T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

KARMAŞIK AĞLARDAKİ MODÜL YAPILARININ VE ANLAMLI ALT-AĞLARIN TESPİTİ

Yılmaz ATAY

DOKTORA TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Nisan-2018
KONYA

TEZ KABUL VE ONAYI

Yılmaz ATAY tarafından hazırlanan "Karmaşık Ağlardaki Modül Yapılarının ve Anlamlı Alt-Ağların Tespiti" adlı tez çalışması 20/04/2018 tarihinde aşağıdaki jüri tarafından oy birliği ile Selçuk Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda DOKTORA TEZİ olarak kabul edilmiştir.

Jüri Üyeleri	**İmza**
Başkan Prof. Dr. Harun UĞUZ	.....
Danışman Doç. Dr. Halife KODAZ	.....
Üye Doç. Dr. İsmail BABAOĞLU	.....
Üye Dr. Öğr. Üyesi Mehmet HACIBEYOĞLU	.....
Üye Dr. Öğr. Üyesi Onur İNAN	.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Mustafa YILMAZ
FBE Müdürü

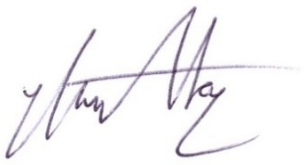
Bu tez çalışması, Türkiye Bilimsel ve Teknolojik Araştırma Kurumu (TÜBİTAK) tarafından 1649B031402383 numaralı 2211–C Yurt İçi Öncelikli Alanlar Doktora Burs Programı ve 1059B141500230 numaralı 2214–A Yurt Dışı Doktora Sırası Araştırma Burs Programı ile desteklenmiştir.

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.



Yılmaz ATAY

Tarih: 20.04.2018

ÖZET

DOKTORA TEZİ

KARMAŞIK AĞLARDAKİ MODÜL YAPILARININ VE ANLAMLI ALT-AĞLARIN TESPİTİ

Yılmaz ATAY

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Halife KODAZ

2018, 247 Sayfa

Jüri
Prof. Dr. Harun UĞUZ
Doç. Dr. Halife KODAZ
Doç. Dr. İsmail BABAOĞLU
Dr. Öğr. Üyesi Mehmet HACIBEYOĞLU
Dr. Öğr. Üyesi Onur İNAN

Anlambilimsel, biyolojik, ekolojik, medikal, sosyal, telekomünikasyon ve ulaşım ağları gibi çok çeşitli alanlarda mevcut olan karmaşık ağ sistemleri ortaya çıkarılabilecek oldukça önemli bilgiler barındırır. Bu tür gerçek ağlardaki milyonlarca nesnenin ve etkileşimin analizi de bir o kadar zordur. Bu noktada fizik, istatistik ve matematik gibi temel bilimlerin yanında bilgisayar bilimleri, biyoinformatik, mühendislik, sosyal ve teknolojik alanlardaki disiplinlerle dinamik olarak sürekli etkileşim halinde olan ağ bilimi kavramı ön plana çıkmaktadır. Farklı alanlardan çeşitli yöntemlere sahip olan bu çok disiplinli bilim dalında gerçek-dünya sistemlerinin modellenmesinde genellikle çizgeler kullanılır. Böylece temelde çizge teorisi ve bilgisayar bilimleri yardımıyla gerçek ağ yapılarındaki gizli ve önemli bilgiler çeşitli yöntemler kullanılarak keşfedilebilmektedir. Bu bilgilerin bilgisayar sistemleri yardımıyla hızlı bir şekilde analiz edilerek ortaya çıkarılabilmesi amacıyla son yıllarda çeşitli çalışmalar yapılmakta ve farklı ağ analiz teknikleri önerilmektedir. Bu doktora tez çalışmasında, gerçek-dünya ağlarından anlamlı bilgilerin ortaya çıkarılmasında kullanılan iki farklı ağ analiz konusuna odaklanılmıştır. Bunlardan ilki, ağ modül ya da topluluk tespiti problemi hakkındadır. İkinci ise klinik kanser verileri ve bunlarla ilişkili biyolojik ağların analizleri hakkındadır. Tezdeki ilk çalışma konusunda, gerçek sistemleri oluşturan nesnelerin fiziksel etkileşim yoğunluklarına ya da fonksiyonel ilişkilerine göre modül yapılarının tanımlanması amaçlanır. Bu problemde kullanılan ağların oldukça karmaşık yapılarda olması ve ortaya çıkarılacak anlamlı bilgilerin ancak yüksek hesaplama maliyeti ile elde edilebilmesi gibi kısıtlamalar sebebiyle klasik ve sezgisel yöntemler yerine bu tez çalışmasında çeşitli metasezgisel yaklaşımlar kullanılmıştır. Bu amaçla sekiz farklı metasezgisel optimizasyon algoritması orijinal mekanizmalarıyla ya da çeşitli hibrit yaklaşımlarla bu probleme adapte edilmiştir. Bu algoritmalar iki farklı model ile üretilen sekiz farklı yapay ağda test edilmiştir. Ayrıca algoritmaların performanslarının gerçek sistemler üzerinde de karşılaştırılabilmesi amacıyla genellikle literatürde tercih edilen altı gerçek-dünya ağı kullanılmıştır. Buradaki karşılaştırmalarda nesnel değerlendirmeler için istatistiksel analizler yapılmıştır. Tüm testlerin

sonuçlarına göre genellikle en başarılı sonuçlara ulaşan *3pHybrid* algoritması, bu problemle ilgili gerçekleştirilen sonraki deneylerde test yöntemi olarak kullanılmıştır. Bahsi geçen testlerin ilkinde, ağ modül tespitinde çoğunlukla tercih edilen ve iyi bilinen on farklı amaç fonksiyonundan genellikle en uygun sonuçları veren fonksiyonun tespitine çalışılmıştır. Bu testlerde biyoloji, gen, protein, radar, sosyal ve tıbbi alanlardan temin edilen bilgi ağları kullanılmıştır. Bu ağlar için en uygun modül yapıları bilinmektedir. Böylece test edilen amaç fonksiyonları ile elde edilen ağ modüllerinin kaliteleri bu tez çalışmasında kullanılan altı farklı küme değerlendirme ölçütü ve gerçek ağ modül yapıları ile test edilmiştir. Bunların sonuçlarına göre genel başarı sıralamaları dikkate alındığında, genellikle en başarılı küme değerlendirme sonuçlarına modülerlik fonksiyonu ile ulaşılmış olsa da buna yakın sonuçlara ulaşan diğer fonksiyonların da bazı durumlarda tercih edilmesinin uygun olacağı sonucuna varılmıştır. Çünkü bazı test ağlarında modülerlik amaç fonksiyonu ile daha az kaliteli ağ modüllerinin tespit edildiği gözlemlenmiştir. Yine de ortalama değerlendirme sonuçları açısından bu fonksiyon ile genellikle en iyi ortalama sonuçlara ulaşıldığından sonraki testlerde uygunluk kriteri olarak aynı fonksiyon kullanılmıştır. Bu problemle ilgili en son çalışmada, genel olarak en iyi sonuçları sunan algoritmanın uygunluk kriteri olarak seçilen ve çoğunlukla en kaliteli ağ modül yapılarını sunan modülerlik amaç fonksiyonu ile elde edilen nihai skorlar literatürdeki diğer algoritmaların skorları ile karşılaştırılmıştır. Bu testlerde 21 farklı algoritma, 13 adet gerçek dünya ağında mevcut olan skorlara göre karşılaştırılmıştır. Tüm sonuçlara göre bu ağlardan 11'inde en yüksek skorlara tez çalışmasında önerilen algoritmalardan biri olan *3pHybrid* algoritması ile ulaşılmıştır.

Bu tezin ikinci çalışma konusunda, klinik kanser verileri ve karmaşık biyolojik ağların sağkalım analizleri ile birlikte değerlendirilmesiyle genlerin fonksiyonel etkileşimlerinden oluşan anlamlı alt-ağların tespitine odaklanılmıştır. Buradaki analizler sağkalım süreleri ile maksimum ilişkili olduğu düşünülen/varsayılan gen listelerinin ortaya çıkarılmasını kapsar. Burada kullanılan tüm genler özellikle kanser hastalıklarında çokça üzerinde durulan kopya sayısı değişikliklerini içerir. Genlerdeki bu değişikliklerle ilgili sunulan problem, ilk kez bu çalışmada anlatılmıştır. Burada, hastalıklarda ciddi klinik etkilere sahip olabilecek kopya sayısı değişikliklerinin temel alınmasıyla hastaların yaşam—*sağkalım* sürelerine etki edebilecek gen gruplarının ortaya çıkarılması amaçlanmıştır. Bununla ilgili gerçekleştirilen deneylerde, beş farklı kanser türü için temin edilen klinik hasta bilgileri ile bu hastaların sahip oldukları gen-gen etkileşim ağları kullanılmıştır. İlgili tez bölümünde bu problemin çözümü amacıyla dinamik programlama ve genetik algoritma temelli iki farklı teknik önerilmiştir. Burada uygunluk fonksiyonu olarak sağkalım analizinde önemli olan log-rank istatistik ölçütü kullanılmıştır. Önerilen iki yöntemin skorlarına ve çalışma sürelerine göre performansları birbirleriyle kıyaslanmıştır. Testler için gerçek verilerin kullanılmasının yanında rastsal olarak üretilen yapay veriler de kullanılmıştır. Bu problemle ilgili tüm deneyler sonunda kaydedilen genlerden bazılarının ilişkili oldukları hastalıkların ya da biyolojik bozuklukların/değişikliklerin listeleri deneysel çalışmalar bölümünde verilmiştir. Son olarak, hem birinci hem de ikinci problemin birlikte dikkate alınmasıyla gen-hastalık ilişkileri ayrıntılı olarak analiz edilmiştir. Bu en son ve yeni problemde, aynı modüllerde bulunan ve ilgili kanser türlerindeki sağkalım analizlerinde en fazla etkili oldukları kabul edilen genlerin tespitine odaklanılmıştır. Bu amaçla düğümler arası etkileşimlerin çok karmaşık olduğu biyolojik ağlardaki modül yapılarının tespiti problemi ile sağkalım analizi probleminin bir arada değerlendirildiği tüm deneysel sonuçlar ilgili eklerde ve ek dosyalarda sunulmuştur.

Anahtar Kelimeler: Ağ bilimi ve analizi, karmaşık ağlar, klinik veriler, kopya sayısı değişikliği, küme değerlendirme, metasezgisel yaklaşımlar, sağkalım analizi, topluluk—modül tespiti.

ABSTRACT

Ph.D THESIS

DETECTION OF MODULE STRUCTURES AND SIGNIFICANT SUB-NETWORKS IN COMPLEX NETWORKS

Yılmaz ATAY

The Graduate School of Natural and Applied Science of Selçuk University
The Degree of Doctor of Philosophy
In Computer Engineering

Advisor: Assoc. Prof. Dr. Halife KODAZ

2018, 247 Pages

Jury
Prof. Dr. Harun UĞUZ
Assoc. Prof. Dr. Halife KODAZ
Assoc. Prof. Dr. İsmail BABAOĞLU
Asst. Prof. Dr. Mehmet HACIBEYOĞLU
Asst. Prof. Dr. Onur İNAN

Complex network systems, which can be exist in a wide range of fields such as semantic, biological, ecological, medical, social, telecommunication and transport networks, contain considerable information that can be uncovered. In such real-networks, the analysis of millions of objects and their interactions is so difficult. At this point, the concept of network science, which constantly is in interaction dynamically with the fundamental disciplines such as physics, statistics and mathematics, as well as disciplines in computer science, bioinformatics, engineering, social and technological fields, comes into prominence. In this multidisciplinary science, which has various methods from different fields, the graphs are generally used in modeling real-world systems. Thus, basically, with the help of graph theory and computer science, hidden and valuable information in real network structures can be discovered using various methods. In recent years, numerous studies have been carried out and different network analysis techniques have been proposed to analyze and to reveal this information with the help of computer systems. In this doctoral thesis, it has been focused on two different network analysis topics that are used to reveal meaningful information from real-world networks. The first of these is about network module or community detection problem. The second is about the analysis of clinical cancer data and related biological networks. In the first study in the thesis, it is aimed to define the module structures according to the physical interaction intensities or functional relations of the objects forming the real systems. Because of the complexity of the networks used in this problem, and the limitations such as meaningful information to be revealed that can be only obtained with high computational cost, various metaheuristic approaches have been used instead of classical and intuitive methods in this thesis. For this purpose, eight metaheuristic optimization algorithms have been adapted to this problem with the original mechanisms or with various hybrid approaches. These algorithms have been tested in eight different random networks

produced with two different models. In addition, six real-world networks, which are generally preferred in the literature, have been used to compare the performance of algorithms on real-systems. In these comparisons, the statistical analyzes have been carried out for objective evaluations. According to the results of all the tests, the *3pHybrid* algorithm, which usually achieves the most successful results, has been used as a test method in subsequent experiments on this problem. In the first of the aforementioned tests, it has been studied to determine the function that gives generally the most appropriate results, from the most commonly preferred, and well-known ten different objective functions in network module detection. In these tests, information networks are used which are obtained from biology, gene, protein, radar, social and medical fields. The most suitable module structures for these networks are known. Thus, the qualities of network modules obtained with the objective functions tested were examined with six different cluster evaluation criteria and real network module structures used in this thesis study. When the rankings of general success are taken into consideration, it is generally concluded that the most successful cluster evaluation results are achieved with modularity function, but other functions that reach comparable results can be preferred in some cases. Because in some test networks it has been observed to obtained less quality network modules with the modularity objective function. However, the same function was used as the eligibility criterion in subsequent tests, since it usually has been achieved the best average results in terms of average evaluation results with this function. In the final study with this problem, the final scores obtained with the modularity objective function, which is selected as the eligibility criterion of the algorithm that provides the best overall results, and which mostly provides the best quality network module structures, are compared with the scores of other algorithms in the literature. In these tests, 21 different algorithms were compared according to the scores available on 13 real-world networks. According to all the results, the highest scores in 11 of these networks were reached with the *3pHybrid* algorithm, one of the algorithms proposed in this study.

In the second study of this thesis, it has been focused on the identification of meaningful sub-networks consisting of functional interactions of genes by evaluating clinical cancer data and survival analysis of complex biological networks. These analyzes include the detection of sub-genomic networks that are assumed/considered to be most relevant to survival times. All genes contain copy number alterations which are particularly important in cancer diseases. The problem with these changes in the genes is presented for the first time in this thesis. Here, it is aimed to reveal gene groups that can affect the life-survival times of patients based on copy number alterations which can have serious clinical effects in diseases. In the related experiments, the clinical patient information provided for five different types of cancer and the gene-gene interaction networks possessed by these patients have been used. Two different methods based on dynamic programming and genetic algorithm have been proposed to solve this problem in the related section of thesis. Here, the log-rank statistical criterion, which is important in survival analysis, is used as a fitness function. The performances of the two proposed methods have been compared with each other according to their scores and their execution time. In addition to the used real data for tests, randomly generated artificial data have been also used in the experiments. The lists of diseases or biological disorders/changes which some of the recorded genes are related at the end of all experiments related to this problem are given in the section of experimental studies. Finally, gene-disease relationships have been analyzed in detail, taking both the first and second problems together. In this last and new problem, it has been focused on identification of genes found in the same network modules that are considered to be most effective in the survival analysis of the related cancer types. For this purpose, all the experimental results in which the problem of detection of module structures in biological networks where interactions between nodes are very complex, and the survival analysis problem are evaluated together, have been presented in the related appendices and in the supplementary files.

Keywords: Network science and analysis, complex networks, clinical data, copy number alteration, cluster evaluation, metaheuristic approaches, survival analysis, community—module detection.

ÖNSÖZ

Bu doktora tez çalışmasının başlangıcından itibaren teşvik edici fikirleriyle bana rehberlik eden, önerilerini ve katkılarını esirgemeyen tez danışmanım ve çok değerli hocam Doç. Dr. Halife Kodaz'a en içten teşekkürlerimi sunarım.

Doktora tez izleme komitesinde yer alan ve tavsiyeleriyle çalışmalarıma destek veren Prof. Dr. Harun Uğuz'a ve Dr. Öğr. Üyesi Mehmet Hacıbeyoğlu'na teşekkür ederim. Ayrıca, çok iyi bir çalışma ortamı sunan Selçuk Üniversitesi'ne ve bu süre boyunca fikirlerini paylaşmaktan çekinmeyen Bilgisayar Mühendisliği Bölümündeki çalışma arkadaşlarıma ve hocalarıma teşekkürlerimi sunarım.

Bu çalışmanın şekillenmesinde 2214/A Yurt Dışı Doktora Sırası Araştırma Burs Programı ile önemli katkılar sağlayan ve 2211/C programı ile de maddi destekler sunan Türkiye Bilimsel ve Teknolojik Araştırma Kurumu'na teşekkürlerimi iletmek isterim. Bir de doktora tez çalışmasının son yılında sundukları sıcak çalışma ortamı için Florida Üniversitesi'ne (UF, USA) ve orada kaldığım süre zarfında çalışmalarıma destek veren Prof. Dr. Tamer Kahveci'ye teşekkür ederim.

Son olarak, bu zorlu süreçte gösterdikleri sonsuz sabır ve anlayışla tez çalışmam boyunca desteklerini esirgemeyen sevgili aileme en kalbi teşekkürlerimi sunarım.

Yılmaz ATAY
Konya-2018

İÇİNDEKİLER

SİMGELER VE KISALTMALAR

CNA : Kopya sayısı değişikliği
PPI : Protein-protein etkileşim
LAR : Yer seçimi tabanlı komşuluk temsili
LFR : Lancichinetti–Fortunato–Radicchi benchmark
TCGA : Kanser Genom Atlası
PSO : Parçacık Sürü Optimizasyonu
GA : Genetik Algoritma
DP : Dinamik Programlama
ER (modeli) : Erdos–Renyi
BA (modeli) : Barabasi-Albert
F_M : Modularity
F_C : Conductance
F_{ID} : Internal Density
F_{GD} : Global Density
F_{CR} : Cut Ratio
F_{NC} : Normalized Cut
F_{FS} : Fitness Score
F_{ODF} : Average Out Degree Fraction
F_S : Significance
F_{Sr} : Surprise
MI : Karşılıklı bilgi, E_{MI}
NMI : Normalize edilmiş karşılıklı bilgi, E_{NMI}
RI : Rastgelelik indeksi, E_{RI}
ARI : Düzenlenmiş rastgelelik indeksi, E_{ARI}
JI : Jaccard indeksi, E_{JI}
P : Kalıcılık, E_P
GBM : Glioblastoma Multiforme
LAML : Acute Myeloid Leukemia
HNSC : Head and Neck Squamous Cell Carcinoma
KIRC : Kidney Renal Clear Cell Carcinoma
STAD : Stomach Adenocarcinoma
BA : Ayrık Yarasa Algoritması
SACO : Seçime Dayalı Karınca Kolonisi Optimizasyonu
EPSO : Zenginleştirilmiş Parçacık Sürü Optimizasyonu
GSA : Ayrık Yerçekimsel Arama Algoritması
ISFLA : Geliştirilmiş Kurbağa Sıçrama Algoritması
AGA : Adaptif Genetik Algoritma
SSA : Dağınık Arama Temelli Algoritma
3pHybrid : Üç Fazlı Hibrit Yaklaşım
DPT : Dinamik Programlama Temelli Yaklaşım
GAT : Genetik Algoritma Temelli Yaklaşım
GS2D : Gene set to diseases
p_{skor} : Normalize edilmiş log-rank istatistik skoru
SNAP : Stanford Ağ Analiz Platformu

1. GİRİŞ

1.1. Tezin Amacı ve Önemi

Biyolojik, ekolojik, ekonomik, sosyal, teknolojik ve benzeri bilgileri barındıran gerçek dünya sistemleri ağ yapıları olarak tanımlanırlar. Bu tür sistemlerin ayrıntılı bir şekilde analizi için disiplinler arası bir iş birliği gerekir ve bu gereksinim ağ analizi problemlerinin bilgisayar sistemleri ile çözümünü zorunlu kılar. Karmaşık sistemlerin analizi konusundaki çalışmalar köklü bir geçmişe dayansa da bilim ve teknolojideki son gelişmeler bu alanın yeni disiplinlerle birlikte güncelliğini koruduğunu göstermektedir. Bu sebeple gerçek dünya problemleri bilgisayar sistemleri yardımıyla modellenmekte ve bu modellere uygun olarak geliştirilen yeni yöntemlerle ya da algoritmalarla mevcut problemlere çözümler üretilmektedir. Ancak gerçek karmaşık sistemlerin modellenmesi ve incelemesi, bu sistemlerdeki elemanların karmaşık ilişkilerinden dolayı oldukça zordur. Çünkü gerçek dünya verilerinin temsilinde kullanılan sistemler; insanlar arası sosyal ilişkiler, canlılar arası yiyecek paylaşımı ve rekabeti, bilgisayarlar arası bilgi alışverişi, vücuttaki moleküler yapıların birbirleriyle etkileşimi gibi oldukça önemli ve elde edilmesi zor bilgiler barındırırlar. Gerçek dünya ağlarının temsil edilmesinde ve modellenmesinde genellikle çizge yapıları tercih edilmektedir (Fortunato 2010). Çizge yapılarındaki düğümler gerçek dünya sistemlerindeki temel nesnelere karşılık gelirken; nesneler arası ortak etkileşimler bağlantılarla temsil edilirler (Boccaletti ve ark 2006, Lancichinetti ve Fortunato 2014). Bu yapılarla ilgili önemsiz gibi görünen ve ağdaki topolojik özelliklerden yararlanılarak basit analiz teknikleri ile ortaya çıkarılamayan önemli bilgiler ağ örüntülerini ifade ederler. Keşfedilecek bilgileri temsil eden bu örüntüler karmaşık yapılardaki dinamik süreçlerin anlaşılmasına yardımcı olurlar (Newman 2001, Fortunato 2010, Gach ve Hao 2012). Gerçek sistemlerdeki anlamlı ve önemli bilgilerin tespiti için öncelikle bu sistemleri ifade eden ağlara/çizgelere özgü hem istatistiksel hem de matematiksel nitelikler ve nicelikler (temsili çizgelerin yönlü ya da yönsüz oluşu, bağlı bileşenlerin sayıları, çap, düğümlerin derece dağılımları, ortalama yol uzunluğu ve diğer önemli topolojik özellikler) belirlenir. Bu özellikler genellikle temsil edilen karmaşık sistemlerin davranışlarının ve eğilimlerinin tahmin edilmesinde kullanılır (Newman 2003). Buradaki istatistiksel ya da matematiksel çıkarımlar dışında ağ toplulukları ya da modülleri diye ifade edilen anlamlı alt-grupların tespitiyle de karmaşık sistemlerde mevcut olan bazı önemli bilgilere ulaşılabilmektedir

(Fortunato 2010, Newman 2010). Herhangi bir karmaşık gerçek dünya ağı bir çizge yapısı ile temsil edildiğinde; elde edilen alt-gruplar kendi içlerinde maksimum etkileşim, konumsal/yapısal benzerlik, ortak görevler gibi temel özelliklere sahip olurlar. Bu alt-grupların üyeleri, kendi kümesindeki (aynı modül/topluluk) üyelerle maksimum ilişkiye sahip olup fonksiyonel ya da yapısal olarak daha fazla ortak özellik barındırırlar; ancak buna karşılık diğer kümelerdeki (farklı modül/topluluk) üyelerle minimum ilişkiye ve daha az ortak özelliğe sahip olurlar (Girvan ve Newman 2002, Newman 2004 (a), Fortunato 2010).

Ağ toplulukları ya da modülleri özellikle gerçek dünya ağlarında nesneler arası yapısal veya fonksiyonel ilişkilerin belirlemesinde ve ağ kimliklerinin tanımlanmasında çok önemli bilgiler sağlar. Bu yüzden bu modüllerin ortaya çıkarılması karmaşık ağların analizi açısından oldukça önemlidir. Örneğin, sosyal gruplardaki arkadaşlıkların tespiti, sosyal etkinlik analizi ve terörist ataklarla ilgili gizli bağların ortaya çıkarılması gibi sosyal ağlardaki analizler (Marcus ve ark 2007) ile hücrelerdeki proteinler veya diğer moleküller arasında gerçekleşen fiziksel, biyolojik ya da fonksiyonel ilişkilerin modellendiği bazı etkileşim ağlarının detaylı analizi ve fonksiyonel tahmini (Lee ve Lee 2013) gibi konularda ağ modüllerinin tespitiyle önemli ve anlamlı bilgilere ulaşılabilmektedir. Ayrıca bir biyolojik ağdaki molekül topluluğunun işlevinin bilindiği varsayılarak bu toplulukla etkileşimde olan fakat işlevi bilinmeyen diğer bir molekülün ya da molekül topluluğunun fonksiyonel işlevi hakkında bazı tahminlerde bulunulabilir. Bu şekilde bir yaklaşımla Palla ve diğerleri (Palla ve ark 2005) tarafından literatüre sunulan bir çalışmada, örnek olarak verilen *Saccharomyces Cerevisiae* isimli bir maya (*yeast*) türünün *PPI* ağıyla ilgili bir grafikte gösterilen (bilgi için (Palla ve ark 2005) tarafından sunulan makaledeki Şekil 3'e bakınız), "*ribosome biogenesis/assembly*" topluluğundaki "*Ycr072c*" proteininin hücre canlılığı için önemli olduğu bilgisinden ve buna benzer diğer bilinenlerden yola çıkılarak fonksiyonu bilinmeyen başka ağ modüllerinin/toplulukların işlevlerinin tahmin edilebileceği belirtilmiştir. Bu şekilde hayati öneme sahip bazı bilgilerin ortaya çıkarılabilmesi için modül yapılarının tespiti ile ağ analizinin gerçekleştirilmesi mümkün olmaktadır.

Gerçek dünya ağlarının modüler yapılarının tespiti problemi son zamanlarda ağ biliminde odaklanılan en temel konulardan biri haline gelmiştir (Amiri ve ark 2013). Topluluk tespitiyle ağa özgü gizli bilgilerin ortaya çıkarılmasına çalışıldığı için ağdaki toplulukların başarılı bir şekilde tespiti oldukça zor bir problemdir. Ağdaki düğümlerin diğer tüm düğümlerle yoğun veya seyrek bir şekilde bağlantılı olma ihtimali göz önüne

alındığında ve bu ağlarda en az binlerce ya da milyonlarca etkileşimin olduğu düşünüldüğünde bu probleminin zorluğu daha iyi anlaşılabilir. Karşılaşılan buna benzer kısıtlamalar dikkate alındığında, bu tür karmaşık problemlerin çeşitli optimizasyon teknikleriyle çözülmeye çalışılması zorunlu hale gelmektedir. Gerçek dünya süreçlerinin temsil edildiği ağ yapıları, düğümleri ve bağlantıları rastsal olarak oluşturulmuş yapay ağlardaki gibi rastgelelik üzerine kurgulanmamıştır. Bu sebeple gerçek ağların analizinin de belli kurallara göre planlanması gerekmektedir. Bugüne kadar çeşitli klasik sezgisel ve metasezgisel yöntemler ağlardaki anlamlı yapıların ortaya çıkarılması için önerilmiştir. Klasik yöntemlerin ağ yapısına olan bağlılıkları sebebiyle birtakım dezavantajlara sahip oldukları bilinmektedir. Ayrıca ağlardaki düğüm ve etkileşim sayılarının artması da klasik yöntemlerdeki başarı oranını düşürmüş ve çözüm elde etme süresinde de ciddi sorunlarla karşılaşılmasına sebep olmuştur. Bilimsel ve teknolojik ilerlemelerin hızla arttığı bir zamanda gerçek dünya sistemlerinin modellenmesi ve analizi gittikçe daha da önemli bir hale gelmektedir. Örneğin; İnsan Genom Projesinin (*The Human Genome Project*) (Collins ve ark 1998) planlanmış önemli amaçlarına ulaşılmasının hemen ardından çeşitli biyolojik ağlardaki analizlerin gerçekleştirilmesinin önü açılmıştır. Ayrıca kanser türü hastalıklardan alzaymır gibi hastalıklara kadar her türlü probleme ışık tutacak bazı gen ağları, protein yolakları (*pathways*) ve protein etkileşim ağları (Collins ve ark 2003) gibi çeşitli biyolojik veriler bilim insanlarının kullanımına sunulmuş ve hastalık tedavisinde önemli bir konu olan ilaç keşfi çalışmaları da hız kazanmıştır (Butcher ve ark 2004). Buna benzer verilerin başarılı yöntemlerle analizi ve bilgi keşfi canlıların yaşamına önemli derecede katkı sağlamakta ve özellikle ağ modüllerinin ya da istatistiksel olarak anlamlı alt-ağların başarılı bir şekilde tespiti için çeşitli yaklaşımlar önerilmektedir. Bu yaklaşımları temel alarak geliştirilen yöntemlerin çoğu basit gerçek dünya ağlarında başarılı sonuçlara ulaşsa da mevcut yöntemlerin çoğunun büyük ve karmaşık ağlardaki performansları beklenenin altındadır. Bununla birlikte, birçok yöntem ağdaki topluluk sayısı ve topluluklardaki üye sayıları gibi bazı temel ön bilgilere de ihtiyaç duyarlar. Yaklaşık olarak 2002 yılı itibariyle (Girvan ve Newman 2002) önerilen topluluk/modül tespiti yöntemleri çeşitli kategorilerde incelenebilir. Örneğin, *Wu* ve *Pan* tarafından sunulan makalede (Wu ve Pan 2015), genellikle çoğu topluluk algılama yöntemi sezgisel ve optimizasyon temelli olmak üzere iki ayrı kategoride incelenmiştir. Sezgisel temelli yöntemler genel amaç fonksiyonlarının optimize edilmesi yerine bazı sezgisel gözlemlere dayanan kurallar ile ağlardan anlamlı yapıların tespit edilmesi mantığına

dayanır. Sezgisel yöntemlerle elde edilen alt-kümelerin sistematik olarak eğilimli (*biased*) olabileceği gerçeğinin göz önünde bulundurulması gerekmektedir (Leskovec ve ark 2010). Çünkü bu tür yöntemler belli ağlarda özellikle çok iyi ya da çok kötü ağ toplulukları elde etme eğilimindedirler. Bu yöntemlerin genel olarak birçok ağda iyi sonuçları elde etmede başarısız oldukları bilinir. Bu yüzden bu yöntemlerle elde edilen ağ modüllerinin uygun çözümleri temsil edip etmediğinin iyi analiz edilmesi gerekmektedir. Bunun aksine optimizasyon temelli yöntemler ise "*modularity*" (Newman 2004 (a)) veya "*global density*" (Zaremotlagh ve ark 2016) gibi amaç fonksiyonlarının maksimize ya da minimize edilmesi mantığıyla en uygun ağ yapılarının ortaya çıkarılmasına çalışırlar. Bu tür yöntemlerin amaç fonksiyonlarının özelliklerine göre bazen tutarsız sonuçlar üretmesi veya en iyi çözümün aranması sırasında yerel maksimuma ya da minimuma takılma gibi problemlerin dikkatlice çözümlenmesiyle genel en iyi çözüme (*global optimum*) sahip modül yapılarının tespiti mümkün hale gelebilmektedir. Bu güne kadar çeşitli klasik sezgisel yöntemlerle (Pothen ve ark 1990, Shi ve Malik 1997, Girvan ve Newman 2002, Clauset ve ark 2004, Radicchi ve ark 2004, Newman ve Girvan 2004 (b), White ve Smyth 2005, Rosvall ve Bergstrom 2007, Blondel ve ark 2008, Ronhovde ve Nussinov 2009) ve optimizasyon temelli birçok algoritma (von Luxburg 2007, Pizzuti 2008, Rosvall ve Bergstrom 2008, Shi ve ark 2009, Shi ve ark 2010, Gong ve ark 2011, Amiri ve ark 2013, Atay ve Kodaz 2015, Cao ve ark 2015, Zhou ve ark 2016, Li ve ark 2016 (c), Babers ve Hassanien 2017, Li ve ark 2017, Zalik KR ve Zalik B 2017, Zheng ve ark 2017, Zhou ve ark 2017, Atay ve ark 2017 (a), Atay ve ark 2017 (b), Atay ve ark 2017 (c), Said ve ark 2018) ile bahsi geçen problemlere benzer birçok kısıtın çözümü için çeşitli yaklaşımlar önerilmiştir. Sezgisel veya klasik yöntemlerin çoğunluğu büyük ve karmaşık ağlar için sınırlı bir kullanım potansiyeline sahiptir. Bunun yanında yüksek zaman karmaşıklığı ile birlikte uygulanacak ağa veya tespit edilmek istenen alt yapılara ait hiçbir ön bilginin olmaması gibi kısıtlamalardan (Li ve ark 2015) dolayı klasik matematiksel veya sezgisel yöntemler yerine son zamanlarda genellikle optimizasyon temelli algoritmalar önerilmektedir.

Tez çalışmasında ele alınan konulardan ilki ağ modüllerinin tespiti problemidir. Bununla ilgili yukarıda özet tanımlamalar yapılmış olup; bu konunun literatürdeki önemi vurgulanmıştır. Problemle ilgili genel tanımlamalar, seçilen uygunluk/amaç fonksiyonları, küme değerlendirme ölçütleri gibi konular tez çalışmasının içeriğinde sunulmuştur. Bu konuların tezdeki sunumuyla ilgili genel açıklamalar sonraki başlık

altında verilmiştir. Ağlardaki modül/topluluk yapılarının ortaya çıkarılması ile ilgili tez çalışmasında probleme adapte edilen sekiz adet metasezgisel optimizasyon algoritması önerilmiştir. Bunlardan ilki, farklı yöntemlerin bazı uygun özelliklerini barındıran ve *Üç Fazlı Hibrit Yaklaşım—3pHybrid* olarak isimlendirilen algoritmadır. Bunlardan bazıları bilinen mevcut algoritmaların orijinal versiyonlarıdır. Son olarak diğer algoritmalar bazı mevcut yöntemlerin probleme özgü iyileştirilmiş veya geliştirilmiş halini ifade ederler. Böylece önerilen çeşitli metasezgisel algoritmalarla ağ modüllerinin tespiti probleminin çözülmeye çalışılması ve elde edilen sonuçlar ışığında bu algoritmaların karşılaştırmalı analizinin yapılması amaçlanmıştır. Daha sonra yapılan deneyler sonucunda, en başarılı sonuçlara ulaşan algoritma ile bu problemde uygunluk fonksiyonları olarak sunulmuş olan 10 adet amaç/kalite fonksiyonu arasından en iyi sonuçların elde edilmesini sağlayan fonksiyon belirlenmiştir. Deneylerde iki farklı modelle rastsal olarak üretilen yapay ağlarla birlikte biyolojik, medikal, radar, sosyal ve gen/protein etkileşimleri gibi gerçek dünya ağları kullanılmıştır. Tez çalışmasında odaklanılan ilk konu ile amaç ağ modül tespitinde hem etkili bir metasezgisel algoritmanın önerilmesi hem de literatürde bilinen amaç fonksiyonlarından en uygun olanının belirlenmesidir.

Doktora tez çalışmasının bir diğer önemli konusu klinik verilere göre temin edilen biyolojik ağların ve hasta bilgilerinin birlikte analiz edilmesiyle ilgili kanser türlerinde bazı çıkarımların yapılmasıdır. Bunun için sonraki deneylerde sağkalım parametresiyle maksimum ilişkili olan genlerin ortaya çıkarılması amaçlanmıştır. Buna benzer bir problemle ilgili *Hansen* ve *Vandin*, mutasyonlu genlerin bazı kanser türlerine etkisi üzerine bir çalışma yapmışlardır (Hansen ve Vandin 2016). Bu tez çalışmasında mutasyonlar yerine kopya sayısı değişikliklerine (*copy number alterations—CNAs*) sahip genlerin tespitine odaklanılmıştır. Bu amaçla beş farklı kanser türü ile ilgili klinik kanser veri setleri üzerinde bazı testler gerçekleştirilmiştir. Ayrıca bu klinik verilerle ilgili biyolojik ağları temsil eden gen-gen etkileşim ağlarının uygun modül yapıları ortaya çıkarılmış ve bu modüllerde belli kriterlere göre seçilen genlerin ilgili hastalıklarla ilişkileri analiz edilmiştir. Böylece hem ağ modül tespiti problemi hem de kanser verilerinde sağkalım ile yoğun ilişkili gen kümelerinin tespiti problemi ele alınmış ve elde edilen sonuçlar bazı web-modül araçları yardımıyla görselleştirilmiş ve ayrıntılı olarak analiz edilmiştir. Elde edilen tüm sonuçlar hem EKLER bölümündeki tablo ve grafiklerde hem de bu tezle birlikte verilen farklı dosyalarda sunulmuştur. Tez çalışmasında odaklanılan ikinci problem ile hem tıbbi verilerin hem de biyolojik ağların birlikte analiz edilmesi sonucu elde edilen bazı önemli genlerin hastalıklarla ilişkilerinin

değerlendirilmesi amaçlanmıştır. Burada, "Gene set to Diseases – GS2D" web analiz aracından yararlanılmıştır. Bu işlemler, kaydedilen test sonuçlarının biyolojik perspektif açısından incelenmesi ve sonraki çalışmalara öncülük etmesi açısından önemlidir.

1.2. Tezin Organizasyonu

Bu çalışma beş ana bölümden oluşmaktadır. İlk bölüm "*giriş*" bölümüdür ve burada tezle ilgili özet bilgiler, çizge teorisi, biyolojik ağlar hakkında kısa tanımlamalar ve detaylı kaynak araştırması sunulmuştur. İkinci bölümde ele alınan problemlerle ilgili gerekli tanımlamalar yapılmış olup; bu bölüm üç alt-bölümden oluşmaktadır. İlk alt-bölüm ağ modüllerinin ortaya çıkarılması problemi hakkındadır. Burada sırasıyla; ağ modülleri ve tanımlamalar, problem tanımı ve karmaşıklık, amaç fonksiyonları ve değerlendirme ölçütleri başlıkları altında açıklamalar yapılmıştır. İlk alt-başlıkta ağ modüllerinin/topluluklarının tanımlanması yapılarak bu problemle ilgili tezde kullanılan temel terimler hakkında kısa bilgiler sunulmuştur. Sonraki iki alt-başlıkta ise literatürde kullanılan 10 temel amaç fonksiyonu ile 6 farklı değerlendirme ölçütünün genel tanımlamaları verilmiştir. İkinci bölümün sonraki alt-bölümünde sağkalım parametresi ile yoğun ilişkili alt-ağların tespiti problemi hakkında tanımlayıcı bilgiler sunulmuştur. İkinci bölümün son alt-bölümünde ise hem ilk problem hem de ikinci problemin birden ele alındığı üçüncü bir problem hakkında gerekli açıklamalar yapılmıştır.

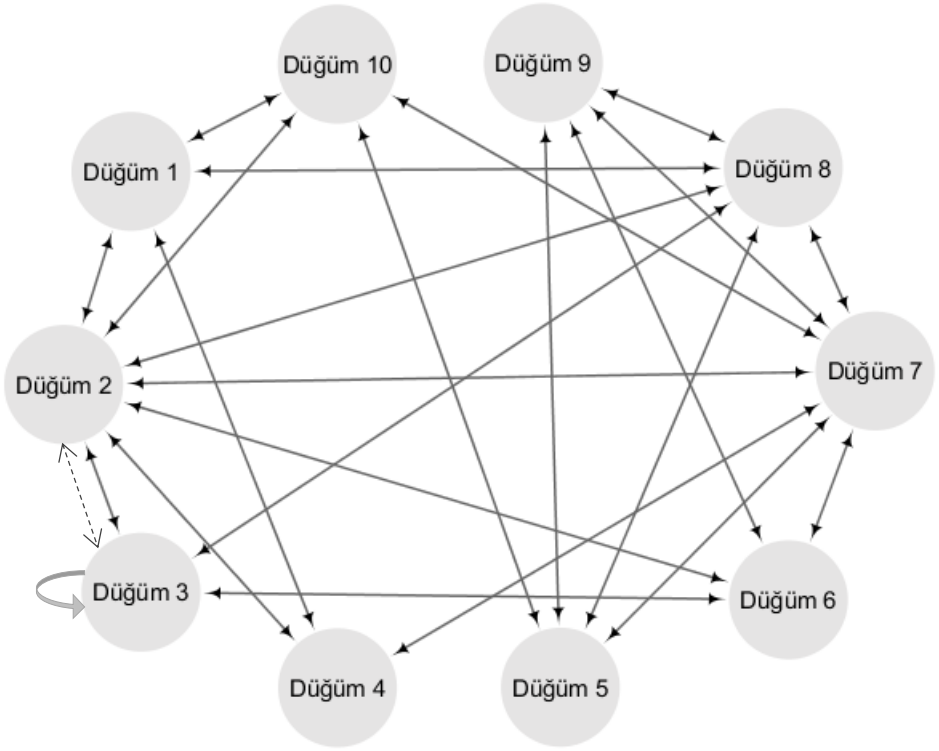
Üçüncü ana bölüm "*materyal ve yöntem*" diye isimlendirilmiştir. Burada ilk iki alt-bölümde sırasıyla; ağ modüllerinin tespiti problemindeki testlerde kullanılan gerçek dünya ağları ile rastsal olarak üretilen yapay ağlar hakkında temel bilgiler sunulmuştur. Bunlar ağların düğüm ve bağlantı sayıları, gerçek dünya ağlarının türleri ve kaynakları ile kesin referanslı ağlarda bilinen modül sayıları gibi tanımlayıcı bilgilerdir. Bu ana bölümün üçüncü alt-bölümünde tezin ikinci problemiyle ilgili testlerde kullanılan klinik hasta bilgileri verilmiştir. Dördüncü alt-bölümde ağ modüllerinin başarılı bir şekilde tespiti amacıyla önerilen sekiz farklı yöntemin çalışma mekanizmaları kendi alt-başlıklarında sunulmuştur. Bu bölümün son kısmı ise "*kanserle ilişkili CNA'lı gen gruplarının tespitinde önerilen yaklaşımlar*" olarak isimlendirilen alt-bölümdür. Burada Bölüm 2.2'deki problemin çözümü için önerilen iki farklı yaklaşım hakkında bilgiler sunulmuştur. Yaklaşımlardan ilki *DP* temelli yöntem; ikincisi *GA* temelli yöntemdir. Bu problemle ilgili iki yöntem de ilk kez bu çalışmada sunulmuştur.

"*Deneysel çalışmalar*" olarak isimlendirilen dördüncü ana bölüm iki alt-bölümden oluşur. Bunlardan ilki ağ modüllerinin tespiti problemi ile ilgili tüm deney sonuçlarının verildiği ve ayrıntılı olarak değerlendirmelerinin yapıldığı bölümdür. Bu alt-bölüm kendi içinde iki ayrı alt-başlıkta incelenmiştir. Birinci alt-başlıkta önerilen yöntemlerin karşılaştırmalı analizleri yapılmıştır. Aynı zamanda burada hem yapay hem de gerçek ağlar üzerinde gerçekleştirilen tüm deneylerin sonuçlarının dikkate alındığı istatistiksel analiz sonuçları da verilmiştir. İkinci alt-başlıkta ise bir önceki alt-başlıkta sunulan istatistiksel test sonuçlarına göre en uygun çıktıları sunan algoritma buradaki deneylerde kullanılmıştır. Bu deneyler için seçilen en başarılı algoritmanın uygunluk kriteri olarak belirlenen çeşitli amaç fonksiyonları her bir test ağında ayrı ayrı çalıştırılmıştır. Daha sonra fonksiyonlara göre kaydedilen ağ modüllerinin kaliteleri iyi bilinen ve çoğunlukla tercih edilen küme değerlendirme ölçütlerine göre incelenmiştir. Böylece gerçekleştirilen tüm deneysel çalışmalarla en iyi sonuçlara ulaşan algoritmanın ve en kaliteli ağ modül yapılarını sunan amaç fonksiyonun belirlenmesi amaçlanmıştır. Bu ana bölümün son alt-bölümü kanserde sağkalım süresi ile maksimum ilişkili alt-ağların tespiti için gerçekleştirilen ayrıntılı deneylerle ilgilidir. Aynı zamanda bu bölümde hem birinci hem de ikinci problemin birden dikkate alındığı üçüncü bir problemle ilgili gerçekleştirilen deneylerin sonuçları verilmiştir. Bu problemin analizi için farklı kanser türleri ile ilişkili karmaşık biyolojik ağlarda testler gerçekleştirilmiştir.

Son olarak, "*sonuçlar ve öneriler*" başlığı altında tez çalışmasının genel katkıları anlatılmış olup; deneysel çıktılara göre genel değerlendirmeler ve bazı çıkarımlar yapılmıştır. Ayrıca burada, yapılabilecek sonraki çalışmalar için öneriler sunulmuştur.

1.3. Çizge Teorisi ve Terminolojisi

Genel olarak bir çizge (*graph*), objeleri ifade eden noktaların ve bu noktalar arasındaki bağlantıları gösteren çizgilerin ortak temsil edildiği bir diyagramdır. Şekil 1.1'de 10 adet düğümden oluşan ve *G* ile temsil edilen örnek bir çizge verilmiştir. Çizge teorisi ise çizgeleri inceleyen bir matematik dalıdır. Temeli, 1736 yılında *Leonhard Euler* tarafından sunulan ve *Königsberg*'in yedi köprüsü olarak bilinen popüler bir problemle alakalı çalışmaya dayanmaktadır (Biggs ve ark 1976).



Şekil 1.1. 10 düğümlü örnek bir çizge

1.3.1. Tanımlamalar

Çizge teorisiyle ilgili bu tez çalışmasında ismi geçen bazı terimler sonraki alt başlıklarda verilmiştir (Biggs ve ark 1976, Biggs 1993, Akgüneş 2013, Bollobas 2013).

1.3.1.1. Düğüm

Çizge terminolojisinde çeşitli bağlantıların buluştuğu nokta veya noktalar topluluğuna düğüm (*vertex/node*) denir. Her bir düğüm bir etiket ile isimlendirilir ve objeleri temsil etmek için kullanılır. Düğümler kümesi, *V* veya *V(G)* ile temsil edilsin.

$$V = \{v_i \mid i = 1, 2, 3, \ldots, n\}$$

Burada *n* toplam düğüm sayıdır. Şekil 1.1'deki Düğüm 1'den Düğüm 10'a kadar olan tüm düğümler *v1*, *v2*, *v3*, *v4*, *v5*, *v6, v7*, *v8*, *v9* ve *v10* ile gösterilsin. Bu durumda verilen çizge için *V = {v1, v2, v3, v4, v5, v6, v7, v8, v9, v10}* şeklinde gösterilir.

1.3.1.2. Bağlantı

Bir bağlantı (*link/edge*) iki düğümü birbirine bağlayan bir çizgiyi ifade eder ve temsil edilen ağ yapısında objeler arasındaki ilişkileri gösterir. Şekil 1.1'deki çizgenin tüm ikili (çift yönlü) bağlantılarının listesi *E* veya *E(G)* ile gösterilir. *m* toplam bağlantı sayısını temsil etsin.

$$E = \{e_j \mid j = 1, 2, 3, ..., m\}$$

Bu durumda bağlantıların tekrarsız listesi *E* = {(*v1*, *v2*), (*v1*, *v4*), (*v1*, *v8*), (*v1*, *v10*), (*v2*, *v1*), (*v2*, *v3*), (*v2*, *v4*), (*v2*, *v6*), (*v2*, *v7*), (*v2*, *v8*), (*v2*, *v10*), (*v3*, *v2*), (*v3*, *v3*), (*v3*, *v6*), (*v3*, *v8*), (*v4*, *v1*), (*v4*, *v2*), (*v4*, *v7*), (*v5*, *v7*), (*v5*, *v8*), (*v5*, *v9*), (*v5*, *v10*), (*v6*, *v2*), (*v6*, *v3*), (*v6*, *v7*), (*v6*, *v9*), (*v7*, *v2*), (*v7*, *v4*), (*v7*, *v5*), (*v7*, *v6*), (*v7*, *v8*), (*v7*, *v9*), (*v7*, *v10*), (*v8*, *v1*), (*v8*, *v2*), (*v8*, *v3*), (*v8*, *v5*), (*v8*, *v7*), (*v8*, *v9*), (*v9*, *v5*), (*v9*, *v6*), (*v9*, *v7*), (*v9*, *v8*), (*v10*, *v1*), (*v10*, *v2*), (*v10*, *v5*), (*v10*, *v7*)} elde edilmiş olur. Burada örnek olarak, (*v1*, *v2*) ifadesi birinci ve ikinci düğüm arasındaki bağlantıyı gösterir.

1.3.1.3. Çizge

Bir *G* çizgesi düğümler ve bağlantılar kümelerinin ortak gösterimiyle *G* = (*V*, *E*) şeklinde temsil edilir. Örnek olarak, Şekil 1.1'de temsili bir çizge yapısı verilmiştir.

1.3.1.4. Döngü

Bir döngü (*self-loop*), düğümün başlangıç ve bitiş bağlantısının kendisini işaret etmesi durumunu tanımlanır. Şekil 1.1'de tek bir döngüye sahip Düğüm 3 (*v3*) örnek olarak gösterilmiştir ve bu bağlantı (*v3*, *v3*) şeklinde temsil edilebilir.

1.3.1.5. Paralel bağlantılar

Paralel bağlantılar birden fazla ilişkiye/etkileşime sahip düğümler arasındaki iletişimi göstermek için kullanılırlar. Şekil 1.1'deki *v2* ile *v3* düğümleri arasında paralel bağlantı bulunmaktadır.

1.3.1.6. Basit çizge

Bu tür çizgeler yönsüz, paralel bağlantıya (*double edges*) sahip olmayan ve döngü içermeyen çizgelerdir.

1.3.1.7. Çoklu çizge

Nesneler arasındaki çoklu ilişkilerin basit çizgelerle temsil edilemediği durumlarda tercih edilen bir çizge türüdür. Bu tür çizgeler paralel bağlantılar içermek dışında diğer özellikleriyle basit çizgelere benzerler.

1.3.1.8. Yönsüz çizge

Düğüm bağlantılarının herhangi bir yön belirtmediği çizgelere yönsüz (*undirected*) çizge denir. İki düğüm arasındaki bağlantının yönsüz olması sebebiyle her iki düğüm de hem kaynak hem de hedef olarak düşünülebilir. Bu tür çizgelerdeki bağlantılar çift yönlü ok işareti veya sade düz çizgilerle gösterilirler.

1.3.1.9. Yönlü çizge

Bu tür çizgeler, bağlantıların iki taraftan birine yönelimli (*directed/digraph*) olduğu çizgeleri ifade ederler. Böyle çizgelerdeki bağlantılar, kaynak ve hedef düğümler arasında sadece tek bir yönelime sahip olurlar.

1.3.1.10. Ağırlıksız çizge

Düğümler arasındaki bağlantıların herhangi bir ağırlıkla ya da değerle ifade edilmediği çizgeler ağırlıksız (*unweighted*) çizge türlerindendir. Bu tür çizgelerde ağırlıklar ikili olarak bir veya sıfır olarak tanımlanır. Şekil 1.1'de gösterilen çizge ağırlıksız bir çizge örneğidir.

1.3.1.11. Ağırlıklı çizge

Tüm düğüm bağlantıları belli sayılarla etiketlenmiş olan çizge ağırlıklı (*weighted*) çizge olarak tanımlanır. Bir ağırlıklı çizgede düğümler arasındaki yolu oluşturan bağlantıların ağırlıkları toplamı o yolun uzunluğu verir. Arkadaş ilişkilerinin modellendiği sosyal bir ağdaki düğüm bağlantılarının yakın arkadaşlıkları yüksek sayıda ağırlıklı; daha az ilişkide olan kişilerin temsil ettiği bağlantıların ise düşük sayıda ağırlıklı olması bu tür çizgeler için örnek olarak verilebilir.

1.3.1.12. Sözde çizge

Döngü içeren ve hem yönsüz hem de paralel bağlantıları bulunan çizgeler sözde (*pseudo*) çizgeleri temsil ederler. Bu tür çizgeler basit olmayan yapılardadır. Ayrıca bir çizgenin sözde çizge olabilmesi için gerekli şart, yapısında en az bir tane kendisine bağlı düğümün (döngülü) bulunmasıdır.

1.3.1.13. Bağlı çizge

Bir çizgedeki tüm düğümler arasında en az bir bağlantı varsa bu tür çizgeler bağlı (*connected*) olarak tanımlanır. Şekil 1.1'deki çizge bağlı bir çizge örneği iken; Şekil 1.2'deki çizge bağlı olmayan bir çizgeyi göstermektedir.

1.3.1.14. Kesitleme noktası

Başlangıçta bağlı olan bir çizgeden herhangi bir düğümün çıkarıldığı farz edilsin. Eğer çıkarılan bu düğümden sonra kalan çizge, bağlı bir çizge değilse çıkarılan düğüm kesitleme noktası (*cut point*) olarak tanımlanır.

1.3.1.15. Bağlı bileşen

Bağlı olmayan bir çizgede bağlı alt-grupların mevcut olması durumunda her bir alt-grup bağlı bileşenlere (*connected component*) ayrılabilir. Özetle diğer bileşenlerdeki düğümlerle ilişkisiz ve sadece kendi bileşeni içindeki düğümlerle bağlantıları bulunan çizgedeki her bir ayrık alt-çizge bağlı bileşenleri temsil eder.

1.3.1.16. Topolojik parametreler

Derece: Bir düğümün diğer düğümlerle olan bağlantılarının toplamı düğüm derecesini (*degree*) ifade eder. Düğüm derecesi, *deg(v)* veya k_i ile gösterilir. Burada *i*, düğüm indislerinin gösterimi için kullanılır. Örneğin; Şekil 1.1'deki çizgenin yönsüz, paralel bağlantılar ve döngüler içermeyen özelliklere sahip olduğu düşünülürse bu çizgenin toplam derece sayısı 46 olur. *v1* düğümü 4 bağlantıya sahiptir. Böylece düğüm derecesi *deg(v1)* = 4 ile gösterilir. Derece hesabı için $\deg(v) \leq n - 1 \mid v \in G$ tanımı yapılır.

Ortalama derece: Bu tez çalışmasında ele alınan ağlar yönsüz ve ağırlıksız ağlardır. Bu sebeple ortalama derece (*average degree*), toplam bağlantı sayılarının ikiye bölünmesiyle elde edilir. Ortalama derecenin hesaplanmasında ise her bir bağlantının ait olduğu iki düğümünün ilişkilendirilmesi dikkate alınır ve bağlantı sayılarının iki katı hesaba katılır. Böylece toplam m bağlantıya sahip n adet düğümlü bir ağın ortalama derecesi şu şekilde hesaplanabilir: $(2 \times m/n)$.

Ortalama kümelenme katsayısı: Herhangi bir çizgenin kümelenme eğiliminin derecesi ortalama kümelenme katsayısı (*average cluster coefficient*) ile temsil edilir. Çizgedeki kümelenme oranı üç düğüme sahip bağlı alt-çizgelerin (üçgenler—*triangles*) yoğunluğunun bir ölçütüyle ifade edilir.

Mesafe: İki düğüm arasındaki en kısa yolun uzunluğu mesafeyi (*distance*) verir.

Ağ çapı: Bir çizgede bulunan herhangi iki düğüm arasındaki maksimum mesafe çap (*diameter*) ile ifade edilir.

Ağ bağlantı yoğunluğu: Yoğunluk kavramı bir ağda bulunan gerçek bağlantıların tüm potansiyel bağlantılara oranını gösterir. Buradaki potansiyel bağlantı kavramı, mevcut olup olmadığına bakılmaksızın düğümler arasında olabilecek olası tüm bağlantıları ifade eder. n düğümlü ve m bağlantılı bir ağın bağlantı yoğunluğu şöyle hesaplanır: $(2 \times m/(n^2 - n))$.

Ortalama yol uzunluğu: Ağ topolojisindeki ortalama yol uzunluğu (*average path length*) kavramı bir çizgedeki olası tüm düğüm çiftleri için mevcut en kısa yol boyunca ortalama adım sayısını ifade eder.

Merkezilik: Çizge teorisi ve ağ analizinde çeşitli ölçümlerle bir çizgedeki en önemli düğümler merkezilik (*centrality*) kavramı ile tanımlanır. Merkeziliğe ilişkin çeşitli ölçümler bulunmaktadır. Bu ölçümlerden bazıları derece merkeziliği, yakınlık merkeziliği ve arasındalık merkeziliği ölçümleridir.

Arasındalık: Bir düğümün çizgedeki diğer düğümler arasında bulunmasının derecesi arasındalık (*betweenness*) değerini verir. Yüksek arasındalık değeri bu düğümün çizge içinde önemli bir bağlantı noktasında bulunduğunu gösterir.

Yakınlık: Herhangi bir düğümün diğer düğümlerle mevcut olan ve doğrudan ya da dolaylı olarak ölçülebilen derecesi, yakınlığı (*closeness*) ifade eder. Bir düğüm için yakınlık ölçütü bu düğümün çizgedeki diğer düğümlere olan en kısa uzaklık ölçümlerinin terslerinin belirlenerek toplanmasıyla hesaplanır.

1.3.1.17. Uç düğüm

Düğüm derecesi bir olan düğümler uç düğümler olarak tanımlanırlar. Bu tür düğümlerin tek bir komşu düğümleri mevcuttur.

1.3.1.18. İzole düğüm

Derecesi sıfır olan düğümlere izole düğümler denir. Bu düğümler diğer tüm düğümlerden ayrık bir durumdadır ve bunlarla hiçbir bağlantısı bulunmamaktadır.

1.3.1.19. Komşuluk matrisi

Düğüm ilişkilerinin 0 ve 1'lerle gösterildiği komşuluk matrisi düğümlerin aktif var olan bağlantılarını gösterir ve *adj* ile temsil edilir. Bu matris *nxn* boyutludur. Burada *n*, toplam düğüm sayısını gösterir. Komşuluk matrisi Denklem 1'e göre oluşturulur.

$$adj = \begin{cases} 1 & \text{eğer } i.\text{ve } j.\text{düğümler bağlantılı ise,} \\ 0 & \text{aksi durumda.} \end{cases} \quad (1)$$

Her bir düğümün derecesi komşuluk matrisinde yararlanarak Denklem 2'ye göre hesaplanır. Burada *i* ve *j* düğümlerin indislerini gösterir. Çizelge 1.1'de, Şekil 1.1'deki çizgenin komşuluk matrisi verilmiştir.

$$k_i = \sum_j adj_{(i,j)} \tag{2}$$

Çizelge 1.1. Şekil 1.1'deki çizgenin komşuluk matrisi

adj	*v1*	*v2*	*v3*	*v4*	*v5*	*v6*	*v7*	*v8*	*v9*	*v10*
v1	*0*	*1*	*0*	*1*	*0*	*0*	*0*	*1*	*0*	*1*
v2	*1*	*0*	*1*	*1*	*0*	*1*	*1*	*1*	*0*	*1*
v3	*0*	*1*	*0*	*0*	*0*	*1*	*0*	*1*	*0*	*0*
v4	*1*	*1*	*0*	*0*	*0*	*0*	*1*	*0*	*0*	*0*
v5	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*1*	*1*	*1*
v6	*0*	*1*	*1*	*0*	*0*	*0*	*1*	*0*	*1*	*0*
v7	*0*	*1*	*0*	*1*	*1*	*1*	*0*	*1*	*1*	*1*
v8	*1*	*1*	*1*	*0*	*1*	*0*	*1*	*0*	*1*	*0*
v9	*0*	*0*	*0*	*0*	*1*	*1*	*1*	*1*	*0*	*0*
v10	*1*	*1*	*0*	*0*	*1*	*0*	*1*	*0*	*0*	*0*

Burada, döngülü ve paralel bağlantılar ihmal edilmiştir.

1.3.1.20. Komşuluk listesi

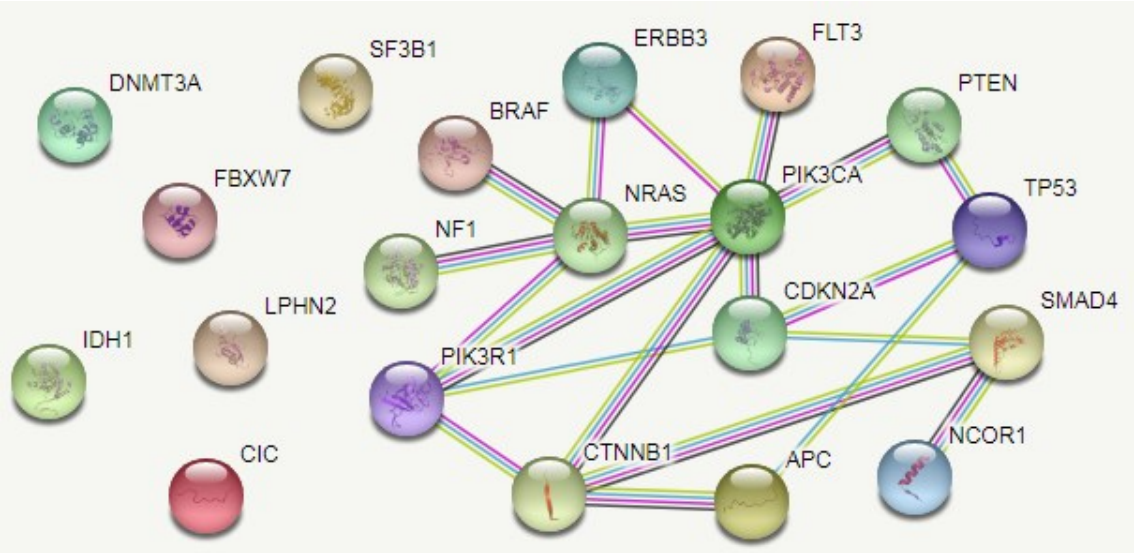
Bir çizgedeki düğümlerin bağlı olduğu komşu düğümlerinin tutulduğu liste komşuluk listesi olarak tanımlanır. Bu liste *nxn*'lik bir matris gerektirmediği için bellek yönetimi açısından daha avantajlıdır. Şekil 1.1'de gösterilen çizgedeki tüm düğümlerin komşuluk listeleri Çizelge 1.2'de sunulmuştur. Bu liste $adjL$ ile temsil edilmiştir.

Çizelge 1.2. Şekil 1.1'deki çizgenin $adjL$ komşuluk listesi

v1	→	*v2, v4, v8, v10*
v2	→	*v1, v3, v4, v6, v7, v8, v10*
v3	→	*v2, v6, v8*
v4	→	*v1, v2, v7*
v5	→	*v7, v8, v9, v10*
v6	→	*v2, v3, v7, v9*
v7	→	*v2, v4, v5, v6, v8, v9, v10*
v8	→	*v1, v2, v3, v5, v7, v9*
v9	→	*v5, v6, v7, v8*
v10	→	*v1, v2, v5, v7*

1.3.2. Çizge modelleme

Örnek olarak alınan bir gerçek dünya sistemini temsil eden ve nesneler arasında var olan çeşitli ilişkileri bir çizge yardımıyla gösteren *N* ağı, *adj* komşuluk matrisine sahip $G = (V, E)$ çizgesi ile modellenebilir. Örneğin *STRING* (Szklarczyk ve ark 2014, Szklarczyk ve ark 2016) veri tabanından temin edilen 20 adet en sık mutasyona uğramış insan kanser geni ve bunlar arasındaki bağlantılar Şekil 1.2'de gösterilmiştir (STRING 2017). Bunlar; *APC, BRAF, CDKN2A, CIC, CTNNB1, DNMT3A, ERBB3, FBXW7, FLT3, IDH1, LPHN2, NCOR1, NF1, NRAS, PIK3CA, PIK3R1, PTEN, SF3B1, SMAD4, TP53* isimli genler/proteinlerdir. Proteinler veya genler arası etkileşimler anlamlı olabilir ve bunların ayrıntılı analizi gereklidir. Ağlardaki ortak bağlantılar iki proteinin paylaştığı fonksiyonel etkileşimi gösterir. Buradaki bağlantının fiziksel olması zorunluğu bulunmamaktadır. Aradaki etkileşim sadece fonksiyonel düzeyde de olabilir. Örneğin, Şekil 1.2'de en fazla bağlantıya sahip iki düğüm *PIK3CA* ile *NRAS*'dir ve aralarındaki fonksiyonel veya moleküler etkileşim çizge ile modellenmiştir. Şekil 1.2'deki örneğe benzer şekilde çeşitli hastalıklarda ortak etkisi olan mutasyona uğramış gen/protein gruplarının tespiti için bu etkileşimlerin çizgelerle modellenmesi ve analizlerinin gerçekleştirilmesi önemlidir.



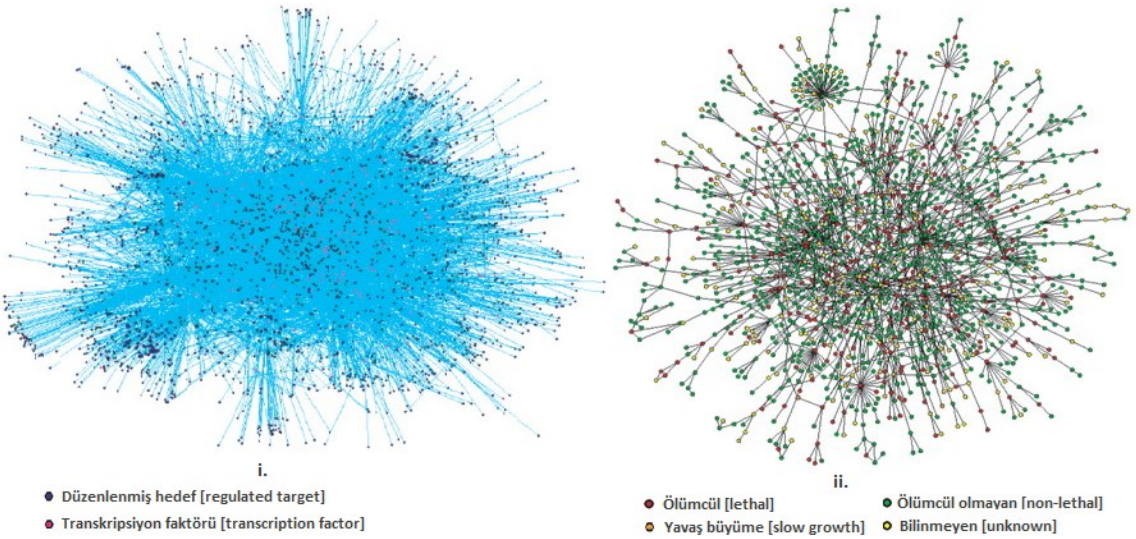
Şekil 1.2. En sık mutasyona uğramış 20 adet insan kanser geni (STRING 2017)

Çizge teorisi çeşitli alanlarda doğrudan ya da dolaylı olarak kullanılmaktadır. Bununla ilgili Bilgisayar Bilimleri ve Mühendisliğinde çizgeleri temel alan *Kruskal*, *Prim* ve *Dijkstra* gibi çizge algoritmaları, bilgisayar oyunları, ağlardaki bilgisayarların bağlantılarının tasarımı, ağ ve veri paylaşımı, veri tabanında ilişkilerin temsili gösterimi, bilimsel makalelerdeki atıfların çizgelerle modellenmesi gibi çalışmalar; Teknolojik alanlarda telefon veya e-mail kayıtlarının çizgeler ile tasarımı ve incelenmesi, uçak rotalarının belirlenmesi, trafik akışının hızlandırılması için yolların düğüm/bağlantı ikilisi ile gösterimi; Elektrik Mühendisliğinde devre tasarımı ve topolojiler konuları; Biyolojik sistemlerdeki gen/protein etkileşimleri, organizmaların DNA yapıları veya moleküllerinin gösterimi, sinir ağlarının modellenmesi, ekosistemde birbirleriyle beslenen canlılar arasındaki ilişkisel ağlar, birbirleriyle sosyal iletişim kurabilen yunuslar gibi canlıların davranışlarının incelenmesi; Sosyal alanlarda insanlar arası ilişkilerin analizi, kişi eğilimlerinin ortaya çıkarılması, terörist ataklara ve suça meyilli olan şahısların incelenmesi gibi önemli konular veya elektrik-su-doğalgaz gibi kaynakların dağıtımı, posta ve kargo sistemlerinin modellenmesi, büyük ve kurumsal firmalardaki personellerin görev ve iş ilişkilerinin çizgelerle analizi gibi genel konularla ilgili ağ yapıları uygulamaları bu alanlarda verilebilecek örneklerden bazılarıdır. Tüm bahsi geçen uygulama örnekleri incelendiğinde, çizge teorisinin hayatın hemen hemen her noktasında önemli bir yere sahip olduğu anlaşılabilir.

1.4. Biyolojik Ağlar

Gerçek biyolojik sistemlerinin modellendiği ağ yapıları çizgelerle temsil edilirler. Bu sistemlerdeki nesneler veya elemanlar arasındaki mevcut etkileşimlerin uygun şekilde gösterimi için çeşitli türlerdeki çizgeler kullanılır. Ekolojik/yiyecek etkileşim ağları, sinir ağları, biyokimyasal ağlar biyolojik ağlardan bazılarıdır. Biyokimyasal ağlar biyolojik bir hücredeki moleküler seviyede gerçekleşen etkileşim ve kontrol mekanizmalarını temsil ederler. Bu tür ağlardan bazıları metabolik ağlar, protein etkileşim ağları ve gen düzenleyici ağlardır. Bu tez çalışmasında sosyal, teknolojik, bilimsel ve mühendislik gibi çeşitli alanlardan temin edilen ağların yanında klinik kanser verilerinden temin edilen genlerin birbirleriyle olan etkileşimlerini temsil eden gen-gen etkileşim ağları kullanılmıştır. Bu tür biyokimyasal etkileşim ağlarının analizinde amaç, seçilen kanser türüne en fazla etki ettiği düşünülen gen/protein gruplarının tespit edilmesidir. Aşağıda moleküler etkileşim ağlarından bazıları hakkında kısa bilgiler verilmiştir.

Gen düzenleyici ağlar (GRNs): Transkripsiyon faktörleri (*TFs*) ile düzenleyici elemanlar arasında bulunan fiziksel ve/veya fonksiyonel etkileşimleri içeren gen düzenleyici ağları, canlının gelişimi ve fizyolojisinde kritik fonksiyonlara sahiptir. Örneğin, uygun olmayan gen düzenlemesi insanlardaki çeşitli hastalıkların temelini oluşturur (Bass ve ark 2015). Düzenleyici ağlar, hücrelerde mevcut olan gen ekspresyonunun kontrolü ile ilgili bilgiler içerirler. Bu tür ağlar proteinlerin ve diğer biyomoleküllerin gen ekspresyonuna dahil olma şeklini modellemek ve bu sürecin farklı aşamalarında gerçekleşen zincirleme olaylar serisini simüle etmek amacıyla yönlü çizgelerin gösterimiyle modellenebilirler. Gen düzenleyici ağlarındaki düğümler proteinleri veya bunlara eşdeğer olarak görülen genleri kodlayan başka genleri temsil ederler. Örneğin *X* geninden *Y* genine yönlü bir bağlantı, *X*'in *Y* geninin ekspresyonunu düzenlediğini gösterir. Şekil 1.3–i'de, *Saccharomyces Cerevisiae* isimli maya türüne ait iki farklı veri setinin görselleştirilmiş transkripsiyon faktörü bağlayıcı (gen düzenleyici) ağı verilmiştir (Zhu ve ark 2007).



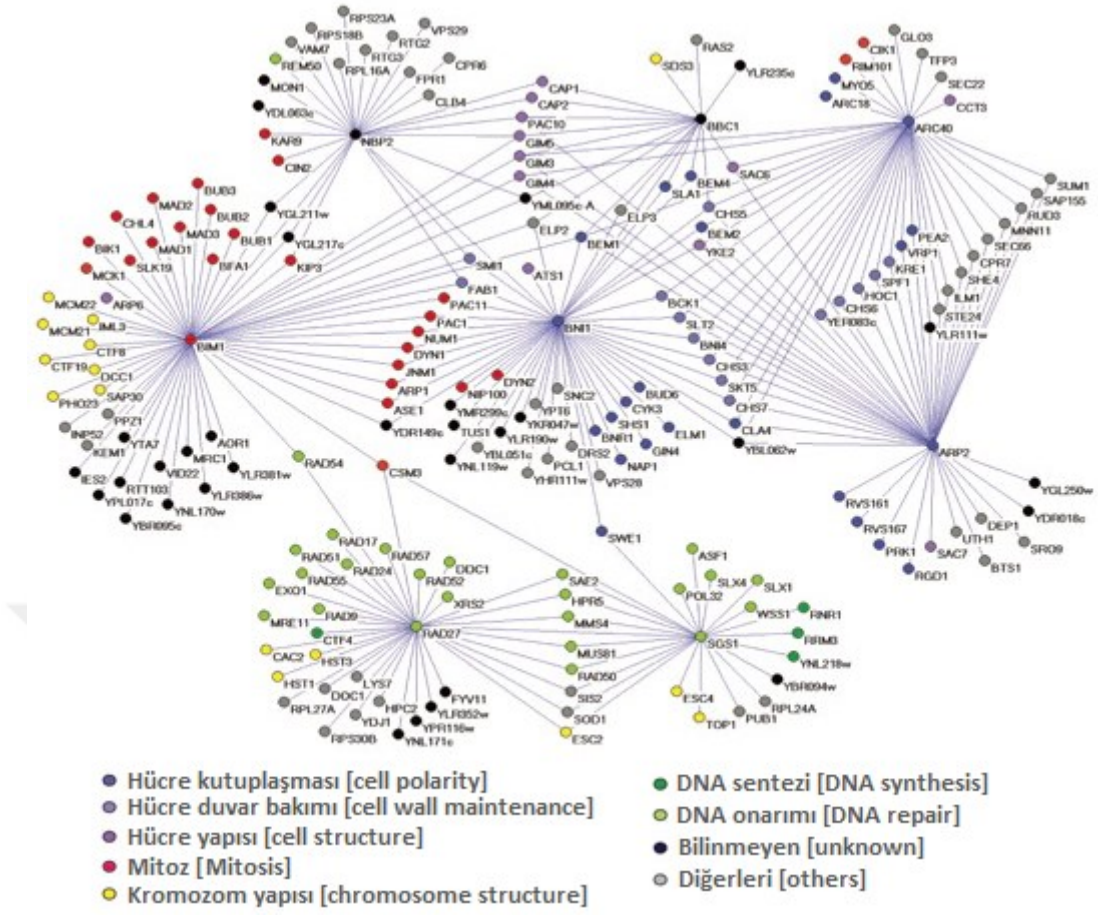
Şekil 1.3. i. Bir maya (yeast) canlısı için örnek olarak, transkripsiyon faktörü bağlayıcı ağı, **ii.** Çeşitli canlılara ait *PPI* ağı (Zhu ve ark 2007)

Protein-protein etkileşim ağları (protein-protein interactions—PPIs): Tüm canlılardaki proteinler hücreler için en önemli moleküler gruplardandır ve canlıların yaşamsal fonksiyonlarıyla doğrudan veya dolaylı olarak ilişkilidirler. Bu yapılar birbirleriyle ve diğer moleküllerle kimyasal veya kimyasal olmayan çeşitli etkileşimlere girerler. Örneğin, metabolik süreçlerin katalizörü, sinyal maddeleri (hormonlar), yapısal veya mekanik materyal (saç) veya diğer maddeler için oksijen gibi ihtiyaçları yüklenen taşıyıcılar olarak enzimler adı altında görev yaparlar (Bachmaier 2013). Proteinler, esasen yirmi farklı amino asitin peptit bağlarıyla birbirlerine bağlandığı bir takım baz ünitenin bir araya getirilmesiyle oluşan uzun zincirli molekül topluluklarıdır. Bunlar farklı şekillere biçimlenerek katlanırlar. Katlanan form çeşitli moleküllerle uygun durumlarda etkileşime girerler. Bu yüzden proteinlerin birbirleriyle veya diğer biyomoleküllerle girdikleri etkileşimler biyokimyasal veya fiziksel etkileşimi kapsayabilir. Ayrıca bazı protein veya gen etkileşimleri de ne kimyasal ne de fiziksel bir özelliğe sahiptir. Bu tür etkileşimler gen/protein gruplarının birlikte etki ettiği fonksiyonel birliktelikleri vurgularlar. Herhangi bir protein etkileşim ağında düğümler proteinleri gösterirken; proteinlerin komşularıyla var olan ilişkiler/etkileşimler de bağlantılarla temsil edilirler. Protein etkileşimleri genellikle yönsüz ve ağırlıksız ağlardır. Bu ağlar bugüne dek elde edilen en fazla veri kümesini temsil eden biyolojik ağ türüdür. Şekil 1.3–ii'de *Saccharomyces Cerevisiae* isimli maya türüne, *Drosophila* isimli sinek türüne, Ev faresine (*Mus Musculus*), *Caenorhabditis Elegans* isimli solucan türüne ve insanlara ait proteinlerin ve bağlantıların bir arada görselleştirildiği etkileşim

ağı verilmiştir (Zhu ve ark 2007). Doktora tez çalışmasında insanlara ait gen/protein etkileşim ağları bazı testlerde kullanılmıştır.

Metabolik ağlar: Metabolizmalar hücrelerdeki besin maddelerini kullanılabilir yapı taşlarına dönüştürdükleri veya yapı taşlarını birleştirerek biyolojik molekülleri oluşturuldukları yaşamsal ve kimyasal değişimlerin bütünü olarak tanımlanabilirler. Genel olarak bu ayrışma ve birleşme süreçleri çeşitli zincirler, yollar/yolaklar veya başarılı kimyasal reaksiyon adımlarını içerirler. Bir metabolik reaksiyon maddelerin veya metabolitlerin genellikle enzimler tarafından katalize edilen diğer maddelere dönüşümü olarak düşünülebilir. Metabolik reaksiyonlar enerji üretimi ve maddelerin sentezi gibi yaşamsal süreçlerinin temelini oluştururlar. Reaksiyonların ilk girdilerinden itibaren sürecin son basamağına kadar gerçekleşen işlemler kompleksi ve bu reaksiyonların tamamı metabolik ağı tanımlar (Bachmaier 2013).

Genetik ve küçük molekül etkileşim ağları: Birbirleriyle işlevsel olarak ilişkili olan gen veya küçük moleküllerin etkileşimlerinin modellendiği çizgeler çeşitli şekillerde sunulmaktadır. Sekiz adet maya geninde sentetik öldürücü etkileşimlerle oluşturulmuş bir maya genetik ağı Şekil 1.4'te sunulmuştur (Tong ve ark 2001, Zhu ve ark 2007). Bu çizgedeki genler, düğümlerle (204 gen); etkileşimler ise bağlantılar (291 etkileşim) ile temsil edilir. Burada, genler hücresel görevlerine uygun olarak renklendirilmiştir (Tong ve ark 2001). Siyah renkte verilen ve hücresel rolleri bilinmeyen genlerin işlevi, benzer bağlantıyı gösteren (*komşu*) genlerin rolleri ile ilişkili olduğu öngörülür. Diğer genlerin işlevleri ise Şekil 1.4'teki etkileşim ağının altında verilmiştir.



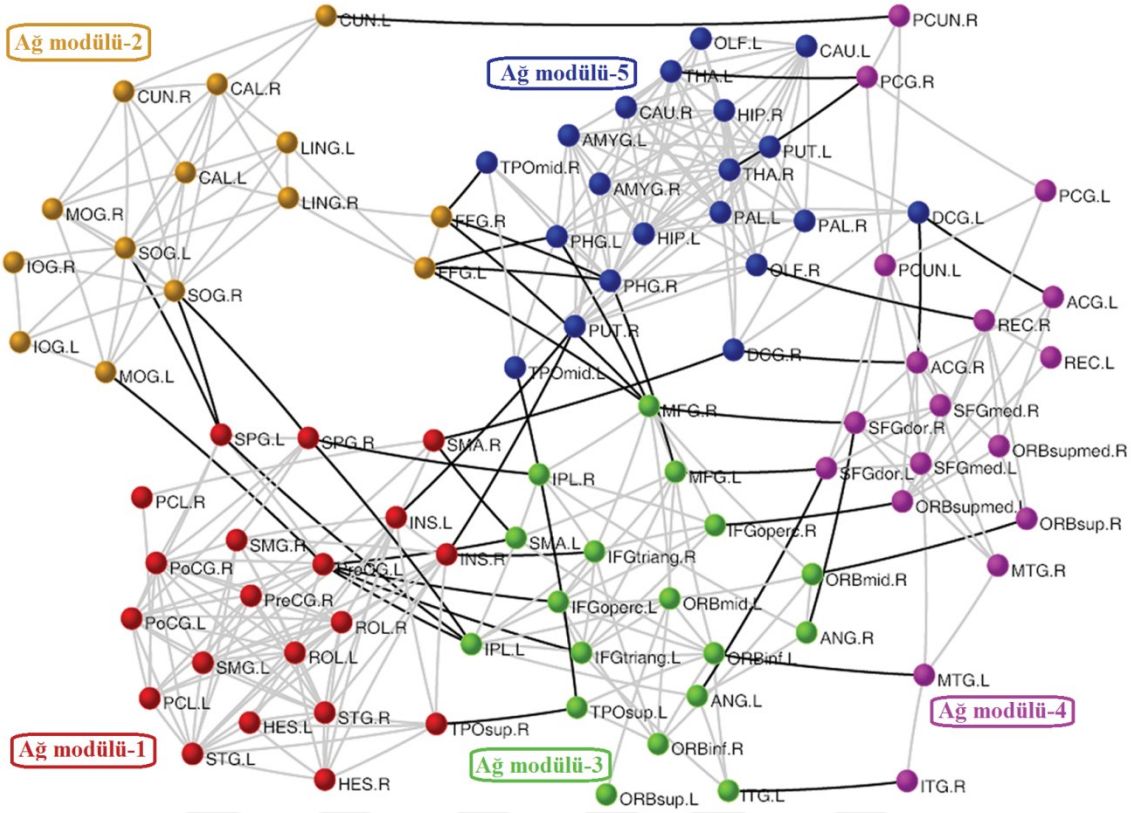
Şekil 1.4. *Saccharomyces Cerevisiae* isimli maya türüne ait genetik etkileşim ağı (Tong ve ark 2001, Zhu ve ark 2007)

1.5. Biyolojik Ağ Motifleri ve Modülleri

Karmaşık ağlarda bulunan istatistiksel olarak anlamlı ve bu ağda bulunma sayısı rastsal olarak üretilen herhangi bir ağdaki bulunma sayısının ortalamasından büyük olan alt-çizgeler, *ağ motifleri* olarak isimlendirilirler. Bu motifler ilk kez *Milo ve diğerleri* (Milo ve ark 2002) tarafından çizgelerdeki temel yapı taşları ve karmaşık ağ desenleri olarak tanıtıldı. Örneğin, bir gen düzenleyici ağındaki biyolojik fonksiyonların temsili olarak motifler gen düzenleyici ilişkilerin temsilinde modellenmiştir. Aynı zamanda bu motifler ilgili çalışmada (Milo ve ark 2002) nöronlar, yiyecek/gıda ağları, elektronik devreler ve dünya çapında web (*world wide web*) ağlarında da ilgili fonksiyonel yapıların tanımlanması için kullanılmıştır.

