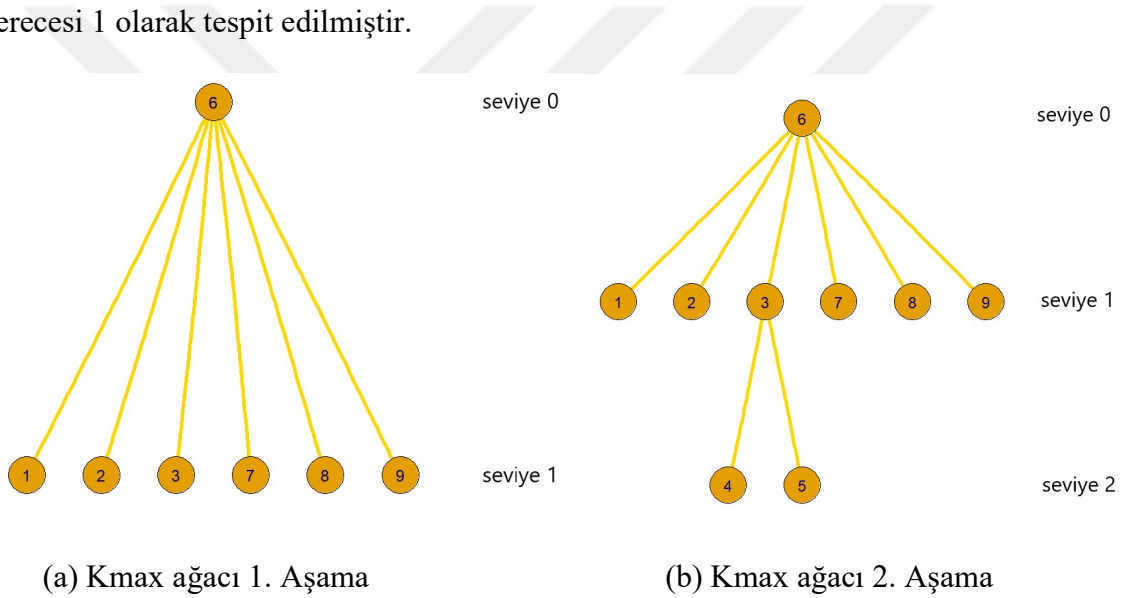
Düğüm ismi	1	2	3	4	5	6	7	8	9
Düğüm derecesi	2	4	5	2	4	6	4	3	2

4.2 Karcı Maksimum Kapsayan Ağaç(Karci Maximum Spanning Tree)

İkinci aşamada Karcı maksimum kapsayan (Kmax) ağaç oluşturuldu. Kmax ağacı düğüm dereceleri yüksek olan düğümleri önceliklendirerek oluşturulan özel bir kapsayan ağaçtır. Oluşturulan Kmax ağacı temel kesmelerin gerçekleştirilmeside kullanılacaktır. Kmax ağacı oluşturulurken çizgedeki en yüksek dereceli düğümden başlanarak komşu düğümlere ulaşılır. Ulaşılan komşu düğümün derecesi 1 azaltılarak, ulaşılan bütün komşu düğümler kuyruğa güncel dereceleri ile eklenir. Bir sonraki iterasyon başlamadan kuyruktaki düğümlerin düğüm dereceleri güncellenir. Derece güncellemesinde temel amaç sadece daha önce ulaşılmayan düğümler ile varolan kenar bağlantılarının düğüm derecesi olarak sayılmasıdır. O yüzden her iterasyon sonrasında ulaşılan düğümlerin birbirleri arasındaki bağlantıları kesilmektedir. Ulaşılan düğümlerin kuyruğa atılmasının sebebi herhangi bir iterasyonda düğüm dereceleri eşit olan iki veya daha fazla düğüm var ise düğümlerin kuyruk sırasına (İlk giren ilk çıkar prensibine) göre seçim önceliği almasıdır. Kuyruktaki bulunan komşu düğümler arasından bir sonraki maksimum derecedeki düğümü seçerken eşitlik olması durumunda öncelikle ağaçtaki seviyesine bakılır. Ulaşılan komşu düğümlerden ağaçtaki seviyesi düşük olan düğüme öncelik verilir, eğer iki veya daha fazla düğümün derecesi ve ağaçtaki seviyesi eşit ise, o zaman kuyruktaki sırasına gözönünde bulundurularak bir sonraki maksimum düğüm seçimi yapılır.

Şekil 4.1’de verilen çizge üzerinde Kmax ağacının oluşturulma süreci açıklanacak olursa, Çizelge 4.1’de belirtildiği gibi düğüm derecesi en yüksek olan düğümün 6 numaralı düğüm olduğu görülmektedir. Şekil 4.2’nin a bölümünde görüldüğü üzere 6. düğümden başlanıp komşularına ulaşılarak Kmax ağacının ilk aşaması tamamlanmıştır. İlk aşama incelendiğinde 6. Düğümün komşuları (1,2,3,7,8 ve 9) nolu düğümlerdir. Kmax ağacı oluşturulurken maksimum düğümün komşuları indeks sırasına göre kuyruğa eklenir.

Sonraki iterasyona geçmeden önce (1,2,3,7,8 ve 9) numaralı düğümlerin derecesi 1 azaltılır (6 numaralı düğümün tüm kenar bağlantıları kesilir) ve düğüm dereceleri güncellenir. Düğüm dereceleri güncellenirken daha önce ziyaret edilen düğümlere bağlanan düğümlerin kenarları silinir. Bu güncellemenin sebebi bir düğüme bir kenar üzerinden ulaşıldığı için diğer alternatif yollara artık ihtiyaç kalmamış olmasıdır. Örneğin çizgeye göre 7. düğümün düğüm derecesi (2,3,5,6) düğümlerine olan bağlantıdan dolayı 4'dür. Kmax ağacında ilk olarak maksimum düğüm olan 6 nolu düğüm ile bağlantılı kenarlar silinir ve bu sayede 6. düğümün komşuları olan 2, 3 nolu düğümlere ulaşıldığı için 7. düğüm üzerinden 2 ve 3 nolu düğümlere tekrar ulaşmaya gerek kalmamıştır. Bu sayede 2. ve 3. düğümlerin 7 düğüm ile bağlantısı kesilir ve 7 düğümün daha önce Kmax ağacı üzerinden ulaşılmamış olan 5. düğüm ile bağlantısı kalmıştır. Bu sayede düğüm güncelleme işlemi sonrasında 7. düğümün düğüm derecesi 1 olarak tespit edilmiştir.



Şekil 4.2. Kmax ağacı aşamaları

Bu güncelleme işlemi Kmax ağacı tamamlanana kadar her iterasyonda gerçekleştirilir. 2. iterasyon öncesi güncellenen düğüm dereceleri Çizelge 4.2'de verilmiştir.

Çizelge 4.2. Komşu düğümlerin güncel düğüm dereceleri

Düğüm ismi	1	2	3	7	8	9
Düğüm derecesi	0	0	2	1	1	0

Çizelge 4.2'ye göre 1, 2 ve 9 numarası düğümlerin daha önce ulaşılmamış herhangi bir düğüm ile bağlantısı kalmadığı için bu düğümlerin düğüm dereceleri 0 dır. Düğüm 3'ün 2, düğüm 7 ve düğüm 8'in daha önce ulaşılmamış bir düğüm ile bağlantısı bulunmaktadır. Güncelleme sonrası 1,2,3,7,8,9 numaralı düğümlerin kuyrukdaki durumu Çizelge 4.3'de belirtilmiştir. Normalde derecesi 0 olan düğümlerin kuyruktan direk silinirken algoritmanın daha anlaşılabilir olması için Çizelge 4.3'de ve Çizelge 4.4'de gösterilmiştir.

Çizelge 4.3. 1.iterasyon komşu düğümlerin kuyrukdaki durumu

Kuyruk sırası	1	2	3	4	5	6
Düğüm ismi	1	2	3	7	8	9
Düğüm derecesi	0	0	2	1	1	0

Bu aşamada 1,2,3,6,7,8,9 numaralı düğümlere ulaşılmıştır. Eğer bütün düğümlere ulaşılsaydı, Kmax algoritması çalışmayı durduracaktı. 2. iterasyonda düğüm derecesi 2 ile en yüksek olan düğüm, düğüm 3 olarak tespit edilmiştir. Eğer düğüm derecesi 2 olan başka bir düğüm olsaydı önce bu düğümlerin Kmax ağacındaki seviyelerine bakılır, eğer seviyeleri de aynı ise bu düğümlerin kuyruktaki önceliğine göre maksimum düğüm seçimi yapılır.

Maksimum düğüm olarak seçilen düğüm 3'ün komşuları düğüm 4 ve düğüm 5 dir. Bu düğümlerin Kmax ağacındaki görünümü Şekil 4.2'nin b bölümünde verilmiştir. Düğüm 4 ve düğüm 5'e ilk defa ulaşıldığı için ulaşılrken kullanılan kenar bağlantısı silinir diğer bir deyişle bu düğümlerin düğüm dereceleri 1 azaltılır. Ayrıca maksimum dereceye sahip olan düğüm 3'ün bağlantıları silinir. Son durumda kuyruktaki düğümlerin güncel hâli Çizelge 4.4'de verildiği gibi olmaktadır.

Çizelge 4.4. 2. iterasyon komşu düğümlerin kuyrukdaki durumu

Kuyruk sırası	1	2	3	4	5	6	7	8
Düğüm ismi	1	2	3	7	8	9	4	5
Düğüm derecesi	0	0	0	0	0	0	0	0

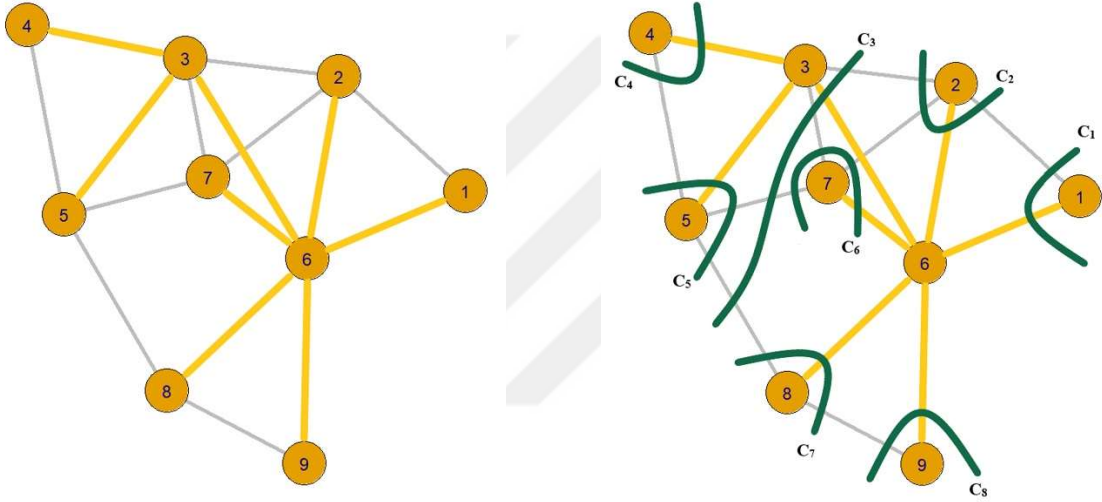
Kuyrukta bulunan bütün düğümlerin dereceleri 0 olduğu için Kmax ağacı oluşturma süreci tamamlanmıştır ve çizgedeki bütün düğümlere ulaşım sağlanmıştır. Oluşturulan Kmax ağacının 2 önemli işlevi vardır. Birincisi Kmax ağacının düğüm derecelerinin belirlenmesi, ikincisi ise Kmax ağacındaki kenar bağlantılarının temel kesme işlemlerinde kullanılacak olmasıdır. Kmax ağacının düğüm dereceleri Çizelge 4.5'de verilmiştir.

Çizelge 4.5. Kmax ağacı düğüm dereceleri

Düğüm ismi	1	2	3	4	5	6	7	8	9
Düğüm derecesi	1	1	3	1	1	6	1	1	1

4.3 Temel Kesmeler

Temel kesme dereceleri hesaplanırken Kmax ağacının tüm kenarları sırası ile kesilerek (silinerek) çizgeden çıkarılır ve bu kesme işlemi sonucunda çizge iki alt çizgeye bölünür. Bu oluşan iki alt kümedeki düğümlerin birbirleri ile arasındaki bağlantılar belirlenerek kesme işlemi uygulanır ve etkilenen düğümler belirlenir. Örneğin Şekil 4.3'ün a bölümünde Kmax ağacının çizge üzerindeki durumu verilmiştir. Belirlenen Kmax ağacına göre gerçekleştirilecek tüm temel kesmeler Şekil 4.4'ün b kısmında verilmiştir. Her kesme işleminde Kmax ağacında belirtilen sadece bir kenar kesilecektir.