Sosyal ağlar, bilgisayar ağları ve metabolik/düzenleyici etkileşimler gibi çeşitli biyolojik ağlar da dahil olmak üzere sosyal ve fen bilimlerindeki birçok gerçek dünya ağı, doğal bir şekilde topluluklara veya modüllere bölünmüş olarak bulunur (Newman

2006 (a)). Ağ perspektifinden bakıldığında, ağ modülleri ya da toplulukları üyeleri kendi aralarında güçlü ilişkilerle birbirlerine bağlı olan yoğun alt-çizgelere karşılık gelir. Buradaki alt-çizgeler (*subgraphs*); topluluklar—*communities* (Fortunato 2010), ağ modülleri—*network modules* (Lecca ve Re 2015), topluluk yapıları—*community structures* (Newman 2004 (a)), yapısal alt birimler—*structural subunits* (Palla ve ark 2005), kümeler—*clusters* (Leskovec ve ark 2010), kompleksler—*complexes* (Li ve Chen 2013 (b)), gruplar—*groups* (Subelj ve Bajec 2014) ve bağlı bileşenler—*connected components* (Duch ve Arenas 2005) gibi ifadelerle temsil edilmektedir. Verilen bu terimler, bazı özellikleriyle birbirlerinden farklı olsalar da genelde çeşitli çalışmalarda ağ topluluklarının temsilinde kullanılmıştır. Bu çalışmada çizgelerde bulunan alt-yapılar, çoğunlukla ağ modülleri ismiyle temsil edilmişlerdir. En genel tanımıyla bu ağ modülleri çizgelerdeki bağlantı sayısı, ağırlıkların yoğunluğu, düğüm sayısı, komşuluk ilişkisi ve bunun gibi birtakım topolojik özelliklere göre elde edilmiş düğüm kümelerini ifade ederler. Bu alt-çizgeler özellikle gerçek dünya ağlarında objeler arası yapısal veya fonksiyonel ilişkilerin belirlemesinde ve ağın kimliğinin belirlenmesinde çok önemli bilgiler sağlamaktadırlar. Ağ motifleri genelde üç, dört ve beş düğüme sahip alt-çizgeleri temsil ederler. Ancak ağ modülleri motiflerden çok daha büyük yapılardır ve genellikle 100'den fazla düğüme sahiptirler (Milo ve ark 2002, Ma ve Gao 2012). Ağ modülleri tanımının anlaşılabilmesi için insan beynindeki fonksiyonel modülleri içeren örnek bir çizge Şekil 1.5'te verilmiştir (He ve ark 2009). Burada *Ağ modülü-1*, *Ağ modülü-2*, *Ağ modülü-3*, *Ağ modülü-4* ve *Ağ modülü-5* isimli düğüm toplulukları farklı renklerde sunulmuştur. Her bir düğüm tüm ağdaki bilgi akışını kontrol altında tutmada önemli olan beyin bölgelerini temsil etmektedir. Kendiliğinden gerçekleşen beyin fonksiyonlarının bölgeler arası ilişkileri ise bağlantılar ile sunulmuştur. Modüller içi bağlantılar açık gri renkte çizgilerle ve modüller arası bağlantılar ise koyu renk çizgilerle gösterilmiştir. Birinci modül 20 adet bölge (düğüm) içermekte ve bu modül çoğunlukla motor, duyu ve işitsel işlevlerle ilişkili bölgelere sahiptir. Diğer modüller de içerdikleri bölgelerin (düğümlerin) işlevsel etkileşimleriyle modellenmiştir ve ilgili çalışmada (He ve ark 2009) bu modüllerin ayrıntılı özelliklerine ulaşılabilir. Her bir modülün üyeleri fonksiyonel ortaklıklarına göre ilişkisel olarak Şekil 1.5'teki 5 adet modülden birinde yer alırlar. Böylece bu beyin bölgelerinin kendi modüllerindeki bölgelerle (doğrudan ya da dolaylı komşu düğümler) yoğun etkileşimlerde (*gri renkli bağlantılar*); diğer modüllerdeki bölgelerle daha seyrek etkileşimlerde (*koyu renkli bağlantılar*) bulundukları anlaşılabilir.



Şekil 1.5. İnsan beyninin fonksiyonel ağının çizge ile oluşturulmuş modüler yapısı (He ve ark 2009)

1.6. Kaynak Araştırması

Bu bölümde, tez çalışmasında ele alınan konularla ilgili literatür araştırması yapılmıştır. Gerçek dünya sistemlerinin modellendiği karmaşık ağ yapıları, içerisinde ortaya çıkarılabilecek önemli bilgiler barındırmaktadırlar. Bu bilgilerin farklı ağ analiz teknikleriyle keşfedilebilmesi için son zamanlarda çeşitli çalışmalar yapılmakta ve birçok yöntem önerilmektedir. Bu çalışmanın temelini oluşturan konulardan ilki olan ağ modüllerinin veya topluluklarının ortaya çıkarılması problemi özellikle son yıllarda oldukça dikkat çeken bir konu haline gelmiştir. Bu konuda son zamanlarda birçok çalışma yapılmış olup halen de yeni araştırmalar ve çalışmalar yapılmaktadır. Bunlardan dikkate değer olan çalışmaların özet bilgileri literatüre sunulma tarihlerine göre bu bölümde verilmektedir.

Ağlardaki modüllerle veya topluluk yapılarıyla ilgili çalışmaların geçmişi çizge teorisinin çeşitli bilim dallarında fark edilir bir şekilde kullanıldığı tarihlere dayanmaktadır. Bu konudaki çalışmaların hızla yaygınlaşması ise *2002* yılında *Girvan*

ve *Newman* tarafından sunulan *"Community structure in social and biological networks"* isimli makale ile başlamıştır (Girvan ve Newman 2002). Bu makalede, topluluk yapısının çeşitli tanımlamaları ve bu yapıların tespitinde önerilen klasik yaklaşımlar özetlenmiştir. Aynı zamanda, toplulukların tespiti için çizge teorisinde merkezilik (*centrality*) özelliğini referans alan yeni bir yöntem sunulmuştur. Bu yöntem *GN* (*Girvan* ve *Newman*) olarak isimlendirilmiştir (Newman 2004 (a)). *GN* algoritması, test verisi olarak alınan giriş ağından bağlı bileşenlerin elde edilmesine kadar düğümler arasındaki bağlantıları adım adım kaldırarak nihai toplulukları tespit eden ayırıcı (*divisive*) bir kümeleme algoritmasıdır. Algoritmanın dört temel adımı bulunmaktadır. Bunlar: (*a*) Ağdaki tüm bağlantılar için arasındalık (*betweenness*) değeri hesaplanır, (*b*) En yüksek arasındalık değerine sahip bağlantı veya bağlantılar ağdan silinir, (*c*) Seçilen bağlantıların ağdan silinmesinden sonra mevcut tüm bağlantılar için arasındalık değerleri yeniden hesaplanır, (*d*) Bağlı bileşenlerin elde edilmesine kadarki süreçte ağda hiçbir bağlantı kalmayana dek *b* ve *c* adımları tekrar edilir. Yazarlar önerdikleri algoritmayı rastsal olarak üretilmiş ağlarda, futbol ağında (*American College Football*) (Girvan ve Newman 2002), bilinen sosyal bir ağda (*Zachary's Karate Club*) (Zachary 1977) ve diğer çeşitli ağlarda test etmiş ve elde edilen sonuçları ilgili makalede ayrıntılarıyla sunmuşlardır (Girvan ve Newman 2002).

Newman ve *Girvan* tarafından önerilen modülerlik—*modularity* (Newman ve Girvan 2004 (b)) fonksiyonu ağdaki toplulukların/modüllerin tespitinde en fazla bilinen ve diğer çalışmalara kaynak teşkil eden bir amaç fonksiyonudur. Bu fonksiyon orijinal *GN* algoritmasında bir sonlandırma kriteri olarak sunulmuştur (Newman ve Girvan 2004 (b)). Çeşitli kümeleme yöntemleri, aç gözlü yaklaşımlar ve optimizasyon teknikleri gibi birçok yöntem bu fonksiyonun en iyilenmesi üzerine modellenmiştir. Bu şekilde maksimum modülerlik skoru ağdaki en uygun modülerliğin sağlanması için gerekli şart haline gelmiştir. İlgili çalışmada modülerlik fonksiyonu en temel hali ile sunulmuştur. Ayrıca bu çalışmada yazarlar, önerdikleri çeşitli yöntemlerin başarılarını, yapay ve gerçek dünya ağları üzerinde test ederek göstermişlerdir.

Literatürdeki klasik birçok algoritma büyük ve karmaşık ağlarda, küçük ağlarda elde edilen başarılı kümelemeleri sağlayamamışlardır (Newman 2004 (a)). Örneğin, *GN* algoritması küçük ağlar için uygun alt grupları doğru bir şekilde sağlasa da büyük ağlarda benzer başarıyı gösterememiştir. *Newman* tarafından önerilen diğer bir yeni algoritma (*Fast Newman—FN*) ise yukarıda bahsedilen probleme çözüm önermektedir. Buna ek olarak, *FN* algoritmasının ağdaki modüllerin tespitinde önceki yöntemlerden

neredeyse binlerce kez daha hızlı olduğu ilgili çalışmada (Newman 2004 (a)) vurgulanmıştır. Önerilen algoritmanın test çalışmaları klasik gerçek dünya ağlarında ve bilgisayar tarafından üretilen ağlarda gerçekleştirilmiştir. Buradaki sonuçlar *GN* algoritmasının sonuçları ile karşılaştırılmıştır. Verilen tüm test ağlarından elde edilen kümeleme sonuçlarının değerlendirilmesi konusunda ilgili çalışmada (Newman 2004 (a)) önerilen yaklaşıma göre modülerlik değeri yaklaşık olarak *0.3* ve üzeri olan kümelemeler, anlamlı ağ modüllerini (güçlü ağ topluluklarını) önermektedirler. Diğer bir çalışmada (Newman ve Girvan 2004 (b)) ise en güçlü topluluk yapısı için modülerlik değeri *1* ile, en zayıf topluluk yapısı ve rastgeleliği temsil eden değer *0* ile gösterilirken, bu değerin *0.3* ile *0.7* arasında olması normal anlamlılık düzeyinin sağlandığını gösterir.

2005 yılında *Duch* ve *Arenas* (Duch ve Arenas 2005), ekstremal optimizasyon algoritmasına dayanan sezgisel bir arama kullanarak modülerlik fonksiyonunu optimize eden ayırıcı bir algoritma önermişlerdir. Önerilen algoritmanın bilgisayar simülasyonlu ve gerçek dünya ağlarında gerçekleştirilen deney sonuçları mevcut yaklaşımların sonuçlarıyla karşılaştırılmıştır. Yazarlar, diğer algoritmalar tarafından bulunan optimal modülerlikten daha iyi sonuçların üretildiğini belirtmişlerdir.

Tasgin tarafından sunulan tez çalışmasında (Tasgin 2005) modülerlik amaç fonksiyonu, *GA* temelli yaklaşımla karmaşık ağlardaki alt-toplulukların bulunması amacıyla kullanılmıştır. Bu çalışmada *GN* (Girvan ve Newman 2002), *FN* (Newman 2004 (a)) ve *Extremal Optimization* (Duch ve Arenas 2005) gibi algoritmalara değinilmiş ve topluluk tespitinde karşılaşılan çeşitli problemlerden bahsedilmiştir. Gerçek hayattan modellenen birçok ağ yapısı, binlerce veya daha fazla düğüme sahip olacak kadar karmaşık ve büyük olabilmektedir. Ancak çoğu klasik yöntem oldukça yüksek zaman karmaşıklığına sahiptir. Bu yöntemler daha karmaşık ağlarda uygun çözümler üretemediklerinden ve ağa veya ortaya çıkarılacak topluluklara ait ön bilgilere ihtiyaç duyduklarından ağ topluluklarının tespitinde yeterli olmamaktadırlar (Tasgin 2005). Bahsi geçen problemlerin çözümü amacıyla ağdaki global özelliği (*network modularity* (Newman ve Girvan 2004 (b))) optimize eden *GA* temelli bir yaklaşım (*GACD—Genetic Algorithm in Community Detection*) önerilmiştir. *GACD* algoritması gerçek dünya verilerinden Zachary karate kulübü ve Enron e-posta ağı gibi karmaşık ağlarda test edilmiştir. Önerilen algoritmanın hem ağ topluluklarının tespitindeki başarısı hem de düşük zaman karmaşıklığına [*O(n) time-complexity*] sahip olması gibi avantajları ilgili çalışmada (Tasgin 2005) yazar tarafından vurgulanmıştır.

Ağlardaki alt-grupları keşfetme yöntemleriyle ilgili geçmişteki çalışmalar *Newman* tarafından (Newman 2006 (a)) iki ana alt başlıkta sınıflandırılmıştır. İlk başlık çizge bölümlemedir ve bu konuda özellikle bilgisayar bilimleri ve ilgili alanlarda paralel hesaplama, tümleşik devrelerin tasarımı gibi uygulama alanlarında çalışılmaktadır. İkincisi ise blok modelleme, hiyerarşik kümeleme veya topluluk yapısının tanımlanması gibi isimlerle tanımlanmıştır. Bu alanda, sosyologlar ve daha yakın zamanda fizikçiler ve uygulamalı matematikçiler tarafından özellikle sosyal ve biyolojik ağlarda uygulamalar gerçekleştirilmiştir. Bu çalışmada yukarıda verilen iki alt başlıkla ilgili benzer ve farklı özellikler hakkında bilgiler verilmiştir. Ayrıca, modülerlik amaç fonksiyonunun yeniden formülize edildiği de belirtilmiştir (Newman 2006 (a)).

2006 yılında *Gunes* tarafından yazılım ajanları kullanılarak karmaşık ağlarda alt-toplulukların bulunması konusunda tez çalışması yapılmıştır (Gunes 2006 (a)). Yazılım ajanları ilgili çalışmada genel olarak bilgisayar bilimlerinde kullanıcılar adına çeşitli görevleri yerine getirmek için belirli bir özerkliğe sahip olan karmaşık bir yazılımın varlığı tanımlanmıştır. Aynı şekilde ağ topluluğu terimi de kendi üyeleri aralarında yoğun olarak bağlı olan; ancak diğer gruplardaki üyelerle seyrek olarak bağlı olan alt-gruplar olarak tarif edilmiştir. Bu alt-grupların tespiti amacıyla modülerliği temel alan yeni bir algoritma önerilmiştir. Algoritmanın hem sonuçlarının doğruluğu hem de karmaşıklığı-ölçeklenebilirliği sırasıyla; *Zachary* karate kulübü ve *Reuters* ağlarında test edilmiştir. Yazar, algoritmanın kabul edilebilir sürede optimal çözüme yakın alt-toplulukların tespit edildiğini belirtmiştir (Gunes 2006 (a)). Tez çalışmasında çeşitli gerçek dünya ağları hakkında kısa bilgiler verilmiş ve ağlar hakkında bazı özellikler de sunulmuştur. Buna ek olarak, *Gunes* ve *Bingol* tarafından aynı tez çalışmasıyla ilgili bilimsel bir çalışma da yayımlanmıştır (Gunes ve Bingol 2006 (b)).

Ağ modülerliğini optimal hale getirmeye dayanan *GA* yaklaşımı karmaşık ağlarda toplulukları tespit etmek amacıyla *Tasgin* ve diğerleri tarafından sunulmuştur (Tasgin ve ark 2007). Bu çalışmada, ağlardaki toplulukların tespiti problemine genetik algoritmanın adapte edilmesi açıklanmış ve gerekli algoritma adımları ayrıntılarıyla verilmiştir. Önerilen yaklaşım; (*i*) üretilmiş kromozomlara uygunluk fonksiyonunun uygulanması, (*ii*) popülasyondaki kromozomların uygunluklarına göre sıralanması ve en uygun kromozomların belirlenmesi, (*iii*) daha sonrası için tüm popülasyondaki en iyi belli sayıdaki kromozomların sonraki nesile aktarılması, (*iv*) uygunluk değerlerine göre sıralanmış kromozomların eşleştirilmesi ve çaprazlama işleminin seçilen çiftlere uygulanması, (*v*) mutasyon işlemi ile popülasyondaki çeşitliliğin arttırılması, (*vi*)

çaprazlama ve mutasyon işlemleri sonrası oluşturulan yeni kromozomlar ile popülasyondaki mevcut kromozomların birleştirilmesi ve bu işlemin maksimum nesil sayısına ulaşılıncaya kadar devam etmesi şeklinde çalışma adımlarından oluşur (Tasgin ve ark 2007).

Büyük ağların topluluk yapısını ortaya çıkarmak amacıyla modülerlik amaç fonksiyonunu kullanan ve daha hızlı çalışan yeni bir hiyerarşik optimizasyon algoritması *Blondel* ve diğerleri tarafından önerilmiştir (Blondel ve ark 2008). Önerilen algoritma *2.6* milyon müşteriden oluşan bir Belçika cep telefonu ağındaki dil topluluklarının belirlenmesi işleminde ve *118* milyon düğümden oluşan bir web ağının analiz edilmesinde kullanılmıştır. Burada önerilen algoritmanın elde ettiği başarılar geçici modüler ağlarla (*ad-hoc modular networks*) da doğrulanmıştır.

Ağ modüllerinin tespiti konusunda sıklıkla önerilen metasezgisel yaklaşımlardan biri *GA*'dır. Bu algoritma çeşitli ayrık problemlere kolayca uygulanabilmesi, tatmin edici sonuçlar vermesi ve diğer algoritmalar için örnek teşkil etmesi sebebiyle bu konuda literatürde oldukça fazla tercih edilmektedir (Tasgin ve ark 2007, Fortunato 2010, Li ve ark 2013 (c), Atay ve Kodaz 2016). *Pizzuti* tarafından sunulan bir çalışmada *GA* temelli yeni bir yöntem önerilmiştir (Pizzuti 2008). Bu çalışmada, topluluk skoru (*community score*) isimli yeni bir uygunluk fonksiyonu önerilmiştir. Bu çalışmada, ağlar çizgelerle modellenmiş ve popülasyondaki genotiplerin dizi yapısına uygun bir şekilde temsil edilebilmesi için *LAR* gösterimi tercih edilmiştir (Park ve Song 1998, Pizzuti 2008).

Yudong tarafından sunulan doktora tez çalışmasında, karmaşık ağların dinamiği ve yapısı ile ilgili çeşitli optimizasyon problemlerine odaklanılmıştır. Ayrıca karmaşıklık analizi, çeşitli optimizasyon algoritmalarının açıklamaları, ortaya çıkarılan ağ topluluklarından biyolojik çıkarım yapma gibi konularda ek bilgiler verilmiştir. Bu tez çalışmasında (Sun 2009) ele alınan optimizasyon problemlerinden biri ağdaki fonksiyonel toplulukların bulunmasıdır. Bu problemle ilgili önerilmiş mevcut klasik algoritmaların birçoğu ağdaki modülleri yinelemeli bir şekilde tüm ağı bölme yoluyla tespit ederler. Bu durum modüllerin boyutlarında bir yanlılığa/sapmaya (*bias*) yol açar ve algoritmaların etkinliğini kısıtlar. Bu çalışmada bahsi geçen kısıtlamalarla ilgili problemleri çözen yeni bir algoritma önerilmiştir. Önerilen modül algılama algoritması gen ekspresyon profillerinden oluşturulmuş etkileşim ağlarında fonksiyonel modülleri bulmak amacıyla kullanılmıştır. Deneyler sonucu elde edilen toplulukların Gen

Ontolojisi (*GO*) ile karşılaştırıldığında istatistiksel olarak anlamlı bir şekilde biyolojik olarak ilişkili olduğu gösterilmiştir (Sun 2009).

Ağ modülerliğini optimize ederek topluluk yapılarını tespit etmeye çalışan diğer bir metasezgisel yaklaşım *PSO*'dur. Bununla ilgili sunulan bir çalışmada (Shi ve ark 2009), önerilen algoritma iki temel özelliğe sahiptir. Birincisi, toplulukların sayısı otomatik olarak belirlenir. İkincisi ise popülasyondaki parçacıklar öz vektörün ifade ettiği bileşenleri kullanarak topluluk merkezlerini belirleyen düşük boyutlu yapılara sahiptirler. Algoritmanın test sonuçları *GN* (Girvan ve Newman 2002) ve *FN* (Newman 2004 (a)) gibi diğer metotların sonuçlarıyla karşılaştırılmış ve *PSO* temelli algoritmanın daha iyi performansa ulaştığı vurgulanmıştır.

Modüllerin veya topluluk yapılarının tespit edilmesi biyoloji, fizik ve sosyal bilimlerdeki karmaşık ağların temel özelliklerinin anlaşılması açısından önemlidir. Ağın topolojik özelliklerinin bir ölçümü olarak modülerlik fonksiyonunun önerilmesinden itibaren bu fonksiyonun maksimize edilmesine dayanan çeşitli metodolojiler geliştirilmiştir. Ancak gerçek ağlardaki karmaşıklığın büyüklüğü düşünüldüğünde, ilgili optimizasyon probleminin *NP-zor* olması sebebiyle özellikle her bir ağ için en uygun çözüme ulaşmak oldukça zordur. *Xu* ve diğerleri (Xu ve ark 2010) belirtilen problemleri dikkate alan karışık tam sayılı optimizasyon algoritmasını daha büyük ağlarda modüllerin algılanması amacıyla kullanmışlardır.

"Community detection in graphs" (Fortunato 2010) isimli makalede, ağdaki modüllerin veya topluluk yapılarının ortaya çıkarılması konusunda ayrıntılı bir literatür araştırması ve kapsamlı bir çalışma yapılmıştır. Bu bilimsel makale önerdiğimiz tez çalışmasına da oldukça önemli katkılar sağlamıştır. Bu makalede gerçek ağlardaki toplulukların özelliklerinden, topluluk tespitindeki temel elemanlardan, literatürdeki topluluk bulma yöntemlerinden, anlamlılık analizinden, bu alandaki gerçek dünya uygulamalarından, karmaşıklığın değerlendirilmesinden ve temel çizge teorisi gibi benzer birçok konudan oluşan temel ve gerekli bilgiler verilmiştir.

Leskovec ve diğerleri ağ topluluk tespitinde yaklaşım algoritmalarını ve sezgisel yöntemleri karşılaştırmak, tanımladıkları alt kümelerin göreceli performanslarını değerlendirmek ve bu kümelerdeki sistematik eğilimleri anlamlandırmak amacıyla karşılaştırmalı bir çalışma yapmışlardır (Leskovec ve ark 2010). Bu çalışmada, aynı zamanda tek ve çok kriterli ölçütler olmak üzere iki kategoride incelenen amaç fonksiyonlarının karşılaştırılması da yapılmıştır.

Ağdaki modülleri tanımlamak için *Cytoscape* (Shannon ve ark 2003, Cytoscape 2017) yazılımındaki *Glay* (Su ve ark 2010) eklentisi *Su* ve diğerleri tarafından önerilmiştir. *Cytoscape*, biyolojik ağların analizi ve görselleştirilmesi amacıyla gerçekleştirilmiş açık kaynak kodlu bir yazılım temelidir. Bu yazılımda sunulan *Glay* eklentisi *Cytoscape* kullanıcılarına ağ kümeleme ve yapısal görselleştirme için çok yönlü topluluk yapısı algoritmaları ve çizge yerleştirme fonksiyonları sunan çeşitli koleksiyonlara sahiptir. Bu eklentide çeşitli ağ topluluk tespiti algoritmalarının yanında kullanıcılar için zengin görselleştirme seçenekleri de sunulmaktadır (Su ve ark 2010).

Karmaşık ağlarda örüntü tanıma için büyük bir ilgi konusu olan topluluk algılama problemi pek çok gerçek dünya ağında var olan sıkıca bağlı alt grupların gözetimsiz keşfedilmesini temsil eder (Steinhaeuser ve Chawla 2010). İlgili çalışmada, ağlardaki topluluk yapısının tespiti için önerilen farklı ölçütler karşılaştırılmış ve sonuçlar değerlendirilmiştir. Bu amaçla bazı yöntemler incelenmiş, *Zachary karate kulübü* (Zachary 1977) ve *Amerikan Kolej Futbolu* (Girvan ve Newman 2002) gibi bilinen gerçek ağlar üzerinde testler gerçekleştirilmiştir. Testlerdeki algoritmaların toplam çalışma süreleri, performansları ve bunun gibi kriterleri temel alan karmaşıklık ve ölçeklenebilirlik analizleri de yapılmıştır (Steinhaeuser ve Chawla 2010).

Sadi ve diğerlerinin bilimsel çalışmasında (Sadi ve ark 2010 (b)) ve *Sadi* tarafından sunulan tezde (Sadi 2010 (a)), güçlü bağlı alt-çizgeleri (klik—*clique*) arayan paralel karıncalar kullanılarak sosyal ağlardaki topluluk yapılarının tespiti konusuna odaklanılmıştır. Doğal esinli bir optimizasyon aracı olarak karınca kolonisi yöntemi seçilmiştir. Önerilen tez çalışmasındaki algoritma beş temel adımdan oluşmaktadır (Sadi 2010 (a)). Bunlar: (*i*) Ağın boyutuna bağlı olarak özellikle büyük ölçekli ağlardaki yüksek işlem maliyetini önlemek amacıyla temsil edilen çizge, daha küçük alt-çizgelere indirgenir ve oluşan her alt-çizge üzerinde sonraki adımlar için birbirleriyle paralel olacak şekilde iş parçacıkları çalıştırılır. (*ii*) Her iş parçacığında karınca kolonisi yöntemleri çalıştırılır. Popülasyondaki karıncalar her alt-çizgede en iyi yarı-bağlı alt-çizgeler listesini oluşturmaya çalışırlar. Çizgelerin tam bağlı olması amaçlanır, fakat önerilen bir eşik değer ile belli oranda ayrıtın eksik olmasına da izin verilir ve bu işlem boyunca yarı-bağlı alt-çizgeler de listeye dahil edilirler. Çözüm listesindeki alt-çizgelerin kalitesine bağlı olarak en fazla puana sahip üye en iyi karınca olarak işaretlenir. (*iii*) Bu karıncalar tarafından temin edilen listedeki alt-çizgelerde üstüste binme—*overlapping* olarak isimlendirilen ve bir düğümün paylaşılmasını ifade eden problemin düzeltilmesi gereklidir. Bu problem en fazla düğüme sahip olan ve örtüşen

düğümü (*overlapping node*) de barındıran bir alt-çizgeye, bu düğümün diğer alt-çizgelerden silinerek aktarılmasıyla düzeltilir. (*iv*) Düzeltilen alt-çizgeler (örn. *cliques*) tek bir düğüm haline dönüştürülür. Diğer düğüm grupları ve atanmamış düğümlerle birlikte çizge dönüştürme işleminde kullanılacak bu yeni düğüme *çizge-düğüm* adı verilir. Bu aşamada, indirgenmiş ana çizgedeki tüm bağlantılar silinir ve bağlantı-arasındalık (*edge-betweenness*) kavramından esinlenerek belirlenen ağırlık değeri ile çizge-düğüm ve atanamamış düğümler arasında yeni bağlantılar oluşturulur. (*v*) Son aşamada ise yeni indirgenmiş çizge topluluk tespiti yapan hızlı açgözlü bir algoritma ile işlenir ve elde edilen sonuçlar orijinal çizgenin sonuçları ile karşılaştırılır. Yukarıdaki adımları ortak olarak kullanan karınca kolonisi optimizasyonu yaklaşımının üç farklı tekniği, literatürdeki popüler orta ve büyük ölçekli sosyal ağlar üzerinde test edilmiş ve elde edilen sonuçlar ayrıntılı olarak tez çalışmasının deney sonuçları ve tartışma bölümlerinde değerlendirilmiştir (Sadi 2010 (a)).

Çeşitli gerçek dünya sistemlerini temsil eden karmaşık ağların analizi ve bu ağlardan anlamlı modüllerin ortaya çıkarılması gibi konularda son zamanlarda birçok çalışma yapılmıştır (Pereira-Leal ve ark 2004, Hutchings 2011, Jia ve ark 2012, Leung ve ark 2014, Bennett ve ark 2015, Ghiassian ve ark 2015, Lecca ve Re 2015, Cai ve ark 2016, Sporns ve Betzel 2016, Li ve ark 2016 (a), Khan ve Niazi 2017, Yan ve ark 2017, Zheng ve ark 2017, Atay ve ark 2017 (a), Atay ve ark 2017 (b)). Bununla ilgili *Cerami* tarafından önerilen doktora tezinde, insan biyolojik sistemlerinin etkileşim ağları ve yolak verilerinin ayrıntılı analizi ile ilgili bir çalışma yapılmıştır (Cerami 2011). Bu çalışmada, klinik kanser veri setleri üzerinde birtakım çalışmalar ve değerlendirmeler yapılmıştır. Ağ modüllerinin tespiti konusunda *Cytoscape* ve gen ontolojisi zenginleştirme aracı gibi yazılımlar kullanılarak analizler gerçekleştirilmiştir.

Karmaşık ağlardaki topluluk yapılarının ortaya çıkarılması için tek bir fonksiyonunun optimize edilmesi genel olarak uygulanan bir yaklaşım olsa da bu alanda çok amaçlı optimizasyon yaklaşımı da özellikle son yıllarda önemli hale gelmiştir. *Pizzuti*, önerdiği *GA* temelli yaklaşım (Pizzuti 2012) ile ağlardaki toplulukların bulunması için iki farklı amaç fonksiyonunun optimizasyonuna odaklanmıştır. Burada, topluluk yapılarının tespiti çok amaçlı bir optimizasyon problemi olarak formüle edilmiş ve optimum *Pareto* yapısı ile amaç fonksiyonlarının optimize edilmesi için hedefler arasında en iyi uzlaşmaya karşılık gelen bir dizi çözüm önerilmiştir. Bununla ilgili ayrıntılı algoritma adımları bu çalışmada (Pizzuti 2012) sunulmuş olup, önerilen

yöntemin başarısı literatürde sıklıkla tercih edilen bir küme değerlendirme ölçütüyle (*NMI*) test edilmiştir.

Diğer bir çok amaçlı optimizasyon yaklaşımı *Gong* ve diğerleri tarafından sunulmuştur (Gong ve ark 2012). Bu çalışmada *Negative-Ratio-Association* ve *Cut-Ratio* isimli amaç fonksiyonlarını uygunluk ölçütleri olarak kullanan ve birbirine zıt amaçları optimize eden *MOEA/D-Net* isimli ayrıklaştırmaya dayalı çok amaçlı evrimsel bir algoritma ağdaki toplulukların tespiti amacıyla sunulmuştur.

Yang ve *Leskovec* dört farklı kategoride inceledikleri 13 adet topluluk puanlama fonksiyonunu 230 tane farklı büyüklükteki sosyal, işbirliği ve bilgi ağları seti üzerinde test eden bir değerlendirme metodolojisi önermişlerdir (Yang ve Leskovec 2012 (a)). Bu çalışmada, farklı puanlama kriterleri kullanılarak bulunan topluluk yapıları, veri kümesi üzerinde öncesinden belirlenmiş olan referans ağ topluluk yapılarıyla (*ground-truth communities*) (Arınık ve ark 2015) karşılaştırılmıştır. Tüm test sonuçlarına göre topluluk puanlama işlevlerinin-fonksiyonlarının davranışlarında büyük farklılıklar olduğu vurgulanmıştır.

Sosyal ve biyolojik ağlar gibi bazı karmaşık sistemlerin temsil edildiği çizgelerin düğümleri arasındaki etkileşimlerin zamanla değişmesine izin veren bir ağ topolojisine sahip olması gerçek dünya sistemlerinin temsiline daha yakındır. Verilen bu ağların zaman bağımlılığının anlaşılması, zamana göre değişen ağın fonksiyonel özellikleri ve genel yapısı hakkında oldukça önemli bir bakış açısı sağlayabilir. *Afsariardchi* tarafından sunulan tezde, çeşitli statik ve dinamik kümeleme algoritmaları gözden geçirilmiş ve dinamik ağlarda örtüşen toplulukları ortaya çıkaran *Bayes* modeline dayanan yeni bir yöntem önerilmiştir (Afsariardchi 2013). İlgili tezde, önerilen yaklaşımın deneysel çalışmaları yapay ve gerçek ağlar üzerinde gerçekleştirilmiştir. Yazar, elde edilen sonuçlarla bu yaklaşımın son zamanlarda sunulmuş iyi bilinen algoritmalardan daha iyi performans sergilediğini belirtmiştir (Afsariardchi 2013).

GA temelli diğer bir yaklaşımda—*MENSGA*, düğümler arasında geçişi sağlayan matris kodlama benimsenmiş ve başlangıç popülasyonunun üretilmesi için genotiplerin çeşitliliğini arttıran düğüm benzerliği seçilmiştir (Li ve ark 2013 (c)). Bu yaklaşımın çalışma adımları ilgili makalede ayrıntılarıyla verilmiş ve *FN* (Newman 2004 (a)), *GN* (Girvan ve Newman 2002), *TGA* (Cog ve ark 2007) klasik algoritmalarından daha iyi sonuçlar verdiği vurgulanmıştır. Ayrıca, *MENSGA* algoritmasının topluluk yapılarının açıkça belli olmadığı karmaşık ağlarda *CCGA* (HE Dong-Xiao 2010) ve *LGA* (Jin ve ark 2011) algoritmalarından daha iyi sonuçlara ulaşamamasına rağmen; gerçek ağların

topluluk yapılarının bu algoritmalarla aynı doğruluk seviyesine göre elde edildiği gözlemlenmiştir (Li ve ark 2013 (c)).

Bugüne dek ağlardaki modüllerin tespiti için birçok yöntem önerilmiştir. Bunlar içerisinde en fazla umut verici sonuçlara ulaşan yaklaşımlardan bazıları pratikte oldukça iyi sonuçlar veren ve sağlam matematiksel temele dayalı istatistiksel çıkarım yöntemleridir. *"Community detection and graph partitioning"* isimli çalışmada (Newman 2013), en fazla kullanılan iki adet çıkarım yönteminin standart minimum-kesit çizge bölümleme probleminin versiyonlarına doğrudan eşleştirilebildiğini gösterilmiştir. Bahsi geçen çalışmada önerilen yeni yöntemle hem sonuçların kalitesi hem de çalışma süresi açısından önceki en iyi yöntemlere kıyasla başarılı sonuçların elde edildiği belirtilmiştir.

Yüksek verimli teknolojinin gelişimi ile birlikte genler arası, proteinler arası, metabolitler arası ve benzeri etkileşimleri temsil eden biyolojik verilerin kullanılabilirliği önemli ölçüde artmıştır. Teknolojik ve bilimsel gelişimle birlikte saklanan verilerin optimum bir şekilde modellenip analiz edilebileceği birtakım teknikler, araçlar ve algoritmalar geliştirilmiştir ve halen de bu süreç devam etmektedir. *Bennett* tarafından sunulan doktora tezinde, biyolojik ağların analizinde kullanılan ve ağ teorisinin belirli bir yönünü temsil eden modüllerin veya topluluk yapılarının tespiti konusunda çalışmalar gerçekleştirilmiştir (Bennett 2013). Bu amaçla ayrık ve örtüşen topluluk yapılarını hem yönsüz hem de ağırlıksız ağlarda tespit eden matematiksel programlama modelleri geliştirilmiştir. Diğer bir yaklaşımla da dinamik ağlardaki modüllerin tespiti üzerine bir yöntem geliştirilmiştir. Ayrıca bahsi geçen tez çalışmasıyla ilgili çeşitli bilimsel makaleler de yayımlanmıştır (Xu ve ark 2010, Bennett ve ark 2012 (a), Bennett ve ark 2012 (b), Bennett ve ark 2012 (c)).

Karmaşık bir ağdaki toplulukları saptama problemi doğrusal olmayan ve zorluk derecesi oldukça yüksek (*NP-zor problem sınıfında*) olan bir optimizasyon problemi olarak modellenebilir (Li ve Song 2013 (a)). Bu problemin çözümü için umut verici sonuçlar veren ve geleneksel çaprazlama-mutasyon adımlarını kullanmak yerine olasılık dağılımlarına göre genotipleri örnekleyerek yeni bireyler üreten istatistiksel öğrenme mekanizması temelli yeni bir yaklaşım önerilmiştir. Bu çalışmada önerilen algoritma, genişletilmiş kompakt *GA* (*extended compact genetic algorithm—ECGA*) olarak isimlendirilmiştir (Li ve Song 2013 (a)). *Ubuntu Linux* ortamında *C++* programlama dili ile kodlanan bu algoritma hem elde edilen ağ modüllerin kaliteleri hem de çalışma süreleri açısından yine aynı programlama diliyle kodlanan *GATHB* (Tasgin ve ark

2007), *CNM* (Clauset ve ark 2004), *GA-Net* (Pizzuti 2008) ve *GN* (Girvan ve Newman 2002) algoritmalarıyla kıyaslanmıştır. Tüm deneyler, *GN* (Girvan ve Newman 2002) ve *LFR* (Lancichinetti ve ark 2008) karşılaştırma ağlarında ve altı adet gerçek dünya test verisinde yapılmıştır. Elde edilen sonuçlara göre *ECGA*'nın önceki bazı topluluk algılama algoritmalarından daha etkili olduğu vurgulanmıştır.

Narayanan tarafından önerilen doktora tezinde, biyolojik ağların analizi için bir hesaplama tekniği olarak topluluk algılama yaklaşımı ele alınmıştır (Narayanan 2013). Ağdaki modül yapılarının ortaya çıkarılması ağ yapısı ve işlevi arasında var olan etkileşimi anlamak için bir temel oluşturabilir. Bu tez çalışmasındaki yaklaşımla biyolojik ağların barındırdığı ulaşılabilir ve önemli sayılabilecek alt toplulukların ortaya çıkarılması için bazı uygulamalar gerçekleştirilmiştir. Ayrıca burada topluluk tespitinde modülerlik temelli optimizasyon amaçlanmıştır. Önerilen yeni yaklaşımın geçerlilik testleri için *Yeast*, *C. elegans* and *Drosophila* canlılarına ait *PPI* ağları kullanılmıştır. Ayrıca bu tez çalışmasında, lineer cebir ve çizge teorisinden elde edilen tekniklerden oluşan bir kombinasyon kullanılarak *Duchenne Muscular Dystrophy* (*DMD*) hasta gen ekspresyon verilerini analiz etmek için yeni bir yaklaşım geliştirilmiştir. Bu genetik kas hastalığı verisinin kullanımı ile ağdaki modüllerin tespiti sayesinde bu hastalığın patolojisinin anlaşılması üzerine odaklanılmıştır. *DMD* iskelet kaslarının gen ekspresyonu ölçümleri patolojiye yol açan sebeplerin anlaşılması ve altta yatan mekanizmaların tespiti açısından bazı fırsatlar verebilir. Bahsi geçen tezde sunulan 3. bölümdeki bulgularla, *DMD* hastalığı hedefli ilaç geliştirmeyi içeren tedavi edici uygulamaların umut verici olduğu belirtilmiştir. Bu tezdeki çalışmalar dört ayrı bölüme ayrılmıştır. Her bir bölümle ilgili önerilen tekniklere ve ulaşılan sonuçlara bu çalışmada (Narayanan 2013) ulaşılabilir.

Literatürde önerilen yaklaşımlardan çoğu ağlardaki modüllerin tespitinde yüksek hesaplama maliyeti problemiyle karşı karşıyadır. En uygun çözümlerin elde edilmesi yanında hesaplama karmaşıklığı da önemli bir değerlendirme kriteridir. *Subelj* ve *Bajec* tarafından önerilen yayılım tabanlı algoritma ile ağdaki grupların tespitine çalışılmış ve yukarıda bahsedilen hesaplama karmaşıklığının maliyeti hakkında bu yeni algoritma neredeyse ideale yakın bir çözüm sunmuştur (Subelj ve Bajec 2014).

Orman ve diğerleri, karmaşık ağlardaki topluluk tespiti probleminde önerilen yöntemlerin daha nesnel değerlendirilebilmesi için geniş kapsamlı gerçekçi koşullar altında testlerin yapılması gerektiğini belirtmişlerdir (Orman ve ark 2013). Yapay test ağları genellikle bu amaç için kullanılmaktadır (*LFR için bakınız* (Lancichinetti ve ark

2008)). Buna rağmen bu gerçekçi yöntem (*LFR*) dahi gerçek dünya ağlarının bazı tipik özelliklerine uygun yapay test ağları üretememektedir. Bu istenmeyen durumun üstesinden gelmek için *LFR* yönteminin iki alternatif modifikasyonu sunulmuştur. Değişiklikler sayesinde önerilen yaklaşımlarla oluşturulan ağların orijinal *LFR* ile üretilen ağlardan daha fazla gerçek ağlardaki özelliklere sahip olması amaçlanmıştır (Orman ve ark 2013). Zaten deneysel sonuçlar da üretilen ağların merkezileşme ve derece korelasyon değerlerinin gerçek dünya ağlarında karşılaşılan değerlere daha yakın olduğunu göstermiştir.

Sosyal ağlarda bulunan önemli modüllerin/toplulukların tanımlanması problemi artan sosyal ağ etkileşimleri ile daha da önemli hale gelmiştir. Sosyal bilgileri barındıran kayıtların ve ilişkisel bağlantıların analiz edilerek gizli ve anlamlı toplulukların keşfedilmesine yönelik çeşitli yaklaşımlar geliştirilmektedir. Bu amaçla yapılan yüksek lisans tez çalışmasında *Kakışım* (Kakışım 2013), suç ağı olduğu kanıtlanmamış olası şüpheli ağların analiz edilmesi için bir üretici olasılıklı topluluk tanıma modeli geliştirmiştir. Bu çalışmada, bir sosyal ağda bulunan kişilere ait yönlü mesaj verisi kullanılarak benzer konulara ilgi duyan üyelerin aynı topluluklar altında kümelenmeye çalışıldığı bir yaklaşım önerilmiştir. Önerilen yöntem—*YATK*, *Enron* şirketi bünyesinde çalışan 158 kişi tarafından üretilen 600.000'den fazla e-postanın bulunduğu bir gerçek dünya ağında (Klimt ve Yang 2004) test edilmiştir. Elde edilen deney sonuçlarına göre *Enron* ağında hem içerik hem de topolojik olarak benzer özellikli üyelerin başarıyla gruplandırıldığı gözlenmiştir (Kakışım 2013).

Huang tarafından sunulan doktora tez çalışmasında, sistem biyolojisinde karşılaşılan başlıca zorluklardan biri olan fonksiyonel modüllerde veya hastalık modüllerinde genlerin veya proteinlerin rollerini ortaya çıkarma problemine odaklanılmıştır (Huang 2014). Bu tezde gerçekleştirilen temel çalışmalar üç ana başlıkta ele alınmıştır. İlk bölümde, hastalık popülasyonlarında aktif düzenleyicileri etkili bir şekilde bulmak için veri güdümlü (*data-driven*) bir yöntem olan Korelasyon Kümesi Analizi (*CSA*) sunulmuştur. İkinci ve üçüncü bölümlerde sırasıyla; ilişkili gen modüllerini saptamak için yeni bir algoritma ile fonksiyonel ağ modüllerini genişletmek amacıyla modüle dayalı yeni bir yaklaşım sunulmuştur. Deney setleri, *KEGG* (*Kyoto Encyclopedia of Genes and Genomes*) veri tabanı ve *TCGA* (*The Cancer Genome Atlas*) gibi platformlardan temin edilmiştir.

Dinamik ağlarda gelişen toplulukların keşfi problemi oldukça zor olan önemli bir araştırma konusudur. Son zamanlarda evrimsel kümeleme temelli yaklaşımlar bu

alanda oldukça tercih edilmektedir (Folino ve Pizzuti 2014). Ağ topluluklarının zamansal değişimle birlikte tespit edilebilmesi için bu problem çok amaçlı olarak *Folino* ve *Pizzuti* tarafından formülize edilmiş ve *GA* temelli evrimsel bir yöntem önerilmiştir (Folino ve Pizzuti 2014). Önerilen yöntem *DYNMOGA* (*DYNamic MultiObjective Genetic Algorithms*) olarak isimlendirilmiştir. *Modularity* (Newman ve Girvan 2004 (b)), *conductance* (Kannan ve ark 2004), *normalized cut* (Shi ve Malik 2000) ve *community score* (Folino ve Pizzuti 2010) amaç fonksiyonlarıyla diğer değerlendirme kriterleri ilgili makalede verilmiştir. *DYNMOGA* ile elde edilen sonuçlar bazı en yeni algoritmaların sonuçlarıyla karşılaştırılmıştır. Sonuç olarak, önerilen bu çok amaçlı optimizasyon yaklaşımının dinamik ağlardaki topluluk yapılarının ortaya çıkarılmasında başarılı bir şekilde uygulanabileceği vurgulanmıştır (Folino ve Pizzuti 2014).

PPI ağları, biyolojik ağlar içinde en fazla ulaşılabilir veri setlerinden biridir. Bu ağlardaki üye sayıları da çok büyük olabilmektedir. Bu durum yüksek hesaplama karmaşıklığını beraberinde getirmektedir. Yüksek maliyete rağmen bu tür biyolojik etkileşim ağlarından fonksiyonel olarak anlamlı sayılabilecek modüllerin ortaya çıkarılması önemlidir. Bu amaçla, *Liu* ve diğerleri tarafından sürü zekası temelli karınca kolonisi kümeleme algoritması—*ACCMLF* önerilmiştir (Liu ve ark 2014). Yazarlar önerdikleri algoritmanın yüksek boyutlu ağlarda düşük çalışma süresine sahip olduğunu vurgulamışlardır. Bu yaklaşımla giriş ağını temsil eden karmaşık ve büyük *PPI* verisini daha küçük bir hale çeviren yeni bir eşleştirme stratejisi geliştirilmiştir. Bu ağın kümeleme işlemi için karınca koloni algoritması tercih edilmiştir. Son olarak, lokal minimuma takılmadan kümeleme sonuçlarını ilk işleme tabi tutarak en uygun çözümün elde edilmesi sağlanmıştır. Deneyler için kıyaslama amacıyla tercih edilen büyük boyutlu maya canlısı *PPI* veri setleri kullanılmıştır. Farklı değerlendirme kriterlerinin referans alındığı sonuçlara göre *ACCMLF* algoritmasının hem çalışma süresinin uygunluğu hem de fonksiyonel modüllerin tespiti açısından *MCODE* (Bader ve Hogue 2003) ve *MINE* (Rhrissorrakrai ve Gunsalus 2011) gibi diğer kabul görmüş yöntemlere göre başarılı sonuçlar verdiği anlaşılmıştır.

Harenberg ve diğerleri tarafından (Harenberg ve ark 2014) yapılan derleme çalışmasında, büyük ölçekli gerçek dünya ağlarındaki ayrık ve örtüşen ağ topluluklarının tespiti için ayrıntılı bir literatür araştırması yapılmıştır. Bu çalışmada, sekizi en son önerilen ve beşi de geleneksel yaklaşımlar sunan toplam 13 adet algoritmanın sonuçları değerlendirilmiştir. Bu algoritmalar iki farklı özelliğe göre kıyaslanmıştır. İlki tanımlanmış toplulukların yapısal özelliklerini ölçen *modularity* gibi

puanlama ölçütleri; diğeri ise elde edilen ağ topluluklarını daha öncesinden belirlenmiş olan referans ağ topluluklarıyla karşılaştıran küme değerlendirme ölçütleridir. İlgili çalışmada, bu algoritmalar hakkında özet bilgilere ve deneysel sonuçlara ulaşılabilir (Harenberg ve ark 2014).

Yapay Arı Kolonisi (*ABC*) optimizasyon tekniği (Karaboga ve Basturk 2007) çeşitli gerçek dünya problemlerinin en optimal şekilde çözümlenebilmesi için başarılı bir şekilde kullanılan doğa esinli algoritmalardan biridir. Birçok optimizasyon yönteminin ağlardaki modül yapılarının tespit edilmesinde kullandıkları mekanizmalara benzer şekilde *ABC* algoritması da topluluk sayılarının otomatik olarak belirlendiği bir yaklaşımla bu probleme uygun çözümler sunmaktadır (Hafez ve ark 2014). Herhangi bir ağda ortaya çıkarılan toplulukların en optimal yapılarda olması, seçilen optimizasyon fonksiyonlarına doğrudan bağlıdır. Bu amaçla bahsi geçen çalışmada (Hafez ve ark 2014), iki ayrı kategoride incelenen çeşitli amaç fonksiyonlarını kullanan *ABC* temelli yaklaşımla bu fonksiyonların performansları karşılaştırılmıştır.

Ağlardaki toplulukların ortaya çıkarılması için seçilebilecek iki bilgi kaynağı mevcuttur. Bunlar (*i*) ağ yapısı veya (*ii*) düğüm özellikleri/nitelikleridir. Topluluk bulma algoritmaları bilindiği üzere ağ yapısına odaklanırken; kümeleme algoritmaları çoğunlukla düğüm niteliklerini göz önünde bulundururlar. *Yang* ve diğerleri (Yang ve ark 2013), düğüm özniteliklerine sahip ağlardaki örtüşen toplulukları tespit edebilmek için doğru ve ölçeklenebilir olan *CESNA* (*Communities from Edge Structure and Node Attributes*) isimli algoritmayı geliştirmişlerdir. Bu algoritma ağ yapısı ve düğüm özellikleri arasındaki etkileşimi istatistiksel olarak modeller ve bu da ağ yapısında doğal olarak bulunan gürültülü verilere karşı sağlamlılık ve kararlılık sunar. Böylece bu yaklaşım daha başarılı topluluk algılamasını sunar. Yazarlar önerilen yaklaşımın matematiksel modelini sunup; bu algoritmanın lineer karmaşıklığa sahip olduğunu belirtmişlerdir. Bununla birlikte önerilen algoritmanın Google ve Facebook gibi büyük ve karmaşık ağlarda da oldukça iyi sonuçlara ulaştığı vurgulanmıştır.

Orman tarafından sunulan doktora tezinde (Orman 2014), toplulukların ortaya çıkarılması problemi bu toplulukların en karakteristik özelliklerinin belirlemesinden oluşmakta ve bu yapıların yorumlanması işlemi tespit sürecinden bağımsız bir problem olarak ele alınmaktadır. Bahsi geçen problemler iki alt başlıkta ele alınmıştır. İlki topluluğun temsili için uygun bir yolun bulmasıdır. İkincisi ise bu temsilin en karakteristik bölümlerinin objektif olarak seçilmesidir. Bu problemlerin çözümü için dinamik ağ yapılarının karakteristik özelliklerinden yararlanılmıştır. Bu tezde, karmaşık

ağlarla ilgili modelleme, topolojik özellikler ve ölçütler hakkında bilgilerin yanında düz (*plain*), özellikli ve dinamik ağlardan oluşan veri setlerinde toplulukların tespit edilmesiyle ilgili açıklamalar yapılmıştır. Ayrıca, elde edilen toplulukların yorumlanması hakkında da ayrıntılı çalışmalar yapılmış olup; bu tezde çeşitli topluluk bulma yaklaşımları ilgili alt başlıklar altında sunulmuştur. Bu tez çalışması dışında ağlardaki topluluk veya modül yapılarının bulunması ve gerçek dünya sistemlerinden anlamlı bilgilerin ortaya çıkarılması amacıyla çeşitli alanlarda birçok tez çalışması yapılmıştır. Yönlü ağlarda topluluk algılama ile sosyal ağ analizi (Kim 2014), *PPI* ağlarında çizge madenciliği ve modül algılama (Shen 2014), sosyal ağlarda toplulukların bulunması (Özturk 2014), farklı türlerden biyolojik ağ modüllerinin tespiti (Wang 2015) ve *"modularity density"* amaç fonksiyonun optimize edilmesiyle kaliteli toplulukların keşfi (Chen 2015) gibi konulardaki çalışmalar literatüre sunulan tezlerden bazılarıdır.

Çoklu topluluk yapılarının tanımlaması için tek bir amacı en iyileyen yöntemlerin fonksiyon özelliklerine oldukça bağlı olmalarından dolayı çok amaçlı optimizasyona odaklanan bazı yöntemler önerilmiştir. Çok amaçlı olarak tasarlanan memetik algoritma temelli yaklaşım (*MMCD*), *Wu* ve *Pan* tarafından önerilmiştir (Wu ve Pan 2015). Bu algoritma, üç temel işleme sahip güçlü bir yerel arama prosedürüne sahiptir. İlk işlemde, baskın bir popülasyonda gerçekleştirilen evrimsel süreçlerle üretilen baskın olmayan çözümler lokal arama prosedürü için başlangıç popülasyonunu temsil eder. İkincisinde, *pseudonormal* olarak isimlendirilen yeni bir yön vektörü iki objektif fonksiyonu bir araya getirerek yeni bir uygunluk fonksiyonu oluşturur. Sonuncusunda ise etiket yayılım kuralına dayalı arama stratejisi yerel optimal çözümlerde en iyi çözümleri etkin bir şekilde aramak için bu çözüm uzayını genişletir. Gerçek ve yapay ağlar üzerinde gerçekleştirilen testlerin sonuçları *MMCD* algoritmasının önerilen yerel arama stratejisi sayesinde daha iyi çözümlere hızlıca yakınsadığını göstermiştir.

Klonal seçim algoritması (de Castro ve Von Zuben 2002) edinilmiş bağışıklığın klonal seçimi teorisinden esinlenen yapay bağışıklık sistemleri alanlarından biridir. Bu bağışıklık sistemlerindeki temel elemanlar ağ topluluklarının tespiti problemine adapte edilmiş ve modülerlik amaç fonksiyonunu en iyilemeye çalışan antikor-antijen uygunluğuna göre uygun topluluk yapılarının ortaya çıkarılması amaçlanmıştır. Bu makalede sunulan deneysel sonuçlara göre önerilen algoritmanın daha düşük bir

hesaplama maliyetiyle daha ölçeklenebilir ve daha doğru çözümler elde ettiği gözlemlenmiştir (Cai ve ark 2015).

Son zamanlarda önerilen topluluk bulma algoritmaları teknolojik gelişmelerle birlikte daha çok büyük ve karmaşık ağlardaki problemlere odaklandıklarından dolayı uygun çözümü çeşitli kalite fonksiyonlarının optimize edilmesinde aramaktadırlar. Bu fonksiyonlar farklı çizgelerin yapısal özelliklerini ölçerler ve bu özniteliklere karşılık gelen her bir tanımlama ile topluluk yapılarının ortaya çıkarılmasını sağlarlar. Bu çıkarım seçilen fonksiyonun özelliğine birebir bağlı olur. *Creusefond* ve diğerleri yukarıda bahsedilen amaç fonksiyonlarının davranışlarına göre bu ölçütleri gruplandıran (*modularity optimization*, *random walks* ve *heuristics* kategorilerinde) bir metodoloji önermişlerdir (Creusefond ve ark 2016). Bu metodolojide amaç işlevsel özelliklerine göre seçilen ölçütleri farklı kategorilerde tanımlamak ve her kategori için en iyi kalite fonksiyonlarını belirlemektir. Tüm testler yapay olarak üretilen *LFR* çizgeleri ile Youtube, Amazon ve bunlara benzer gerçek dünya ağlarında gerçekleştirilmiştir. Bu makalede elde edilen gruplamanın doğruluğu *NMI* kriteriyle test edilmiş ve sonuçlara göre aynı kategorideki fonksiyonların *NMI* değerlerinin istisnalar haricinde benzer olduğu anlaşılmıştır (Creusefond ve ark 2016).

Modern ağ biliminin en popüler konularından biri olan ağ modüllerinin tespitiyle ilgili *Fortunato* ve *Hric* tarafından sunulan "*Community detection in networks: A user guide*" isimli makale, bu alanda önemli bir çalışmalardan biridir (Fortunato ve Hric 2016). Bu çalışmada, öncelikle literatürde önemli sayılabilecek bazı çalışmalar hakkında bilgiler verilmiş olup; topluluk tespitinde klasik ve modern yaklaşımlar ile bu konuyla ilgili genel bakış açıları sunulmuştur. Ayrıca, ağ topluklarının tespitinde önerilen metotların ve algoritmaların erişilebilir yazılımları ile online olarak sunulan test ağları (*LFR gibi*) bu çalışmada farklı kategorilerde verilmiştir.

GA ile bütünleştirilmiş çok etmenli bir yaklaşım sunan ve *MAGA-Net* (*Multi-Agent Genetic Algorithm*) olarak isimlendirilen yeni bir algoritma ağlardaki topluluk yapılarının tespit edilmesi amacıyla önerilmiştir (Li ve Liu 2016 (b)). Bu çalışmada sunulan etkili komşuluk temelli operatörler yardımıyla arama uzayındaki en iyi çözümün arayışı sağlamıştır. *MAGA-Net* algoritmasının performansı çeşitli ağlar üzerinde test edilmiş ve algoritmanın test sonuçları *GA-Net* (Pizzuti 2008) ve *Meme-Net* (Gong ve ark 2011) algoritmalarının sonuçlarıyla karşılaştırılmıştır. Son olarak, algoritmanın hızlı bir şekilde çözüme ulaşması ve hem doğru hem de istikrarlı topluluk

yapılarının ortaya çıkarılması gibi konulardaki başarısı yazarlar tarafından vurgulanmıştır (Li ve Liu 2016 (b)).

Botta ve *Genio*, yakın zamanda sunulan *"modularity density"* isimli amaç fonksiyonunu optimize eden yeni bir kuadratik topluluk tespit algoritmasını önermiştir (Botta ve del Genio 2016). Burada seçilen fonksiyon modülerlik amaç fonksiyonuna göre bazı avantajlara sahiptir. Bu fonksiyon referans olan topluluklara ihtiyaç duymadan farklı boyutlardaki ağlar için kolaylıkla karşılaştırma olanağı sağlamıştır. Önerilen algoritmanın teoriksel yaklaşımını doğrulamak ve başarısını test etmek için yapılan deneylerde, *ER* modeli ile üretilen test ağları ve bilinen gerçek veriler kullanılmıştır. Buna ek olarak, bu algoritmanın en kötü durumdaki hesaplama karmaşıklığının $O(N^2)$ olduğu belirtilmiştir (Botta ve del Genio 2016). Bu algoritmanın ulaşılabilir versiyonuna *"www.fedebotta.com"* adresinden ulaşılabilmektedir (en son erişim tarihi: 19.01.2018).

Girvan ve *Newman* tarafından önerilen *GN* algoritmasının büyük ve karmaşık ağlardaki kısıtlarını çözmeyi amaçlayan *Moon* ve diğerleri bu algoritmanın iki farklı paralel versiyonunu geliştirmişlerdir (Moon ve ark 2016). Bunlardan ilki, *MapReduce* programlama modelini kullanan *SPB-MRA* (*Shortest Path Betweenness MapReduce Algorithm*) isimli algoritmadır. İkincisi ise *SPB-VCA* (*Shortest Path Betweenness Vertex-Centric Algorithm*) olarak adlandırılan düğüm merkezli paralel algoritmadır. Bu çalışmada, ağ topluluğu bulma sürecinde hızlı yakınsama sağlayan yeni bir yaklaşım da önerilmiştir. İlk algoritma *Hadoop* teknolojisi ile çoklu bilgisayarda; diğeri *GraphChi* ile tek bir bilgisayarda uygulanmıştır. Tüm algoritmaların başarıları F-score ölçütü kullanılarak belirlenmiştir. Sonuçlara göre *SPB-MRA*'nın çalışma süresinin indirgeyici sayılarının artışına paralel olarak lineer eğilimde azaldığı fakat yalnızca bir bilgisayarda işleyen *SPB-VCA* algoritmasının *SPB-MRA*'dan dört ile altı kat arasında daha başarılı sonuçlara ulaştığı gözlemlenmiştir. Sonuç olarak, yazarlar testlerde kullanılacak verinin sadece bir bilgisayarda işlenebilecek büyüklükte olması durumunda, *SPB-VCA* algoritmasının seçilmesinin uygun olacağını; aksi durumda *SPB-MRA* algoritmasının uygulanmasının daha doğru olacağını belirtmişlerdir (Moon ve ark 2016).

Ağ topluluk tespiti konusunda 2016 yılındaki bir literatür araştırmasında (Cai ve ark 2016) modül yapısını optimizasyon problemi olarak modelleyen çeşitli evrimsel algoritmalar incelenmiştir. Bu ayrıntılı literatür taraması kapsamında, tekli ve çoklu optimizasyon modelleri ile ağırlıklı/ağırlıksız, işaretli/işaretsiz, örtüşen/örtüşmeyen ve statik/dinamik olmak üzere her türlü ağ yapısıyla ilgili çalışmalara odaklanılmıştır.

Sosyal ağlardaki topluluk yapılarının ortaya çıkarılması konusunda *Cui* (Cui 2016) ve *Zadeh* (Moradian Zadeh 2017) tarafından sunulan iki farklı doktora tez çalışması bu alanın sosyal ağ analizinde tercih edilen popüler konulardan biri olduğunu göstermektedir. Her iki çalışmada da bireylerin etkileşim yoğunluklarına göre sosyal ağları anlamlı alt gruplara bölen bazı mevcut yöntemler incelenmiş ve yeni yaklaşımlar sunulmuştur.

Evrimsel kümeleme yöntemleri dinamik ağlardaki toplulukların tespiti için geçici ve süreçsel akıcılık kavramını referans almaktadırlar. Bu tür yaklaşımlar genellikle anlık ve geçici görüntü maliyetini kontrol eden bir giriş parametresine ihtiyaç duyarlar. Buradaki değişkene bağlı seçim kısıtlılığının ortadan kaldırılması ve dinamik ağlardaki toplulukların ortaya çıkarılmasındaki başarının arttırılması amacıyla *MODCS* olarak temsil edilen çok amaçlı ayrık guguk kuşu arama algoritması önerilmiştir (Zhou ve ark 2017). Öncelikle başlangıç popülasyonunun oluşturulması için yuvanın yerini kodlamada sıralı komşu listesi yöntemi kullanılmıştır. Daha sonra bir değiştirilmiş yuva yeri güncelleme stratejisi ve yuvayı bırakma operatörü ile guguk kuşu arama algoritmasının ayrık yapısı probleme uyarlanmıştır. Son olarak, önerilen bu yapı temel alınarak baskın olmayan sıralama yöntemini ve kalabalık mesafe (*crowding distance*) metodunu entegre eden çok amaçlı ayrıklaştırılmış guguk arama algoritması ağ topluluklarının tespiti için uygulanmıştır. Önerilen algoritma, *modularity* ve *NMI* ölçütlerini iki farklı amaç fonksiyonu olarak aynı anda optimize etmeyi amaçlar. Rastsal olarak üretilmiş test ağlarındaki ve bir futbol ağındaki deneysel sonuçlar *MODCS* algoritmasının diğer algoritmalarla rekabet edebileceğini göstermiştir. Bunlara ek olarak bu algoritmanın *FacetNet*, *DYNMOGA* ve *DYN-DMLS* algoritmalarından daha yüksek *NMI* skorlarına ve daha düşük ortalama hata oranlarına sahip olduğu belirtilmiştir (Zhou ve ark 2017).

Proteinler arası etkileşim ağlarındaki protein modüllerinin incelenmesi biyolojik sistemlerin ve mekanizmaların anlaşılmasına önemli ölçüde katkılar sağlar (Yu ve ark 2017). *Yu* ve diğerleri tarafından sunulan çalışmada, ağırlıklı *PPI* ağlarındaki modüllerin tespiti için dizi (*amino asit frekansı*) ve Gen Ontolojisi kombinasyonunu birleştiren yeni bir yaklaşım önerilmiştir. Önerilen yaklaşım; öznitelik çıkarımı, ağırlıklı çizge yapısının oluşturulması ve protein kompleksinin algılanması şeklinde üç temel adımdan oluşmaktadır. İlk adımda, protein kompleksinin özelliğini belirtmek amacıyla topoloji dizi bilgisi kullanılır. İkincisinde, *PPI* ağı ile gen ontolojisi bilgisi birleştirilerek altı farklı türde ağırlıklı çizgeler oluşturulur. Son adımda ise protein kompleks

algoritması; yoğunluk, ağ çapı ve dahil edilen açı kosinüsü—*included angle cosine* olarak belirlenen üç koşula göre ağırlıklı çizgeden uygun kümeleri bulur. Deneyler, bir maya canlısına (*yeast*) ait iki protein veri seti üzerinde sürdürülmüştür. *F-measure* ve *precision* kriterleri önerilen yöntemin beş farklı klasik algoritma ile karşılaştırılması için seçilmiştir. Tüm sonuçlara göre önerilen algoritmanın diğerlerinden daha etkili olduğu belirtilmiştir (Yu ve ark 2017).

Li ve diğerleri karmaşık ağlarda topluluk tespiti yardımıyla veri madenciliği konusuna odaklanmışlardır (Li ve ark 2017). Sunulan çalışmada, diferansiyel evrim temelli sosyal örümcek optimizasyonu (*DESSO/CD*) topluluk yapılarının ortaya çıkarılması amacıyla geliştirilmiştir. Sonuçlarına göre *DESSO/CD* algoritmasının mevcut yöntemlerle karşılaştırıldığında, topluluk yapılarının daha yüksek doğrulukla ve daha düşük hesaplama maliyetiyle tespit edilebildiği yazarlar tarafından belirtilmiştir.

Omar tarafından sunulan tezde, sosyal ağlardaki toplulukların tespiti üzerine çalışmalar yapılmıştır (Omar 2017). Yazar, sosyal ağlardaki alt grupların zamana göre gelişimlerini ve değişimlerini çizge yapılarıyla göstermiştir. Ayrıca, bu tezde sosyal ağlar hakkında genel bilgiler verilmiş olup; ağ türleri, sosyal ağ analizi, ağlardaki topluluk kavramı, problem tanımı, literatürdeki topluluk tespit yöntemleri ve bu konuda önerilen yaklaşımın deneysel sonuçları gibi alt-balıklarda bilgiler verilmiştir. Önerilen yaklaşım bir açıdan *modularity*, *network diameter*, *graph density*, *page rank* ve benzeri fonksiyonları temel alan bazı çizge teorisi temelli ölçütlerin analizini ifade etmektedir. *Zachary's karate club* isimli sosyal etkileşim ağındaki test sonuçları tez çalışmasındaki ilgili şekillerde ve tablolarda özet olarak sunulmuştur.

Ağlardaki modül veya topluluk yapılarının ortaya çıkarılması için önerilen diğer bir yöntem havai fişek algoritmasıdır (*FWA*). Bu algoritma doğa olaylarından esinlenen sürü zekasına dayalı optimizasyon tekniklerinden biridir. *Guendouz* ve diğerleri tarafından sunulan ayrık güncellenmiş *FWA* (Tan ve Zhu 2010) çözüme yakınsamayı hızlandırmak ve algoritmayı zenginleştirmek için etiket yayılım stratejisine dayalı yeni bir popülasyon başlatma ve mutasyon stratejisi önerilmiştir (Guendouz ve ark 2017). Önerilen yeni mutasyon stratejisi ile çözüm uzayındaki çeşitliliğin arttırılması amaçlanmıştır. *Modularity* fonksiyonundaki kısıtlamaların çözümü amacıyla *Li* ve diğerleri (Li ve ark 2008) tarafından önerilen *modularity density* (*D*) ölçütü *FWA* için amaç fonksiyonu olarak seçilmiştir. Önerilen algoritma, *Infomap*, *CNM* ve *GA-net* isimli üç algoritma ile yapay ve gerçek dünya ağlarının sonuçlarına göre karşılaştırılmıştır. Değerlendirme kriterleri olarak *modularity* (*Q*) ve *NMI* seçilmiştir.

Sonuç olarak, önerilen algoritmanın diğerlerine göre daha başarılı sonuçlar verdiği yazarlar tarafından belirtilmiştir (Guendouz ve ark 2017).

Karmaşık ağlardaki toplulukların veya modüllerin ortaya çıkarılması konusunda kapsamlı bir literatür araştırması yakın zamanda (2017'nin sonuna kadar) *Khan* ve *Niazi* tarafından yapılmıştır (Khan ve Niazi 2017). Bu çalışmaya benzer olarak görülebilecek bazı çalışmaların ((Newman 2006 (a), Fortunato 2010, Leskovec ve ark 2010, Harenberg ve ark 2014, Lancichinetti ve Fortunato 2014, Fortunato ve Hric 2016) gibi) daha öncesinden sunulmasına rağmen bu yeni literatür incelemesinin diğerlerinden daha güncel ve kapsayıcı olduğu anlaşılmaktadır. Çünkü sunulan araştırma çalışmasının en önemli özelliği bilimsel literatürün görselleştirilmesi amacıyla kullanılan *CiteSpace* programı yardımıyla temel literatür kaynağı olan *Web of Science* isimli veri tabanından konuyla ilgili karmaşık ağ analizlerini içeren tüm makalelerin bu çalışmada gözden geçirilmesidir (Khan ve Niazi 2017). Ayrıca bu çalışmada, topluluk tespiti alanında çalışan yazarlar ile kurum ve ülke bilgileri, temel makaleler, atıf yapılan referanslar, konu kategorileri ve önemli dergiler gibi bilgiler ele alınmıştır. *CiteSpace* programının alan bazlı literatür araştırmasına göre en merkezde bulunan temel düğümün *Yong Wang* olduğu belirtilirken; *Mark Newman* isimli kişinin bu ağda en çok bahsedilen yazar olduğu gözlemlenmiştir. "*Reviews of Modern Physics*" dergisinin diğer dergilere göre en güçlü atıf sayısına sahip olduğu da belirtilmiştir. Tüm bunlara ek olarak, "*Bilgisayar Bilimi*" ve "*Mühendislik*" kategorilerinin frekans ve merkezde olma özellikleriyle diğer disiplinlere yön verdiği belirtilmiştir. Yazarlar çalışmalarında, topluluk tespiti için genel bilgiler verirken bu amaçla önerilen çeşitli optimizasyon fonksiyonlarının gelişimi ve iyileştirmelerinden bahsetmişlerdir. İlgilenilen veri setinin yüzde *94.802*'si makaleleri, yüzde *4.517*'si incelemeleri, yüzde *4.455*'i konferans bildirilerini, yüzde *0.495*'i editöryal çalışmaları, yüzde *0.155*'i kitap bölümlerini, yüzde *0.093*'ü mektupları ve yüzde *0.062*'si notları içermektedir. *SCI*, *SCI-Expanded*, *SSCI* ve *A&HCI* kategorilerinde yayımlanmış *3168* adet veri seti, ayrıntılı olarak analiz edilmiş ve çalışmanın genelinde çok amaçlı ve zengin içerikli bir literatür incelemesi olarak araştırmacıların imkanına sunulmuştur.

Bu çalışmanın sonraki bölümünde, büyük ve karmaşık yapılı gen etkileşim ağlarında *survival* (*hayatta kalma*, *yaşam* veya *sağkalım* gibi ifadelerle temsil edilebilmektedir) ile ilişkili *CNA*'lar içeren alt-ağların tanımlaması problemi ele alınmıştır. Tez çalışmasında, *survival* ifadesinin karşılığı olarak *sağkalım* (Kazancı 2003) terimi tercih edilmiştir. Sağkalım süresi, bir bireyin belirli bir girişime ya da etkene maruz kaldıktan sonra iyileşmesine, hastalığın tekrarlamasına ya da ölümüne

kadar geçen zamana denir (Yayla 2013). Yaşam çözümlemesi ile ilgili analizde (*survival analysis*), her biri için genellikle başarısızlık olarak nitelenen bir eşiğe göre gruplanan farklı bireyler kümesi odağı oluşturur. Başarısızlık (ölüm, vazgeçme ve diğer durumlarda) belirli bir süreden sonra oluşur ve bu başarısızlık zamanı *"failure time"* olarak isimlendirilir. Bir diğer tanıma göre "*sağkalım süresi*" yaşayan bir organizmanın ya da bir nesnenin belirli bir başlangıç ile ölüm (başarısızlığı, iyileşmesi vb.) zamanları arasındaki süreyi ifade eder (Sertkaya ve ark 2005). Bugüne kadar genomik çalışmalarda *hayatta kalma* ile ilişkili mutasyonlara sahip gen gruplarının tespitine yönelik az da olsa bazı çalışmalar yapılmıştır. Tüm çalışmalardan en dikkate değer olanlarında biri bu tez çalışmasının Bölüm 2.2'sindeki probleme de ilham kaynağı olan ve 2016 yılında *Hansen* ve *Vandin* tarafından sunulan makaledir (Hansen ve Vandin 2016). Bu makaleye benzer çalışmalar (Vandin ve ark 2012, Wu ve Stein 2012, Hofree ve ark 2013, Patel ve ark 2013, Reimand ve Bader 2013, Leung ve ark 2014, Jeong ve ark 2015) literatürde mevcuttur. Burada, benzer kanser türlerine sahip birçok hastanın biyolojik ağ bilgilerine göre analiz yapılması hedeflenmiştir. Bu analizin temel amacı, *sağkalım süresi* gibi klinik parametrelerle ilişkili gen gruplarının tespit edilmesidir. Ancak genlerin ve mutasyonların (veya diğer genetik değişikliklerin) yolaklardaki davranış durumlarının karmaşıklığı gerçeğine bağlı olarak kanser mutasyonlarının genetik heterojenliği sebebiyle bu amacın gerçekleştirilmesinin oldukça zor olduğu yazarlar tarafından belirtilmiştir (Hansen ve Vandin 2016). Burada, büyük miktardaki kanser hizalama çalışmalarından ve genom-ölçekli etkileşim ağlarından elde edilen veri kümeleri kullanılmıştır. İlgili makaledeki (Hansen ve Vandin 2016) problemle ilişkili skor hesaplama ölçütü olarak *log-rank istatistik testi* temelli bir fonksiyon önerilmiştir. Bu tez çalışmasında ilgili problemdeki uygunluk kriteri için bu ölçüt amaç fonksiyonu olarak seçilmiştir. Ele alınan problem *NP-zor* türü bir problemdir ve çözüm amaçlı önerilen algoritma *NoMAS* (*Network of Mutations Associated with Survival*) olarak isimlendirilmiştir (Hansen ve Vandin 2016). Bu algoritma renk-kodlama (Alon ve ark 1994) tekniğinin probleme uyarlanmasına dayanır. *NoMAS* algoritması maksimum bağlantılı *k-kümeli log-rank* (*max connected k-set log-rank* (Hansen ve Vandin 2016)) problemini çözmek için önerilmiştir. Bu algoritmanın deney sonuçları daha önce literatüre sunulan aç gözlü yaklaşımların sonuçlarıyla karşılaştırılmış ve iki adet kanser verisindeki sağkalımla ilişkili gen grupları tespit edilmiştir. Bu tez çalışmasında ele alınan ikinci konuda, (Hansen ve Vandin 2016) referanslı çalışma temel almakla birlikte diğer çalışmalar (Kim ve ark 2016, Ajwad 2017, Vandin 2017) da referans alınmıştır.

2. PROBLEMLERİN TANIMLANMASI VE KARMAŞIKLIK ANALİZİ

Bu tez çalışması üç temel bölümü kapsamaktadır. İlk bölüm karmaşık ağ analizinde modül yapılarının ortaya çıkarılması problemini temsil eder. İkinci bölüm gen etkileşim ağlarındaki sağkalım parametresiyle ilişkili *CNA*'lara sahip gen gruplarının tespiti problemini ifade eder. Son bölüm ise birinci ve ikinci konuları birden ele alan yeni bir problemle ilgilidir. Bu bölümde, sağkalım ile en fazla ilişkili gen grupları elde edildikten sonra bu genlerin ilk bölümdeki tekniklerle elde edilen kanserle ilişkili biyolojik ağlardaki modüllerden herhangi birinde maksimum sayıda birlikte olmaları durumuna göre yapılacak işlemler anlatılmıştır. Aşağıdaki alt-bölümlerde odaklanılan problemlerin ayrıntılı açıklamaları yapılmıştır. Bu açıklamalar çeşitli matematiksel ve istatistiksel yaklaşımların yanında grafiksel gösterimlerle de desteklenmiştir.

2.1. Ağ Modüllerinin Ortaya Çıkarılması

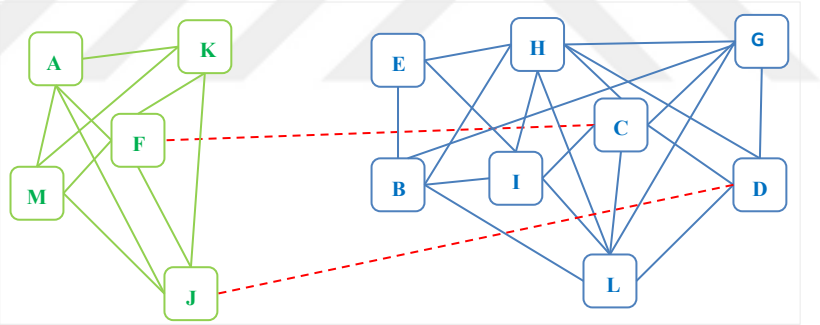
Çoğu karmaşık gerçek dünya sistemi kendisini temsil edebilecek ağ yapıları ile modellenebilir. Bu ağ yapıları anlamlı olabilecek küçük alt-ağlar veya modüller içerebilmektedir. Çizge teorisinde bir ağ modülü sahip olduğu düğümleri kendi aralarında güçlü ilişkilerle birbirlerine bağlı olan bir alt-çizgeyi ifade eder. En genel tanımıyla herhangi bir modül, ağdaki bağlantı sayısı, ağırlıkların yoğunluğu, düğüm sayısı, komşuluk ilişkisi ve bunun gibi belli özelliklere göre elde edilir.

Daha genel bir tanıma göre herhangi bir ağ temsili bir çizge ile gösterildiğinde, bir ağ modülü kendi içinde maksimum ortak öznitelik, bağlantı sayısı, konumsal benzerlik vb. nitelik veya niceliklere sahip alt-çizge yapıları olarak düşünülebilir. Bu yapıların elemanları olan düğümler kendi modüllerindeki düğümlerle (*modül içi*) maksimum ilişkiye veya daha çok ortak özelliğe sahipken; diğer modüllerdeki düğümlerle (*modül dışı*) minimum ilişkiye veya daha az ortak özelliğe sahip olmalıdırlar. Sosyal etkileşim ortamlarında güçlü ilişkilere sahip kişilerin birbirleriyle oluşturdukları ortaklıktan dolayı benzer topluluklarda yer almaları; çevresel ağlarda birbirleriyle beslenen canlıların benzer gruplarda olmaları; bilgisayar ağ yapılarındaki cihazların birbirleriyle oluşturdukları iş birliğinin veya veri alışverişinin maksimum olması sebebiyle aynı modüllerde olmaları ve benzeri örnekler ağ modülleri ile ilgili

verilebilecek örneklerden birkaçıdır. Bu ağ modülleri (Lecca ve Re 2015); topluluklar (Fortunato 2010), alt-çizgeler, topluluk yapıları (Newman 2004 (a)), yapısal alt birimler (Palla ve ark 2005), kümeler (Leskovec ve ark 2010), kompleksler (Li ve Chen 2013 (b)), gruplar (Subelj ve Bajec 2014) ve bağlı bileşenler (Duch ve Arenas 2005) gibi isimlerle literatürde ifade edilmektedir.

2.1.1. Ağ modülleri ve tanımlamalar

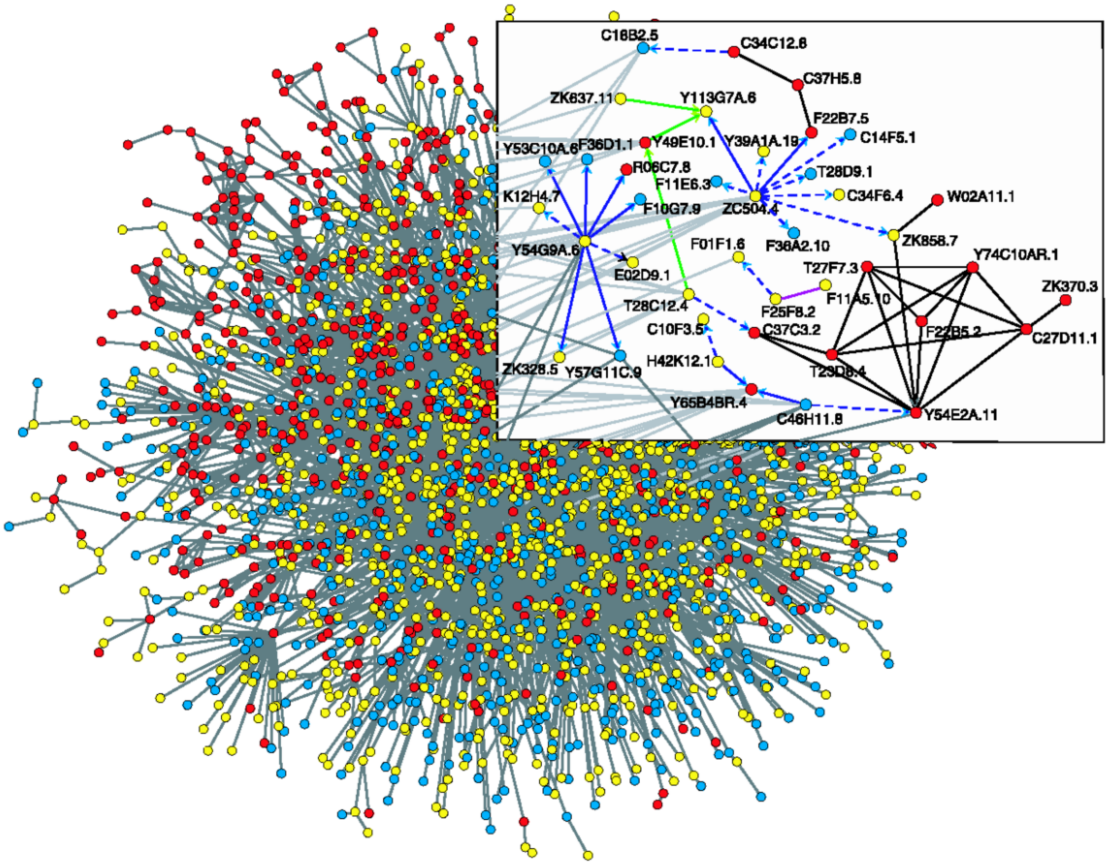
Örnek bir ağ modül yapısı Şekil 2.1'de verilmiştir. Bu şekildeki yeşil ve mavi renkler kendi modül yapılarını gösterirken; kırmızı renkli bağlantılar modül dışı veya önemsiz bağlantıları gösterir. Burada iki adet modül olduğu farz edilirse, birinci modül yeşil renkli düğümlerden—*A*, *F*, *J*, *K*, *M*; ikinci modül mavi renkli düğümlerden—*B*, *C*, *D*, *E*, *G*, *H*, *I*, *L* oluşur. Elde edilen modül üyeleri kendi grubundaki üyelerle maksimum etkileşime sahipken; diğer modüllerdeki üyelerle oldukça az etkileşime sahiptir veya herhangi bir etkileşime sahip değildir.