(a) Örnek çizge ve Kmax ağacının kesişimi

(b) Çizgedeki tüm temel kesmeler

Şekil 4.3. Örnek çizgeye ait Kmax ağacı ve tüm kesmeler

1. temel kesme incelendiğinde Kmax ağacında düğüm 1 ve düğüm 6'yı bölmektedir. Bundan sonraki süreçte n. temel kesme C_n olarak adlandırılacaktır. C_1 (1. Temel kesme) = $\{\{1\}, \{2,3,4,5,6,7,8,9\}\}$ numaralı düğümler iki farklı gruba dönüşecektir. İki grup arasındaki bağlantılar incelendiğinde düğüm 1-2 ve düğüm 1-6 arasındaki kenar bağlantıları tespit edilmiştir. İfade kolaylığı olması açısından düğüm a ve düğüm b arasındaki kenar bağlantısı "a-b" olarak ifade edilecektir. C_1 uygulandıktan sonra 1-2 ve 1-6 bağlantıları düğüm 1 için 2, düğüm 2 ve düğüm 6 için 1 temel kesme derecesi vermiştir. Bu şekilde C_1 den C_8 'e kadar bütün kesme işlemleri açıklanarak uygulanmıştır.

- C_2 (2. Temel kesme) = $\{\{2\}, \{1,3,4,5,6,7,8,9\}\}$ olacak şekilde düğümleri iki gruba bölmektedir. Bu iki grubun birbirleri arasındaki kenar bağlantıları incelendiğinde 1-2, 2-6, 2-7, 2-3 kenarları kesme işleminden etkileniyor. Bu kesme işlemi neticesinde düğüm 2 nin temel kesme derecesi 4 olurken, düğüm 1, 3, 6 ve 7 nin temel kesme derecesi 1 olarak hesaplanır.
- $C_3 = \{\{3,4,5\}, \{1,2,6,7,8,9\}\}$ numaralı düğümleri 2 gruba bölerken bu kesme işleminden 3-2, 3-6, 3-7, 5-7, 5-8 kenarları etkilenir. C_3 kesmesi düğüm 3'e 3, düğüm 5 ve 7'ye 2, düğüm 2, 6 ve 8'e 1 temel kesme derecesi kazandırmıştır.
- $C_4 = \{\{4\}, \{1,2,3,5,6,7,8,9\}\}$ numaralı düğümleri 2 gruba böler ve bu kesme işlemi neticesinde düğüm 4'e 2, düğüm 3 ve 5'e 1 kesme derecesi kazandırmıştır.
- $C_5 = \{\{5\}, \{1,2,3,4,6,7,8,9\}\}$ numaralı düğümleri iki gruba bölerek düğüm 5'e 4 kesme derecesi, düğüm 3,4,7 ve 8'e ise 1 kesme derecesi kazandırmıştır.
- $C_6 = \{\{7\}, \{1,2,3,4,5,6,8,9\}\}$ numaralı düğümleri iki gruba bölerek düğüm 7'ye 4 kesme derecesi, düğüm 2,3,5 ve 6 'ya 1 kesme derecesi kazandırmıştır.
- $C_7 = \{\{8\}, \{1,2,3,4,5,6,7,9\}\}$ numaralı düğümleri iki gruba bölerek düğüm 8'e 3, düğüm 5,6 ve 9'a 1 kesme derecesi kazandırmıştır.
- Son olarak $C_8 = \{\{9\}, \{1,2,3,4,5,6,7,8\}\}$ olacak şekilde ilgili düğümleri iki gruba bölerek düğüm 9'a 2 kesme derecesi, düğüm 6 ve 8'e 1 kesme derecesi kazandırmıştır. C_1 den C_8 e kadar tüm kesmeler için kesme dereceleri hesaplanmış ve Çizelge 4.6 da verilmiştir.

Çizelge 4.6. Tüm temel kesme düğüm dereceleri

Düğüm/ Kesme Dereceleri	1	2	3	4	5	6	7	8	9
C_1	2	1	-	-	-	1	-	-	-
C_2	1	4	1	-	-	1	1	-	-
C_3	-	1	3	-	2	1	2	1	-
C_4	-	-	1	2	1	-	-	-	-
C_5	-	-	1	1	4	-	1	1	-
C_6	-	1	1	-	1	1	4	-	-
C_7	-	-	-	-	1	1	-	3	1
C_8	-	-	-	-	-	1	-	1	2
C_1-C_8	3	7	7	3	9	6	8	6	3

4.4 Dügüm Baskınlık Deęeri(Karcı Centrality Value)

Karcı merkezlilik algoritması düęüm baskınlık deęeri (Γ) simgesi ile ifade edilmektedir. Örnek çizgeye ait düęüm baskınlık deęeri (Γ) Çizelge 4.7’de verilmiştir. Formül 4.1’de düęüm baskınlık deęerinin formülü verilmiştir.

$$\text{Dügüm Baskınlık Deęeri } (\Gamma) = \text{Çizge Derecesi} + \text{Kmax Derecesi} + \text{Kesme Derecesi} \quad (4.1)$$

Çizelge 4.7. Dügüm baskınlık deęeri (Γ)

Dügüm Adı	$\Gamma =$	Çizge derece	Kmax derece	Kesme derece
1	6	2	1	2
2	12	4	1	7
3	15	5	3	7
4	6	2	1	3
5	14	4	1	9
6	18	6	6	6
7	13	4	1	8
8	10	3	1	6
9	6	2	1	3

Dügümlerin baskınlık deęerlerinin belirlenmesi ile hakim kümenin tespit edilmesi için bir sonraki sürece geçilmiştir.

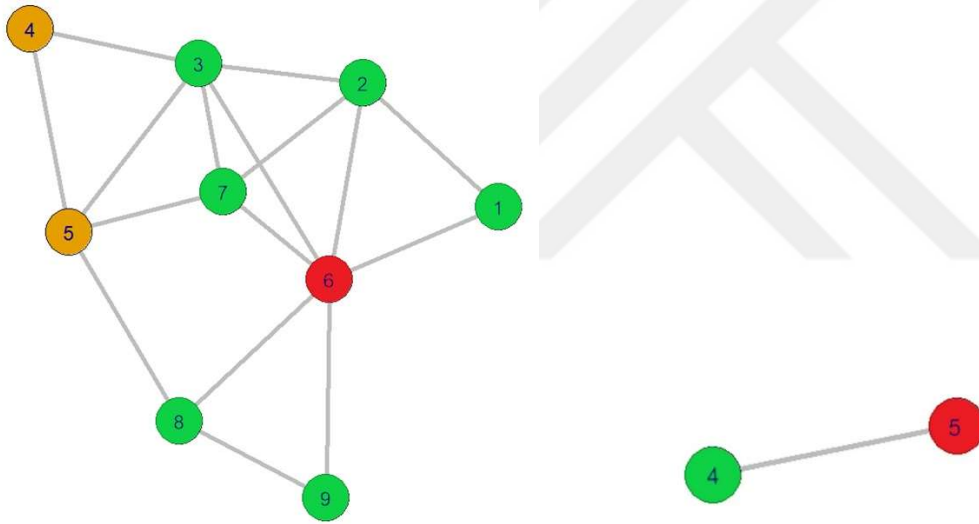
4.5 Hakim Küme (Dominating Set) Üyelerinin Tespiti

Bu aşamada ilk önce çizge üzerinde pendant düęüm kontrolü yapılır. Dügüm derecesi 1 olan düęümlere Pendant düęüm denir [63]. Eđer çizgede pendant düęüm var ise, pendant düęümün atası (komşu düęüm) baskın düęüm olarak hakim kümenin üyeleri arasına eklenir. Baskın düęümün belirlenmesinden sonra baskın düęüm ve baskın düęümün komşuları tüm kenar bağlantıları ile çizgeden ve Kmax ağacından silinir.

İkinci aşamada eđer çizgede pendant düęüm yok ise, güncel çizgenin düęüm baskınlık deęerleri hesaplanır. Dügüm baskınlık deęeri en yüksek olan düęüm baskın düęüm olarak seçilerek hakim küme üyeleri arasına eklenir. Dügüm baskınlık deęeri eşit olan birden fazla düęüm olması durumunda Kmax derecesi düşük olan düęüm seçilir. Eşitlik hala devam ediyorsa herhangi bir düęüm baskın düęüm olarak seçilir. Seçilen baskın düęüm ve komşuları tüm kenar bağlantıları ile silinerek çizge ve Kmax ağacından çıkarılır. Çizge ve Kmax ağacı güncellenir.

Her iterasyonun başlangıcında pendant düğüm kontrolü yapılır. Eğer herhangi bir iterasyonda pendant düğüm var ise ilk aşama uygulanır, pendant düğüm yok ise ikinci aşama uygulanır. İşlemler çizgedeki tüm düğümler bitene kadar devam eder. Örnek çizge üzerinde açıklayacak olursak, çizgede pendant düğüm olmadığı için ikinci aşama uygulanmıştır. Çizge üzerinde düğüm baskınlık değeri en yüksek olan düğüm 6 dır. Düğüm 6 hakim kümenin ilk baskın düğümü olarak seçilerek, gerekli güncelleme işlemleri yapılır. Şekil 4.4'ün a bölümünde hakim küme algoritmasının birinci iterasyonuna ait görsel verilmiştir. Şekilden de anlaşılacağı üzere düğüm 6 dan, düğüm 1,2,3,7,8,9'a erişim sağlanmaktadır.

Şekil 4.4'ün a bölümünde verilen görselde kırmızı renkli düğümler hakim kümenin üyeleri olan düğümleri, yeşil düğümler baskın düğümler tarafından erişilen düğümleri, turuncu düğümler ise henüz erişilmemiş olan düğümleri ifade etmektedir. Baskın düğüm seçimi sonrası yapılan güncelleme neticesinde Şekil 4.4 (b)'deki görsel çizge elde edilmiştir.



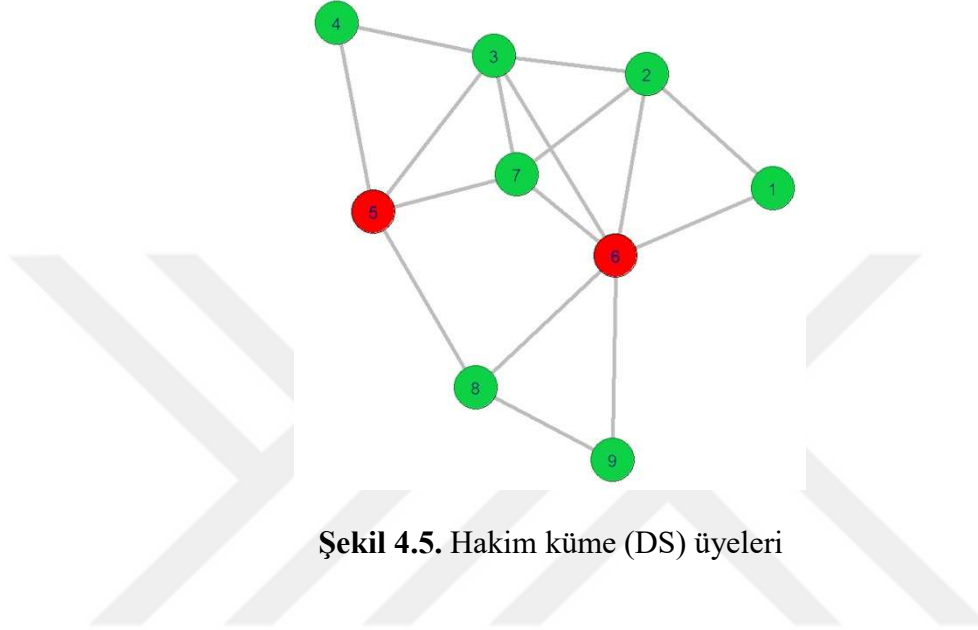
(a) Hakim küme algoritması 1.iterasyon

(b) Hakim küme algoritması 2.iterasyon

Şekil 4.4. Hakim küme iterasyonları

Düğüm 4 ile düğüm 5 ulaşılmayan düğümler olarak tek kaldığı için özel bir durum oluşmuştur ve pendant özelliği uygulanmaz, Düğüm baskınlık değerleri hesaplanarak değeri yüksek olan baskın düğüm olarak seçilir. Yapılan hesaplama neticesinde her iki düğümün Kmax ağacı üzerinde bağlantısı bulunmamaktadır. Bu sebepten dolayı bu iki düğümlü çizge üzerinde kesme işlemi uygulanamaz ve Kmax dereceleri ile temel kesme dereceleri hesaplanamaz. Geriye kalan düğüm dereceleri incelendiğinde iki düğümün derecesi 1 olarak tespit edilmiştir.

Bu eşitlikten dolayı iki düğümden herhangi birisi hakim kümenin 2.baskın düğümün olarak seçilir. Düğüm 5'in hakim kümenin 2. baskın düğümü olarak seçildiği düşünülürse düğüm 6 ve düğüm 5 üzerinden çizgenin bütün düğümlerine doğrudan ulaşılmaktadır. Örnek çizgede minimum hakim küme 2 den daha az sayıda düğüm ile oluşturulamayacağı için bu örnek için geliştirilen algoritma minimum hakim küme'ye sağlamaktadır. Şekil 4.5'de hakim küme üyeleri kırmızı ile, eriştikleri komşu düğümler ise yeşil düğüm ile gösterilmiştir.



Şekil 4.5. Hakim küme (DS) üyeleri

4.6 Sözde Kodlar

Önerilen algoritmanın tüm aşamalarını içeren sözde kodları aşağıdaki bölümlerde ayrı ayrı açıklanarak verilmiştir. Verilen çizgenin hakim küme üyelerini belirlemek için ilk aşamada Kmax ağacının oluşturulması gerekmektedir. Algoritma 4.1'de Kmax ağacının sözde kodu verilmiştir. Bu kod bloğunda Kmax ağacı ile komşu düğümlerin kuyruğa eklenme işlemleri ve ilgili düğümlerin derece azaltmaları gerçekleştirilir. Bu bölümdeki tüm algoritmaların zaman karmaşıklığı değeri algoritmanın başlık bilgisinin verildiği bölüme eklenmiştir.