Şekil 2.1. Temsili modül yapıları

Bu bölümde ele alınan problemin biyolojik ağlardaki temsilinin anlaşılabilmesi açısından Şekil 2.2'de *c. elegans* etkileşim ağı verilmiştir (Li ve ark 2004, Albert 2005, Bennett 2013). Verilen ağ, düğümleri proteinleri temsil eden örnek bir *PPI* ağıdır. Burada düğümler için atanan renkler filogenetik sınıflara göre belirlenmiştir. Şekil 2.2'deki antik—proteinler (*"ancient"*) kırmızı; çok hücreli—proteinler (*"multicellular"*) sarı; solucan—proteinler (*"worm"*) mavi renklerle gösterilmiştir (Li ve ark 2004). Ayrıca, grafiğin sağ üst köşesindeki küçük ve yakınlaştırılmış ayrıntılı resim bu ağın küçük parçasını (*alt-ağ*) temsil eder. Verilen bu biyolojik ağdan yararlanılarak yapılacak çeşitli analizlerle bu canlının hücrelerindeki biyolojik sürecin

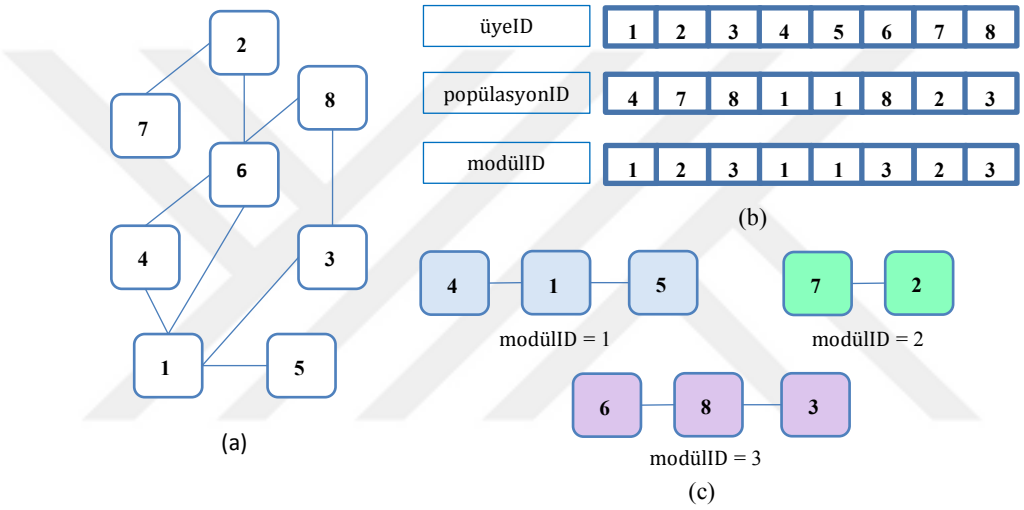
anlaşılması mümkün olabilmektedir. Birbirleriyle etkileşim halinde olan proteinlerin fonksiyonel benzerliklerinden yola çıkılarak özellikleri veya biyolojik sürece katkıları tam olarak bilinmeyen diğer moleküllerin özelliklerinin anlaşılması mümkün olabilmektedir (Bennett 2013).



Şekil 2.2. *c. elegans* protein etkileşim ağı (Li ve ark 2004, Albert 2005)

Bu tezdeki Bölüm 3.4'te verilen tüm algoritmalar çizgelerin gösterimi için "yer seçimi tabanlı komşuluk temsilini" (*locus-based adjacency representation—LAR*) (Park ve Song 1998) kullanırlar. Popülasyondaki her bir üye dizi yapılarıyla temsil edilirse dizilerdeki her bir hücre bir düğüme karşılık gelir. Ayrıca burada tanımlanan üyeler *GA*'daki kromozomlara karşılık gelirken; üyelerdeki tüm elemanlar (*dizilerdeki hücreler*) kromozomlardaki genlere karşılık gelirler. Örneğin tanımlanan bir üyenin (üyeID) herhangi bir hücresi düğü$m-i$'yi göstersin. Burada i, düğümün sıra numarasını gösterir. Her üye alt alta gelecek şekilde popülas$yonID$ ve modülID isimlerinde iki farklı bilgi tutar. Birinci bilgi düğü$m-i$'nin komşuları arasından rastgele seçilen düğümün numarasını tutar (popülas$yonID-i$). İkinci bilgi ise birinci

bilgiye göre oluşturulan modüller için $düğüm - i$'nin ait olduğu modülün *ID*'sini tutar ($modülID - i$). Bu bilgilere göre Şekil 2.3-a'da, sekiz adet düğümden oluşan bir çizge yapısı; Şekil 2.3-b'de, bu ağa göre oluşturulmuş örnek bir üye yapısı ve Şekil 2.3-c'de ise bu üyenin bilgilerine göre oluşturulan alt-çizgeler (*modüller*) verilmiştir. Elde edilen bu modüller farklı renklerde sunulmuştur. Şekil 2.3-b'de üç ayrı bilgiye sahip örnek bir üye sırasıyla; $üyeID$, $popülasyonID$ ve $modülID$ dizilerine sahip olsun. Birinci dizideki bilgiler düğüm sıralarını; ikinci dizideki bilgiler düğümlerin seçilen komşularını ve üçüncü dizideki bilgiler ise bu düğümlerin hangi modüllere ait olduklarını gösterir.



Şekil 2.3. Örnek bir çizge yapısı (**a**), temsili üye yapısı (**b**) ve modül bilgileri (**c**)

Temsili bir *G* çizgesi (*ağ*) ve *C* alt-çizgesi (*modül*) tanımlansın. Toplam düğüm ve bağlantı sayıları, *G* çizgesi için *n* ve *m*; *C* modülü için n_c ve m_c olsun. Bu çizgenin komşuluk (*bitişiklik* veya *yakınlık*) matrisi *A* olsun. Bu durumda eğer *i* ve *j* düğümleri komşu ise A_{ij}, *1* değerini; aksi halde *0* değerini alır. Normalde modüllerin veya toplulukların bağlı olmalarından dolayı *C* alt-çizgesinin de bağlı olduğu farz edilsin (Fortunato ve Hric 2016). Şekil 2.1'deki iki bağlı modüle sahip çizge referans olarak alınabilir.

C modülü ile uyumlu bir *G* çizgesinin iç ve dış dereceleri toplamı sırasıyla; d_C^{int} ve d_C^{ext} olarak ifade edilsin. Ortalama dereceler ise $d_C^{int^{avg}}$ ve $d_C^{ext^{avg}}$ ile temsil edilsin. Son olarak, iç ve dış bağlantı yoğunlukları φ_C^{int} ve φ_C^{ext} ile gösterilsin. Buna göre,

$$d_C^{int} = \sum_{i,j \in C} A_{ij} \tag{3}$$

$$d_C^{int^{avg}} = \frac{d_C^{int}}{n_c} \tag{4}$$

$$\varphi_C^{int} = \frac{d_C^{int}}{n_c \times (n_c - 1)} \tag{5}$$

$$d_C^{ext} = \sum_{i \in C \,\&\, j \notin C} A_{ij} \tag{6}$$

$$d_C^{ext^{avg}} = \frac{d_C^{ext}}{n_c} \tag{7}$$

$$\varphi_C^{ext} = \frac{d_C^{ext}}{n_c \times (n - n_c)} \tag{8}$$

şeklinde denklemler elde edilir. *i.* düğümün ait olduğu modüldeki düğümlerle olan toplam bağlantı sayısı d_C^{int} ile ifade edilirken; d_C^{ext} değişkeni bu düğümün ait olduğu modül dışındaki diğer tüm düğümlerle olan toplam bağlantı sayısını verir. Diğer değerler de yukarıda verilen denklemlere göre hesaplanır. Ayrıca hem iç hem de dış bağlantıların bir arada değerlendirildiği durumlarda, toplam derece d_C^{sum}, ortalama derece d_C^{avg} ve iletkenlik (*conductance*) C^C sayıları, Denklemler 9, 10 ve 11'e göre hesaplanır. Burada iletkenlik, *C* modülünün dış derecesi ile toplam derecesi arasındaki oranı gösterir.

$$d_C^{sum} = d_C^{int} + d_C^{ext} \tag{9}$$

$$d_C^{avg} = \frac{d_C^{sum}}{n_c} \tag{10}$$

$$C^C = \frac{d_C^{ext}}{d_C^{sum}} \tag{11}$$

2.1.2. Problem tanımı ve karmaşıklık

Ağlardaki toplulukların veya modüllerin tespitiyle ilgili problemin standart bir tanımı bulunmamaktadır (Chen 2015, Fortunato ve Hric 2016). Buna rağmen popüler tanımlardan bir tanesi şudur: Bir modül rastgele ortaya çıkmasından daha çok birbirlerine çok güçlü bağlantılara sahip düğümler topluluğunu ifade eder (Newman ve Girvan 2004 (b)). Bulunmak istenen bu topluluk üyeleri güçlü olarak birbirine bağlı olan ve klik olarak ile isimlendirilen bir alt-çizge olarak düşünülebilir. Bu *k-klik topluluğu*, birbirine komşuluğu bulunan *k* adet düğümlü bir alt-kümedir (Chen 2015). Ancak bir modül veya topluluk yapısı bir klik'ten daha zayıf bir yapıya sahiptir. Bu yüzden bu zayıflık ağ modüllerinin tespit edilmesini zorlaştırmaktadır. *Chen* tarafından önerilen tez çalışmasındaki ağ topluluklarının tanımlanması ile ilgili bilgiler aşağıda sunulmuştur (Chen 2015).

Radicchi ve diğerleri (Radicchi ve ark 2004) ağdaki topluluk yapıları için niceliksel bir tanımlama sunmuşlardır. Bu tanımlamaya göre topluluk düğümleri arası bağlantıların yoğunluğu ağın geri kalanına oranla daha yoğundur. $\partial_i^{iç}$ ve $\partial_i^{dış}$ değişkenleri düğüm-*i* için sırasıyla; iç ve dış dereceler toplamını temsil etsin. Denklem 12'deki şartı sağlayan bir *C* modülü, *G* çizgesi için yoğun bir alt-çizgeyi temsil eder.

$$\partial_i^{iç} > \partial_i^{dış}, \forall\, i \in C \tag{12}$$

Yukarıdaki tanımlama dışında zayıf yoğunluklu ilişkilere sahip ağ modülleri için de çeşitli tanımlamaları yapılmıştır. Bir *G* çizgesi için *C*, zayıf bir modülü temsil etsin. Denklem 13'e göre $E_{iç}$, bu modüldeki düğümler arası toplam bağlantı sayısını; $E_{dış}$, modüldeki düğümlerin ağın kalanındaki düğümlerle olan bağlantı sayısını ifade etsin.

$$2 \times |E_{iç}| > |E_{dış}| \tag{13}$$

Zayıf bir toplulukta, *C* içindeki derecelerin toplamı ağın geri kalanının temel alındığı derece toplamından daha büyüktür. Güçlü bir topluluk, bu zayıf topluluğun özelliğine sahipken; bunun aksi doğru değildir. Zayıf ve güçlü/yoğun toplulukların tanımlanması ile ilgili diğer bir yaklaşım *Hu* ve diğerleri tarafından sunulmuştur (Hu ve ark 2008). Bu yaklaşıma göre *C* modülündeki herhangi bir düğümün iç derecesi bu

düğümün diğer bir modülle paylaştığı bağlantı sayısından büyük veya en azından eşit ise *C* güçlü bir düğüm topluluğu olarak tanımlanır.

Li ve diğerleri bir düğüm topluluğunun geçerli bir modül olmasının, Denklem 14'e göre ortalama modülerlik derecesinin olabildikçe yüksek olmasına bağlı olduğunu belirtmişlerdir (Li ve ark 2008). Bu denklemde $k_{iç}^{C}$, *C* modülünün iç derecelerinin toplamının ortalamasını ve $k_{dış}^{C}$, bu modülün dış derecelerinin toplamının ortalamasını gösterir.

$$k^{C} = k_{iç}^{C} - k_{dış}^{C} \tag{14}$$

Yukarıda verilen tanımlamalar dışında literatürde farklı yaklaşımları temel alan birçok topluluk tanımı yapılmıştır ((Chen 2015) ve (Papadopoulos ve ark 2012) referansları incelenebilir).

Ağ bilimlerinde topluluk algılama probleminin temel amacı bu tür toplulukları tespit etme yollarının araştırılmasıdır. Bu yapıların analizi ve incelenmesi ağ işlevinin anlaşılmasına yardımcı olabilmektedir. Aynı zamanda topluluk/modül tespiti problemi bir çeşit çizge kümeleme ve gözetimsiz öğrenme problemi olarak da görülebilir.

Bir algoritmanın veya problemin karmaşıklığı gerçekleştirilen işlemi tamamlamak için ihtiyaç duyduğu kaynak miktarının ve çalışma süresinin niceliksel ölçüsüdür. Çizgelerle ilgili problemlerde, genellikle hem *n* düğüm sayısı hem de *m* bağlantı sayısı büyüklüğü temsil eder. Hesaplama karmaşıklığının artmasıyla bazı durumlarda karmaşıklık hesaplanamaz. Örneğin; büyük ölçekli (*large-scale*) ağlardaki toplulukların tespit edilmesinde böyle bir durum vardır (Khan ve Niazi 2017). Polinom karmaşıklığına sahip algoritmalar *P* (*polynomial*) sınıfını oluşturur. Fakat bazı önemli karar verici ve optimizasyon problemlerinin çözümü için *P* sınıfında herhangi bir algoritma bulunmaz (Fortunato 2010). Bu tür bir problemin çözümünde en kötü durumda, olasılıksal çözüm uzayının detaylı bir şekilde aranması (*exhaustive search*) ile sistem boyutunun herhangi bir polinomal fonksiyonundan daha hızlı büyüyen bir sürede (örneğin *üssel* bir sürede) çözümün bulunması amaçlanabilir. Ancak bu teorik yaklaşım sistem kısıtları sebebiyle uygulanamaz. Polinomal zamanda çözülemeyen problem kümesi *NP* (*non-deterministic polynomial time*) sınıfını oluşturur. Bu tür problemler polinomsal sürede çözülebilen karar problemlerini (*P*) de kapsarlar (Fortunato 2010). Aynı zamanda içerisinde *NP* sınıfından problem veya problemler barındıran karmaşıklık sınıfı, *NP-zor* (*NP-hard*) olarak ifade edilir ve bu tür problemler polinomsal zamanda

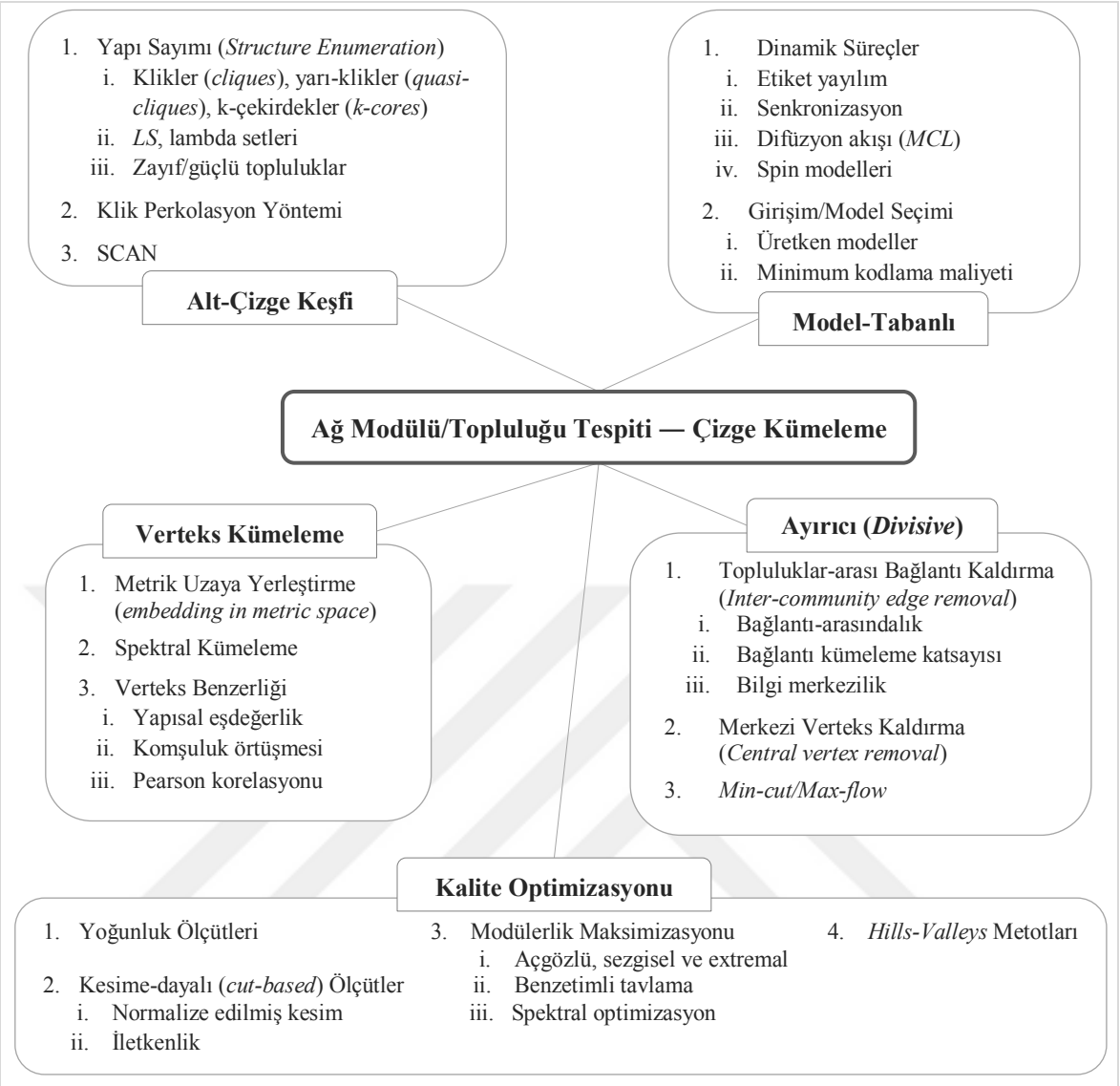
çözülemeyen problemler sınıfını kapsar. Bir problem hem *NP* hem de *NP-zor* sınıfındaysa *NP-tam* (*NP-complete*) problem türüne dahil edilebilir. Bir diğer ifadeyle, *NP-tam* problemler kümesi *NP* ve *NP-zor*'un kesişimiyle oluşan sınıfı ifade eder. Örneğin; bilgisayar bilimlerindeki gezgin satıcı problemi ve doğrusal programlama gibi pek çok iyi bilinen problemin/yaklaşımın dahil olduğu sınıf *NP-tam*'dır (Fortunato 2010). Bunlara ek olarak, bazı kümeleme problemleri de *NP-zor* sınıfına (*the sparsest cut problem* (Matula ve Shahrokhi 1990) vb.) ya da *NP-tam* sınıfına (*maksimum klik problemi* (Pardalos ve Xue 1994) gibi) dahildir.

Modül veya topluluk tespitiyle ağa özgü gizli bilgilerin ortaya çıkarılmasına çalışıldığı için başarılı bir şekilde çözüme ulaşmak oldukça zordur. Ağdaki düğümlerin diğer tüm düğümlerle bağlantılı olma olasılığı göz önüne alındığında ve bu ağlarda en az binlerce veya milyonlarca etkileşimin olduğu düşünüldüğünde ağ modüllerinin tespiti probleminin zorluğu anlaşılabilir. Bu problem özellikle büyük ölçekli ağlar için oldukça karmaşıktır. Çözüm için seçilen yaklaşımların ya da optimize edilecek fonksiyonların sahip olduğu karmaşıklığa göre bu problem tam bir çözümün sunulması durumunda *NP-tam* olarak görülebilir (Brandes ve ark 2006, Cog ve ark 2007, Porter ve ark 2009, Wang ve Hopcroft 2010) ya da *N-zor* sınıfına dahil edilebilir (Duch ve Arenas 2005, Brandes ve ark 2008, Fortunato 2010, Yang ve Leskovec 2012 (a)). Burada bahsedilen karmaşıklık sebebiyle ağ modül yapılarının ortaya çıkarılması problemine kesin olarak çözüm üreten bir yaklaşım henüz önerilememiştir. Literatüre sunulan birçok çalışma bu alanda önerilen yöntemlerin karşılaştırılması veya kategorize edilmesi amacıyla sunulmuştur (Lancichinetti ve ark 2008, Fortunato 2010, Leskovec ve ark 2010, Orman ve ark 2013, Fortunato ve Hric 2016, Khan ve Niazi 2017).

Son zamanlarda yayımlanan bir literatür araştırmasında (Khan ve Niazi 2017), bugüne kadar önerilen temel teknikler veya algoritmalar iki ana başlıkta incelenmiştir. Bunlar klasik topluluk tespiti algoritmaları ve modülerlik benzeri amaç fonksiyonlarını optimize eden tekniklerdir. Bu algoritmalardan klasik olanları çizge bölümleme, hiyerarşik kümeleme, parçalı kümeleme, spektral kümeleme ve ayrıştırıcı algoritmalar olmak üzere beş gruba ayrılmıştır. Optimizasyon temelli algoritmalar ise açgözlü teknikler, benzetilmiş tavlama, ekstremal optimizasyon, spektral optimizasyon ve evrimsel algoritmalar olmak üzere beş farklı grupta ele alınmıştır. Her bir gruptaki algoritmalarla ilgili literatüre sunulmuş çalışmalar hakkında ayrıntılı bilgi için *Khan* ve *Niazi* tarafından hazırlanan kapsamlı literatür incelemesine bakılabilir (Khan ve Niazi

2017). Doktora tezinin modül tespitiyle ilgili bölümünde bazı evrimsel algoritmalara da odaklanılmıştır.

Yukarıda verilen çeşitli tanımlamalar da dahil olmak üzere kabul gören topluluk tanımlamaları ve altta yatan metodolojiler, *Papadopoulos* ve diğerleri tarafından incelenmiş ve topluluk tespit yöntemleri farklı bir sınıflandırma ile sunulmuştur (Papadopoulos ve ark 2012). Bu sınıflandırmaya göre topluluk tespiti problemleri—çizge kümeleme yöntemleri farklı bir bakış açısıyla Şekil 2.4'te gösterilmiştir. Burada ele alınan problem bir çizge kümeleme problemi olarak da sunulmuş ve bununla ilgili mevcut tüm yöntemler beş farklı sınıfta sunulmuştur. Bunlar; alt-çizge keşfi, model-tabanlı, verteks kümeleme, ayırıcı ve kalite optimizasyonu kategorileridir. Bunlardan kalite (fonksiyonlarının) optimizasyonu kısmı bu tez çalışmasında odaklanılan ağ modül tespiti probleminde önerilen yaklaşımların ait olduğu kategoriyi temsil etmektedir. Tüm yaklaşımlar modülerlik amaç fonksiyonunun maksimizasyonuna çalışmaktadır. Bu konuda hem Bölüm 2.1'de hem de ilgili deneysel çalışmalar bölümünde ayrıntılı bilgiler sunulmuştur. Ayrıca, bu kategoriye ait olan 10 farklı amaç fonksiyonunun optimizasyonu ile ilgili deneysel çıktılar ayrıntılarıyla değerlendirilmiştir. Bir sonraki bölümde, en iyilenmesi amaçlanan fonksiyonlar anlatılmaktadır.



Şekil 2.4. Topluluk tespiti—çizge kümeleme yöntemlerinin sınıflandırılması (Papadopoulos ve ark 2012)

2.1.3. Amaç fonksiyonları

Bölüm 2.1.2'de sunulan problemin çözümü amacıyla Bölüm 3.4'te çeşitli metasezgisel yöntemler önerilmiştir. Bu yöntemler aşağıda verilen her bir kalite fonksiyonunu (Chen 2015) ağ modüllerinin ortaya çıkarılmasında uygunluk kriteri olarak kullanmıştır. Literatürde kabul gören amaç fonksiyonları ağlardaki yerel ve/veya global özellikleri dikkate alacak şekilde tasarlanmıştır. Ancak bu fonksiyonların tüm ağlar için uygun modülleri ortaya çıkaracak şekilde başarılı sonuçlar verdiğini söylemek oldukça zordur. Böylece tek bir amaç fonksiyonu ile ağ modüllerini tespit etmek yerine farklı fonksiyonların sonuçlarının değerlendirilmesi önem arz etmektedir. Bu yüzden bu

çalışmada farklı özelliklere sahip 10 adet amaç fonksiyonu gerçek dünya ağlarında test edilmiş ve sonuçlar deneysel çalışmalar bölümünde ayrıntılı olarak analiz edilmiştir.

2.1.3.1. Modülerlik (*M, Modularity*)

Literatürde en iyi bilinen ve en çok kullanılan amaç fonksiyonu *modülerlik* fonksiyonudur (Fortunato 2010). Bu kavram 2004 yılında *Girvan* ve *Newman* tarafından tanıtılmıştır. Modülerlik, ağlardaki modülleri ifade eden alt-kümelerdeki üyelerin birbirleriyle etkileşimlerinin maksimum olduğu; fakat diğer alt-kümelerdeki üyelerle mevcut etkileşimlerin minimum olduğu optimizasyona dayalı bir amaç fonksiyonudur (Newman 2004 (a), Newman ve Girvan 2004 (b)). Bu fonksiyon rastgelelikten uzak ve matematiksel bir model üzerine kuruludur. Büyük ve karmaşık çizge yapılarının analizlerinin zorluğu göz önüne alındığında, modülerlik maksimizasyonunun *NP-zor* sınıfı bir problem olduğu anlaşılmaktadır (Brandes ve ark 2008). En yalın ve orijinal haliyle modülerlik fonksiyonu, Q_{Basic} ile temsil edilmekte ve temel formülü Denklem 15'te sunulmaktadır. Burada k, toplam modül sayısını; e_{ii}, i. modüldeki ikili düğümlerin bağlantı olasılıklarını gösterirken; a_i, modül içinde en az bir bitiş noktasına sahip bağlantıların grubunu temsil eder (Newman 2004 (a), Newman ve Girvan 2004 (b)).

$$Q_{Basic} = \sum_{i=1}^{k} (e_{ii} - a_i^2) \tag{15}$$

Verilen bir *G(V, E)* çizge yapısı örnek bir ağı temsil etsin. Burada *G*, çizgeyi; *V*, düğümler kümesini ve *E* ise kenarlar kümesini temsil eder.

$V = \{v_i \mid i = 1, 2, 3, ..., n\}$ ve $E = \{e_j \mid j = 1, 2, 3, ..., m\}$ kümelerinde *n* ve *m* değerleri sırasıyla, *G*'deki toplam düğüm ve bağlantı sayılarını içerir. Bu kümelerdeki *i* ve *j* ikilisi düğüm ve bağlantı indislerini gösterir.

Ayrıca, A isimli bir komşuluk matrisi tanımlansın ve bu matris nxn boyutlu olsun. A matrisi V kümesinin elemanlarının E kümesinin elemanlarına göre ilişkilerini göstersin. Bu matris Denklem 16'ya göre oluşturulur.

$$A = \begin{cases} 1 & eğer \;\; i \; ve \; j \; düğümleri \; bağlantılı \; ise, \\ 0 & aksi \; durumda. \end{cases} \tag{16}$$

G çizgesi için önerilen en son modülerlik fonksiyonu Denklem 17'de verilmiştir.

$$F_M = \frac{1}{2 \times m} \sum_{ij} \left(A_{(i,j)} - \frac{k_i \times k_j}{2 \times m} \right) \times \delta(C_i, C_j) \tag{17}$$

Burada F_M, maksimize edilecek amaç fonksiyonunu ve m, yönsüz bir ağ için Denklem 18'e göre hesaplanan toplam bağlantı sayısını temsil eder. k_i, *i.* düğümün derecesini, k_j ise *j.* düğümün derecesini gösterir. C_i ve C_j sırasıyla; *i* ve *j* düğümlerinin ait oldukları modülleri gösterirler. $\delta(C_i, C_j)$ ise bu düğümlerin aynı modülde bulunup bulunmadığını gösteren kontrol fonksiyonunu ifade eder ve bu fonksiyon Denklem 19'a göre 0 ya da 1 çıktısı verir.

$$m = \frac{1}{2} \sum_{ij} A_{(i,j)} \tag{18}$$

$$\delta = \begin{cases} 1 & eğer \;\; C_i = C_j, \\ 0 & eğer \;\; C_i \neq C_j, \end{cases} \tag{19}$$

Modülerlik en fazla kullanılan amaç fonksiyonu olsa da bilinen iki önemli kısıta sahiptir (Labatut ve Balasque 2012). Kısıtlardan ilkinde, bu fonksiyon maksimum değeri dikkate alınan ağ yapısına oldukça bağlıdır. Bu bağlılık farklı ağlar arasında elde edilen değerleri karşılaştırmayı imkansız hale getirebilir. İkincisi ise bu fonksiyonun bir çözünürlük sınırına/kısıtına sahip olmasıdır. Bu durum ağdan kendi yapısına bağlı olarak eşik bir sınır değerinden daha küçük bir modülün ortaya çıkarılamaması sorununa sebep olur (Fortunato ve Barthelemy 2007, Labatut ve Balasque 2012). Amaç fonksiyonlarının bunlara benzer kısıtlarının olması sebebiyle çizgelerin farklı özelliklerini referans alan fonksiyonların dikkate alınması önemlidir. Bu sebeple bu çalışmada, modülerlik dışında dokuz farklı amaç fonksiyonu daha ele alınmış ve test sonuçları üzerinden analizler ve çıkarımlar yapılmıştır.

2.1.3.2. İletkenlik (*C, Conductance*)

İletkenlik, yerel ağ topluluk tespitinde birçok algoritma tarafından kullanılan popüler bir amaç fonksiyonudur (Leskovec ve ark 2010, Yang ve Leskovec 2012 (a)).

Bu fonksiyon, ağ modülünün dışında kalan kısımların toplam bağlantı oranını ölçer ve Denklem 20'deki gibi hesaplanır. Bu denklemdeki *Conductance* fonksiyonu, F_C ile temsil edilir.

$$F_C = \frac{c_S}{2 \times m_S + c_S} \tag{20}$$

$G(V, E)$ yönsüz çizgesinde, V düğüm kümesinin toplam sayısı $n = |V|$ ile gösterilsin. E bağlantılarının toplam sayısı ise $m = |E|$ ile temsil edilsin. Sırasıyla; S, modülü temsil eden düğümler kümesini; c_S, S kümesinin sınırındaki bağlantıların sayısını ve m_S, bu kümedeki toplam bağlantı sayısını gösterir. $m_S = \{(u, v) : u, v \in S\}$ kümesinden oluşur ve *u*-*v* ikilisi, S kümesindeki düğümleri gösterir. Bu denkleme göre grup içindeki ve grup dışındaki düğümler ayrı ayrı hesaplanır. Böylece tüm modüllerdeki düğümler dikkate alınarak minimize edilmesi amaçlanan iletkenlik fonksiyonu skoru elde edilmiş olur. Bu yaklaşıma göre bir düğümler topluluğunun modül olarak kabul olarak edilebilmesi için bu düğümlerin modül dışındaki düğümlerle bağlantısının olabildiğince düşük olması gerekir. Denklem 20'deki işlem tüm ağ toplulukları için uygulanır ve sonuçlar toplanır.

2.1.3.3. İç Yoğunluk (*ID, Internal Density*)

Bu fonksiyon ağ modüllerindeki bağlantıların iç yoğunluğunu temel alır (Leskovec ve ark 2010) ve tez çalışmasında F_{ID} ile gösterilir. F_{ID} değerinin hesaplanabilmesi için gereken denklem aşağıda verilmiştir. Bu denklemde, sadece tek bir modül için yapılan işlem gösterilmiştir. Fonksiyonun değeri tüm modüller dikkate alınarak hesaplanır.

$$F_{ID} = 1 - \frac{m_S}{n_S \times (n_S - 1)/2} \tag{21}$$

Denklem 21'de sunulan formüle göre bir ağdaki yoğunluk oranı, örnek bir S kümesindeki bağlantı sayısının (m_S), tüm düğümler arasındaki olası bağlantıların toplam sayısına $(n_S \times (n_S - 1)/2)$ bölünmesine bağlıdır. Böylece iç yoğunluk skoru, yoğunluk oranının 1'den çıkarılmasıyla elde edilir. Deneylerde yönsüz ağların kullanılması sebebiyle burada ikiye bölme işlemi gerçekleştirilmiş ve böylece iki kez

hesaba katılan bağlantılar sadece bir kez dikkate alınmıştır. Bu fonksiyonun *iletkenlik* (*C*) fonksiyonundaki gibi minimize edilmesi amaçlanır.

2.1.3.4. Global Yoğunluk (*GD, Global Density*)

Global yoğunluk kalite fonksiyonu (Mitalidis ve ark 2014) referanslı çalışmada sunulan ve literatürde çok fazla kullanılmayan bir amaç fonksiyonudur. Bu fonksiyon "*ağlardaki topluluklar, kendileri dışındaki ağın kalanıyla kıyaslandığında daha yoğun bir şekilde bağlantılara sahip olan düğümlerin alt-kümeleri olarak tanımlanabilir*" yaklaşımını (Danon ve ark 2005) temel alır. Global yoğunluk fonksiyonu Denklem 24'teki F_{GD} ile temsil edilir. Bu fonksiyon, global olarak tanımlanan iç yoğunluk (*internal density*)—$F_{GD}^{iç}$ ve dış yoğunluk (*external density*)—$F_{GD}^{dış}$ fonksiyonlarının birlikte optimize edilmesi mantığına dayanır.

$$F(A,v)_{GD}^{iç} = \frac{\sum_{a=1}^{M}\sum_{i\in v_a}\sum_{j\in v_a} A_{i,j}}{\sum_{a=1}^{M}|v_a|^2} \tag{22}$$

$$F(A,v)_{GD}^{dış} = \frac{\sum_{a=1}^{M}\sum_{i\in v_a}\sum_{j\in (v-v_a)} A_{i,j}}{\sum_{a=1}^{M}|v_a| \times |v-v_a|} \tag{23}$$

Burada *A*, *G* çizgesinin komşuluk matrisini; *i-j* ikilisi, seçilen düğümün matristeki indislerini; *v*, verilen çizgeden elde edilen topluluk yapısını veya modülleri temsil eden vektörü; *M* ise *v* vektöründeki maksimum sayıyı temsil eder. Denklem 22'de sunulan $F(A,v)_{GD}^{iç}$, ağdaki iç bağlantıların toplamını, tüm kümelerin toplam alanına böler. Denklem 23'teki $F(A,v)_{GD}^{dış}$ fonksiyonu ise aynı işlemi dış bağlantılar için gerçekleştirir.

$$F_{GD} = \frac{1}{2}\left[F(A,v)_{GD}^{iç} + 1 - F(A,v)_{GD}^{dış}\right] \tag{24}$$

Hem iç hem de dış yoğunluk değerleri 0 ile 1 arasındadır. Global iç yoğunluk fonksiyonu için 1 değeri en iyi sonucu gösterirken; aksine global dış yoğunluk fonksiyonu için en iyi sonuç 0'dır. Denklem 24'te önceki iki denklemden elde edilen değerlerin sonuçlarına göre nihai F_{GD} değeri elde edilir. Bu değerin optimum olması düğümleri birbirlerine yoğun olarak bağlı modüllere sahip bir çizgenin global yoğunluk

sonucunun 1'e eşit veya buna yakın bir değerde olmasıyla elde edileceğini anlamına gelir.

2.1.3.5. Kesme Oranı (*CR, Cut Ratio*)

Cut ratio veya *external density* (dış yoğunluk) diye isimlendirilen bu fonksiyon, F_{CR} ile gösterilir ve ağın ayrılabilirlik ölçüsünü belirler. F_{CR} bir çizgenin olası toplam bağlantı sayısına göre bir alt-çizgenin (S) dış bağlantılarının oranını yüzde olarak ifade eder (Wei ve Cheng 1989, Leskovec ve ark 2010). Bu fonksiyonla ilgili hesaplama Denklem 25'e göre yapılır. Bu formüldeki işlemler her ağ modülü için yapılır ve sonuçlar toplanarak F_{CR} değeri elde edilmiş olur.

$$F_{CR} = \frac{c_S}{n_S \times (n - n_S)} \tag{25}$$

Burada n, G yönsüz ağındaki toplam düğüm sayısını ve c_S, S alt-çizgesinin sınırındaki bağlantıların sayısını gösterir ve $c_S = |\{(u, v): u \in S, v \notin S\}|$ şeklinde ifade edilir. n_S ise S alt-çizgesindeki toplam düğüm sayısını temsil eder. *Cut ratio* değerinin olabildiğince düşük olması elde edilen ağ modüllerinin bir o kadar başarılı bir kümeleme neticesinde ortaya çıkarıldığını gösterir.

2.1.3.6. Normalize Edilmiş Kesim (*NC, Normalized Cut*)

Bu fonksiyon, çizge teorisinde bir çizgenin köşelerinin (*vertices*—düğümler) iki ayrı bölüme ayrılması olarak ifade edilen kesmelerle ilgili özellikleri kullanan amaç fonksiyonunu ifade eder. Burada, bir kesimin olabildiğince geniş olması yerine mümkün olduğunca küçük olması amaçlanır. Buna göre herhangi bir modül/topluluk tespit problemi, minimum kesim (*minimum cut*) (Ford ve Fulkerson 1956) yaklaşımıyla çözülebilir. Bu yaklaşım, kesme oranı (Wei ve Cheng 1989) ve normalize edilmiş kesim (Shi ve Malik 1997, 2000) gibi fonksiyonların odak noktasını oluşturur.

Bu amaç fonksiyonu, gruplamanın çizge teorisine dayalı formülasyonunu kullanır ve çizgeyi birkaç alt kümeye (modüller) ayıran bir *minimum kesim* elde etmeyi amaçlar. Bu özelliği kullanan formül F_{NC} ile gösterilir ve Denklem 28'e göre hesaplanır (Shi ve Malik 2000, Leskovec ve ark 2010). Normalize edilmiş kesim fonksiyonu

sıklıkla görüntü segmentasyonunda kullanılır ve bu amaç fonksiyonunun optimizasyonu problemi *NP-tam* türü bir problem olarak dikkate alınır (Blake ve Zisserman 1987, Fortunato 2010).

$$cut(Q,P) = \sum_{u \in Q, v \in P} w(u,v) \tag{26}$$

Verilen *G* çizgesi birbirinden ayrık iki gruba bölünebilir—(*Q* ve *P*). *Q* ve *P* parçaları arasındaki farklılık derecesi silinen bağlantıların toplam ağırlığı olarak hesaplanır ve bu değer $cut(Q,P)$ ile elde edilir. Denklem 26'da verilen $w(u,v)$, *u* ve *v* düğümleri arasındaki benzerliği gösteren bir fonksiyondur.

$$assoc(S,V) = \sum_{u \in S, t \in V} w(u,t) \tag{27}$$

Denklem 27'deki $assoc(S,V)$, *S* kümesindeki düğümleri birleştiren bağlantıların toplam ağırlığını verir. Burada *S*, *Q* veya *P* gruplarından seçilen kümeyi; *V*, *Q* ile *P*'nin kesişimini ve F_{NC} ise minimize edilecek amaç fonksiyonunu ifade eder.

$$F_{NC} = \frac{cut(Q,P)}{assoc(Q,V)} + \frac{cut(Q,P)}{assoc(P,V)} \tag{28}$$

F_{NC} fonksiyonunun minimize edilmesi, herhangi bir çizgeden elde edilen alt-gruplar arasındaki ayrışmayı en aza indirgeyen ve doğal olarak bu gruplar arasındaki ilişkiyi en üst düzeye çıkaran bir bölümün tespit edilmesi anlamına gelir.

2.1.3.7. Uygunluk Skoru (*FS, Fitness Score*)

Bu tez çalışmasında kullanılan diğer bir amaç fonksiyonu *Uygunluk Skoru*'dur ve bu fonksiyon farklı çalışmalarda (Lancichinetti ve ark 2009, Pizzuti 2009, Deng ve ark 2015, Wu ve Pan 2015) değişik isimlerle ifade edilmiştir. Bu uygunluk skoru ölçütü Denklem 29'da F_{FS} ile temsil edilmektedir (Kaur ve ark 2016). Sırasıyla d_S ve k_S parametreleri *S* alt-çizgesinin üyesi olan düğümlerin iç ve dış derecelerinin toplamını saklar. α ise modüllerin boyutunu kontrol eden çözünürlük parametresini gösterir (Lancichinetti ve ark 2009).

$$F_{FS} = \sum_{i \epsilon S} \frac{{d_S}^i}{\left({d_S}^i + {k_S}^i\right)^{\alpha}} \quad (29)$$

Bu denkleme göre F_{FS}'nin değeri maksimuma ulaştığında, her bir modül için iç bağlantıların yoğunluğu maksimum olurken; modüller arası yoğunluk minimum olur.

2.1.3.8. Ortalama Çıkış Derecesi Fraksiyonu (*ODF, Average Out Degree Fraction*)

Ortalama *ODF*, *S* alt-kümesinin dışına işaret eden bir bölgedeki düğüm bağlantılarının ortalama fraksiyonunu ölçer (Leskovec ve ark 2010). Daha açık bir ifadeyle, *S* alt-kümesindeki herhangi bir düğümün toplam bağlantıları üzerinden topluluğun dışındaki bağlantıların kesiri (*oranı*) hesaplanır. Bu amaç fonksiyonu F_{ODF} ile temsil edilir ve Denklem 30'a göre hesaplanır (Flake ve ark 2000, Leskovec ve ark 2010).

$$F_{ODF} = \frac{1}{n_S} \sum_{u \in S} \frac{|\{(u,v): v \notin S\}|}{d_u} \quad (30)$$

Burada (u, v) ikilisi çizgedeki düğümleri; n_S, *S* alt-kümesindeki düğümler toplamını; d_u, u düğümünün derecesini temsil eder. Bu fonksiyona göre ağın en uygun modül yapısının ortaya çıkarılabilmesi için F_{ODF} skorunun en küçük değerini öneren bir kümelemenin sağlanması amaçlanır.

2.1.3.9. Anlamlılık (*S, Significance*)

Çizgede bulunan anlamlı alt-ağların tespiti için alt-çizge olasılıklarına dayanan *Significance* amaç fonksiyonu, *Traag* ve diğerleri (Traag ve ark 2013) tarafından önerilmiştir. Standart ölçütlerde, bu fonksiyonun (F_S) optimizasyonuyla mükemmel bir performansın elde edildiği bilgisi ilgili makalede belirtilmiştir (Traag ve ark 2013). Bu optimizasyon temelli amaç fonksiyonu yoğun modül veya modüllerin rastgele bir çizgede bulunma olasılıklarına bakar ve bu özelliğiyle modülerlik (Newman 2004 (a), Newman ve Girvan 2004 (b)) fonksiyonu da dahil olmak üzere diğer birçok amaç fonksiyonundan farklılık gösterir.

$$p_v = \frac{m_v}{\binom{n_v}{2}} \tag{31}$$

$$D(p_v \parallel p) = p_v \, log \frac{p_v}{p} + (1 - p_v) \, log \frac{(1-p_v)}{(1-p)} \tag{32}$$

Burada *v*, ağ modülünü temsil eden vektör olsun. p_v, bu vektörün yoğunluğunu gösterir ve Denklem 31'deki formüle göre hesaplanır. n_v ve m_v sırasıyla bu vektördeki düğüm ve bağlantı sayılarını gösterirler. Denklem 32'de verilen $D(p_v \parallel p)$, iki olasılık dağılımı arasındaki doğal uzaklık metriğini—*Kullback-Leibler (KL) divergence* (bilgi için (Cover ve Thomas 2012) referanslı çalışmaya bakınız) temsil eder. Bu değer verilen iki olasılıksal dağılımın birbirlerine olan uzaklığının bir ölçüsüdür. Bu denklemdeki *p* ise verilen çizgenin yoğunluğunu temsil eder. Modülün yoğunluğu ile çizgenin yoğunluğunun *KL* uzaklık metriği fonksiyonuna girdi olarak verilmesiyle elde edilen sonuç Denklem 33'te gösterilen F_S amaç fonksiyonu için giriş parametresini oluşturur. Bu şekilde en uygun modüllerin tespiti için *anlamlılık* değeri arttırılır ve bu işlem ağdaki tüm olasılıksal modüllerin sayısı kadar sürdürülür.

$$F_S = \sum_{v} \binom{n_v}{2} D(p_v \parallel p) \tag{33}$$

2.1.3.10. Sürpriz (*Sr, Surprise*)

Newman tarafından sunulan modülerlik fonksiyonu gibi çeşitli global amaç fonksiyonlarının maksimize edilmesine dayanan çizge bölümleme yöntemlerinin tüm ağın boyutuna göre tanımlanmış bir ölçekten daha küçük olan modülleri tespit etmede yetersiz kaldıkları veya başarısız oldukları bilinmektedir (Nicolini ve Bifone 2016). Bu problem çözünürlük kısıtından kaynaklanır. *Nicolini* ve *Bifone* tarafından sunulan bir çalışmada (Nicolini ve Bifone 2016), bu kısıtın etkilerinden bahsedilmiş ve yakın zamanda önerilen olasılık teorisine dayalı bir uygunluk fonksiyonu olan *Surprise* ölçütünün bu kısıtlamalardan etkilenmediği belirtilmiştir. Ayrıca bahsi geçen çalışmada, bu fonksiyonun mevcut birçok fonksiyona göre daha uygun sonuçlara ulaştığı gözlemlenmiştir. Bu tez çalışmasında, diğer dokuz fonksiyon için yapılan analizler gibi *Sr* fonksiyonunun da çıktıları dikkate alınarak ulaşılan performansın değerlendirilmesi

amaçlanmıştır. Buradaki kıyaslama işlemi tüm amaç fonksiyonları için ayrı ayrı gerçekleştirilmiştir.

Surprise amaç fonksiyonu küme başına benzer düğüm dağılımına sahip elde edilmiş bir kümeleme düzeninin rastgele bir ağa göre istatistiksel olarak ne kadar düşük olasılık taşıdığını gösterir (Del Ser ve ark 2016).

(Aldecoa ve Marin 2011) referanslı çalışmadan yararlanılarak *Surprise* (F_{Sr}) uygunluk değerinin hesaplanması için gereken formül Denklem 34'te verilmiştir. Bu denkleme göre simetrik olarak birbirine bağlı birimler içeren yönsüz bir ağın en uygun topluluk yapısının ortaya çıkarılması F_{Sr} parametresinin maksimize edilmesine eşdeğerdir.

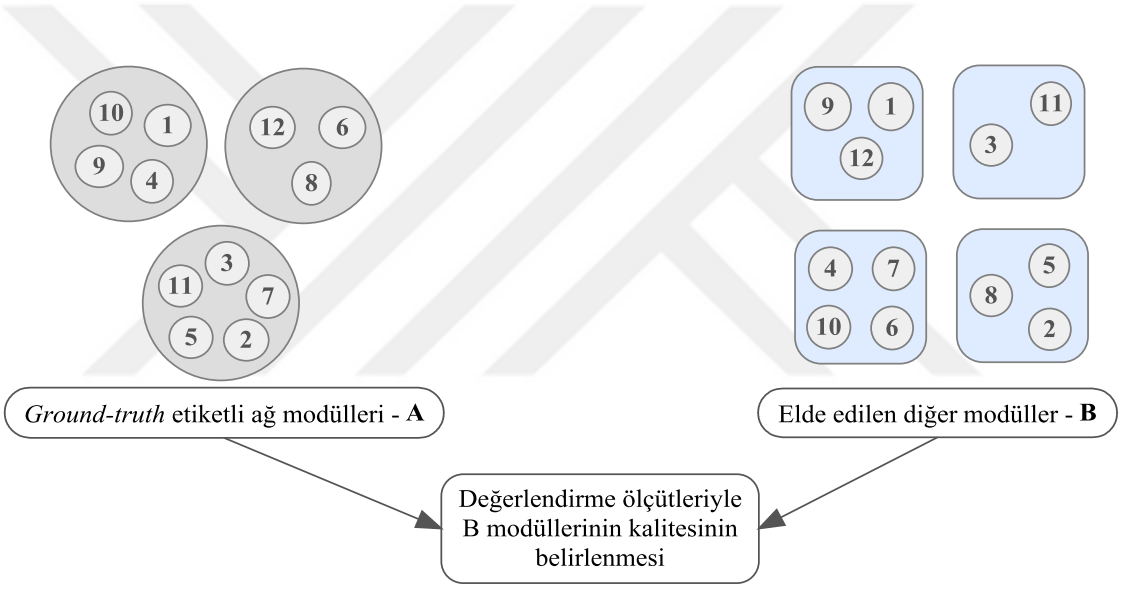
$$F_{Sr} = -log \sum_{i=k}^{min(M,n)} \frac{\binom{M}{i}\binom{T-M}{n-i}}{\binom{T}{n}} \tag{34}$$

Burada T, bir çizgedeki olası en fazla bağlantı sayısını; n, gözlenen bağlantı sayısını; M, belirli bir bölüme ait modül içi bağlantıların maksimum olası sayısını ve k ise bu bölümde aktif olarak gözlemlenen modül içi bağlantıların toplam sayısını gösterir. Ayrıca, Denklem 34'te verilen ve *sürpriz* anlamına gelen F_{Sr} parametresi rastgele üretilmiş bir çizgedeki iç bağlantıların olabildiğince zenginleştiği bir bölümü tesadüfen bulmanın "*surprise—olası olmama*" durumunu ölçer. Bu çalışmada son amaç fonksiyonu olarak *Surprise* fonksiyonu önerilmiş ve bu fonksiyonun sonuç değerinin optimize edilebilmesi için Bölüm 3.4'teki yaklaşımlardan en uygun olanı kullanılmıştır.

2.1.4. Değerlendirme ölçütleri

Bölüm 2.1.3'te verilen amaç fonksiyonları ile elde edilen tüm ağ modüllerinin uygunlukları bu bölümde sunulan altı adet değerlendirme ölçütünün yardımıyla test edilmiştir. Bu test bize amaç fonksiyonlarının kalite açısından değerlendirilebilmesi ve birbirleriyle kıyaslanabilmesi olanağı sağlamıştır. Böylece bu çalışmada önerilen en uygun algoritma en iyi uygunluk fonksiyonuyla kullanılabilecek ve sonraki bölümlerde sunulan daha büyük ve karmaşık ağlardaki kaliteli modüllerin ortaya çıkarılmasına yardımcı olabilecektir.

Herhangi bir ağdan elde edilen modüller veri üzerinde daha önceden belirlenmiş referans modül yapıları (*ground-truth*) ile karşılaştırılıp bu modüllerin hangilerinin uygun olduğuna dair karar verilebilmektedir (Arınık ve ark 2015). "*Ground-truth*" terimi, bir test sonucunun doğrulanması amacıyla tercih edilen kesin referansı (Demir 2015) ifade eder. Şekil 2.5'te sunulmuş *10* düğümlü bir ağ için kesin referanslı topluluklar/modüller (*A*) ve herhangi bir algoritma ile elde edilen temsili topluluklar/modüller (*B*) verilmiştir. Bu bölümde sunulan değerlendirme ölçütleri ile Şekil 2.5'teki gibi bir karşılaştırmada elde edilmiş olan toplulukların kaliteleri ölçülür. Böylece çeşitli amaç fonksiyonlarının kullanılması sonucu elde edilmiş ağ topluluklarının uygun olanları hem kesin referanslı topluluklar hem de aşağıda sunulan değerlendirme ölçütlerinin yardımlarıyla belirlenir.



Şekil 2.5. Kesin referanslı ağ modülleri ile diğer modüllerin temsili karşılaştırılması

Karşılaştırma test verisi olarak kullanılan *ground-truth* etiketli bir ağdan elde edilen modüller daha öncesinden bilinen en iyi referans sonuçlarıyla karşılaştırılır ve böylece önerilen yöntemin başarısı test edilir. Bu doğrulama yaklaşımı özellikle çeşitli kümeleme yöntemlerinden elde edilen sonuçların doğruluğunun gösterilmesinde kullanıldığı gibi modül veya topluluk bulma algoritmalarının karşılaştırılması için de uygulanmaktadır (Yang ve ark 2012 (b), Fu ve ark 2017). Bu amaçla Bölüm 3.1.3'te test verileri sunulmuş ve bu bölümdeki değerlendirme fonksiyonları göre elde edilen sonuçlar da Bölüm 4.1.2'de ayrıntılı olarak analiz edilmiştir. Bu bölümde incelenen fonksiyonlar aşağıda verilmiştir.

Bu fonksiyonlardan ilki *Mutual Information* (*MI*—karşılıklı bilgi) isimli fonksiyondur (Shannon 1997, Cover ve Thomas 2012). Bu fonksiyon iki rastgele parametre verisi arasındaki entropik korelasyonun bir ölçütünü belirler. Olasılık ve bilgi teorisinde, iki farklı giriş değişkeninin *MI* sonucu bu değişkenler arasındaki karşılıklı bağımlılığının bir ölçüsünü gösterir. Bu çalışmada, farklı kümeleme sonuçlarını tutan bilgilerin *karşılıklı bilgi* değerini hesaplayan fonksiyon E_{MI} ile temsil edilmiştir. *K* ve *L* ile temsil edilen iki ayrık dizi değişkenin karşılıklı bilgi değeri Denklem 35'e göre hesaplanır. Burada *k* ve *l* parametreleri sırasıyla; *K* ve *L* dizilerindeki indis bilgilerini tutarlar. $P(k,l)$, *K* ve *L* değişkenlerinin ortak olasılık fonksiyonunu gösterirken; $P(k)$ ve $P(l)$ fonksiyonları ise bu değişkenlerin marjinal olasılık dağılımlarını gösterirler.

$$E_{MI} = \sum_{l \in L} \sum_{k \in K} P(k,l) \log_2 \frac{P(k,l)}{P(k)P(l)} \tag{35}$$

Denklem 35'e göre hesaplanan E_{MI} değerinin yüksek olması iki değişken arasındaki belirsizliğin veya farklılığın önemli ölçüde az olduğunu; düşük ise tam tersi sonucun elde edildiğini gösterir. İki değişken arasındaki karşılıklı bilgi değerinin sıfır olması, bu değişkenlerin birbirlerinden bağımsız olduğu anlamına gelir. Sonucun *1*'e yakınlaşması ise benzerliğin maksimum oranda olduğunu belirtir.

Bir sonraki ölçüt ise *Normalized Mutual Information* (*NMI*—normalize edilmiş karşılıklı bilgi) fonksiyonudur (Vinh ve ark 2010). Bu fonksiyon kümeleme veya topluluk/modül algılama gibi alanlarda yaygın olarak kullanır. Bu çalışmada, *NMI* değeri Denklem 36'da sunulan $\mathrm{E_{NMI}}$ ile temsil edilir. $\mathrm{E_{NMI}}$, C_1 ve C_2 olarak temsil edilen iki farklı kümenin benzerlik sonuçlarını *0* ile *1* arasında ölçeklendiren *MI* (Cover ve Thomas 2012) skorunun normalleştirilmiş halini ifade eder (Vinh ve ark 2010, Sun ve ark 2014). Sırasıyla; *N*, konfüzyon matrisini; $N_{i,j}$, C_1 ve C_2 alt-kümelerinde bulunan toplam düğüm sayısını; N_i^s, *N*'nin *i.* sıradaki toplamını; N_j^s, *N*'nin *j.* sütunundaki toplamını ifade eder.

$$E_{NMI} = \frac{-2 \sum_{i,j} N_{i,j} \, log\left(\frac{N_{i,j} \times N}{N_i^s \times N_j^s}\right)}{\sum_i N_i^s \, log(N_i^s/N) + \sum_j N_j^s \, log(N_j^s/N)} \tag{36}$$

Rand Index—RI (rastgelelik indeksi) (Rand 1971, Hubert ve Arabie 1985, Wagner ve Wagner 2007) üçüncü değerlendirme ölçütü olarak sunulmuş ve bu çalışmada E_{RI} ile temsil edilmiştir. Rastgelelik indeksi 0 ile 1 arasında bir değer alır. 0 değeri iki veri kümesinin herhangi bir düğüm çiftinde aynı olmadıklarını; 1 değeri ise bu veri kümelerinin tam olarak aynı olduğunu gösterir. Hesaplama formülü Denklem 37'de sunulmuştur.

$$E_{RI} = \frac{n_{00} + n_{11}}{n_{11} + n_{00} + n_{10} + n_{01}} \tag{37}$$

Burada verilen n_{00}, n_{01}, n_{10} ve n_{11} parametreleri, düğüm çiftlerinin iki ayrı kümedeki durumlarına göre elde edilen sayıları tutarlar. Bu amaçla S_{00}, S_{01}, S_{10} ve S_{11} kümeleri tanımlanmış olsun. S_{11} kümesi hem tahmini topluluğun hem de referans topluluğunun aynı alt kümelerinde birden bulunan düğüm çiftlerini içerir. S_{00} kümesi bu iki topluluğun farklı alt kümelerinde bulunan tüm düğüm çiftlerinin listesini tutar. S_{10} kümesi ise ilk topluluğun aynı alt kümesinde olup ikinci topluluğun farklı bir alt kümesinde bulunan düğüm çiftleri saklar. Son olarak S_{01} kümesi ise ilk topluluğun farklı; ikinci topluluğun aynı alt kümelerinde olan düğüm çiftlerini tutar. Sonuç olarak; $n_{11} = |S_{11}|$, $n_{00} = |S_{00}|$, $n_{10} = |S_{10}|$ ve $n_{01} = |S_{01}|$ şeklinde belirtilen kümelerdeki düğüm çiftlerinin toplam sayıları elde edilmiş olur. E_{RI} ölçütünün skoru herhangi bir algoritma ile elde edilen ağ topluluk yapısı ve referans topluluk yapısı tarafından benzer şekilde kümülatif olarak sınıflandırılmış düğüm çiftlerinin oranını temsil eder. Böylece, verilen bir çift düğüm için işleme tabi tutulan iki farklı topluluk yapısına göre bu iki düğümün de aynı toplulukta olması veya ikisinin de farklı topluluklarda bulunması durumunda bir kümeleme uyumundan bahsedilebilir (Arınık ve ark 2015).

Dördüncü değerlendirme kriteri olarak *Adjusted Rand Index* (*ARI*—düzenlenmiş rastgelelik indeksi) fonksiyonu önerilmiştir. Bu fonksiyon *RI* (Rand 1971) ölçütünün düzenlenmiş versiyonu olarak görülebilir ve tez çalışmasında E_{ARI} ile gösterilmiştir. Bu fonksiyonun formülü Denklem 38'de verilmiştir. Bir *G* çizgesi için *n* = |*G*| olacak şekilde *n*, toplam eleman sayısını ifade eder. *i*, C_1 kümesi için; *j*, C_2 kümesi için indisleri gösterir ve L_1, C_1 için; L_2, C_2 için maksimum sınır değerlerini tutar. M_{ij}, konfüzyon matrisini ya da diğer bir ifade ile olasılık tablosunu (Wagner ve Wagner 2007) ifade eder. İki küme karşılaştırıldığında *RI*, sıfır ile bir arasında değer alırken; *ARI* bunlardan

farklı olarak negatif değerler alabilir. Böyle bir durum, mevcut indeks değeri beklenen indeks değerinden daha az olduğunda gerçekleşebilir.

$$E_{ARI} = \frac{2\left(n(n-1)\left(\sum_{i=1}^{L_1}\sum_{j=1}^{L_2}\binom{M_{ij}}{2}\right) - 2\left(\sum_{i=1}^{L_1}\binom{|C_{1,i}|}{2}\sum_{j=1}^{L_2}\binom{|C_{2,j}|}{2}\right)\right)}{n(n-1)\left(\sum_{i=1}^{L_1}\binom{|C_{1,i}|}{2} + \sum_{j=1}^{L_2}\binom{|C_{2,j}|}{2}\right) - 4\left(\sum_{i=1}^{L_1}\binom{|C_{1,i}|}{2}\sum_{j=1}^{L_2}\binom{|C_{2,j}|}{2}\right)} \quad (38)$$

Beşinci kriter olarak *Jaccard Index* (*JI*—Jaccard indeksi) seçilmiştir ve formülü Denklem 39'da verilmiştir. *JI*, farklı çalışmalarda *Jaccard Similarity Index* veya *Jaccard Similarity Coefficient* olarak da isimlendirilmiştir. Bu fonksiyon iki gruptaki üyelerin aynı kümede bulunup bulunmadığına göre bu grupları birbirleriyle karşılaştıran bir yöntem sunar. Yani, her iki gruptaki düğümlerden aynı kümede sınıflandırılan düğüm çiftlerinin sayısının en az bir gruptaki düğümlerden aynı kümede sınıflandırılan düğüm çiftlerinin sayısına oranı Jaccard indeksi (E_{JI}) sonucunu verir (Fortunato 2010).

$$E_{JI} = \frac{J_{C_1,C_2}}{J_{C_1,C_2} + J_{C_1} + J_{C_2}} \quad (39)$$

Denklem 39'daki E_{JI} değeri, sıfır ile bir arasındadır. Sırasıyla, J_{C_1,C_2}, her iki gruptaki düğümler için aynı kümede bulunan düğüm çiftlerinin sayısını; J_{C_1}, ilk gruptaki düğümlerden aynı kümede bulunan düğüm çiftlerinin sayısını; J_{C_2}, ikinci gruptaki düğümlerden aynı kümede bulunan düğüm çiftlerinin sayısını gösterir.

Son olarak, bu çalışmada *permanence* (*P*—kalıcılık) ölçütü ele alınmakta ve Denklem 40'ta E_P olarak temsil edilmektedir. Bu kriter ağlardan elde edilen alt-kümelerin uygunluğunu değerlendirmek için önerilen düğüm tabanlı yeni bir yaklaşım sunmaktadır. Bu yaklaşım kesin referanslı alt-kümelerin kalitesi ile uygun bir korelasyona sahiptir ve alt-kümelerdeki bozulmalara karşı duyarlıdır (Chakraborty ve ark 2014).

$$E_P(v) = \frac{X_{int}}{X_{ext} \times d} + U_{in} - 1 \quad (40)$$

Denklem 40'ta, v düğümüne göre *kalıcılık* skorunu temsil eden $E_P(v)$'nin hesaplaması verilmiştir. Bu formül, verilen alt kümelerin iç ve dış bağlantılarına göre hesaplanır. Sırasıyla; X_{int}, v düğümünün ait olduğu alt kümedeki düğümlerle olan iç

(*internal*) bağlantıların sayısını; X_{ext}, kendisi dışındaki ve birbirinden bağımsız alt kümelerle olan maksimum bağlantı sayısını; d, v'nin toplam derecesini; U_{in}, v'nin aynı modüldeki komşu düğümlerine göre iç kümeleme katsayısını ifade eder. Ayrıca verilen formülde bazı kısıtlar bulunmaktadır. İlk kısıtta, eğer v düğümünün harici bağlantıları yoksa X_{ext} değeri sıfır olacağından bu denklemdeki formülün hesaplanabilmesi için değer bir olarak alınır. Diğer bir kısıt ise her modülün minimum üç düğüm ve üç dahili bağlantı içermesi gerekliliğidir. Aksi durumda, U_{in} değerinin 0 olduğu varsayılır (Chakraborty ve ark 2014).

2.2. Kanser Verilerinde Sağkalım ile İlişkili Alt-Ağların Belirlenmesi

Tez çalışmasının bu bölümünde sağkalım süresi (*survival time*) ile ilişkili kopya sayısı değişikliklerine sahip büyük ve karmaşık gen etkileşim ağlarından anlamlı alt-ağların ortaya çıkarılması problemine odaklanılmıştır. *Hansen* ve *Vandin* tarafından sunulan çalışmada (Hansen ve Vandin 2016) bu problem, *NP-zor* kategorisinde tanımlanmış ve sağkalımla ilişkili hesaplama probleminin teorik altyapısı açıklanmıştır. Bu amaçla literatürde yaygın olarak kullanılan *log-rank test istatistiği* iki farklı gruba ayrılmış hasta popülasyonunun sağkalım parametresini karşılaştırmak için kullanılır. İlgili çalışmada (Hansen ve Vandin 2016), istatistiksel test skorunun kullanılmasıyla ilgili ayrıntılı bilgiler sunulmuştur.

Bu bölümde tanımlanan problem, genler arası etkileşim ağlarındaki sağkalım ile ilişkili *CNA*'lara sahip gen gruplarının tanımlanması hakkındadır. Bu problemi tanımlamadan önce somatik kopya sayısı değişiklikleri—*CNAs* hakkında gerekli bilgiler aşağıda verilmiştir.

Genomik değişiklikler; (1) kopya sayısı değişiklikleri, (2) mutasyonlar, (3) *mRNA* veya *miRNA* ekspresyonu değişiklikleri ve (4) protein/fosfoprotein seviyesindeki değişiklikler şeklinde dört adet veri türünde gruplandırılabilir (Gao ve ark 2013). Tez çalışmasında, genlerdeki *CNA*'lar dikkate alınmıştır. Şekil 2.6'da örnek olarak sunulan klinik verilerdeki genler, *CNA*'lar olarak isimlendirilen somatik türdeki değişikliklere sahiptirler. Bu tür değişikliklerde (somatik türde mutasyonlar gibi), hücrenin kalıtım malzemesi olan *deoksiribonükleik asit* (*DNA*)'nın çoğaltılması sırasındaki bir hata/değişiklik sonucu bir genin birden çok kopyalanması veya azaltılması durumu ortaya çıkar. Meydana gelen değişiklikler somatik değişiklikleri, yani bir organizmayı

oluşturan üreme hücreleri dışındaki hücrelerde meydana gelen değişiklikleri ifade ederler. *CNA*'lar, kanser genomunda yaygın bir olgudur (Yuan ve ark 2012). Somatik *CNA*'lar (*CNAs*) neredeyse tüm insan kanser türlerinde genomun her tarafına yayılmıştır (Beroukhim ve ark 2010). *cBioportal* isimli web platformunda (Cerami ve ark 2012, cBioPortal 2016) elde edilen *TCGA* verilerindeki *CNA* bilgileri ve temsil edilen değerler 5 grupta kategorize edilir (cBioPortal 2016): homozigot silme—*homozygous deletion* (-2), hemizigöz silme—*hemizygous deletion* (-1), nötr/değişimsiz—*neutral/no change* (0), artış/kazanç—*gain* (+1), yüksek seviyeli amplifikasyon—*high-level amplification* (+2). Bu tez çalışmasında hastaların gen bilgilerindeki değişiklikleri temsil eden *CNA* değerleri beş kategori yerine iki kategoride ele alınmıştır. Eğer gen bilgisinde *CNA* değeri 0 (nötr) ise bu bilgi 0; diğer durumlarda 1 olarak alınmıştır. Yani *CNA* bilgisi 1 ise hastanın ilgili geninin somatik değişikliğe sahip olduğu; 0 ise ilgili genin somatik değişiklik içermediği anlaşılır.

Bu bölümde sunulan problem sağkalım analizi ile ilgilidir. Sağkalım analizini temel alan bilgisayımsal problem ile bu problemin hesaplama karmaşıklığı aşağıda sunulmuştur.

Bu tür bir analiz yaklaşımında, P_0 ve P_1 isimli iki adet hasta popülasyonu tanımlansın ve i, hastayı tanımlayıcı *ID* numarasını temsil etsin. Her bir popülasyondaki hastalar $(i \in P_0 \cup P_1)$ kendi sağkalım süresine ve sansürlü bilgiye (*censoring information*) sahip olsun. Sağkalım süresi ve sansürlü bilgi sırasıyla; t_i ve c_i ile temsil edilmiş olsun. Burada c_i bilgisi için Denklem 41'deki eşitlik verilebilir.

$$c_i = \begin{cases} 0, & eğer\ \langle hasta, \boldsymbol{alt\ sınır}\ t_i\ sağkalım\ süresine\ sahipse \rangle, \\ 1, & eğer\ \langle hasta, \boldsymbol{tam}\ t_i\ sağkalım\ süresine\ sahipse \rangle, \end{cases} \tag{41}$$

$c_i = 0$ durumunda, hastanın **sansürlü bilgiye** sahip olduğu; $c_i = 1$ durumunda ise hastanın tam bir sağkalım süresine ve **sansürsüz bilgiye** sahip olduğu anlaşılır.

Sırasıyla; m_0 ve m_1 parametreleri, P_0 ve P_1 popülasyonlarındaki hasta sayılarını temsil eder ve deneylerdeki toplam hasta sayısı ise $m = m_0 + m_1$ ile elde edilir. Örnek olarak, hastalar $\{1, 2, 3, \dots, m\}$ numaraları ile gösterilirken; sağkalım süreleri *düşükten yükseğe* doğru $\{t = 1, 2, 3, \dots, m\}$ ile temsil edilebilir (Hansen ve Vandin 2016). Sonuç olarak, hastalar için sağkalım bilgileri c ve x isimli iki adet vektörde tutulur. Burada c_i, i. hasta için sansür bilgisini tutarken; x_i, hastanın popülasyon bilgisini gösterir. İlk

popülasyon (P_0) için bu değer 0 iken; ikinci popülasyon (P_1) için bu değer 1 olarak tutulur.

Verilen iki farklı popülasyondaki sağkalım bilgileri göz önüne alındığında, P_0 ve P_1 arasındaki sağkalım farklılığının anlamsal değerlendirilmesi için *log-rank testi* (Mantel 1966, Peto ve Peto 1972, Zırhlıoğlu ve Kara 2004, Harrington 2005, Karasoy 2008, Karasoy ve Tilki 2013) tercih edilebilir. Bu test iki adet popülasyonun veya grubun sağkalım dağılımlarını karşılaştırmak için kullanılan ve parametrik olmayan bir hipotez testidir. *Log-rank* özellikle kanser vakaları gibi klinik araştırmalarda çokça tercih edilen istatistiksel test türüdür. Log-rank istatistik testi *Hansen* ve *Vandin* tarafından sunulmuş bir çalışmada (Hansen ve Vandin 2016) açıklanan ve bu bölümde de özet bilgilerle tanıtılan probleme özgü bir ölçüttür.

$$V_{x,c} = \sum_{j=1}^{m} c_j \times \left(x_j - \frac{m_1 - \sum_{i=1}^{j-1} x_i}{m - j + 1} \right) \tag{42}$$

$$\sigma_{x,c} = \sqrt{\frac{m_0 \times m_1}{m \times (m-1)} \times \left(\left(\sum_{j=1}^{m} c_j \right) - \left(\sum_{j=1}^{m} c_j \times \frac{1}{m - j + 1} \right) \right)} \tag{43}$$

$$p = \frac{V_{x,c}}{\sigma_{x,c}} \tag{44}$$

P_0 ve P_1 grupları arasındaki sağkalım bilgilerinde herhangi bir fark bulunmamasını temsil eden sıfır hipotezi altında; problemin çözümünde amaç bu iki grup arasında makul bir skor elde etmektir. Bu makul skor maksimum değeri gösterir. Log-rank istatistik skoru Denklem 44'teki p ile temsil edilmekte ve bu skorun sonucu Denklem 42'ye ve Denklem 43'e göre belirlenmektedir. Hesaplamalar sonucu elde edilen sonuç *normalleştirilmiş log-rank istatistiği skorunu* ifade eder. Denklem 43'teki formül ile standart sapma değeri hesaplanır. Literatürde, log-rank istatistik dağılımının normal yaklaşımı için *permütasyonel* ve *koşullu* olmak üzere iki farklı standart sapma türü önerilmiştir. Tez çalışmasında, *Hansen* ve *Vandin* tarafından kullanılan ve bu problemin yapısına uygun olarak tercih edilen permütasyonel dağılımlı standart sapma türü tercih edilmiştir. Bunun sebebini Hansen ve Vandin, (Vandin ve ark 2015) referanslı çalışmadan da yararlanarak permütasyonel dağılımlı standart sapma türünün

genomik çalışmalarda daha uygun olması şeklinde açıklamışlardır (Hansen ve Vandin 2016). Denklem 44'teki formül kullanılarak verilen iki farklı grubun sağkalım bilgilerine göre aralarındaki farkın maksimize edilmesi amaçlanır. Böylece genellikle sansürlü ve daha az sağkalım süresine sahip hastalar (P_0 popülasyonu) ile sansürsüz ve daha uzun sağkalım süresine sahip hastalar (P_1 popülasyonu) arasındaki en uygun eşik sınırın belirlenmesine çalışılır.

Bu tez çalışmasındaki genomik verilerde, n adet gen içeren etkileşim ağı G ile gösterilir. m adet hasta içeren örnekler kümesi P değişkeniyle ve *CNA* durumlarını gösteren matris bilgisi ise M ile temsil edilir. Bu matris Denklem 45'teki eşitlikten yararlanılarak oluşturulur.

$$M_{i,j} = \begin{cases} 0, & eğer \quad j.hastanın\ i.geni\ CNA'lı\ değilse, \\ 1, & eğer \quad j.hastanın\ i.geni\ CNA'lı\ ise, \end{cases} \tag{45}$$

P_0 ve P_1 popülasyonlarındaki hastalarda bulunan genlerin *CNA* durumlarına göre belirlenen S alt-kümesi, gen etkileşim ağında sağkalım parametresine göre kanserle maksimum ilişkiye sahip olduğu düşünülen genlerin listesini tutar. Bu alt-küme, aşağıda verilen P_0^S ve P_1^S listelerine göre elde edilir.

$$\begin{aligned} P_0^S &= \{j \in P : \textstyle\sum_{i \in S} M_{i,j} = 0\} \\ P_1^S &= \{j \in P : \textstyle\sum_{i \in S} M_{i,j} \geq 1\} \end{aligned} \tag{46}$$

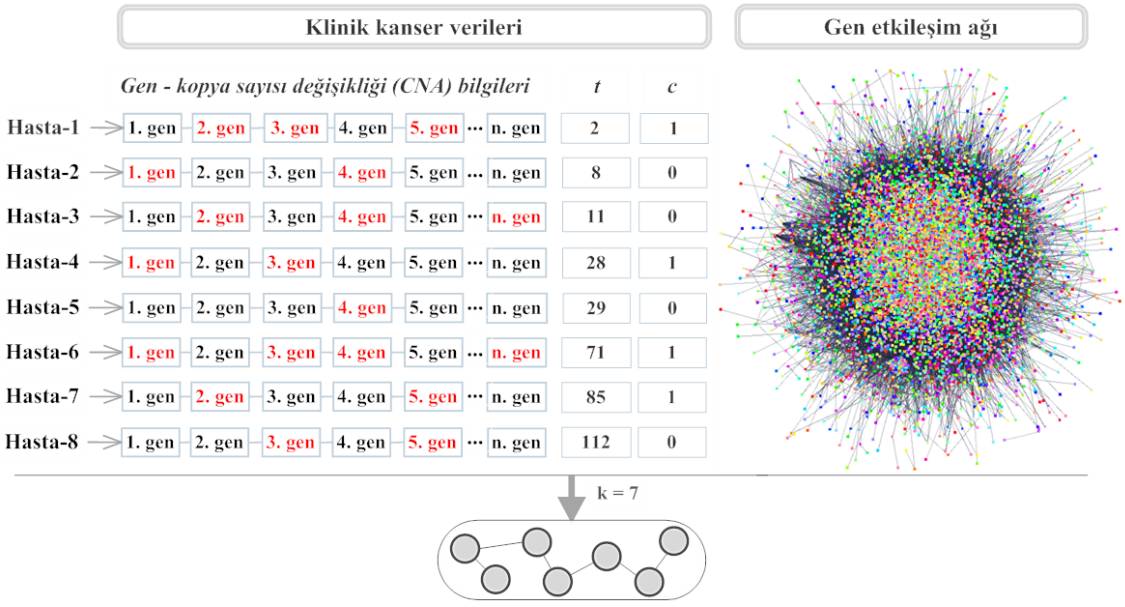
$S \subset G$ gen alt-kümesi seçildikten sonra bu genlerde hiç *CNA* içermeyen hastalar ($\sum_{i \in S} M_{i,j} = 0$) Denklem 46'da gösterilen P_0^S alt-popülasyonunu; bu genlerden en az birinin *CNA*'lı olması durumunda bu hastalar ($\sum_{i \in S} M_{i,j} \geq 1$) P_1^S alt-popülasyonunu oluşturur. Bu bölümde ilgilenilen problemde, G'deki tüm genlerin genomik kopya sayısı değişikliklerinin (*CNAs*) mevcut durumları göz önüne alınarak; $|S| = k$ adet gen grubuna sahip $S \subset G$ kümesinin bulunması amaçlanır. Burada k isteğe bağlı olarak belirlenen ve sağkalım parametresiyle maksimum ilişkide olduğu düşünülen gen sayısını ifade eder. S kümesi normalize edilmiş log-rank istatistiğinin mutlak değerini maksimize eden k adet *CNA*'lı gen içerir.

Verilen S alt-kümesi için x^S vektörü hem P_0 hem de P_1 gruplarındaki hastaların gen ve *CNA* bilgilerine göre oluşturulan vektörü temsil eder. Örneğin; j. hasta için S'de belirlenen genlerden en az biri *CNA*'lı ise $x_j^S = 1$ olurken; aksi durumda $x_j^S = 0$ olur.

Böylece Denklem 46'daki P_0^S ve P_1^S hasta popülasyonları göz önüne alındığında, x^S vektörüne göre normalize edilmiş log-rank istatistik skoru Denklem 47'ye göre hesaplanır. Denklem 44'te sunulan ilk hesaplamadaki c sansür bilgisi hasta verisi olarak sabit alınırsa yeni hesaplama,

$$p_{skor} = \frac{V_x s}{\sigma_x s} \tag{47}$$

şeklinde olur. Burada sırasıyla $V_x s$ ve $\sigma_x s$ değerleri, Denklem 42 ve Denklem 43'e göre x^S girdi vektörü dikkate alınarak hesaplanır. Ele alınan problem için p_{skor} ile ilgili verilen Denklem 47'deki fonksiyon, *uygunluk fonksiyonu* olarak kabul edilir ve bu değerin maksimize edilmesi amaçlanır. Maksimum k kümeli log-rank problemi (*max k-set log-rank problem*) şeklinde ifade edilen bu problemin ilgili çalışmada ((Hansen ve Vandin 2016) referanslı çalışmaya bakınız) *NP-zor* kategorisinde olduğu teorik bilgilerle açıklanmıştır. Ayrıca, birbirine bağlı alt bileşenler içeren bu problemin çizgeler üzerindeki hesaplama maliyetinin büyüklüğü de yine problemin oldukça karmaşık olduğunu açıklar niteliktedir. Bu problemin temsili gösterimi için Şekil 2.6'da giriş ve çıkış verileri gösterilmiştir. Bu temsili şekil (Hansen ve Vandin 2016) referanslı makaleden ve (Vandin 2016) referanslı sunumdan yararlanılarak hazırlanmıştır.



Şekil 2.6. Problemin temsili gösterimi ile giriş ve çıkış verileri

Şekil 2.6'da n adet gene sahip etkileşim ağı ve 8 adet hastaya sahip klinik verilerle ilgili temsili bir grafik sunulmuştur. Buradaki gen etkileşim ağı klinik hasta verilerinden temin edilen gen listesinin *STRING* veri tabanı (bakınız (Szklarczyk ve ark 2014, Szklarczyk ve ark 2016, string-db 2018)) yardımı ile görselleştirilmiş ağ yapısını temsil eder.

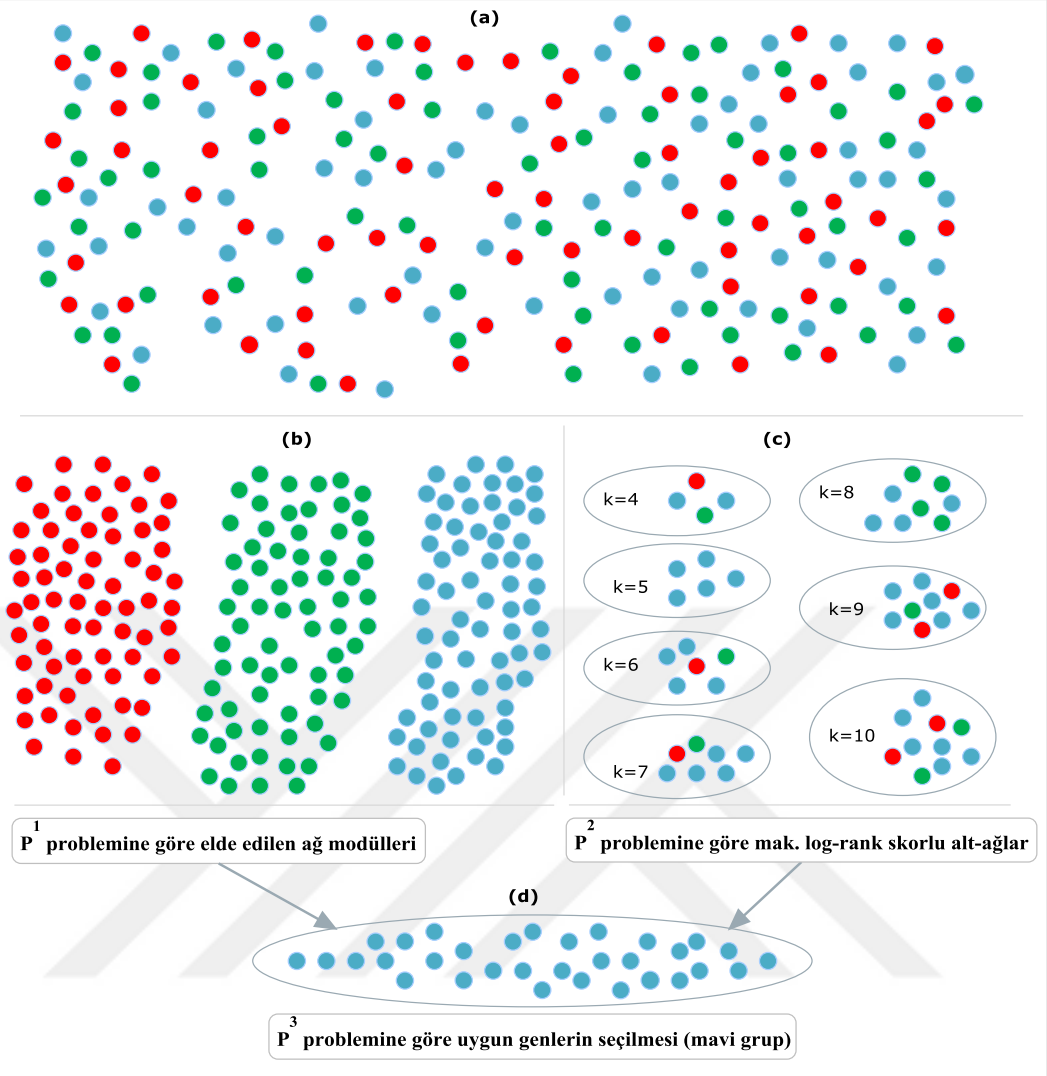
Klinik verilerden sağkalım süresi ve sansürlü bilgi dahil diğer tüm hasta bilgileri *"http://www.cbioportal.org/data_sets.jsp"* web sitesinden temin edilmiştir (cBioPortal 2016). Bunlar, *TCGA* veri kümeleridir ve bu veri kümeleri hakkında ayrıntılı bilgiler Bölüm 3.3'te verilmiştir. Şekil 2.6'da giriş verilerinin analizi sonucu en yüksek *log-rank* skoruna sahip $k = 7$ adet gene sahip temsili bir alt-ağ elde edilmiştir.

Şekil 2.6'daki grafikte örnek olarak verilen hastalar için kırmızı renkli genler *CNA* içeren; siyah renkli genler *CNA* içermeyen genleri ifade eder. Ayrıca t ve c sembolleri sırasıyla; hastanın sağkalım süresi ve sansürlü bilgi durumunu ifade eder. Bir hastanın sansürlü bilgi değeri 1 ise yani hasta sansürsüz bilgiye sahipse sağkalım süresinin tam olduğu; 0 ise hastanın tüm gözlem süresince çeşitli sebeplerle bu işlemleri tamamlamadan ayrıldığı ve tam bir sağkalım süresine sahip olmadığı yani hastanın sansürlü bilgiye sahip olduğu anlaşılır. *Log-rank* fonksiyonunun çıktı değerinin yüksek olabilmesi için olabildiğince sansürsüz ve yüksek sağkalım süresine sahip hastaların bir alt-popülasyonda; sansürlü ve daha az sağkalım süresine sahip hastaların ise diğer alt-popülasyonda gruplanması amaçlanır. Buna göre Şekil 2.6'daki örnek için sağkalım parametresine göre kanser ile maksimum ilişkide olduğu düşünülen yedi adet gen, birbirine bağlı düğümlerle temsil edilerek çıktı olarak verilmiştir. Şekil 2.6'da örnek olarak verilen hastaların *CNA*'lı değişiklik bilgileri, sağkalım bilgileri ve gen etkileşim ağları giriş verileri olarak alınmaktadır. Bu verilere göre Bölüm 3.5'te sunulan *DP* ve *GA* temelli iki farklı yaklaşımdan seçilen algoritmaya özgü bir çıkış verisi elde edilir. Bu çıkış verisi p_{skor} değişkeninde tutulur. Buradaki p_{skor} değerinin maksimize edilmesi amaçlanır ve en son değere göre ilgili kanser hastalığı ile en fazla ilişkili k adet genin/proteinin listesi çıktı olarak sunulur.

2.3. Ağ Modüllerinin ve Sağkalımla İlişkili Alt-Ağların Birlikte Değerlendirilmesi

Bu bölümde, Bölüm 2.1'de ve Bölüm 2.2'de sunulan problemlerin birlikte ele alındığı yeni bir problem üzerine odaklanılmıştır. Bu problem iki temel adımdan

oluşmaktadır. İlk adım Bölüm 2.1'de sunulan problemdeki çözüme uygun bir şekilde giriş verisi olarak alınan gen-gen etkileşim ağlarının uygun modül yapılarının ortaya çıkarılmasıdır. Bu adımdaki işlemler Bölüm 3.3'te verilen *LAML-net*, *HNSC-net*, *KIRC-net* ve *STAD-net* biyolojik ağları için uygulanmıştır. Ağ modül yapıları ortaya çıkarılan bu ağlar için aynı zamanda ikinci adımda sağkalım analizi işlemleri uygulanır. Böylece ikinci adım gerçek klinik veriler için *maksimum k-kümeli log-rank probleminin* uygulanmasıyla farklı k değerlerine sahip alt-ağların elde edilmesi işlemlerini kapsar. Bu iki adımdan sonra ilgili kanser türü ile ilişkili kaydedilen tüm alt-ağlardaki genlerin büyük çoğunluğunun dahil olduğu ağ modülü belirlenir. Bu iki adım yeni problemi tanımlar. Burada hem aynı modülde bulunan hem de ilgili kanser türünde sağkalım süresine maksimum etki ettiği düşünülen genlerle ilgili analiz işlemine geçilir. Hem aynı modülde bulunup bağlantılarına göre birbirleriyle etkileşimde olduğu kabul edilen hem de giriş kanser türü ile maksimum ilişkili oldukları kabul edilen gen kümelerinin birlikte dikkate alındığı yeni problem için çözümler değerlendirilir. Burada, aynı modülde yoğun bağlantılarla birbirleriyle ilişkili olan genlerin birlikte ilgili kanser hastalığındaki etkilerinin analiz edilmesi hedeflenir. Ağ modül tespiti problemine göre belirlenen en uygun algoritma ile Bölüm 3.3'te verilen dört farklı klinik kanser verisinden elde edilen gen etkileşim ağları için en uygun modül yapılarının ortaya çıkarılması sağlanır. Her ağ modülüne göre k adet düğümden oluşan gen alt-ağlarının modülleri belirlenir. Burada; P^1, P^2 ve P^3 ifadeleri sırasıyla; Bölüm 2.1'deki, Bölüm 2.2'deki ve bu bölümdeki problemleri temsil ederler. P^3 probleminin temel çalışma mekanizması Şekil 2.7'de gösterilmiştir. İlk iki problemin birlikte uygulanmasını ifade eden üçüncü problemle ilgili test sonuçları Bölüm 4.2.2'de sunulmuştur. Giriş verilerine göre elde edilen çıktıların neler olduğu ilgili deneysel sonuçlar kısmında açıklanmıştır.



Şekil 2.7. P^3 problemi için temsili gösterim

Şekil 2.7-(a)'da üyeleri rastgele atanmış bir ağ yapısı verilmiştir. P^1 probleminin çözümünde belirlenen en başarılı algoritma (**a**)'da verilen ağın modül yapılarının ortaya çıkarılmasını sağlar. Bu ağ modül yapıları (**b**) sırasıyla; kırmızı, yeşil ve mavi renklerde sunulmuştur. Aynı renk üyelerin (aynı modüllerdeki) birbirleriyle yoğun işlevsel veya fiziksel bağlantılarının olduğu; diğer üyelerle (farklı modüllerdeki) daha az işlevsel veya fiziksel ilişkilerin olduğu varsayılır. Buna göre verilen örnek için P^1 probleminin çözümüyle önerilen yaklaşımın uygulanması sonrası tüm ağın üç farklı gruba ayrıldığı ağ modülleri elde edilir (**b**). Daha sonra temsili ağ için $k = 4, 5, 6, 7, 8, 9, 10$ sayılarında maksimum log-rank istatistik skorlu alt-ağlar (**c**) P^2 problemine göre ortaya çıkarılsın. Burada *maksimum k kümeli log-rank problemi* için seçilen k sayısı 7 farklı değer için uygulanmıştır. Bu parametre seçilen kanser türünde sağkalım süresiyle maksimum ilişkili genlerin sayısını ifade eder. Son aşamada ise kaydedilen farklı k değerlerine göre

elde edilen maksimum log-rank skorlu alt-ağlardaki her bir gen/düğüm kendi modül yapısına göre P^3 problemine uygun olarak gruplandırılır. Böylece **c**'deki alt-ağlarda bulunan düğümler, **b**'deki modül yapılarına göre aynı renkte (aynı modülde) olmak şartıyla; en fazla eleman sayısına ulaşan ağ modülündeki elemanlar, **d**'de seçilen grubu oluşturur. Bu şartı sağlayan en yüksek sayıdaki alt-ağ elemanları (örneğin; mavi renkli düğümler) P^3'teki seçilen grubu temsil eder. Şekil 2.7'deki örnek için alt-ağlardaki elemanlardan aynı modülde olan ve sağkalım süresine en yoğun etkisi olduğu varsayılan alt-ağlardaki elemanlardan aynı modülde bulunan en yüksek sayıdaki düğümlerin ait olduğu modül üçüncüsüdür (mavi grup). Buna göre **d**'deki küme toplam 32 adet düğümden oluşur. Ayrıca, seçilen genlerin (**d**'deki son grup) birbirleriyle ilişkilerinin analizi için genlerin en az üçünün etkili olduğu fonksiyonel eksiklikler veya hastalıklar "*http://cbdm-01.zdv.uni-mainz.de/~jfontain/cms*" isimli web sitesindeki *GS2D* modülü ile tespit edilmiştir (Fontaine ve Andrade-Navarro 2016, GS2D 2018).

Bu bölümdeki yeni problemin (P^3) ilk adımını ifade eden ağ modüllerinin tespiti probleminin (P^1) karmaşıklığı uygulanan çözümün yaklaşımına veya seçilen uygunluk fonksiyonuna bağlı olarak *NP-zor* veya *NP-tam*'dır. Bu bölümdeki problemin ikinci adımını temsil eden sağkalım ile maksimum ilişkili alt gen-ağlarının ortaya çıkarılması probleminin (P^2) karmaşıklığının da Bölüm 2.2'de *NP-zor* olduğu belirtilmiştir. Burada sunulan P^3 problemi için $P^1 \subset P^3$, $P^2 \subset P^3$ ve $P^3 = P^1 \cup P^2$ tanımlamaları yapılabilir. Böylece P^1 probleminin *NP-zor* veya *NP-tam* olduğu ve P^2 probleminin de yine *NP-zor* olduğu bildiğine göre P^3 probleminin hesaplama karmaşıklığı bu iki probleme göre belirlenebilir. Burada P^3 probleminin hesaplama karmaşıklığı K_{P^3} ile temsil edilsin ve minimum hesaplama maliyeti $K_{P^3} = \{(NP_{zor} \cap NP_{tam}) \cup (NP_{zor})\}$ olsun. Sonuç olarak, P^3 probleminin karmaşıklığının en azından *NP-zor* türü bir problem sınıfına dahil olduğu anlaşılır.

3. MATERYAL VE YÖNTEM

3.1. Gerçek Dünya Verileri

Bu çalışmada karmaşık ağlardan anlamlı modüllerin tespiti için testlerde kullanılacak gerçek dünya ağları iki ayrı kategoride ele alınmıştır. İlk kategori *Grup-A* ile temsil edilirken; ikinci kategori *Grup-B* ile ifade edilmiştir. İlk kategorideki ağlardan küçük ve orta boyutluların çoğu modül/topluluk tespiti konusunda oldukça fazla kullanılan verilerdir ve bunlar Çizelge 3.1'de sunulmuştur. Bu veriler önerilen algoritmaların başarı kıyaslamaları açısından önemlidir. Çünkü literatürde önerilen birçok algoritma bu ağlarla test edilmekte ve algoritmaların başarıları bu ağlardaki sonuçlar göz önüne alınarak değerlendirilmektedir. Tez çalışmasında bu ağlar düğüm ve bağlantı yoğunluklarına göre gruplandırılmış ve aşağıdaki alt başlıklarda sunulmuştur. *Grup-B* kategorisinde ise kesin referanslı veri kümeleri olarak etiketlenen ağlar incelenmiş ve bununla ilgili veriler kümesi Çizelge 3.2'de verilmiştir.

3.1.1. Grup-A kategorisi

Bu grup "*küçük ve orta boyutlu etkileşim ağları"* ve "*büyük ve karmaşık ağlar*" olarak iki ayrı alt başlıkta incelenmiştir.

3.1.1.1. Küçük ve orta boyutlu etkileşim ağları

Bu kategori *Zachary's Karate Club*, *Bottlenose Dolphins*, *American College Football*, *Books About US Politics*, *E. Coli Transcription* ve *Helicobacter Pylori Protein-Protein Interaction* (*PPI*) ağlarından oluşmaktadır. Ağların temsili kısaltmaları, düğüm—bağlantı sayıları ve ait oldukları kategorileri Çizelge 3.1'de verilmiştir. Bu çizelgedeki düğüm ve bağlantı sayıları ağların ağırlıksız ve yönsüz hale dönüştürülmesiyle ve döngülü düğümlerin listeden çıkarılmasıyla elde edilmiştir.

Çizelge 3.1. Küçük ve orta boyutlu gerçek dünya ağları

Ağlar		Düğüm sayısı	Bağlantı sayısı	Ağ kategorisi
Zachary's Karate Club	[A-1]	*34*	*78*	*sosyal*
Bottlenose Dolphins	[A-2]	*62*	*159*	*sosyal - ekolojik*
American College Football	[A-3]	*115*	*613*	*sosyal*
Books about US Politics	[A-4]	*105*	*441*	*sosyal*
E. Coli Transkripsiyon	[A-5]	*418*	*519*	*biyolojik - transkripsiyon*
Helicobacter Pylori PPI	[A-6]	*732*	*1403*	*biyolojik - protein etkileşim*

Bu gruptaki ilk test verisi bu tez çalışmasında [A-1] ile temsil edilen *Zachary's Karate Club* sosyal ağıdır. 1970'li yıllarda bir üniversite karate kulübündeki *34* adet üyenin birbirleriyle olan *78* adet etkileşimi *Wayne W. Zachary* tarafından ağ yapısı olarak sunulmuştur (Zachary 1977). Bu veri *karate* ismiyle de ifade edilir ve topluluk tespitinde en çok tercih edilen ağlardan biridir (karate 2017).

Bir diğer sosyal-ekolojik etkileşim verisi *Bottlenose Dolphins* ağıdır. Bu ağ birbirleriyle yoğun olarak ilişkili *62* adet şişe burunlu yunusun birbirleriyle oluşturdukları çok yönlü etkileşimleri temsil eder (Lusseau ve ark 2003). Bu yunuslarla ilgili gözlemler 1994 ile 2001 yılları arasında Yeni Zelanda'da yapılmıştır. Normalde yönlü bir ağ olan *dolphins* ağı, tez çalışmasında *159* adet bağlantıya sahip yönsüz bir ağa dönüştürülerek [A-2] ile temsil edilmiştir. Ayrıca bu ağ *dolphins* ismiyle de temsil edilir. *Dolphins* ile ilgili genel bilgilere (dolphins 2017) referanslı web adresinden ulaşılabilir.

Çizelge 3.1'de [A-3] ile gösterilen *American College Football* ağı yönsüz ve ağırlıksız bir ağdır. Bu ağ 2000 yılının güz döneminde *Division IA* kolejleri arasındaki Amerikan futbol oyunlarını içermektedir (Girvan ve Newman 2002). Bu veri deneysel çalışmalarda *football* ile ifade edilmiştir ve bu ağdaki düğümler toplam *115* tane kolej takımını temsil eder. Bağlantılar ise bu takımlar arasında gerçekleşen *613* adet sezonluk futbol oyunlarının eşleştirmelerini ifade eder.

Bu kategorideki dördüncü test ağı [A-4] kısaltmasıyla sunulan *Books About US Politics* etkileşim ağıdır (polbooks 2017). *Polbooks* ismi ile de temsil eden bu ağ çevrimiçi kitap satıcısı Amazon tarafından satılan ve son zamanların US politikasıyla ilgili kitaplarını içeren bir sosyal etkileşim ağıdır. Kitaplar arasındaki bağlantılar, aynı alıcılar tarafından ortak olarak satın alınan kitapların sıklığını ifade eder. Bu ağdaki düğüm ve bağlantı sayıları sırasıyla; 105 ve 441 olarak alınmıştır.

E. coli transkripsiyonel (hücrelerdeki gen ekspresyonunu) düzenleme ağı [A-5] topluluk yapılarının ortaya çıkarılmasında test ağı olarak yaygın bir şekilde kullanılır. Bu ağdaki düğümler operonları temsil eder. Düğümler arası her bir bağlantı bir transkripsiyon faktörünü kodlayan bir operondan direk olarak düzenlenen diğer bir operonda yönlendirilir (Shen-Orr ve ark 2002). Bu ağ toplam *418* düğüme ve *519* bağlantıya sahiptir.

Çizelge 3.1'de gösterilen son test verisi *Helicobacter Pylori PPI* (helico 2017) biyolojik etkileşim ağıdır. Proteinler arası fonksiyonel veya fiziksel etkileşimleri modelleyen bu tür ağlar canlılardaki enzim faaliyetlerinin düzenlenmesinden hücre döngülerindeki mekanizmaların kontrol edilmesine kadar birçok önemli süreçte doğrudan ya da dolaylı olarak görev alırlar. Tez çalışmasında odaklanılan ağ modül yapılarının tespit edilmesi probleminin testlerinde gerçek dünya ağlarından *Helicobacter Pylori* [A-6] canlısının protein etkileşimlerini modelleyen bir temsili ağ kullanılmıştır. Burada toplam düğüm sayısı ve döngü içermeyen bağlantı sayısı 732 ile 1403'tür.

3.1.1.2. Büyük ve karmaşık ağlar

Bu bölümdeki ağların düğüm-bağlantı sayıları, bir önceki bölümde sunulan ağların düğüm-bağlantı sayılarından daha fazladır ve bu ağlar daha karmaşık yapılara sahiptir. Ağlar hakkında kısa bilgiler aşağıda verilmiştir.

Ağların uygun topluluk yapılarının ortaya çıkarılması amacıyla bu tez çalışmasında önerilen metasezgisel yaklaşımlar bir önceki bölümde verilen gerçek dünya ağları ile Bölüm 3.2'de sunulan yapay ağlar üzerinde karşılaştırmalı testlere tabi tutulmuştur. *Rastgele üretilmiş yapay ağlar* bölümündeki veriler rastsal olarak üretilmiş ve çeşitli büyüklükteki test ağlarını temsil ederler. Tüm ağlardaki deneysel sonuçlara göre en uygun ve en başarılı sonuçlara ulaşan algoritma bu bölümde tanıtılan büyük ve karmaşık ağlarda başarı testine tabi tutulmuş ve bu algoritmanın ölçeklendirilebilirliği de test edilmiştir. Bununla ilgili test sonuçları Bölüm 4.1.1.3'de sunulan istatistiksel analiz kısmında verilmiştir. Bölüm 4.1.1.1 ve Bölüm 4.1.1.2'deki ilk test sonuçlarına göre seçilen algoritmanın büyük ağlardaki başarısının ölçülmesi amacıyla sonraki testte, *Facebook/Twitter/Google+ ego-networks* (Leskovec ve Mcauley 2012) bireysel ağları kullanılmıştır. Burada Facebook, Twitter ve Google+ web platformlarından temin edilen ve kişilerin sosyal çevreleri (*social circles*) hakkında bilgiler içeren veriler üzerinde

deneyler gerçekleştirilmiştir. Bir ego ağı her bir bireyin kendi arasında yoğun olarak etkileşimlere sahip olduğu alterler ile ilişkilerini modelleyen sosyal etkileşim ağıdır (Arnaboldi ve ark 2016). Bu tür ağlarda "*ego*" sosyal ağdaki düğümdür. Özellikle online sosyal ağ analizinde "*ego*" odaklanılan bireyi temsil ederken; *alterler* bu bireyin çevresini tanımlar. Bu ağlarda egolar özellikli sosyal çevrelere sahiptir.

Facebook ve Google+ ağlarındaki düğüm özellikleri; arkadaşlık bilgileri, kişi eğilimleri, meslekleri, cinsiyetleri, aile bilgileri, kültürleri, yaşam standartları ve ilgi alanları gibi spesifik özellikli kullanıcı profillerinden gelir. Twitter'da ise düğüm özellikleri çevrimiçi web platformunda kullanıcı tarafından kendi tweet'lerinde kullanılan hashtag'lar ile tanımlanır. Bu ağlardaki düğüm ve bağlantı sayıları; Facebook için 4039/88234; Twitter için 81306/1768149 ve Google+ için 107614/13673453'tür (SLNDC-sosyal ego ağları 2017). Bu ağlardan yönsüz ve ağırlıksız olan Facebook ego ağı 10 adet ayrık ağdan oluşur. Fakat bu çalışmada bu ego ağları birleştirilerek tek bir ağ olarak testlerde kullanılmıştır. Diğer iki ağ ise normalde yönlü ağlardır (SLNDC-sosyal ego ağları 2017). Bu ağlar tez çalışmasında yönsüz olacak şekilde yeniden düzenlenmiştir. Ayrıca, bu ağlarda kendi kendine döngülere ve birden fazla bağlantıya da izin verilmemiştir.

3.1.2. Grup-B kategorisi

Deneysel çalışmalardan elde edilen ağ modül yapıları herhangi bir veri kümesi üzerinde daha önceden tanımlanmış kesin referanslı modül yapıları ile karşılaştırılıp bu modüllerin uygunluğuna yönelik değerlendirmeler yapılabilmektedir. Bununla ilgili "*ground-truth*" ifadesi herhangi bir test sonucunun (özellikle kümeleme işlemlerinde) doğrulanması amacıyla başvurulan kesin referansı (Demir 2015) gösterir. Bu bölümde kullanılan test ağları Çizelge 3.2'de sunulmuş olup; bu ağlardaki modüllerin sayıları, modüllerin üyeleri ve en uygun kümeleme sonuçlarını gösteren uygunluk değerleri bilinmektedir. Bu verilerle ilgili ayrıntılı bilgilere ulaşmak için *"Neural Information Processing Systems"* isimli konferansta sunulan (Yang ve ark 2012 (b)) referanslı makalenin *"Suppemental Document"* isimli eki incelenebilir. Buradaki ağlar üzerinden gerçekleştirilen deneysel çalışmalarla en uygun çözümü sunan amaç fonksiyonunun elde edilmesi amaçlanmıştır.

Çizelge 3.2. Kesin referanslı test ağları (Yang ve ark 2012 (b))

		# düğüm sayısı	# bağlantı sayısı	# modül sayısı	Ağ türü	Veri kaynağı
Strike	[B1]	*24*	*38*	*3*	*Sosyal*	*PAJEK*
Amlall	[B2]	*38*	*190*	*3*	*Gen*	*AMLALL*
Duke	[B3]	*44*	*220*	*2*	*Tıbbi*	*LIBSVM*
Khan	[B4]	*83*	*415*	*4*	*Gen*	*ELML*
Cancer	[B5]	*198*	*990*	*14*	*Tıbbi*	*ELML*
Ecoli	[B6]	*327*	*1635*	*5*	*Protein*	*UCI*
Ionosphere	[B7]	*351*	*1755*	*2*	*Radar*	*UCI*
Yeast	[B8]	*1484*	*7420*	*10*	*Biyoloji*	*UCI*

3.2. Rastgele Üretilmiş Yapay Ağlar

Bu bölümde rastsal bir şekilde oluşturulan yapay çizgelerin üretilmesinde tercih edilen iki farklı model hakkında bazı bilgiler verilmiştir. Bunlar: Erdos–Renyi (*ER*) ve Barabasi-Albert (*BA*) modelleridir. Bu iki model yardımıyla üretilen yapay ağlar Bölüm 3.4'te sunulan algoritmaların performans karşılaştırmasında kullanılmıştır. Bu ağlar üzerinde modül tespiti problemine (P^1 *problemi*) göre önerilen algoritmalarla testler gerçekleştirilmiş ve en yüksek amaç fonksiyonu değerine ulaşan algoritma ya da algoritmaların belirlenmesi hedeflenmiştir. Bu amaçla hem *ER* hem de *BA* yaklaşımlarına göre 100, 250, 500 ve 1000 düğümden oluşan yapay ağlar üretilmiştir. Bu ağlar basit, yönsüz ve bağlı çizgeler türündedir. Üretilen ağlarla ilgili genel özellikler aşağıda verilmiştir.

3.2.1. Erdos–Renyi modeli

Erdos ve *Renyi* tarafından 1959 yılında tanıtılan yeni bir yaklaşıma göre rastgele üretilen basit çizgeler kavramı literatüre sunulmuştur (Erdos ve Renyi 1959). Sunulmuş olan bu yaklaşım tez çalışmasında, *ER* modeli olarak isimlendirilmiştir. Bu modelde her bir düğümün olası kenar sayısı için p olasılık değeri tanımlanır. Bu değer iki düğüm arasında bir bağlantının olup olmadığı ihtimalini belirtir. Bu model ile yönsüz yapay ağlar üretilir.

Verilen pozitif n düğüm sayısı ve $0 \leq p \leq 1$ arasında tanımlanan olasılık değeri için $G(n, p)$ yönsüz çizgesi tanımlansın. Burada n adet düğüm dikkate alınarak; her bir

$v - w$ düğüm çifti arasında p olasılıklı (v, w) bağlantısı bulunur. Bu metoda göre üretilen tüm düğümler için bağlantı sayısı eşit olur. Böylece belli bir düğüm veya düğümler grubunun etrafında yoğunluk olmaz. Her bir kaynak düğüm için hedef düğüm rastsal bir şekilde seçilir. Bu tez çalışmasında ağlardaki topluluk tespitinde önerilen algoritmaların daha nesnel olarak test edilebilmesi için *ER* modeline göre *4* adet rastgele ağ üretilmiştir. Burada amaç, gerçek dünya ağlarının dışında farklı boyutlarda ve farklı topolojik özelliklere sahip ağlarda algoritmaların performanslarının analiz edilmesidir. Bu yaklaşıma göre oluşturulan ağlar ve genel özellikleri Çizelge 3.3'te verilmiştir.

Çizelge 3.3. *ER* modeline göre üretilen yapay ağlar

Temsili ağ İsmi	***Düğüm sayısı***	***Bağlantı sayısı***	***Ortalama derece***	***Ağ çapı***	***Ağ bağlantı yoğunluğu***	***Kümeleme katsayısı***	***Ortalama yol uzunluğu***
ER-1	100	318	6.360	5	0.064242	0.067709	2.671
ER-2	250	644	5.152	7	0.020691	0.023166	3.549
ER-3	500	1547	6.188	7	0.012401	0.008223	3.603
ER-4	1000	3270	6.540	7	0.006547	0.005652	3.888

3.2.2. Barabasi-Albert modeli

BA modeli, Albert-Laszlo Barabasi ile Reka Albert tarafından önerilen ve tercihli bir bağ kurma mekanizması kullanarak rastgele ölçeksiz (*random scale-free*) ağlar üretmek için kullanılan bir yaklaşımdır (Albert ve Barabasi 2002). *BA* yaklaşımı bağlantı sayısı dağılımını daha gerçekçi bir şekilde gerçek dünya ağlarındaki gibi modeller. *ER* modelinde bir çizgenin üretilmesi tüm düğümlerin eşit özellikli olarak aynı anda oluşturulması şeklinde gerçekleşirken; *BA* modelinde bundan farklı olarak düğümler çizgeye birer birer dahil edilirler. Her bir yeni düğüm çizgeye mevcut düğümlerin sahip olduğu bağlantı sayısı ile orantılı olarak eklenir. Bu metoda göre bağlantı sayısı yüksek olan düğümler yeni eklenen düğümler için daha yüksek olasılıklı etkileşim ihtimaline sahip olur. Bundan dolayı bağlantı sayısı daha fazla olan düğümlerin bağlantı sayısı çizgenin büyüklüğüne bağlı olarak daha fazla artar. Tıpkı *ER* modeliyle üretilen rastgele ağlar gibi *BA* modeline göre de dört adet test ağı üretilmiştir. Üretilen rastgele ağlar ve bunların genel karakteristik özellikleri Çizelge 3.4'te sunulmuştur.

Hem *ER* hem de *BA* modellerine göre üretilen ağlardaki testlere göre bu tez çalışmasında önerilen tüm algoritmaların başarı karşılaştırmaları ve değerlendirmeleri deneysel çalışmalar bölümünde sunulmuştur. Her iki ağ üretim modeliyle de farklı topolojik ve karakteristik özellikli ağlar oluşturulmuştur. Böylece önerilen algoritmalar gerçek dünya ağları dışında da test edilmiştir. Bunun sonucunda tüm deneysel sonuçlar ağ özelliklerinden bağımsız olarak daha nesnel bir yaklaşımla analiz edilmiştir.

Çizelge 3.4. *BA* modeline göre üretilen yapay ağlar

Temsili ağ İsmi	*Düğüm sayısı*	*Bağlantı sayısı*	*Ortalama derece*	*Ağ çapı*	*Ağ bağlantı yoğunluğu*	*Kümeleme katsayısı*	*Ortalama yol uzunluğu*
BA-1	100	296	5.920	5	0.059798	0.088355	2.751
BA-2	250	757	6.056	6	0.024321	0.037686	3.247
BA-3	500	1506	6.024	6	0.012072	0.013821	3.633
BA-4	1000	3031	6.062	7	0.006068	0.005941	3.968

3.3. Kanserle İlişkili Klinik Veri Kümeleri

Bu bölümde *Glioblastoma Multiforme* (*GBM* 2016), *Acute Myeloid Leukemia* (*LAML* 2016), *Head and Neck Squamous Cell Carcinoma* (*HNSC* 2016), *Kidney Renal Clear Cell Carcinoma* (*KIRC* 2016) ve *Stomach Adenocarcinoma* (*STAD* 2016) olarak isimlendirilen klinik kanser veri kümeleri hakkında bilgiler verilmiştir. Bu kanser türleri hakkında daha ayrıntılı bilgilere (Cerami ve ark 2012, Gao ve ark 2013, cBioPortal 2016, TCGA 2016, *TCGA geçici-2016*) referanslı çalışmalardan ve web sitelerinden ulaşılabilir. Ayrıca, bu verilerle ilgili ek tanımlayıcı bilgiler Bölüm 2.2'de sunulmuştur.

Klinik hasta verilerinde sağkalım parametresini en fazla etkileyen genlerin tespiti problemi (P^2) ile ilgili test çalışmaları bu bölümde verilen beş adet gerçek dünya verisi üzerinde gerçekleştirilmiştir. Her bir veri seti klinik kanser verisine ve bu kanser verileriyle ilgili gen etkileşim ağına sahiptir. Kanser verileri hastalara ait sağkalım süresi, sansürlü bilgi, cinsiyet, *CNA*'lı genler ve mutasyon sayısı gibi bilgiler sunar. Bu bilgilerin hangi aşamalarda kullanıldığı Bölüm 2.2'de açıklanmıştır. Tüm kanser türlerine göre hastaların genel klinik bilgileri Çizelge 3.5'te sunulmuştur. Hastaların sağkalım süreleri orijinal hallerinde ay bilgisine göre sunulmaktadır. Tez çalışmasında bu süre bilgisi günlük olarak sunulmuştur. Bu çizelgedeki ortalama sağkalım süresi tüm hastaların yaklaşık olarak gün bazındaki sağkalım sürelerinin ortalamasını gösterir.

Buradaki yaklaşık ifadesi ise aylık sürelerin günlük olarak sunulmasında kalanların (*saatler*) en yakın ondalık sayıya yuvarlanması işlemini ifade eder. Klinik hasta bilgileri ile ilgili olarak orijinal verilerdeki toplam hasta sayıları Çizelge 3.5'te *L* ile temsil edilmiştir. Tüm klinik veriler problemin yapısına uygun hale getirilmiş ve buradaki veri hazırlama işlemi *filtreleme* olarak isimlendirilmiştir. Veri hazırlama işlemleri sonunda elde edilen toplam hasta sayıları *T* ile gösterilirken; cinsiyet bilgilerine göre hasta sayıları erkekler için *E*, kadınlar için *K*, cinsiyet bilgisi bulunmayan hastalar için *B* ile temsil edilmiştir. Sansürlü bilgilerle ilgili sansürlü hasta sayısı *S*, sansürsüz hasta sayısı *Y* ve sansür bilgisi olmayan hasta sayısı *Z* ile gösterilmiştir. Sansür bilgileri orijinal verilerdeki toplam hasta sayısına göre sunulmuştur. Her bir kanser türü için nihai (*veri hazırlama sonrası*) toplam hasta sayısı *T* ile gösterilmiştir. *T* değerleri belirlenirken ilk aşamada, hastaların en az bir *CNA*'lı gen içerip içermediği kontrol edilir. Hastanın genomik yapısında hiç *CNA*'lı gen bulunmuyorsa bu hasta listeden çıkarılır. İkinci aşamada ise sansür bilgisi belirsiz (*Z*) olan hastaların listeden çıkarılması işlemi gerçekleştirilir. Sonuç olarak, iki aşamada toplam hasta sayısı (*T*) belirlenmiş olur. Tüm verilerin saf ve orijinal hallerine (*GBM* 2016, *HNSC* 2016, *KIRC* 2016, *LAML* 2016, *STAD* 2016) referanslı web adreslerinden ulaşılabilir.

Glioblastoma Multiforme veya *Glioblastoma* (Chin ve ark 2008, *GBM* 2016) bir tür beyin tümörüdür ve GBM ile temsil edilir. Bu hastalık türünde toplam 206 hasta verisi bulunmaktadır. Tüm hastaların en az bir *CNA*'lı gene sahip olduğu bilinmektedir. Bu verilere GMB ağ verisinin küçük olmasından dolayı herhangi bir filtreleme uygulanmamıştır.

Acute Myeloid Leukemia (Ley ve ark 2013, *LAML* 2016), Akut Miyeloid Lösemi olarak isimlendirilir ve LAML kısaltmasıyla gösterilir. Bu kanser türünde toplam hasta sayısı 200'dür. Burada hiç *CNA*'lı gen içermeyen hastalar listeden çıkarıldıktan sonra toplam 191 hasta bilgisi elde edilmiştir.

Head and Neck Squamous Cell Carcinoma (Lawrence ve ark 2015, *HNSC* 2016), HNSC kısaltmasıyla ifade edilir ve Baş ve Boyun Skuamöz Hücreli Karsinom kanser olarak da bilinir. Bu hastalığa sahip kişilerin cinsiyet bilgileri verilmemiştir. Veri hazırlama süreci öncesi toplam hasta sayısını ifade eden *L* sayısı 279'dur. Çizelge 3.5'teki cinsiyet bilgisinin belirsiz olduğu hasta sayısı (*B*) da 279 olarak sunulmuştur. Bu tür hastalığa sahip kişiler listesinden sansürlü bilgi verisi belirsiz olan 120 kişinin çıkarılması sonucu işleme alınan toplam hasta sayısı (*T*) 159 olarak belirlenmiştir.

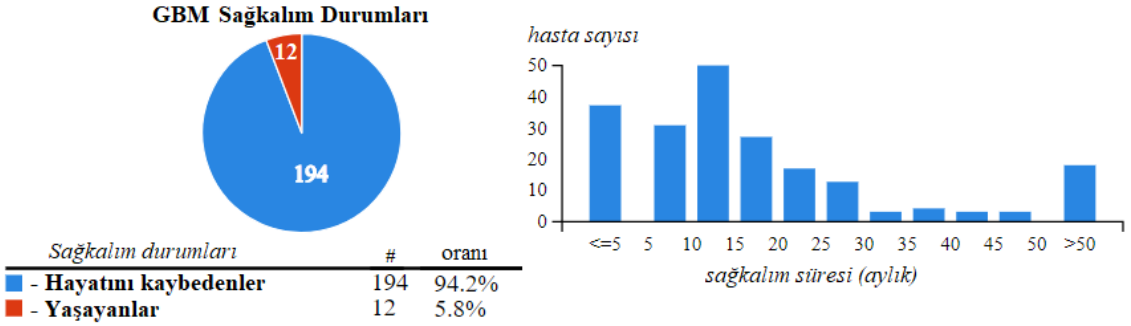
Kidney Renal Clear Cell Carcinoma (CGARN 2013, *KIRC* 2016) isimli kanser türü Böbrek-Renal Hücreli Karsinom olarak bilinir ve KIRC ile gösterilir. Veri hazırlama işlemi öncesi bu hastalık türüne sahip toplam hasta sayısı (*L*) 499'dur. Herhangi bir *CNA*'lı gen içermeyen hastalar ile sağkalım bilgileri belirsiz olan hastaların kesişimlerinin dikkate alınarak bu listeden çıkarılmasıyla (*veri hazırlama sonrası*) en son hasta sayısı (*T*) 433 olarak tanımlanmıştır. Hiçbir hastanın cinsiyet bilgisi veri setinde sunulmamıştır.

Stomach Adenocarcinoma (*STAD* 2016), Mide Adenokarsinoması olarak bilinen kanser türüdür ve STAD ile ifade edilir (*TCGA geçici-2016*). Bu kanser türüne sahip orijinal veri setinde toplam (*L*) 478 hasta bulunmaktadır. Bu hastalardan 46 tanesi ya hiç *CNA*'lı gen içermemekte ya da sağkalım bilgisi bulunmamaktadır. Bu yüzden bu 46 hastanın listeden çıkarılmasıyla elde edilen en son hasta sayısı (*T*) 432'dir. Deneysel çalışmalar bölümündeki sağkalım süreleriyle ilgili analiz testleri çizelgenin *T* sütununda belirtilen sayıdaki hastaların klinik bilgilerine göre gerçekleştirilmiştir.

Çizelge 3.5. Her bir kanser türüne göre klinik hasta bilgileri

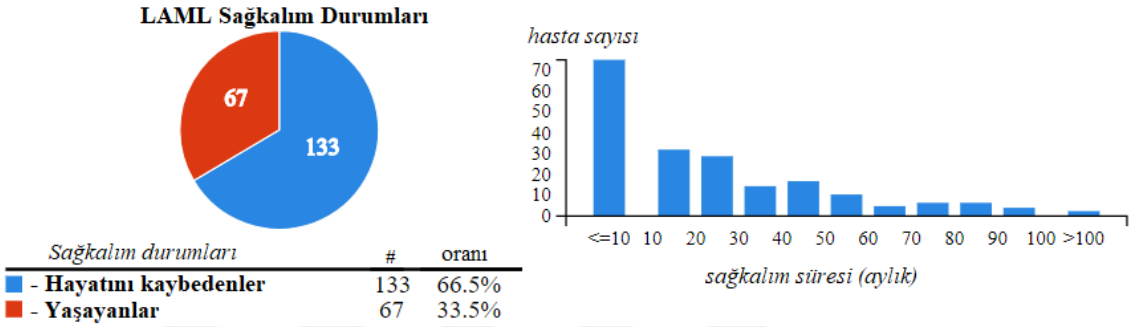
Kanser Türü	***Klinik Hasta Sayısı***					***CNA'lı Hasta Sayısı***	***Mutasyonlu Hasta Sayısı***	***Ortalama Sağkalım Süresi (gün)***	***Sansürlü Bilgi***		
	T	*L*	*E*	*K*	*B*				*S*	*Y*	*Z*
GBM	206	206	128	78	-	206	91	576	12	194	-
LAML	191	200	108	92	-	191	200	774	67	133	-
HNSC	159	279	-	-	279	279	279	729	90	69	120
KIRC	433	499	-	-	479	436	424	1125	297	148	54
STAD	432	478	284	158	36	441	395	566	267	175	36

Şekil 3.1'te GBM hastalık türü, Şekil 3.2'de LAML hastalık türü, Şekil 3.3'te HNSC hastalık türü, Şekil 3.4'de KIRC hastalık türü ve Şekil 3.5'te STAD hastalık türü ile ilgili sağkalım durumları hakkında bilgiler verilmiştir. Bu şekillerdeki diyagramlarda her bir hastalık için hayatını kaybeden, yaşayan ve sağkalım bilgisi belirsiz olan hastaların sağkalım durumları sayısal verilerle gösterilmiştir. Ayrıca, bu şekillerde sağkalım sürelerine göre hasta sayıları sunulmuştur. Bu süreler aylık olarak tutulmuştur.



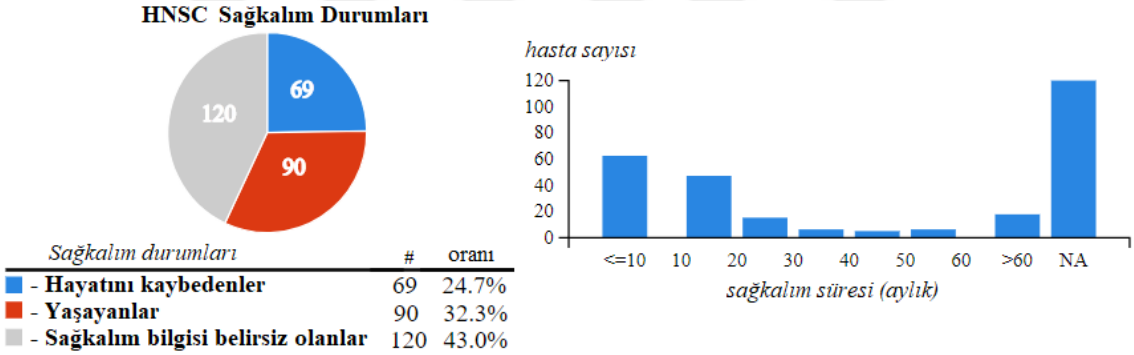
Sağkalım durumları	#	oranı
- Hayatını kaybedenler	194	94.2%
- Yaşayanlar	12	5.8%

Şekil 3.1. *GBM* için sağkalım durumları (*GBM* 2016)



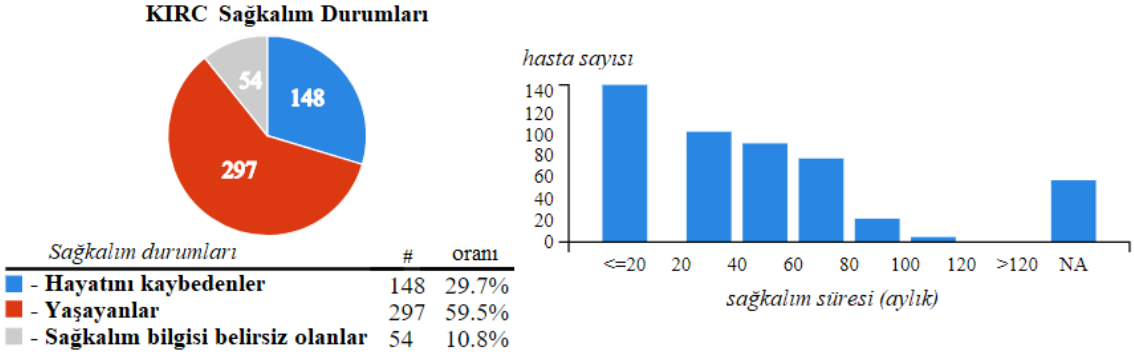
Sağkalım durumları	#	oranı
- Hayatını kaybedenler	133	66.5%
- Yaşayanlar	67	33.5%

Şekil 3.2. *LAML* için sağkalım durumları (*LAML* 2016)



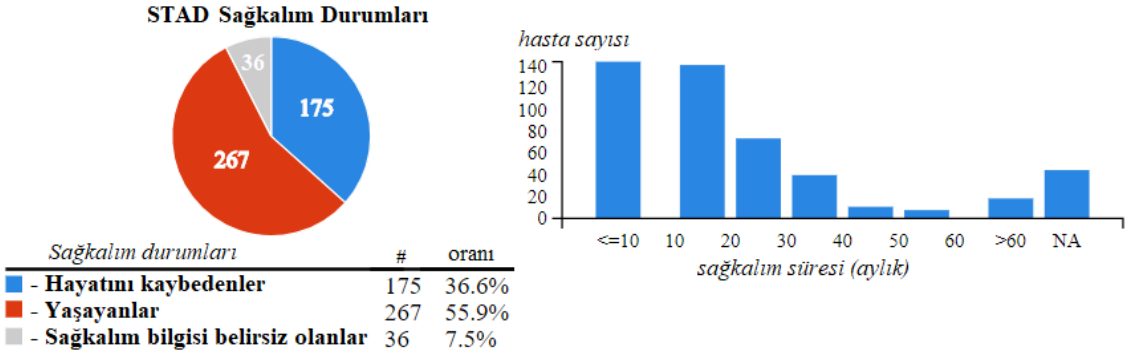
Sağkalım durumları	#	oranı
- Hayatını kaybedenler	69	24.7%
- Yaşayanlar	90	32.3%
- Sağkalım bilgisi belirsiz olanlar	120	43.0%

Şekil 3.3. *HNSC* için sağkalım durumları (*HNSC* 2016)



Sağkalım durumları	#	oranı
- Hayatını kaybedenler	148	29.7%
- Yaşayanlar	297	59.5%
- Sağkalım bilgisi belirsiz olanlar	54	10.8%

Şekil 3.4. *KIRC* için sağkalım durumları (*KIRC* 2016)



Şekil 3.5. *STAD* için sağkalım durumları (*STAD* 2016)

Çizelge 3.6'da, *Cytoscape* (Shannon ve ark 2003, Cytoscape 2017) yardımıyla *STRING*'ten (string-db 2018) elde edilen gen etkileşim ağları ve bu ağların genel karakteristik özellikleri verilmiştir. Bunlar: ağdaki düğüm ve bağlantı sayısı, ortalama derece, ağın bağlantı durumu, çapı, bağlantı (*bağ.*) yoğunluğu, kümeleme katsayısı ve ortalama (*ort.*) yol uzunluğu bilgileridir. Bu ağların tam bağlı olup olmadıkları *bağlantı durumu* ile gösterilmiştir. Bu bilgi 1 ise ağ, bir tam bağlı çizgeyi; 0 ise bağlı olmayan çizgeyi ifade eder. Bu çizelgedeki Inf ifadesi ilgili ağlarda elde edilemeyen değerleri ya da sonsuz (*infinite*) ölçümleri gösterir. "*Tam bağlı olmayan bir çizgenin sonsuz çapı vardır*" (Bollobas 1981) cümlesine göre HNSC, KIRC ve STAD veri setlerindeki ağların tam bağlı özelliklere sahip olmamalarından dolayı bunların ağ çapı ve ortalama yol uzunluğu Inf ile temsil edilmiştir.

Tez çalışmasında her bir kanser türü için etkileşim ağlarının isimleri; GBM için *GBM-net*, LAML için *LAML-net*, HNSC için *HNSC-net*, KIRC için *KIRC-net* ve STAD için *STAD-net* olarak belirlenmiştir. Bu ağlar tüm hasta popülasyonunun iki alt-popülasyona ayrılması için gerekli olan bilgileri sunmanın yanında kanserde sağkalım ile maksimum ilişkiye sahip genlerin listelerini de sunar. Bunun için komşu genlerin (çizgelerdeki düğümler) *CNA* bilgilerine göre bu ayırma işlemi gerçekleşir. Ana popülasyonun ayrılması sonucu Bölüm 2.2'de tanımlanmış P_0 ve P_1 alt-popülasyonları oluşturulur. Bu işlem seçilen k adet gen için herhangi bir kanser türüne göre Çizelge 3.6'da verilen ilgili gen etkileşim ağı yardımıyla yapılır. Ana popülasyondan seçilen genlerin hiçbirinin *CNA*'lı olmaması durumunda bu hastalar P_0'a, genlerden en az birinin *CNA*'lı olması durumunda ise bu hastaların P_1'e dahil edilmesi sağlanır. Böylece *CNA*'lı genler içermeyen hastalar P_0 alt-popülasyonunu; en az bir *CNA*'lı gen içeren hastaların ise P_1 alt-popülasyona eklenir. Sonraki işlemler bu iki ayrı popülasyondaki hastaların sağkalım bilgilerine göre hesaplanan Denklem 47'deki *log-rank istatistik*

skorunun belirlenmesi ile devam eder. Sonuç olarak, en yüksek skora ulaştıran gen listesinin elde edilmesine çalışılır. Tüm işlemler sonucu kaydedilen en son liste ise ilgili kanser türünün sağkalım parametresi ile maksimum ilişkiye sahip olduğu düşünülen ve birbiriyle etkileşimde olan k adet gene sahip alt-ağı ifade eder.

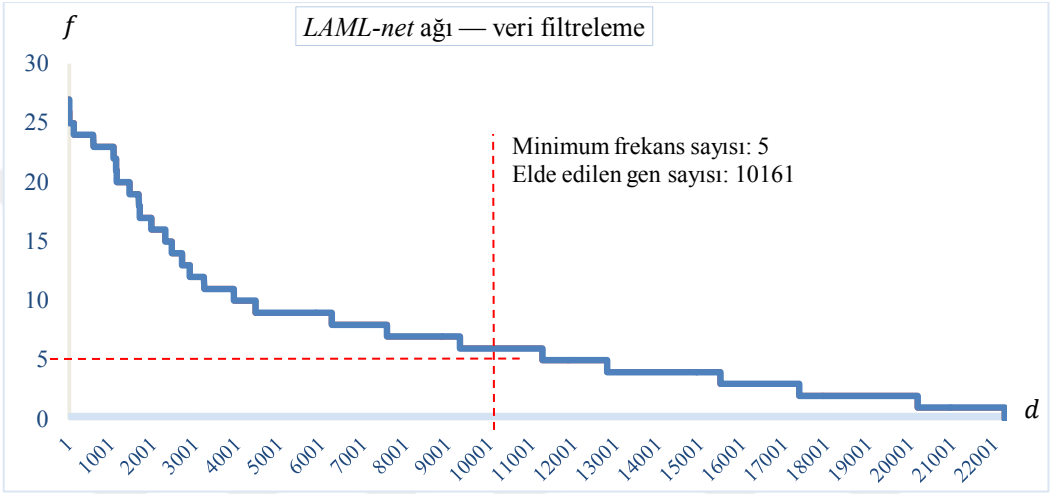
Çizelge 3.6'daki her bir ağ için verilen nihai düğüm (*gen*) ve bağlantı sayıları orijinal sayılarından genellikle farklıdır. Buradaki farklılıklar klinik hasta verilerindeki bazı genlerin *STRING* veri tabanında bulunmamasından dolayı bu genlerin mevcut listeden çıkarılmasından veya hem *"https://string-db.org"* web sayfası üzerinden hem de *Cytoscape* (Shannon ve ark 2003, Cytoscape 2017) yazılımı üzerinden büyük gen/protein etkileşim ağlarının temini sırasında karşılaşılan bazı sistem (bellek ve işlemci kısıtları gibi) kısıtlamaları sebebiyle veri filtrelemesinin yapılmasından kaynaklanmıştır. Bu filtreleme diğer genlere göre daha az sayıda hastada bulunan *CNA*'lı genlerin listeden çıkarılmasını kapsar. Bu sebeple teste tabi tutulacak ağ verileri bir filtreleme (*veri düzenleme*) sürecine alınmıştır. Böylece genlerin hastalara göre *CNA*'lara sahip olma sayıları dikkate alınarak yüksekten düşüğe doğru bir veri sıralama yapılmıştır. Burada her genin hastalarda bulunma sayıları genlerin frekanslarını oluşturur. Bu frekanslar f ile temsil edilmiştir. Daha sonra her ağa özgü belli oranlarda daha az frekansa sahip *CNA*'lı genler listeden çıkarılır.

Çizelge 3.6. Klinik kanser verileri için gen etkileşim ağları

Etkileşim ağları	Düğüm sayısı	Bağlantı sayısı	Ortalama derece	Bağlantı durumu	Ağ çapı	Ağ bağ. yoğunluğu	Kümeleme katsayısı	Ort. yol uzunluğu
GBM-net	*107*	*2103*	*39.3084*	*1*	*4*	*0.3708300*	*0.67726*	*1.6565*
LAML-net	*10161*	*229867*	*45.2450*	*1*	*9*	*0.0044532*	*0.23858*	*3.1691*
HNSC-net	*13828*	*425037*	*61.4748*	*0*	*Inf*	*0.0044460*	*0.23985*	*Inf*
KIRC-net	*14161*	*447977*	*63.2691*	*0*	*Inf*	*0.0044682*	*0.24250*	*Inf*
STAD-net	*10911*	*210245*	*38.5382*	*0*	*Inf*	*0.0035324*	*0.23859*	*Inf*

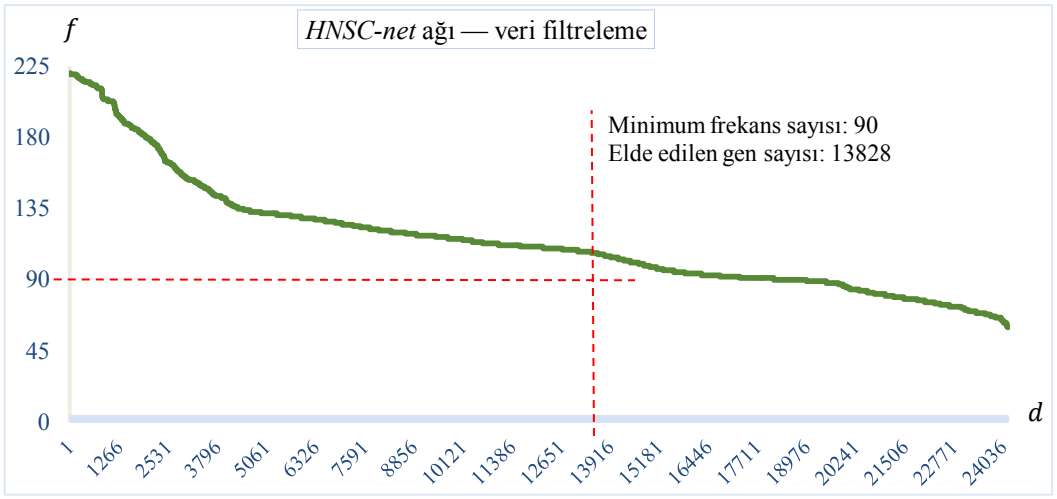
GBM-net ağının düğüm sayısının az olmasından dolayı bu ağda herhangi bir gen sayısı azaltılmasına gerek duyulmamıştır. Bu kanser türüyle ilgili gen listesi *STRING* veri tabanında sorgulanmış ve bilinen tüm genlerin oluşturduğu bir etkileşim ağı elde edilmiştir. 206 hastanın bulunduğu GBM veri seti için elde edilen gen-gen etkileşim ağında toplam 107 gen ve 2103 etkileşim bulunmaktadır. Bu kanser türü dışındaki diğer kanser türlerinin sunulan gen sayıları çok büyük olduğu için bu dört etkileşim ağı (*LAML-net*, *HNSC-net*, *KIRC-net* ve *STAD-net*) için gen sayısının azaltılması amacıyla

veri filtrelemesi işlemi yapılmıştır. Ayrıca, bu işlem yapılırken olabildiğince daha çok hastada bulunan *CNA*'lı genlerin seçilmesi amaçlanmıştır. Böylece daha az hastada bulunan genlerin ağ listesinden çıkarılması sağlanmıştır. Aşağıdaki dört ayrı filtreleme grafiğindeki "*f*" frekansları ve "*d*" gen *ID* bilgileri dikkate alınarak işlemler yapılmıştır. Buradaki *d* değişkeni 0'dan başlayarak en sonuncu gen *ID* numarasının gösterilmesinde kullanılır. Burada her bir genin hastalardaki *CNA*'lı durumlarının sayısı verilmiştir. İlk gen en yüksek *CNA* sayısına sahipken; en son sıradaki gen en düşük sayıya sahip olur.



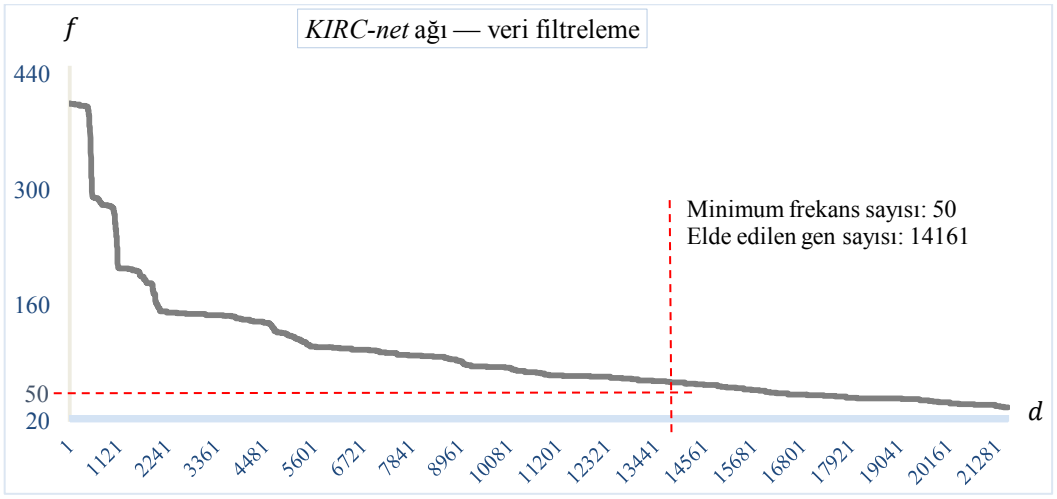
Şekil 3.6. *LAML-net* gen etkileşim ağı için veri filtreleme

Şekil 3.6'da sunulan grafikte, LAML veri setindeki tüm *CNA*'lı genlerin hastalarda bulunma sayıları (*frekansları*) gösterilmiştir. Orijinal gen listesinde toplam 22246 adet gen vardır. Bu genlerden *CNA* içermeyenler ile tekrar edenler listeden çıkarılmış ve her genin hastalarda bulunma sayıları elde edilmiştir. *STRING* veri tabanında bulunan tüm genler sorgulanmış ve bu genlerin birbirleriyle olan fonksiyonel ilişkilerini modelleyen gen ağı elde edilmiştir. *LAML-net* ağ için minimum frekans sayısı 5 olarak belirlenmiştir. Bu sayı en az 5 hastada bulunan *CNA*'lı genlerin ağı oluşturan gen listesine dahil edilmesini ifade eder. Burada *STRING* veri tabanının web sayfası ve *Cytoscape* yazılımının sistem kısıtları sebebiyle minimum 5 frekansa sahip genlerin bazıları listeden çıkarılmıştır. Böylece olabildiğince büyük bir gen etkileşim ağı elde edilmiştir. Filtreleme işleminin yapıldığı minimum frekansı göstermek için kesikli kırmızı çizgiler kullanılmıştır. İki kesikli çizginin kesişim yeri 5 frekanslı 10161 genin konumunu gösterir. Böylece 22246 gen yerine 10161 genden oluşan *LAML-net* ağı elde edilmiş olur. Bu ağda toplam 229867 adet genler arası bağlantı bulunmaktadır.



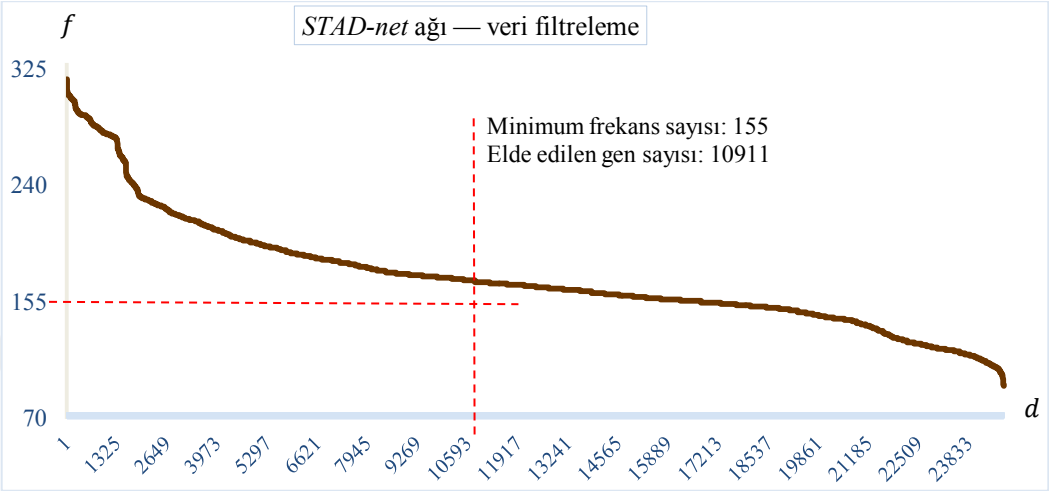
Şekil 3.7. *HNSC-net* gen etkileşim ağı için veri filtreleme

Şekil 3.7'de HNSC veri seti için gen etkileşim ağının seçilen toplam gen sayısı ve minimum frekans sayısı gösterilmiştir. Belirlenen minimum frekans sayısı 90'dır ve orijinal verideki 24174 adet genden bu şarta uyan 13828 gen etkileşim ağının üretilmesi için seçilmiştir. Bu genlerin toplam etkileşim sayısı 425037'dir. Bu ağdaki genlerin hastalarda bulunma sıklıkları *LAML-net* ağına göre daha yüksektir. Örneğin *LAML-net* ağındaki genlerden en fazla hastada bulunan *CNA*'lı gen, *CNTNAP2*'dir ve bu genin frekansı 27 iken; *HNSC-net* ağında 221 frekans değerine sahip *USP19*, *LAMB2*, *SEMA3F* ve *GNAT1* gibi genler bulunmaktadır. Şekil 3.6 ve Şekil 3.7 incelendiğinde *CNA*'lı genlerin LAML verisine kıyasla HNSC verisinde daha fazla hastada bulunduğu anlaşılmaktadır.



Şekil 3.8. *KIRC*-net gen etkileşim ağı için veri filtreleme

KIRC veri seti için filtreleme işlemi öncesi orijinal klinik verideki hastaların gen sayısı 21526'dır. Bu verinin filtrelenmesinde Şekil 3.8'deki minimum frekans değeri 50 olarak belirlenmiştir. Seçilen genlere göre *STRING* veri tabanından elde edilen *KIRC-net* gen etkileşim ağında 14161 gen ile 447977 bağlantı bulunmaktadır.

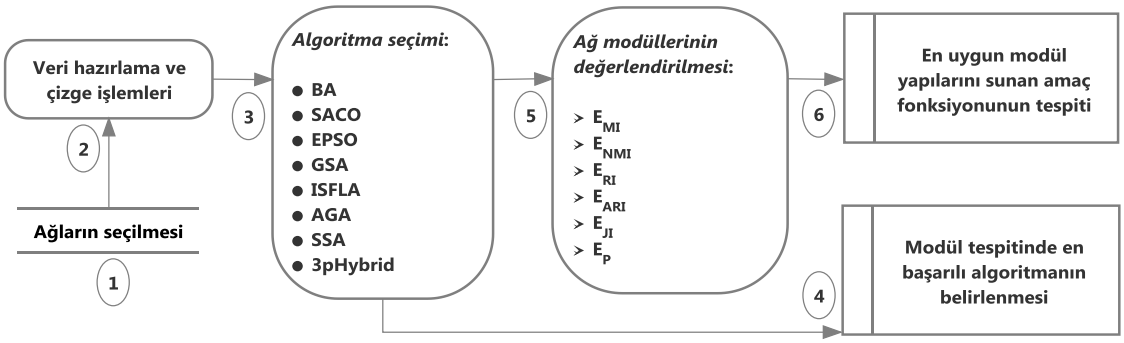


Şekil 3.9. *STAD-net* gen etkileşim ağı için veri filtreleme

Şekil 3.9'da STAD veri setinde bulunan tüm hastalara ait *CNA*'lı genlerin oluşturduğu etkileşim ağının filtreleme işlemi gösterilmiştir. Bu ağda filtreleme öncesi gen sayısı 24776 iken; filtreleme sonrası gen sayısı 10911'e indirgenmiştir. Bu veri için belirlenen minimum frekans sayısı 155'tir. 10911 ve 155 değerlerinin kesişimi kesikli çizgilerle vurgulanmıştır. *STAD-net* ağındaki en son toplam bağlantı sayısı 210245'tir.

3.4. Ağ Modüllerinin Tespitinde Kullanılan Yöntemler

Bu bölümde ağ modüllerinin tespiti amacıyla seçilen metasezgisel yöntemler hakkında bilgiler verilmiştir. Bu yöntemler, ayrık yarasa algoritması, seçime dayalı karınca kolonisi optimizasyonu, zenginleştirilmiş parçacık sürü optimizasyonu, ayrık yerçekimsel arama algoritması, geliştirilmiş kurbağa sıçrama algoritması, adaptif genetik algoritma, dağınık arama temelli algoritma ve üç fazlı hibrit yaklaşımdır. Bu bölümde, birbirlerinden farklı mekanizmalara sahip optimizasyon algoritmalarının modül/topluluk tespiti problemine uyarlanması ve gerçekleştirilen testlerin sonuçlarına göre birbirleriyle kıyaslanmaları amaçlanmıştır. Bu algoritmalarla ağ modül yapılarının ortaya çıkarılmasında takip edilen temel işlem adımları Şekil 3.10'da verilmiştir.



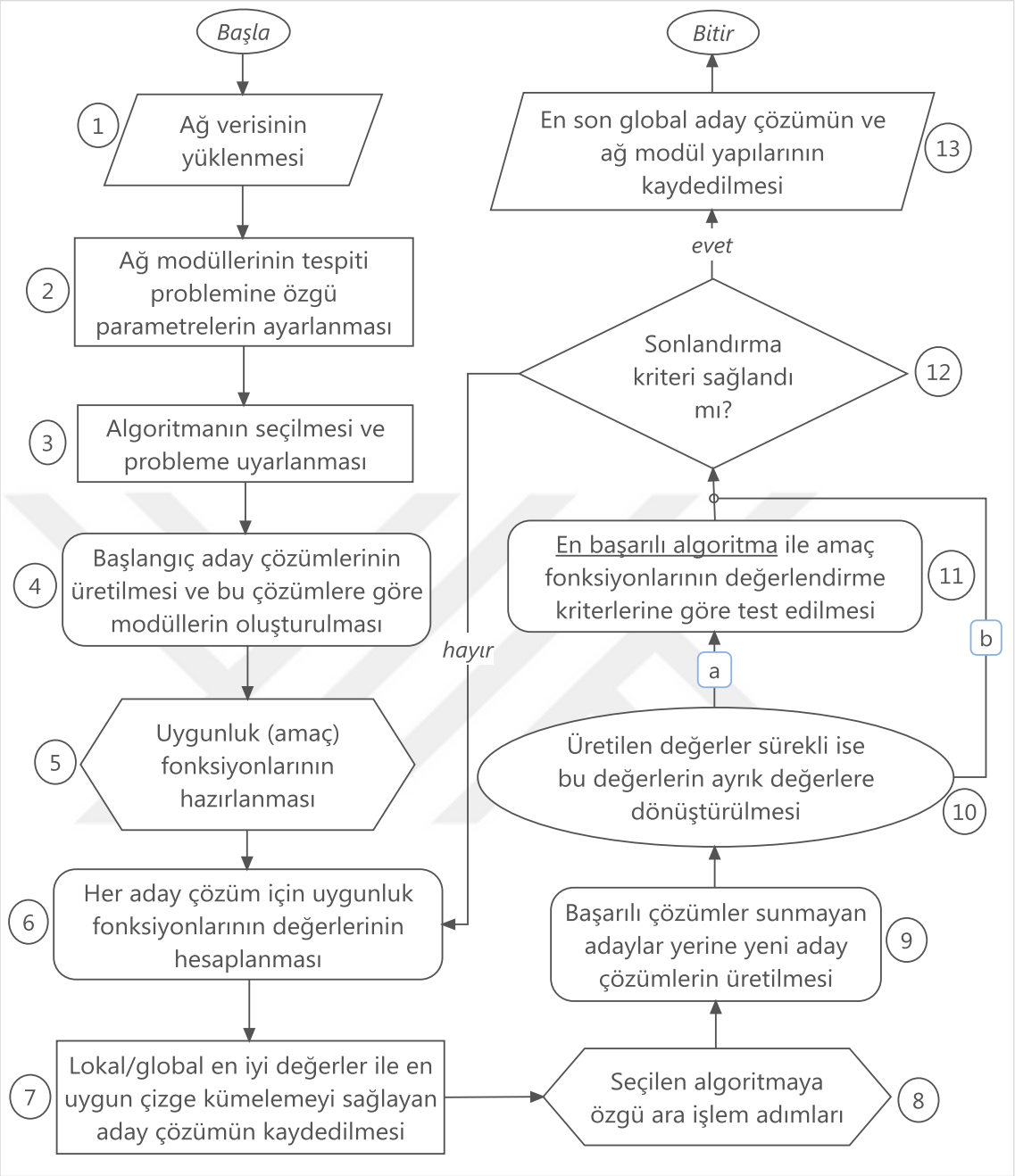
Şekil 3.10. Ağ modüllerinin tespitinde uygulanan temel işlem adımları

Şekil 3.10'da altı işlem adımı sunulmuş ve her bir adım bir sıra numarası ile temsil edilmiştir. İlk adımda Bölüm 3.1 ve Bölüm 3.2'de verilen gerçek dünya ağları ile rastsal olarak üretilen yapay test ağlarından biri seçilir ve sisteme yüklenir.

İkinci adımda seçilen ağ algoritma yapılarına ve çizge teorisi fonksiyonlarına uygun olarak yönsüz ve ağırlıksız çizge yapısına dönüştürülür ve bu yapı veri hazırlama işlemlerine tabi tutulur. Veri hazırlama sürecinde çizgedeki döngülü ve paralel düğümler listeden çıkarılır ve çizgenin topolojik özellikleri algoritmalar için giriş parametrelerini oluşturur.

Üçüncü işlem adımında, bu bölümde sunulan tüm algoritmalar yardımıyla seçilen ağın modül yapıları ortaya çıkarılır. Bu algoritmalar için uygunluk fonksiyonu olarak modülerlik (Newman ve Girvan 2004 (b)) fonksiyonu tercih edilmiştir. Bu fonksiyonunun literatürde en çok kullanılan ve en iyi bilinen kalite fonksiyonu (Fortunato 2010) olması sebebiyle tez çalışmasında da bu amaç fonksiyonu temel kriter olarak seçilmiştir. Bu işlem adımında tüm algoritmaların test sonuçları sonraki sürece aktarılır. Böylece dördüncü işlem adımında, en yüksek modülerlik değerine ulaşan algoritma belirlenir ve bu algoritmanın önerdiği ağ modülleri kaydedilir.

Beşinci adımda giriş ağının en uygun modül yapısını ortaya çıkaran amaç fonksiyonunun tespiti amacıyla Bölüm 2.1.3'de sunulan 10 adet fonksiyon bir önceki adımda seçilen algoritmanın uygunluk fonksiyonları olarak seçilir. Burada farklı amaç fonksiyonlarıyla (F_M, F_C, F_{ID}, F_{GD}, F_{CR}, F_{NC}, F_{FS}, F_{ODF}, F_S, F_{Sr}) elde edilen modüller altı farklı değerlendirme fonksiyonu (E_{MI}, E_{NMI}, E_{RI}, E_{ARI}, E_{JI}, E_P) ile test edilmektedir. Son işlem adımında ise her bir amaç fonksiyonunun farklı değerlendirme kriterlerine göre kaydedilen test sonuçları analiz edilmekte ve en başarılı modülerliği sunan amaç fonksiyonu belirlenmektedir.



Şekil 3.11. Tüm algoritmaların kullandıkları probleme özgü genel akış şeması

Ağ modüllerinin tespiti için önerilen tüm algoritmalar Şekil 3.11'de sunulan akış şemasının genel işlem adımlarını aynen kullanırlar. Bu şemadaki her sürece bir numara atanmıştır. Aşağıda, sonraki alt başlıklarda açıklanan algoritmaların çalışma adımları anlatılırken Şekil 3.11'de gösterilen akış şemasındaki tüm işlemler için atanmış adım numaraları dikkate alınmıştır.

Ağ modül tespitindeki ilk süreç ağ verisinin belleğe yüklenmesi ile başlar. Ağ verileri düğümlerin birbirleriyle bağlantılarının bulunduğu dosyalarda tutulmuştur. Bu

veri belleğe yönsüz ve ağırlıksız çizge olarak kaydedilir. Ayrıca en iyi amaç fonksiyonunun tespiti amacıyla ilgili ağa özgü daha öncesinden elde edilmiş referans ağ modüllerinin sayıları ve modüllerin bilgileri de sisteme yüklenir. 2. süreçte, tanımlanan problem için gerekli düğüm/bağlantı sayıları, komşuluk listesi ve matrisi gibi parametrelerin içerikleri belirlenir. Bu parametreler probleme özgü değişkenlerdir. 3. süreçte sonraki bölümlerde sunulan 8 adet algoritmadan biri seçilir ve bu algoritmanın parametreleri probleme göre tanımlanır. 4. işlem sürecinde algoritmaların başlangıç popülasyonları oluşturulur. Bu işlem algoritmaların tercihlerine göre iki şekilde yapılır. İlk tercih "$mekanizma-1$" diye isimlendirilmiştir. Bu mekanizmada, popülasyondaki bireyler ağdaki düğümlerin komşuluk ilişkileri dikkate alınarak seçilir ve bireyler rastgele üretilirler. Bununla ilgili ek açıklama Bölüm 3.4.1'de sunulmuştur. İkinci tercih ise "$mekanizma-2$" olarak isimlendirilmiştir. Burada rastgelelik içermeden düğümlerin komşuluk yoğunlukları ve olasılık matrisleri dikkate alınarak yeni bireyler üretilir. Bununla ilgili açıklamalar Bölüm 3.4.2'de yapılmıştır. İki farklı popülasyon üretim mekanizmasından sadece *SACO* algoritması ikincisini kullanırken, *3pHybrid* yöntemi hem birinci hem de ikinci mekanizmayı kullanır. Diğer altı algoritmanın başlangıç popülasyonlarının üretimi ise ilk mekanizmaya göre gerçekleştirilir, yani düğüm komşulukları temel alınarak rastgele bireyler üretilir. Popülasyondaki bireylerin oluşturulmasıyla ilgili temsili örnek Bölüm 2.1.1'de sunulmuştur. 5. ve 6. süreçlerde, üretilen bireylerin 10 farklı amaç fonksiyonuna göre uygunluk değerleri hesaplanır ve bu değerlere göre seçilen en iyi birey sonraki sürece aktarılarak kaydedilir. 8. süreçte seçilen algoritmanın kendi mekanizmasına göre birey geliştirme işlemleri uygulanır. Bu adım algoritmalara göre şekillenmekte ve algoritmaların ilgili çalışma mekanizmaları kendi alt bölümlerinde açıklanmaktadır. 9. süreçte ise iyi çözümler sunamayan ve geliştirilemeyen bireylerin popülasyondan silinmesi ve yerine yeni bireylerin dahil edilmesi anlatılır. 10. süreç *BA* ve *GSA* algoritmaları için üretilen kesirli sayıların doğal sayılara yuvarlanmasını; *EPSO* için ise Bölüm 3.4.3'te sunulan sekiz farklı transfer fonksiyonu ile ayrıklaştırılmasını içerir. Modül tespiti probleminde düğüm komşularının seçimi için bu üç algoritma kendi yapılarına uygun olarak kesirli değerler üretirler. Bu süreçte düğüm bilgilerinin ayrık olması sebebiyle üretilen sürekli değerler komşu düğümlerin sıra numaralarına uygun olarak bir çevirme işlemine tabi tutulurlar. Bu algoritmaların orijinal/standart versiyonları sürekli problemlere uygundur. Böylece 10. süreçteki çevirme işlemi ile üç algoritma temel olarak ayrıklaştırılmış olur. 11. süreçte on adet amaç fonksiyonu içinden en uygun modül tespiti yapan fonksiyonun

belirlenmesi için değerlendirme kriterlerine göre testler gerçekleştirilir. Buradaki kriterlerin temsil edildiği denklemler Bölüm 2.1.4'te sunulmuştur. Bu işlem adımı tüm algoritmalar için uygulanmamaktadır. Bunun yerine tüm test ağlarında modülerlik değerlerine göre en iyi sonuçlara ulaşan algoritma "*en başarılı algoritma*" olarak etiketlenmektedir ve bu süreçte sadece seçilen algoritma kullanılmaktadır (Şekil 11'deki "a" işlem adımı). Diğer algoritmalar doğrudan "b" işlem adımıyla devam etmektedirler. 12. süreçte sonlandırma şartı kontrol edilmektedir. Bu şart seçilen algoritmanın toplam iterasyon sayısı kadar çalıştırılmasını ifade eder. Son süreçte ise giriş ağının en uygun modül yapısını sunan nihai birey ve buna göre oluşturulan modül yapıları çıktı olarak sunulur.

Aşağıdaki alt başlıklarda Şekil 11'deki ortak süreçlerle birlikte kendi yapılarına özgü mekanizmaları kullanan tüm algoritmaların genel özellikleri verilmiştir. Ayrıca, her bir algoritma için testlerde kullanılan parametrelerin değerleri literatürdeki önerilere ve testlerdeki çözümlerin kalitelerine göre belirlenerek alt başlıklar altında sunulmuştur.

3.4.1. Ayrık Yarasa Algoritması

Yarasa Algoritması (*Bat Algorithm*), yarasaların ekolokasyon olarak ifade edilen sesin yankılanmasından faydalanılarak bir cismin bulunduğu yönün ve uzaklığın belirlenmesi gibi davranışlarına göre modellenen ve bu canlıların karanlık ortamlarda dahi hareket edebilmelerine imkan sağlayan temel özelliklerinden esinlenen sürü zekasına dayalı metasezgisel optimizasyon algoritmasıdır (Yang 2010). Bu algoritma *Xin-She Yang* tarafından önerilmiştir. Buradaki ekolokasyon mekanizması en basit tanımıyla ses titreşimi ile yer tespit edilmesi işlemidir. Doğal yaşamlarında yarasalar yol ve yiyecek arayışı sırasında yansıyan ses titreşimlerinin yoğunluğuna göre bir yön haritası oluşturur ve uygun yöne doğru yakınsarlar. Bahsi geçen kurala göre tasarlanan yöntemde her bir yarasa hız, konum, frekans, dalga boyu ve ses şiddeti gibi parametrelere sahiptir. *Yang*, yarasaların ekolokasyon davranışlarını örnek alan yarasa algoritmasını takip eden cümledeki üç temel kurala göre modellemiştir (Yang 2010, Koç 2016). Bu kurallar şunlardır: (1) Yarasalar mesafeyi algılamak için ekolokasyon mekanizmasını kullanırlar ve av ile arka plandaki engeller arasındaki mesafeyi yapısal özellikleriyle kavrayabilirler. (2) Avın aranması için her yarasa x pozisyonuna, v hızına, sabit f_{min} frekansına, λ değişken dalga boyuna ve A ses şiddetine sahiptir ve avın

konumuna göre yarasalar rastsal olarak uçarlar. Modellenen algoritmanın yapısına uygun olarak bu canlılar f frekansını (*dalga boyu*) ve 0 ile 1 aralığındaki r emisyon oranını hedef bölgenin yakınlığına göre otomatik olarak ayarlarlar. (3) Ses şiddeti parametresinin farklı şekillerde değişmesine rağmen önerilen algoritmada bu parametrenin değerlerinin en yüksek ses şiddeti (A_0) ile sabit en düşük ses şiddeti (A_{min}) aralığında değiştiği varsayılır.

Denklem 50'de t anındaki bir yarasanın konum değişikliği gösterilmiştir. Burada t değişkeni mevcut anda ulaşılan bilgileri gösterirken; $t-1$, bir önceki andaki konum ve hız bilgilerini tutar. Denklem 48'e ve Denklem 49'a göre sırasıyla; yarasanın frekans ve hız değerleri hesaplanır. χ değeri rastgele üretilen frekans katsayısını temsil eder. Başlangıçta, tüm yarasalar için rastsal olarak $[f_{min}, f_{max}]$ aralığında frekans değerleri üretilir. x_g, tüm yarasalar arasındaki aday çözümlerin karşılaştırılması ile belirlenen o anki en iyi global aday çözümü (*global-best-bat*) ifade eder. Bu denklemlere göre frekansların yarasaların hızlarını ve hızların da yeni konumları etkilediği anlaşılır.

$$f_i = f_{min} + (f_{max} - f_{min})\chi \tag{48}$$

$$v_i^t = v_i^{t-1} + \left(x_i^t - x_g\right)f_i \tag{49}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{50}$$

Modül tespiti problemine adapte edilen standart yarasa algoritmasının yukarıda verilen parametreleri bu probleme uygun şekilde ayarlanmıştır. Ayrıca, orijinalinde sürekli problemlerin yapısına uygun olan bu algoritma, tez çalışmasında ayrık hale dönüştürülmüş ve *BA* ile temsil edilmiştir. Bu algoritmanın probleme uyarlanmış sözde-kodu (*pseudo-code*) Çizelge 3.7'de sunulmuştur. Bu çizelgeye göre başlangıçta tüm yarasalar için ($i = 1 \ldots n$) *BA*'nın başlangıç parametreleri tanımlanır. Burada gösterilen i değişkeni yarasaların temsili sıra numaralarını; n ise popülasyondaki toplam yarasa sayısını ifade eder. Çizelge 3.7'deki ikinci işlem satırında tanımlanan d değişkeni problem boyutunu tutar ve bu boyut giriş ağındaki toplam düğüm sayısına karşılık gelir. Böylece toplam d düğümlü (boyutlu) n adet yarasa popülasyonu oluşturur. Uygunluk fonksiyonu olarak da modülerlik amaç fonksiyonu (F_M) tanımlanmıştır. Tüm bunlara ek olarak algoritmadaki parametrelerin değerleri; $f_{max} = 1$, $f_{min} = 0$, $r = 0.5$, $A = 0.5$ ve $\lambda = 0.9$ olarak belirlenmiştir.

Çizelge 3.7. Ayrık yarasa algoritmasının sözde-kodu

1. x_i ve v_i değerlerine sahip ve komşuluk listesi ile uyumlu başlangıç popülasyonunu üret
2. Modülerlik amaç fonksiyonunu tüm yarasalar için tanımla —$F(x_i)$, $x \rightarrow (x_1, \ldots, x_d)$
3. Frekansı belirle→ $f_i \in [f_{min}, f_{max}]$, artış oranı—$r_i$ ile ses şiddetini—A_i tanımla
4. Ağ yapısına ve probleme özgü diğer tanımlamaları yap ve t sayacına 1 sayısını ata
5. Döngüyü başlat $(t < maksimum\ iterasyon\ sayısı)$
6. Frekansı ayarlayarak yeni çözümler üret
7. Yarasa hızlarını ve çözümlerini/konumlarını 48., 49. ve 50. denklemlere göre güncelle
8. Şekil 3.11'deki 10. süreç adımına göre sürekli değerleri ayrıklaştır (1)
9. 'Şart-1' $\langle rastgele\ üretilen\ sayı, emisyon\ oranından\ yüksekse—rand(0,1) > r_i \rangle$
10. En iyi aday çözümler içerisinden bir aday seç ve bunun çevresinde lokal arama yap
11. 'Şart-1' sonu
12. Rastgele uçuşla yeni bir aday çözüm üret
13. Şekil 3.11'deki 10. süreç adımına göre yeni üretilen sürekli değerleri ayrıklaştır (2)
14. 'Şart-2' $\langle rand(0,1) < A_i\ ve\ f(x_i) < f(x_g) \rangle$
15. Yeni aday çözümleri kabul et (eğer daha büyük modülerlik değerine sahipse)
16. r_i değerini arttır ve A_i değerini azalt
17. 'Şart-2' sonu
18. Yarasaları modülerlik değerlerine göre sırala ve en başarılı aday çözümü bul
19. t sayacını bir arttır: $t = t + 1$
20. Döngüyü sonlandır ve en uygun sonucu veren yarasaya göre ağ modüllerini oluştur

Başlangıçta rastsal olarak üretilen her bir yarasa komşuluk listesine/matrisine uygun olarak bir çözüm dizisini temsil eder. Bu dizinin uzunluğu maksimum düğüm sayısı kadardır ve 1'den başlayarak her bir yarasa için rastgele bir şekilde uygun komşular seçilir. Bu şekilde oluşturulan aday çözümlerin Denklem 17'ye göre uygunluk değerleri (F_M) hesaplanır. Bu değerlere göre yarasaların hızları ve konumları güncellenerek en uygun modül yapılarının ortaya çıkarılmasına çalışılır. Komşuluk matrisine göre yeni yarasaların (bireylerin) üretilebilmesi için sınır kontrolü yapan ve kesirli değerleri doğal sayılara yuvarlayan ek bir fonksiyon tanımlanmıştır.

Belli kısıtlara göre rastgele birey üretim işlemi $mekanizma - 1$'i ifade eder ve Çizelge 3.7'deki 1. ve 12. adımlarda uygulanır. Bölüm 2.1.1'de verilen Şekil 2.3-a'daki 8 düğümden oluşan bir çizge için örnek bir birey üretim ve ayrıklaştırma işlemi Şekil 3.12'de gösterilmiştir.