DeQueue_Max(Algoritma 4.2) sayesinde en yüksek dereceleri düğümlerden biri Kmax ağacında açılacak sonraki düğüm olarak seçilir ve seçilen düğüm kuyruktan direk silinir. Kuyrukta birden fazla maksimum düğüm olması durumunda, önce ağaçtaki seviyesine sonra kuyruktaki sıralamasına bakılır. Kmax ağacı Algoritma 4.1 ve 4.2 kullanılarak oluşturulur. Algoritma 4.2'nin 5.satırında [...] fonksiyonu, verilen koşul yerine getirildiğinde 1, sağlanmadığında ise sonucu 0 olan bir toplam fonksiyonudur.

Algoritma 4.1: Kmax ağacının oluşturulması – $O(n^3)$

Kmax_Tree(A,AT,D)

1. $Q \leftarrow \emptyset$
2. $r \leftarrow \max(D)$ // D derece matrisidir
3. $T=(V,E1)$ and $E1 \leftarrow \emptyset$
4. $i \leftarrow 1, \dots, |V|$
5. $j \leftarrow 1, \dots, |V|$
6. $AT(i,j) \leftarrow 0$
7. EnQueue(Q,r)
8. $Q2 \leftarrow \emptyset$
9. for $i \leftarrow 1, \dots, |N(r)|$
10. EnQueue(Q2, v_i), $v_i \in N(r)$
11. if $AT(r,s)=0$ then
12. $AT(r,s)=1, AT(s,r)=1$
13. EnQueue(Q2,s)
14. while $Q2 \neq \emptyset$
15. $v \leftarrow \text{DeQueue_Max}(Q2,A,AT)$
16. EnQueue(Q,v)
17. for $i \leftarrow 1, \dots, |N(v)|$
18. if not($v_i \in Q2$) and $v_i \in N(v)$
19. EnQueue(Q2, v_i)
20. $AT(v,v_i)=1, AT(v_i,v)=1$
21. Remove_Zero(Q2,A,AT)

Algoritma 4.2: Kalan derecesi maksimum olan düğümün seçimi - $O(n^2)$

DeQueue_Max(Q,A,AT)

1. $m \leftarrow 0$
2. node $\leftarrow \emptyset$
3. $i \leftarrow 1, \dots, |Q|$
4. $v \leftarrow \text{DeQueue}(Q)$
5. $t_1 \leftarrow \sum_{j=1}^n A(v,j), t_2 \leftarrow \sum_{j=1}^n [A(v,j) \text{ and } j \notin Q]$
6. if $m < |t_1 - t_2|$ then
7. $m = t$
8. node = v
9. EnQueue(Q,v)
10. else EnQueue(Q,v)
11. DeQueue(Q,node) // Q dan düğüm(node) kaldırılıyor.
12. return node

Kuyruktaki eleman sayısını gereksiz yere artırmamak için , kalan derecesi 0 olan düğümler kuyruktan silinirler. Bu işlemi gerçekleştiren algoritma Algoritma 4.3’de verilmiştir.

Algoritma 4.3: Kalan derecesi 0 olan düğümlerin kaldırılması - $O(n^2)$

RemoveZero(Q,A,AT)

1. $i \leftarrow 1, \dots, |Q|$
 2. $v \leftarrow \text{DeQueue}(Q)$
 3. $t_1 \leftarrow \sum_{j=1}^n A(v,j), t_2 \leftarrow \sum_{j=1}^n [A(v,j) \text{ and } j \notin Q]$
 4. if $t_1 - t_2 \neq 0$ then
 5. EnQueue(Q,v)
-

K_{max} dolanım ağacı oluşturulduktan sonra, K_{max} ağacı kullanılarak temel kesme kümeleri elde edilmektedir. Algoritma 4.4 bu amacı gerçekleştirmek için kullanılır.

Algoritma 4.4: Temel(Fundamental) kesmelerin oluşturulması - $O(n^3)$

FundamentalCutSets(AT,A,B,C)

1. $B \leftarrow \text{IncidenceMatrix}(A)$ // A'nın boyutu $n \times n$ dir
 2. for $i \leftarrow 1, \dots, n$
 3. for $j \leftarrow 1, \dots, m$ // B'nin boyutu $n \times m$ ve kenar sayısı m dir.
 4. $C(i,j) = 0$
 5. for $i \leftarrow 1, \dots, m_2$
 6. $\text{EnQueue}(E_2, e_i)$ // T'deki kenar sayısı m_2 dir ve $T = (V, E_2)$
 7. while $E_2 \neq \emptyset$
 8. $e = \text{DeQueue}(Q_2)$
 9. if $e = (u, v)$ and $(d_v = 1 \text{ or } d_u = 1)$ // d_u ve d_v , T'deki u ve v nin derecelerini gösterir
 10. if $d_u = 1$ then
 11. $i \leftarrow 1, \dots, m$ // m , $G = (V, E)$ 'deki kenar sayısıdır
 12. $C(u, i) = B(u, i)$
 13. else $\text{GenerateInternalCut}(AT, A, C, B, e)$
-

Temel kesme kümeleri elde edilirken, K_{max} ağacında yaprak düğümlerle denk gelen kenarlar temel kesme kümesini oluşturur. İkinci temel kesme türü, K_{max} ağacını iki alt ağaca bölen kesme tipidir.

İlk tür kesme(fundamental cut-set) doğrudan Algoritma 4.4 ile elde edilirken ikinci tür(internal cut-set) Algoritma 4.5 ile elde edilir.

Algoritma 4.5: Dâhili(Internal) kesmelerin oluşturulması - $O(n^3)$

GenerateInternalCut(AT,A,C,B,e)

1. Assume $e = (u, v)$
 2. $Q_2 \leftarrow \emptyset$
 3. $\text{EnQueue}(Q_2, u)$
 4. while $Q_2 \neq \emptyset$
 5. $u \leftarrow \text{DeQueue}(Q_2)$
 6. $\text{EnQueue}(Q, u)$
 7. for $i \leftarrow 1, \dots, n$ // $G = (V, E)$ 'deki düğüm sayısı
 8. if $AT(u, i) = 1$ and $i \neq v$
 9. $\text{EnQueue}(Q_2, i)$
 10. while $Q \neq \emptyset$
 11. $u \leftarrow \text{DeQueue}(Q)$
 12. for $i \leftarrow 1, \dots, n$ // $G = (V, E)$ 'deki düğüm sayısı n dir
 13. if $A(u, i) = 1$ and $i \notin Q$
 14. $C(u, k) = B(u, k)$ // k , $e = (u, i)$ kenarını gösterir
-

Algoritma 4.6 (Γ) simgesi ile ifade edilen Karcı merkezlilik algoritmasına ait düğüm basınlık değerlerinin belirlenmesi için kullanılır.

Algoritma 4.6: Her düğüm için baskın düğüm derecesinin hesaplanması - $O(n^3)$

- 1) $G=(V,E)$ ve $T=Kmax_Tree(A,AT,D)$ varsayılırsa $T=(V,E_1)$, G çizgesinin bir $Kmax$ ağacıdır
 - 2) B incidence matrisidir, G ve C_{max} , $Kmax$ ağacına karşılık gelir.
 - 3) $E_{max} = B * C_{max}^T$ // E_{max} baskınlığa karşılık gelir ve C_{max}^T , C_{max} 'in transpozudur.
 - 4) $i \leftarrow 1, \dots, n$
 - 5) $\pi(v_i) = 0$
 - 6) $j \leftarrow 1, \dots, m$
 - 7) $\pi(v_i) = \pi(v_i) + E_{max}(i, j)$
 - 8) $i \leftarrow 1, \dots, n$
 - 9) $\pi(v_i) = \pi(v_i) + d_G(v_i) + d_T(v_i)$ // $d_G(v_i)$, v_i 'nin G deki düğüm derecesidir, $d_T(v_i)$ $Kmax$ düğüm derecesidir.
i.e. $\pi(v_i) = \text{Cut-Set Effectiveness} + \text{Graph Effectiveness} + \text{Kmax_Tree Effectiveness}$.
-

Algoritma 4.7 çizge üzerindeki pendant düğümlerin belirlenmesi için kullanılır.

Algoritma 4.7: Pendant düğüm - $O(n^2)$

Pendant(G,A,DS,D)

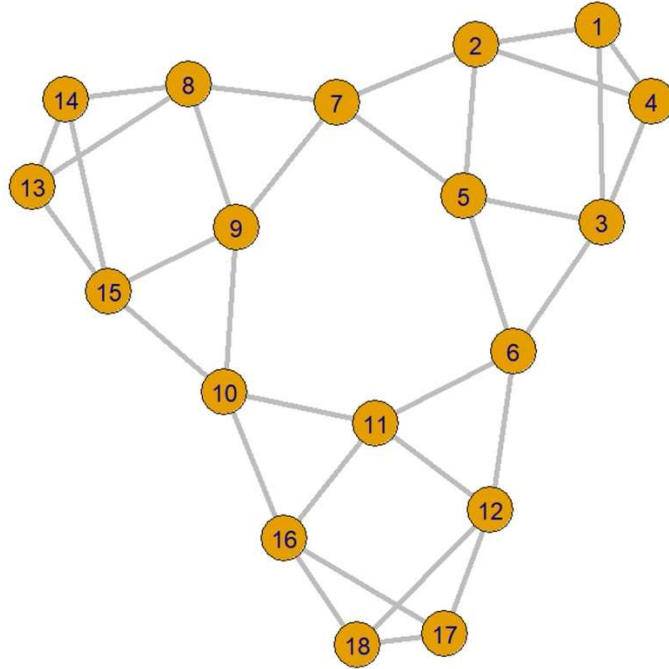
1. $i \leftarrow 1, \dots, |V|$
 2. if $D(v_i)=1$ then
 3. $DS = DS \cup \{v_j\}$ where $N(v_i) = \{v_j\}$
 4. $k \leftarrow 1, \dots, |V|$
 5. if $A(v_i, v_k)=1$ then
 6. $A(v_i, v_k)=0, A(v_k, v_i)=0$
-

5. UYGULAMA SONUÇLARI

Tezin önceki bölümlerinde hem baskın düğüm hemde hakim küme algoritmalarının çalışma prensipleri ayrıntılı olarak örneklendirilerek anlatılmıştır. Bu bölümde bazı özel çizgeler üzerinde bu algoritmaların nasıl sonuçlar vereceği özet bir şekilde aşama aşama anlatılmıştır.

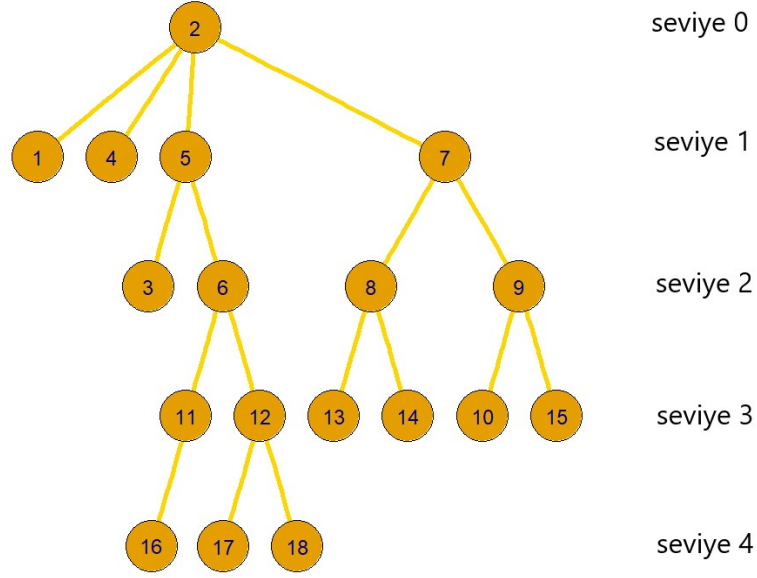
5.1 Özel Çizgelerde Hakim Küme Tespiti

Bu bölümde 18 düğüm ve 33 kenar bağlantısına sahip pençesiz (Claw free) çizge üzerinde algoritmalar aşama aşama uygulanmıştır. Şekil 5.1’de verilen örnek çizge çalışma içerisinde A çizgesi olarak isimlendirilmiştir.



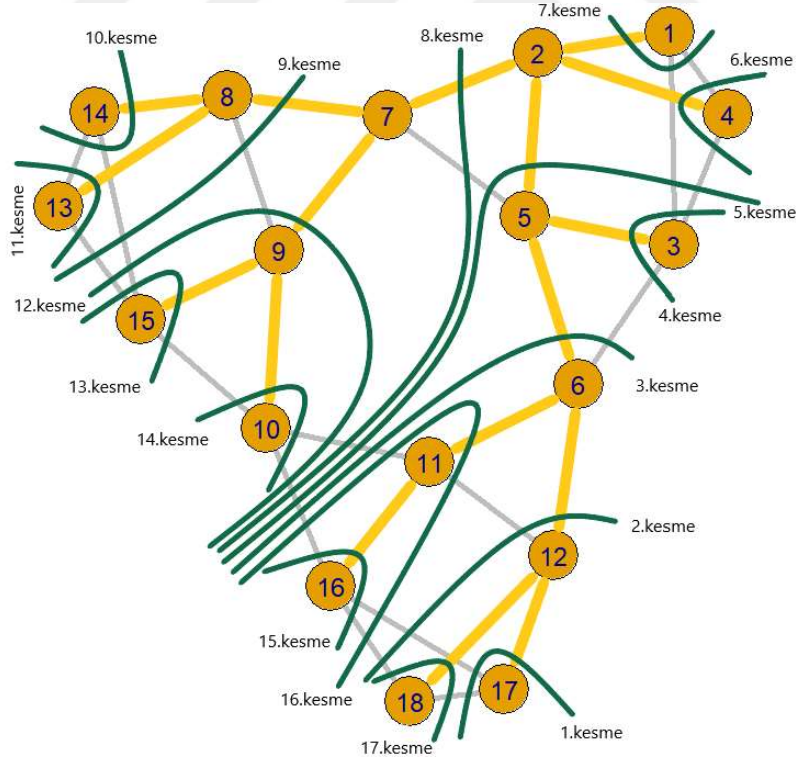
Şekil 5.1. A çizgesi

A çizgesi incelendiğinde pendant düğüm olmadığı görülmektedir. Bu yüzden hakim kümenin ilk üyesinin belirlenmesi için baskın düğüm değerlerinin hesaplanması gerekmektedir. A çizgesinin Kmax ağacı Şekil 5.2’de verilmiştir.



Şekil 5.2. A çizgesine ait Kmax ağacı

Tespit edilen Kmax ağacı neticesinde gerçekleştirilecek bütün temel kesmeler Şekil 5.3’de verilmiştir.



Şekil 5.3. A çizgesine ait tüm temel kesmeler

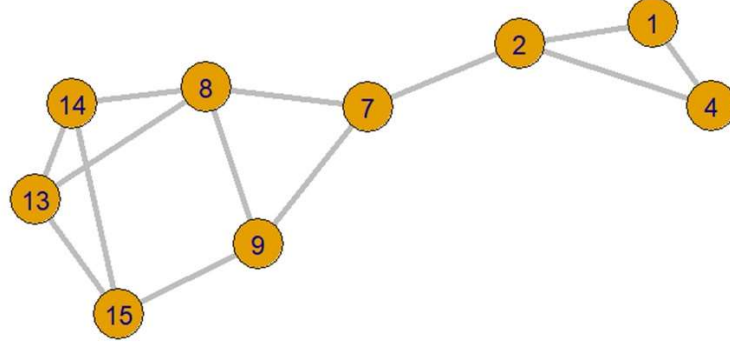
Verilen A çizgesinin düğüm dereceleri, Kmax düğüm dereceleri, düğümlerin kesme dereceleri ve düğüm baskınlık değerleri Çizelge 5.1’ de verilmiştir.

Çizelge 5.1. A çizgesi düğüm baskınlık değerleri (Γ)

Düğüm Adı	$\Gamma =$	Çizge derece +	Kmax derece +	Kesme derece
1	10	3	1	6
2	12	4	4	4
3	14	4	1	9
4	10	3	1	6
5	12	4	3	5
6	12	4	3	5
7	12	4	3	5
8	12	4	3	5
9	12	4	3	5
10	21	4	1	16
11	16	4	2	10
12	12	4	3	5
13	11	3	1	7
14	11	3	1	7
15	16	4	1	11
16	21	4	1	16
17	11	3	1	7
18	11	3	1	7

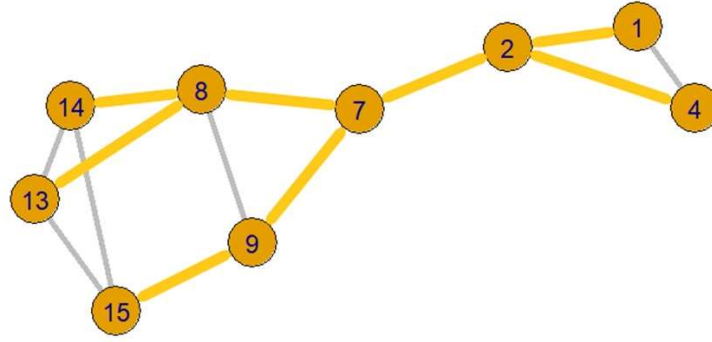
A çizgesine ait düğüm baskınlık değerlerinin hesaplanması ile birlikte hakim küme üyelerinin belirlenmesi gerçekleştirilir. Çizelge 5.1'deki değerler incelendiğinde düğüm 16 ve düğüm 10 'un düğüm baskınlık değerinin 21 olduğu görülmektedir. İlk iterasyon seçimi için baskınlık değeri eşit olan bu iki düğümden Kmax derecesi küçük olan düğüm seçilir. Kmax derecesi bu iki düğüm için eşit olduğundan, düğümlerden herhangi bir tanesi hakim kümenin üyesi olarak seçilebilir. Bu süreçte düğüm 16 hakim kümenin ilk üyesi olarak seçilmiştir. Şekil 5.4'de gösterildiği üzere hakim kümenin üyeleri kırmızı renk ile gösterilirken, komşu düğümler ise yeşil renk ile ifade edilmiştir.

Üçüncü iterasyon için çizgenin güncel hâli Şekil 5.6'da verilmiştir. Şekilde görüldüğü üzere pendant düğüm bulunmamaktadır. Bu nedenle Kmax ağacı Şekil 5.6'da verilen güncel çizgeye göre yeniden oluşturularak baskın düğüm dereceleri hesaplanır.



Şekil 5.6. 3. iterasyon için çizgenin güncel hâli

Şekil 5.7'de güncel çizge üzerinde Kmax ağacı sarı çizgiler ile gösterilmiştir.



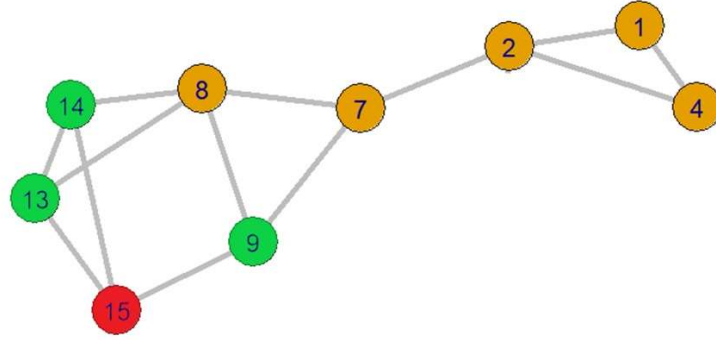
Şekil 5.7. Hakim küme 3. iterasyon

Hesaplamalar neticesinde Şekil 5.6'da verilen çizgenin düğüm baskınlık değerleri Çizelge 5.2'de açıklanmıştır. Çizelgede düğüm 15'in baskınlık değeri en yüksek düğüm olduğu görülmektedir.

Çizelge 5.2. A çizgesi 3.iterasyon düğüm baskınlık değerleri (Γ)

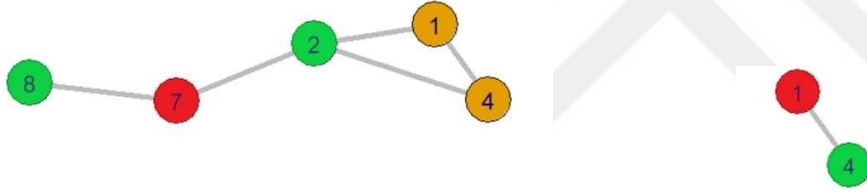
Düğüm Adı	$\Gamma =$	Çizge derece +	Kmax derece +	Kesme derece
1	6	2	1	3
2	9	3	3	3
4	6	2	1	3
7	9	3	3	3
8	12	4	3	5
9	9	3	2	4
13	11	3	1	7
14	11	3	1	7
15	13	3	1	9

Şekil 5.8’de görüldüğü üzere hakim kümenin 3.üyesi olarak düğüm 15 seçilmiştir. Seçim işlemi sonrasında çizge ve Kmax ağacı güncellenmiştir.



Şekil 5.8. A çizgesi hakim küme (DS) algoritması 3. iterasyon

Pendant düğüm kontrolü sonrası 4. iterasyon’ a başlanır. Şekil 5.9’un a kısmında belirtildiği üzere çizgede pendant düğüm bulunmaktadır. Pendant düğümün atası konumundaki düğüm 7 hakim kümenin 4. üyesi olarak seçilir. Çizge ve Kmax ağacı güncellenerek 5. iterasyona geçilir.

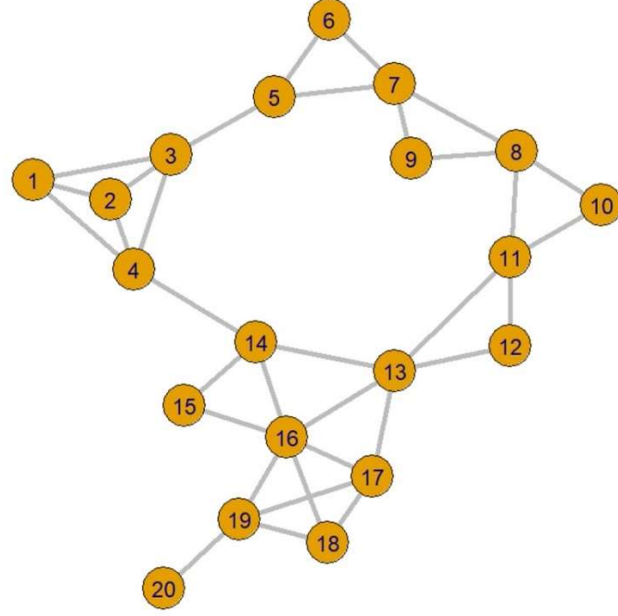


(a) A çizgesi DS algoritması 4. iterasyon (b) A çizgesi DS algoritması 5. iterasyon

Şekil 5.9. Hakim küme iterasyonları

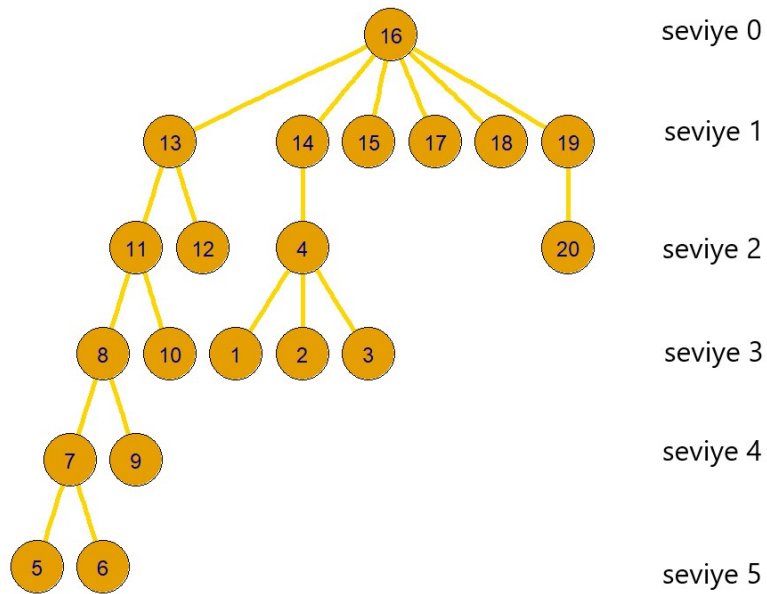
Şekil 5.9’un b kısmında güncel çizgede 2 düğüm kalmıştır. Bu koşullar altında pendant düğüm kontrolü yapılamaz. Yeni hakim küme üyelerinin belirlenmesi için baskın düğüm derecelerine bakılır. Her iki düğümünde baskın düğüm dereceleri eşit olduğu için düğüm seçiminde öncelik sunmamaktadır. Bu koşullarda düğüm 1 ve düğüm 4 ‘den herhangi bir tanesininin seçilmesi yeterli olacaktır. Bu çalışmada düğüm 1 hakim kümenin 5. üyesi olarak seçilmiştir. Sonuç olarak hakim kümenin baskın düğümleri sırası ile düğüm 16,6,15,7 ve 1 olarak belirlenmiştir. Eğer 1. iterasyonda hakim küme üyesi olarak düğüm 16 yerine düğüm 10 seçilseydi hakim kümenin üyeleri sırasıyla düğüm 10,3,7,13 ve 17 olarak tespit olacaktır. Her iki seçimde de hakim küme üye sayısı 5 olmaktadır. Aynı zamanda bu çizgede hakimiyet, 5 düğümden daha az sayıda düğüm ile sağlanamayacağı için belirlenen düğümler minimum hakim kümeyi sağlamaktadır.

Algoritmanın farklı çizge türlerindeki başarısını göstermek için örnek çalışmalar yapılmıştır. Şekil 5.10'da karmaşık bir örnek çizge verilmiştir. Bu çizge B çizgesi olarak isimlendirilmiştir. B çizgesi 20 düğüm ve 33 kenar bağlantısından oluşmaktadır.