Bireylerin düğüm sıraları →	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Üretilen ondalıklı değerler →	**2.57**	**6.01**	**1.248**	**5.9**	**0.75**	**3.84**	**1.92**	**6.435**
Seçilen komşu düğümler →	**3**	**6**	**1**	**6**	**1**	**4**	**2**	**6**

Şekil 3.12. $mekanizma - 1$'e göre rastgele birey üretimi ve ayrıklaştırma işlemi

Şekil 3.12'de üretilen birey *BA* algoritmasında aday çözümü ifade eden bir yarasayı temsil eder. Bu aday çözüm Şekil 3.4'deki sekiz düğümlü örnek çizgeye göre oluşturulmuştur. Tez çalışmasında ayrık çözümler üreten algoritmalar rastgele birey üretim işlemi sırasında bir *dönüşüm*/*ayrıklaştırma* işlemine ihtiyaç duymadan doğrudan ağdaki düğümlerin komşuluk ilişkilerine göre aday çözümler üretirler. Şekil 3.12'deki örnekte her bir birey 8 boyutludur. Düğümler kendi sıra numaralarına göre bağlı oldukları komşu düğümlerden rastsal olarak bir aday düğüm seçerler. Böylece birbirlerine bağlı düğümler topluluğu elde edilmiş olur. Daha sonra üretilen bireydeki düğümlerin oluşturdukları ağ modüllerine göre F_M fonksiyonu değeri elde edilmiş olur. Her bir modül birbirleriyle yoğun ilişkili fakat diğer modüllerdeki düğümlerle daha az ilişkide olan düğümlere sahip olur. Bu şekilde modüller arası ilişkiler minimize edilip; modüller içerisindeki etkileşimler maksimize edilmektedir. Tüm popülasyon belirtilen biçimde oluşturulduktan sonra en yüksek modülerlik skorunu sunan aday çözüm global en iyi birey olarak kaydedilir. *BA* algoritmasında ayrık yapıdaki algoritmalardan farklı olarak her yarasanın frekans ve hız değerlerine göre üretilen yeni konumlar (*dönüşüm öncesi aday çözüm*) sürekli değerlere sahip olurlar. Bu değerler ilgili düğümlerin komşuluk listelerine uygun olarak doğal sayılara (*komşu düğüm sıra numaralarına*) yuvarlanırlar. Üretilen sayının ondalık kısmı ―*0.5*― değerinden küçük ise yuvarlama işlemi bir alt doğal sayıya; değilse bir üst doğal sayıya doğru yapılır. Şekil 3.12'de örnek olarak üretilen bireyde 1. düğüm için *2.57* yeni konum değeri üretilmiş olsun. Bu değer bir üst doğal sayı olan 3 sayısına yuvarlanır. Bölüm 2.1.1'de verilen Şekil 2.3-a'daki çizge incelendiğinde, 1. düğümün komşu düğümleri 3., 4., 5. ve 6. düğümlerdir. Buna göre ilk düğümün komşu düğümü olarak 3. düğümün seçilebileceği anlaşılır. Böylece 1. düğümün 3. düğüm ile bağlantısı olduğu varsayılır ve bu düğümler aynı modüle dahil edilir. 2. düğüm için üretilen 6.01 değeri 6 sayısına yuvarlanır ve 6. düğüm komşu düğüm olarak seçilir. Aynı şekilde 3. düğüm için 1.248 değeri 1 sayısına yuvarlanır ve 1. düğüm komşu düğüm olarak seçilir. Bu şekilde tüm düğümlerin üretilen değerleri yuvarlanır ve uygun komşular atanmış olur. Anlatılan ayrıklaştırma işlemini *BA* dışında *GSA* algoritması da uygulamaktadır. *EPSO* algoritması ise ayrıklaştırma işlemi için farklı transfer fonksiyonlarını kullanır. *BA*, *GSA* ve *EPSO* haricinde kalan diğer beş algoritma ayrıklaştırma işlemine gerek duymaz. Bu algoritmalar her bir düğüm için kendi komşu düğümleri içinden doğrudan seçim yapar.

3.4.2. Seçime Dayalı Karınca Kolonisi Optimizasyonu

Karınca Kolonisi Optimizasyonu (*ACO*), çeşitli optimizasyon problemlerine yaklaşık çözümler sunabilmek amacıyla kullanılan ve karıncaların yiyecek arama davranışlarını model alan popülasyon temelli bir metasezgisel algoritmadır. Bugüne dek literatürde önerilmiş birçok karınca kolonisi optimizasyon algoritması bulunmasına rağmen en önemlileri Karınca Sistemi (Dorigo ve ark 1991, Dorigo 1992, Dorigo ve ark 1996), Karınca Kolonisi Sistemi (Dorigo ve Gambardella 1997) ve Maksimum-Minimum Karınca Sistemidir (Stutzle ve Hoos 2000).

Karınca kolonisi sisteminin geliştirilmesinde gerçek karıncaların yiyecek arama yöntemlerinden yararlanılmıştır. Karıncaların oluşturduğu kolonilerin çoğunda üyeler yiyeceklerini ararken öncelikle kendi yuvalarının çevresinde rastsal olarak keşfetme işlemine başlarlar. Karıncalar çevrede yiyecek bulduklarında bu yiyeceğin kalitesini ve miktarını değerlendir ve bu yiyeceğin bir kısmını kendi yuvalarına götürürler. Yuvaya dönüş sürecinde diğer karıncaların da aynı yiyecek kaynağına ulaşabilmeleri için kaynağının kalitesine ve miktarına göre geçtikleri rotalara kimyasal feromen (*pheromone*) denilen maddeler bırakırlar. Bırakılan bu feromenler diğer karıncalar için izlenmesi gereken yolları belirler. Böylece bırakılan feromen miktarına göre karıncaların yiyecek kaynağı ile yuva arasındaki en kısa yolu bulmaları kolaylaşır. Burada bahsedilen gerçek karınca kolonilerinin yiyecek arama yöntemi *Dorigo* tarafından önerilen karınca sisteminin geliştirilmesine ilham kaynağı olmuştur (Dorigo 1992).

Tez çalışmasında karıncaların yiyecek arama ve feromen bırakma davranışlarına göre modellenen ve modül tespiti probleminin çözümünde bazen makul çözümler sunan karınca kolonisi optimizasyonu temelli yeni bir algoritma önerilmiştir. Bu çalışmada, orantılı uygunluk seçimi olarak bilinen rulet tekerleği seçim yöntemini (Sastry ve ark 2014) kullanan algoritma, *SACO* (*Selection based Ant Colony Optimization*) olarak isimlendirilmiştir.

Önerilen algoritmanın temel çalışma mantığı *Honghao* ve diğerleri tarafından sunulan çalışmadaki (Honghao ve ark 2013) yöntemin ağ topluluk yapılarının ortaya çıkarılması ile ilgili çalışma mekanizmasını temel alır.

SACO algoritmasında popülasyondaki her karınca (*birey*) çizgedeki düğümlerin ilişkisel gösterimi için *LAR* (Park ve Song 1998) gösterimini kullanmıştır. Bu tür bir temsili birey gösterimi için Bölüm 2.1.1'deki Şekil 2.3-b'de verilen diziler incelenebilir.

Popülasyondaki karıncaların oluşturulması işlemi diğer algoritmalardan farklıdır. Bu algoritmada birey üretimi düğümlerin komşuluk listelerinin ve feromen miktarlarının temel alınmasıyla oluşturulan seçim olasılıklarına göre yapılır. Bu seçim olasılıkları optimizasyon sisteminde karıncaların tercih edebilecekleri yolların geçiş olasılıklarını (*transition probability*—$p(i,j)$) ifade eder. Çizgedeki her düğüm için karınca temelli $p(i,j)$ olasılığı Denklem 51'e göre ayrı ayrı hesaplanır. Burada p, olasılık matrisini ve v_i, i. düğümü temsil ederken; $L(i)$, v_i'nin komşularının listesini ifade eder. $p(i,j)$, i. ve j. düğümler arası bağlantının hesaplanacak seçilme olasılığını gösterir. $[\tau(i,j)]$, iki düğümün kesişimindeki feromen miktarını; $\eta(i,j)$ ise sezgisel seçimliliği yani uygunluk fonksiyonunun etki değerini temsil eder. Burada τ ve η, NxN'lik matrislerdir. N, çizgedeki/ağdaki toplam düğüm sayısını gösterir. α ve β sırasıyla; feromen miktarının ve sezgisel benzerlik hesaplama ölçütünün önem derecelerini belirtir. Bu iki parametre değerinin uygun şekilde ayarlanması gerekir. Eğer α değeri düşük oranda seçilirse p olasılık tablosu sezgisel fonksiyon değerine göre ağırlıklandırılmış olur ya da β değeri düşük oranda seçilirse feromen yoğunluğuna göre bu tablo ağırlıklandırılmış olur. Tez çalışmasında $\alpha = 1$ ve $\beta = 0.75$ olarak seçilmiştir. Seçilen değerler, gerçekleştirilen testlerde ortalama olarak en başarılı sonuçları veren parametre değerleridir.

$$p(i,j) = \begin{cases} \dfrac{[\tau(i,j)]^{\alpha} \times [\eta(i,j)]^{\beta}}{\sum_{v_u \in L(i)} [\tau(i,u)]^{\alpha} \times [\eta(i,u)]^{\beta}}, & eğer \quad v_j \in L(i) \\ 0 \qquad\qquad\qquad\qquad\quad , & eğer \quad v_j \notin L(i) \end{cases} \tag{51}$$

İlk iterasyonda her bir düğümün seçilebilecek komşu düğümleri için eşit miktarda sabit feromen miktarı tanımlanır. *SACO* algoritmasının başlangıç feromen miktarı tüm düğümler için $\tau = 10$ olarak belirlenir. Ardışık olarak iterasyonlara göre feromen miktarları güncellenir ve bu güncelleme ile değişen p seçim olasılık matrisi dikkate alınarak önceki iterasyonlardan bağımsız bir şekilde yeni karınca popülasyonu oluşturulur. Bir önceki iterasyondan sadece en uygun modül yapısını sunan karınca lokal en iyi olarak kaydedilir ve diğer karıncalar popülasyondan silinir. Her iterasyonda güncellenen p matrisine göre sırasıyla ilk düğümden son düğüme kadar gerçekleşen komşu seçimi sürecinde rulet tekerleği seçim yöntemi kullanılır. Bu işlemde komşusu belirlenecek düğüm i'nin $p(i,:)$ olasılık değerleri seçim yöntemine göre işleme alınır ve sezgisel rastsallığa göre bir komşu düğüm seçilir. Bu işlem N adet üyeli bir ağdaki tüm düğümler için sırasıyla gerçekleşir ve popülasyon sayısı kadar $1xN$ boyutlu karıncalar

üretilir. Buradaki feromen güncellemesi temelli birey üretim yöntemi tez çalışmasında $mekanizma-2$ olarak isimlendirilmiştir. Tüm işlem adımlarını gerçekleştiren *SACO* algoritmasının genel çalışma mantığını gösteren sözde-kod Çizelge 3.8'de verilmiştir.

Çizelge 3.8. *SACO* algoritmasının sözde-kodu

1. Ağ verisini sisteme yükle ve bu veriyi yönsüz/ağırlıksız çizge olarak modelle
2. Karınca sayısı—P_{max}'ı, α ve β oranlarını ve b buharlaşma oranını tanımla
3. Çizge yapısına ve probleme özgü tanımlamaları yap → $örn.$ düğüm $sayısı: N$
4. Başlangıç feromen miktarını 10 olarak belirle → $\tau = 10$
5. Modülerlik amaç fonksiyonunu tanımla → F_M
6. η sezgisel benzerlik matrisini Denklem 57'ye göre oluştur
7. $z = 1$ atamasını yap ve iterasyonu başlat $(z < maksimum\ iterasyon\ sayısı)$
8. p olasılık matrisini Denklem 51'e göre oluştur/güncelle
9. Boş bir karınca dizisi oluştur ve ara sayaç $t = 1$ tanımlamasını yap
10. Popülasyonu oluşturmak için ara döngüyü başlat $(t < P_{max})$
11. Her düğümün $p(i,:)$ olasılık vektörüne göre komşusunu rulet tekerleği ile seç
12. Seçilen düğümü karıncaya ekle ve bu işlemi karınca boyutu (N) kadar tekrarla
13. Ara sayacı bir artır: $t = t + 1$
14. Popülasyon oluşturmayı tamamla ve ara döngüyü sonlandır
15. Tüm karıncaların modülerlik skorlarını hesapla
16. Lokal ve global en iyileri sakla ve popülasyonu sıfırla
17. Feromen miktarlarını Denklem 58'e göre güncelle
18. z sayacını bir arttır: $z = z + 1$
19. İterasyonu sonlandır ve global en iyiye göre ağ modüllerini oluştur

Uygunluk fonksiyonunun birey üretim sürecine etki oranını temsil eden sezgisel bilgi (*heuristic information*—η) (Honghao ve ark 2013) parametresi tanımlansın. Buna göre tüm i ve j düğümleri için sezgisel görünürlük oranları düğümler arasındaki benzerlik ölçütüne göre hesaplanır. Benzerlik ölçütü kavramı literatürde topolojik indeks diye de ifade edilir. Bu çalışmada benzerlik ölçütü olarak, Korelasyon Katsayısı Benzerlik Ölçüsü seçilmiştir. Bu ölçü Pearson Korelasyon Katsayısı (Hall 2015) olarak da isimlendirilir ve bu çalışmada r ile temsil edilmiştir. Burada "r" tüm düğümlere göre oluşturulan benzerlik matrisini gösterir ve içeriği Denklem 52'deki formüle göre belirlenir. Bu denklemdeki $K_{i,j}$, $K_{i,i}$ ve $K_{j,j}$ değerleri Denklem 53 ile Denklem 56 arasındaki formüllere göre hesaplanır. Aşağıdaki formüllerde N adet düğüme göre $i = \{1, 2, \dots, N\}$ ile $j = \{(i+1),\ (i+2), \dots, N\}$ düğümleri arasındaki benzerlik ilişkisi oranı "$r_{i,j}$" ile temsil edilir.

$$r_{i,j} = \frac{K_{i,j}}{\sqrt{K_{i,i}K_{j,j}}} = \frac{K_{i,j}}{\sigma_i \sigma_j} \tag{52}$$

$$\bar{i} = \frac{1}{N}\sum_a i_a \qquad ve \qquad \bar{j} = \frac{1}{N}\sum_a j_a \tag{53}$$

$$K_{i,j} = \frac{1}{N-1}\sum_a (i_a - \bar{i})(j_a - \bar{j}) \tag{54}$$

$$K_{i,i} = \sigma_i^2 = \frac{1}{N-1}\sum_a (i_a - \bar{i})^2 \tag{55}$$

$$K_{j,j} = \sigma_j^2 = \frac{1}{N-1}\sum_a (j_a - \bar{j})^2 \tag{56}$$

Yukarıda verilen hesaplamalar dikkate alınarak üretilecek karıncalar için çizgedeki her bir düğümün sezgisel seçim olasılığını içeren η matrisi Denklem 57'ye göre oluşturulur. Bu denklemde i ve j bağlı düğümleri için $e^{-r(i,j)}$ ifadesindeki e sayısı *Euler sayısı* olarak bilinir ve bu denklemde yaklaşık olarak *2.71828* olarak alınmıştır. Burada birbirleriyle yoğun etkileşimli düğümlerin seçim olasılığını temsil eden sezgisel benzerlik oranının yüksek hesaplanabilmesi için korelasyon katsayısının eksi değeri $(-r(i,j))$ dikkate alınmıştır. Böylece $1/e^{-r(i,j)}$ değeri yüksek benzerliği yani yoğun ilişkiyi vurgulamaktadır. Denklem 51'deki tüm $\eta(i,j)$ değerleri η sezgisel benzerlik matrisinden temin edilir.

$$\eta(i,j) = \frac{1}{1 + e^{-r(i,j)}} \tag{57}$$

Her yeni iterasyonda düğümlere göre feromen miktarlarını temsil eden τ matrisi güncellenir. (i,j) düğümlerinin bağlantısı için güncellenen feromen miktarı hesaplaması Denklem 58'e göre yapılır. Bu hesaplama ağdaki tüm düğümler için yapılarak τ matrisinin yeni iterasyondaki değerleri belirlenmiş olur.

$$\tau(i,j) = (1-b) \times \tau(i,j) + f(i,j) \tag{58}$$

$$f(i,j) = \begin{cases} 0 & , \quad e(i,j) \notin e_k \\ (1/\sqrt{b}) + F_M^k & , \quad e(i,j) \in e_k \ \cap \ \phi(i,j) = 0 \ ise \\ (1/\sqrt{b}) + 2 \times F_M^k & , \quad e(i,j) \in e_k \ \cap \ \phi(i,j) = 1 \ ise \end{cases} \tag{59}$$

$$\phi(i,j) = \begin{cases} 1, & v(i) \text{ ve } v(j) \ aynı \ modüle \ aitse \\ 0, & v(i) \text{ ve } v(j) \ farklı \ modüllerdeyse \end{cases} \tag{60}$$

Denklem 58'deki $\tau(i,j)$ güncellenen feromen miktarını gösterir. b, buharlaşma oranını ifade eder ve testlerde 0.25 olarak alınmıştır. $f(i,j)$ ise i ve j düğümlerinin bağlantısı için güncellenen feromen farkını gösterir.

Güncellenen feromen miktarı Denklem 59'a göre hesaplanır. Bu denklemde k, işlem yapılan karıncayı temsil eder. $e(i,j)$, i. ve j. düğümler için temsili bağlantıyı; e_k ise bu karıncanın sahip olduğu düğümlerin birbirleriyle olan tüm bağlantılarını ifade eder. F_M^k ise ilgili karıncanın elde edilen modülerlik değerini gösterir. Denklem 59'daki $\phi(i,j)$ parametresi iki düğümün aynı modülde olup olmadığını belirleyen fonksiyonu ifade eder ve Denklem 60'a göre belirlenir. Bu denklemdeki $v(i)$ ve $v(j)$, i ve j düğümlerini temsil ederler. Eğer bu iki düğüm aynı modülün üyeleri ise $\phi(i,j)$ değeri 1; aksi durumda 0 olur.

Feromen güncellemesi sırasında, ilk önce tüm bağlantılar üzerindeki feromen miktarları dörtte bir oranında azaltılır. Daha sonra k karıncası için Denklem 59'daki üç duruma göre kontrol yapılır. İlk durumda eğer i ve j düğümlerinin bağlantısı seçilen karıncadaki tüm bağlantılar içinde mevcut değilse bu durumda feromen miktarı farkı hesaplanmaz. İkinci durumda ilgili bağlantı karıncadaki bağlantılar içindeyse ve bu iki düğüm farklı modüllerin üyeleri ise bu durumda fark $[(1/\sqrt{b}) + F_M^k]$ oranında değişir. Üçüncü durumda ise hem ilgili bağlantı mevcut ise hem de iki düğüm aynı modülün üyeleri ise bu durumda feromen farkı $[(1/\sqrt{b}) + 2 \times F_M^k]$ değerinde olur. Seçilen karıncanın modülerlik skoru yüksek ise bu karıncadaki düğümler ile bunların komşu düğümlerinin feromen miktarları pozitif yönde etkilenirken; düşük modülerlik skorunda bu işlemin tam tersi gerçekleşir. Böylece birbirleriyle etkileşimde olmaları uygunluk fonksiyonunun değerini arttıran düğümlerin feromen miktarlarında artış gerçekleşir. Bu da ilgili düğümlerin sonraki iterasyonlarda seçilme olasılığını arttırır.

3.4.3. Zenginleştirilmiş Parçacık Sürü Optimizasyonu

Ağ modüllerinin tespiti amacıyla tez çalışmasında önerilen diğer bir metasezgisel yöntem *PSO* temelli hesaplama yaklaşımıdır. Bu optimizasyon yöntemi ilk kez 1995 yılında *Kennedy* ve *Eberhart* tarafından tanımlanmıştır (Kennedy ve Eberhart 1995). Sürüler halinde hareket eden kuş ve balık gibi canlıların sosyal etkileşimlere sahip oldukları ve yiyecek arama gibi önemli etkinliklerin bu sürülerde sezgisel veya rastgele gerçekleştiği görülmüştür. Ayrıca, sürülerdeki bireylerin birbirlerini etkiledikleri ve böylece amaçlarına daha kolay ulaşabildikleri de gözlemlenmiştir. *Kennedy* ve *Eberhart* sürülerin davranışlarını model alan ve *PSO* ile ifade edilen popülasyon temelli metasezgisel optimizasyon algoritması geliştirmişlerdir. Sürülerdeki canlıların sergiledikleri davranışların temsil edilebilmesi karmaşık görünse de geliştirilen sürü optimizasyonu yöntemi basit fakat etkili bir çalışma mantığına sahiptir. Bu yöntemde, bireyler arasındaki bilginin paylaşımı esas alınır. Her bir birey parçacık diye isimlendirilir ve bu parçacıkların oluşturduğu popülasyon da sürüyü temsil eder. Bu yöntemde temel amaç sürüdeki en iyi konuma sahip parçacığın yerinin (*aday çözüm*) tespit edilip diğer parçacıkların da bu yöne hareketlerinin sağlanmasıdır (Mirjalili ve Lewis 2013). Parçacıklar kendi pozisyonlarını sürüdeki en iyi pozisyona doğru ayarlarken bir önceki tecrübelerden de yararlanırlar. Bu tecrübeler algoritmadaki önceki iterasyonlara karşılık gelir. Bu optimizasyon algoritması Denklem 61'deki hız ve Denklem 62'deki konum formüllerine göre modellenir.

$$v_i^{t+1} = w_i \times v_i^t + c_1 \times r_1^t \times (p_l - x_i^t) + c_2 \times r_2^t \times \left(p_g - x_i^t\right) \tag{61}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{62}$$

$$w_i = w_{max} - iter \times ((w_{max} - w_{min})/iter_{max}) \tag{63}$$

Denklem 61'deki v_i^{t+1}, son hız değerini; w, önceki hızın yeni hız üzerindeki etkisini kontrol eden ağırlık değerini; v_i^t, önceki hız değerini; c_1 ve c_2, ölçeklendirme faktörlerini; r_1^t ve r_2^t, t anında üretilen ve 0 ile 1 arasında olan rastsal değerleri; p_l, mevcut iterasyondaki parçacıklar için en iyi konumu (*lokal en iyi*) ve p_g, tüm iterasyonlar için ulaşılmış en iyi konumu (*global en iyi*) temsil eder. Denklem 62'deki x_i^t, önceki konumu ifade ederken; x_i^{t+1}, elde edilen son konumu gösterir. Ayrıca her bir

iterasyon sayısına göre değişen ağırlıklandırma değeri w_i, Denklem 63'e göre hesaplanır. Bu denklemde $iter$, mevcut iterasyon sırasını; $iter_{max}$, maksimum iterasyon sayısını ve w_{max} ile w_{min} ise alt ve üst sınır ağırlıklandırma değerlerini tutarlar.

Tez çalışmasında ayrık yapıdaki probleme uyarlanan *PSO* temelli yaklaşım, Zenginleştirilmiş Parçacık Sürü Optimizasyonu (*Enhanced Particle Swarm Optimization*) olarak isimlendirilmiş ve *EPSO* ile temsil edilmiştir. Bu algoritmanın seçilen parametre değerleri (Shi ve Eberhart 1999) referanslı çalışmada önerilen uygun değerlerdir. Böylece c_1 ve c_2 için 2; w_{min} için 0.4 ve w_{max} için 0.9 değerleri seçilmiştir. *EPSO*'ya dahil edilen transfer fonksiyonları bu algoritmanın arama kapasitesini zenginleştirerek sürekli verilerin ayrıklaştırılmasını da sağlamıştır. Her bir parçacığın ayrıklaştırılmasının aynı transfer fonksiyonuna göre yapılması veya güncellenen konumun bir komşu düğümün indis değerine yuvarlama işlemleri popülasyondaki birey çeşitliliğini sağlayamamıştır. Bunun yerine transfer fonksiyonlarının rastsal olarak seçilmesi ve bunların ürettiği sonuçlara göre komşu düğümün seçilmesi popülasyona daha fazla çeşitlilikte yeni parçacıkların dahil edilmesini sağlamıştır. Bu fonksiyonlar (Mirjalili ve Lewis 2013) referanslı çalışmadan alınmıştır.

EPSO algoritmasında popülasyonu oluşturan parçacıklar, aday çözümleri temsil ederler. Bir parçacığın boyutu ağdaki toplam düğüm sayısı—N kadardır. Parçacıktaki indisler 1'den başlayarak N'e kadar giden düğüm sıra numaralarını; parçacıkların içerikleri ise her düğüm için komşuluk listesinden rastsal olarak seçilen komşu düğümleri içerir. Bu algoritmanın başlangıç popülasyonu *BA* algoritmasında olduğu gibi $mekanizma - 1$'e göre oluşturulur. Bölüm 2.1.1'de gösterilen Şekil 2.3-a'daki 8 düğümlü bir çizgeden elde edilen örnek bir parçacığın temsili sunumu Şekil 3.13'te verilmiştir. Parçacık sıra numaralarından ilki 1'dir. Bu sayı ağdaki ilk seçilen düğümü ifade eder. Şekil 2.3-a'daki çizgede ilk düğümün komşuluk listesi 3., 4., 5. ve 6. düğümleri içerir. Bunlardan rastgele 5. düğüm seçilerek komşu düğümler listesinin ilk konumuna eklenmiştir. Bu işlem son düğüme kadar seçilen düğümün komşuluk listesine uygun olarak tekrarlanır. İlk parçacığın içeriği Şekil 3.13'teki ikinci diziyi temsil eder. Buradaki işlem toplam popülasyon sayısı kadar tekrarlanır ve popülasyon ilk parçacığın üretilmesine benzer şekilde oluşturulur. Daha sonra üretilen parçacıkların ağ modül yapıları ortaya çıkarılır. Bu modüllere ve sahip oldukları üyelere göre her bir parçacığın modülerlik skoru hesaplanır. Örnek olarak oluşturulan temsili modüller için Bölüm 2.1.1'de verilen Şekil 2.3 ve ilgili açıklamalar incelenebilir.

Parçacıktaki sıra numaraları :	1	2	3	4	5	6	7	8
Parçacıktaki komşu düğümler:	5	7	8	1	1	8	2	3

Şekil 3.13. *EPSO* algoritmasına göre başlangıç popülasyonu için örnek bir parçacık üretimi

Bu algoritmada başlangıç popülasyonu Şekil 3.13'teki parçacıkların üretimine benzer şekilde yapılırken; sonraki iterasyonlarda yeni parçacıkların oluşturulması işlemi için Denklem 61, Denklem 62 ve Denklem 63'teki formüller kullanılır. Burada eski parçacıkların güncellenmesi ve yeni parçacıkların üretilmesi sırasında uygulanan ayrıklaştırma işlemi *BA*'da uygulanan işlemden farklıdır. Bu ayrıklaştırma işlemi için S-şekilli (*S-shaped*) ve V-şekilli (*V-shaped*) olmak üzere 2 farklı kategoride toplam sekiz adet transfer fonksiyonu seçilmiştir (Mirjalili ve Lewis 2013). Bu transfer fonksiyonları Denklem 64 ile Denklem 71 arasındaki formüllerle ifade edilirler. Bunlardan ilk dördü S-şekilli (f_{S1}, f_{S2}, f_{S3} ve f_{S4}); ikinci dördü V-şekilli (f_{V1}, f_{V2}, f_{V3} ve f_{V4}) fonksiyonlardır. Bu fonksiyonlarla ilgili bilgilere (Mirjalili ve Lewis 2013) referanslı makaleden ulaşılabilir. Denklemlerdeki x, işlem yapılan parçacığın Denklem 62'de güncellenen son konum değerini gösterirken; Denklem 68'deki erf ise Gauss hata fonksiyonunu ifade eder.

$$f_{S1} = \frac{1}{1 + e^{-2x}} \tag{64}$$

$$f_{S2} = \frac{1}{1 + e^{-x}} \tag{65}$$

$$f_{S3} = \frac{1}{1 + e^{\left(\frac{-x}{2}\right)}} \tag{66}$$

$$f_{S4} = \frac{1}{1 + e^{\left(\frac{-x}{3}\right)}} \tag{67}$$

$$f_{V1} = \left|\operatorname{erf}\left(\frac{\sqrt{\pi}}{2}x\right)\right| = \left|\frac{\sqrt{2}}{\pi}\int_0^{\frac{\sqrt{\pi}}{2}x} e^{-t^2}\,dt\right| \tag{68}$$

$$f_{V2} = |\tanh(x)| \tag{69}$$

$$f_{V3} = \left|\frac{x}{\sqrt{1+x^2}}\right| \tag{70}$$

$$f_{V4} = \left|\frac{2}{\pi}\arctan\left(\frac{\pi}{2}x\right)\right| = \left|\frac{2}{\pi}\tan^{-1}\left(\frac{\pi}{2}x\right)\right| \tag{71}$$

EPSO algoritmasında bir "i" parçacığı güncellenirken ilk olarak w_i ağırlık değeri Denklem 63'e göre hesaplanır. Bu parametrenin içeriğine göre Denklem 61'deki v_i hız değeri güncellenir. Daha sonra ilgili düğüm için seçilecek komşu düğüm değerini temsil eden x_i'nin içeriği Denklem 62'ye göre sürekli (*kesirli*) bir değer olarak hesaplanır. Bu hesaplamadan sonra ayrıklaştırma işlemine geçilir.

Ayrıklaştırma işlemi her düğümün komşusu için üretilen ondalık değerlerin Denklem 64 ile Denklem 71 arasında verilen transfer fonksiyonlarına ve düğümlerin komşuluk listelerine göre doğal sayılara dönüştürülmesini kapsar. Bu sayılar her bir düğüm için kendi komşuları içinden seçilen komşu düğüm indisini ifade eder. Buradaki işlem için ilk olarak 1 ile 8 arasında rastgele bir sayı üretilir. 1'den 8'e kadarki sayılar sırasıyla; f_{S1}, f_{S2}, f_{S3}, f_{S4}, f_{V1}, f_{V2}, f_{V3} ve f_{V4} fonksiyonlarını temsil eder. Rastgele üretilen sayının temsil ettiği transfer fonksiyonu seçilir ve bu fonksiyona göre giriş parametresini ifade eden "x" son konum değeri için 0 ile 1 arasında bir değer hesaplanır. Seçilen transfer fonksiyonuna göre hesaplanan değer f_s parametresinde saklanır. Bu parametre değeri temel alınarak Denklem 72'deki dört farklı duruma göre uygun işlem yapılır. Transfer fonksiyonları ile hesaplanan değere göre Denklem 72 kullanılarak yeni bir komşu düğüm seçilir. Bu denklemdeki $s^{i,u}$, $i.$ parçacığın $u.$ konumu temsil eden düğüm için seçilen yeni komşu düğümün indis numarasını ifade eder. r ise 0 ve 1 arasında rastgele üretilen sayıdır.

$$s^{i,u} = \begin{cases} minimum\ bağlantılı\ komşu\ düğüm, & r \le \left(\frac{f_s}{4}\right) \\ orta\ sayıda\ bağlantıya\ sahip\ komşu\ düğüm, & \left(\frac{f_s}{4}\right) < r \le \left(\frac{f_s}{2}\right) \\ maksimum\ bağlantılı\ komşu\ düğüm, & \left(\frac{f_s}{2}\right) < r \le 3 \times \left(\frac{f_s}{4}\right) \\ rastgele\ seçilmiş\ komşu\ düğüm, & 3 \times \left(\frac{f_s}{4}\right) < r \le 1 \end{cases} \tag{72}$$

Denklem 72'ye göre rastsal olarak üretilen r sayısı hesaplanan f_s değerinin dörtte birinden küçük veya eşitse u düğümünün komşuları içinden en az bağlantıya sahip düğüm, komşu düğüm olarak atanır. Eğer r'in değeri f_s'nin değerinin dörtte

birinden büyük ve yarısından küçük veya eşitse komşuluk listesindeki düğümler bağlantı yoğunluklarına göre sıralanırlar ve ortadaki düğüm komşu düğüm olarak seçilir. Üçüncü durumda, r değeri f_s'nin değerinin yarısından büyük ve $3 \times (f_s/4)$ sayısından küçük veya eşitse en fazla bağlantıya sahip komşu düğüm seçilir. Son olarak, r değeri $3 \times (f_s/4)$ sayısından büyükse "u" düğümünün komşuluk listesine uygun bir şekilde rastsal olarak yeni bir komşu düğüm seçilir. Bu işlem, parçacığın $N.$ düğümüne kadar devam eder ve birey güncelleme ile yeni birey üretim işlemi popülasyondaki tüm parçacıklar için tekrarlanır. Böylece her bir parçacığa göre rastsal olarak seçilen farklı transfer fonksiyonları ile çözüm uzayı zenginleştirilmiş olur. Tüm iterasyonlar tamamlanana kadar parçacıkların modülerlik kaliteleri geliştirilir ve en son döngüde *EPSO* algoritmasına göre giriş ağının en uygun modül yapısı ortaya çıkarılır. Bu algoritmanın tüm işlem adımlarıyla ile ilgili genel çalışma mantığı Çizelge 3.9'da sunulmuştur. Bu çizelgedeki P, popülasyondaki toplam parçacık sayısını; i, parçacığın indisini; j, parçacıktaki sıralı düğümlerin indisini; N, ağdaki toplam düğüm sayısını; F_l, lokal en iyi parçacığın modülerlik skorunu; F_g, global en iyi parçacığın modülerlik skorunu gösterir.

Çizelge 3.9. *EPSO* algoritmasının çalışma mantığına uygun sözde-kod

1. EPSO algoritmasının parametrelerini ve giriş ağının bilgilerini tanımla
2. Düğümlerin komşuluk listesine göre başlangıç parçacıklarını oluştur
3. Probleme özgü tanımlamaları yap ve h sayacına 1 ata
4. Lokal ve global en iyi parçacıklar için parametreleri (p_l ve p_g) tanımla
5. İterasyonu başlat ($h < maksimum\ iterasyon\ sayısı$)
6. Her bir iterasyonda Denklem 63'e göre w ağırlık değerini hesapla
7. Popülasyondaki parçacıklar için $\langle$döngü$-a\rangle$*'yı başlat* ($i \rightarrow 1'den\ P'e\ kadar$)
8. Parçacıklardaki düğümler için $\langle$döngü$-b\rangle$*'yi başlat* ($j \rightarrow 1'den\ N'e\ kadar$)
9. Parçacıklardaki düğümlere göre hız—v ve konum—x değerlerini güncelle
10. Rastsal olarak 0 ile 1 arasında r sayısını üret
11. r ve f_s değerlerini temel alarak Denklem 72'ye göre yeni komşu düğümleri seç
12. $\langle$döngü$-b\rangle$*'yi sonlandır*
13. Tüm parçacıkları güncellemeyi tamamla ve modülerlik skorlarını hesapla
14. $\langle$döngü$-a\rangle$*'yı sonlandır*
15. Lokal en iyi parçacığı belirle
16. 'Şart' $\langle F_l > F_g\ ise\rangle$
17. En iyi aday çözümü güncelle
18. 'Şart' sonu
19. $h = h + 1$
20. İterasyonu tamamla ve global en iyi parçacığa göre ağ modüllerini kaydet

3.4.4. Ayrık Yerçekimsel Arama Algoritması

Ağ modüllerinin tespiti için önerilen diğer bir yöntem Ayrık Yerçekimsel Arama Algoritmasıdır. Bu algoritma tez çalışmasında, *GSA* (*Gravitational Search Algorithm*) ile temsil edilmiştir. Yerçekimsel Arama Algoritması, *Rashedi* ve arkadaşları tarafından *Newton*'un hareket ve yerçekimi kanunlarından esinlenerek geliştirilmiş evrimsel bir optimizasyon algoritmasıdır (Rashedi ve ark 2009).

Yerçekimi kanununa göre kütleye sahip her bir nesne birbirine doğru hızlanır. Buna ek olarak, bu kanuna göre kütleye sahip nesneler diğer nesneleri belli bir güçle çeker. Buradaki kural yerçekimsel güç olarak isimlendirilir. Yerçekimi ve hareket kanunlarından esinlenerek geliştirilen *GSA* algoritmasındaki ajanlar kütleleri temsil eder ve popülasyonu oluştururlar. Bu algoritmanın başlangıç popülasyonu $mekanizma-1$'e göre oluşturulur. Bununla ilgili işlemler Bölüm 3.4.1'de açıklanmıştır. Bireylerin güncelleştirilmesi süreci ise *GSA*'ya özgü gerçekleşir ve sonraki ayrıklaştırma işlemi için *BA*'daki yöntem aynen kullanılır. Temsili bir ajanın gösterimi için Şekil 3.12'deki temsili birey incelenebilir. Aşağıda verilen denklemler için N, toplam düğüm sayısını gösterirken; popülasyonda bulunan P sayıdaki ajan N boyutlu dizilerle temsil edilir. Dizilerin indis değerleri düğümlerin sıralarını gösterirken; içerikleri seçilecek komşu düğümleri tutar. *GSA* algoritmasında bir bireyi ifade eden "i" ajanın gösterimi x_i olsun ve bu ajan Denklem 73'e göre oluşturulsun. Buradaki formüller ve tanımlamaların sunumunda (Koç 2016) referanslı çalışmadan yararlanılmıştır.

$$x_i = \left(x_i^1,\ x_i^2, \dots, x_i^d, \dots, x_i^N\right),\ i = 1, 2, 3, \dots, P \tag{73}$$

$$q_i^t = \frac{f_i^t - f_w^t}{f_b^t - f_w^t} \tag{74}$$

$$M_i^t = \frac{q_i^t}{\sum_{j=1} q_j^t} \tag{75}$$

$$f_b^t = maksimum\left(f_j^t\right)\ ve\ j \in 1,2,3, \dots, P \tag{76}$$

$$f_w^t = minimum\left(f_j^t\right)\ ve\ j \in 1,2,3, \dots, P \tag{77}$$

Denklem 73'teki x_i^d, $d.$ boyuttaki $i.$ kütlenin konumudur. Her bir ajanın kütlesi, bir uygunluk değerine sahiptir. Ajana ait mevcut kütle dizisindeki diğer ajanların uygunluğu Denklem 74'e göre hesaplanır. Denklem 75'teki M_i, mevcut kütlenin toplam kütleye oranını verir. Bu denklemlerde M_i^t ve f_j^t sırasıyla; t anındaki $i.$ ajanın kütlesini ve modülerlik değerini ifade eder. Burada uygunluk fonksiyonunun maksimize edilmesi amaçlanır. Denklem 76'daki f_b^t ve Denklem 77'deki f_w^t parametreleri t anındaki en yüksek ve en düşük uygunluk fonksiyonlarının değerlerini gösterir. Denklem 78 ile Denklem 81 arasındaki formüllerle $t.$ iterasyondan sonraki iterasyon için $i.$ bireyin $d.$ konumu güncellenir (Rashedi ve ark 2009, 2010).

$$F_i^d(t) = \sum_{j \in L \,\&\, j \neq 1} r_j G^t \frac{M_i^t M_j^t}{R_{ij}^t + \varepsilon} \left(x_j^d(t) - x_i^d(t) \right) \tag{78}$$

$$a_i^d(t) = \frac{Fi^d(t)}{M_i^t} \tag{79}$$

$$\mathrm{v_i^d(t+1)} = \mathrm{r_i} \times \mathrm{v_i^d(t)} + \mathrm{a_i^d(t)} \tag{80}$$

$$\mathrm{x_i^d(t+1)} = \mathrm{x_i^d(t)} + \mathrm{v_i^d(t+1)} \tag{81}$$

Yukarıdaki denklemlerde kullanılan r_i ve r_j değerleri 0 ile 1 aralığında üretilen rastsal sayılardır. Denklem 78'deki ε küçük bir değerdir ve bu değerle 0'a bölme probleminin önüne geçilir. Denklem 82'deki R_{ij}^t, $i.$ ve $j.$ ajanlar arasındaki Öklid uzaklığını temsil eder ve Denklem 78'de kullanılır. $F_i^d(t)$'nin değeri hesaplanırken yüksek uygunluk değerlerine sahip parçacıkların kümesini oluşturan L listesinin birey sayısını ifade eden L_d kadar işlem yapılır. Denklem 83'teki G^t, elde edilecek yerçekimsel sabiti; G^0, başlangıç değerini; β, sabit katsayıyı; t, mevcut iterasyon sayısını ve t_{son}, son iterasyon sayısını (*bitiş iterasyon sayısını*) ifade eder. Bu çalışmada $G^0 = 10$, ajan sayısı 30 ve maksimum iterasyon sayısı 1000 olarak alınmıştır.

$$R_{ij}^t = \| x_i(t), x_j(t) \|_2 \tag{82}$$

$$G^t = G^0 e^{-\beta \left(\frac{t}{t_{son}} \right)} \tag{83}$$

Bir ajanın ivme hesabı yapılırken diğer ajanlar tarafından kuvvet kanununa göre bahsi geçen ajanın üzerine etki eden toplam güç Denklem 78'deki formüle göre hesaplanır. Bu hesaplamadan sonra hareket kanununa göre ajanın ivmesi hesaplanır ve bu hesaplama Denklem 79'a göre yapılır. Denklem 80'e göre bu ajanın ivme değeri mevcut hızıyla toplanır ve yeni hız vektörü hesaplanır. Daha sonra Denklem 81'e göre yeni hız değeri ile önceki konum toplanarak yeni konum hesaplanmış olur. Tüm bu işlemler popülasyondaki ajanların tamamının güncelleştirilmesine ve ayrıklaştırılmasına kadar devam eder. Maksimum iterasyon sayısına ulaşılınca algoritma sonlandırılır. *GSA* algoritmasının Şekil 3.11'deki genel akış şemasına uygun olarak hazırlanan *sözde-kodu* Çizelge 3.10'da verilmiştir. Ayrıca, bu algoritmayla ilgili parametreler ve formüller hakkında ek açıklamalar için (Rashedi ve ark 2009) ve (Rashedi ve ark 2010) referanslı çalışmalar incelenebilir.

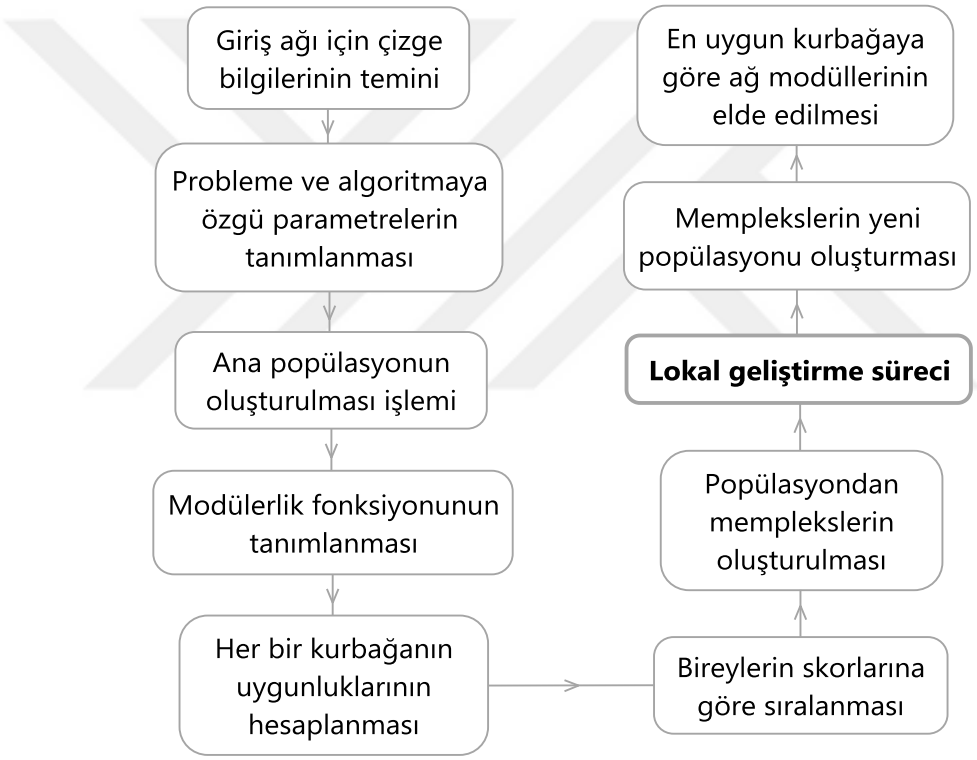
Çizelge 3.10. *GSA*'nın probleme özgü sözde-kodu

1. Denklem 73 ile Denklem 83 arasındaki GSA'ya özgü parametreleri tanımla
2. Başlangıç ajanlarını $mekanizma - 1$ *göre oluştur*
3. Ağ topluluk tespiti için ağ bilgilerini al
4. Algoritmayı $t = 1$ *ataması ile başlat* $(t < maksimum\ iterasyon\ sayısı)$
5. Her bir ajan için uygunluk skorunu belirle
6. Denklem 81'deki konum güncelleştirilmesini tüm bireyler için tamamla
7. Sürekli konum değerlerini Şekil 3.11'deki 10. süreç adımına göre ayrıklaştır
8. En iyi lokal ve global ajanları modülerlik değerlerine göre belirle
9. $t = t + 1$
10. Eğer $(t == maksimum\ iterasyon\ sayısı)$ *ise algoritmayı sonlandır*
11. Global en iyi ajana göre ağ modüllerini ve toplam modül sayısını belirle

3.4.5. Geliştirilmiş Kurbağa Sıçrama Algoritması

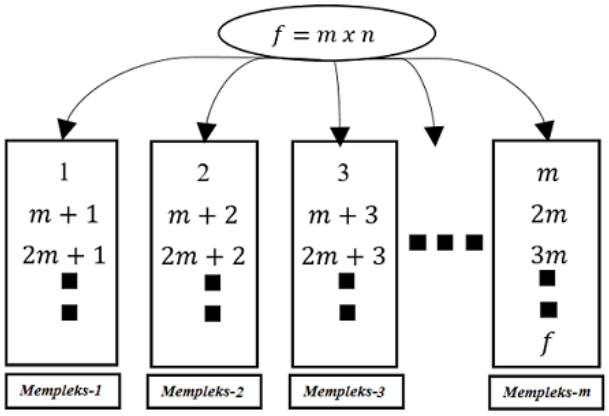
Kurbağaların doğadaki hareketlerinden esinlenen memetik tabanlı yeni bir yaklaşım olan Kurbağa Sıçrama Algoritması (*SFLA*), memetik evrimleri kullanarak yerel arama uzayında bir bireyden diğer bir bireye bilgi geçişini sağlar (Eusuff ve Lansey 2003, Eusuff ve ark 2006). *SFLA*'da ana popülasyon mempleks—*memeplex* olarak ifade edilen alt-gruplardan oluşur. Memetik evrim aşamasında oluşturulan her bir mempleks grubu diğer mempleks gruplarından bağımsız olarak kendi grupları içerisinde birbirlerine bilgi aktarımı gerçekleştirirler. Her mempleks grubu içerisinde belirlenen üye sayısının bir eksiği kadar yani her bir çevrimde alt-grup içindeki en iyi çözüme sahip kurbağa dışındaki üyelerin daha iyi çözümlere ulaşabilmesi için sahip oldukları

mem bilgileri güncellenir. Böylece *mempleklerde* uygun olmayan kurbağalar (*bireyler*) güncellenirken Şekil 3.14'deki "*lokal geliştirme süreci*" uygulanır. Tez çalışmasında önerilen yöntem Geliştirilmiş Kurbağa Sıçrama Algoritması (*Improved Shuffled Frog-Leaping Algorithm*) olarak isimlendirilmiştir ve *ISFLA* ile temsil edilmiştir. Bu algoritmanın geliştirilmiş olarak ifade edilmesinin temel sebebi, metasezgisel *SFLA* stratejisinin temel adımlarına ek olarak "*lokal geliştirme süreci*" olarak isimlendirilen önemli bir işlem adımını içermesindendir. Bu adım aynı zamanda *benzetim işlemi* olarak da isimlendirilir. Sonraki paragraflarda bu adım ayrıntılı olarak ele alınmaktadır. Ayrıca, algoritmanın çalışma mantığını gösteren basit diyagram Şekil 3.14'te sunulmaktadır.



Şekil 3.14. *ISFLA*'nın işleme mekanizması

Şekil 3.14'te ana popülasyonun üretilmesi işlemi $mekanizma - 1$'e göre gerçekleşir. Bu popülasyon aday çözümleri temsil eden kurbağalardan oluşur. Her kurbağanın uygunluğu da modülerlik skoruna göre hesaplanır. Bu skorlara göre Şekil 3.15'teki işlem temel alınarak ana popülasyon alt-popülasyonları temsil eden m adet memplekse ayrılır. Bu işlem en iyi bireyden başlanarak en kötü bireye doğru sırasıyla tüm mempleklere üyelerin atanması şeklinde gerçekleşir. Böylece aday çözümler adil ve homojen bir dağılımla alt-gruplara paylaştırılır.



Şekil 3.15. Mempleksl̇erin oluşturulması süreci

Şekil 3.15'te gösterilen f değişkeni toplam kurbağa (*aday çözümler*) sayısını; m, toplam mempleks sayısını ve n ise mempleksl̇erdeki üye sayılarını temsil eder. En iyi birey ilk memplekse; ikinci en iyi birey ikinci memplekse şeklinde çözüm kümesindeki son birey olan f kurbağası m. memplekse atanıncaya kadar bu işlemler döngüsel olarak devam eder. Burada 1. kurbağa en yüksek modülerlik skoruna sahipken; f kurbağası en düşük skora sahip bireyi temsil eder. Bu çalışmada, *ISFLA* için toplam kurbağa sayısı 30 ve mempleks sayısı ise 5 olarak belirlenmiştir.

Her bir memplekse üyeler atandıktan sonra Şekil 3.14'deki lokal geliştirme süreci ile devam edilir. Bu süreçte, her mempleks için en iyi birey ile bu gruptaki diğer bireyler arasında bir tür çaprazlama süreci olan *benzetim işlemi* uygulanır. Bunun için bir benzetim parametresi tanımlanmış ve ℓ ile temsil edilmiştir. Bu parametre 0.25 olarak alınmıştır. Bunun anlamı en iyi bireyin olasılıksal olarak dört birinin seçilen bireye aktarılmasıdır. Bir x bireyindeki komşu düğümlerin seçim işlemi Denklem 84'e göre yapılır. Bu denklemdeki "j" seçilen bireydeki düğümlerin indisini gösterir ve maksimum değeri N'dir. N ise ağdaki toplam düğüm sayısını gösterirken; r parametresi 0 ile 1 aralığında rastsal olarak üretilen sayıyı; $S_{i,j}$, i. bireyin j. düğümü için seçilen komşu düğümün indisini ve B_j, en uygun bireyin j. düğümünün seçilmiş olan komşu düğümünün indisini ifade eder.

$$S_{i,j} = \begin{cases} B_j, & eğer\ r \leq \ell\ ise \\ S_{i,j}, & aksi\ durumda \end{cases} \tag{84}$$

Lokal geliştirme sürecinde, tüm mempleksler için yukarıdaki işlemlere göre her bir birey kendi mempleksindeki en uygun bireye olasılıksal olarak benzetilir. Tüm bireyler için işlemler tamamlandıktan sonra mempleksledeki kurbağalar ana popülasyonda tekrardan bir araya getirilirler ve uygunluklarına göre sıralanırlar. Önerilen algoritmanın tüm işlem adımları belirlenen bir maksimum iterasyon (*jenerasyon*) sayısına kadar devam eder. Bu sayı testlerde 1000 olarak belirlenmiştir. Tüm işlemler sonunda popülasyondaki global en iyiyi ifade eden kurbağa, girdi olarak verilen ağdaki *ISFLA*'ya göre en uygun alt-gruplamayı sağlar. Bu ağdaki düğümlerin gruplamasına göre elde edilen ağ modülleri ise çıktı olarak verilir.

3.4.6. Adaptif Genetik Algoritma

Bu bölümde modül tespiti probleminin çözümü için *GA* (Goldberg ve Holland 1988, Holland 1992) temelli bir yöntem önerilmiştir. Bu yöntem ele alınan problemin yapısına uygun olarak ayarlanmış ve tez çalışmasında Adaptif Genetik Algoritma (*Adaptive Genetic Algorithm*, *AGA*) olarak isimlendirilmiştir. Önerilen algoritma lokal en iyi çözüme takılmadan en iyi global modülerlik değerine hızlı yakınsama özelliğine sahiptir. Burada standart *GA*'nın temel parametre ve operatörleri kullanılırken buna ek olarak operatörlerde probleme özgü değişiklikler yapılmış ve yönteme yeni parametreler dahil edilmiştir. Bu algoritmanın çalışma adımları ayrıntılı olarak aşağıda açıklanmıştır.

Popülasyonun oluşturulması aşamasında ilk olarak kromozomların temsili sunumu amacıyla *LAR* (Park ve Song 1998) yaklaşımı tercih edilmiştir. Popülasyondaki kromozomların boyutu ağdaki düğüm sayısı kadardır. Bir kromozomdaki genlerin sırası ağdaki düğümlerin indislerini; genlerin içerikleri ise her düğüm için seçilen komşu düğümleri gösterir. Örnek bir kromozomun oluşturulması ve gösterimi için Şekil 2.3'te verilen temsili örnek incelenebilir. Bu örnekte, giriş ağını temsil eden çizge yapısına uygun olarak üretilen kromozom yapısı Şekil 3.16'da gösterilmiştir.

1	2	3	4	5	6	7	8
4	7	8	1	1	8	2	3
1	2	3	1	1	3	2	3

Şekil 3.16. Örnek olarak üretilmiş kromozom yapısı

Burada, kromozom yapısı temsili olarak üç diziden oluşur. İlk dizi düğümlerin sıralı indislerini, ikinci dizi düğümlerin seçilen komşularını ve üçüncü dizi ise ikinci diziye göre oluşturulan ağ modüllerinin indislerini tutar. Bu örnekte 8 adet düğümden oluşan bir ağ için örnek kromozomun içeriği ikinci dizide verilmiştir. Buna göre 1. düğümün Şekil 2.3'e göre rastsal olarak seçilen komşusu 4. düğümdür. 1. düğümün ait olduğu modül numarası (*ID*) ise 1'dir. 2. düğüm için seçilen komşu düğüm 7'dir ve ait olduğu alt grup 2. modüldür. Bu şekilde son gene kadar tüm düğümler uygun komşuları seçerler. Son olarak, birinci dizideki son geni ifade eden 8. düğüm için seçilen komşu düğüm 3'tür ve ait olduğu modülün *ID*'si 3'tür. Bu şekilde tüm düğümler uygun modüllere dahil edilirler. Son durumda elde edilen modüllerin grafiksel gösterimi Şekil 2.3-c'de sunulmuştur. Burada anlatılan yönteme göre *AGA*'nın başlangıç kromozomları $mekanizma - 1$'e uygun olarak üretilir. Uygunluk fonksiyonu olarak modülerlik skoru referans alınmıştır. Bu skor (-1) ve (+1) arasında değişen değerler alır.

Bu algoritmada kromozomların geliştirilebilmesi amacıyla probleme uygun olarak güncellenen elitizm, seçim, çaprazlama ve mutasyon genetik operatörleri kullanılmıştır. Her bir operatör işleminde kullanılan parametreler uygun çözüme ulaşabilecek şekilde ayarlanmıştır. *AGA* algoritmasında standart genetik algoritmadan farklı olarak elitizm, çaprazlama ve mutasyon operatörlerine yeni parametreler dahil edilmiştir. Kullanılan operatörler ve parametreler aşağıda ayrıntılı olarak açıklanmıştır.

Elitizm: Bu operatör algoritmanın iki farklı aşamasında kullanılmıştır. Birinci aşamada popülasyondaki en iyi modülerlik değerlerine sahip $e \times 100$ adet kromozomun sonraki nesile aktarılması sağlanır. Buradaki e, belli sayıdaki en uygun kromozomun bir sonraki jenerasyona aktarılmasını sağlayan parametredir ve test çalışmalarında bu değer 0.05 olarak alınmıştır. İkinci aşamada ise üretilen ve daha iyi modülerlik değerlerine sahip yeni kromozomlar yine aynı oranda kötü kromozomlarla yer değiştirir. Kullanılan elitizm operatörü popülasyonda uygun çözümler sunmayan kromozomların elimine edilmesini sağlamaktadır. Buradaki kromozom çeşitliliğinin azaltılmaması için e değerinin küçük oranlarda seçilmesi önemlidir. Parametrenin değeri [0-1] aralığındadır.

Seçim: Yeni nesilin üretilmesi aşamasında birey seçim işlemi için rulet tekerleği seçimi (*RTS*) (Sastry ve ark 2014) kullanılmıştır. Bu işlem çaprazlama için uygun bireyin seçilmesini kapsar. Seçim işlemi şu şekilde gerçekleşir: İlk önce her kromozomun uygunluk değeri hesaplandıktan sonra tüm kromozomların toplam uygunluk değerini temsil eden t_F Denklem 85'e göre belirlenir. Burada i, kromozomun

popülasyondaki sırasını (indisini) tutar ve P ise toplam kromozom sayısını gösterir. İlk işlemden sonra her kromozomun seçilme olasılığı R, Denklem 86'ya göre hesaplanır. Üçüncü işlemle Denklem 87'deki hesaplamaya göre her kromozom için kümülatif olasılık değeri belirlenir.

$$t_F = \sum_{i=1}^{P} {F_M}^i \tag{85}$$

$$R_i = \frac{{F_M}^i}{t_F} \tag{86}$$

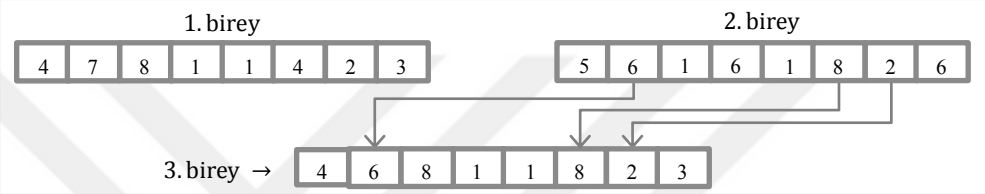
$$Q_i = \sum_{k=1}^{i} R_k \tag{87}$$

$$c = \{i, \quad eğer \;\; Q_{i\text{-}1} < r \leq Q_i \;\; ise\} \tag{88}$$

Son olarak, Denklem 88'e göre uygun aralığa gelen kromozom seçilir. Burada r, her kromozom için yeniden belirlenen ve 0 ile 1 aralığında üretilen rastsal sayıdır. c, seçilen kromozomu ifade eder. x^{i-1} ve x^i ise sırasıyla; popülasyondaki $(i-1)$. ve i. kromozomları gösterir. Eğer r değeri x^i'nin kümülatif oranından (Q_i) küçük veya eşitse ve x^{i-1}'in kümülatif oranından $\left(Q_{i\text{-}1}\right)$ büyükse i. kromozom seçilir ve sonucu tutan $c = x^i$ olur. Böylece bir sonraki işlem süreci için *RTS* yöntemi kullanılarak seçilecek ikinci kromozom çaprazlamaya alınır. Bu süreç her bir kromozom için tekrarlanır.

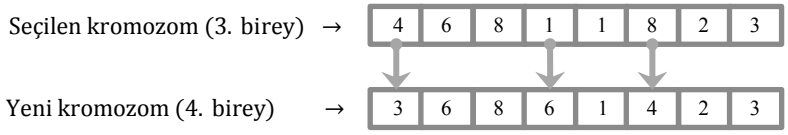
Çaprazlama: Yeni nesilin üretilmesi aşamasında iki birey arasında bilgi alışverişi için bu operatör kullanılır. Bunun için iki bireyden ilki (1. birey) popülasyondaki kromozomlardan sırasıyla temin edilir. Ancak burada tüm bireyler işleme alınmaz. Bu operatör için tanımlanmış *çaprazlama oranı* parametresi ile popülasyondaki belli sayıda birey işleme alınır. Burada çaprazlama oranı "c" ile temsil edilir ve tez çalışmasında bu değişkenin değeri 0.8 olarak alınmıştır. Toplam kromozom sayısı P ise çaprazlama için seçilecek popülasyondaki toplam kromozom sayısı, $P \times c$ kadardır. Bu operatör için seçilecek 2. birey ise popülasyondaki kromozomların uygunlukları temel alınarak oluşturulan kümülatif toplama göre seçim işleminde belirlenir. Daha sonra iki birey arasında bilgi alışverişi oranını belirleyen *çaprazlama sayısı* parametresi tanımlanır. Bu parametre s ile gösterilir. Giriş ağının düğüm sayısı N ile temsil edilirse çaprazlanacak

toplam gen oranı $[s \div N]$ ile belirlenir. Bu çalışmada maksimum s değeri N'nin yarısı oranında belirlenmiştir. Böylece *RTS* ile seçilmiş uygun kromozomun (2. birey) tüm genlerinden rastsal olarak seçilen s tanesi üretilecek yeni kromozoma (3. birey) aktarılır. İlk kromozomdaki (1. birey) diğer genler 3. bireye olduğu gibi aktarılır. Bölüm 2.1.1'de sunulan Şekil 2.3-a'daki temsili çizgeden üretilen bireyler arasında gerçekleşen çaprazlama işlemi Şekil 3.17'de gösterilmiştir. Buradaki örnekte $N = 8$ değeri için s sayısı 3 olarak belirlenmiştir. Ok işaretleri *RTS* ile seçilen kromozomdan üç adet genin yeni kromozoma aktarılmasını gösterir. Diğer genler ilk kromozomdan alınır. Üretilen kromozomdaki düğümlerin komşuluk listelerine uygun olmasına dikkat edilir.



Şekil 3.17. *AGA*'ya göre çaprazlama işlemi

Mutasyon: Bu operatör işlemi çaprazlama ile üretilen kromozomlardan bazılarının olasılıksal mutasyona tabi tutulmasını sağlar. Bu amaçla değişen oranlarda çoklu mutasyon uygulanır. Üretilen yeni kromozomlardan mutasyona uğrayacakların seçilebilmesi için bir mutasyon oranı parametresi tanımlanır. Bu parametre m ile ifade edilir ve düşük oranlarda seçilir. Tez çalışmasında bu oran 0.1 olarak alınmıştır. Böylece popülasyon boyutu 30 iken genel olarak sadece 3 birey mutasyon işlemine alınacaktır. Buradaki işlemde mutasyona uğrayacak gen sayısını belirleyen ikinci bir parametre tanımlanmış ve g ile temsil edilmiştir. Ağdaki toplam düğüm sayısı N ile ifade edildiğinde; g parametresi 1 ile $(N/4)$ arasında rastsal olarak seçilir. Böylece en az 1 ve en çok düğüm sayısının dörtte birisi kadar gen mutasyona uğrar. Herhangi bir kromozomdaki temel gen yapısının korunması amacıyla tüm genlerden mutasyon işlemi için seçilecek maksimum gen sayısı $(N/4)$ oranıyla kısıtlanmıştır. Şekil 3.18'de *AGA*'nın Şekil 2.3-a'daki çizgeye uygun olarak uygulanan mutasyon operatörüyle ilgili temsili bir örnek verilmiştir. Bu örnekte $g = 3$ olarak belirlenmiştir.



Şekil 3.18. Mutasyon operatörü için uygulanan örnek işlem

Şekil 3.18'de mutasyona uğrayan genlerin konumları kalın oklarla gösterilmiştir. Bu örnekte, 1. düğümün komşusu olarak seçilmiş olan 4. düğüm ilk mutasyon işlemi sonrası 3. komşu düğümle yer değiştirmiştir. Sonraki mutasyon işlemiyle 4. gen (4. düğüm) konumundaki 1. komşu düğüm 6. komşu düğümle değiştirilmiştir. Son olarak dizideki 6. konumu temsil eden altıncı düğümün atanmış olan komşusu 8. düğüm mutasyon işlemiyle 4. komşu düğüme dönüştürülmüştür. Bu işlemle üçüncü birey üç noktalı mutasyona uğrayarak yeni bir birey (4. birey) oluşturur. Seçilen kromozomlar için bu operatör tekrar çalıştırılarak yeni nesilin popülasyonu oluşturulmuş olur. Daha sonra her bir birey için uygunluk değeri hesaplanır ve en iyi lokal ve global bireyler kaydedilerek sonraki nesile aktarılır. Bu işlem tanımlanmış maksimum iterasyon sayısı kadar tekrarlanır. İşlemler sonunda kaydedilmiş olan en iyi modülerlik skoruna sahip bireye göre ağın modül yapısı ortaya çıkarılır.

3.4.7. Dağınık Arama Temelli Algoritma

Bu bölümde açıklanan ve *SSA* (*Scatter Search based Algorithm*) olarak ifade edilen Dağınık Arama Temelli Algoritma, P adet üyeli popülasyondan (*DiverseSet*) belli sayıdaki en iyi kromozomun (referans kümesi, *ReferenceSet*) dağınık arama (Glover 1977, Marti ve ark 2006) temelli bir yöntemle geliştirilmesini kapsar. Bu algoritmanın başlangıç aşamasında, birey üretimi $mekanizma - 1$ diye isimlendirilen bir yaklaşımla gerçekleştirilir. *SSA* ile referans kümesinden rastsal olarak seçilen bireylerin (*alt-kümeler*) problemin yapısına uygun olarak önerilen *GA*'daki (Goldberg ve Holland 1988, Holland 1992) çaprazlama ve mutasyon işlemlerine alınması, birey geliştirme aşamasını oluşturur.

SSA'nın genel çalışma prensibi en iyi bireylerin diğer bireylere oranla daha fazla geliştirilmesine dayanır. Bu algoritmanın uygulanan işlem adımlarını içeren sözde-kod Çizelge 3.11'de gösterilmiştir. Bu çizelge, Şekil 3.11'deki genel işlem adımlarına uygun olarak hazırlanmıştır. Buna göre önerilen algoritmanın ilk aşaması $mekanizma - 1$

kullanılarak ana popülasyonun oluşturulmasıdır. Şekil 3.16'da temsili olarak oluşturulan bir birey yapısı incelenebilir.

Çizelge 3.11. *SSA*'ya göre işlem adımlarını gösteren sözde-kod

1. Ağ topluluk tespiti problemine ve algoritmaya özgü parametreleri tanımla
2. Ana popülasyonunu $(DiverseSet)$ *"*$mekanizma - 1$*" göre üret*
3. Bireylerin modülerlik skorlarını hesapla → F_M^i
4. Bireyleri uygunluklarına göre iyiden kötüye doğru sırala (1)
5. x *sayacına 1 sayısını ata* $(x = 1)$
6. Döngüyü başlat $(x < x_{max})$
7. $\mathcal{T}$ *adet bireye sahip referans kümesini* $(ReferenceSet)$ *oluştur*
8. Referans kümesinden rastsal olarak alt kümeleri $(subSets)$ *oluştur*
9. Alt kümelere Bölüm 3.4.6'daki çaprazlama operatörünü uygula
10. Çaprazlama sonrası bireyleri, referans kümesinde birleştir
11. Referans kümesindeki bireyleri "$m - mekanizma$*" ile mutasyona uğrat*
12. Referans kümesi dışındaki K adet bireyi al → $(ekPop)$
13. $ekPop$*'taki bireylere Bölüm 3.4.6'daki mutasyon operatörünü uygula*
14. Ana popülasyonu güncelle → $DiverseSet = ReferenceSet + ekPop$
15. Bireyleri uygunluklarına göre iyiden kötüye doğru sırala (2)
16. En iyi lokal ve global bireyleri seç
17. $x = x + 1$
18. $(x == x_{max})$ *şartı sağlanırsa döngüyü sonlandır*
19. Global en iyi bireye göre ağ modüllerini oluştur

Çizelge 3.11'e göre ana popülasyondaki tüm birey sayısı P ile gösterilsin. $\mathcal{T}$ ise referans kümesindeki toplam birey sayısıdır. Ana popülasyon üretildikten sonra bu popülasyondan alınan $(\mathcal{T} = \rho \times P)$ adet birey referans kümesini oluşturur. Burada ρ, referans kümesine seçilecek en iyi bireylerin seçim oranını gösterir ve tez çalışmasında bu oran 0.4 olarak alınmıştır. Bu işlem için ana popülasyondaki tüm bireyler modülerlik skorlarına göre yüksekten düşüğe doğru sıralanır ve en iyi $\mathcal{T}$ tanesi seçilir.

Sonraki işlem adımında, referans kümesinden rastsal olarak seçilen bireylerle $[n \times \mathcal{T}]$ adet alt-küme oluşturulur. n, alt-kümelerin üretim oranını gösterir ve tez çalışmasında, bu parametrenin değeri 0.3 olarak belirlenmiştir. Örneğin, toplam birey sayısı $P = 30$ seçilirse; referans kümesindeki birey sayısı $[\mathcal{T} = 0.4 \times 30 = 12]$; alt-küme sayısı ve bu alt-kümelerdeki birey sayıları sırasıyla; $[0.3 \times 12 = 4]$ ve $[12/4 = 3]$ olur. Buradaki kesirli ifadeler bir üst doğal sayıya yuvarlanmıştır. Ayrıca, bireylerin geliştirilmesi için aşağıdaki mutasyon ve çaprazlama işlemleri uygulanmıştır.

Alt-kümelerdeki her bir bireyin geliştirilmesi için öncelikle Bölüm 3.4.6'daki çaprazlama operatörü *SSA* algoritması için de uygulanır. Burada sadece *AGA*'dan farklı olarak seçilen 2. birey kendi alt-kümesinden rastsal olarak seçilir. Örnek çaprazlama

operatörü Şekil 3.17'de gösterilmiştir. Bu operatör referans kümesindeki tüm alt-kümelerin bireyleri için uygulanır. Çaprazlama işlemi için her alt-kümedeki bireyler kendi alt-kümelerindeki bireylerle çaprazlanırlar. Burada 1. birey, alt kümedeki sırasına göre seçilen ilk bireyi; 2. birey, çaprazlama işlemi için yine kendi alt kümesinden rastsal olarak seçilen ikinci bireyi temsil eder. Bu işlem sonrası mutasyon operatörüne geçilir.

Yukarıdaki çaprazlama işlemi tamamlandıktan sonra tüm alt-kümeler yeni referans kümesini oluşturur ve mutasyon işlemine geçilir. *SSA*'nın önerilen mutasyon işlemi diğer algoritmaların sahip olduğu mutasyon işlemlerinden farklıdır. Bu işlem "$m-mekanizma$" olarak isimlendirilmiştir. Referans kümesindeki her bir birey modülerlik değerlerine göre yüksekten düşüğe doğru sıralanır ve Denklem 89'da gösterildiği gibi bunlara sırasıyla 1'den $\mathcal{T}$'ye kadar indeksler atanır. Bu indekslere göre Denklem 90'daki hesaplama ile her bir bireyin mutasyon işlemine alınacak düğüm sayısı belirlenir. Denklem 89'da her bir bireyin indeksi (*sırası*) ve sahip olduğu modülerlik skoru gösterilmiştir. Burada i, referans kümesindeki bireyleri gösterirken; d_i, i. bireyin indeksini; $\mathcal{T}$, bu kümedeki toplam birey sayısını; F_i, i. bireyin uygunluk değerini; F_1, en yüksek skoru ve $F_{\mathcal{T}}$ ise en düşük skoru temsil eder.

$$\begin{matrix}\langle indeksler \rightarrow d_i \rangle & : 1, & 2, \dots, \mathcal{T} \\ \langle skorlar \rangle & : F_1, & F_2, \dots, F_{\mathcal{T}}\end{matrix} \quad , \quad F_i > F_{i+1}, \forall i \in ReferenceSet \tag{89}$$

Denklem 90'da ise i. birey için mutasyon ile değiştirilecek düğüm (*gen*) sayısı gösterilir. Burada N, bireyi oluşturan toplam düğüm sayısını; M, mutasyon oranını; r_i^1, i. birey için 0 ve 1 arasında üretilen rastsal seçim oranını; r^2, [0,1] aralığındaki rastsal mutasyon miktarı oranını ve $d_{\mathcal{T}}$ ise en kötü skora sahip bireyin indeksini gösterir. Sonucu tutan m_i, i. bireydeki mutasyona uğrayacak düğüm sayısını ifade eder. Burada elde edilen ondalık sayılar bir üst doğal sayıya yuvarlanır. Bir bireyin mutasyona uğratılması sırasında temel gen yapısının korunması amacıyla bireyde maksimum değiştirilecek düğüm sayısı, $\left(r^2 \times (N \div 4)\right)$ oranıyla kısıtlanmıştır.

$$m_i = r^2 \times \left(\frac{d_i \times (N \div 4)}{d_{\mathcal{T}}}\right), \quad eğer \ (r_i^1 < M) \ ise \tag{90}$$

Denklem 90'a göre referans kümesinde mutasyon işlemine alınacak bireyler $(r_i^1 < M)$ şartına göre seçilir. Bu işlemde, her bir birey için rastsal olarak 0 ve 1

arasında bir sayı (r_i^1) üretilir. Bu sayı sabit olarak belirlenmiş mutasyon oranından (M) küçükse $i.$ birey için mutasyon işlemi gerçekleşir; aksi durumda sonraki bireye geçilir. Burada şart sağlandığında, bu bireyin Denklem 89'daki sırası dikkate alınarak Denklem 90'a göre mutasyona uğrayacak gen sayısı belirlenir. Daha sonra bu sayı kadar rastsal olarak konumlar seçilir ve mutasyonlar gerçekleşir. Böylece tüm bireyler için mutasyon şartına göre işlem yapılarak sonraki adıma geçilir.

Bir sonraki işlemde referans kümesine seçilmeyen $[K = (1 - \rho) \times P]$ adet birey grubu $(ekPop)$ için çaprazlama işlemi yapılmadan doğrudan *AGA* algoritmasının Bölüm 3.4.6'da anlatılan mutasyon işlemi uygulanır. Burada çaprazlama işleminin yapılmamasının temel sebebi bu bireylerin uygun olmayan modülerlik değerlerine sahip olmalarından dolayı birbirleriyle bilgi alışverişinin gelişimi sağlayamamasıdır. Bunun yerine doğrudan çoklu mutasyon operatörü ile işlem yapılır. Böylece $P = 30$ için $\mathcal{T} = 12$ adet birey, referans kümesi ile geliştirilirken; kalan $K = 18$ bireyin uygunlukları buradaki mutasyon ile iyileştirilmeye çalışılır.

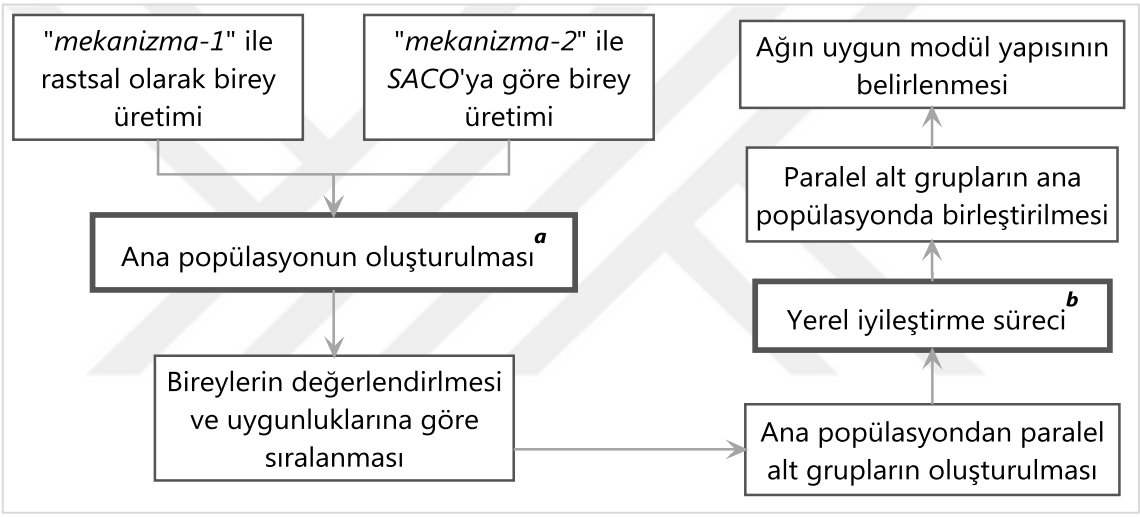
Yukarıdaki işlemler sonunda en başarılı çözümleri sunan $(ReferenceSet)$ ve en başarısız sonuçları veren $(ekPop)$ iki grup tekrardan ana popülasyonda $(DiverseSet)$ birleştirilir. Popülasyondaki bireyler uygunluklarına göre sıralandıktan sonra tekrardan referans kümesine en iyi $\mathcal{T}$ adet birey seçilir ve bu işlemler döngüsel olarak devam eder. Çizelge 3.11'in 6. ve 18. adımındaki x_{max}, sonlandırma kriteri olan maksimum nesil sayısını gösterir. Burada x sayacı x_{max}'a ulaştığında $(x == x_{max})$ döngü sonlandırılır.

3.4.8. Üç Fazlı Hibrit Yaklaşım

Ağ modül tespiti probleminin çözümü amacıyla önerilen son yöntem Üç Fazlı Hibrit Yaklaşım (*Three-Phase Hybrid Approach*) olarak isimlendirilen ve bu çalışmada "$3pHybrid$" ile ifade edilen algoritmadır. Bu yöntem; kurbağa sıçrama algoritmasını (Eusuff ve Lansey 2003, Eusuff ve ark 2006), karınca kolonisi optimizasyonunu (Dorigo ve ark 1991, Dorigo 1992, Dorigo ve ark 1996, Dorigo ve Gambardella 1997) ve genetik algoritmayı (Goldberg ve Holland 1988, Holland 1992) içeren hibrit bir yaklaşım sunar. Bu sebeple *3pHybrid* algoritması üç aşamalı (*üç fazlı*) hibrit bir yaklaşım olarak ifade edilmiştir. Bu algoritma diğer önerilen algoritmalara göre daha başarılı sonuçlar sunmuştur. Bu algoritmanın diğerlerine göre başarı testi *Grup-A* kategorisindeki ağlar üzerinde gerçekleştirilmiştir. Bu kategorideki tüm test sonuçlarına

göre bir diğer test grubunu temsil eden *Grup-B* kategorisinde kullanılacak en uygun algoritma *3pHybrid* olarak belirlenmiştir. Tüm test sonuçları dördüncü bölümde ayrıntılı olarak sunulmaktadır.

Bu algoritmanın genel çalışma prensibi Şekil 3.19'da gösterilmiştir. Buradaki işlemleri özetleyen sözde-kod ise Çizelge 3.12'de sunulmuştur. *3pHybrid* algoritması *SFLA*'nın temel çalışma mekanizmasına dahil edilen *ACO*'nun feromen güncellemesi mantığına dayanan birey üretim işlemiyle ve *GA*'da birey iyileştirme sürecini oluşturan *seçim—çaprazlama—mutasyon* mekanizmaları ile zenginleştirilmiştir. Ayrıca, burada *3pHybrid* için tanımlanmış parametre değerleri *SACO*, *ISFLA* ve *AGA* algoritmaları için atanmış değerlerle aynıdır.



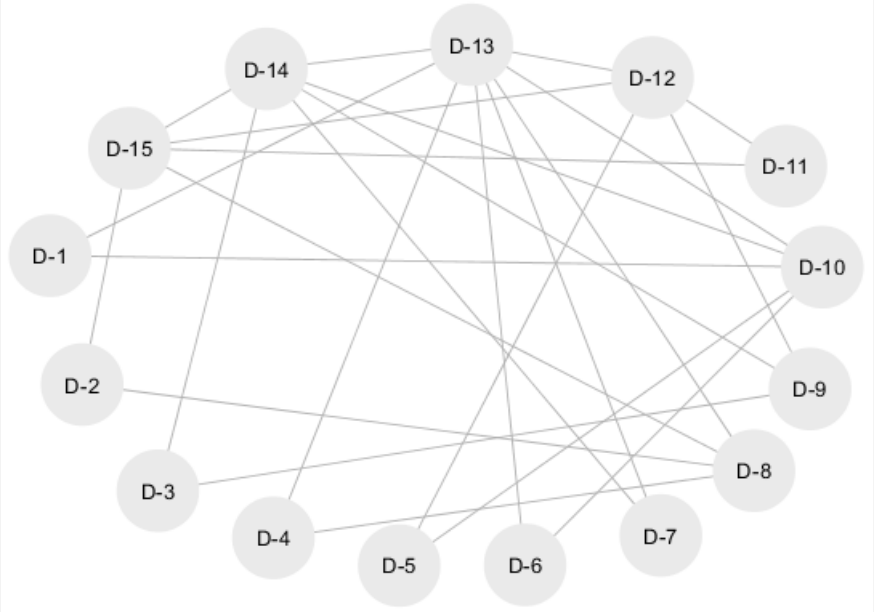
Şekil 3.19. *3pHybrid* algoritmasının genel işlem adımları

Bu bölümde önerilen algoritma metasezgisel *SFLA* stratejisinin temel adımlarına ek olarak iki önemli işlem adımını içerir. İlk adım ana popülasyondaki bireylerin üretilmesi sürecini (Şekil 3.19-a) içerirken; ikinci adım her bir paralel gruptaki (*mempleks*) bireyler için "yerel arama" sürecinden "global arama" sürecine doğru birey geliştirme sürecini (Şekil 3.19-b) kapsar. Bu işlemler *yerel iyileştirme sürecini* temsil eder. Bunlar aşağıda açıklanmıştır.

Ana popülasyonun oluşturulması süreci (***a***): *3pHybrid* algoritmasının başlangıç popülasyonundaki aday çözümlerin üretilmesi amacıyla bu tez çalışmasında önerilen iki farklı mekanizma bir arada kullanılmıştır: "$mekanizma-1$" ve "$mekanizma-2$". Ana popülasyondaki elemanlar; bireyler, aday çözümler veya üyeler olarak isimlendirilmiştir. Bu popülasyondaki toplam eleman sayısı P ile gösterilirse; ana

popülasyonun yarısı $(P/2)$, $(mekanizma-1)$ ile oluşturulurken; diğer bir yarısı $(P/2)$, $(mekanizma-2)$ ile oluşturulur. İlk olarak aday çözümlerin çeşitliliğinin arttırılması ve algoritmanın yerel en iyi çözümlere takılmaması için ana popülasyona sadece komşuluk matrisine uygun olarak rastgele üretilmiş aday çözümler eklenir $(mekanizma-1)$. Bu süreçteki birey üretim işlemi önceki algoritmalarda da kullanılmış ve bu konu hakkında kısa bilgiler ilgili bölümlerde sunulmuştur. Ayrıca "$mekanizma-1$" ile birey üretim sürecinin daha iyi anlaşılabilmesi için gerekli ek bilgiler ve örnekler bu bölümde de verilmiştir. Bir diğer birey üretim süreci olan "$mekanizma-2$" ile birey üretim işlemleri ise Bölüm 3.4.2'deki *SACO* algoritmasının işlem adımlarında örneklerle açıklanmıştır.

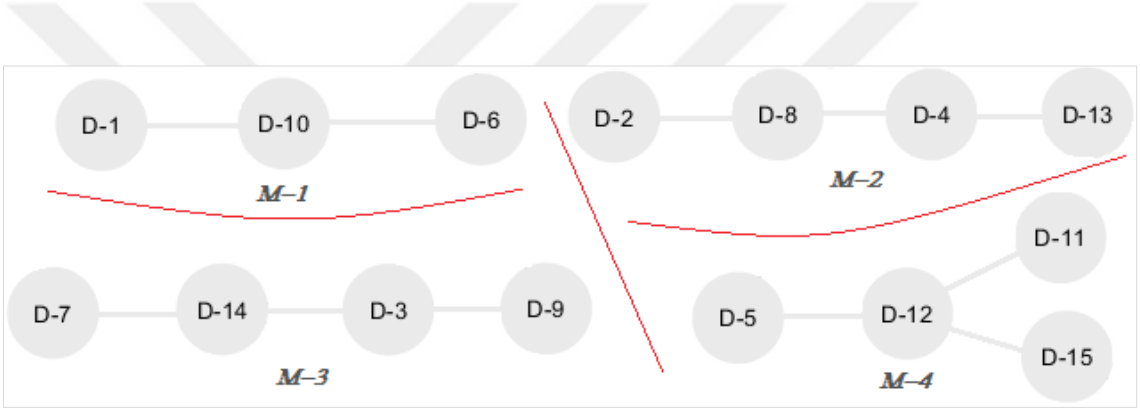
Şekil 3.20'de *BA* modeli ile oluşturulmuş örnek bir yönsüz ve ağırlıksız çizge verilmiştir. Bu çizge 15 düğüme ve 26 bağlantıya sahiptir. Düğümler $D-1$'den $D-15$'e kadar sıra numaralarıyla isimlendirilmiştir.