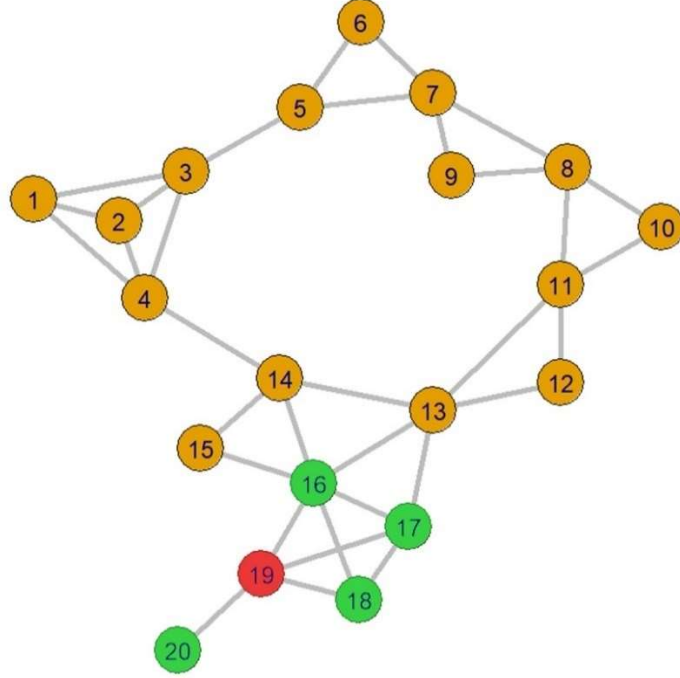
Şekil 5.10. B çizgesi

Şekil 5.10'da ki çizge incelendiğinde düğüm 20'nin pendant düğüm olduğu görülmektedir. Bu nedenle 1. iterasyonda düğüm baskınlık değerlerinin hesaplanmasına gerek yoktur. Şekil 5.11'de B çizgesine ait Kmax ağacı ilk iterasyonda kullanılmayacak olsada bilgilendirme amaçlı verilmiştir.



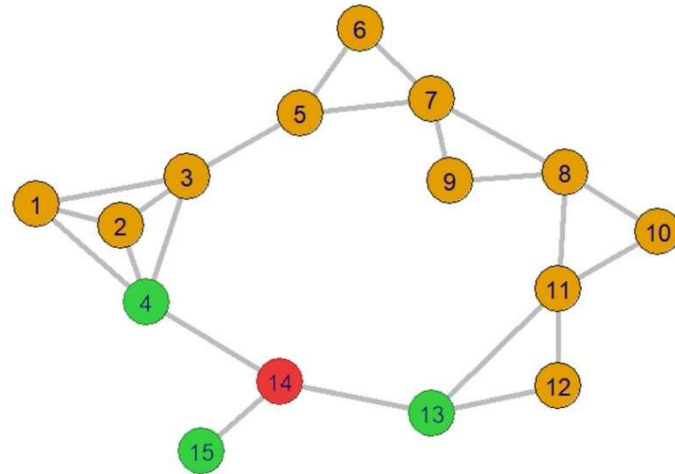
Şekil 5.11. B çizgesine ait Kmax ağacı

Şekil 5.12’de verildiği üzere düğüm 20’nin atası konumunda bulunan düğüm 19, hakim kümenin ilk üyesi olarak seçilir ve çizge ile Kmax ağacı üzerinde gerekli güncelleme işlemleri yapılır.



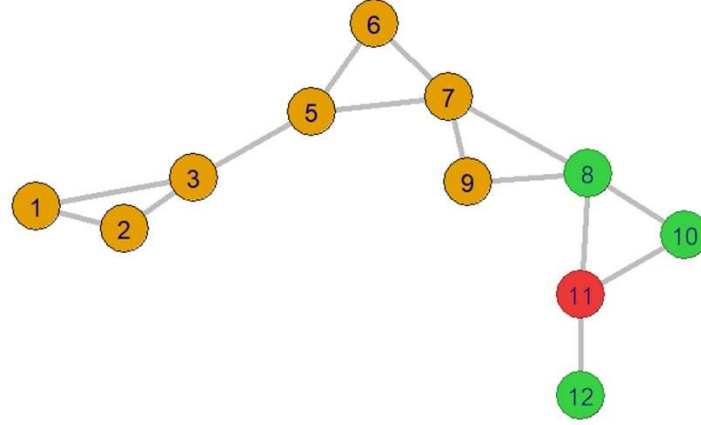
Şekil 5.12. B çizgesi DS algoritması 1. iterasyon

Güncellenen B çizgesi Şekil 5.13’de verilmiştir. Çizge incelendiğinde düğüm 15’in pendant düğüm olduğu tespit edilmiştir. Pendant düğümün atası olan düğüm 14, hakim kümenin ikinci baskın düğümü olarak seçilir. Çizge ve Kmax ağacı bir sonraki iterasyon için güncellenir.



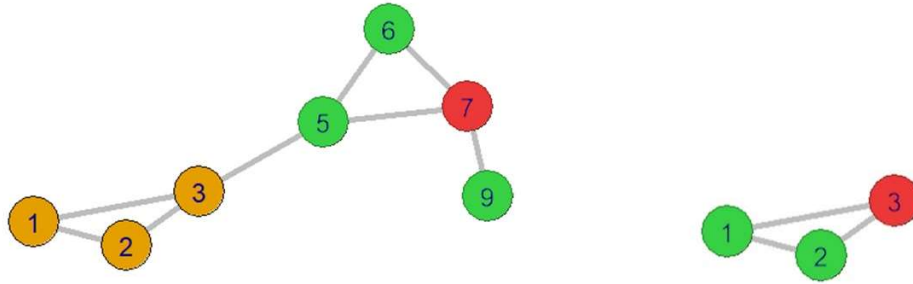
Şekil 5.13. B çizgesi DS algoritması 2. iterasyon

Şekil 5.14 incelendiğinde düğüm 12'nin pendant düğüm olduğu görülmektedir. Yine pendant düğüm koşulu yerine getirilir ve düğüm 12'nin atası konumundaki düğüm 11, hakim kümenin 3. baskın düğümü olarak belirlenir. Güncelleme işlemleri tamamlandıktan sonra sonraki iterasyona geçilir.



Şekil 5.14. B çizgesi DS algoritması 3. iterasyon

Şekil 5.15'in (a) bölümünde görüldüğü üzere 4. iterasyonda düğüm 9 pendant düğüm olarak seçilir. Düğüm 9'un atası, 4. baskın düğüm olarak hakim küme üyeleri arasına eklenir. Gerekli güncelleme işlemleri yapılır.



(a) B çizgesi DS algoritması 4. iterasyon (b) B çizgesi DS algoritması 5. iterasyon

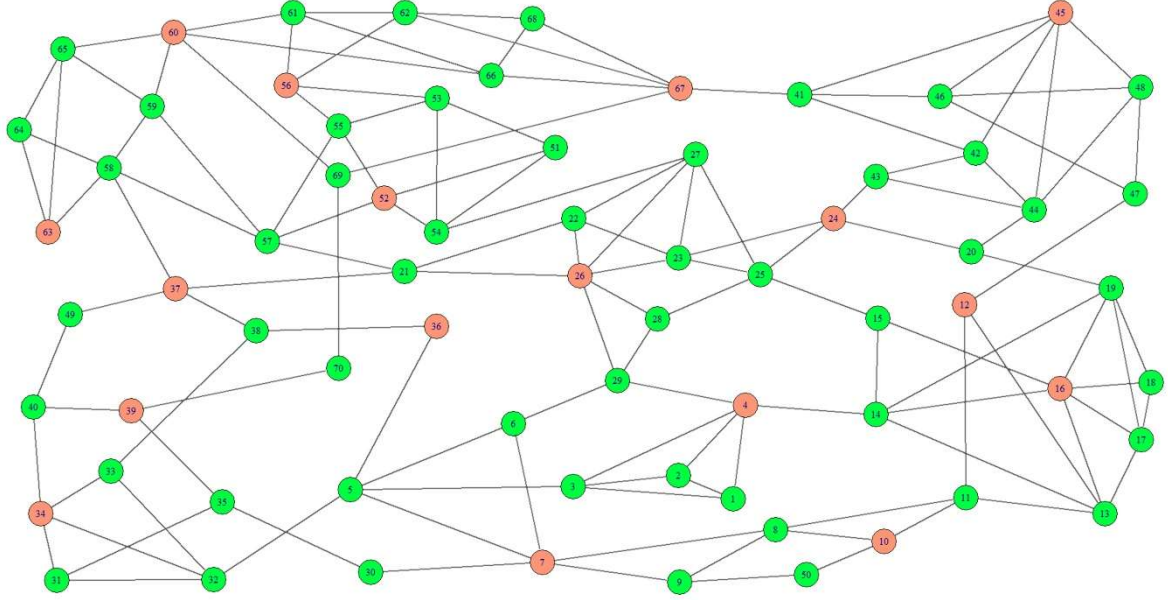
Şekil 5.15. B çizgesi DS iterasyonlar

Şekil 15'in (b) bölümünde pendant düğüm bulunmamaktadır. Yeni hakim küme üyesinin belirlenmesi için baskın düğüm değerlerinin hesaplanması gerekmektedir.

Şekil 15 (b) incelendiğinde tüm düğümlerin dereceleri 2'dir. Kmax ağacı dereceleri 0 olduğu için kesme dereceleri de hesaplanamamaktadır ve 0 olarak ifade edilir. Sonuç olarak düğümlerin baskınlık değerleri her düğüm için 2 olarak hesaplanmıştır. Bu eşitlikten dolayı herhangi bir düğüm hakim küme üyesi olarak seçilebilir. Bu çalışma için düğüm 3 hakim kümenin 5. baskın düğümü olarak seçilir. 5. iterasyon itibari tüm düğümlere ulaşıldığı için algoritmanın çalışması tamamlanmıştır. B çizgesi 5 düğümden daha az sayıda düğüm ile hakimiyet altına alınamayacağı için hakim küme üyeleri olan düğüm 19, 14, 11, 7, 3 aynı zamanda minimum hakim kümeyi ifade etmektedir.

5.2 Karma Çizgelerde Hakim Küme Tespiti

Önerilen algoritma 70 düğümlü daha büyük bir çizge üzerinde uygulanarak başarısı test edilmiştir. Şekil 5.16'da verilen çizge C çizgesi olarak isimlendirilmiştir. C çizgesi grid, lineer, tree, cycle gibi özel çizgelerin birleşiminden oluşturulan karmaşık bir çizgedir. Şekil 5.16'da gösterilen kırmızı düğümler C çizgesi üzerinde önerilen algoritmanın belirlediği hakim kümenin üyelerini ifade ederken, yeşil düğümler ise hakim küme üyelerinin komşu düğümlerini ifade etmektedir.



Şekil 5.16. C çizgesinin baskın ve komşu düğümleri

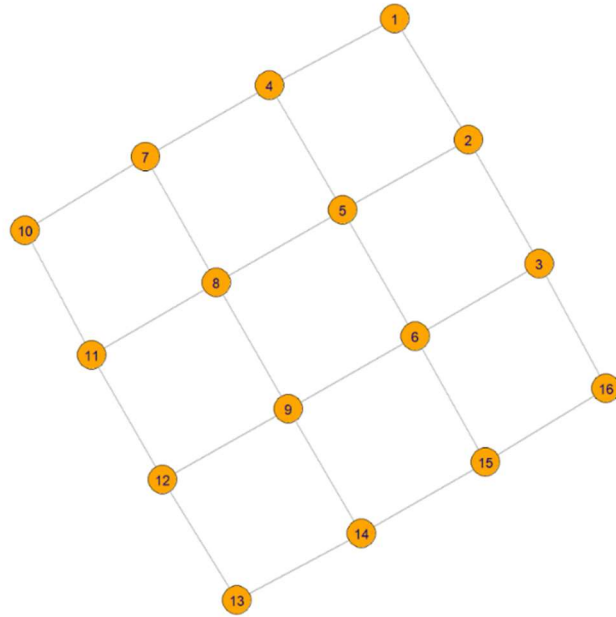
Çizelge 5.3'de algoritmanın C çizgesi üzerinde her iterasyonda hangi üyenin ne şekilde tespit edildiğine dair bilgiler verilmiştir.

Çizelge 5.3. C çizgesi hakim küme üye tespit yöntemi

İterasyon	Hakim Küme Üyeleri	Tespit Yöntemi
1	60	Baskınlık değeri
2	39	Pendant atası
3	37	Pendant atası
4	7	Pendant atası
5	63	Baskınlık değeri
6	36	Tek düğüm
7	10	Pendant atası
8	56	Baskınlık değeri
9	52	Pendant atası
10	67	Pendant atası
11	45	Baskınlık değeri
12	12	Pendant atası
13	24	Pendant atası
14	4	Baskınlık değeri
15	16	Pendant atası
16	26	Pendant atası
17	34	Baskınlık değeri

5.3 Izgara Çizgelerde Uygulanması

Önerilen hakim küme algoritmasının uygulandığı bir diğer özel çizge türü ızgara (grid) çizgelerdir. Şekil 5.17’de verildiği üzere 16 düğüm ve 24 kenar bağıntısından oluşan ızgara çizgeye ait hakim küme üyeleri önerilen algoritma ile tespit edilmiştir. Izgara çizgi üzerinde pendant düğümün olmadığından ilk aşamada Γ değerleri hesaplanmıştır.