Şekil 3.20. 15 düğümlü yönsüz ve ağırlıksız bir çizge

Burada 15 düğüme uygun bir aday çözümün üretilmesi işlemi şöyle gerçekleşir. İlk aşamada, ilk düğüm için kendi komşuluk listesinden bir komşu rastsal olarak seçilir. Çizgedeki 1. düğümün $(D-1)$ komşuları: $\{D-10,\ D-13\}$'tür. Birinci düğümün seçilen komşusu rastgele bir şekilde "$D-10$" olarak seçilsin ve sonraki düğümlerin komşulukları da benzer şekilde belirlensin. Son düğüm $(D-15)$ için komşuluk listesi: $\{D-2,\ D-8,\ D-11,\ D-12,\ D-14\}$ olur. Bu liste içinden rastsal olarak 12.

düğümün seçildiği farz edilsin. Buna göre 15. düğümün dahil edildiği modülün numarası 4 olur. Son olarak, oluşturulan 15 boyutlu bir bireyin dizi yapısı şu şekilde gösterilir: $[1_1^{10}\ 2_2^8\ 3_3^{14}\ 4_2^8\ 5_4^{12}\ 6_1^{10}\ 7_3^{14}\ 8_2^2\ 9_3^3\ 10_1^6\ 11_4^{12}\ 12_4^5\ 13_2^4\ 14_3^7\ 15_4^{12}]$. Bu dizide ilk düğüm için seçilen komşu düğüm ve ait olunan modül şu şekilde ifade edilir: 1_1^{10}. Bu gösterimdeki taban, birinci düğümü; üst ifade, bu düğümün seçilen komşusunu $(D-10)$; alt ifade ise bu düğümün ait olduğu ağ modülünün numarasını gösterir. 1. düğümün ait olduğu modülün numarası 1'dir ve bu modül $(M-1)$ ile ifade edilir. Üretilen bireydeki tüm düğümlerin seçilen komşularına göre oluşturulan modül yapıları Şekil 3.21'de gösterilmiştir. Dizide gösterilen komşuluklara göre oluşturulan modül sayısı 4 olur ve bunlar sırasıyla; $(M-1)$, $(M-2)$, $(M-3)$ ve $(M-4)$ ile isimlendirilir. Modüller arası düğümler arasındaki sınırlar *ara çizgilerle* vurgulanmıştır.



Şekil 3.21. Üretilen aday çözümün elde edilen ağ modülleri

Modüller içindeki düğümlerin birbirleriyle ilişkileri açık gri renk çizgilerle gösterilmiştir. Kırmızı çizgilerle ayrılan modüllerin üyeleri kendi grubundakilerle maksimum ilişkilere sahipken; diğer modüllerin üyeleriyle daha az ilişkidedirler veya bunlarla hiçbir ilişkiye sahip değillerdir. Burada belirtilen durum ağ modül tespiti probleminin çözümünde ana hedefi oluşturur. Bu bölümde sunulan *3pHybrid* algoritması da burada verilen örnekteki gibi ağdaki düğümlerin en uygun şekilde gruplandırılmasını amaçlar. Modülerlik amaç fonksiyonu da burada belirtilen en uygun gruplandırmayı matematiksel modelleme ile temsil eden bir fonksiyondur. Bu fonksiyonun çıktı değerinin maksimum yapılması amaçlanır ve bu amacı gerçekleştiren yaklaşımlar da önerdiğimiz algoritmaların nihai hedeflerini oluşturur.

Yukarıda örnek olarak gösterilen birey üretim aşaması bu çalışmada sunulan $mekanizma-1$'i temsil eder. Bu mekanizma ile ana popülasyondaki bireylerin yarısı üretilen örnek bireye benzer şekilde oluşturulur. Diğer yarısı ise $mekanizma-2$

kullanılarak temin edilir. Böylece Şekil 3.19-a'daki süreç tamamlanır ve *P* boyutlu bir başlangıç popülasyonu oluşturulmuş olur.

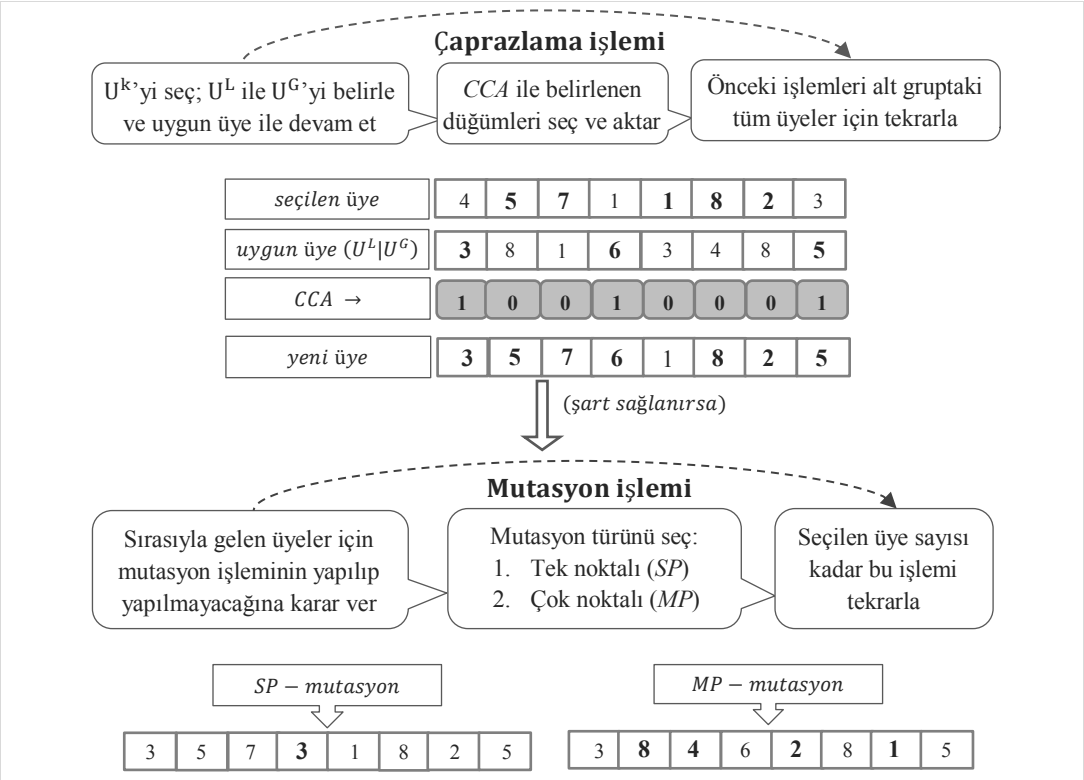
Sonraki süreçte her bir aday çözümün uygunluk değeri hesaplanır. Buradaki uygunluk kriteri olarak *Grup-A* kategorisi ağlarındaki testler için sadece modülerlik amaç fonksiyonu; *Grup-B* kategorisi ağlarındaki testler için Bölüm 2.1.3'teki 10 farklı fonksiyon kullanılmıştır. Bu durum Şekil 3.10'da gösterilmiştir. Bu şekilde ifade edilen 5. adımda sadece *3pHybrid* algoritmasının kullanıldığı işlem gösterilmiştir. Bir sonraki işlem adımı olan 6. adımda ise bu algoritmanın kullanılmasıyla elde edilen test sonuçları gösterilir. *Grup-B*'de verilen ağlar için diğer algoritmalar yerine bu algoritmanın seçilmesinin temel sebebi *Grup-A*'daki sonuçlara göre *3pHybrid* algoritmasının en iyi test sonuçlarına ulaşmasıdır. Deneysel Çalışmalar bölümünde, elde edilen bu sonuçlar ayrıntılarıyla sunulmuş ve değerlendirilmiştir. *3pHybrid* algoritması kullanılarak *Grup-A* kategorisindeki tüm ağlarda diğer algoritmalarda anlatıldığı gibi modülerlik uygunluk fonksiyonunun maksimize edilmesi amaçlanmıştır. *Grup-B* kategorisindeki ağlardan 10 amaç fonksiyonunun kullanılmasıyla elde edilen modül yapılarının kaliteleri Bölüm 2.1.4'teki değerlendirme fonksiyonlarıyla analiz edilmiştir. Böylece ilk olarak en uygun sonuçlara ulaşan algoritma tespit edilmiş ve daha sonra da bu algoritma kullanılarak en uygun amaç fonksiyonu belirlenmiştir. Bu işlem adımları Çizelge 3.12'de gösterilmiştir. *Grup-A* ve *Grup-B* kategorisinde yapılan testlerde kullanılan *3pHybrid* algoritmasının bundan sonraki birey iyileştirme süreci de dahil diğer tüm işlem adımları aynı süreçleri izler ve bu süreçler Şekil 3.11'deki akış şemasına uygun bir şekilde gerçekleştirilir.

Bir sonraki işlem adımında popülasyondaki bireyler uygunluklarına göre sıralanırlar. Daha sonra *ISFLA* algoritmasının ana popülasyonu mempleklere ayırma stratejisine (Bölüm 3.4.5'e bakınız) göre bu popülasyondaki bireyler uygun paralel alt-gruplara dahil edilirler. Bu işlem kullanılan amaç fonksiyonlarına göre oluşturulan popülasyonlar kullanılarak Şekil 3.15'e göre yapılır. Her bir fonksiyon için algoritma kendi popülasyonuna sahip olur. Bu adımdaki işlem, sırası gelen amaç fonksiyonu için oluşturulan ana popülasyona uygulanır. Paralel alt-gruplar oluşturulduktan sonra her bir aday çözümün uygunluğunun geliştirilmesi amacıyla *yerel iyileştirme sürecine* geçilir.

Yerel iyileştirme süreci (***b***): Bu süreçte gerçekleşen işlemler Şekil 3.19-b'deki adımı temsil eder. Bu adım normalde *PSO* algoritmasının hız ve konum güncellemesine benzerlik gösteren *SFLA*'nın yerel arama işlemlerini içerir. *Eusuff* ve arkadaşları kendi çalışmalarında (Eusuff ve Lansey 2003, Eusuff ve ark 2006), *SFLA*'daki yerelde

gerçekleşen işlemler ve karıştırma—*shuffling* yapısı sayesinde bireyler (*temsili kurbağalar*) arasında gerçekleşen bilgi değişimiyle genel en iyiye ulaşmayı amaçlamışlardır. Tez çalışmasında, bu işlem adımı problemin yapısına uygun olarak değiştirilmiştir. Şekil 3.22'de *yerel iyileştirme sürecini* temsil eden bir grafik verilmiştir.

Çaprazlama işleminde alt-gruplardan işleme alınan sıralı bireyler (*SFLA*'da kurbağalar) üç farklı duruma göre çaprazlamaya tabi tutulur. İlk durumda, seçilen üye kendi grubundaki en iyi üyeye (*yerel en iyi*) göre güncellenir. İkinci durumda, birinci durumdaki işleme göre daha iyi bir birey elde edilmezse bu üye genel en iyi üyeye (*global en iyi*) göre güncellenir. Global en iyi üye tüm popülasyondaki en uygun amaç fonksiyonu değerine sahip bireyi temsil eder. Son durumda ise önceki durumlara göre herhangi bir güncelleme olmazsa (*daha iyi birey elde edilmezse*) seçilen üye yerine rastgele yeni bir üye "$mekanizma - 1$" ile oluşturulur. Şekli 3.22'deki "U^k", Her bir alt-grubun sırasına göre seçilen k. bireyi gösterir. Burada k, 1'den başlayarak ilgili paralel alt-gruptaki en son üyeye kadar seçilen bireyleri temsil eder. "U^L", ilgili alt-gruptaki lokal en iyi bireyi; "U^G", global en iyi bireyi; "$uygun$ üye $(U^L|U^G)$" ise işleme alınan yerel veya global en iyiyi ifade eder. Mutasyon işlemindeki "$SP - mutasyon$" ve "$MP - mutasyon$" ifadeleri uygulanacak mutasyon türünü gösterir ve sırasıyla; tek noktalı (*single-point, SP*) ve çok noktalı (*multi-point, MP*) mutasyonları temsil ederler. *CCA* dizisi ise çaprazlama değişim dizisini (*crossover change array*, *CCA*) gösterir.



Şekil 3.22. *3pHybrid* algoritmasında uygulanan çaprazlama ve mutasyon operatörleri

Önerilen algoritma tarafından kullanılan çaprazlama ve mutasyon operatörleri her bir mempleks için Şekil 3.22'ye uygun olarak gerçekleştirilir. Çaprazlama işleminde kullanılan *CCA* dizisi aday çözümün uzunluğundadır ve içeriği rastsal olarak belirlenir. Bu dizi 0 ve 1'lerden oluşur. Dizinin bir hücresindeki değer 0 ise seçilecek düğüm ilk üyeden alınır. Eğer bu değer 1 ise ikinci üyenin ilgili düğümü yeni üyeye aktarılır. İkinci üyenin belirlenmesi yukarıda açıklanan üç farklı duruma göre yapılır. Bu işlemler sonucu çaprazlama süreci tamamlanır ve yeni birey üretilmiş olur. Bu yaklaşımla yeni birey, kendi türetildiği üyenin ve ait olduğu alt-gruptaki ya da tüm popülasyondaki en iyi üyenin kombinasyonu sonucu üretilir. Daha sonra mutasyon işlemine geçilir.

Mutasyon sürecinde iki farklı duruma göre işlem yapılır. Eğer popülasyondaki bir birey, uygun modüller öneriyorsa daha uygun bir çözüme yakınsayabilmesi için küçük iyileştirmeler amacıyla tekli mutasyona $(SP - mutasyon)$ tabi tutulur. Ancak bu birey uygun olmayan bir gruplama öneriyorsa bireyde daha fazla değişimler sağlayan çoklu mutasyon $(MP - mutasyon)$ işlemi gerçekleştirilir. Örneğin birinci üye, seçilen örnek bir bireyi; ikinci üye, ait olduğu mempleksteki en uygun çözümü sunan bireyi temsil etsin. Rastgele üretilen *CCA* dizisine göre elde edilen yeni birey (*yeni üye*) eğer

şart sağlanırsa iki tür mutasyon işleminden birine tabi tutulur. Buradaki şart Denklem 91'i ve Denklem 92'yi ifade eder.

$$S = \begin{cases} 1, & eğer\ (r^k < M)\ ise \\ 0, & aksi\ durumda \end{cases} \tag{91}$$

$$P^k = \begin{cases} 0, & 1.koşul \rightarrow\ (S == 1)\ ve\ (F^k \geq F^o)\ ise \\ 1, & 2.koşul \rightarrow\ (S == 1)\ ve\ (F^k < F^o)\ ise \end{cases} \tag{92}$$

Mutasyon işlemine geçilebilmesi için $k.$ birey için ilk koşul $r^k < M$'dir. Burada r^k ve M sırasıyla; $k.$ birey için [0-1] arasında üretilen rastsal sayıyı ve algoritmanın mutasyon oranını gösterir. S ise $k.$ birey için mutasyon işleminin gerçekleştirilip gerçekleştirilmeyeceğini belirleyen parametredir. Eğer ilgili birey $(k.)$ için üretilen r^k sayısı M sayısından küçükse mutasyon işlemi gerçekleşir. Bu durumda S parametresinin değeri 1 olur; aksi durumda 0 olur. Hangi tür mutasyon türünün seçileceği de Denklem 92'ye göre belirlenir. Bu denklemdeki F^k, $k.$ bireyin uygunluk değeridir. F^o ise bu bireyin ait olduğu paralel alt-gruptaki tüm bireylere göre belirlenen ortalama uygunluk değeridir. Mutasyon işlemi sonucu ise P^k ile temsil edilir. Buna göre eğer mutasyon işlemine geçilmişse $(S == 1)$ ve $k.$ bireyin uygunluk değeri ortalama değerden fazla $(F^k \geq F^o)$ ise bireyin uygun çözüme yaklaştığı farz edilerek $P^k = 0$ olur. Bu durumda sadece tek bir düğümün komşuluğunun değiştirildiği "$SP - mutasyon$" işlemi gerçekleşir. Aksi durumda, $P^k = 1$ olur ve "$MP - mutasyon$" ile çoklu mutasyon işlemi gerçekleşir. Denklem 92'deki işlemler modülerlik fonksiyonu dikkate alınarak gösterilmiştir. Modülerliğin amacı bu skorun maksimize edilmesidir. Eğer seçilecek uygunluk fonksiyonu için amaç minimize etmek ise Denklem 92'deki ilk koşulun ikinci kısmı $(F^k \leq F^o)$ şeklinde değiştirilir. Aynı şekilde ikinci koşulun ikinci kısmı da $(F^k > F^o)$ olarak değiştirilir. Buna örnek olarak *İletkenlik* (C) fonksiyonu verilebilir.

3pHybrid algoritmasına göre çoklu mutasyon işleminde $(MP - mutasyon)$ değiştirilecek toplam düğüm sayısı — C, Denklem 93'teki hesaplama ile belirlenir.

$$\mathrm{C} = R\ \times r^{[N/2]} \tag{93}$$

Burada, ilk önce rastsal olarak "1" ile "$N/2$" arasında bir tam sayı üretilir. Bu işlem denklemde $r^{[N/2]}$ ile temsil edilir. Bu işlemden sonra 0 ile 1 arasında yine rastsal

olarak oluşturulan R sayısı, ilk oluşturulan sayı ile çarpılır. Son olarak, bu sayı kesirli ise bir üst tam sayıya yuvarlanır. Daha sonra ilgili birey için C sayısı kadar rastsal olarak seçilen düğümler için farklı komşular seçilerek mutasyonlar gerçekleşir. Denklem 93'teki toplam düğüm sayısı N parametresi ile gösterildiğinde, çok düşük bir olasılıkla değiştirilebilecek maksimum düğüm sayısı $N/2$'dir. Buradaki "$N/2$" sınırıyla bireyin temel düğüm yapısı korunmaktadır.

Örnek olarak Şekil 3.22'de gösterilen yeni üye için iki farklı mutasyon türü işlemi şu şekilde gerçekleşir. "$SP-mutasyon$" operatörüne göre yeni üyedeki dördüncü düğümün komşusu olan 6. düğüm komşuluk listesine uygun olması şartıyla rastsal olarak 3. düğüm ile değiştirilir. "$MP-mutasyon$" operatörüne göre ise ikinci, üçüncü, beşinci ve yedinci düğümlerin komşuları olan 5., 7., 1. ve 2. düğümler 8., 4., 2. ve 1. düğümlerle değiştirilir. Böylece, üretilen yeni birey daha uygun bir çözüm sunar ve kendi paralel alt-grubu vasıtasıyla ana popülasyona taşınır. Bireylerin geliştirilmesi işlemlerinden sonraki karıştırma—*shuffling* aşamasında, mempleklerdeki bireyler ana popülasyonda birleştirilirler. Daha sonra tüm bireyler amaç fonksiyonu skorlarına göre iyiden kötüye doğru sıralanırlar. Bu işlemler toplam iterasyon sayısına ulaşılıncaya dek tekrarlanır ve böylece algoritmada gerçekleşen tüm süreçler tamamlanmış olur.

3pHybrid algoritması ile *Grup-A* kategorisinde testler yapıldığında, tüm işlem adımları sonunda sadece *Grup-A*'daki ağlar için modülerlik fonksiyonuna göre popülasyondaki global en iyiyi ifade eden birey ile en yüksek modülerlik skoru çıktı olarak verilir. Bu işlem adımları önceki algoritmaların çalışma prensibi ile aynı olur. Çünkü diğer algoritmalar sadece bu kategoride çalıştırılmıştır. Bu kategorideki sonuç aşamasında ulaşılan aday çözüm giriş ağının en uygun modül yapısını sunar. Önceki algoritmalardan farklı olarak *3pHybrid* algoritması ile *Grup-B* kategorisindeki ağlar üzerinde testler gerçekleştirilmiştir. Buradaki problem Bölüm 2.1.4'te açıklanmıştır. Gerekli bilgiler için ilgili bölüm incelenebilir. Bu kategoriyi temsil eden *Grup-B*'deki ağlarla yapılan testler sonunda ilk olarak 10 farklı amaç fonksiyonunun her birinin önerdiği ağ modül yapıları elde edilir. Daha sonra her bir fonksiyona göre elde edilmiş ağ modül yapıları giriş ağının referans modül yapıları ile kıyaslanır. Bu işlem sonunda bilinen en uygun sonuca ulaşan ya da buna yakınlaşan amaç fonksiyonu en başarılı sonuçları veren kalite fonksiyonu olarak seçilir ve çıktı olarak sunulur. Bu algoritmanın genel işlem adımlarını özetleyen *sözde-kod* Çizelge 3.12'de sunulmuştur.

Çizelge 3.12. *3pHybrid* algoritmasının çalışma prensibini özetleyen sözde-kod

1. 3pHybrid algoritmasını ağ modül tespiti problemine uyarla
2. Algoritmanın başlangıç parametrelerini tanımla
3. "$mekanizma-1$" ve "$mekanizma-2$" ile birey üretim işlemlerini başlat
4. 3pHybrid algoritmasının ana popülasyonunu oluştur → $(Şekil\ 3.19-a)$
5. "$Grup-A$" kategorisi için tek bir amaç fonksiyonunu ayarla → F_M *(modülerlik)*
6. "$Grup-B$" kategorisi için Bölüm 2.1.3'deki 10 adet amaç fonksiyonunu ayarla
↳ F_M, F_C, F_{ID}, F_{GD}, F_{CR}, F_{NC}, F_{FS}, F_{ODF}, F_S, F_{Sr}
7. Önceki adımdaki fonksiyonların kalite testleri için Bölüm 2.1.4'deki fonksiyonları tanımla
↳ $\mathrm{E_{MI}}$, "E_{NMI}", "E_{RI}", "E_{ARI}", "E_{JI}", "E_P"
8. Ana popülasyondaki her bir bireyin uygunluk değerlerini hesapla
9. Bireyleri uygunluklarına göre sırala
10. İlk iterasyon için global en iyi bireyi ya da bireyleri belirle
11. Ana popülasyonu, Şekil 3.15'e uygun olarak m sayıda paralel alt gruba (mempleks) ayır
12. Her bir alt gruptaki bireyler için yerel iyileştirme sürecini tekrarla → $(Şekil\ 3.19-b)$
13. Her alt gruptaki aday çözümler için 3 farklı duruma göre çaprazlama işlemini uygula
14. Üretilen aday çözümler için Denklem 91, 92 ve 93'e göre mutasyon işlemini yap
15. Tüm paralel alt grupları "karıştırma—shuffling" işlemi ile ana popülasyonda birleştir
16. Popülasyondaki yeni aday çözümleri uygunluklarına göre sırala
17. Yerel ve global en iyi aday çözümleri (bireyleri) kaydet
18. Algoritmanın önceki adımlarını sonlandırma kriteri sağlanana kadar tekrarla
19. Tüm iterasyonlar tamamlanınca iki farklı duruma göre sonuçları çıktı olarak ver
↳ $1.durum$: *Grup A'ya göre en yüksek modülerlik değerine sahip bireyi ve skoru göster*
↳ $2.durum$: *Grup B'ye göre en kaliteli sonuçları veren amaç fonksiyonunu göster*

3.5. Kanserle İlişkili CNA'lı Gen Gruplarının Tespitinde Önerilen Yaklaşımlar

Bu tez çalışmasında ele alınan bir diğer problem, klinik kanser verilerinden ve ilgili etkileşim ağlarından yararlanılarak kanserde sağkalım ile maksimum ilişkili *CNA*'lı gen gruplarının tespit edilmesidir. Literatüre sunulan doktora tez çalışmasındaki bu problem, (Hansen ve Vandin 2016) referanslı çalışmadan yararlanılarak önerilmiş ve Bölüm 2.2'de ayrıntılarıyla açıklanmıştır. Bu bölümde ise ilgili probleme tam ya da yaklaşık çözümler öneren yeni algoritmalar açıklanmıştır. Şekil 2.6'da bu problemin giriş verileri ile bu verilere göre elde edilen $k = 7$ adet genden oluşan bir alt-ağ gösterilmiştir. Elde edilen bu alt-ağdaki genler giriş verisindeki kanser türü için etkili olabilecek genleri içermektedir. Genlerin tespiti amacıyla sunulan algoritmalar kendi çalışma prensiplerine uygun bir biçimde farklı çözümler sunmaktadır. Bu algoritmalarla gerçekleştirilen testler, elde edilen p_{skor} sonuçları ve algoritmaların çalışma süreleri dikkate alınarak Bölüm 4.2'de değerlendirilmiştir. Aşağıdaki başlıklarda, Bölüm 2.2'de sunulan problemin yapısına uyarlanan dinamik programlama ve genetik algoritma temelli iki yaklaşım anlatılmıştır.

3.5.1. Dinamik Programlama Temelli Yaklaşım

Hansen ve *Vandin* tarafından *2016* yılında yayımlanan çalışmada (Hansen ve Vandin 2016) tanıtılan problemin *CNA*'lı genlere uyarlanmış hali yeni bir problem olarak bu tez çalışmasında sunulmuştur. Bu problemin çözümü amacıyla önerilen *DP* temelli yaklaşım, tanımlanan k sayısına ve giriş ağına bağlı olarak *Alon* ve diğerleri tarafından önerilen renk-kodlama (*color-coding*) (Alon ve ark 1994) yöntemi yardımıyla dinamik bir tablonun adım adım doldurulmasını içerir. Bu bölümde anlatılan algoritma sezgisel bir yaklaşım sunmakta ve DPT (Dinamik Programlama Tekniği) ile temsil edilmektedir. Algoritmanın uygulanmasında *SNAP* (*Stanford Network Analysis Platform*) kütüphanesinde yararlanılmıştır (SNAP 2016).

Bu bölümdeki yöntem giriş verileri olarak birbirleriyle ilişkili iki farklı bilgi kullanır. Bunlardan ilki hasta bilgileridir $(Giriş\ verisi\colon 1)$. Bu bilgiler, her bir hastaya özgü olarak tutulan *gen-CNA bilgileri*, *sağkalım süresi*$-(t)$ ve *hastanın sansürlü bilgi içerip içermediğini gösteren parametredir*$-(c)$. İkinci bilgi $(Giriş\ verisi\colon 2)$ ise ilgili hastaların sahip oldukları genlerin birbirleriyle bağlantılarını gösteren ağ yapısıdır. Tez çalışmasında Bölüm 2.2'deki problemle ilgili etkileşim ağları *Cytoscape* biyoinformatik yazılım platformu (Shannon ve ark 2003, Cytoscape 2017) yardımıyla *STRING* veri tabanından temin edilmiştir. Bu bölümdeki *DP* temelli yaklaşımın çalışma prensibi Şekil 3.23'te sunulan dokuz hastanın klinik bilgileri referans alınarak aşağıda adım adım anlatılmıştır.

Şekil 3.23'te *H1* ile *H9* arasında isimlendirilen örnek hasta bilgileri verilmiştir. Burada her bir hastanın genlerindeki *CNA* bilgileri bir matris şeklinde sunulmuştur. Eğer ilgili gen, *CNA*'lı ise kırmızı renkte; değilse beyaz renkte gösterilmiştir. Böylece *CNA*'lı durum 1 ile; *CNA*'sız durum 0 ile temsil edilmiştir. Ayrıca her bir hasta, ilgili kanser türü için sağkalım süresi ve sansür bilgisi içerir. Burada hastalar sağkalım sürelerine göre düşükten yükseğe doğru sıralanmıştır. Amaç ise gen-*CNA* bilgisi dikkate alınarak sansürlü ve daha az sağkalım süresine sahip hastaların ilk popülasyonu (P_0); sansürsüz ve yüksek sağkalım süresine sahip hastaların ise ikinci popülasyonu (P_1) oluşturmalarını sağlamaktır. Bu amaçla örnekte verilen hastaların Şekil 3.24'teki genler arası ilişkilerine göre iki popülasyondan birine dahil edilmesi sağlanır. En yüksek p_{skor} değerini veren alt-ağ ise hastaların en uygun şekilde P_0 ya da P_1 popülasyonlarına dahil

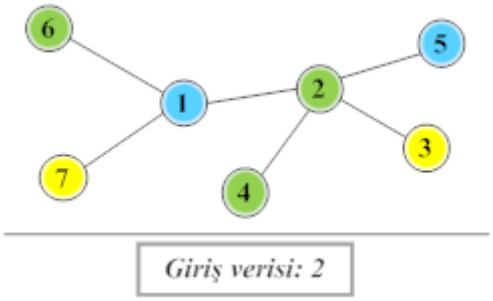
edilmesini sağlar. Böylece tespit edilen son gen grubu (Şekil 3.25'te gösterilen 3 boyutlu alt-ağ) ilgili kanser hastalığı için *DPT*'ye göre sağkalım parametresi ile maksimum ilişkili grubu temsil eder. Böylece bu kanser türünde en fazla etkili olduğu düşünülen gen etkileşimlerinin hastalardaki sağkalım ve *CNA* bilgileriyle birlikte değerlendirilmesinin önemi vurgulanır.

DPT ile işlem yapılmadan önce Şekil 3.23'teki hastaların gen ilişkilerini içeren ve Şekil 3.24'te verilen 7 adet düğüme (*gene*) sahip ağ bilgisi alınır ve komşuluk listeleri olarak belleğe yüklenir. Hastaların hangi popülasyona dahil edileceği bu ağdaki gen ilişkilerine göre belirlenir. Bu algoritma ağdaki düğümlere k adet rengin rastsal olarak atanmasıyla başlar. Burada k, ilgili kanser türünde sağkalım parametresi ile maksimum ilişkide olduğu düşünülen toplam gen sayısıdır. Bu parametre aynı zamanda rastsal olarak üretilecek renk türünün sayısını da ifade eder. Örneğin Şekil 3.24'teki giriş ağı için bu değer 3 olarak belirlenmiştir. Bunlar mavi, yeşil ve sarı renklerdir. Buna göre üç adet ilişkili gen içeren bağlı bir alt-ağ ortaya çıkarılacaktır. Şekil 3.24'te verilen ağdaki genlere 3 farklı renk atandıktan sonra k sayısına göre dinamik tablo oluşturulur.

Gen/CNA bilgisi								t	c
H1:	1. gen	2. gen	**3. gen**	4. gen	**5. gen**	6. gen	7. gen	2	0
H2:	1. gen	**2. gen**	3. gen	4. gen	5. gen	6. gen	7. gen	4	0
H3:	1. gen	2. gen	3. gen	**4. gen**	5. gen	6. gen	7. gen	13	0
H4:	**1. gen**	**2. gen**	3. gen	4. gen	5. gen	6. gen	**7. gen**	48	1
H5:	1. gen	2. gen	3. gen	4. gen	**5. gen**	6. gen	7. gen	50	0
H6:	1. gen	2. gen	3. gen	**4. gen**	5. gen	**6. gen**	7. gen	71	1
H7:	1. gen	2. gen	**3. gen**	4. gen	5. gen	6. gen	**7. gen**	108	1
H8:	**1. gen**	2. gen	3. gen	4. gen	5. gen	**6. gen**	**7. gen**	120	1
H9:	1. gen	2. gen	3. gen	4. gen	**5. gen**	**6. gen**	7. gen	185	1

Giriş verisi: 1

Şekil 3.23. Temsili hastaların sağkalım süreleri ve sansür bilgileri ile gen-*CNA* matrisi



Şekil 3.24. Birinci giriş verisindeki (*Giriş verisi*: 1) genlere göre oluşturulan temsili ağ

Yukarıdaki şekillerde gösterilen temsil örnek için oluşturulan dinamik bir tablo Çizelge 3.13'te verilmiştir. Bu tablo $[d \times n]$ boyutludur. Burada d, toplam renk kombinasyonu sayısını; n, toplam gen sayısını temsil eder. Denklem 94'e göre d parametresi belirlenir.

$$d = 2^k - 1 \tag{94}$$

Buradaki örnekte $k = 3$ için $d = 7$ sonucu elde edilir. Çizelge 3.13'teki n parametresinin değeri ise 7'dir. Bu sayı Şekil 3.23'teki en son düğümün (gen) sırasını, yani maksimum gen sayısını ifade eder. Tablodaki genleri temsil eden düğümler, 1'den 7'ye kadar numaralandırılmıştır. Renklerin gösterimi için ise mavi, yeşil ve sarı renkler sırasıyla; M, Y ve S ile temsil edilmiştir. Tüm bu bilgilere göre ağdaki genlerin birbirleriyle etkileşimlerine uygun olarak doldurulan $7X7$'lik bir matris elde edilmiştir.

Çizelge 3.13. Üç adet $(k = 3)$ renge göre oluşturulan dinamik tablo

	1. adım			2. adım			3. adım
	M	**Y**	**S**	**M-Y**	**M-S**	**Y-S**	**M-Y-S**
1	p^1			$p^{[1-2]}\_p^{[1-6]}$	$p^{[1-7]}$		$p^{[1-2-3]}\_p^{[1-6-7]}$
2		p^2		$p^{[1-2]}\_p^{[5-2]}$		$p^{[2-3]}$	$p^{[1-2-3]}\_p^{[5-2-3]}$
3			p^3			$p^{[2-3]}$	$p^{[1-2-3]}\_p^{[5-2-3]}$
4		p^4					
5	p^5			$p^{[5-2]}$			$p^{[5-2-3]}$
6		p^6		$p^{[1-6]}$			$p^{[1-6-7]}$
7			p^7		$p^{[1-7]}$		$p^{[1-6-7]}$

Bu tablonun doldurulması şu şekilde gerçekleşir: Her bir genin hastalardaki *CNA* durumlarına göre hesaplanan *normalize edilmiş log-rank istatistik değeri—*p^i, kendi renginin bulunduğu hücreye yazılır. Burada i, genin sıra numarasını ve p^i, i. gene göre hesaplanan skoru gösterir. Tablodaki 1'li, 2'li ve 3'lü renklerin sütün başlıklarının arka planları sırasıyla; açık tonlarda sarı, yeşil ve gri renklerle gösterilmiştir. Bu renklere göre işlemler; *M*, *Y*, *S* sütunları için 1. adımda; *M-Y*, *M-S*, *Y-S* sütunları için 2. adımda ve son olarak *M-Y-S* sütunu için 3. adımda gerçekleşir. Böylece farklı renkler farklı adımlardaki işlemleri göstermiş olur. Örneğin ilk adımda; birinci gen, mavi renge sahip olduğu için *M* sütunu ile 1. satırın kesişimine "p^1" şeklinde; ikinci gen, yeşil renkli olduğu için *Y* sütunu ile 2. satırın kesişimine "p^2" şeklinde yazılır. Bu şekilde tüm tekli genlere göre elde edilen skorlar 1. adımdaki uygun hücrelere yazılır. İkinci adımda; ikili gen kombinasyonları alınır. Bu işlem ve sonraki işlemler için hem genlerin birbirleriyle doğrudan komşuluklarının bulunması hem de renk kombinasyonlarına uygun olmaları temel şarttır. Yani her renge uygun bir genin bulunması gereklidir. Bu işlem ikinci adım için ikili renk şartına göre gerçekleşir. Örneğin; arka plan rengi açık yeşil olan 2. adımda *M-Y* sütunu, mavi ve yeşil renkli komşu düğümlere göre doldurulacaktır. Şekil 3.24'te bu şarta uyan üç adet ikili komşu genler bulunmaktadır. Bunlar 1-2, 1-6 ve 5-2'dir. Buna göre *M-Y* sütunu için 1. satır ile *M-Y* sütunun kesişimine $p^{[1-2]}$ ve $p^{[1-6]}$ skorları; 2. satır ile *M-Y* sütunun kesişimine $p^{[1-2]}$ ve $p^{[5-2]}$ skorları; 5. satır ile *M-Y* sütunun kesişimine $p^{[5-2]}$ skoru ve 6. satır ile *M-Y* sütunun kesişimine ise $p^{[1-6]}$ skoru ayrı ayrı yazılır. Bu adımdaki *M-S* ve *Y-S* sütunlarındaki şartlara uyan hücreler de aynı şekilde doldurulur ve böylece ikinci adım da tamamlanır. Bu örnek için son işlemleri ifade eden 3. adımda (arka plan rengi açık gri olan adım) ise 3'lü renk kombinasyonunu ifade eden *M-Y-S* renklerine sahip genler kontrol edilir. Bu şartlara uygun olan hücreler doldurulur. Tablodaki 4. genin hizasındaki son hücre hariç diğer tüm son hücreler uygun skorlarla doldurulur. Her bir satırda en yüksek skoru veren gene ya da genlere sahip hücrelerdeki skorlar kırmızı renklerle vurgulanmıştır. Temsili örnekte bu skorların diğerlerine göre daha yüksek oldukları varsayılmıştır.

Çizelge 3.13'e dikkat edilirse 1. satırın 1. hücresindeki p^1, 4. hücresindeki $p^{[1-6]}$ ve 7. hücredeki $p^{[1-6-7]}$ ifadeleri kırmızı renkle yazılmıştır. Burada ilk adım için 1. geni *CNA*'lı olmayan hastalar P_0 popülasyonuna; bu geni *CNA*'lı olan hastalar ise P_1 popülasyonuna dahil edilir. Bu örnek için ilk adımda Şekil 3.23'teki hasta bilgilerine göre 4. (*H-4*) ve 8. (*H-8*) hastaların ilk genleri *CNA*'lı olduğu için P_1 popülasyonuna;

diğerleri P_0 popülasyonuna eklenir. Bu işlem için Denklem 95'e bakılabilir. Daha sonra güncellenen P_0 ve P_1 popülasyonlarındaki hasta bilgilerine göre Bölüm 2.2'deki Denklem 47'de gösterilen formül ile p_{skor} değerleri hesaplanır. Bu skor, temsil örnekte p^1'de tutulur. Bu adımdaki p^1 skorunun diğerlerinden yüksek olduğu varsayılmıştır.

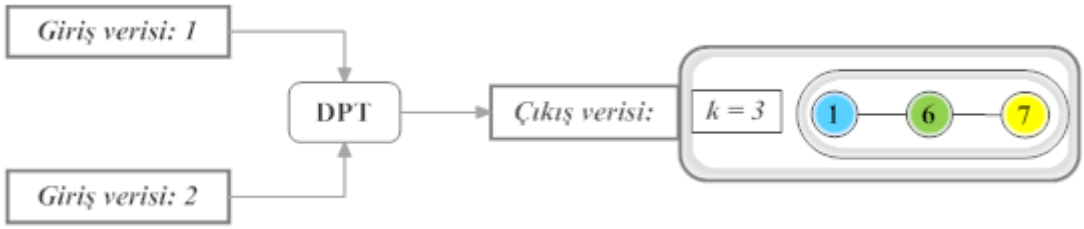
İkinci adımda yine ilk adımdaki gibi her bir uygun hücredeki gen bağlantılarına ve *CNA* bilgilerine göre hastalar iki popülasyondan birine aktarılır ve p_{skor} hesaplaması yapılır. Bu adımda da önceki adımdaki gibi kırmızı renkli skorun ilgili sütundaki en yüksek skor olduğu farz edilmiştir. Bu yüzden örnek olarak, $p^{[1-6]}$ skorunun hesaplanması şöyle gerçekleşir: İlk önce tüm popülasyondaki hastaların 1. ve 6. genlerinin *CNA*'lı olma durumları kontrol edilir. 1. ve 6. genlerden en az birinin *CNA*'lı olması durumunda ise ilgili hasta P_1 popülasyonuna; aksi durumda P_0 popülasyonuna dahil edilir. Böylece Şekil 3.23'e göre 4. hastanın 1., 2. ve 7. genlerinin; 6. hastanın 4. ve 6. genlerinin; 8. hastanın 1., 6. ve 7. genlerinin; 9. hastanın 5. ve 6. genlerinin *CNA*'lı olduğu bilinmektedir. Bu durumda 4., 6., 8. ve 9. hastalar P_1 popülasyonuna; diğerleri P_0 popülasyonuna eklenir. Bu durumda hesaplanan skor, $p^{[1-6]}$'da saklanır. Bu adımda tüm uygun hücreler için benzer hesaplamalar yapılır ve en yüksek skor saklanır.

Son adımda ise aynı işlem, tüm 3'lü genlere sahip hücreler için yapılır ve en yüksek skor kaydedilir. Yukarıdaki örnekte kırmızı renkle vurgulanan $p^{[1-6-7]}$ skorunun diğerlerinden daha yüksek olduğu varsayılmıştır. Örnek olarak bu genlere göre yapılan hesaplama şöyle olur: İlk önce 1. 6. ve 7. genlerden en az birinin *CNA*'lı olması durumunda uygun hastalar P_1'e; diğerleri P_0'a aktarılır. Daha sonra elde edilen 2 popülasyona göre en yüksek değere sahip olduğu varsayılan $p^{[1-6-7]}$ skoru hesaplanır. Bu skor aynı zamanda temsili örnek için algoritmanın elde ettiği son değeri temsil eder.

Yukarıdaki üç farklı adımda gerçekleşen işlemlere göre güncellenen alt-popülasyonlara uygun hastalar Denklem 95'te gösterildiği gibi dahil edilirler. Bu sonuçlar Şekil 3.23'teki hasta bilgilerine göre dikkatlice incelendiğinde, ilerleyen her bir adımda genellikle *DPT* ile yüksek sağkalım sürelerine ve sansürsüz bilgilere sahip hastaların diğerlerinden ayrılıp P_1'e dahil edildiği anlaşılabilir. İkinci adım sonunda bir önceki adımda P_0'da bulunan 6. ve 9. hastaların P_1'e aktarılması ve üçüncü adımdaki işlem ile önceki adımlarda P_0'a ait olan 7. hastanın P_1 popülasyonuna dahil edilmesi yukarıdaki açıklamayı doğrulamaktadır.

$$\begin{aligned} &1.\text{adım} \rightarrow \left\{\begin{matrix} P_0 = [H1; H2; H3; H5; H6; H7; H9] \\ P_1 = [H4; H8] \end{matrix}\right\} \\ &2.\text{adım} \rightarrow \left\{\begin{matrix} P_0 = [H1; H2; H3; H5; H7] \\ P_1 = [H4; H6; H8; H9] \end{matrix}\right\} \\ &3.\text{adım} \rightarrow \left\{\begin{matrix} P_0 = [H1; H2; H3; H5] \\ P_1 = [H4; H6; H7; H8; H9] \end{matrix}\right\} \end{aligned} \quad (95)$$

Üçüncü adım sonunda en uygun alt-ağın 1., 6. ve 7. genlerden oluştuğu görülür. Şekil 3.23'teki hasta bilgileri de incelendiğinde, genellikle sansürlü ve daha az sağkalım süresine sahip hastaların ilk popülasyonu (P_0); sansürsüz ve yüksek sağkalım süresine sahip hastaların ise ikinci popülasyonu (P_1) oluşturduğu anlaşılmıştır. Böylece hangi bağlı genlerin kanserdeki yaşam süresinin eşik sınırını en uygun şekilde belirlediği anlaşılır. Bu eşik sınır, P_0'da sansürlü ve en kısa sağkalım süresine sahip hastalar ile P_1'de sansürsüz ve en yüksek sağkalım süresine sahip hastalar arasındaki belirgin farkı temsil eder. Burada anlatılan örnek için 1., 6. ve 7. genlerin ilgili hastalıkta sağkalım parametresiyle en fazla ilişkili genler oldukları varsayılır. Son olarak, burada açıklanan tüm işlem adımları sonunda ortaya çıkarılan $k = 3$ boyutlu bir alt-ağ için temsili grafik Şekil 3.25'te gösterilmiştir.



Şekil 3.25. *DPT* algoritması ile uygun alt-ağın ortaya çıkarılması

Son olarak *DP* temelli algoritmanın k sayısına göre belirlenen uygun maksimum çalıştırılma sayısı T_m ile temsil edilmektedir. Bu sayı Denklem 96 ile Denklem 100 arasındaki hesaplamalara göre belirlenir.

$$p = \frac{k!}{k^k} \quad (96)$$

$$q = 1 - p \quad \rightarrow \quad t \ için, \quad q^t \quad (97)$$

$$q^t \leq \varepsilon \quad (98)$$

$$t \times \log q \leq \log \varepsilon \quad \rightarrow \quad t \leq \frac{\log \varepsilon}{\log q} \tag{99}$$

$$T_m = M(t) \tag{100}$$

Denklem 96'daki p parametresi en başarılı sonucun renkli kümelerde bulunma olasılığını gösterir ve bu parametre k sayısına göre belirlenir. $k!$ ve k^k sayıları ise Çizelge 3.14'teki olasılık tablosuna göre belirlenir. Buradaki $k!$, farklı renk kombinsyonlarının sayısını; k^k ise tüm olasılıkları içeren toplam sayıyı ifade eder. $C^{k!}$ kümesi farklı renklerdeki elemanları içerir. Buna göre Çizelge 3.14'teki k sayısı 3 olarak alınırsa, farklı renklerden oluşan tüm eleman sayısı $k! = 3! = 6$ olarak bulunur. Böylece $C^{k!=6} = \{(1,2,3), (1,3,2), (2,1,3), (2,3,1), (3,1,2), (3,2,1)\}$ kümesi oluşturulur. Bu kümedeki tüm elemanlar farklı sayılarla oluşturulmuştur. Renk kodlama tekniği ile işlem yapılabilmesi için k adet rengin tümünün elemanlara dağıtılması gerekir. $k!$ kümesindeki dışındaki diğer elemanlar bu şartı sağlamamaktadır. *DPT*, önerilen renk-kodlama tekniği ile genlere k adet rengin rastsal olarak atanması prensibine göre çalıştığı için başarılı sonucun bulunma olasılığı p ile hesaplanır. p parametresinin değerine göre Denklem 97'de başarısız olma (*hata*) olasılığı hesaplanır ve sonuç, q parametresinde tutulur. Denklemde t adet başarısızlık için hata olasılığı q^t ile gösterilir. Denklem 98'deki ε, düşük hata olasılığını belirleyen sayıdır ve bu sayı Denklem 99'a göre t'nin en alt sınırını belirler. Son olarak, maksimum t sayısına göre algoritmanın toplam çalıştırılma sayısını ifade eden T_m değeri hesaplanır. Bu sayı Denklem 100'deki "$M(t)$" ifadesi ile belirlenir. Buradaki ifade, t ondalıklı sayısının bir üst tam sayıya yuvarlanmasını temsil eder. Tüm bu hesaplamalar ile algoritmanın başarılı bir sonuca ulaşabilmesi için en az kaç kez çalıştırılması gerektiği tespit edilir.

Çizelge 3.14. $k = 3$ değerine göre üçlü olasılıkların oluşturulması

k_1	k_2	k_3	
1	2	3	$k!$
1	3	2	
2	1	3	
2	3	1	
3	1	2	
3	2	1	
1	1	1	k^k
1	1	2	
1	1	3	
.	.	.	
.	.	.	
.	.	.	
3	3	3	

Örneğin, $k = 3$ ve $\varepsilon = 0.001$ şeklinde alınırsa önerilen dinamik programlama temelli algoritmanın uygun çalıştırılma sayısı Denklem 100 yardımıyla $T_m = 27$ olarak bulunur. Buradaki hata olasılığı sadece binde bir (*0.001*) olarak belirlenmiştir. Böylece sağkalım ile ilişkili üç genli bir alt-ağın tespiti için Çizelge 3.14'te renk kodlama tekniği kullanılarak binde 99 başarı olasılığı ile işlem gerçekleşir. Böylece normalde $k^k = 3^3 = 27$ adet deneme yerine $k! = 3! = 6$ adet deneme ile uygun çözümlere yaklaşılmıştır.

3.5.2. Genetik Algoritma Temelli Yaklaşım

Bölüm 2.2'deki problemin çözümü amacıyla bu bölümde, *GA* (Goldberg ve Holland 1988, Holland 1992) temelli metasezgisel bir yaklaşım önerilmiştir. Bu yaklaşım tez çalışmasında *GAT* (Genetik Algoritma Tekniği) kısaltmasıyla ifade edilmektedir. Bu tekniğin uygulanmasında, çizgelerin temsili ve topolojik özelliklerin elde edilmesi gibi temel işlemler için *SNAP* kütüphanesindeki bazı fonksiyonlardan yararlanılmıştır (SNAP 2016). Bu algoritmanın temel işlem adımları; kromozomların temsili gösterimi, popülasyonun oluşturulması, uygunluk değerlerinin hesaplaması ve seçim süreci ile çaprazlama ve mutasyon operatörlerinden oluşur. Gerçekleştirilen her

bir işlem aşağıda adım adım açıklanmıştır. Algoritmanın sonlandırma kriteri olarak maksimum iterasyon sayısı temel alınmıştır. Bu sayı 1000'dir. Ayrıca, algoritmadaki toplam birey sayısı 50 olarak belirlenmiştir. *GAT* için seçilen uygun parametre değerleri literatürdeki öneriler de dikkate alınarak deneme yanılma yoluyla belirlenmiştir.

3.5.2.1. Kromozomların temsili gösterimi

Algoritmadaki aday çözümler, kromozomları temsil ederler ve bunların genetik temsili gösterimi Şekil 3.26'da verilmiştir. Kromozomdaki gen sayısı ağdaki düğüm sayısına denk gelir. Düğüm/gen sayısı ise temsili örnek için Şekil 3.24'teki ağın toplam düğüm sayısıdır ve bu değer d parametresinde tutulur. Buna göre temsil bireyin boyutu $d = 7$ olur. Her bir sıra, genin indisini (ID: sıra numarası) gösterir. Gen içeriği ise 0 veya 1'lerden oluşur. Buradaki 1 sayısı bağlantının olması durumunu; 0 ise bağlantısız durumu gösterir. Gen içerikleri ağdaki düğümlerin komşuluklarına göre rastsal olarak belirlenir. Şekil 3.26'daki örnekte 1., 6. ve 7. genlere rastsal olarak 1 sayısı atanmıştır. Diğer genlerin içerikleri 0 olarak belirlenmiştir. Buradaki örnek için k sayısı 3 olarak belirlenmiştir. Tanımlanan k parametresi giriş verisindeki kanser türünde sağkalım parametresi ile maksimum ilişkide olduğu düşünülen toplam gen sayısını gösterir. Bu sayı kullanıcı tarafından algoritmanın başlangıcında belirlenir.

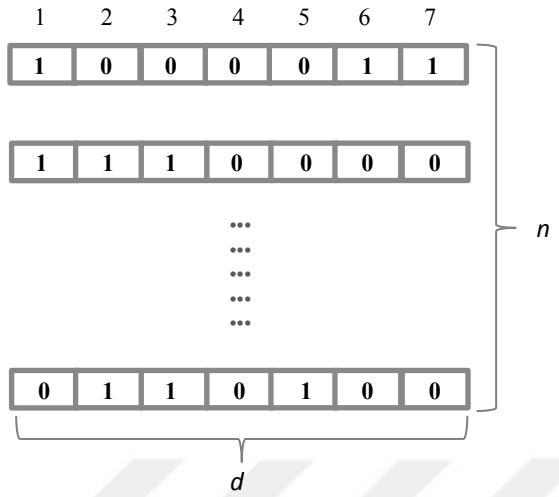
ID:	1	2	3	4	5	6	7
	1	**0**	**0**	**0**	**0**	**1**	**1**

Şekil 3.26. *GAT* algoritmasına göre temsili kromozomun gösterimi

3.5.2.2. Popülasyonun oluşturulması

Popülasyondaki bireylerin temsili gösteriminden sonra bu bireylerin içeriklerine k sayısı kadar rastsal olarak 0 ya da 1 sayıları atanır. Bu işlem için temel şart atanan 1 sayılarına denk gelen düğümlerin doğrudan veya dolaylı olarak komşu olmaları gerekir. Örneğin, Şekil 3.26'daki kromozom için seçilen düğümlerin (1 - 6 - 7) Şekil 3.24'teki ağda birbirlerine bağlı oldukları (komşuluklarının bulunduğu) bilinmektedir. Bu şekilde popülasyona d boyutlu n sayıda kromozom eklenir. n, popülasyondaki toplam aday çözüm (*kromozom*) sayısıdır. Her bir aday çözümde k adet komşuluk belirlenir. Böylece

k sayısı kadar 1 değeri kromozomlara rastsal olarak atanır. Şekil 3.24'teki ağda bulunan düğümlerin komşuluklarına göre oluşturulan popülasyon Şekil 3.27'de gösterilmiştir.



Şekil 3.27. Popülasyondaki temsili kromozomlar

3.5.2.3. Uygunluk değerlerinin hesaplaması ve seçim süreci

Önceki adımda üretilen bireylerin uygunluk değerleri Bölüm 2.2'de sunulan Denklem 47'deki p_{skor} hesaplama formülüne göre belirlenir. Uygunluk fonksiyonunun hesaplanabilmesi için gereken temsili giriş bilgileri ve sunulan deneysel çıktılar Bölüm 3.5.1'de açıklanmıştır. *GAT* algoritması ile amaç, yüksek p_{skor} değerlerine sahip aday çözümlerin elde edilmesidir. Böylece ulaşılan değerlerin yüksek olması bireylerin daha kaliteli olduğunu gösterir. Hesaplama işleminden sonra bireyler uygunluklarına göre yüksekten düşüğe doğru sıralanırlar. Ana popülasyondaki bireylerin en iyi "%β" tanesi P^{β} popülasyonunu (*en iyiler*) oluşturur ve bunlar sonraki nesile aktarılmak üzere kaydedilir. β değişkeni ana popülasyondaki en iyi bireylerin saklanması amacıyla kullanılır. Bu değişkenin değeri kullanıcı tarafından belirlenir. Kalan bireyler içinden çaprazlama olasılığı şartını sağlayanlar da çaprazlama sürecine alınırlar. Örneğin popülasyondaki birey sayısı 100 ise ve $\beta = 10$ olarak belirlenmişse en iyi bireylerin yüzde 10'u sonraki nesile aktarılır. Böylece $P^{\beta=10}$'daki 10 adet birey sonraki iterasyondaki ana popülasyona dahil edilir.

İki farklı bireyden yeni bir bireyin elde edilmesi sürecini temsil eden ve sonraki alt başlıkta anlatılan çaprazlama adımı yüksek uygunluk değerlerine sahip bireylerin seçilmesi aşamasından sonra gerçekleşir. Çaprazlamadaki ilk bireyler çaprazlama

olasılığı şartını (bu şart sonraki alt başlıkta anlatılmıştır) sağlayan bireylerin oluşturduğu P^C'den alınırken; ikinci bireyler doğal seçilimdeki rulet tekerleği yöntemi ile belirlenen P^{RTS} popülasyonundan alınır. Bu adım için Bölüm 3.4.6'daki seçim aşamasında kullanılan *RTS*'nin işlemleri uygulanır. Ana popülasyondaki her bir aday çözümün uygunluğuyla doğru orantılı olarak belirlenen kümülatif olasılık değerleri bu bireylerin seçim olasılıklarını belirler. Çaprazlama amacıyla seçilen bireyler genelde yüksek değerli p_{skor}'lara sahip aday çözümleri ifade eder. İkinci bireyler (*kromozomlar*) Bölüm 3.4.6'da açıklanan Denklem 88'e göre seçilir ve P^{RTS}'e dahil edilir. Bu popülasyondaki birey sayısı P^C popülasyonunki ile aynı olur.

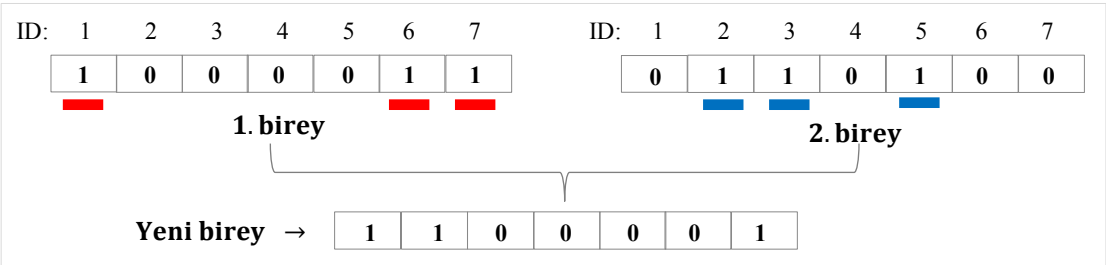
3.5.2.4. Çaprazlama operatörü

Bu adım aday çözümlerin geliştirilmesi amacıyla kullanılan ve 2 farklı bireyden (*ebeveyn kromozom*) yeni bir bireyin (*çocuk kromozom*) elde edilmesi sürecini temsil eden işlem adımıdır. Buradaki ilk birey (1. ebeveyn) popülasyondaki sırasına uygun olarak çaprazlama olasılığına şartına göre işleme alınan aday çözümdür. Bu bireyler P^C popülasyonunda tutulurlar. İkinci birey (2. ebeveyn) ise bir önceki işlem sürecinde (Bölüm 3.5.2.3'teki adımda) kümülatif olasılık değerine göre seçilen uygun çözümlü bireydir. İkinci bireyler P^{RTS}'den alınırlar. Çaprazlamadaki her ilk birey için işleme alınan ikinci birey, P^{RTS}'den rastsal olarak seçilir. Bu işlemler aşağıda açıklanmıştır.

Çaprazlama operatöründeki işlemler popülasyondaki toplam birey sayısının yaklaşık olarak "$C \times n$" tanesi için uygulanır ve seçilen bireyler P^C popülasyonunda tutulur. Bu tez çalışmasında C değeri 0.8 olarak alınmıştır. Örneğin $C = 0.8$ ve ana popülasyondaki toplam birey sayısını ifade eden n 100 ise her bir birey için rastsal olarak çaprazlama olasılığı şartını sağlayan bireylerin oluşturduğu $P^{C=0.8}$ popülasyonu yaklaşık olarak 80 bireyden oluşur. Çaprazlama olasılığı şartı ise $R \leq C$ ile temsil edilir. Buradaki R ifadesi 0 ile 1 arasında rastsal olarak üretilen sayıdır. Ana popülasyondaki her birey için bir R sayısı üretilir. Bu sayı çaprazlama olasılığından küçük veya eşitse bu birey P^C'e dahil edilir. Aksi durumda diğer bireye geçilir. Böylece şartı sağlayanlar çaprazlama operatöründeki birinci bireyleri temsil ederler. İkinci bireyler ise P^{RTS} popülasyondan rastsal olarak seçilirler. P^{RTS}'deki toplam aday çözüm sayısı n_{RTS} ile temsil edilsin. Çaprazlamadaki ikinci bireyin seçilmesi için öncelikle 1 ile n_{RTS} arasında rastsal olarak bir tam sayı üretilir. Bu sayı P^{RTS} popülasyonundaki herhangi bir bireyin

sıra numaralarına denk gelir ve bu sıradaki aday çözüm ikinci birey olarak alınır. İki farklı birey seçildikten sonra çaprazlama operatörüne geçilir.

Çaprazlama operatöründe ilk işlem, seçilen birinci ve ikinci bireylerdeki gen içerikleri "1" olan konumları bulmaktır. Bu konumlar, kromozomlardaki bağlı genlerin *ID*'lerine karşılık gelirler. Her bir birey, k adet "1" içerir. k parametresi ile ilgili tanımlama Bölüm 3.5.1'de yapılmıştır. Bulunan konumlardaki gen grupları ilgili kromozomlar için temsil edilen k boyutlu alt-ağları ifade ederler. *GAT* algoritmasındaki çaprazlama işlemini gösteren temsili örnek Şekil 3.28'de verilmiştir. Buradaki 2 birey de Şekil 3.24'teki temsili ağdaki düğümlerin komşuluklarına göre oluşturulmuştur. 1. bireyin P^C popülasyonundan sırasına göre işleme alındığı; 2. bireyin ise P^{RTS}'den rastsal olarak seçildiği varsayılmıştır. Aşağıdaki şekilde bireylerdeki alt-ağları oluşturan genler birinci birey için kırmızı; ikinci birey için mavi kalın alt-çizgilerle vurgulanmıştır. Buna göre birinci bireydeki 1., 6. ve 7. genlerden 1. ve 7. genler ile ikinci bireydeki 2., 3. ve 5. genlerden 2. gen çaprazlama işleminde rastsal olarak seçilerek yeni üretilen bireye aktarılmıştır. Sonuç olarak yeni birey 1., 2. ve 7. genlerden oluşan bağlı bir alt-ağı temsil eder. Bu temsili örnekte de gösterildiği gibi üretilecek yeni bireyler için "1" olarak belirlenecek konumlar (seçilecek gen ID'leri), rastsal olarak seçilen iki bireyden birindeki "1" konumlarından (var olan gen ID'lerinden) alınırlar. Böylece k adet gen çaprazlama prensibine göre rastsal olarak seçilir.



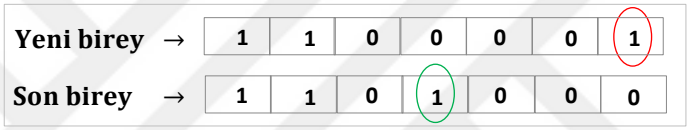
Şekil 3.28. Önerilen çaprazlama operatörü

Şekil 3.28'de gösterilen örnekteki gibi P^C popülasyonundaki bireylere karşılık P^{RTS} popülasyonundan uygun bireylerin seçilmesiyle çaprazlama işlemi, n_{RTS} sayısı kadar tekrarlanır. Burada dikkat edilmesi gereken şart ise oluşturulan bireylerdeki genlerin birbirleriyle komşuluklarının bulunması zorunluluğudur. Bu yüzden her bir alt-ağın bağlı yapıda olması durumu aranır. Buna göre üretilen tüm yeni bireyler P^{yeni} isimli popülasyonu oluşturur ve sonraki adımdaki mutasyon sürecine aktarılırlar.

3.5.2.5. Mutasyon operatörü

Bu adımda P^{yeni}'deki bireyler için mutasyon işlemi gerçekleştirilir. Bunun için mutasyon operatörüne alınacak bireyler $r_M < M$ şartına göre belirlenir. M, mutasyon oranını; r_M, 0 ile 1 arasında üretilen sayıyı gösterir. Buradaki r_M sayısı her birey için tekrar üretilir. M değeri ise 0.05 gibi düşük bir oranda seçilmiştir. Böylece sadece şartı sağlayan bireyler mutasyona uğratılır. Diğerleri ise mutasyon operatörüne alınmadan sonraki adıma geçerler. Üretilecek rastsal sayılara ve olasılık değerlerine bağlı olarak P^{yeni}'deki bireylerin yaklaşık olarak sadece yüzde 5'i mutasyona uğrar.

Mutasyon işlemi seçilen kromozomda bulunan ve sadece tek bir bağlantısı olan genlerden yalnızca birinde gerçekleşir. Bu işlem Şekil 3.29'a benzer şekilde yapılır.



Şekil 3.29. Tek bir gen için gerçekleşen mutasyon işlemi

Şekil 3.29'a göre bir önceki çaprazlama sürecinde gösterilen yeni birey burada temsili olarak mutasyona tabi tutulmuştur. Burada işleme alınan genin tek bir bağlantısının olması şartı göz önüne alınmıştır. Çünkü sadece bir bağlantısı olan genin değiştirilmesi durumunda bireyin genel yapısı mutasyon doğasına uygun olarak korunur ve sadece küçük değişimler gerçekleşir. Şekil 3.29'daki yeni bireydeki (*çaprazlamadan gelen*) son gen kırmızı halka ile belirtilmiştir. Son bireydeki 4. gen ise yeşil halka ile vurgulanmıştır. Bunların anlamı ilk durumda alt-ağdaki üçüncü düğüm 7. gen ile temsil edilirken; tekli mutasyonla bu gen 4. gen ile değiştirilmiştir. Son aşamadaki alt-ağ 1., 2. ve 4. genlerden oluşur. Yukarıdaki şekilde gösterilen iki birey de Şekil 3.24'teki temsili ağdaki komşuluklara uygun olarak oluşturulmuştur. Mutasyon işlemi öncesi mevcut bireyden (*yeni birey*) rastsal olarak seçilen 7. genin tek bir komşusunun olduğu ve bunun 4. gen ile değiştirilebileceği Şekil 3.24'ten anlaşılabilir. Son bireyde bulunan 4. genin de tek bir komşuluğunun bulunduğu bilinmektedir. Buradaki tüm mutasyon işlemleri P^{yeni}'de bulunan ve $r_M < M$ şartını sağlayan bireylere aynı şekilde uygulanır.

GAT algoritmasının en son aşamasında ise uygun bireyleri içeren alt-ağlardaki genlerin *CNA* durumlarına göre Bölüm 3.5.1'deki temsili örnekteki gibi hem P_0 hem de

P_1 hasta popülasyonları belirlenir ve p_{skor} hesabı yapılır. Daha sonra işlem yapılan nesilde en uygun çözümü sunan birey, yerel en iyi birey olarak kaydedilir. Aynı şekilde global en iyi çözümü sunan birey de yerel en çözüme göre güncellenir. Maksimum iterasyon sayısına ulaşılıncaya kadar yukarıdaki tüm işlemler tekrarlanır. Bitirme şartı sağlandığında ise önerilen algoritma sonlandırılır. Böylece giriş verisindeki kanser türü için bu algoritmanın başarısına göre sağkalım parametresiyle maksimum ilişkili olduğu düşünülen k boyutlu bir alt-ağ elde edilmiş olur.

4. DENEYSEL ÇALIŞMALAR

Bu bölümde tez çalışmasında ele alınan problemlerle ilgili deneysel çalışmaların sonuçları ve bu sonuçların ayrıntılı değerlendirmeleri sunulmuştur. Sonuçlar genellikle grafiklerde ve tablolarda gösterilmiştir. Deneysel çalışmalar iki temel başlık altında verilmiştir. Bunlardan ilki ağ modüllerinin tespiti konusunda yapılan karşılaştırmalı testlerdir. Diğeri ise kanser hastalıkları ile ilişkili *CNA*'lı gen gruplarının tespit edilmesi konusundaki çalışmalardır. Bunlar sırasıyla; Bölüm 4.1 ve Bölüm 4.2'de açıklanmıştır. Ayrıca her bir bölüm de kendi alt başlıklarına ayrılmıştır.

4.1. Ağ Modüllerinin Bulunmasıyla İlgili Deneysel Çalışmalar

Bu bölümde Bölüm 2.1'deki problemin çözümü amacıyla sunulan algoritmaların test çalışmaları verilmiştir. Bu algoritmaların genel çalışma prensipleri ile giriş ve çıkış verileri Bölüm 3.4'te ayrıntılı olarak anlatılmıştır. Burada ele alınan problemin test çalışmalarında kullanılan algoritmalar *Matlab* ile kodlanmıştır. Deneyler, *~8 GB* kapasiteli bellek, *Intel(R) Xeon(R) CPU* (*12 CPUs*) *E5-1650 ~3.20GHz* özellikli işlemci ve *Windows 10 Pro 64-bit* işletim sistemine sahip bir bilgisayarda gerçekleştirilmiştir. Önerilen algoritmalar tüm test ağları için 30 kez çalıştırılmıştır. Her bir algoritmaya ait parametrelerin değerleri literatürdeki öneriler de dikkate alınarak deneme ve yanılma yoluyla belirlenmiştir. Bu yüzden en uygun çözümlere yaklaştıran değerler ya da değer aralıkları ilgili parametreler için atanmıştır. Bununla ilgili genel bilgiler Bölüm 3.4'te verilmiştir. Tüm algoritmalar için popülasyondaki toplam birey sayısı ve maksimum iterasyon sayısı sırasıyla; 30 ve 1000 olarak belirlenmiştir. Buradaki iterasyon kavramı literatürde nesil ya da jenerasyon terimleriyle de ifade edilmektedir. Ayrıca, testlerde kullanılan ağların genel özellikleri ve bunlarla ilgili tanımlayıcı açıklamalar ise Bölüm 3.1 ve Bölüm 3.2'de sunulmuştur.

4.1.1. Önerilen yöntemlerin karşılaştırmalı analizi

Bu bölümde modül tespitinde kullanılan algoritmaların deney sonuçları verilmiştir. Bu sonuçlara göre karşılaştırmalı analizler yapılmış ve değerlendirme sonuçları da sonraki alt başlıklarda sunulmuştur. Burada, ağ modül/topluluk tespiti için literatürde en fazla kullanılan uygunluk fonksiyonu olan modülerlik ölçütü, tez

çalışmasında sunulan sekiz adet algoritma için amaç fonksiyonu olarak seçilmiştir. Bölüm 4.1.1.1'deki rastgele üretilmiş ağlarda ve Bölüm 4.1.1.2'deki gerçek ağlarda gerçekleştirilen deneylerde en uygun sonuçlara ulaşan algoritma Bölüm 2.1.3'te anlatılan tüm amaç fonksiyonlarının uygulanacağı yöntem olarak seçilmiştir. Bölüm 4.1.2'deki diğer bir test çalışmasının sonuçlarına göre seçilen algoritmanın kullanılmasıyla en uygun modül yapılarının tespit edilmesini sağlayan amaç fonksiyonu belirlenmiştir. Böylelikle genellikle en başarılı sonuçları öneren fonksiyonunun 10 farklı fonksiyon içinden seçilebilmesi sağlanmıştır. Ayrıca, bu bölümde elde edilen sonuçlara göre belirlenen en uygun amaç fonksiyonunu uygunluk ölçütü olarak kullanan en başarılı algoritmanın sonuçları, literatürdeki diğer yöntemlerin sonuçları ile karşılaştırılmıştır. Bu sonuçlar Çizelge 4.19'da verilmiştir.

Bu bölümdeki testlerin sonuçları aşağıda sunulan üç ayrı başlıkta incelenmiştir. Bu başlıklarda yapay ağlar ve gerçek dünya ağları üzerinde yapılan testlerin sonuçları ile bu sonuçlarla ilgili yapılan değerlendirmeler sunulmuştur. Sonraki ilk iki bölümde her bir test ağı için algoritmaların başlangıçtan maksimum iterasyon sayısına kadar elde edilen tüm genel (*global*) ve yerel (*lokal*) F_M skorları kaydedilmiştir. Ulaşılan test sonuçları ayrı çizelgelerde verilmiştir. Çizelgelerdeki tüm algoritmaların ortalama skorları, standart sapma değerleri, en kötü (*başarısız*) skorları, en yüksek (*başarılı*) skorları ve global en iyi skorlara göre belirlenen toplam ağ modül sayıları sırasıyla; *ort.*, *std.*, *min.*, *mak.* ve *tms.* ile temsil edilmiştir. Çizelgelerde gösterilen maksimum (*en iyi*) ve minimum (*en kötü*) skorlar, son iterasyonlardaki yerel ve genel en iyi bireylerin nihai skorlarıdır. 4.1.1.3'te ise ilk iki bölümdeki test sonuçlarına göre değerlendirilen istatistiksel sonuçlar ile en başarılı algoritma olarak belirlenen yöntemin daha karmaşık düğüm ilişkileri içeren büyük ağ verilerindeki deney sonuçları verilmiştir.

4.1.1.1. Rastsal olarak üretilen ağlar üzerinde yapılan testlerin sonuçları

Bölüm 3.2'de verilen ağlar *ER* ve *BA* ile ifade edilen iki farklı modele göre rastsal olarak üretilen yapay test ağlarıdır. Bu modellere göre oluşturulan 100, 250, 500 ve 1000 düğümlü ağlar sırasıyla; Çizelge 3.3'te ve Çizelge 3.4'te sunulmuştur. Bu ağlar önerilen yöntemlerin başarı kıyaslaması amacıyla kullanılmıştır. Gerçek verilerdeki testlerden önce ilk olarak farklı modellere göre üretilen farklı özellikteki rastgele ağlar üzerinde testler yapılmıştır. Böylece algoritmaların çeşitli topolojik ve karakteristik

özellikli yapay ağlardaki performansları belirlenen en başarılı yöntemin seçilmesinde bir kriter olarak kullanılmıştır. Diğer bir kriter ise bilinen gerçek ağlardaki sonuçlardır.

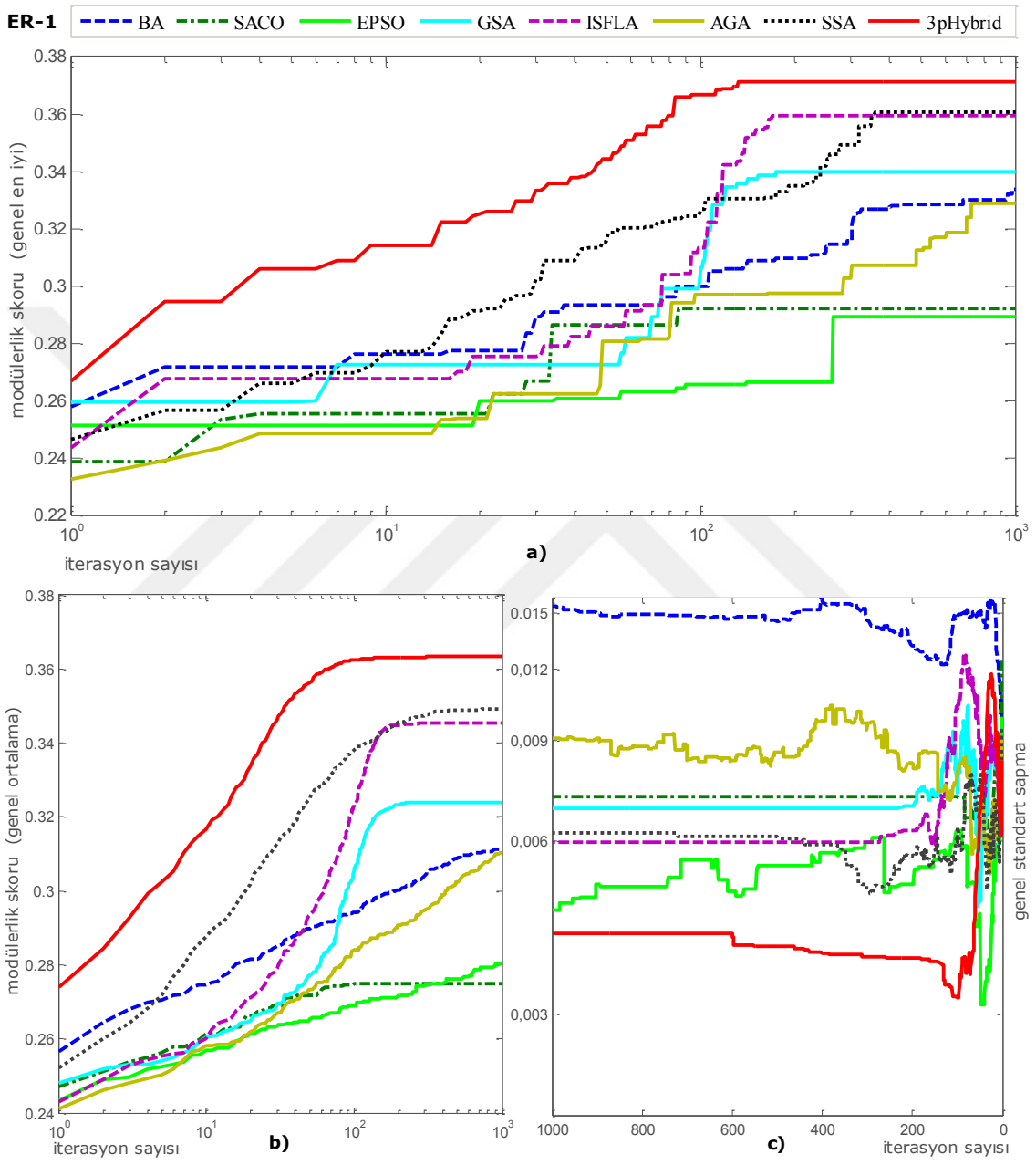
Çizelge 4.1'de *ER* modeline göre oluşturulan verilerdeki deneylere göre tüm algoritmaların genel test sonuçları gösterilmiştir. Çizelgedeki değerler; *BA*, *SACO*, *EPSO*, *GSA*, *ISFLA*, *AGA*, *SSA* ve *3pHybrid* algoritmalarının *ER-1*, *ER-2*, *ER-3* ve *ER-4* ağlarına göre global modülerlik skorlarına göre oluşturulmuştur. Her bir ağ için algoritmalar 1000 iterasyonluk işlemlere göre 30 kez çalıştırılmıştır. Buradaki *ort.* değerleri tüm çalıştırmalar sonundaki en iyi ortalama skorların ortalamasını gösterirken; *std.* sonuçları bu değerlere göre elde edilen standart sapmaları sunar. *std* değerlerinin düşük olması ilgili algoritmanın sonuçları arasındaki farkın tutarlılığını gösterir. *min.* ve *mak.* sayıları algoritmaların tüm çalıştırmalardaki en iyi ve en kötü skorlarını gösterirler. Son sıradaki *tms* değerleri ise algoritmalar tarafından belirlenen global en iyi skorlara göre üretilen ağ modül yapılarındaki toplam modül sayılarıdır.

Çizelge 4.1. *ER* modeline göre üretilen ağların genel test sonuçları

		BA	**SACO**	**EPSO**	**GSA**	**ISFLA**	**AGA**	**SSA**	**3pHybrid**
ER-1	*ort.*	0.3115	0.275	0.2804	0.3239	0.3452	0.3103	0.3491	**0.3636**
	std.	0.01548	0.007187	0.004555	0.006861	0.005983	0.009102	0.006225	**0.004142**
	min.	0.2797	0.2644	0.274	0.3108	0.3357	0.2963	0.3388	**0.3569**
	mak.	0.3337	0.2921	0.2896	0.34	0.3595	0.3291	0.3607	**0.3711**
	tms.	6	17	9	7	5	5	5	6
ER-2	*ort.*	0.3738	0.3364	0.3456	0.3949	0.4156	0.3646	0.4379	**0.4418**
	std.	0.00685	0.003329	**0.002875**	0.007506	0.007814	0.006228	0.00426	0.009761
	min.	0.3635	0.3316	0.3418	0.3812	0.3998	0.3545	0.4296	**0.4302**
	mak.	0.3894	0.3426	0.3519	0.4083	0.4266	0.3786	0.4464	**0.4567**
	tms.	19	18	26	13	11	22	10	9
ER-3	*ort.*	0.3086	0.2855	0.2925	0.3348	0.3529	0.3002	0.3475	**0.3878**
	std.	0.005277	0.001609	**0.001518**	0.005505	0.005436	0.002517	0.00303	0.002619
	min.	0.2971	0.2832	0.2898	0.323	0.3438	0.2957	0.3444	**0.384**
	mak.	0.3155	0.2883	0.2961	0.3446	0.3633	0.3049	0.3537	**0.393**
	tms.	30	37	40	25	16	33	17	12
ER-4	*ort.*	0.288	0.2756	0.2799	0.2946	0.3076	0.2847	0.3121	**0.3633**
	std.	0.002233	0.001212	**0.0008103**	0.003429	0.004776	0.002076	0.009992	0.001904
	min.	0.2853	0.2748	0.2788	0.2903	0.2997	0.2813	0.3106	**0.3604**
	mak.	0.2911	0.2777	0.2808	0.2998	0.3136	0.2874	0.349	**0.3649**
	tms.	52	70	74	52	46	67	19	17

Yukarıdaki çizelgede *ER-1* ağının tüm sonuçlarına göre en uygun değerlere *3pHybrid* algoritması ulaşmıştır. En iyi *ort.*, *std.*, *min.* ve *mak.* sayıları sırasıyla; 0.3636, 0.004142, 0.3569 ve 0.3711'dir. En iyi maksimum skora (*0.3711*) göre elde edilen ağ

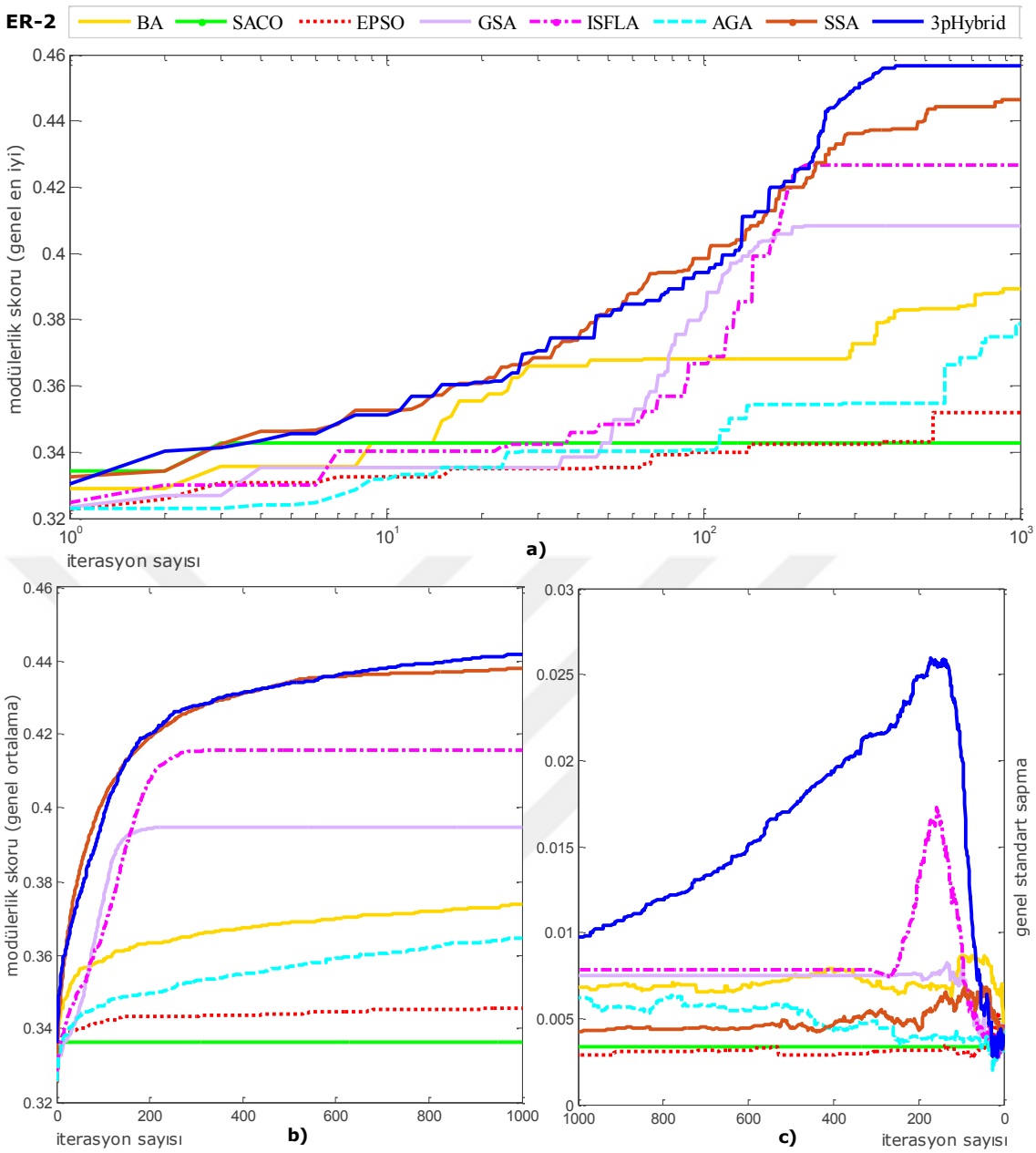
modülleri sayısı ise 6'dır. Bu sayı ağın 6 parçaya ayrılmasının uygunluğunu temsil eder. Her bir parça ağdaki düğümlerin etkileşimlerine göre uygun düğümlere sahiptir ve böylece kendi modüllerindeki diğer düğümlerle yoğun ilişkili fakat diğer modüllerdeki düğümlerle daha az ilişkili topluluk yapıları elde edilmiş olur.