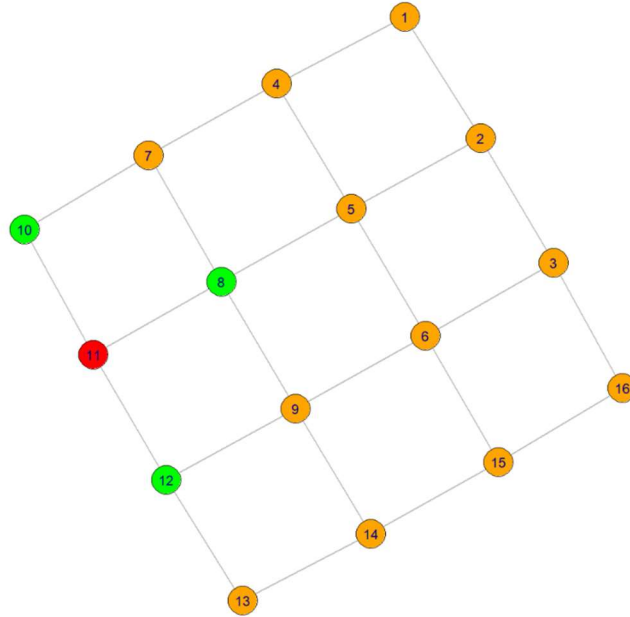
Şekil 5.17. Izgara çizge

Çizelge 5.4’de ızgara (Grid) çizgeye ait baskınlık değerleri (Γ) verilmiştir. Düğüm 8 ve düğüm 11 en yüksek düğüm baskınlık değerine sahiptir. 2 düğümün aynı baskınlık değerine sahip olması neticesinde iki düğümden Kmax ağaç derecesi küçük olan düğüm seçilir.

Çizelge 5.4. Izgara (Grid) çizgeye ait baskınlık değerleri (Γ)

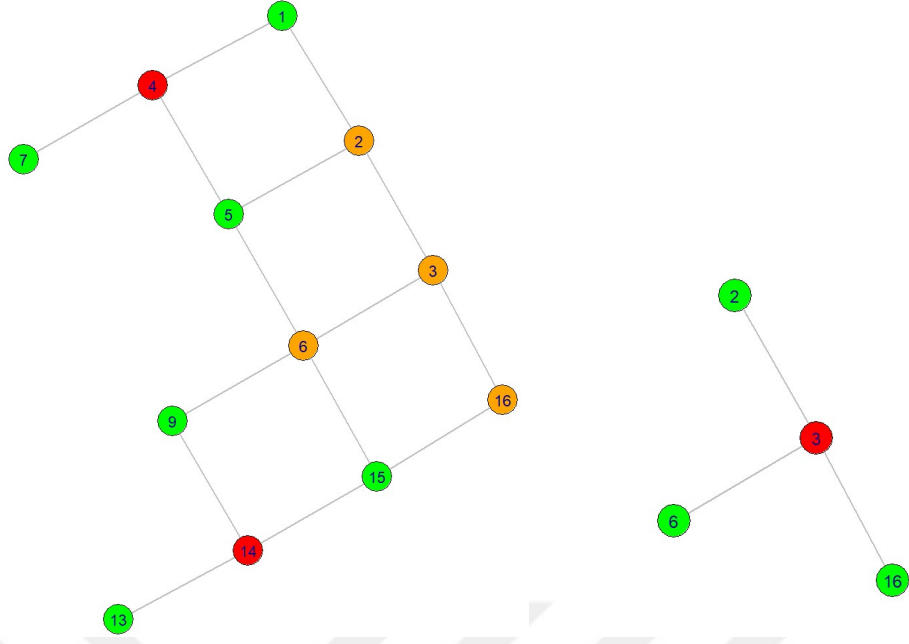
Düğüm No	Γ
1	7
2	11
3	10
4	9
5	12
6	12
7	10
8	17
9	13
10	11
11	17
12	9
13	7
14	11
15	11
16	7

Bu durumda düğüm 11 hakim kümenin ilk üyesi olarak seçilir. Şekil 5.18’de düğüm 11 ve komşu düğümler vurgulanmıştır. Bu düğümler, bağlantıları ile çizgeden çıkarılarak güncelleme işlemleri gerçekleştirilir.



Şekil 5.18. Izgara çizge DS algoritması 1. iterasyon

Şekil 5.19 (a) da belirtildiği üzere güncelleme sonrası 2 adet pendant düğüm bulunmaktadır.

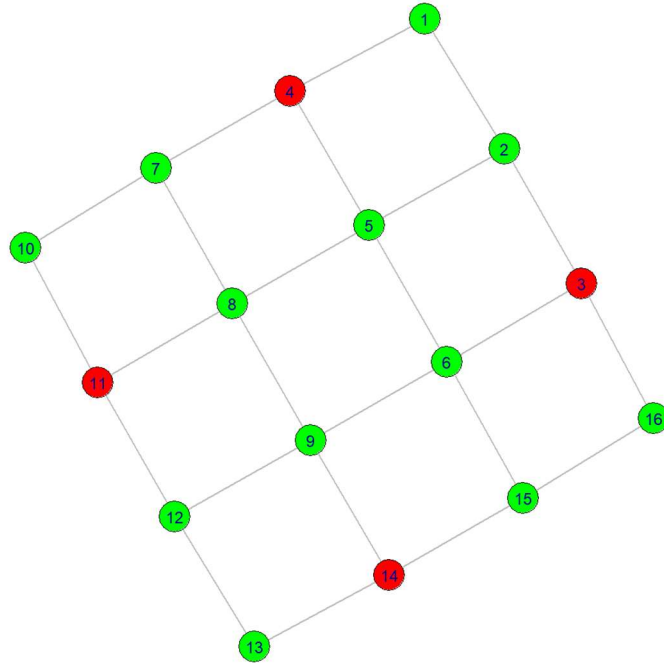


(a) Izgara çizge 2. iterasyon

(b) Izgara çizge 3. iterasyon

Şekil 5.19. Izgara çizge hakim küme iterasyonları

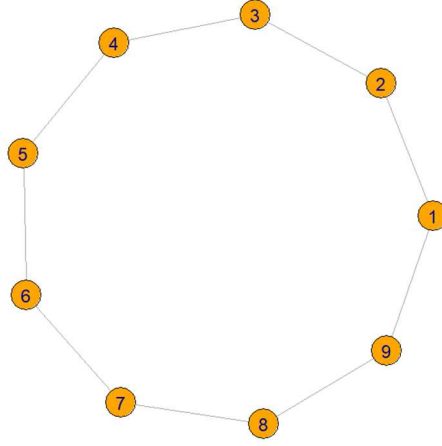
Pendant düğümlerin ataları konumunda bulunan düğüm 4 ve düğüm 14 hakim kümenin diğer üyeleri olarak belirlenir. Güncelleme işlemi sonrasında çizge Şekil 5.19 (b) deki durumdadır. Bu aşamada 3 pendant düğümün atası konumunda bulunan düğüm 3 hakim kümenin son üyesi olarak seçilir. Şekil 5.20’de gösterildiği gibi ızgara çizgeye ait hakim küme üyeleri düğüm 3, 4 ,11 ve 14 olarak tespit edilmiştir.



Şekil 5.20. Izgara çizge hakim küme üyeleri

5.4 Dairesel Çizgelerde Uygulanması

Dairesel çizgelerdeki başarısını göstermek için Şekil 5.21 deki 9 düğüm ve 9 kenar bağıntısından oluşan çizge üzerinde önerilen algoritma uygulanmıştır.



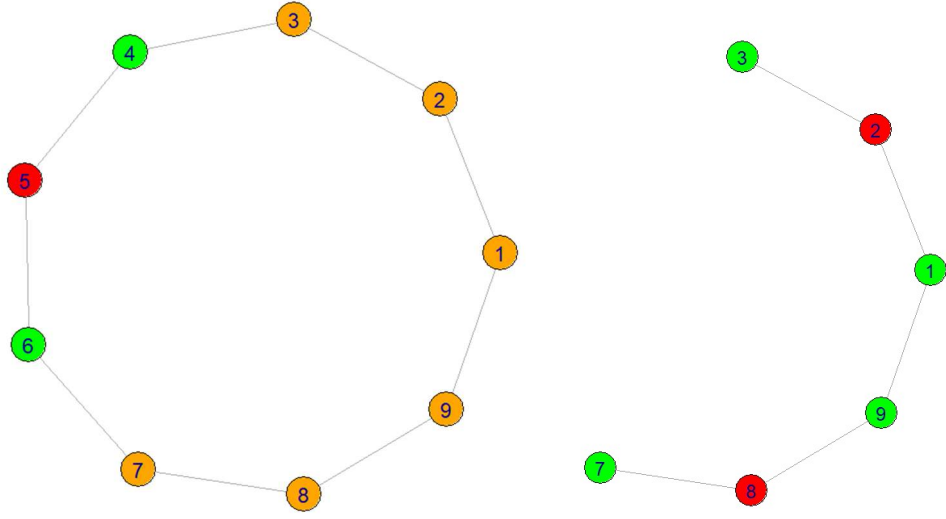
Şekil 5.21. Dairesel çizge

Çizgede pendant düğüm olmadığı için Karcı baskınlık değeri hesaplanır. Çizelge 5.5’de düğümlerin baskınlık değerleri verilmiştir. Bu sonuçlar doğrultusunda düğüm 5 ve düğüm 6 en yüksek baskınlık değerlerine sahiptir. Eşitlik olduğu için Kmax tree düğüm derecelerine bakılır. Kmax tree düğüm dereceleride eşit olduğundan düğümlerden herhangi birisi hakim kümenin ilk üyesi olarak seçilebilir.

Çizelge 5.5. Dairesel (Circle) çizgeye ait baskınlık değerleri (Γ)

Düğüm No	Γ
1	6
2	6
3	6
4	6
5	12
6	12
7	6
8	6
9	6

Şekil 5.22 (a) da belirtildiği gibi düğüm 5’in hakim küme üyesi seçilmesi durumunda düğüm 4 ve düğüm 5 bağlantıları ile çizgeden çıkarılır. Çizge güncellenir ve sonraki iterasyona geçilir. Şekil 5.22 (b) deki güncel çizgede düğüm 3 ve düğüm 7 pendant düğümlerdir. Bu düğümlerin ata düğümleri olan düğüm 2 ve düğüm 8 hakim kümenin diğer üyeleri olurlar.

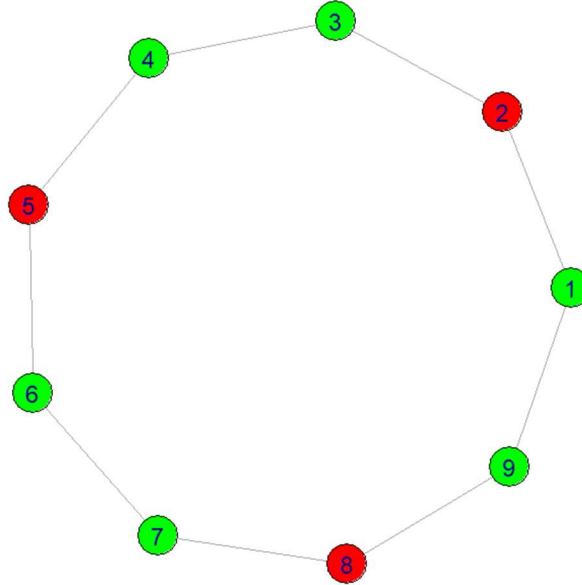


(a) Dairesel çizge 1. iterasyon

(b) Dairesel çizge 2. iterasyon

Şekil 5.22. Dairesel çizge hakim küme iterasyonları

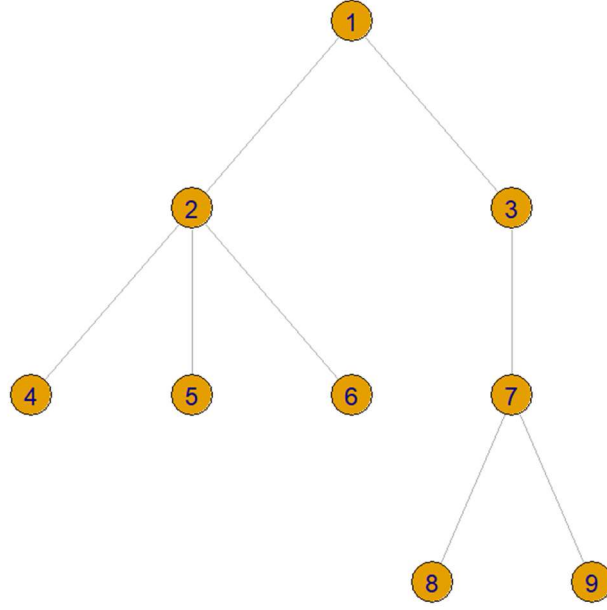
Şekil 5.23’de dairesel çizgeye ait hakim küme üyeleri düğüm 2, 5 ve 8 olarak tespit edilmiştir.



Şekil 5.23. Dairesel çizge hakim küme üyeleri

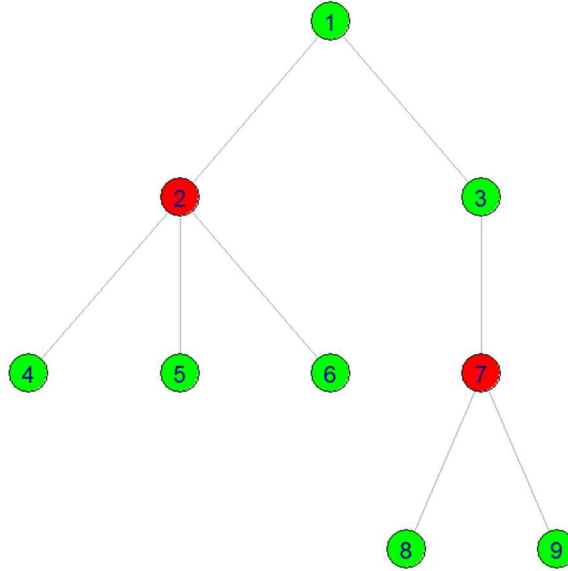
5.5 Ağaç Çizgelerde Uygulanması

9 düğüm ve 8 kenar bağıntısına sahip Şekil 5.24’de belirtilen ağaç çizge üzerinde önerilen hakim küme algoritması uygulanmıştır.