Şekil 4.1. ER-1 yapay ağındaki karşılaştırmalı test sonuçları: **a)** iterasyon sayısına göre genel en iyi skorlar, **b)** genel ortalama skorlar, **c)** genel standart sapma değerleri

ER-1 ağı üzerinde gerçekleştirilen testlerde her bir algoritmanın iterasyon sayısına göre global en iyi modülerlik skorları Şekil 4.1-a'da verilmiştir. Bu sonuçlar sekiz algoritmanın 30 kez çalıştırılması sonundaki maksimum skorlara göre

oluşturulmuştur. Şekil 4.1-b'de ve Şekil 4.1-c'de ise sırasıyla *ER-1* ağı için her bir iterasyona göre global ortalama skorlar ve global standart sapma değerleri sunulmuştur. Hem en iyi hem de ortalama skorlar açısından *3pHybrid* algoritmasının diğerlerine göre yüksek değerlere ulaştığı ve standart sapma açısından da en düşük sonucu verdiği gözlemlenmiştir. Ayrıca bu algoritmanın elde ettiği en düşük skoru (*min.*) 0.3569'dur ve bu sonuç diğer algoritmaların sonuçlarına kıyasla en yüksek değeri temsil eder. Burada *3pHybrid* algoritması en başarılı yöntem olarak etiketlenmiştir. En başarısız sonuçlar ise maksimum değer için *EPSO* algoritmasının; ortalama ve minimum değerler için *SACO* algoritmasının sonuçlarıdır. *SACO*'nun *tms.* değeri 17'dir ve bu, diğer algoritmaların sonuçlarına kıyasla en yüksek değerdir. Bundan dolayı bu algoritmanın *ER-1* ağı için uygun bir gruplama sağlamadığı anlaşılmaktadır. Çizelge 4.1'deki genel standart sapma sonuçları için en yüksek değer (*0.01548*) ise *BA* algoritmasına aittir.

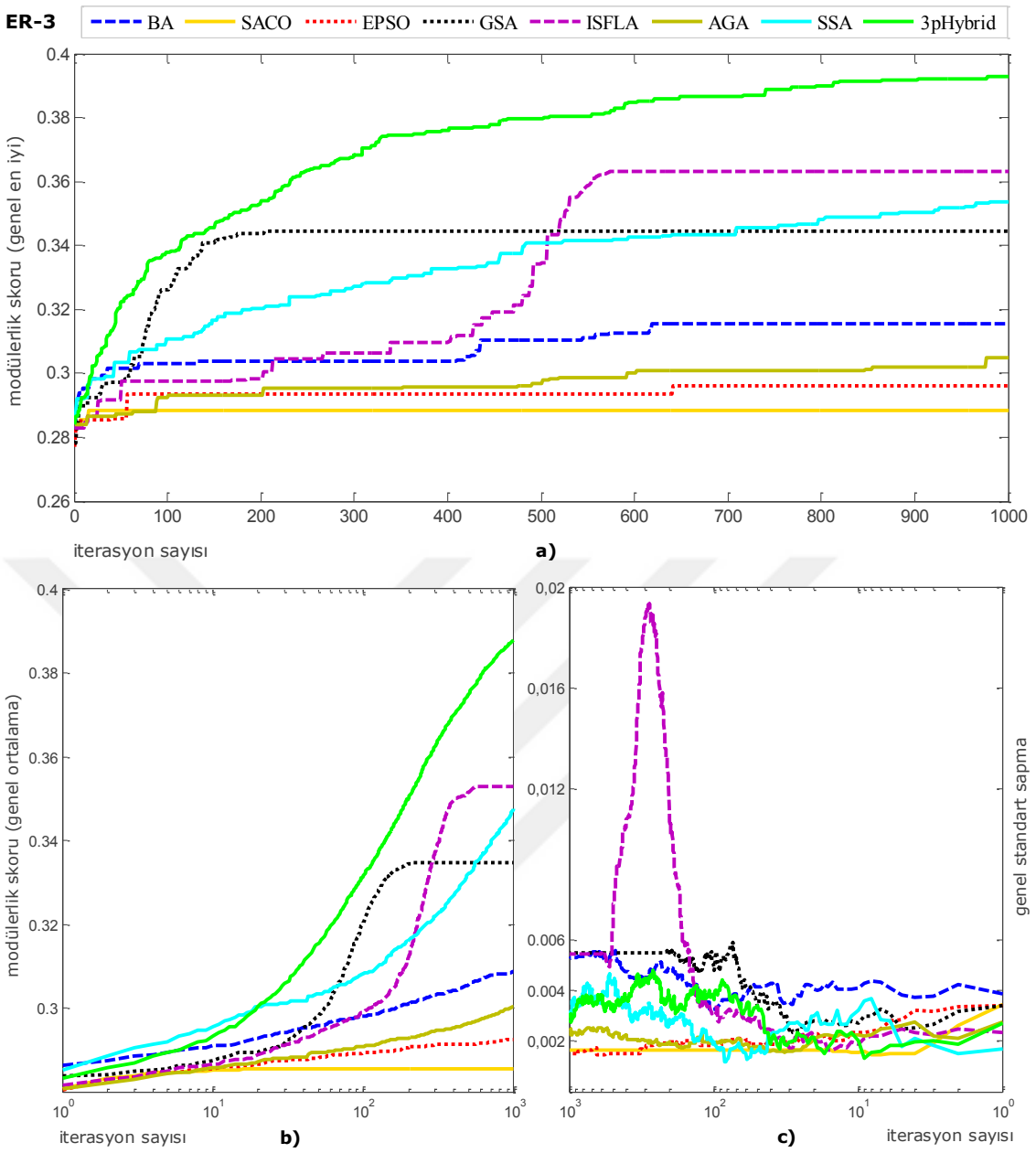


Şekil 4.2. ER-2 ağı için iterasyonlara göre **a)** genel en iyi modülerlik skorları, **b)** genel ortalama skorlar, **c)** genel standart sapma değerleri

ER-2 ağının giriş verisi olarak kullanıldığı deneylerde tüm algoritmalara göre elde edilen global en yüksek ve en düşük modülerlik skorları, ortalama skorlar ve standart sapma değerleri Çizelge 4.1'de sunulmuş ve bu sonuçlara göre oluşturulan karşılaştırma grafikleri Şekil 4.2'de gösterilmiştir. Bu ağda en yüksek değerlere göre en iyi sonucu *3pHybrid* algoritması elde etmiştir. Bu algoritma aynı zamanda ortalama ve minimum skorlara göre de en uygun sonuçları vermiştir. Standart sapma sonuçları açısından ise en düşük (*uygun*) değere *EPSO* algoritması ulaşmıştır. *3pHybrid* algoritmasının global standart sapma değeri diğer algoritmaların *std.* değerlerinden daha

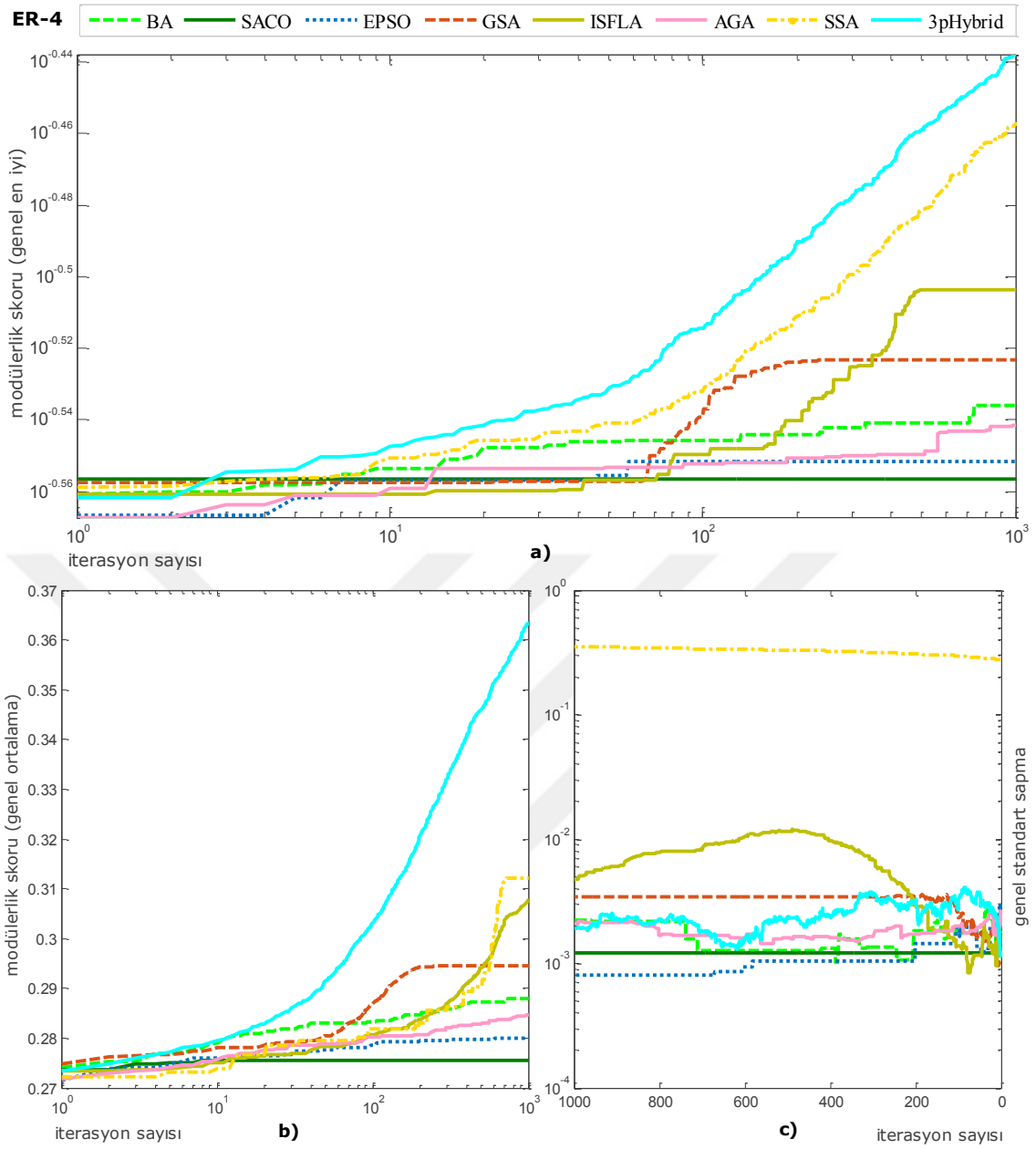
yüksektir. Bu algoritmanın global en yüksek modülerlik skoru ise 0.4567'dir ve buna göre kaydedilen *tms.* değeri 9'dur. Böylece *ER-2* ağının 9 farklı boyuttaki bölümlere ayrılmasının modülerlik açısından daha uygun olduğu sonucuna varılmıştır. *BA*, *SACO*, *EPSO*, *GSA*, *ISFLA*, *AGA* ve *SSA* algoritmaları için bu sayılar sırasıyla; 19, 18, 26, 13, 11, 22 ve 10'dur. Şekil 4.2-a'daki genel maksimum ve Şekil 4.2-b'deki genel ortalama skorlara göre *3pHybrid* ile *SSA* algoritmalarının sonuçlarının birbirlerine oldukça yakın oldukları gözlemlenmiştir. Bu durum ağlardaki düğümlerin yoğunluklarının homojen ya da heterojen dağılımlara sahip olmalarından veya önerilen metasezgisel yaklaşımlar için başlangıç popülasyonlarının rastsal olarak oluşturulmasından kaynaklanmış olabilir.

Şekil 4.3'te tüm algoritmaların *ER-3* yapay ağına göre gerçekleştirilen test sonuçları karşılaştırmalı olarak sunulmuştur. Önceki iki ağ verisinin sonuçlarında olduğu gibi burada da *3pHybrid* algoritması genel en iyi, en kötü ve ortalama eğerler açısından maksimum sonuçlara ulaşmıştır. Sonraki en iyiler *ISFLA* ve *SSA* algoritmalarıdır. En kötü sonuçlar da *EPSO* ve *SACO* algoritmalarına aittir. Bu ağdaki en uygun modül sayısı (*tms.*) 12'dir ve bu sonuç 0.393 maksimum skoruna göre *3pHybrid* algoritması ile elde edilmiştir. Diğer algoritmaların *tms.* değerleri daha yüksektir. Global en uygun standart sapma sonuçlarının ortalamaları açısından da *EPSO* algoritması en uygun değere (*0.001518*) sahiptir. Şekil 4.3-a'ya göre *3Hybrid*, *SSA* ve *AGA* algoritmalarının global en iyi skorları maksimum iterasyon sayısına dek genelde sürekli artış gösterirken; diğer algoritmalar için belli iterasyonlardan sonra aday çözümler iyileştirilememiş ve daha uygun sonuçlara ulaşılamamıştır. Bu durumda *ER-3* ağı üzerindeki deneyler için *3Hybrid*, *SSA* ve *AGA* algoritmalarının popülasyonlarındaki bireylerin ya da maksimum iterasyon sayılarının artırılmasıyla daha uygun sonuçlara ulaşılabileceği anlaşılmaktadır.



Şekil 4.3. ER-3 yapay test ağında elde edilen sonuçlar: **a)** genel en iyi F_M skorları, **b)** genel ortalama F_M skorları, **c)** genel standart sapma değerleri

Aşağıda gösterilen Şekil 4.4'deki üç grafik (*a, b, c*), *ER-4* test verisi için global en iyi skorlara göre oluşturulan maksimum, ortalama ve standart sapma değerlerinin karşılaştırılmasını ifade eder. Ayrıca bu ağ için tüm sonuçlar Çizelge 4.1'de verilmiştir.



Şekil 4.4. ER-4 ağı için **a)** genel en iyi modülerlik skorları, **b)** genel ortalama skorlar, **c)** genel standart sapma değerleri

ER-4 yapay ağı üzerinde gerçekleştirilen testlerdeki genel en iyi F_M skorlarına göre tüm algoritmaların başarı sıralamaları şöyledir: *3pHybrid*, *SSA*, *ISFLA*, *GSA*, *BA*, *AGA*, *EPSO*, *SACO*. En yüksek F_M skoru (*mak.*) 0.3649'dur ve bu skora göre elde edilen *tms.* sayısı 17'dir. Şekil 4.4-a'da gösterilen maksimum skorlara ve Şekil 4.4-b'de verilen genel ortalama skorlara göre *3pHybrid* algoritmasının *ER-4* ağı üzerindeki test sonuçları diğer algoritmaların sonuçlarından daha iyidir. Ancak global standart sapma sonuçları açısından bu algoritma aynı başarıya sahip değildir.

Çizelge 4.1'deki ağların global sonuçları incelendiğinde, *ER-1* ağı hariç global en iyi skorlara göre giriş ağının minimum sayıda bölümlere ayrıldığı gözlemlenmiştir. Bu yüzden genelde daha düşük *tms.* değerlerinin daha uygun modülerlik skorlarına denk geldiği anlaşılmıştır. Örneğin *ER-1* ağı için *ISFLA*, *AGA* ve *SSA* algoritmaları bu ağı 5 bölüme ayırırken; en iyi skoru sunan *3pHybrid* algoritması 6 bölüme ayırmıştır. *tms* değerini 5 olarak öneren algoritmalardan *SSA* ve *ISFLA*, sırasıyla sonraki ikinci ve üçüncü hem en iyi hem de ortalama modülerlik skorlarını sunan algoritmalardır. Bu sebeple genelde giriş ağını daha az bölümlere ayıran algoritmaların daha uygun sonuçlar verdiği söylenilebilir. Bu tespit, *ER* modeline göre üretilen yapay ağların sonuçları için geçerlidir. Genel en iyi değerlere göre elde edilen sonuçların yanında bu modele göre üretilen dört farklı boyuttaki ağın yerel test sonuçları Çizelge 4.2'de sunulmuştur.

Çizelge 4.2. *ER* modeline göre üretilen ağların yerel test sonuçları

		BA	**SACO**	**EPSO**	**GSA**	**ISFLA**	**AGA**	**SSA**	**3pHybrid**
ER-1	*ort.*	0.2998	0.2323	0.2422	0.3222	0.3327	0.3012	0.3368	**0.3605**
	std.	0.01674	0.007775	0.01096	0.007199	0.006035	0.008224	0.006982	**0.004089**
	min.	0.2706	0.2171	0.2231	0.3108	0.3161	0.2806	0.3208	**0.3544**
	mak.	0.3276	0.2479	0.2647	0.34	0.3423	0.3101	0.3521	**0.3698**
ER-2	*ort.*	0.3645	0.2626	0.3263	0.3944	0.4099	0.3521	0.4201	**0.4321**
	std.	0.008444	0.004215	**0.003503**	0.007637	0.005183	0.008001	0.00409	0.008046
	min.	0.3538	0.2532	0.3195	0.3805	0.4001	0.3337	0.4175	**0.4301**
	mak.	0.385	0.2696	0.3351	0.4077	0.4202	0.3601	0.4395	**0.4563**
ER-3	*ort.*	0.3011	0.2044	0.2813	0.3346	0.3499	0.2986	0.3439	**0.3877**
	std.	0.005652	**0.002388**	0.003884	0.005423	0.005077	0.00419	0.00407	0.003043
	min.	0.2853	0.2007	0.2759	0.323	0.3397	0.2801	0.343	**0.3833**
	mak.	0.309	0.2083	0.2902	0.3438	0.3518	0.3032	0.3509	**0.3927**
ER-4	*ort.*	0.282	0.1818	0.2725	0.2945	0.3012	0.2841	0.3119	**0.3625**
	std.	0.00323	0.001406	0.002982	0.003431	0.003991	0.001509	0.009809	**0.001404**
	min.	0.2775	0.1795	0.2701	0.2903	0.2908	0.28	0.3088	**0.3601**
	mak.	0.2854	0.1828	0.2776	0.2998	0.3066	0.2863	0.3201	**0.3639**

ER-1 ve *ER-4* ağları için *3pHybrid* algoritması *yerel*—en iyi, en kötü ve ortalama en yüksek skorlara sahiptir. Aynı şekilde bu algoritma en uygun yerel standart sapma değerlerine de sahiptir. *ER-2* ve *ER-3* ağları için bu algoritma standart sapma değerleri açısından daha kötü sonuçlar elde etmiştir. Ancak diğer skorlara göre yine en başarılı algoritmadır. *ER-2* verisi için 0.003503 *std.* değeriyle *EPSO* algoritması; *ER-3* verisi için 0.002388 *std.* değeriyle *SACO* algoritması en uygun sonuçları vermiştir. Ayrıca, Bölüm 4.1.1.3'te *ER* ve *BA* ağlarının global en iyi sonuçlarını birden dikkate alan istatistiksel testler yapılmış ve sonuçlara göre en uygun algoritma belirlenmiştir.

Bundan sonraki deneylerde yapay veriler olarak Bölüm 3.2.2'deki *BA* modeline göre üretilen *BA-1*, *BA-2*, *BA-3* ve *BA-4* test ağları kullanılmıştır. Bu dört farklı ağın global en iyi sonuçlarına göre Çizelge 4.3'teki tablo oluşturulmuştur. Tablodaki *BA-1*, *BA-3* ve *BA-4* ağları üzerindeki testlerde *3pHybrid* algoritması genel maksimum, minimum ve ortalama F_M skorları açısından en uygun sonuçlara ulaşmıştır. *BA-2* ağındaki testler için ise bu algoritma ortalama ve maksimum en iyi sonuçları elde etmiştir fakat en uygun *min.* değerine (*0.3848*) *SSA* ulaşmıştır. *3pHybrid* algoritmasının global standart sapma değerleri ise genelde yüksektir. Ayrıca her bir algoritmanın global maksimum skorlarına göre belirlenen toplam ağ modül sayıları (*tms.*) aşağıdaki çizelgede sunulmuştur.

Çizelge 4.3. Dört farklı *BA* modeli ağı için elde edilen genel test sonuçları

		BA	**SACO**	**EPSO**	**GSA**	**ISFLA**	**AGA**	**SSA**	**3pHybrid**
BA-1	*ort.*	0.3345	0.2999	0.2992	0.3549	0.3692	0.3301	0.3693	**0.3844**
	std.	0.01475	0.007274	0.004082	0.01023	0.005027	0.009005	**0.003626**	0.00464
	min.	0.3087	0.2893	0.2912	0.3305	0.3586	0.3165	0.3636	**0.373**
	mak.	0.3589	0.3179	0.3068	0.3756	0.3776	0.3557	0.3759	**0.3923**
	tms.	6	19	9	6	7	7	6	8
BA-2	*ort.*	0.3258	0.2911	0.3003	0.3571	0.3744	0.3194	0.3916	**0.4077**
	std.	0.00962	**0.003691**	0.003888	0.005706	0.006074	0.004355	0.003985	0.01087
	min.	0.3038	0.2858	0.2934	0.3472	0.3631	0.3105	**0.3848**	0.3823
	mak.	0.3381	0.2998	0.3076	0.3682	0.3824	0.3274	0.3996	**0.4187**
	tms.	15	15	17	13	11	18	7	10
BA-3	*ort.*	0.3186	0.2948	0.3013	0.3467	0.367	0.311	0.3998	**0.402**
	std.	0.005946	**0.001436**	0.001614	0.005541	0.003493	0.002947	0.003147	0.009021
	min.	0.3057	0.2924	0.2984	0.3343	0.3605	0.306	0.3626	**0.3754**
	mak.	0.3283	0.2974	0.304	0.3553	0.3712	0.3163	0.4032	**0.4129**
	tms.	32	39	43	21	20	32	11	11
BA-4	*ort.*	0.308	0.2951	0.299	0.3153	0.3327	0.3052	0.3644	**0.3837**
	std.	0.002811	0.001521	**0.000858**	0.002946	0.004307	0.002006	0.01442	0.001188
	min.	0.3041	0.2926	0.2979	0.3099	0.325	0.3028	0.328	**0.3815**
	mak.	0.3131	0.2982	0.3005	0.3188	0.3367	0.309	0.3734	**0.385**
	tms.	72	76	78	64	54	77	26	18

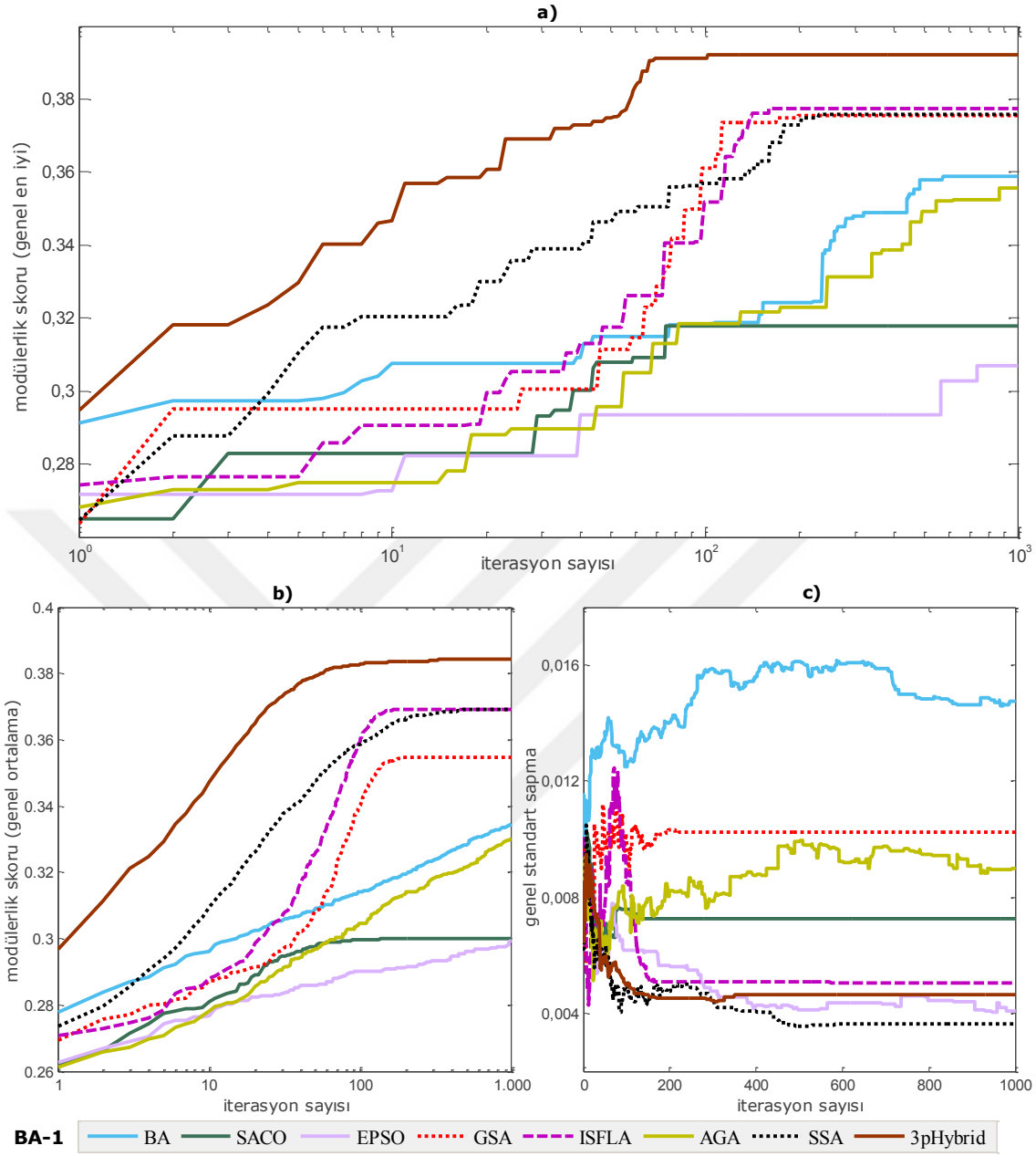
Çizelge 4.4'te ise önerilen algoritmaların *BA* ağlarında elde ettikleri yerel en iyi skorlara göre *ort.*, *std.*, *min.* ve *mak.* sonuçları sunulmuştur. *BA-1* ve *BA-4* ağlarındaki deneylerde yerel en iyi sonuçların hepsinde *3pHybrid* algoritması en uygun skorlara ulaşmıştır. *BA-3* ağındaki testler için en iyi *min.* değerini *SSA* algoritması elde etmiştir. Yerel ortalama standart sapma değerleri için *BA-2* ağında yine *SSA* algoritması; *BA-3* ağında ise *SACO* algoritması en uygun sonuçları vermiştir.

Çizelge 4.4. *BA* modeli ile oluşturulan ağların yerel test sonuçları

		BA	SACO	EPSO	GSA	ISFLA	AGA	SSA	3pHybrid
BA-1	*ort.*	0.3262	0.2489	0.2641	0.3537	0.3588	0.3279	0.3682	**0.381**
	std.	0.01932	0.008543	0.01017	0.01017	0.008027	0.007522	0.003906	**0.003871**
	min.	0.2946	0.2331	0.2393	0.3298	0.3499	0.3177	0.3634	**0.3732**
	mak.	0.355	0.2653	0.2821	0.3756	0.3762	0.3459	0.3701	**0.3921**
BA-2	*ort.*	0.3153	0.2213	0.2772	0.3568	0.3688	0.3004	0.3825	**0.4071**
	std.	0.01046	0.0039	0.005075	0.005609	0.005597	0.006018	**0.002791**	0.01989
	min.	0.2917	0.2125	0.2669	0.3472	0.359	0.2997	0.3751	**0.3755**
	mak.	0.3325	0.2276	0.2859	0.3681	0.3703	0.3101	0.3899	**0.4187**
BA-3	*ort.*	0.3109	0.2095	0.2898	0.3466	0.3602	0.3083	0.3902	**0.3936**
	std.	0.005728	**0.002345**	0.002851	0.005575	0.005493	0.006947	0.003147	0.01799
	min.	0.297	0.2042	0.285	0.3343	0.3583	0.2996	**0.3837**	0.3626
	mak.	0.3213	0.2138	0.2957	0.3553	0.3651	0.3109	0.402	**0.4129**
BA-4	*ort.*	0.3029	0.1965	0.2911	0.3151	0.3233	0.3045	0.3563	**0.3835**
	std.	0.002481	0.001894	0.001932	0.002901	0.009706	0.002085	0.005168	**0.001024**
	min.	0.2993	0.1937	0.2879	0.3099	0.3189	0.3021	0.3399	**0.3823**
	mak.	0.307	0.1992	0.2951	0.3184	0.3367	0.3087	0.3722	**0.3846**

Aşağıda sunulan karşılaştırma grafiklerinde *BA-1*, *BA-2*, *BA-3* ve *BA-4* ağlarında gerçekleştirilen testlerin iterasyonlarına göre *global*—en iyi skorlar (***a***), ortalama skorlar (***b***) ve standart sapma (***c***) değerleri verilmiştir. Bu sonuçlar her bir algoritmanın 30 kez çalıştırılması sonrası elde edilmiştir.

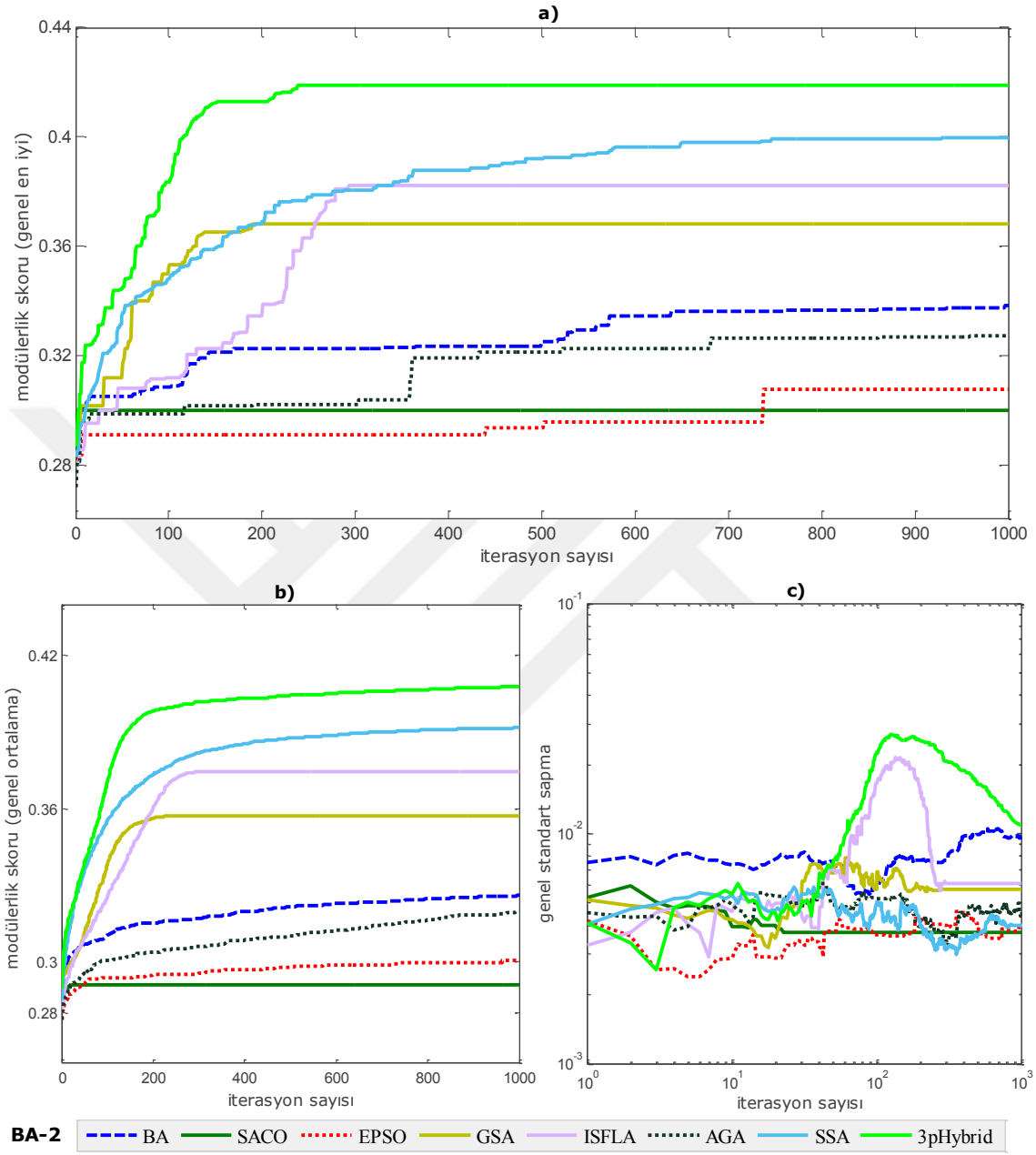
Şekil 4.5-a'da *BA-1* yapay ağı üzerindeki testler için iterasyon sayılarına göre en iyi sonuca belirgin farkla *3pHybrid* algoritması ulaşmıştır. Bu grafik incelendiğinde ilk iterasyondan itibaren son iterasyona kadar en iyi modülerlik skorları bu algoritmaya aittir. Aynı tespitler genel ortalama skorları için de geçerlidir. Global ortalama değerlere göre sonraki en iyi üç algoritma sırasıyla *SSA*, *ISFLA* ve *GSA* algoritmalarıdır. Bu sıralama yerel skorlar için de aynıdır. Şekil 4.5-c'deki global ortalama—standart sapma sonuçları açısından ise en iyi algoritma *SSA*'dır. En kötü *std.* değerleri ise *BA* algoritmasına aittir. Global en iyi ortalama F_M değerleri açısından son dört algoritmanın (*BA*, *AGA*, *SACO* ve *EPSO*) iterasyonlara göre elde edilen skorları birbirlerine yakındır. Çizelge 4.3'e uygun bir şekilde Şekil 4.5-a'daki en yüksek (*uygun*) F_M skorlarına göre algoritmaların başarı sıralamaları: *3pHybrid* (*0.3923*), *ISFLA* (*0.3776*), *SSA* (*0.3759*), *GSA* (*0.3756*), *BA* (*0.3589*), *AGA* (*0.3557*), *SACO* (*0.3179*) ve *EPSO* (*0.3068*) şeklinde olur. Parantez içinde gösterilen değerler algoritmaların en iyi modülerlik skorlarını ifade ederler. Son olarak, en başarılı algoritmanın sunduğu *tms.* sayısı 8'dir. Buna göre *BA-1* ağının 8 farklı bölüme ayrılmasının uygun olduğu anlaşılır.



Şekil 4.5. BA-1 yapay ağı üzerindeki testler için iterasyon sayılarına göre —genel— **a)** en iyi skorlar, **b)** ortalama skorlar, **c)** standart sapma değerleri

BA-2 ağının iterasyonlara göre elde edilen karşılaştırmalı sonuçları Şekil 4.6'da verilmiştir. Şekil 4.6-a'da ve Şekil 4.6-b'de verilen skorlara göre her bir iterasyondaki en uygun sonuçlar yeşil çizgilerle gösterilmiştir. Bunlar *3pHybrid*'in test sonuçlarıdır Şekil 4.6-b'de gösterilen genel ortalama skorlara dikkat edildiğinde, genelde belirli iterasyonlardan sonra algoritmaların ulaştıkları ortalama değerlerin eğilimleri yaklaşık olarak sabitlenmiştir. Şekil 4.6-c'deki *std.* sonuçlarına göre kıyaslama yapıldığında,

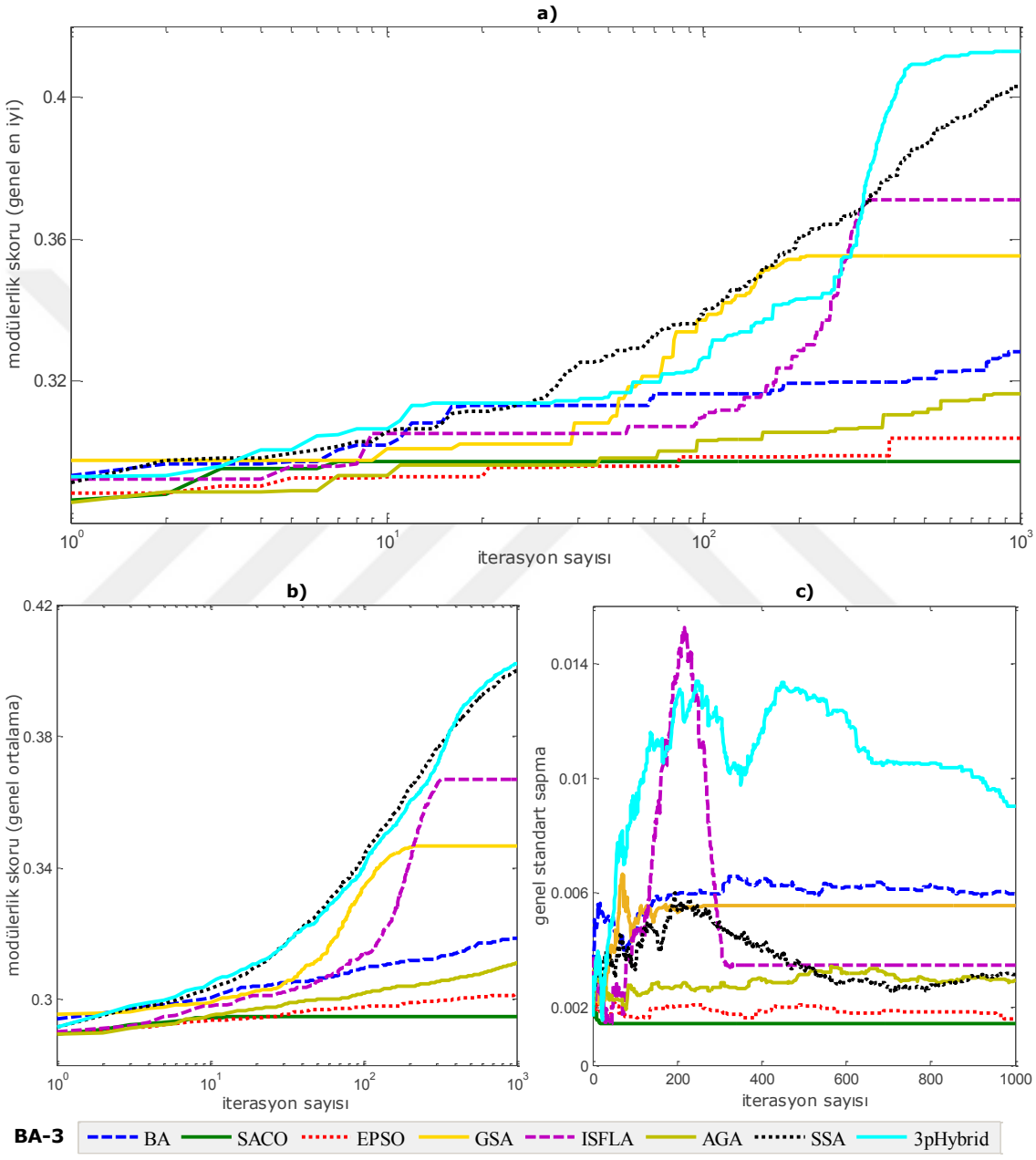
genelde en iyi sonuçlara *SACO* ve *EPSO* algoritmalarının ulaştıkları gözlemlenmiştir. En yüksek değerler (*başarısız*) ise *3pHybrid*, *BA* ve *ISFLA* algoritmalarına aittir.



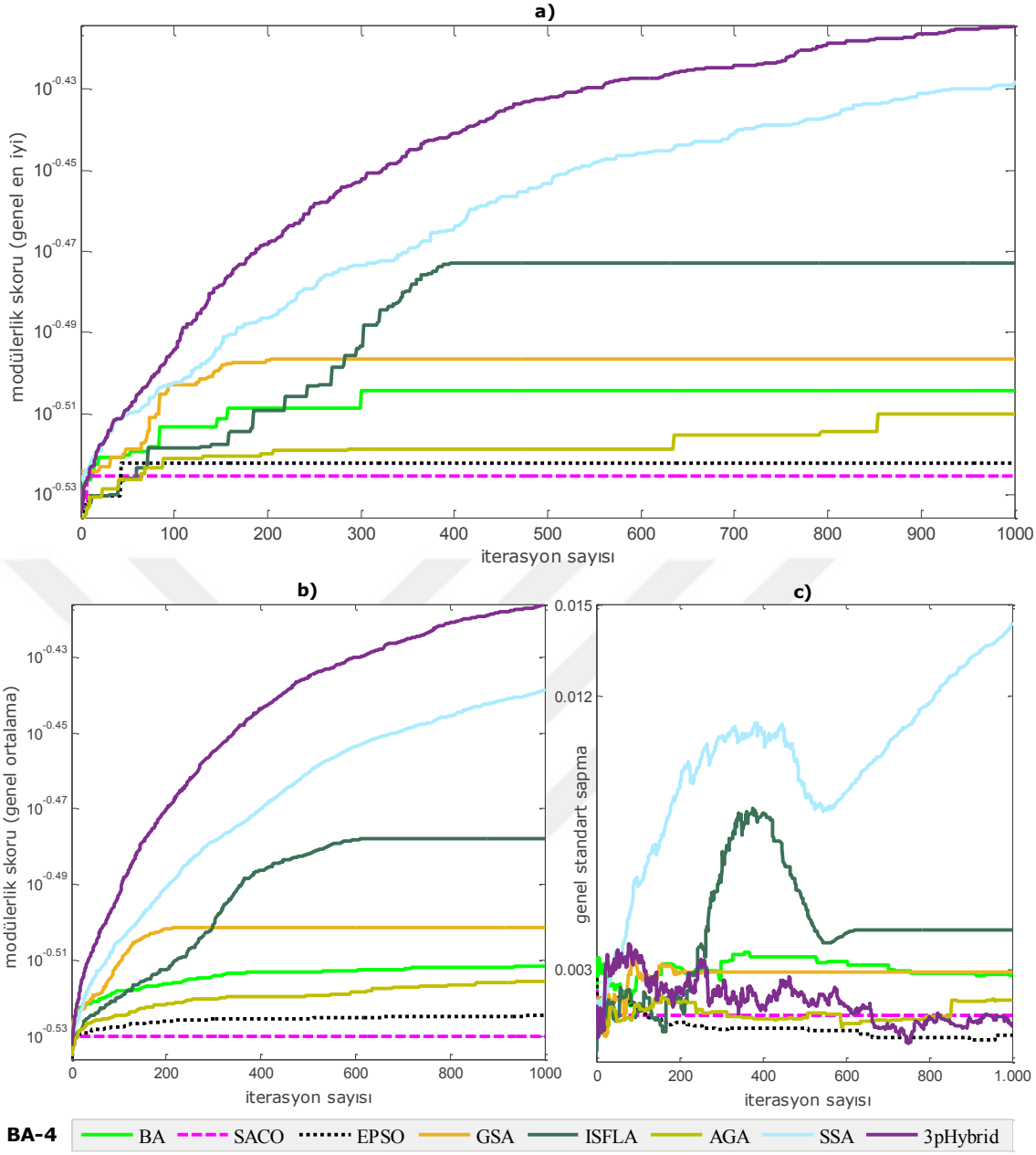
Şekil 4.6. BA-2'deki test sonuçları için her bir iterasyona göre **a)** genel en iyi skorlar, **b)** genel ortalama skorlar, **c)** genel standart sapma değerleri

Aşağıda sunulan Şekil 4.7'de *BA-3* test ağının tüm algoritmalara göre elde edilen sonuçları gösterilmiştir. Şekil 4.7-a'daki genel en yüksek skorların kıyaslanmasında *3pHybrid* algoritmasının yaklaşık olarak 375. iterasyondan sonra *SSA* algoritmasından daha iyi sonuçları elde ettiği anlaşılmaktadır. Önceki iterasyonların genelinde, *SSA*'nın sonuçlarının daha iyi olduğu görülmektedir. Bu duruma benzer diğer bir grafik Şekil

4.7-b'de gösterilmiştir. *BA-3* ağı için global en iyi değerlerin ortalamasında *3pHybrid* ile *SSA* algoritmalarının sonuçları birbirlerine oldukça yakındır. *3pHybrid* algoritması *SSA*'ya kıyasla en iyi maksimum skorlara benzer şekilde son iterasyonlarda ulaşmıştır. Standart sapma değerlerine göre algoritmalar kıyaslandığında en iyiden en kötüye doğru sıralama genelde şöyle olur: *SACO*, *EPSO*, *AGA*, *SSA*, *ISFLA*, *GSA*, *BA* ve *3pHybrid*.



Şekil 4.7. BA-3 yapay ağında iterasyonlara göre ulaşılan: **a)** genel en yüksek F_M skorları, **b)** genel ortalama F_M skorları, **c)** genel standart sapma değerleri



Şekil 4.8. BA-4 ağındaki test sonuçlarına göre her iterasyon için **a)** genel en iyi skorlar, **b)** genel ortalama skorlar, **c)** genel standart sapma değerleri

Son olarak, *BA-4* yapay ağı üzerindeki testlerde kullanılan 8 adet algoritmanın her bir iterasyona göre genel en iyi ve genel ortalama skorları ile genel standart sapma değerleri karşılaştırmalı olarak Şekil 4.8'deki grafiklerde gösterilmiştir. İlk iki grafikte *3pHybrid* algoritmasının diğer algoritmalara kıyasla belirgin bir başarıya sahip olduğu gözlemlenmiştir. Sonraki en iyi başarı sıralamaları *BA-3* ağı hariç genelde önceki ağlardaki sıralamalar gibidir. Ayrıca *BA-4* ağı için *EPSO*, *AGA*, *3pHybrid* ve *SACO* algoritmalarının *std.* değerleri genelde birbirlerine yakındır.

4.1.1.2. Gerçek dünya ağları üzerinde yapılan testlerin sonuçları

Bu bölümde, çeşitli alanlardan gerçek dünya ağlarının temsil edildiği *Grup-A* kategorisindeki test verilerine göre elde edilen deney sonuçları verilmiştir. Burada altı adet ağ kullanılmıştır. Burada, *Zachary's Karate Club*, *Bottlenose Dolphins*, *American College Football* ve *Books About US Politics*, *E. coli Transkripsiyon* ve *Helicobacter Pylori PPI* ağları sırasıyla; "A-1", "A-2", "A-3", "A-4", "A-5" ve "A-6" kısaltmaları ile temsil edilmiştir. Bu ağlarla ilgili açıklayıcı bilgilere Bölüm 3.1.1.1'de ulaşılabilir. Verilen küçük ve orta boyutlu etkileşim ağları ağlarındaki modüllerin tanımlanması amacıyla tüm algoritmaların test sonuçları ayrıntılı olarak aşağıda verilmiştir.

Çizelge 4.5'te küçük ve orta boyutlu gerçek dünya ağlarının genel (*global*) sonuçları gösterilmiştir. Ayrıca Çizelge 4.6'da yerel sonuçlar da sunulmuştur. Burada, tüm algoritmaların ortalama, en düşük ve en yüksek modülerlik skorları; *ort.*, *min.*, *mak.* ile gösterilmiştir. 30 kez çalıştırma sonunda ortalama standart sapma değerleri ise *std.* ile temsil edilmiştir. Hem genel hem de yerel sonuçlara göre genellikle en iyi sonuçlar, *3pHybrid* algoritması ile elde edilmiştir. Sonraki en yüksek modülerlik skorlarına ise genel olarak *SSA* ile ulaşılmıştır. Genellikle buradaki tüm test sonuçları için ortalama, en iyi ve en kötü F_M değerlerine göre en başarılı sonuçlara bir önceki yapay verilerdeki testlerde olduğu gibi yine *3pHybrid* algoritması ile ulaşılmıştır. Rastgele ağlardaki sonuçlardan farklı olarak gerçek dünya ağlarında *3pHybrid* algoritmasının genel standart sapma değerleri hariç diğer üç kriterlere göre başarılar daha belirgindir. Böyle bir çıkarım bu bölümdeki karşılaştırma grafiklerinden de anlaşılmaktadır. Bunlara ek olarak, tüm algoritmaların gerçek ve rastgele üretilmiş ağlardaki başarı sıralamaları ile birbirleri arasındaki farklar sonraki bölümde verilen istatiksel analizlerle de değerlendirilmiştir.

Aşağıda verilen genel test sonuçlarına göre *karate* (A-1) ağındaki testlerde tüm algoritmalar en yüksek F_M skoru olan 0.4198 değerine ulaşmışlardır. Bu değere göre ağ 4 farklı boyuttaki modüllere ayrılmaktadır. *E. coli* (A-5) ve *Helicobacter P.* (A-6) ağlarındaki sonuçlar hariç; 3*pHybrid* algoritması diğer tüm ağlardaki kriterlerin hepsinde bilinen en iyi sonuçları elde etmiştir. Buradaki iki ağın sonucu için sadece *std.* değerleri daha yüksektir. Diğer kriterler için bu algoritma en uygun skorlara ulaşmıştır.

Çizelge 4.5. Küçük veya orta büyüklükteki ağların genel test sonuçları

		BA	SACO	EPSO	GSA	ISFLA	AGA	SSA	3pHybrid
	ort.	0.4004	0.4185	0.4185	0.4191	0.4197	**0.4198**	**0.4198**	**0.4198**
	std.	0.01829	0.002891	0.001335	0.001305	0.0005329	**0**	**0**	**0**
A-1	*min.*	0.3589	0.407	0.4156	0.4151	0.4174	**0.4198**	**0.4198**	**0.4198**
	mak.	**0.4198**	**0.4198**	**0.4198**	**0.4198**	**0.4198**	**0.4198**	**0.4198**	**0.4198**
	tms.	4	4	4	4	4	4	4	4
	ort.	0.4881	0.4761	0.4988	0.5236	0.5265	0.5271	0.527	**0.5285**
	std.	0.02396	0.01223	0.007219	0.005106	0.001683	0.00063	0.00223	**0**
A-2	*min.*	0.4407	0.4571	0.4879	0.5127	0.522	0.5259	0.5186	**0.5285**
	mak.	0.5208	0.4994	0.5146	0.5277	**0.5285**	**0.5285**	**0.5285**	**0.5285**
	tms.	4	5	5	5	5	5	5	5
	ort.	0.5261	0.4754	0.4487	0.5811	0.599	0.5671	0.5866	**0.6046**
	std.	0.03323	0.01496	0.01514	0.01861	0.004602	0.01037	0.009975	**3.63E-05**
A-3	*min.*	0.4608	0.4465	0.4234	0.5315	0.59	0.5509	0.5605	**0.6044**
	mak.	0.578	0.5096	0.4886	0.6016	**0.6046**	0.5928	0.6006	**0.6046**
	tms.	8	12	9	9	10	8	7	10
	ort.	0.5051	0.4716	0.5018	0.5211	0.5266	0.5269	0.5268	**0.5272**
	std.	0.01173	0.01172	0.009429	0.006391	0.001049	0.0002115	0.000728	**0**
A-4	*min.*	0.4855	0.4513	0.4877	0.5062	0.5237	0.5264	0.5247	**0.5272**
	mak.	0.5262	0.4924	0.5207	**0.5272**	**0.5272**	**0.5272**	**0.5272**	**0.5272**
	tms.	4	5	5	5	5	5	5	5
	ort.	0.7698	0.7695	0.739	0.778	0.782	0.7693	0.7849	**0.7856**
	std.	0.004918	0.005871	0.003793	0.002551	0.001959	0.0033	**0.001429**	0.001966
A-5	*min.*	0.7634	0.7607	0.7337	0.7745	0.7779	0.7636	**0.7818**	**0.7818**
	mak.	0.7764	0.7768	0.7481	0.7823	0.7837	0.7742	0.7869	**0.7873**
	tms.	50	50	62	48	47	51	46	45
	ort.	0.4838	0.4828	0.4646	0.4874	0.5009	0.4845	0.5374	**0.543**
	std.	0.005703	0.008958	**0.001759**	0.003802	0.00453	0.004392	0.004049	0.003319
A-6	*min.*	0.4757	0.4695	0.4627	0.4815	0.4944	0.4803	0.5313	**0.5364**
	mak.	0.4918	0.4938	0.4675	0.4929	0.5093	0.4933	0.5422	**0.5452**
	tms.	63	65	66	56	56	60	34	32

Çizelge 4.6'daki yerel en iyi skorlar dikkate alındığında, ilk ağda (*A-1*) en yüksek modülerlik skoruna *SACO* ve *EPSO* hariç diğer altı algoritma ulaşmıştır. Bu ağda, "0" std. değerlerine sahip *3pHybrid*, *SSA* ve *AGA* algoritmalarının tümü genetik algoritma temellidir. Tablodaki sonuçlarda, genel skorlardaki gibi yine son iki ağda *3pHybrid* algoritmasının genel ortalama standart sapma sonuçları beklenenin altındadır. Ancak ağlardaki genel sonuçlar açısından bu algoritmanın yerel en iyilerde de en başarılı yöntem olduğu anlaşılmaktadır. Bu algoritmadan sonra genel anlamda uygun sonuçlara ulaşan algoritmalar *SSA*, *ISFLA*, *AGA* ve *GSA* algoritmalarıdır. Bunlardan ilk üçü ağ modül tespiti problemine kolayca uyarlanmış ayrık mekanizmalara sahiplerdir. Ancak bunlardan farklı olarak çalışma mekanizması gereği *GSA*, sürekli problemler için çözüm üreten bir mekanizmaya sahip olduğu için buradaki problem için diğerlerinden

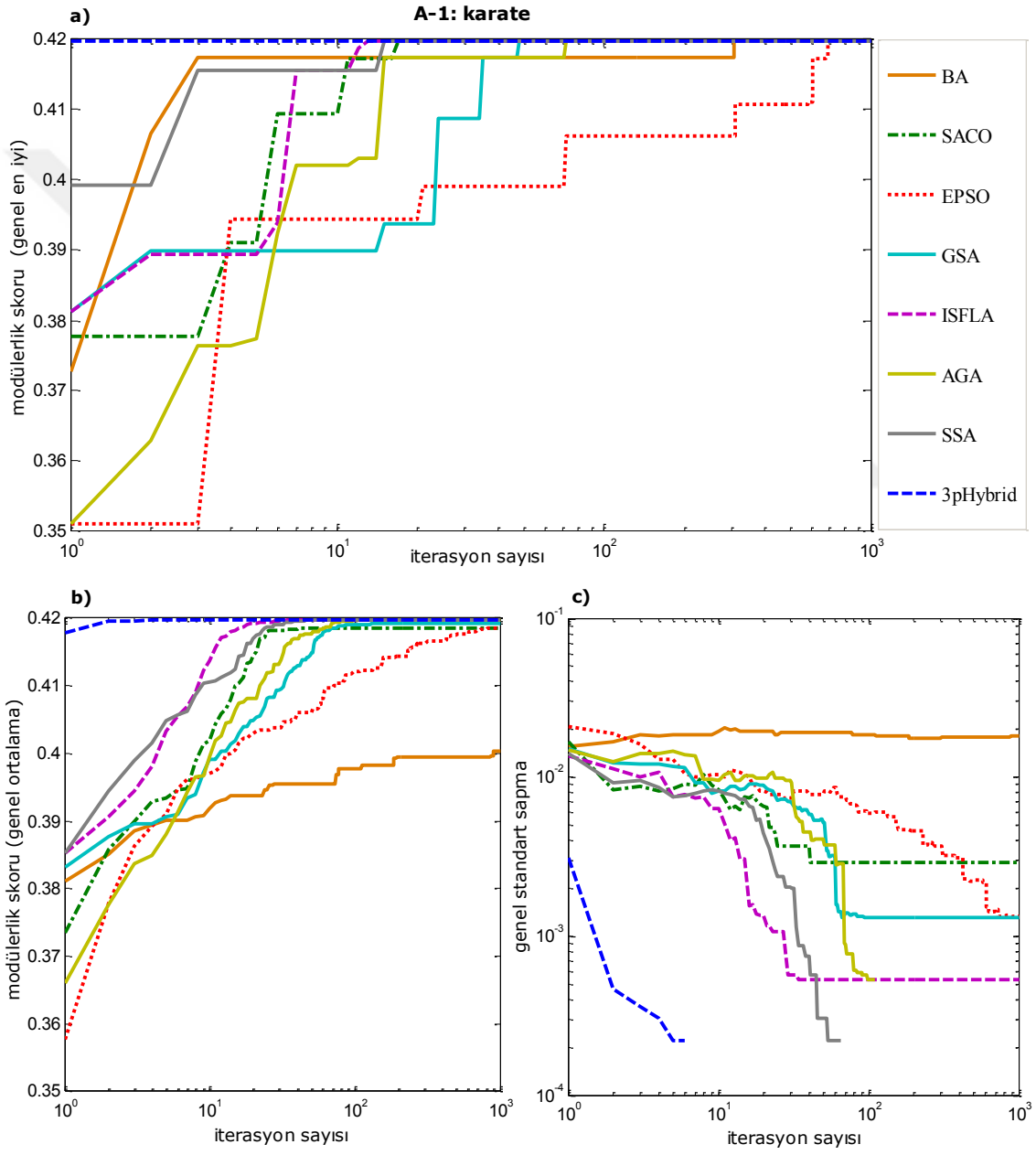
daha fazla rastsallığa sahiptir. Gerçek ağlardaki test sonuçlarına göre *SACO* ve *EPSO* algoritmaları genellikle uygun olmayan ağ modül yapılarını önermişlerdir. Bunun temel sebebi bu algoritmaların probleme uyarlanan yapılarının rastsallılıktan daha çok belli prensiplere dayanmasıdır. Örneğin, *SACO* algoritması feromen güncellenmesi mantığıyla uygun çözümü aramaktadır. Ancak buradaki test sonuçlarına göre bu tür bir arama işlemi yerine rastsal olarak çözüm aranması daha uygun skorlara yaklaştırmıştır. Feromen güncellemesi yapılarak uygun ağ modül yapılarının tespit edilebilmesi için algoritmanın daha fazla iyileştirilmesi gerekmektedir. Örneğin arama uzayının daha uygun şekilde taranabilmesi için *3pHybrid* algoritmasında probleme özgü geliştirilen seçim ve mutasyon operatörlerinden yararlanılabilir. Buna ek olarak, *SACO* algoritması için de buna benzer bir işlem gerçekleştirilebilir.

Çizelge 4.6. Küçük veya orta büyüklükteki ağların yerel test sonuçları

		BA	SACO	EPSO	GSA	ISFLA	AGA	SSA	3pHybrid
A-1	*ort.*	0.3997	0.3296	0.3703	0.4144	0.4167	**0.4198**	**0.4198**	**0.4198**
	std.	0.01786	0.02481	0.01385	0.007391	0.004011	**0**	**0**	**0**
	min.	0.3589	0.2973	0.3474	0.4018	0.4021	**0.4198**	**0.4198**	**0.4198**
	mak.	**0.4198**	0.3789	0.3944	**0.4198**	**0.4198**	**0.4198**	**0.4198**	**0.4198**
A-2	*ort.*	0.4855	0.3414	0.4186	0.5221	0.5201	0.5206	0.5261	**0.5285**
	std.	0.02452	0.01841	0.01415	0.005468	0.006972	0.004172	0.001555	**0**
	min.	0.4407	0.3076	0.3956	0.5108	0.5109	0.5001	0.5099	**0.5285**
	mak.	0.5208	0.3768	0.4429	0.5277	0.528	0.5283	0.5284	**0.5285**
A-3	*ort.*	0.5091	0.2156	0.3552	0.579	0.5988	0.5669	0.579	**0.6044**
	std.	0.04177	0.01517	0.02177	0.01894	0.000401	0.00204	0.003446	**0.000054**
	min.	0.4082	0.1892	0.3154	0.523	0.5872	0.5503	0.5521	**0.6043**
	mak.	0.5652	0.2419	0.3984	0.5984	0.6021	0.5899	0.6001	**0.6045**
A-4	*ort.*	0.5041	0.2129	0.4307	0.5183	0.5258	0.5262	0.526	**0.5272**
	std.	0.0125	0.01245	0.01925	0.007756	0.006107	0.000014	0.007221	**0**
	min.	0.4855	0.1896	0.4017	0.5023	0.5137	0.5259	0.5201	**0.5272**
	mak.	0.5262	0.245	0.4776	0.527	0.527	**0.5272**	0.5271	**0.5272**
A-5	*ort.*	0.7652	0.764	0.7105	0.7769	0.7803	0.7688	0.7835	**0.7845**
	std.	0.006275	0.007953	0.009975	0.003765	**0.00012**	0.006101	0.001429	0.001341
	min.	0.7551	0.75	0.6941	0.7702	0.7775	0.762	0.7816	**0.7817**
	mak.	0.7723	0.7725	0.7237	0.7823	0.7831	0.7723	0.7841	**0.7872**
A-6	*ort.*	0.4772	0.476	0.4474	0.4872	0.4998	0.4712	0.5303	**0.5429**
	std.	0.005521	0.00988	0.003379	0.003688	0.021099	0.021137	**0.00301**	0.005022
	min.	0.4701	0.4607	0.4424	0.4815	0.4701	0.4697	0.5285	**0.5363**
	mak.	0.4868	0.4889	0.4509	0.4929	0.5082	0.489	0.5401	**0.545**

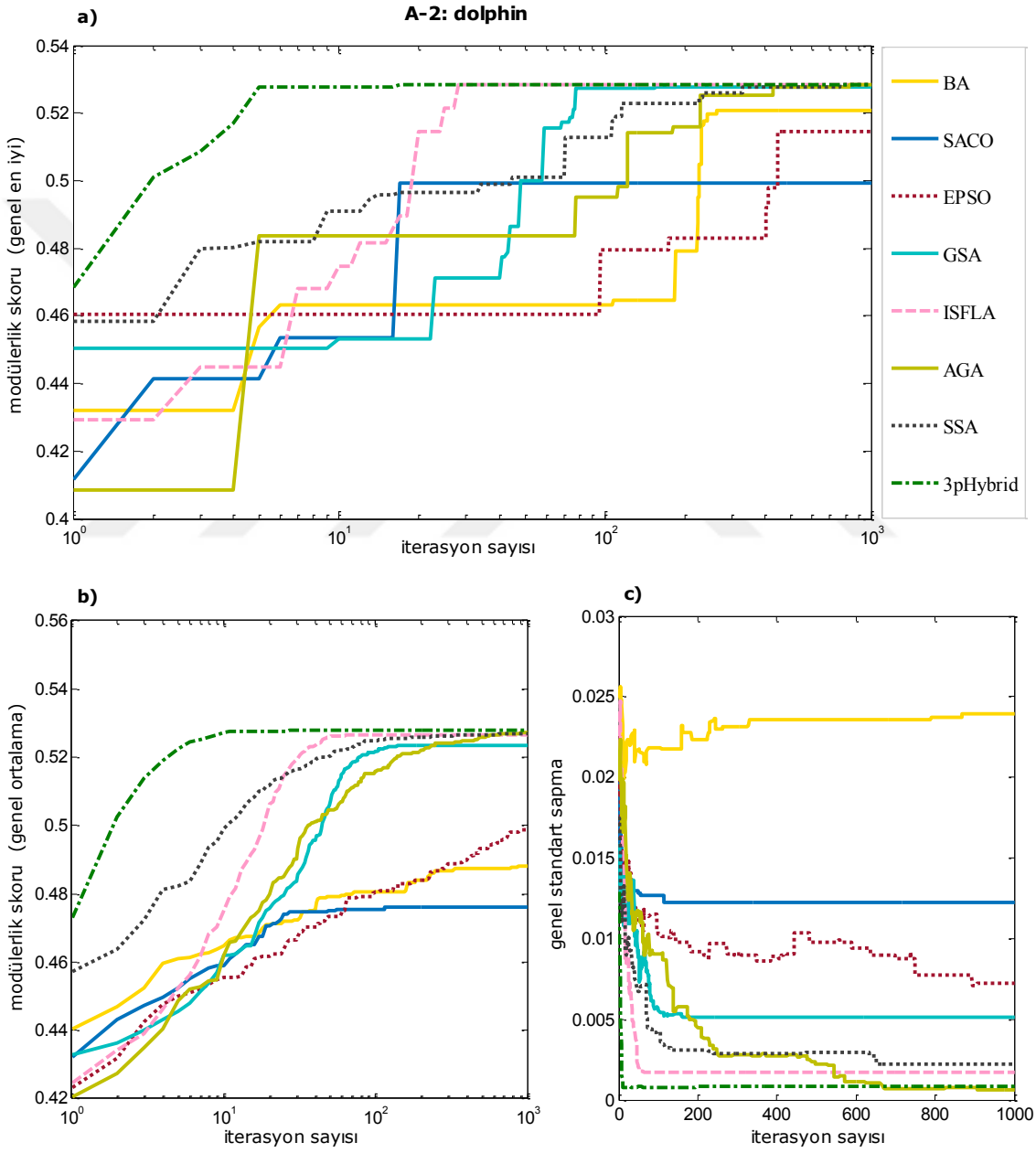
Küçük ve orta büyüklükteki gerçek dünya ağlarındaki her bir iterasyona göre elde edilen genel sonuçlar aşağıdaki grafiklerde verilmiştir. Burada genel maksimum ve

ortalama F_M skorları ile elde edilen standart sapma sonuçları temel alınarak aşağıdaki grafikler oluşturulmuştur. Şekil 4.9 ile Şekil 4.14 arasında A-1'de başlayıp A-6'ya kadar tüm ağlardaki iterasyonlara göre algoritmaların sonuçları verilmiştir. Her bir grafikte, bu algoritmalar farklı renklerde ve biçimlerde sunulan çizgiler gösterilmiştir. Algoritmaların farklı gerçek ağlardaki sonuçlarının iterasyonlara göre karşılaştırılması amacıyla aşağıdaki şekiller incelendiğinde, *3pHybrid* algoritmasının çoğu test ağında ilk iterasyonlarda hızlı bir yakınsamayla genel en iyi sonuçlara ulaştığı gözlemlenmiştir.

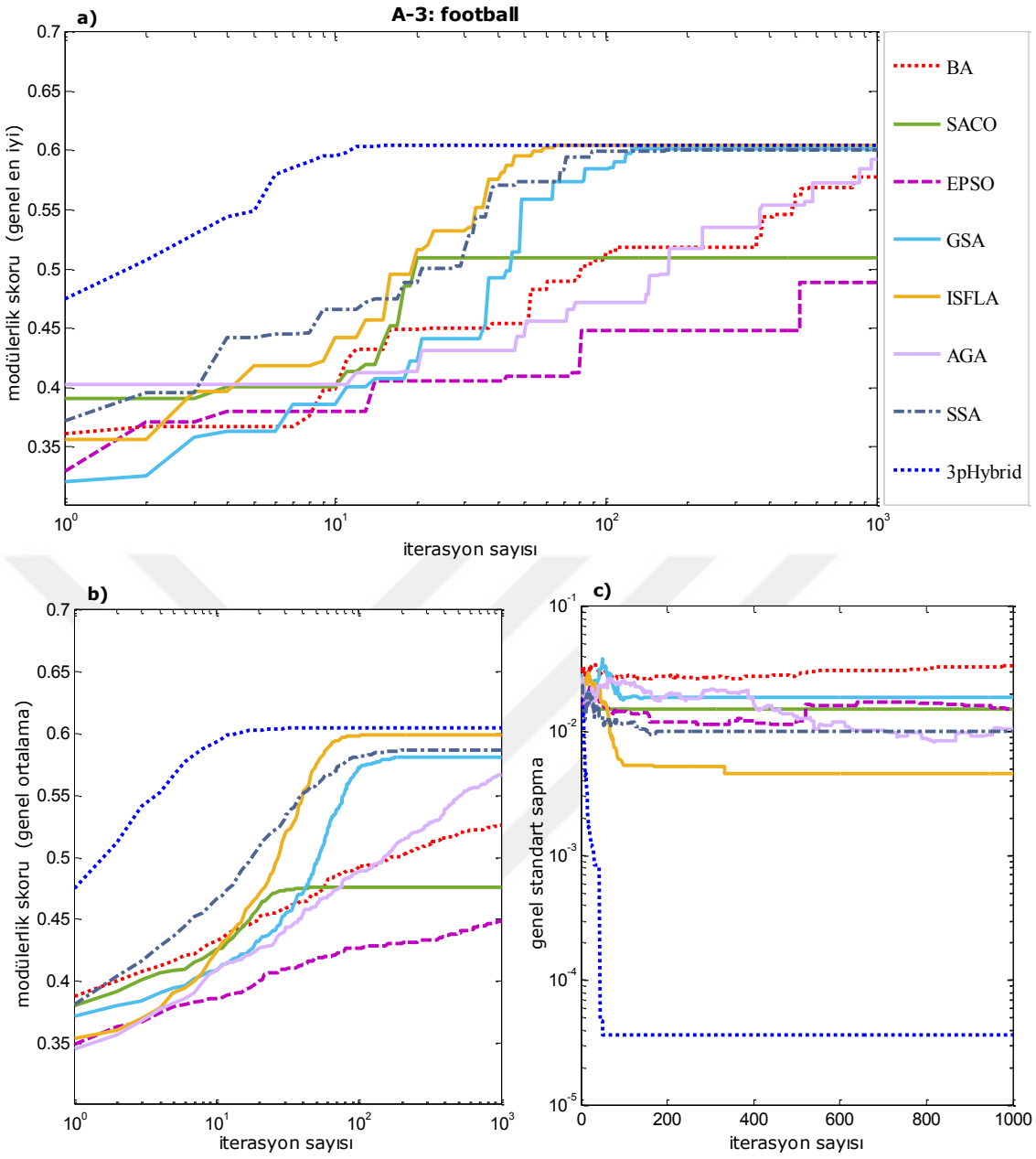


Şekil 4.9. A-1 ağındaki test sonuçlarına göre her iterasyon için **a)** genel en iyi F_M skorları, **b)** genel ortalama F_M skorları, **c)** genel standart sapma değerleri

Şekil 4.9'daki *karate* ağındaki ve Şekil 4.10'daki *dolphin* ağındaki sonuçlara bakıldığında, *3pHybrid* algoritmasının ilk iterasyonlarda en uygun sonucu elde ettiği görülmektedir. Bu iki ağdaki sonuçlarda, *std.* değerleri tüm çalıştırmalar sonunda sıfır olarak hesaplanmıştır. En kötü *std.* değerleri ise *BA* ile kaydedilmiştir. Bu iki ağdaki düğüm sayılarının az olması ve bağlantı karmaşıklıklarının düşük olması sebebiyle çoğu metasezgisel algoritma bu ağlarda modül yapılarını başarılı bir şekilde tespit etmiştir.

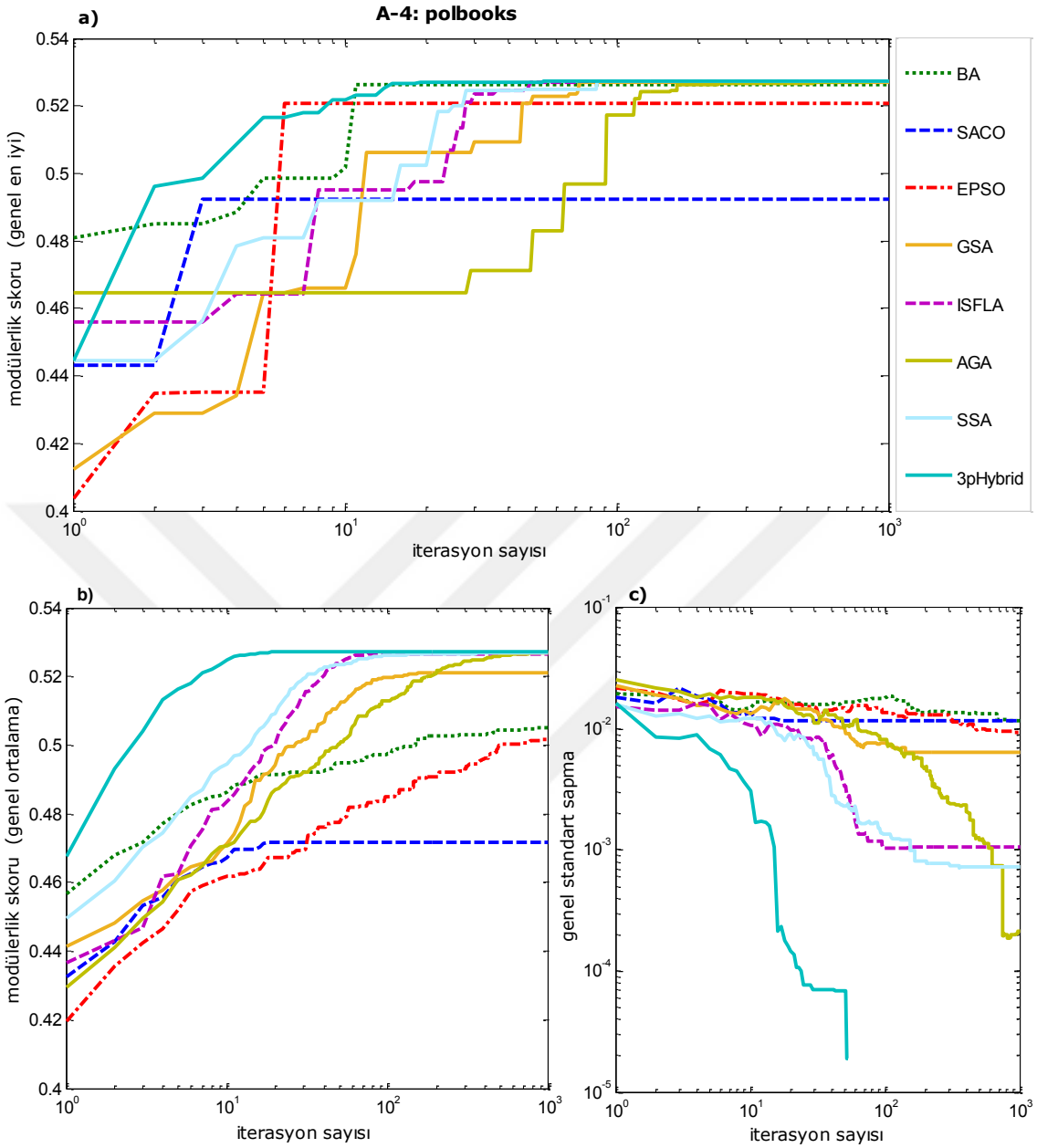


Şekil 4.10. A-2 ağındaki deneyler için her bir iterasyona göre **a)** genel en iyi F_M skorları, **b)** genel ortalama değerler, **c)** genel standart sapma değerleri



Şekil 4.11. A-3 (*football*) gerçek dünya ağındaki karşılaştırmalı sonuçlar; **a)** genel en iyi skorlar, **b)** genel ortalama skorlar, **c)** genel standart sapma değerleri

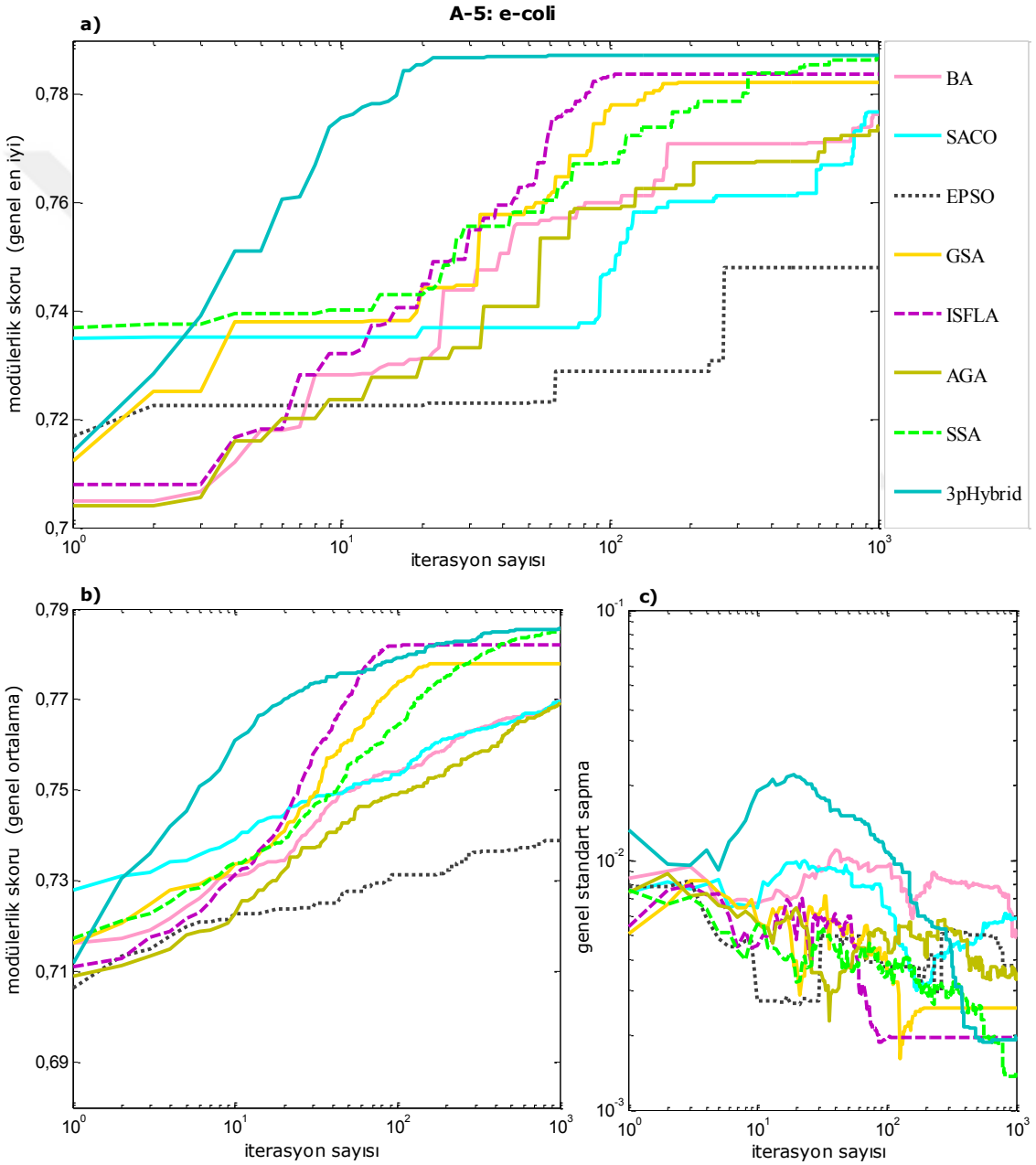
football ve *polbooks* gerçek dünya ağlarındaki test sonuçları Şekil 4.11 ve Şekil 4.12'de sunulmuştur. Burada, *football* ağı A-3 ile gösterilmiştir. Bu ağdaki *std.* değerlerine göre *3pHybrid* algoritması belirgin bir farkla en uygun sonuçları vermiştir. Diğer kriterler için de bu algoritma en başarılı sonuçlara ulaşmıştır. Sonraki en uygun sonuçlar *ISFLA* ve *SSA* ile elde edilmiştir. A-4 ile temsil edilen *polbooks* ağındaki en uygun skora 0.5272 olarak *3pHybrid* ile ulaşılmıştır. *AGA*, *SSA* ve *ISFLA* algoritmaları da bu ağdaki diğer başarılı yöntemlerdendir. Kesikli problemler için genellikle uygun çözümler sunan algoritmalar bu ağlarda da başarılı kümeleme sonuçlarına ulaşmıştır.



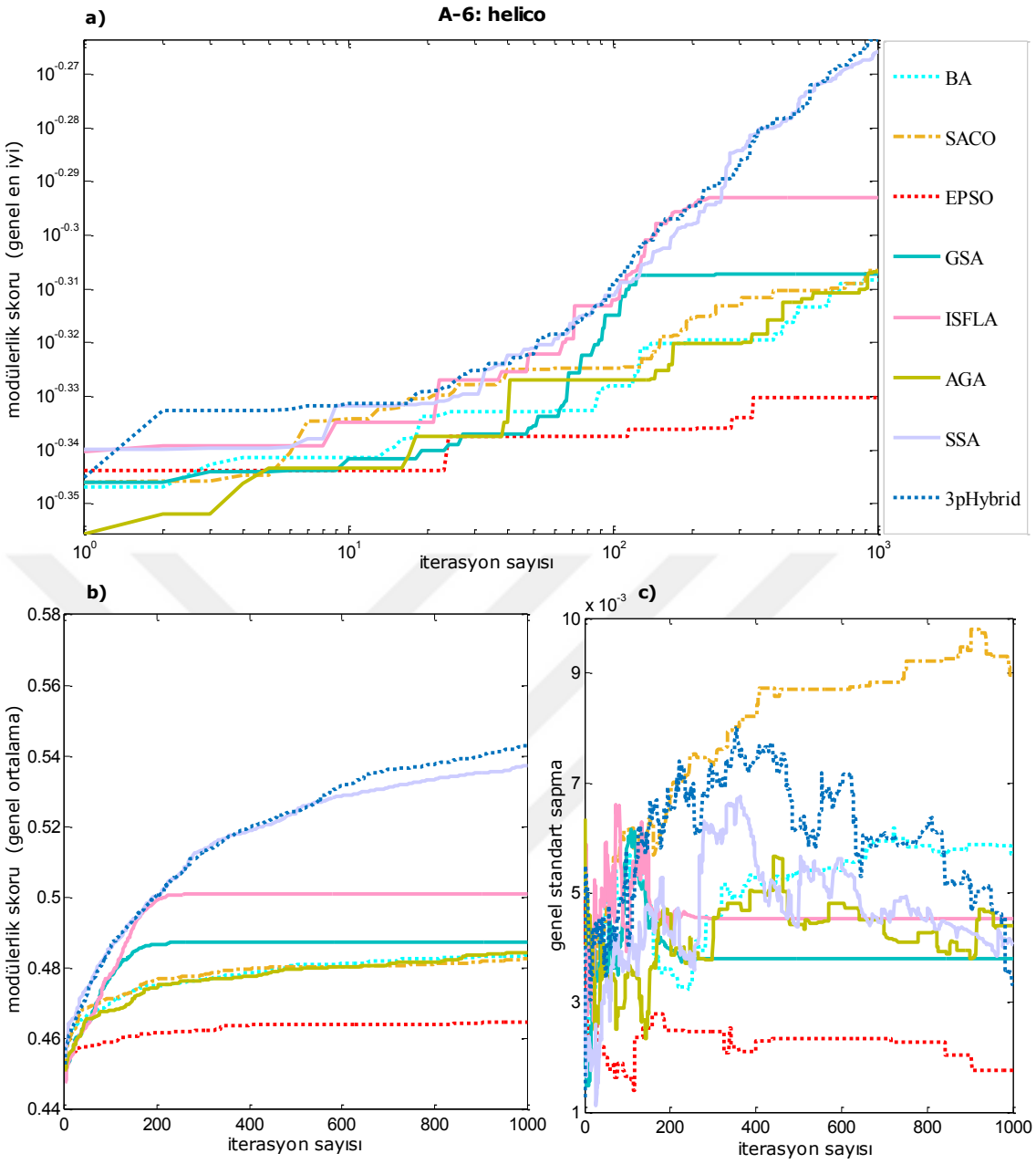
Şekil 4.12. A-4 ağı üzerinde gerçekleştirilen testlerin F_M sonuçları için **a)** genel en iyi skorlar, **b)** genel ortalama skorlar, **c)** genel standart sapma değerleri

Son olarak, gerçek dünya ağlarında karşılaştırmalı testlerin gerçekleştirildiği ağlar; *e-coli* ve *helico* isimleri ile temsil edilen ve sırasıyla; Şekil 4.12'de A-5 ve Şekil 4.13'te A-6 kısaltmaları ile gösterilen biyolojik ağlardır. Bu ağlardaki düğüm ve bağlantı sayıları önceki ağlardan daha fazladır. Bu sebeple değerlendirme açısından daha anlamlı sonuçlar bu testlerde elde edilmiştir. A-4'teki testler için *ort.*, *std.*, *mak.* ve *min.* değerlerine göre yerelde 0.7845, 0.001341, 0.7817 ve 0.7872; genelde 0.7856, 0.001966, 0.7818 ve 0.7873 F_M skorları ile tüm ağ, 45 *tms.* değeri ile farklı gruplara

ayrılmıştır. *helico* ağı için en uygun skora göre elde edilen *tms.* sayısı 32'dir. Bu ağ için en uygun *global—mak.*, *min.* ve *ort.* değerleri *3pHybrid* ile elde edilirken; en düşük standart sapma değeri 0.001759'dur ve bu sonuç *EPSO* ile kaydedilmiştir. A-5'teki testlerde önceki ağlarda elde edilen sonuçların benzerlerine ulaşılmıştır. Ancak A-6'daki testlerden *3pHybrid* ile *SSA* algoritmalarının sonuçlarının birbirlerine oldukça yakın olduğu anlaşılmıştır. Önceki testlerde genellikle diğer algoritmalara kıyasla, *3pHybrid* ile en uygun gruplamaların yapıldığı belirgin bir şekilde fark edilmektedir.