Şekil 5.24. Ağaç çizge

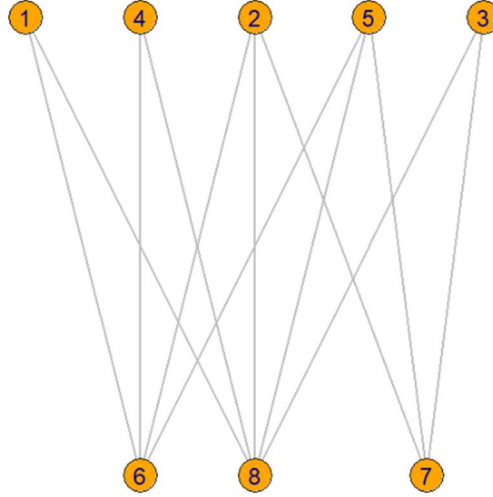
Ağaç çizgeler yapıları gereği süreklilik arzeden pendant düğümlere sahiptirler. İlk iterasyonda düğüm 4, 5, 6, 8 ve 9 pendant düğümler tespit edilmiştir. Bu düğümlerin ataları olan düğüm 2 ve düğüm 7 hakim küme üyeleri olarak tespit edilir. Çizgedeki tüm düğümlere bu iki düğümden ulaşıldığı için algoritma sonlanır. Şekil 5.25’de hakim küme üyeleri ve komşu düğümlerine ait çizge yapısı verilmiştir.



Şekil 5.25. Ağaç çizge hakim küme üyeleri

5.6 İki parçalı Çizgelerde Uygulanması

Algoritma diğer bir özel çizge türü olan Şekil 5.26'da verilen iki parçalı çizge üzerinde uygulanmıştır.



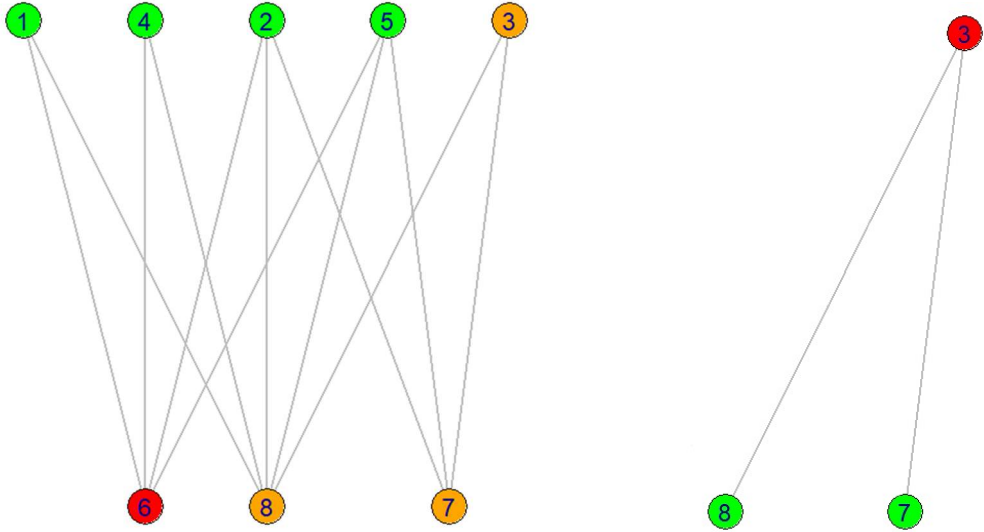
Şekil 5.26. İki parçalı çizge

İlk iterasyonda pendant düğüm bulunmadığı için Karcı baskınlık değeri hesaplanmıştır. Çizelge 5.6' da verilen Karcı baskınlık değerine göre düğüm 6 ve düğüm 8 en yüksek baskınlık değerine sahip olan düğümlerdir. Baskınlık değerleri eşit olduğu için Kmax derecesi küçük olan düğüm 6, Şekil 5.27 (a) da belirtildiği gibi çizgenin ilk hakim küme üyesi olarak seçilir.

Çizelge 5.6. İki parçalı (Bipartite) çizgeye ait baskınlık değerleri (Γ)

Düğüm No	Γ
1	7
2	9
3	7
4	7
5	11
6	15
7	11
8	15

Çizgedeki güncelleme sonrası Şekil 5.27 (b) deki pendant düğümlerin oluşturduğu çizge kalmaktadır. Düğüm 3 ikinci hakim küme üyesi olarak seçilir ve algoritma çalışmayı tamamlar.

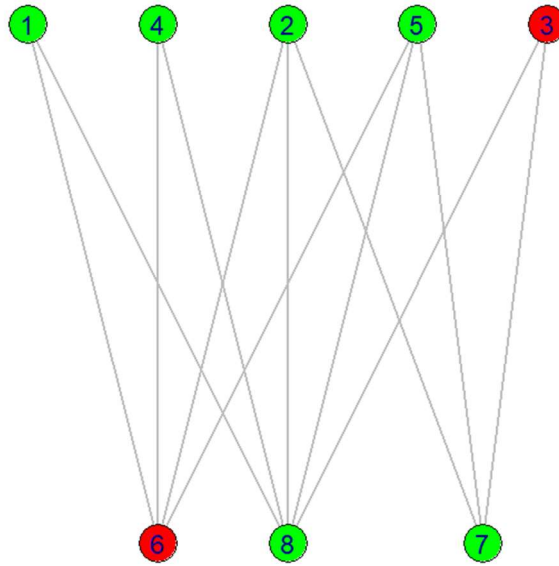


(a) İki parçalı çizge 1. iterasyon

(b) İki parçalı çizge 2. iterasyon

Şekil 5.27. İki parçalı çizge hakim küme iterasyonları

Şekil 5.28’de çizgedeki hakim küme üyeleri ve bu üye düğümlerin ulaşabildikleri komşu düğümler verilmiştir.



Şekil 5.28. Ağaç çizge hakim küme üyeleri

Çizgedeki tüm düğümlere tek bir düğüm üzerinden ulaşamadığından, seçilen düğüm 6 ve düğüm 3 için aynı zamanda minimum hakim küme üyeleridir denir.

6. SONUÇ VE ÖNERİLER

Sosyal ağlarda kişi ve nesnelere arasındaki ilişkiler çizgeler ile sayısal olarak modellenmektedir. Geliştirilen çizge algoritmaları sayesinde sosyal ağlardaki bir çok gizli örüntü keşfedilmektedir. En popüler örüntü keşifleri arasında çizgedeki baskın düğümlerin belirlenmesi, topluluk keşfi ve düğümler arasındaki benzerliklerin çıkarılması sayılabilir. Bu tez çalışmasında herhangi bir çizgedeki baskın düğüm ve hakim kümenin belirlenmesi için özgün bir algoritma önerilmiştir. Önerilen algoritmanın tüm aşamaları uygulamalı olarak açıklanmıştır. Çizgelerdeki hakim kümenin belirlenmesi için önerilen algoritma Karcı merkezlilik algoritması ve hakim küme üye seçim algoritmaları olmak üzere 2 temel aşamadan oluşmaktadır. Çalışmada önerilen Karcı merkezlilik algoritması ağırlıksız ve bağlı çizgeler üzerinde çalışmaktadır. Karcı merkezlilik algoritması çizgedeki tüm düğümler için baskınlık değeri üretmektedir. Üretilen Karcı merkezlilik (baskınlık) değerleri hakim küme üyelerinin belirlenmesinde seçim önceliği için kullanılmaktadır. Karcı merkezlilik algoritması literatürdeki diğer algoritmalar ile karşılaştırıldığında özgün bir yapıya sahiptir. Düğümlere ait baskınlık değerlerini belirlerken bu çalışmada önerilen alt algoritmalarından temel kesme algoritması ve Kmax ağaç oluşturma algoritmasının ürettiği değerler kullanılmıştır. Kmax ağacı düğümlerin dereceleri ve ağaçtaki seviyeleri göz önünde bulundurularak kuyruk sırasına göre oluşturulan bir kapsayan ağaç yapısıdır. Kesme dereceleri ise kapsayan ağacın kenar bağlantıları göz önünde bulundurularak üretilen değerlerdir. Kmax, kesme ve çizge düğüm derecelerinin toplamı Karcı merkezlilik değerini oluşturmaktadır.

Hakim kümenin belirlenmesi için geliştirilen Karcı merkezlilik algoritması literatürdeki merkezlilik algoritmalarından sayfa değeri, özvektör, yakınlık ve derece algoritmaları ile aynı çizge üzerine uygulanarak karşılaştırılmıştır. Uygulamadan elde edilen sonuçlar Karcı merkezlilik algoritması ile diğer popüler merkezlilik algoritmaları arasında kısmi benzerlikler olduğunu göstermektedir.

Ayrıca sayfa değeri, özvektör, yakınlık ve arasındalık merkezlilik algoritmaları gerçek bir ulaşım ağında uygulanarak varolan bir probleme çözüm bulmak amacı ile kullanılmıştır. Malatya kenti ulaşım ağına özgü olarak 4 algortmada ayrı ayrı uygulanmış ve elde edilen sonuçların, kent içerisindeki kavşak noktalarının baskı altında kalma diğer bir tanımla etkili olma durumlarının gerçeğe uygun olmadığı tespit edilmiştir. Bu uygulama özelinde önerilen hibrit bir yöntem sayesinde Malatya kentindeki kavşak noktalarının baskınlık değerleri

hesaplanmış elde edilen sonuçlar yandex haritaların gps izi verileri ile karşılaştırılmıştır. Karşılaştırma sonuçlarında önerilen hibrit yöntemin yüksek doğruluk oranında gerçek gps verilerini doğruladığı sonucuna ulaşılmıştır.

Diğer bir çalışmada sayfa değeri ve özvektör algoritmaları kullanılarak Malatya kentinde araç sayımlarının yapılabildiği akıllı kavşaklardaki mevcut sinyalizasyon planları neticesinde araç taşıma kapasiteleri ve yük durumuna ilişkin analizler gerçekleştirilmiştir. Bu analiz neticesinde birim yeşil sn süresince kavşaklara giriş yapan araç verileri dikkate alınarak aşırı yük altında olan kavşak noktaları tespit edilmiştir.

Özetle tez çalışmasında sosyal ağ kavramı açıklanmış ve gerçek dünya problemleri üzerinde başarılı sonuçlar üreten topluluk tespiti, merkezlilik ve benzerlik algoritmalarına uygulamalı olarak yer verilmiştir. Önerilen Karıcı merkezlilik algoritması literatürdeki popüler merkezlilik algoritmaları ile karşılaştırılarak başarısı analiz edilmiştir. Geliştirilen hakim küme algoritması örnekler üzerinde uygulamalı olarak açıklanmıştır. Önerilen algoritma bir çok çizge türünde minimum hakim küme sonuçlarını sağlamasına rağmen algoritmanın asıl hedefi çizgedeki hakim küme üyelerinin optimuma yakın etkili bir şekilde tespit edilmesidir. Algoritmanın sonraki süreçlerde geliştirilmesi ile maximum qlick ve max independet set gibi zor problem türlerinin çözümünde başarılı sonuçlar vereceği öngörülmektedir.

KAYNAKLAR

- [1] S. S. Özdemir, M. Özdemir, E. Polat, R. Aksoy, *Sosyal Medya Kavramı ve Sosyal Ağ Sitelerinde Yer Alan Online Reklam Uygulamalarının İncelenmesi*, **Electronic Journal Of Vocational Colleges**, Aralık, (2014).
- [2] H. Jinyu, G. Zhiwei and W. Pan, *Multiangle Social Network Recommendation Algorithms and Similarity Network Evaluation*, **Journal of Applied Mathematics**, Volume (2013).
- [3] B. M. Brentan, E. Campbell, G. L. Meirelles, E. Luvizotto and J. Izquierdo, *Social Network Community Detection for DMA Creation: Criteria Analysis through Multilevel Optimization*, **Mathematical Problems in Engineering**, Volume (2017).
- [4] R. Yang, Y. Liu, Y. Liu, H. Liu, W. Gan, *Comprehensive Public Transport Service Accessibility Index—A New Approach Based on Degree Centrality and Gravity Model*, **Sustainability**, (2019), 11(20):5634.
- [5] Y. Zhang, X. Wang, P. Zeng, X. Chen, *Centrality Characteristics of Road Network Patterns of Traffic Analysis Zones*, **Transportation Research Record**, (2011), 2256(1):16-24.
- [6] M. S. Demirci, Ş. Sağıroğlu, *Sosyal Ağ Verilerinin Kullanım Alanları Üzerine Kapsamlı Bir İnceleme*, **Gazi Üniversitesi Fen Bilimleri Dergisi, Part C: Tasarım Ve Teknoloji**, (2017), 5(2): 1-21.
- [7] M. Kaygusuz and K. İnce, *Modelling Academic Collaboration Network of Turkey as Graph and Dominating Set Analysis*, International Artificial Intelligence and Data Processing Symposium (IDAP), (2019), pp. 1-5.
- [8] S. Guze, *Graph Theory Approach to Transportation Systems Design and Optimization*, **The International Journal on Marine Navigation and Safety of Sea Transportation**, (2014), Volume 8, Number 4, December.
- [9] W. Xing and A. Ghorbani, *Weighted PageRank algorithm*, **Proceedings. Second Annual Conference on Communication Networks and Services Research**, (2004), Fredericton, NB, Canada, s. 305-314.
- [10] Z. Yuanyuan, J. Xiaohua, H. Yanxiang, *Energy Efficient Distributed Connected Dominating Sets Construction in Wireless Sensor Networks*, IWCNC, (2006), July, s.797–802.
- [11] T. Alahakoon, R. Tripathi, N. Kourtellis, R. Simha, A. Lamnitchi, *K-path centrality: a new centrality measure in social networks*, **SNS '11: Proceedings of the 4th Workshop on Social Network Systems**, (2011), April, s.1–6.
- [12] Y. Ding, E. Yan, A. Frazho, J. Caverlee, *PageRank for ranking authors in co-citation networks*, **Journal of the American Society for Information Science and Technology**, (2009), November, Volume60, Issue11, Pages 2229-2243.
- [13] V. Jagadishwari and S. Chakrabarty, *Link Prediction using Influencer nodes of a Social Network*, Second International Conference on Inventive Research in Computing Applications (ICIRCA), (2020), Coimbatore, India, s. 927-930.
- [14] T. Agryzkov, L. Tortosa, JF. Vicent, R. Wilson, *A centrality measure for urban networks based on the eigenvector centrality concept*, **Environment and Planning B: Urban Analytics and City Science**, (2019), 46(4) s. 668-689.

- [15] W. Maharani, A. A. Gozali, *Degree centrality and eigenvector centrality in twitter*, **8th International Conference on Telecommunication Systems Services and Applications (TSSA)**, (2014), pp. 1-5
- [16] S. Zhao, P. Zhao, Y. Cui, *A network centrality measure framework for analyzing urban traffic flow: A case study of Wuhan, China*, **Physica A: Statistical Mechanics and its Applications**, (2017), Volume 478, pp:143-157.
- [17] E. Mededovic, V. G. Douros and P. Mähönen, *Node Centrality Metrics for Hotspots Analysis in Telecom Big Data*, **IEEE INFOCOM**, (2019), pp. 417-422
- [18] E. Yan, Y. Ding, *Discovering author impact: A PageRank perspective*, **Information Processing & Management**, Volume 47, Issue 1, Pages 125-134
- [19] N. Wang, J. Dai, D. Li and M. Li , *An approximation algorithm for connected dominating set in wireless ad hoc network* ,**IET, ICISCE**, (2012), pp. 1-5.
- [20] L. Ruan, H. Du, X. Jia, W. Wu, Y. Li, K. Ko, *A greedy approximation for minimum connected dominating sets*, **Theoretical Computer Science**, (2004), 329(1-3), pp. 325-330.
- [21] X. Xu, Z. Tang, W. Sun, X. Chen, Y. Li, G. Xia, X. Bi, Z. Zong, *An algorithm for the minimum dominating set problem based on a new energy function*, **SICE 2004 Annual Conference**, (2004), pp. 924-926.
- [22] K.C. Ho, Y. P. Singh, H.T. Ewe, *An Enhanced Ant Colony Optimization Metaheuristic for The Minimum Dominating Set Problem*, **Applied Artificial Intelligence**, (2006), pp. 881-903
- [23] R. Jovanovic, M. Tuba, *Ant Colony Optimization Algorithm with Pheromone Correction Strategy for the Minimum Connected Dominating Set Problem*, **Computer Science and Information Systems**, (2013), 10(1), pp. 133-149.
- [24] D. Chalupa, *An order-based algorithm for minimum dominating set with application in graph mining*, **Information Sciences**, (2018), Volume 426, pp. 101-116.
- [25] Y. Zhou, G. Lv, B. Xiu, W. Zhang and Q. Cheng, *A faster approximate method to identify minimum dominating set*, **IEEE 5th International Conference on Software Engineering and Service Science**, (2014), pp. 443-448.
- [26] F. Grandoni, *A note on the complexity of minimum dominating set*, **Journal of Discrete Algorithms**, (2006), pp. 209-214.
- [27] D. L. Grinstead and P. J. Slater, *On minimum dominating sets with minimum intersection*, **Discrete Mathematics**, (1990), 86(1-3), pp. 239-254.
- [28] F. Oztemiz, A. Karcı, *Akademik Yazarların Yayınları Arasındaki İlişkinin Sosyal Ağ Benzerlik Yöntemleri ile Tespit Edilmesi*, **Uludağ University Journal Of The Faculty Of Engineering**, (2020), 25(1), Pp. 591-608.
- [29] F. Öztemiz, A. Karcı, *Malatya İli Ulaşım Ağı Kavşak Noktalarının Merkezilik Analizi*, **Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi**, (2021).
- [30] A. Şener, *Sosyal Ağ Tabanlı Modelleme Yöntemiyle GSM Kullanıcılarına En Uygun Tarifelerin Önerilmesi*, **Fırat Üniversitesi Fen Bilimleri Enstitüsü**, (2014), Yüksek Lisans Tez.
- [31] H. Sachs, M. Stiebitz, R. J. Wilson, *An Historical Note:Euler's Königsberg Letters*, **Journal of Graph Theory**, (1988), Vol. 12, No. 1, 133-139.

- [32] E. Erođlu, *A Single Chinese Postman Problem With Two Objectives*, ODTÜ, Yüksek Lisans Tezi, Ankara, Mayıs, (2015).
- [33] L. Debnath, *A brief historical introduction to Euler's formula for polyhedra, topology, graph theory and networks*, **International Journal of Mathematical Education in Science and Technology**, (2010), 41:6, 769-785.
- [34] B. Yıldırım, *Unitary cayley grafların baskınlık sayısının incelenmesi*, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya, Yüksek Lisans Tezi, Temmuz (2012).
- [35] İller Arası Mesafe Tespiti , <https://www.kgm.gov.tr/Sayfalar/KGM/SiteTr/Uzakliklar/illerArasiMesafe.aspx>, (Erişim tarihi: 14.08.2021).
- [36] M. Aşçı, C. U. İleri and O. Dağdeviren, *An energy-efficient capacitated minimum spanning tree algorithm for topology control in Wireless Sensor Networks*, 25th Signal Processing and Communications Applications Conference (SIU), (2017), pp. 1-4.
- [37] J. H. S. Alkhalissi, Ayla Saylı, *Negligence Minimum Spanning Tree Algorithm*, **Avrupa Bilim ve Teknoloji Dergisi**, Aralık (2018), Sayı 14, S. 70-76.
- [38] O. Goldreich, *P, NP and NP-Completeness*, **The Basics of Computational Complexity**, (2010), Cambridge University Press
- [39] J. W. Freeman, *Hard random 3-SAT problems and the Davis-Putnam procedure*, **Artificial Intelligence**, Volume 81, Issues 1–2, (1996), Pages 183-198.
- [40] M. Farrell, L. Levine, *The halting problem for chip-firing on finite directed graphs*, Cornell University, Mathematics Department Senior Thesis, December 2, (2014).
- [41] J. Wu and H. Li, *A Dominating-Set-Based Routing Scheme in Ad Hoc Wireless Networks*, **Telecommunication Systems**, (2001), 18:1–3, 13–36.
- [42] K. Kapoor, D. Sharma and J. Srivastava, *Weighted node degree centrality for hypergraphs*, **2013 IEEE 2nd Network Science Workshop (NSW)**, West Point, NY, (2013), pp. 152-155.
- [43] J. Zhou, Z. Zhang, W.Wu, K. Xing, *A greedy algorithm for the fault-tolerant connected dominating set in a general graph*, **J Comb Optim**, (2014), 28:310-319.
- [44] F. Altunbey, B. Alataş, *Sosyal Ağ Analizi İçin Sosyal Tabanlı Yapay Zekâ Optimizasyon Algoritmalarının İncelenmesi*, **International Journal of Pure and Applied Sciences**, (2015), 1: 33-52.
- [45] İ. Kervankıran, E. F. Sert, M. Çuhadar, *Türkiye 'de İç Turizm Hareketlerinin Sosyal Ağ Analizi ile İncelenmesi*, **Turizm Akademik Dergisi**, (2018), 5 (1), 29-50
- [46] Graph Algorithms, <https://neo4j.com/developer/graph-data-science/graph-algorithms/>, (Erişim Tarihi: 15.08.2021).
- [47] K. You, R. Tempo, L. Qiu, *Distributed Algorithms for Computation of Centrality Measures in Complex Networks*, **IEEE Transactions on Automatic Control**, 62(5), 2080-2094, 2017
- [48] M. Xu, J. Wu, M. Liu, Y. Xiao, H. Wang, D. Hu, *Discovery of Critical Nodes in Road Networks Through Mining From Vehicle Trajectories*, **IEEE Transactions on Intelligent Transportation Systems**, (2019), 20(2), 583-593.
- [49] U. Brandes, *On variants of shortest-path betweenness centrality and their generic computation*, **Social Networks**, (2008), 30(2), 136-145.

- [50] P. Bonacich, *Some unique properties of eigenvector centrality*, **Social Networks**, (2007), 29(4), 555-564.
- [51] A. E. Sariyüce, K. Kaya, E. Saule, Ü. V. Çatalyürek, *Incremental algorithms for closeness centrality*, IEEE International Conference on Big Data, Silicon Valley, CA, October (2013), 487-492.
- [52] Yandex Map Editor, <https://mapeditor.yandex.com.tr/>, (Erişim Tarihi: 07.04.2021).
- [53] N. Gürsakal, *Sosyal Ağ Analizi: PAJEK, UNICET ve GMINE Uygulamalı*, Dora Basım Yayın: Bilgisayar Kitapları Dizisi, Bursa, (2009), sayfa 513
- [54] S. Fortunato, *Community detection in graphs*. **arXiv:0906.0612** (2009).
- [55] B. Onur, *SOSYAL AĞLARDA TOPLULUK YAPILARININ ANALİZİ*, İnönü Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ekim (2012), Malatya.
- [56] M. K. Keleş and S. A. Özel, *Similarity detection between Turkish text documents with distance metrics*, International Conference on Computer Science and Engineering (UBMK), (2017), pp. 316-321.
- [57] S. Niwattanakul, J. Singthongchai, E. Naenudorn and S. Wanapu, *Using of Jaccard Coefficient for Keywords Similarity*, Vol I,IMECS, (2013), March 13 - 15,Hong Kong
- [58] M. Shameem and R. Ferdous, *An efficient k-means algorithm integrated with Jaccard distance measure for document clustering*, First Asian Himalayas International Conference on Internet, (2009), pp. 1-6.
- [59] H. Bulut, M. Milli, *İşbirlikçi filtreleme için yeni tahminleme yöntemleri*, **Pamukkale Univ Muh Bilim Derg**, (2016), 22(2), 123-128.
- [60] Kosinü benzerlik, <https://bilgisayarkavramlari.com/2012/11/08/kosinus-benzerligi-cosine-similarity-2/>, (Erişim tarihi:15.08.2021).
- [61] A.P.Plerou, S.Sioutas, V.G.Salaidis, *CSMR: A Scalable Algorithm for Text Clustering with Cosine Similarity and MapReduce*, IFIP International Conference on Artificial Intelligence Applications and Innovations, September (2014).
- [62] M. Işık, A. Y. Çamurcu, *Web Belgeleri Kümelemede Benzerlik Ve Uzaklık Ölçütleri Başarılarının Karşılaştırılması*, **Marmara Üniversitesi Fen bilimleri Enstitüsü Dergisi**, (2008), 20, 35-49
- [63] R. Uehara, S. Toda, T. Nagoya, *Graph isomorphism completeness for chordal bipartite graphs and strongly chordal graphs*, **Discrete Applied Mathematics**, (2005), 145(3), pp. 479- 482.

ÖZGEÇMİŞ

Ad-Soyad : Furkan ÖZTEMİZ

ÖĞRENİM DURUMU:

- **Lisans** : 2012, Fırat Üniversitesi, Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü
- **Yüksek Lisans** : 2017, İnönü Üniversitesi, Bilgisayar Mühendisliği Anabilim Dalı
Kontrol Kumanda Sistemleri Programı
- **Doktora** : 2021, İnönü Üniversite, Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Bilimleri Programı

MESLEKİ DENEYİM:

- 2012 - 2012 BMF Yazılım.
- 2013 - 2017 Esenlik A.Ş.
- 2018 - 2021 Malatya Büyükşehir Belediyesi

DOKTORA TEZİNDEN TÜRETİLEN ÇALIŞMALAR

- **Öztemiz F., Karci A.**, “Malatya ili Ulaşım Ağı Kavşak Noktalarının Merkezlilik Analizi”, Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, (2021).
- **Öztemiz F., Karci A.**, “Bağlı Graflarda Etkili Dğümlerin Belirlenmesinde Yeni Bir Yaklaşım”, Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi, (2021).
- **Öztemiz F., Karci A.**, “Akademik Yazarların Yayınları Arasındaki İlişkinin Sosyal Ağ Benzerlik Yöntemleri İle Tespit Edilmesi”, Uludağ University Journal of The Faculty of Engineering , 25 (1) , 591-608, (2020).
- **Öztemiz F.**, “Sinyalizasyon Verileri İle Malatya Kenti Ulaşım Ağı Kavşak Noktalarının Merkezlilik Analizi” . Computer Science, (2021).