Şekil 4.13. A-5 ağında yapılan testlerdeki sonuçların iterasyonlara göre gösterimi; **a)** genel en iyi skorlar, **b)** genel ortalama skorlar, **c)** genel standart sapma değerleri



Şekil 4.14. Tüm algoritmaların A-6 ağındaki (*helico*) karşılaştırmalı test sonuçları; **a)** genel en iyi F_M değerleri, **b)** genel ortalama F_M değerleri, **c)** genel standart sapma değerleri

Gerçek dünya ağlarının sonuçları için verilen tüm karşılaştırma grafikleri birlikte incelendiğinde, genel olarak algoritmaların elde edebilecekleri en yüksek F_M skorlarına maksimum iterasyon sayılarından önce ulaştıkları gözlemlenmiştir. Buna ek olarak, *3pHybrid* ile diğerlerinden daha önceki iterasyonlarda daha iyi sonuçlara ulaşılmıştır.

Algoritmaların test sonuçlarının karşılaştırılmasında daha nesnel ölçütlere göre değerlendirmelerin yapılabilmesi için tüm genel ve yerel sonuçların yanında bir sonraki bölümde modülerlik skorlarına göre yapılan istatistiksel analiz sonuçları verilmiştir.

4.1.1.3. İstatistiksel analiz

Bu bölümde, sekiz adet algoritma için önceki bölümlerde test verileri olarak kullanılan 14 farklı (4 ER, 4 BA, 6 gerçek-dünya) ağdaki sonuçlar için *Friedman* (Friedman 1937) ve *Wilcoxon rank-sum* (Hollander ve Wolfe 1999, Gibbons ve Chakraborti 2011) testlerinin sonuçları sunulmuştur. *Wilcoxon rank-sum* testi, *Mann–Whitney U* veya *Wilcoxon–Mann–Whitney* olarak da bilinen testlere eşdeğerdir. Burada, ilk önce *Friedman* testi ile algoritmaların başarılarına göre ortalama sıralamalar belirlenir. Daha sonra seçilen en başarılı algoritma ile diğer algoritmaların sonuç değerleri arasında anlamlı ve belirgin bir fark olup olmadığını anlamak için *Wilcoxon rank-sum* testi uygulanır. Bu test, iki algoritmanın sonuçları için uygulanır. Bunun için *Wilcoxon* testi ilk olarak belirlenen en iyi algoritma ile diğerleri için ayı ayrı uygulanır. Çizelge 4.7'de önerilen algoritmaların 14 farklı ağdaki tüm çalıştırmalar (30 kez) sonunda elde edilen en iyi F_M değerlerinin ortalamaları parantez içinde sunulurken; bu değerlere göre belirlenen başarı sıralamaları da parantez dışında verilmiştir. Burada, parantezler içindeki ortalama değerlerin ondalık kısımları üç hane ile gösterilmiştir. Tablodaki sayıların kesirli kısımları 0.5'e eşit veya daha büyükse yukarıya; değilse aşağıya yuvarlanmıştır. Parantezlerin önündeki sayılarla algoritmaların genel başarı sıralamaları gösterilmiştir.

Çizelge 4.7'nin son satırında koyu renklerle gösterilen (#) sayılar, algoritmaların tüm sonuçlar dikkate alınarak belirlenen nihai sıralamalarıdır. Bu kısımdaki parantezler içinde verilen ondalık değerler ise *Friedman* testi dikkate alınarak belirlenen ortalama başarı sıralamalarıdır. Sonuçlara göre tüm algoritmalar 1'den 8'e doğru sıralanmışlardır. Örneğin nihai başarı sırası 1 olan *3pHybrid* algoritması için son satırda, "*1—(1.071)*" bilgisi verilmiştir. Bunun anlamı test sonucuna göre *1.071* ortalama başarı sırasına sahip algoritmanın nihai (*en son*) sıralaması, 1'dir. Böylece tüm algoritmalar en yüksekten en düşüğe doğru şöyle sıralanır: 3pHybrid, SSA, ISFLA, GSA, AGA, BA, EPSO, SACO.

Çizelge 4.7. Tüm algoritmaların başarı sıralamalarının *Friedman* testi ile elde edilmesi

	BA	**SACO**	**EPSO**	**GSA**	**ISFLA**	**AGA**	**SSA**	**3pHybrid**
ER-1	5—(0.312)	8—(0.275)	7—(0.280)	4—(0.324)	3—(0.345)	6—(0.310)	2—(0.349)	1—(0.364)
ER-2	5—(0.374)	8—(0.336)	7—(0.346)	4—(0.395)	3—(0.416)	6—(0.365)	2—(0.438)	1—(0.442)
ER-3	5—(0.309)	8—(0.286)	7—(0.293)	4—(0.335)	2—(0.353)	6—(0.300)	3—(0.348)	1—(0.388)
ER-4	5—(0.288)	8—(0.276)	7—(0.280)	4—(0.295)	3—(0.308)	6—(0.285)	2—(0.312)	1—(0.363)
BA-1	5—(0.335)	7—(0.300)	8—(0.299)	4—(0.355)	3—(0.369)	6—(0.330)	2—(0.369)	1—(0.384)
BA-2	5—(0.326)	8—(0.291)	7—(0.300)	4—(0.357)	3—(0.374)	6—(0.319)	2—(0.392)	1—(0.408)
BA-3	5—(0.319)	8—(0.295)	7—(0.301)	4—(0.347)	3—(0.367)	6—(0.311)	2—(0.400)	1—(0.402)
BA-4	5—(0.308)	8—(0.295)	7—(0.299)	4—(0.315)	3—(0.333)	6—(0.305)	2—(0.364)	1—(0.384)
A-1	8—(0.400)	6—(0.419)	6—(0.419)	5—(0.419)	4—(0.420)	1—(0.420)	1—(0.420)	1—(0.420)
A-2	7—(0.488)	8—(0.476)	6—(0.499)	5—(0.524)	4—(0.527)	2—(0.527)	3—(0.527)	1—(0.529)
A-3	6—(0.526)	7—(0.475)	8—(0.449)	4—(0.581)	2—(0.599)	5—(0.567)	3—(0.587)	1—(0.605)
A-4	6—(0.505)	8—(0.472)	7—(0.502)	5—(0.521)	4—(0.527)	2—(0.527)	3—(0.527)	1—(0.527)
A-5	5—(0.770)	6—(0.770)	8—(0.739)	4—(0.778)	3—(0.782)	7—(0.769)	2—(0.785)	1—(0.786)
A-6	6—(0.484)	7—(0.483)	8—(0.465)	4—(0.487)	3—(0.501)	5—(0.485)	2—(0.537)	1—(0.543)
#	**6—(5.571)**	**8—(7.536)**	**7—(7.179)**	**4—(4.214)**	**3—(3.071)**	**5—(5.071)**	**2—(2.286)**	**1—(1.071)**

[#] *Ki-kare testi = 85.8711, df = 7, p-değeri = 8.6941e-16 (sıfır hipotezinin reddi)*

Yukarıdaki *Friedman* testi sonucuna göre en başarılı sıralamaya sahip olan *3pHybrid* algoritmasının diğer algoritmalarla *Wilcoxon* istatistiksel anlamlılık testine göre karşılaştırılması Çizelge 4.8'de sunulmuştur. Bu istatistiksel test için eşik p değeri (p *value*) 0.01 olarak alınmıştır. Bu değerden yüksek sonuçlar için anlamlılık "✓" işareti ile gösterilmiştir. Buradaki eşik değer, sonuçlar arasındaki istatistiksel anlamlılık farkının %99 (*yüzde doksan dokuz*) olasılıklı güven aralığında olduğunu gösterir. Yani sonuçlar %1 anlamlılık düzeyinde değerlendirilmiştir. Burada 4 adet *ER* türü ağ, 4 adet *BA* türü ağ ve 6 adet gerçek dünya ağının sonuçlarına göre istatistiksel anlamlılık testi yapılmıştır. Bu istatistiksel test, tüm elde edilen deneysel sonuçlara göre *3pHybrid* algoritmasının diğer algoritmalardan (*BA*, *SACO*, *EPSO*, *GSA*, *ISFLA*, *AGA*, *SSA*) istatistiksel olarak anlamlı bir şekilde farklı sonuçlara ulaştığını göstermiştir. Böylece *3pHybrid* algoritmasının diğer algoritmalardan farklı sonuçlara ulaşan özgün bir yöntem olduğu anlaşılmıştır. Ayrıca seçilen algoritmaların sonuçları arasında istatistiksel olarak bir fark olmaması ($p \geq 0.01$) durumu ise Çizelge 4.8'de "–" işareti ile temsil edilmiştir. Çizelgedeki sonuçlara göre *3pHybrid* algoritması ile *BA* ve *SACO* algoritmalarının sonuçlarının birbirlerinden tamamen farklı olduğu anlaşılmaktadır. Yine en iyi algoritmanın sonuçlarının *EPSO* ve *GSA* algoritmalarının sonuçlarında sadece *A*-1

verisindeki test sonuçları hariç diğerlerinde istatistiksel olarak anlamlı bir şekilde farklı olduğu gözlemlenmiştir. Bu algoritmanın *ISFLA* ile 3 ve *AGA* ile 2 ağ verisi üzerindeki test sonuçlarında anlamlı farklılık gözlenmemiştir. *SSA* ile *3pHybrid* algoritmalarının sonuçları kıyaslandığında, test verilerinin yarısından fazlasında anlamlı bir farklılık olduğu görülmektedir. Böylece *3pHybrid* algoritmasının hem başarı sıralamaları testine göre en iyi algoritma olduğu belirlenmiş hem de bu algoritmanın test sonuçları ile diğer algoritmaların sonuçları arasında genellikle %99 olasılıklı güven aralığında istatistiksel olarak anlamlı farklılıkların olduğu sonucuna varılmıştır.

Çizelge 4.8. *Wilcoxon* istatistiksel anlamlılık testi sonuçları

3pHybrid	***BA***	***SACO***	***EPSO***	***GSA***	***ISFLA***	***AGA***	***SSA***
vs	*p (s)*	*p (s)*	*p (s)*	*p (s)*	*p (s)*	*p (s)*	*p (s)*
ER-1	6.80E-4 (✓)	6.80E-4 (✓)	6.80E-4 (✓)	6.80E-4 (✓)	1.43E-3 (✓)	6.80E-4 (✓)	1.92E-3 (✓)
ER-2	6.71E-4 (✓)	6.71E-4 (✓)	6.71E-4 (✓)	6.71E-4 (✓)	6.71E-4 (✓)	6.71E-4 (✓)	3.60E-2 (–)
ER-3	6.63E-4 (✓)	6.54E-4 (✓)	6.63E-4 (✓)	6.63E-4 (✓)	6.63E-4 (✓)	6.63E-4 (✓)	6.47E-4 (✓)
ER-4	5.65E-5 (✓)	5.65E-5 (✓)	5.65E-5 (✓)	5.99E-4 (✓)	5.95E-4 (✓)	5.99E-4 (✓)	5.65E-5 (✓)
BA-1	6.80E-4 (✓)	6.80E-4 (✓)	6.80E-4 (✓)	7.90E-5 (✓)	1.43E-3 (✓)	6.80E-4 (✓)	1.43E-3 (✓)
BA-2	6.44E-4 (✓)	6.44E-4 (✓)	6.44E-4 (✓)	6.44E-4 (✓)	8.58E-4 (✓)	6.44E-4 (✓)	3.94E-2 (–)
BA-3	6.49E-4 (✓)	6.49E-4 (✓)	6.49E-4 (✓)	6.49E-4 (✓)	4.13E-2 (–)	6.49E-4 (✓)	1.90E-1 (–)
BA-4	6.03E-4 (✓)	6.03E-4 (✓)	6.03E-4 (✓)	6.03E-4 (✓)	5.97E-4 (✓)	6.01E-4 (✓)	5.95E-4 (✓)
A-1	3.49E-3 (✓)	2.07E-3 (✓)	2.36E-1 (–)	1.98E-2 (–)	3.42E-1 (–)	1.00E+0 (–)	1.00E+0 (–)
A-2	3.79E-5 (✓)	3.79E-5 (✓)	3.78E-4 (✓)	1.69E-4 (✓)	2.69E-4 (✓)	3.41E-3 (✓)	1.05E-1 (–)
A-3	1.96E-4 (✓)	1.96E-4 (✓)	1.96E-4 (✓)	1.75E-3 (✓)	2.97E-3 (✓)	2.66E-1 (–)	3.28E-3 (✓)
A-4	8.01E-5 (✓)	8.01E-5 (✓)	8.01E-5 (✓)	7.89E-5 (✓)	6.67E-1 (–)	3.39E-3 (✓)	6.65E-1 (–)
A-5	5.45E-5 (✓)	5.45E-5 (✓)	5.45E-5 (✓)	5.45E-5 (✓)	5.45E-5 (✓)	5.45E-5 (✓)	9.57E-3 (✓)
A-6	5.89E-4 (✓)	5.87E-4 (✓)	5.89E-4 (✓)	5.91E-4 (✓)	5.81E-8 (✓)	5.72E-4 (✓)	4.81E-3 (✓)

Son olarak, önceki istatistiksel analiz sonuçlarına ve yapay ağlar ile küçük ve orta boyutlu ağlar üzerindeki test sonuçlarına göre Bölüm 3.1.1.2'de verilen büyük ve karmaşık ağlar, en başarılı sonuçlara ulaşan *3pHybrid* algoritmasının daha karmaşık ağlardaki başarısının ve performansının test edilebilmesi amacıyla kullanılmıştır. Bu testlerde algoritmanın uygunluk fonksiyonu olarak modülerlik amaç fonksiyonu seçilmiştir. Çünkü bu amaç fonksiyonu ile çeşitli çevrimiçi karmaşık ağlarda modül tespitinin daha öncesinden yapıldığı bilinmektedir. Bu algoritmanın mevcut testlere ek olarak daha büyük ve daha karmaşık ağ yapılarındaki performansı test edilmiş ve sonuçlar burada verilmiştir. Bunun için Bölüm 3.1.1.2'de sunulmuş olan Facebook,

Twitter ve Google+ ego-sosyal-etkileşim ağları kullanılmıştır. Bunlarla hakkında genel bilgilere ilgili bölümde ulaşılabilir. Bu karmaşık çevrimiçi sosyal ağ çalışmalarında önerdiğimiz *3pHybrid* algoritmasının kaydedilen en yüksek modülerlik skorları sırasıyla; Facebook, Twitter ve Google+ ego ağları için 0.76, 0.69 ve 0.66'tir. Buradaki sonuçların kesirli kısımlar çift hane olarak sunulmuştur. Bu sonuçlar önerilen algoritmanın düğüm ve bağlantı yoğunluğu oldukça fazla olan ağlarda bile makul sonuçlara ulaştığını göstermektedir. Çünkü bu skorunun 0.3'ten fazla olması elde edilen ağ modüllerinin genellikle anlamlı olabileceğini göstermektedir.

4.1.2. Amaç fonksiyonlarının değerlendirme ölçütlerine göre ayrıntılı analizi

Bu bölümde, kesin referanslı topluluklara sahip gerçek dünya ağlarındaki deneysel çalışmaların sonuçları ayrıntılı olarak analiz edilmiştir. Burada, amaç (*kalite*) fonksiyonlarından hangisinin veya hangilerinin genellikle daha uygun alt-ağ yapılarını tespit etmede başarılı olduklarının anlaşılabilmesi için farklı küme değerlendirme fonksiyonları kullanılmıştır. Bahsi geçen problemin tanımı ve test verileri sırasıyla; Bölüm 2.1.4'de ve Bölüm 3.1.2'de sunulmuştur.

Deneylerde kullanılan algoritma ise Bölüm 4.1.1.1'deki ve Bölüm 4.1.1.2'deki karşılaştırmalı test sonuçlarına ve Bölüm 4.1.1.3'teki istatistiksel analizlere göre belirlenmiştir. Tüm bu sonuçlara göre en başarılı metasezgisel algoritmanın *3pHybrid* olduğu tespit edilmiştir. Bu sebeplerden dolayı aşağıdaki deneylerde bu algoritmanın kullanılmasına karar verilmiştir. 10 adet amaç fonksiyonunu optimize eden *3pHybrid* algoritmasının sonuçları Çizelge 4.9 ile Çizelge 4.16 arasında verilmiştir. Bu sonuçlar *Strike*, *Amlall*, *Duke*, *Khan*, *Cancer*, *Ecoli*, *Ionosphere* ve *Yeast* gerçek dünya ağlarına aittir. Bu ağlar Çizelge 3.2'deki (Bölüm 3.1.2'ye bakınız) sırasına göre *B1*, *B2*, *B3*, *B4*, *B5*, *B6*, *B7* ve *B-8* ile temsil edilirler. Her bir ağa göre tablolarda gösterilen sonuçlar algoritmanın 30 kez çalıştırılması sonrası kaydedilen değerlerin ortalamasını gösterir. Çizelgelerde 6 adet küme değerlendirme ölçütü ile hesaplanan en iyi (*yüksek*), en kötü (*düşük*), ortalama ve standart sapma değerleri sırasıyla; $'mak.'$, $'min.'$ $ort.'$, $'std.'$ ile temsil edilmiştir. Bunlara ek olarak, çizelgelerdeki değerlerden en uygun olanları koyu renklerle vurgulanmıştır. Tüm testler sonunda (algoritmaların 30 kez çalıştırılması sonrası) her bir amaç fonksiyonuna göre belirlenen en iyiler; mak., min. ve ort. kriterleri için en yüksek sayıları; *std.* kriteri için en düşük sayıları temsil eder. Bu tablolarda F_M,

F_C, F_{ID}, F_{GD}, F_{CR}, F_{NC}, F_{FS}, F_{ODF}, F_S ve F_{Sr} amaç fonksiyonlarının kullanılmasıyla elde edilen modüllerin uygunlukları E_{MI}, E_{NMI}, E_{RI}, E_{ARI}, E_{JI} ve E_P küme değerlendirme ölçütleri ile test edilmiştir. Hem amaç hem de küme değerlendirme fonksiyonları hakkında ayrıntılı bilgilere Bölüm 2.1.3'te ve Bölüm 2.1.4'te ulaşılabilir. Ayrıca, parametrik olmayan istatistiksel bir prosedür sunan *Friedman testi* (Friedman 1937) ile amaç fonksiyonlarının küme değerlendirme fonksiyonlarına göre elde edilen sonuçları aşağıda ayrıntıları olarak sunulmuştur.

Çizelge 4.9. *B-1* ağı için amaç fonksiyonlarının değerlendirme sonuçları

		E_{MI}	E_{NMI}	E_{RI}	E_{ARI}	E_{JI}	E_P
F_M	*mak.*	**1.4773345**	**0.8840699**	**0.9130435**	**0.7977899**	**0.7525773**	0.4929563
	min.	**1.4773345**	**0.8840699**	**0.9130435**	**0.7977899**	**0.7525773**	0.4929563
	ort.	**1.4773345**	**0.8840699**	**0.9130435**	**0.7977899**	**0.7525773**	0.4929563
	std.	**0**	**0**	1.17E-16	1.17E-16	**0**	**0**
F_C	*mak.*	0.6500224	0.611108	0.6413043	0.3622444	0.494898	0.4905754
	min.	0.6500224	0.611108	0.6413043	0.3622444	0.494898	0.4905754
	ort.	0.6500224	0.611108	0.6413043	0.3622444	0.494898	0.4905754
	std.	1.17E-16	**0**	**0**	**0**	5.851E-17	5.851E-17
F_{ID}	*mak.*	**1.4773345**	0.6268159	0.7246377	0.263845	0.2164948	0.4272817
	min.	1.3625475	0.5671217	0.6956522	0.1720591	0.1428571	0.2033234
	ort.	1.4325224	0.5966943	0.7094203	0.2161099	0.1784383	0.2946429
	std.	0.0481579	0.0298844	0.013427	0.0421283	0.033437	0.0902303
F_{GD}	*mak.*	**1.4773345**	0.6629414	0.7391304	0.3105267	0.257732	**0.5773313**
	min.	1.3940011	0.6083724	0.7246377	0.2680578	0.2244898	**0.566369**
	ort.	1.4106678	0.6192862	0.7275362	0.2765516	0.2311382	**0.5751389**
	std.	0.0351364	0.0230083	0.0061107	0.0179065	0.0140161	0.0046221
F_{CR}	*mak.*	0.6500224	0.611108	0.6413043	0.3622444	0.494898	0.4905754
	min.	0.6500224	0.611108	0.6413043	0.3622444	0.494898	0.4905754
	ort.	0.6500224	0.611108	0.6413043	0.3622444	0.494898	0.4905754
	std.	1.17E-16	**0**	**0**	**0**	5.851E-17	5.851E-17
F_{NC}	*mak.*	1.0065486	0.5212662	0.6956522	0.2740936	0.3114754	0.3568948
	min.	0.5488587	0.320532	0.5543478	0.0200901	0.208	0.0707837
	ort.	0.8257148	0.4482408	0.6485507	0.1718671	0.2583859	0.2337698
	std.	0.124881	0.0557164	0.0377693	0.0689249	0.0305245	0.1034911
F_{FS}	*mak.*	**1.4773345**	0.6629414	0.7391304	0.3105267	0.257732	0.566369
	min.	**1.4773345**	0.6629414	0.7391304	0.3105267	0.257732	**0.566369**
	ort.	**1.4773345**	0.6629414	0.7391304	0.3105267	0.257732	0.566369
	std.	**0**	1.17E-16	1.17E-16	**0**	**0**	**0**
F_{ODF}	*mak.*	0.6500224	0.611108	0.6413043	0.3622444	0.494898	0.4905754
	min.	0.6500224	0.611108	0.6413043	0.3622444	0.494898	0.4905754
	ort.	0.6500224	0.611108	0.6413043	0.3622444	0.494898	0.4905754
	std.	1.17E-16	**0**	**0**	**0**	5.851E-17	5.851E-17
F_S	*mak.*	**1.4773345**	0.6629414	0.7391304	0.3105267	0.257732	0.566369
	min.	**1.4773345**	0.6629414	0.7391304	0.3105267	0.257732	**0.566369**
	ort.	**1.4773345**	0.6629414	0.7391304	0.3105267	0.257732	0.566369
	std.	**0**	1.17E-16	1.17E-16	**0**	**0**	**0**
F_{Sr}	*mak.*	**1.4773345**	0.6629414	0.7391304	0.3105267	0.257732	**0.5773313**
	min.	1.3940011	0.6083724	0.7246377	0.2680578	0.2244898	**0.566369**
	ort.	1.4690011	0.6574845	0.7376812	0.3062798	0.2544077	0.5674653
	std.	0.0263523	0.0172562	0.004583	0.0134298	0.0105121	0.0034666

Bu çalışmada *B-1* ile temsil edilen *Strike* ağı üzerinde gerçekleştirilen deneysel çalışma sonuçları Çizelge 4.9'da verilmiştir. Burada *3pHybrid* algoritmasında uygunluk fonksiyonları olarak seçilen 10 adet amaç fonksiyonunun (F_M, F_C, F_{ID}, F_{GD}, F_{CR}, F_{NC}, F_{FS}, F_{ODF}, F_S ve F_{Sr}) kullanılmasıyla elde edilen ağ modüllerinin kaliteleri—uygunlukları, *Strike* ağı için daha önceden tanımlanmış olan (Yang ve ark 2012 (b)) ve Bölüm 3.1.2'deki Çizelge 3.2'de sunulan ağ modüllerine/topluluklarına (3 adet) göre altı farklı küme değerlendirme ölçütü ile belirlenmiştir. Bu ölçütler ise Bölüm 2.1.4'te tanımlanan ve sırasıyla; E_{MI}, E_{NMI}, E_{RI}, E_{ARI}, E_{JI}, E_P ile temsil edilen fonksiyonlardır. Tabloda her bir amaç fonksiyonun değerlendirme ölçütlerine göre hesaplanan sonuçları verilmiştir. *Strike* ağının testlerinde F_M (*modülerlik*) fonksiyonu için E_{MI} (*karşılıklı bilgi*) değerleri *mak.*, *min.*, *ort.* ve *std.* için sırasıyla; 1.4773345, 1.4773345, 1.4773345 ve 0'dır. Bu değerler Çizelge 4.9'da koyu renklerle vurgulanmışlardır. *3pHybrid*'in uygunluk kriteri olarak modülerlik amaç fonksiyonunun seçilmesiyle çalıştırmaların tümünde $E_{MI} = 1.4773345$ sonucu elde edilmiştir. Böylece F_M için ortalama standart sapma değeri 0 olur. E_{MI} ölçütüne göre maksimum değere (*1.4773345*), F_M fonksiyonunun yanında F_{ID}, F_{GD}, F_{FS}, F_S ve F_{Sr} fonksiyonlarının da uygunluk fonksiyonu olarak seçilmesiyle de ulaşılmıştır. Bu ölçüte göre en uygun minimum, ortalama ve standart sapma değerleri ise F_M, F_{FS} ve F_S fonksiyonlarının kullanılmasıyla elde edilmiştir.

E_{NMI} (*normalize edilmiş karşılıklı bilgi*) küme değerlendirme fonksiyonuna göre *Strike* ağı üzerindeki deneylerde en uygun sonuçlar F_M fonksiyonu ile belirlenmiştir. Bunlar *mak.*, *min.* ve *ort.* kriterleri için aynı değer iken (*0.8840699*); *std.* için *0* değeridir. Ayrıca F_C, F_{CR} ve F_{ODF} fonksiyonları en uygun değerlendirme sonucuna ulaşamasalar da kendi sonuçlarına uygun olarak *0 std.* değerine sahiplerdir. Bu fonksiyonlarla elde edilen *mak.* sonuçları ise aynı değerdir (*0.611108*).

Bir diğer ölçüt ise E_{RI} (*rastgelelik indeksi*) fonksiyonudur ve burada *std.* değeri hariç yine F_M ile en yüksek değere (*0.9130435*) ulaşılmıştır. Bu fonksiyon için standart sapma değeri *"1.17E-16"* gibi oldukça düşük bir sayı iken; bir önceki ölçütteki gibi F_C, F_{CR} ve F_{ODF} fonksiyonlarıyla *'0' std.* değeri elde edilmiştir. Bunlar için E_{RI} sonucu *0.6413043*'tür.

E_{ARI} (*düzenlenmiş rastgelelik indeksi*) ölçütüne göre *B-1*'deki testlerde *mak.*, *min.* ve *ort.* değerleri için sadece F_M fonksiyonu en uygun değere (*0.7977899*)

ulaşmıştır. Bu fonksiyon için *std.* değeri ise E_{RI}'daki gibi $1.17E-16$'dir. E_{ARI} ölçütüne göre en düşük *std.* değeri F_C, F_{CR}, F_{FS}, F_{ODF} ve F_S ile elde edilmiştir.

E_{JI} (*jaccard indeksi*) değerlendirme fonksiyonu için *mak.*, *min.*, *ort.* ve *std.* değerlerine göre en uygun sonuçlar *'0.7525773'* ve *'0'* ile F_M'e aittir. Standart sapma sonuçları için aynı değeri F_{FS} ve F_S fonksiyonları da *'0.257732'* maksimum değerlendirme sonucuyla elde etmişlerdir. Son olarak, *Strike* ağı ile yapılan deneylerde en uygun sonuçları veren amaç fonksiyonunun belirlenmesi aşamasında E_P (*kalıcılık*) değerlendirme ölçütü kullanılarak F_{GD} ve F_{Sr} fonksiyonları *mak.* değer için *0.5773313* sonucunu vermişlerdir. Hesaplanan en yüksek minimum E_P değerleri (*0.566369*) F_{GD}, F_{FS}, F_S ve F_{Sr} fonksiyonlarına ait iken; en iyi (*yüksek*) ortalama sonuca (*0.5751389*) F_{GD} fonksiyonu ile erişilmiştir. Bunlara ek olarak, en düşük standart sapma değerlerini F_M, F_{FS} ve F_S fonksiyonları vermiştir.

B-2 ile ifade edilen *Amlall* gerçek dünya ağının bilinen en uygun topluluk sayısı Çizelge 3.2'de gösterildiği gibi 3'tür. Bu topluluklara göre algoritmada farklı uygunluk fonksiyonları kullanılarak elde edilen toplulukların/modüllerin kaliteleri test edilmiş ve Çizelge 4.10'daki sonuçlara ulaşılmıştır. Bu ağda E_{MI} değerlendirme fonksiyonuna göre tüm çalıştırmalar sonucu en yüksek *mak.* değeri (*1.4184734*) ile F_{GD} elde edilmiştir. En uygun *min.* ve *ort.* değerleri F_S ile kaydedilirken; en düşük *std.* değeri F_M ve F_{Sr} amaç fonksiyonlarıyla elde edilmiştir.

Bu ağdaki denemelerde E_{NMI} değerlendirme fonksiyonu için en yüksek *mak.*, *min.* ve *ort.* sayılarına (*0.706189*) F_{Sr} ile ulaşılmıştır. Bu değerlendirme fonksiyonuna göre F_C, F_{CR} ve F_{ODF} fonksiyonları için 30 kez çalıştırma sonunda 0 *std.* değeri hesaplanmıştır. Buna benzer bir şekilde yine F_{Sr} için E_{RI}, E_{ARI} ve E_{JI} değerlendirme fonksiyonlarına göre en uygun *mak.*, *min.* ve *ort.* değerleri elde edilmiştir. Tüm çalıştırmalar sonunda ortalamaları belirlenen standart sapma parametreleri için E_{RI} ile E_{JI} için F_M fonksiyonu ile; E_{ARI} için F_{Sr} fonksiyonu ile sıfır değerleri elde edilmiştir.

Çizelge 4.10'da gösterilen son küme değerlendirme ölçütü E_P'dir. Bu ölçüte göre tüm parametreler için en iyi değerlere F_C, F_{CR} ve F_{ODF} amaç fonksiyonları ile ulaşılmıştır. Bu fonksiyonlar için ortalama standart sapma sayıları 0'dır. Böylece *mak.*, *min.* ve *ort.* değerleri eşittir (*0.507018*).

Çizelge 4.10. *B-2* ağı üzerindeki test sonuçları

		E_{MI}	E_{NMI}	E_{RI}	E_{ARI}	E_{JI}	E_P
F_M	*mak.*	1.0221161	0.6752366	0.8520626	0.6750173	0.6510067	0.4666667
	min.	1.0221161	0.6752366	0.8520626	0.6750173	0.6510067	0.4666667
	ort.	1.0221161	0.6752366	0.8520626	0.6750173	0.6510067	0.4666667
	std.	**0**	1.17E-16	**0**	1.17E-16	**0**	1.17E-16
F_C	*mak.*	0.7424876	0.6648771	0.7027027	0.4534921	0.5485961	**0.507018**
	min.	0.7424876	0.6648771	0.7027027	0.4534921	0.5485961	**0.507018**
	ort.	0.7424876	0.6648771	0.7027027	0.4534921	0.5485961	**0.507018**
	std.	1.17E-16	**0**	1.17E-16	1.17E-16	1.17E-16	**0**
F_{ID}	*mak.*	1.2911381	0.556892	0.7197724	0.3366876	0.349835	0.2280702
	min.	0.6673162	0.357229	0.5661451	0.0477512	0.1780303	0.0938596
	ort.	0.967038	0.4465463	0.6570413	0.1753731	0.2326287	0.1757895
	std.	0.2181662	0.0609482	0.0498315	0.0817715	0.0515074	0.0436935
F_{GD}	*mak.*	**1.4184734**	0.6957668	0.7951636	0.4958365	0.4375	0.3245614
	min.	1.13545	0.4950063	0.6970128	0.2201784	0.2052239	0.1921053
	ort.	1.2610621	0.5788208	0.7358464	0.3306601	0.2944258	0.2599561
	std.	0.1030041	0.065369	0.0295355	0.0846078	0.0731262	0.0436447
F_{CR}	*mak.*	0.7424876	0.6648771	0.7027027	0.4534921	0.5485961	**0.507018**
	min.	0.7424876	0.6648771	0.7027027	0.4534921	0.5485961	**0.507018**
	ort.	0.7424876	0.6648771	0.7027027	0.4534921	0.5485961	**0.507018**
	std.	1.17E-16	**0**	1.17E-16	1.17E-16	1.17E-16	**0**
F_{NC}	*mak.*	0.9656836	0.5751823	0.7211949	0.3643937	0.3931889	0.1714912
	min.	0.4140923	0.274188	0.5732575	0.0893861	0.2486772	-0.2732456
	ort.	0.6555302	0.3862731	0.6396871	0.1934797	0.3003901	0.0473684
	std.	0.1694208	0.0893273	0.0486841	0.093675	0.0456663	0.1307021
F_{FS}	*mak.*	1.236984	0.5898975	0.7510669	0.378244	0.3371212	0.3131579
	min.	1.1751809	0.5658525	0.7439545	0.3648611	0.3295455	0.2947368
	ort.	1.2308037	0.5872068	0.7497866	0.3752361	0.3352273	0.3014035
	std.	0.0195438	0.0075267	0.002366	0.0050332	0.0032191	0.0061222
F_{ODF}	*mak.*	0.7424876	0.6648771	0.7027027	0.4534921	0.5485961	**0.507018**
	min.	0.7424876	0.6648771	0.7027027	0.4534921	0.5485961	**0.507018**
	ort.	0.7424876	0.6648771	0.7027027	0.4534921	0.5485961	**0.507018**
	std.	1.17E-16	**0**	1.17E-16	1.17E-16	1.17E-16	**0**
F_S	*mak.*	1.3459763	0.6069819	0.7567568	0.3912557	0.3448276	0.2824561
	min.	**1.236984**	0.5386654	0.7098151	0.2477813	0.2153846	0.2026316
	ort.	**1.288949**	0.5773453	0.7302987	0.3115543	0.2725263	0.2248684
	std.	0.050507	0.0278837	0.0141487	0.0415019	0.0357295	0.0262182
F_{Sr}	*mak.*	1.1720578	**0.706189**	**0.864865**	**0.692755**	**0.653285**	0.3820175
	min.	1.1720578	**0.706189**	**0.864865**	**0.692755**	**0.653285**	0.3820175
	ort.	1.1720578	**0.706189**	**0.864865**	**0.692755**	**0.653285**	0.3820175
	std.	**0**	1.17E-16	1.17E-16	**0**	1.17E-16	5.851E-17

Çizelge 4.11. *B-3* ağındaki değerlendirme ölçütleri sonuçları

		E_{MI}	E_{NMI}	E_{RI}	E_{ARI}	E_{JI}	E_P
F_M	*mak.*	0.104179	0.0852598	0.5327696	0.0610907	0.3050314	0.3973485
	min.	0.0586943	0.049248	0.5095137	0.0156838	0.2861635	0.3848485
	ort.	0.0721009	0.0586945	0.520296	0.0357414	0.2893433	0.3864773
	std.	0.011796	0.0096124	0.0054995	0.010793	0.0068482	0.004001
F_C	*mak.*	0.0004389	0.0004955	0.4894292	-0.0151071	0.3775773	0.4356061
	min.	0.0004389	0.0004955	0.4894292	-0.0151071	0.3775773	0.4356061
	ort.	0.0004389	0.0004955	0.4894292	-0.0151071	0.3775773	0.4356061
	std.	1.143E-19	**0**	**1.17E-16**	3.657E-18	5.851E-17	**0**
F_{ID}	*mak.*	0.4541664	0.2297854	0.5486258	0.0819458	0.259434	0.2306818
	min.	0.1838809	0.1065474	0.5021142	-0.0018167	0.1300191	0.0458333
	ort.	0.3548647	0.19035	0.5274841	0.0427171	0.1808076	0.1245076
	std.	0.0743682	0.0342154	0.0152128	0.0289424	0.0483916	0.0573755
F_{GD}	*mak.*	0.4898605	0.2468243	**0.562368**	**0.110583**	0.1809524	0.2905303
	min.	0.1093729	0.0610931	0.5084567	0.0033261	0.1027132	0.1655303
	ort.	0.2335513	0.1257169	0.5261099	0.0379167	0.1473431	0.2224242
	std.	0.1008538	0.0488858	0.016943	0.0342827	0.0227434	0.0434357
F_{CR}	*mak.*	0.0008029	0.0009285	0.4883721	-0.0160801	**0.386565**	**0.475758**
	min.	0.0008029	0.0009285	0.4883721	-0.0160801	**0.386565**	**0.475758**
	ort.	0.0008029	0.0009285	0.4883721	-0.0160801	**0.386565**	**0.475758**
	std.	**0**	1.143E-19	**1.17E-16**	**0**	**0**	5.851E-17
F_{NC}	*mak.*	0.3290895	0.2412885	0.5369979	0.069504	0.3080569	0.1162879
	min.	0.0004845	0.0003947	0.4883721	-0.0284965	0.2194305	-0.1522727
	ort.	0.1054866	0.0761477	0.5102537	0.0146939	0.26773	-0.01375
	std.	0.1053998	0.0740918	0.0150903	0.029786	0.0301579	0.0881977
F_{FS}	*mak.*	0.3906043	0.1963084	0.5496829	0.0847199	0.1695568	0.3037879
	min.	0.2322269	0.1195442	**0.529598**	**0.044028**	0.1309524	0.2443182
	ort.	0.3430591	0.1728289	0.5404863	0.0659798	0.1404995	0.2832197
	std.	0.0500335	0.0222798	0.0051558	0.010578	0.0115791	0.0160419
F_{ODF}	*mak.*	0.0008029	0.0009285	0.4883721	-0.0160801	**0.386565**	**0.475758**
	min.	0.0008029	0.0009285	0.4883721	-0.0160801	**0.386565**	**0.475758**
	ort.	0.0008029	0.0009285	0.4883721	-0.0160801	**0.386565**	**0.475758**
	std.	**0**	1.143E-19	**1.17E-16**	**0**	**0**	5.851E-17
F_S	*mak.*	**0.543964**	**0.267105**	0.5613108	0.1087992	0.1733068	0.232197
	min.	**0.281156**	**0.141333**	0.5179704	0.0203525	0.107943	0.1659091
	ort.	**0.455087**	**0.22325**	**0.541015**	**0.066599**	0.1318945	0.1929924
	std.	0.0750036	0.0361997	0.0124259	0.0253559	0.020104	0.0194725
F_{Sr}	*mak.*	0.3718112	0.2100254	0.55074	0.0882352	0.1833333	0.2401515
	min.	0.178581	0.1080928	0.5243129	0.0351708	0.1620112	0.2060606
	ort.	0.2760935	0.1601098	0.5354123	0.0577935	0.1754203	0.2216288
	std.	0.0593307	0.0310903	0.0071738	0.0143173	0.0064509	0.0119256

Buradaki testlerde kullanılan *Duke* ağı *B-3* ile temsil edilmiştir. *3pHybrid* algoritması ile ağ üzerinde gerçekleştirilen testlerin (*30 çalıştırma sonunda*) sonuçları Çizelge 4.11'de verilmiştir. Bu sonuçlar tüm amaç fonksiyonlarının uygunluk fonksiyonları olarak kullanılmasıyla kaydedilen ağ modül yapılarının uygunluklarının 8 adet değerlendirme ölçütü ile test edilmesiyle elde edilmiştir. Burada E_{MI} ve E_{NMI} değerlendirme ölçütleri için en yüksek *mak.*, *min.* ve *ort.* parametre değerleri, F_S uygunluk fonksiyonu kullanılarak elde edilmiştir. Bu parametrelerin içerikleri sırasıyla; E_{MI} için 0.543964, 0.281156, 0.455087; E_{NMI} için 0.267105, 0.141333, 0.22325'tir. E_{MI}'ya göre F_{CR} ve F_{ODF} kullanılarak da en düşük *std.* değeri hesaplanmıştır. Bu değer 0'dır. Aynı hesaplama E_{NMI} ölçütüne göre F_C için de geçerlidir. E_{RI} ve E_{ARI} ölçütleri için en uygun *mak.* değerleri F_{GD} ile; en uygun *min.* değerleri F_{FS} ile ve en yüksek *ort.* değerleri F_S ile kaydedilmiştir. Benzer şekilde E_{JI} ve E_P değerlendirme ölçütlerine göre 10 adet amaç fonksiyonunun sonuçları karşılaştırıldığında; *mak.*, *min.* ve *ort.* değerleri açısından en başarılı sıralamaların F_{CR} ve F_{ODF} fonksiyonlarına ait olduğu anlaşılmıştır.

Diğer bir gerçek dünya ağını ifade eden *Khan*'ın test sonuçları Çizelge 4.12'de sunulmuştur. Bu ağ çizelgede *B-4* ile gösterilmiştir. Buradaki değerlendirme sonuçlarına göre E_{MI} açısından en iyi *mak.*, *min.* ve *ort.* değerleri F_{GD} fonksiyonun kullanılmasıyla elde edilmiştir. Bu değerler sırasıyla; 1.63287, 1.499879 ve 1.553307'dir. Bu ölçüte göre en düşük *std.* değeri 0'dır ve bu değer F_{CR} uygunluk fonksiyonu ile elde edilmiştir. E_{NMI} ve E_{RI} ölçütleri açısından *std.* değerleri hariç en iyi sonuçlara F_{FS} fonksiyonu ile erişilmiştir. E_{NMI}'ya göre hem F_C fonksiyonu hem de F_{ODF} fonksiyonu ile oldukça düşük *std.* değeri hesaplanmıştır ve bu değer 5.85E-17'dir. E_{RI}'da ise E_{ARI} ölçütündeki değerlendirme sonucunda olduğu gibi F_C ve F_{ODF} fonksiyonları ile 0 *std.* değeri elde edilmiştir. Sonuçlara dikkat edildiğinde; genellikle en düşük standart sapma değerlerine bu iki amaç fonksiyonunun kullanılmasıyla ulaşılmıştır. Bu durum ilgili fonksiyonların sonuçları arasındaki tutarlılığı gösterir. E_{JI} ve E_P ölçütleri için ise tüm kriterler açısından algoritmanın 30 kez çalıştırılmasıyla en iyi ortalama değerlere F_{CR} fonksiyonuyla erişilmiştir. Ayrıca E_P değerlendirme ölçütüne göre *std.* kriteri açısından F_{CR} fonksiyonunun yanında F_C ve F_{ODF} fonksiyonlarıyla da en uygun sonuca ulaşılmıştır. Bu sonuç, 1.17E-16 gibi oldukça düşük bir sayıdır.

Çizelge 4.12. *B-4* ağında kullanılan amaç fonksiyonlarının değerlendirme sonuçları

		E_{MI}	E_{NMI}	E_{RI}	E_{ARI}	E_{JI}	E_P
F_M	*mak.*	0.7508121	0.3573229	0.7073171	0.1861465	0.22899	0.4582329
	min.	0.7478163	0.3417059	0.7061416	0.1732075	0.21513	0.4562249
	ort.	0.7481159	0.3432676	0.7071995	0.1745014	0.216516	0.4580321
	std.	0.0009474	0.0049385	0.0003717	0.0040917	0.0043829	0.000635
F_C	*mak.*	0.472498	0.3862068	0.477226	0.1729952	0.3369363	0.5022088
	min.	0.472498	0.3862068	0.477226	0.1729952	0.3369363	0.5022088
	ort.	0.472498	0.3862068	0.477226	0.1729952	0.3369363	0.5022088
	std.	5.851E-17	**5.85E-17**	**0**	**0**	5.851E-17	**1.17E-16**
F_{ID}	*mak.*	1.3807893	0.4823709	0.7687335	0.2692874	0.2472325	0.3461847
	min.	1.0784119	0.3983841	0.7258301	0.1471973	0.1547017	0.1712851
	ort.	1.2354266	0.4549533	0.7523656	0.217353	0.2005526	0.2566667
	std.	0.1080519	0.0284126	0.0120955	0.0360861	0.027615	0.0559643
F_{GD}	*mak.*	**1.63287**	0.5506196	0.7731413	0.2337401	0.1822034	0.4333333
	min.	**1.499879**	0.4969468	0.7566853	0.1643364	0.1320755	0.3214859
	ort.	**1.553307**	0.520333	0.7640317	0.1959344	0.1548669	0.3815462
	std.	0.045411	0.0179987	0.0051348	0.0211995	0.0152579	0.0399177
F_{CR}	*mak.*	0.5643364	0.4550262	0.5013224	0.2004489	**0.350057**	**0.502811**
	min.	0.5643364	0.4550262	0.5013224	0.2004489	**0.350057**	**0.502811**
	ort.	0.5643364	0.4550262	0.5013224	0.2004489	**0.350057**	**0.502811**
	std.	**0**	1.17E-16	1.17E-16	2.926E-17	**0**	**1.17E-16**
F_{NC}	*mak.*	1.0287699	0.4527705	0.7472818	**0.322542**	0.3233674	0.3799197
	min.	0.707032	0.3293812	0.6726418	0.1275447	0.1926121	0.0580321
	ort.	0.8800724	0.3890325	0.7133412	0.205509	0.2399766	0.2148193
	std.	0.123824	0.0420191	0.0251738	0.0575693	0.0397593	0.0901223
F_{FS}	*mak.*	1.5878664	**0.55547**	**0.774611**	0.2509453	0.2033024	0.4485944
	min.	1.4544201	**0.51281**	**0.765207**	**0.217367**	0.1779835	0.4088353
	ort.	1.5120761	**0.533031**	**0.770585**	0.2360053	0.1900678	0.4261044
	std.	0.0471274	0.0140265	0.0027288	0.0097979	0.0071095	0.012341
F_{ODF}	*mak.*	0.472498	0.3862068	0.477226	0.1729952	0.3369363	0.5022088
	min.	0.472498	0.3862068	0.477226	0.1729952	0.3369363	0.5022088
	ort.	0.472498	0.3862068	0.477226	0.1729952	0.3369363	0.5022088
	std.	5.851E-17	**5.85E-17**	**0**	**0**	5.851E-17	**1.17E-16**
F_S	*mak.*	1.5723939	0.5489806	0.7737291	0.2481591	0.1987513	0.4590361
	min.	1.4263343	0.4944557	0.7599177	0.1946243	0.161191	0.3564257
	ort.	1.4988277	0.5283385	0.7695563	0.2316606	0.1868618	0.4082329
	std.	0.0473617	0.0162703	0.0040548	0.0164803	0.0123583	0.0309214
F_{Sr}	*mak.*	1.4711288	0.5439862	0.7737291	0.2781698	0.2353525	0.4590361
	min.	1.2079524	0.4648042	0.7481634	0.2144248	0.1910331	0.3837349
	ort.	1.3407264	0.507227	0.7632971	**0.240639**	0.2089216	0.4324498
	std.	0.0867257	0.0258206	0.0081486	0.0203891	0.0131893	0.0212532

Çizelge 4.13. *B-5* ağı için uygunluk fonksiyonlarının değerlendirme sonuçları

		E_{MI}	E_{NMI}	E_{RI}	E_{ARI}	E_{JI}	E_P
F_M	*mak.*	1.5829484	0.5051652	0.8247962	0.2687132	0.2119465	0.4046296
	min.	1.2450942	0.4284253	0.7399887	0.1713027	0.1576412	0.3700337
	ort.	1.3806859	0.4600717	0.7834641	0.2159816	0.1818164	0.387096
	std.	0.1031301	0.0297372	0.0320155	0.0309331	0.0172729	0.0118342
F_C	*mak.*	0.7230443	0.3206201	0.4397785	0.0822365	0.1170909	0.4393098
	min.	0.6641794	0.2956325	0.4248064	0.0714025	0.1114455	**0.435606**
	ort.	0.6848787	0.3042599	0.4316772	0.0757351	0.1136828	0.4369192
	std.	**0.014618**	**0.006232**	**0.003544**	**0.002617**	**0.001366**	**0.000918**
F_{ID}	*mak.*	1.4663138	0.4709606	0.8206943	0.2273062	0.1846584	0.2905724
	min.	0.8629028	0.3053135	0.5732964	0.045792	0.0941548	0.1662458
	ort.	1.2007321	0.3917794	0.7188535	0.1215645	0.129114	0.2354545
	std.	0.2261896	0.0636871	0.0810333	0.0561658	0.0283893	0.0373374
F_{GD}	*mak.*	2.4311473	0.5825128	**0.92273**	0.3272647	0.2231959	0.2344276
	min.	2.2760071	0.5560791	0.9098087	0.233823	0.1602404	0.2010943
	ort.	2.3673856	0.5731839	**0.916346**	0.2710277	0.1849685	0.220463
	std.	0.0535552	0.0075644	0.0037762	0.0309229	0.0211865	0.0132622
F_{CR}	*mak.*	0.7230443	0.3206201	0.4397785	0.0822365	0.1170909	**0.440488**
	min.	0.3994719	0.1888995	0.2678562	0.0209673	0.0869621	0.4342593
	ort.	0.6105657	0.2754313	0.3768548	0.0607344	0.1065407	**0.437912**
	std.	0.1532319	0.0622259	0.0829681	0.0290834	0.0142932	0.002547
F_{NC}	*mak.*	1.5975973	0.4660495	0.8495103	0.2003368	0.1665044	0.1743266
	min.	1.089544	0.354697	0.7705481	0.116035	0.1162169	0.076431
	ort.	1.3574568	0.4091573	0.8069169	0.1549921	0.1406313	0.1204798
	std.	0.1439959	0.0342074	0.0217439	0.0295401	0.0170472	0.0330874
F_{FS}	*mak.*	**2.517087**	**0.609067**	**0.92273**	0.3372531	0.2311224	0.3172559
	min.	**2.291788**	**0.569484**	**0.910168**	0.2492914	0.1728045	0.2693603
	ort.	**2.38571**	**0.58629**	0.9148746	0.2925241	0.2019187	0.2965488
	std.	0.078351	0.0127245	0.0041955	0.0270409	0.0184048	0.0154298
F_{ODF}	*mak.*	0.7230443	0.3206201	0.4397785	0.0822365	0.1170909	0.4393098
	min.	0.3764946	0.1786665	0.2584731	0.0171225	0.0850889	0.4308923
	ort.	0.6524071	0.2910311	0.4137005	0.0694948	0.1106289	0.4362121
	std.	0.098262	0.0400636	0.0546994	0.0186377	0.009105	0.0021239
F_S	*mak.*	2.4310202	0.5953687	0.918269	0.3540751	0.2503329	0.3267677
	min.	2.1176734	0.5515414	0.9062708	0.2673806	0.1864952	0.2800505
	ort.	2.2476657	0.5704635	0.9117931	0.3107566	0.2174987	0.3095707
	std.	0.0819623	0.0138578	0.0039668	0.0220451	0.0160669	0.0139762
F_{Sr}	*mak.*	2.0919838	0.569477	0.9089371	**0.401325**	**0.291699**	0.3548822
	min.	1.8586277	0.5179217	0.8951956	**0.319552**	**0.231152**	0.3231481
	ort.	1.969421	0.5419871	0.9016562	**0.360106**	**0.26085**	0.3339899
	std.	0.065581	0.0156792	0.0048424	0.0253791	0.018289	0.0116229

Çizelge 4.13'te, *B-5* ile gösterilen *Cancer* ağı üzerinde gerçekleştirilen testlerin sonuçları verilmiştir. Testler her bir amaç fonksiyonunun bu ağdaki düğüm ilişkilerine göre belirlenen ağ modüllerinin/topluluklarının uygunluklarının değerlendirme ölçütleri yardımıyla analiz edilmesini içerir. Bunun için tabloda gösterildiği gibi her bir değerlendirme ölçütüne göre amaç fonksiyonlarının *mak.*, *min.*, *ort.* ve *std.* değerleri tespit edilmiş ve bunlara göre genellikle en iyi sonuçları öneren amaç fonksiyonunun belirlenmesi amaçlanmıştır. Bu amaçla *Cancer* ağındaki sonuçlar incelendiğinde, E_{MI} ve E_{NMI}'ya göre ilk üç sonuçta en iyi değerler F_{FS} fonksiyonu ile elde edilmiştir. Bunlar Çizelge 4.13'teki ilk iki sütunda koyu renklerle vurgulanmıştır. Bu ölçütlere göre en uygun *std.* değerleri ise F_C ile belirlenmiştir. Bunlar; E_{MI} için 0.014618 ve E_{NMI} için 0.006232'dir. Bunlara ek olarak F_C amaç fonksiyonu ile tüm değerlendirme ölçütlerine göre en düşük standart sapma değerlerine ulaşılmıştır. E_{RI}'ya göre en yüksek *mak.* değeri 0.92273'tür ve bu değer hem F_{GD} hem de F_{FS} ile elde edilmiştir. Yine bu ölçüte göre en yüksek *ort.* değeri F_{GD} ile; en yüksek *min.* değeri ise F_{FS} ile belirlenmiştir.

E_{ARI} ve E_{JI} değerlendirme ölçütleri için *B-5*'deki testlere göre *std.* dışındaki en uygun değerler F_{Sr} ile elde edilmişken; F_M fonksiyonu ile bu ağdaki sonuçlar için tüm ölçütler açısından genelde ortalama değerlere ulaşılmıştır. E_P'ye göre en yüksek *mak.* ve *ort.* değerleri F_{CR}'nin; en iyi *min.* ve *std.* değerleri ise F_C'nin uygunluk fonksiyonları olarak seçilmeleriyle elde edilmiştir.

Ecoli biyolojik ağındaki denemelere göre amaç fonksiyonlarının kullanılmasıyla elde edilen değerlendirme sonuçları Çizelge 4.14'te sunulmuştur. Bu ağ, sonuçlar tablosunda *B-6* kısaltmasıyla temsil edilmiştir. Buradaki deneylerin test sonuçları incelendiğinde, genellikle en uygun sonuçlara F_{CR} ile erişildiği gözlemlenmiştir. Bunun dışında E_{MI}'ya göre en yüksek *mak.*, *min.* ve *ort.* sonuçları F_{FS} ile elde edilirken; bu ölçüte göre en düşük standart sapmaya F_S ile ulaşılmıştır. E_P değerlendirme ölçütü açısından dört kritere göre en başarılı sonuçlar farklı fonksiyonların kullanılması sonucu elde edilmiştir. Buna göre elde edilen değerler ve kullanılan amaç fonksiyonları şunlardır: 0.399134 *mak.* değeriyle F_{ODF}; 0.272885 *min.* değeriyle F_M; 0.307752 *ort.* değeriyle F_{CR} ve 0.015621 *std.* değeriyle F_S. Genellikle tüm tablolardaki sonuçlar incelendiğinde, F_{CR} uygunluk fonksiyonunun kullanılmasıyla *mak.*, *min.*, *ort.* ve *std.* değerlerine göre bazı ağlarda en iyi sonuçlara ulaşılsa da tüm ağlardaki deney sonuçları birden dikkate alındığında, F_M fonksiyonunun seçilmesiyle ortalama olarak daha makul sonuçların elde edildiği gözlemlenmiştir. Bunun için EK-1, EK-2 ve EK-3 incelenebilir.

Çizelge 4.14. *B*-6 ağında kullanılan amaç fonksiyonlarının değerlendirme ölçütleri sonuçları

		E_{MI}	E_{NMI}	E_{RI}	E_{ARI}	E_{JI}	E_{P}
F_M	*mak.*	1.498897	0.5886363	0.7883905	0.4104666	0.3669884	0.3200306
	min.	1.2637377	0.5090264	0.7445451	0.2698866	0.2553555	**0.272885**
	ort.	1.3738003	0.5403798	0.7686685	0.3388039	0.3066275	0.2971611
	std.	0.0795917	0.0237168	0.0153037	0.0416999	0.0333416	0.0192348
F_C	*mak.*	1.4196275	0.6088953	0.8233804	0.5409924	0.4907773	0.32263
	min.	1.1424049	**0.531077**	**0.774582**	0.3495707	0.3102359	0.2610092
	ort.	1.279752	0.5582198	0.7968481	0.4444297	0.399754	0.2988022
	std.	0.0912178	0.021922	0.0166548	0.0569853	0.0528371	0.020673
F_{ID}	*mak.*	1.5274201	0.4637285	0.7424063	0.1724876	0.146675	0.1824159
	min.	1.3491051	0.420781	0.7283728	0.1106503	0.097839	0.0785423
	ort.	1.4544417	0.4388715	0.7352076	0.1389491	0.1206032	0.1359072
	std.	0.0585424	0.013868	0.0052178	0.0198416	0.0162478	0.0291743
F_{GD}	*mak.*	1.6859677	0.4490153	0.7312996	0.0920986	0.072169	0.2346585
	min.	1.5833726	0.4300238	0.7275848	0.0760641	0.06061	0.1641182
	ort.	1.6143991	0.4383462	0.729204	0.083795	0.0668359	0.191473
	std.	0.0282648	**0.005963**	**0.001145**	**0.004975**	**0.00366**	0.0185254
F_{CR}	*mak.*	1.4280782	**0.650609**	**0.861297**	**0.675492**	**0.632536**	0.369419
	min.	0.8541233	0.4519271	0.7321251	**0.373884**	**0.36643**	0.2600917
	ort.	1.2013281	**0.566106**	**0.80058**	**0.489081**	**0.454718**	**0.307752**
	std.	0.1580201	0.0564517	0.0365485	0.0913948	0.082626	0.0382214
F_{NC}	*mak.*	1.4923107	0.5242749	0.7698542	0.3189647	0.278836	0.1875127
	min.	1.3386157	0.4397755	0.7275661	0.1540829	0.1534425	0.0962283
	ort.	1.4100613	0.47472	0.7548883	0.2442374	0.2121867	0.1563863
	std.	0.0599685	0.0248826	0.0138767	0.0529946	0.0415054	0.0298344
F_{FS}	*mak.*	**1.702015**	0.4578782	0.732444	0.0987417	0.0776743	0.2461774
	min.	**1.594975**	0.4351803	0.7280539	0.0781074	0.0622654	0.1747706
	ort.	**1.652327**	0.4492334	0.7303465	0.0885952	0.0700656	0.225
	std.	0.0316693	0.0065542	0.0015938	0.0064332	0.0055504	0.0210958
F_{ODF}	*mak.*	1.351407	0.6034366	0.802105	0.479597	0.4686348	**0.399134**
	min.	0.7344827	0.4696075	0.6916943	0.2714879	0.2684258	0.2138634
	ort.	1.0973323	0.5223066	0.7551097	0.383984	0.378679	0.3072375
	std.	0.1886473	0.0463314	0.0390083	0.0787461	0.0667678	0.0505235
F_S	*mak.*	1.6476158	0.4667677	0.7379974	0.1255755	0.0981595	0.2738022
	min.	1.5806678	0.439745	0.7283916	0.0826979	0.0673238	0.2301223
	ort.	1.6203406	0.4542705	0.7338155	0.1070882	0.0848447	0.2503211
	std.	**0.019035**	0.0085988	0.0029714	0.0136258	0.0100954	**0.015621**
F_{Sr}	*mak.*	1.6426668	0.483391	0.7417497	0.1527	0.1235275	0.2777268
	min.	1.5500198	0.457215	0.7334947	0.10805	0.0866714	0.2189602
	ort.	1.5994752	0.4711471	0.7389355	0.1343371	0.1069675	0.254103
	std.	0.0264695	0.0080411	0.0026121	0.0130161	0.010465	0.0168982

Çizelge 4.15. *B-7* ağındaki test sonuçları

		E_{MI}	E_{NMI}	E_{RI}	E_{ARI}	E_{JI}	E_P
F_M	*mak.*	0.4254524	0.2378769	0.5600326	0.1602469	0.257853	0.3905983
	min.	0.3145318	0.1754702	0.5367359	0.1145006	0.221325	0.3623457
	ort.	0.3632981	0.2021486	0.5490175	0.1386452	0.2375695	0.3752374
	std.	0.0328125	0.0171247	0.0079204	0.0148853	0.0114339	0.0109661
F_C	*mak.*	0.3887184	0.2790052	0.614652	0.2393244	0.4240736	0.4123457
	min.	0.292503	0.2066914	0.5610094	0.1399758	0.3456368	**0.399668**
	ort.	0.3352234	0.2549343	0.5975857	0.2079532	0.3976222	**0.406021**
	std.	0.033552	0.0235147	0.0168448	0.0316258	0.0262712	**0.004913**
F_{ID}	*mak.*	0.4248892	0.1817315	0.5245258	0.1008473	0.1654713	0.2575024
	min.	0.3247825	0.1425695	0.4913146	0.0444963	0.0967015	0.0991928
	ort.	0.3834743	0.163256	0.5050517	0.0672841	0.1278461	0.1909592
	std.	0.027894	0.0125667	0.010365	0.0173548	0.0236392	0.0500467
F_{GD}	*mak.*	0.4552096	0.1527448	0.4831095	0.0342318	0.058143	0.2424027
	min.	0.3514507	0.1229727	0.4765812	0.0228414	0.0453476	0.1696581
	ort.	0.3943888	0.1380563	0.4789369	0.0270617	0.0502779	0.2135565
	std.	0.0274747	0.0085739	**0.001968**	**0.003462**	**0.003846**	0.0188099
F_{CR}	*mak.*	0.3594645	0.2844259	0.645812	0.2925232	0.4883108	0.4107787
	min.	0.3162672	**0.239704**	**0.576199**	**0.17063**	**0.357568**	0.3929725
	ort.	0.3402741	0.2572155	0.5988816	0.2110497	0.3968882	0.4055033
	std.	**0.014683**	0.0144251	0.0210146	0.0366577	0.0417925	0.0058862
F_{NC}	*mak.*	0.377809	0.1864836	0.5344729	0.114708	0.206695	0.3083571
	min.	0.3171733	0.1394019	0.4993407	0.0585824	0.1103622	0.1132953
	ort.	0.3454615	0.1631086	0.5160114	0.0837912	0.1618566	0.2315385
	std.	0.022542	0.0154657	0.011507	0.017864	0.0345377	0.060121
F_{FS}	*mak.*	**0.486738**	0.171442	0.5024664	0.0664033	0.1019395	0.2933523
	min.	**0.41049**	0.1512492	0.4835816	0.0352291	0.0578854	0.2367521
	ort.	**0.449498**	0.1618218	0.4910509	0.0476371	0.0747751	0.2654891
	std.	0.022423	0.0069257	0.0064167	0.0108358	0.0141067	0.0197881
F_{ODF}	*mak.*	0.3908625	**0.301861**	**0.650403**	**0.302468**	**0.490594**	**0.415195**
	min.	0.2990961	0.2305402	0.5674562	0.1522107	0.3536309	0.3969136
	ort.	0.3434875	**0.261469**	**0.600607**	**0.213876**	**0.400734**	0.4055318
	std.	0.0252057	0.0226624	0.0233548	0.0419033	0.0443284	0.0068569
F_S	*mak.*	0.4337681	0.1692006	0.5080505	0.0759065	0.1147484	0.3216049
	min.	0.3601285	0.1473835	0.4963289	0.0562991	0.0873746	0.2900285
	ort.	0.3971905	0.159083	0.5019243	0.0645632	0.1062978	0.3005793
	std.	0.0212618	**0.006161**	0.0038315	0.0068735	0.0082401	0.0115374
F_{Sr}	*mak.*	0.3658567	0.1628568	0.5145136	0.0833528	0.1473136	0.3345204
	min.	0.3151811	0.1402021	0.5031502	0.0652922	0.1146637	0.2973409
	ort.	0.3439706	0.1523202	0.5085584	0.0746783	0.1267534	0.3162678
	std.	0.0152417	0.0071585	0.0035892	0.0057966	0.009674	0.0137789

Ionosphere gerçek dünya ağının test verisi olarak kullanılması ile tüm amaç fonksiyonlarının küme değerlendirme sonuçları Çizelge 4.15'te gösterilmiştir. Bu ağ çizelgede *B-7* ile ifade edilmiştir. Değerlendirme ölçütlerinin ilki—E_{MI} ile *mak.*, *min.* ve *ort.* sonuçlarına göre en uygun amaç fonksiyonunun F_{FS} olduğu belirlenmiştir. Bu ölçüte göre en uygun *std.* değeri (*0.014683*) F_{CR} ile elde edilmiştir. E_{NMI} ölçütüne göre sırasıyla; en yüksek *mak.* ve *ort.* değerlerine F_{ODF}; en yüksek *min.* değerine F_{CR} ve en düşük *std.* değerine F_S fonksiyonlarının kullanılmasıyla ulaşılmıştır. E_{RI}, E_{ARI} ve E_J değerlendirme ölçütleri için en iyi *mak.* ve *ort.* sonuçları yine F_{ODF} ile elde edilmiştir. Aynı şekilde bu ölçütlerin değerlendirme sonuçlarına göre en iyi *min.* ve *std.* değerleri sırasıyla; F_{CR} ve F_{GD} amaç fonksiyonları ile kaydedilmiştir. E_P ölçütünün sonuçlarına göre ise en yüksek mak. değeri 0.415195'tir ve bu değer F_{ODF}'nin kalite fonksiyonu olarak kullanılmasıyla elde edilmiştir. Ayrıca E_P ölçütüne göre en uygun *min.*, *ort.* ve *std.* değerlerine F_C ile ulaşılmıştır.

Bu bölümde kullanılan en son test verisi *Yeast* ağıdır. Bu ağda gerçekleştirilen deneylerin sonuçları Çizelge 4.16'da sunulmuştur ve bu biyolojik ağ, ilgili tabloda *B-8* ile temsil edilmiştir. Bu ağdaki test sonuçlarına göre E_{MI} ve E_{NMI} ölçütleri açısından en yüksek *mak.*, *min.* ve *ort.* değerlerine F_{FS} ile ulaşılmıştır. Aynı ölçütlere göre en düşük standart sapma değerleri ise F_{Sr} ile hesaplanmıştır. Bu değerler E_{MI} ve E_{NMI} ölçütleri için sırasıyla 0.01819 ve 0.003022'dir.

B-8'deki test sonuçlarında E_{RI} değerlendirme ölçütüne göre tüm kriterler için en iyi sonuçlar F_{FS} ile elde edilmştir. E_{ARI} ölçütüne göre ise en uygun sonuçlar farklı amaç fonksiyonları ile elde edilmiştir. Bu fonksiyonlara göre elde edilen en iyi değerler; F_{CR}—0.104018(*mak.*), F_{Sr}—0.042501(*min.*), F_{CR}—0.051802(*ort.*), F_{CR}—0.00167(*std.*) şeklindedir. E_{JI} ve E_P değerlendirme ölçütlerine göre en uygun *std.* değerleri (*0.001159* ve *0.007165*) F_{GD}'nin uygunluk fonksiyonu olarak seçilmesiyle elde edilmiştir. E_{JI} için diğer en iyi sonuçlara F_{CR} ile ulaşılmıştır. Bunlar *mak.*: *0.12356*, *min.*: *0.058607* ve *ort.*: *0.074765* değerleridir. E_P ölçütüne göre en yüksek *mak.*, *min.* ve *ort.* sayıları tablodaki sıralarına göre; 0.163623, 0.128571, 0.146897'dir ve bu sonuçlar F_S amaç fonksiyonu ile kaydedilmiştir.

Çizelge 4.16. *B-8*'deki sonuçlara göre değerlendirme ölçütlerinin farklı değerleri (*mak.*, *min.*, *ort.* ve *std.*)

		E_{MI}	E_{NMI}	E_{RI}	E_{ARI}	E_{JI}	E_P
F_M	*mak.*	1.0437185	0.258417	0.7723017	0.0701816	0.0715674	0.1512354
	min.	0.949278	0.2379696	0.7679978	0.0362558	0.0430665	0.122664
	ort.	1.0008207	0.2484736	0.7707044	0.0489668	0.0547864	0.1328347
	std.	0.0308092	0.0071034	0.0013979	0.0104484	0.0093919	0.0100022
F_C	*mak.*	0.978796	0.2424185	0.7726434	0.0635728	0.0941051	0.133367
	min.	0.8144196	0.2168092	0.7508156	**0.042501**	0.0460175	0.0742138
	ort.	0.8706442	0.229669	0.7644849	0.0496062	0.0641696	0.1110726
	std.	0.0513869	0.0089139	0.0059706	0.0071679	0.0122415	0.0167885
F_{ID}	*mak.*	1.138586	0.2552352	0.7754234	0.0291184	0.0337224	0.1026056
	min.	1.0384908	0.2369743	0.7699525	0.0196826	0.02369	0.0761905
	ort.	1.0794334	0.2444888	0.7731829	0.0236396	0.0282574	0.0876516
	std.	0.0310621	0.0055198	0.0019485	0.0029348	0.0039143	0.0089543
F_{GD}	*mak.*	1.1994465	0.258441	0.7759677	0.0208536	0.0211282	0.1176213
	min.	1.1297717	0.243219	0.7751571	0.0144059	0.0167157	0.0941038
	ort.	1.1729319	0.251864	0.7756098	0.0172901	0.0185328	0.1083434
	std.	0.0217333	0.0050963	0.0002961	0.001745	**0.001159**	**0.007165**
F_{CR}	*mak.*	0.9211762	0.2484441	0.7659748	**0.104018**	**0.12356**	0.1278302
	min.	0.7164923	0.2054733	0.7443025	0.0335246	**0.058607**	0.0874326
	ort.	0.8262447	0.2255184	0.7580522	**0.051802**	**0.074765**	0.1099135
	std.	0.0654696	0.0138419	0.0067712	0.0201378	0.0213045	0.0123764
F_{NC}	*mak.*	1.0716346	0.2550634	0.7736131	0.0469839	0.0510042	0.105593
	min.	0.989699	0.2372342	0.7711567	0.0252925	0.0300344	0.0642071
	ort.	1.0329571	0.2431387	0.7721859	0.0329876	0.0382372	0.0889196
	std.	0.0287155	0.0050972	0.0008563	0.0072279	0.0069608	0.0132928
F_{FS}	*mak.*	**1.24079**	**0.263552**	**0.776123**	0.0200151	0.0213359	0.1443396
	min.	**1.170251**	**0.250536**	**0.775402**	0.0142785	0.0149388	0.1184299
	ort.	**1.211319**	**0.25761**	**0.775836**	0.0174236	0.0182638	0.1313241
	std.	0.0238681	0.0047775	**0.000226**	0.0017481	0.0018384	0.0092343
F_{ODF}	*mak.*	0.8980402	0.2387931	0.7676779	0.0782495	0.1028202	0.121889
	min.	0.741359	0.2158602	0.7366351	0.0356627	0.0531716	0.0911051
	ort.	0.8332471	0.2270142	0.7583096	0.0502977	0.0728108	0.1027482
	std.	0.0567881	0.0080894	0.0102771	0.0121655	0.0169236	0.0108469
F_S	*mak.*	1.1763831	0.2576287	0.775855	0.0260805	0.0260308	**0.163623**
	min.	1.1086916	0.2441585	0.7743801	0.0164362	0.0189307	**0.128571**
	ort.	1.1384119	0.2505059	0.7750835	0.0203188	0.0221496	**0.146897**
	std.	0.0206322	0.0041893	0.0004589	0.0027428	0.0021789	0.0143494
F_{Sr}	*mak.*	1.1591123	0.2565727	0.7758723	0.0250945	0.0253204	0.1453504
	min.	1.1042386	0.2469784	0.7749263	0.0202912	0.0217584	0.1176999
	ort.	1.12939	0.2507645	0.7753774	0.0224354	0.0235546	0.1329403
	std.	**0.01819**	**0.003022**	0.0003017	**0.00167**	0.001504	0.0100246

Yukarıda sunulan deneysel sonuçlara ek olarak, parametrik olmayan *Friedman testi* (Friedman 1937) ile *3pHybrid* algoritmasında uygunluk fonksiyonu olarak seçilen amaç fonksiyonlarının önerdikleri ağ modüllerinin/topluluklarının kalitelerine göre belirlenen başarı sıralamaları aşağıda analiz edilmiştir.

Her bir ağda seçilen amaç fonksiyonlarının değerlendirme fonksiyonlarına göre belirlenen değerlendirme sonuçları yukarıdaki çizelgelerde *mak.*, *min.*, *ort.* ve *std.* değerleri ile sunulmuştur. Bu değerler algoritmanın ilgili amaç fonksiyonları ile 30 kez çalıştırılması sonunda elde edilen nihai sonuçları gösterir. İstatistiksel test işleminde ise her bir ağdaki deney sonuçlarının ayrı ayrı değerlendirilmesi yerine bunların ortalamaları temel alınmıştır. Tüm ağlardaki test sonuçları birden dikkate alındığında, tablolardaki toplam maksimum (*mak.*) değerlerin ortalaması $\underline{\langle mak \rangle}$ ile; toplam minimum (*min.*) değerlerin ortalaması $\underline{\langle min \rangle}$ ile; toplam ortalama (*ort.*) değerlendirme sonuçlarının ortalaması $\underline{\langle ort \rangle}$ ile ve tüm standart sapma (*std.*) değerlerinin ortalaması $\underline{\langle std \rangle}$ ile temsil edilmiştir. Buradaki sonuçlar Çizelge 4.17'de gösterilmiştir. Çizelgede sunulan en iyi ortalamalara ($\langle mak \rangle, \langle min \rangle, \langle ort \rangle \; ve \; \langle std \rangle$) göre amaç fonksiyonlarının değerlendirme sonuçları temel alınarak belirlenen başarı sıralamaları parantez içerisinde koyu renklerle vurgulanmıştır.

Modülerlik amaç fonksiyonunun (F_M) ortalama sonuçları incelendiğinde, E_{RI} ve E_{ARI} değerlendirme sonuçlarına göre $\langle std \rangle$ değerleri hariç en başarılı sonuçlar, bu fonksiyonun uygunluk fonksiyonu olarak seçilmesi ile elde edilmiştir. Yine diğer değerlendirme ölçütlerinin sonuçları da F_M fonksiyonunun makul sonuçlara ulaştığını doğrulamaktadır. Bir diğer amaç fonksiyonu olan F_{CR}'nin E_{JI} ve E_P değerlendirme ölçütlerine göre genellikle en uygun sonuçları elde ettiği anlaşılmıştır. Ancak bu fonksiyonunun diğer ölçütlere göre kaydedilen sonuçları, F_M fonksiyonunun sonuçları kadar başarılı değildir. Ayrıca hemen hemen tüm değerlendirme ölçütlerine göre ortalama başarıya sahip F_{Sr} uygunluk fonksiyonu genel olarak uygun sonuçlar önermektedir. F_{ID} ve F_{NC} fonksiyonlarının değerlendirme sonuçları incelendiğinde, bu iki fonksiyonunun genellikle çoğu ölçüte göre en kötü kalitede ağ modüllerini önerdikleri anlaşılmıştır. Bunlara ek olarak, diğer amaç fonksiyonlarının tüm ölçütlere göre başarı sıralamaları aşağıdaki tabloda gösterilmiştir. Son olarak, E_{JI} ve E_P ölçütleri kullanılarak yapılan bazı değerlendirme testlerinin sonuçları ile diğer ölçütlerin (E_{MI}, E_{NMI}, E_{RI} ve E_{ARI}) sonuçları arasında bazı tutarsızlıkların olduğu dikkate değer bir analiz sonucu olarak not düşülebilir.

Çizelge 4.17. Amaç fonksiyonlarının tüm ağlardaki sonuçlarına göre ortalama değerleri ve sıralamaları

		E_{MI}	E_{NMI}	E_{RI}	E_{ARI}	E_{JI}	E_P
F_M	⟨mak⟩	0.98818 (**6**)	0.44900 (**3**)	0.74384 (**1**)	0.32871 (**1**)	0.35575 (**4**)	0.38521 (**4**)
	⟨min⟩	0.88483 (**5**)	0.41264 (**1**)	0.72125 (**1**)	0.28171 (**1**)	0.32278 (**2**)	0.36608 (**4**)
	⟨ort⟩	0.92978 (**6**)	0.42654 (**3**)	0.73306 (**1**)	0.30318 (**1**)	0.33628 (**4**)	0.37456 (**4**)
	⟨std⟩	0.03239 (**2**)	0.01153 (**3**)	0.00781 (**5**)	0.01411 (**4**)	0.01033 (**3**)	0.00708 (**2**)
F_C	⟨mak⟩	0.67195 (**9**)	0.38920 (**10**)	0.62014 (**10**)	0.23747 (**5**)	0.36051 (**3**)	0.40538 (**3**)
	⟨min⟩	0.59737 (**8**)	0.36411 (**6**)	0.60273 (**8**)	0.19713 (**4**)	0.32142 (**3**)	0.38824 (**3**)
	⟨ort⟩	0.62949 (**8**)	0.37622 (**7**)	0.61266 (**8**)	0.21892 (**4**)	0.34165 (**2**)	0.39853 (**3**)
	⟨std⟩	0.02385 (**1**)	0.00757 (**1**)	0.00538 (**4**)	0.01230 (**3**)	0.01159 (**4**)	0.00541 (**1**)
F_{ID}	⟨mak⟩	1.14508 (**5**)	0.40844 (**6**)	0.70310 (**7**)	0.18519 (**10**)	0.20044 (**7**)	0.25816 (**9**)
	⟨min⟩	0.85843 (**6**)	0.31687 (**9**)	0.63158 (**7**)	0.07323 (**9**)	0.11475 (**10**)	0.11681 (**9**)
	⟨ort⟩	1.01349 (**5**)	0.36587 (**9**)	0.67233 (**6**)	0.12537 (**10**)	0.14978 (**9**)	0.18770 (**9**)
	⟨std⟩	0.09905 (**10**)	0.03114 (**9**)	0.02364 (**10**)	0.03565 (**9**)	0.02914 (**9**)	0.04660 (**9**)
F_{GD}	⟨mak⟩	1.34879 (**1**)	0.44986 (**1**)	0.72286 (**3**)	0.20314 (**8**)	0.17913 (**9**)	0.30686 (**8**)
	⟨min⟩	1.18491 (**3**)	0.37671 (**5**)	0.69699 (**5**)	0.12538 (**8**)	0.11843 (**9**)	0.23431 (**8**)
	⟨ort⟩	1.25096 (**3**)	0.40570 (**5**)	0.70670 (**5**)	0.15503 (**8**)	0.14355 (**10**)	0.27161 (**8**)
	⟨std⟩	0.05193 (**8**)	0.02281 (**8**)	0.00811 (**6**)	0.02489 (**8**)	0.01937 (**7**)	0.02367 (**8**)
F_{CR}	⟨mak⟩	0.67368 (**8**)	0.40450 (**7**)	0.63082 (**8**)	0.26930 (**2**)	0.39270 (**1**)	0.41558 (**2**)
	⟨min⟩	0.53050 (**9**)	0.35224 (**7**)	0.58177 (**9**)	0.19989 (**3**)	0.33121 (**1**)	0.39386 (**1**)
	⟨ort⟩	0.61701 (**9**)	0.38203 (**6**)	0.60851 (**9**)	0.22660 (**3**)	0.35163 (**1**)	0.40466 (**1**)
	⟨std⟩	0.04893 (**7**)	0.01837 (**7**)	0.01841 (**8**)	0.02216 (**7**)	0.02000 (**8**)	0.00738 (**3**)
F_{NC}	⟨mak⟩	0.98368 (**7**)	0.40280 (**8**)	0.70357 (**6**)	0.21394 (**6**)	0.25489 (**5**)	0.22505 (**10**)
	⟨min⟩	0.67569 (**7**)	0.26195 (**10**)	0.63215 (**6**)	0.07031 (**10**)	0.15985 (**6**)	0.00668 (**10**)
	⟨ort⟩	0.82659 (**7**)	0.32373 (**10**)	0.67023 (**7**)	0.13769 (**9**)	0.20242 (**6**)	0.13494 (**10**)
	⟨std⟩	0.09734 (**9**)	0.04260 (**10**)	0.02184 (**9**)	0.04470 (**10**)	0.03077 (**10**)	0.06861 (**10**)
F_{FS}	⟨mak⟩	1.32993 (**2**)	0.43832 (**5**)	0.71853 (**5**)	0.19336 (**9**)	0.17497 (**10**)	0.32913 (**6**)
	⟨min⟩	1.22583 (**1**)	0.40845 (**2**)	0.70939 (**3**)	0.16421 (**6**)	0.15051 (**7**)	0.28920 (**6**)
	⟨ort⟩	1.28277 (**1**)	0.42637 (**4**)	0.71401 (**3**)	0.17924 (**6**)	0.16107 (**7**)	0.31193 (**6**)
	⟨std⟩	0.03413 (**3**)	0.00935 (**2**)	0.00284 (**1**)	0.00893 (**1**)	0.00773 (**1**)	0.01251 (**6**)
F_{ODF}	⟨mak⟩	0.65365 (**10**)	0.39098 (**9**)	0.62120 (**9**)	0.23940 (**4**)	0.36827 (**2**)	0.41889 (**1**)
	⟨min⟩	0.50216 (**10**)	0.34472 (**8**)	0.57048 (**10**)	0.18114 (**5**)	0.31591 (**4**)	0.38854 (**2**)
	⟨ort⟩	0.59904 (**10**)	0.37062 (**8**)	0.60467 (**10**)	0.21129 (**5**)	0.34123 (**3**)	0.40341 (**2**)
	⟨std⟩	0.04611 (**6**)	0.01464 (**6**)	0.01592 (**7**)	0.01893 (**6**)	0.01714 (**6**)	0.00879 (**4**)
F_S	⟨mak⟩	1.32856 (**3**)	0.44687 (**4**)	0.72139 (**4**)	0.20505 (**7**)	0.18299 (**8**)	0.32823 (**7**)
	⟨min⟩	1.19862 (**2**)	0.40253 (**4**)	0.70403 (**4**)	0.14951 (**7**)	0.13780 (**8**)	0.27751 (**7**)
	⟨ort⟩	1.26548 (**2**)	0.42827 (**2**)	0.71283 (**4**)	0.17788 (**7**)	0.15998 (**8**)	0.29998 (**7**)
	⟨std⟩	0.03947 (**5**)	0.01415 (**5**)	0.00523 (**3**)	0.01608 (**5**)	0.01310 (**5**)	0.01651 (**7**)
F_{Sr}	⟨mak⟩	1.21899 (**4**)	0.44943 (**2**)	0.73369 (**2**)	0.25402 (**3**)	0.23970 (**6**)	0.34638 (**5**)
	⟨min⟩	1.09758 (**4**)	0.40622 (**3**)	0.72109 (**2**)	0.21545 (**2**)	0.21063 (**5**)	0.31192 (**5**)
	⟨ort⟩	1.16252 (**4**)	0.43090 (**1**)	0.72822 (**2**)	0.23613 (**2**)	0.22627 (**5**)	0.33011 (**5**)
	⟨std⟩	0.03724 (**4**)	0.01351 (**4**)	0.00391 (**2**)	0.01175 (**2**)	0.00876 (**2**)	0.01112 (**5**)

Çizelge 4.17 ile ilgili tüm açıklamalara ek olarak amaç fonksiyonları ile önerilen ağ modüllerinin kalite testlerinde, farklı değerlendirme ölçütlerine göre bazen oldukça farklı değerlendirme sonuçlarına ulaşıldığı gözlemlenmiştir. Bu durum değerlendirme fonksiyonlarının genellikle farklı matematiksel yaklaşımlara sahip olduklarını gösterir ve bu, ilgili sonuçların karşılaştırılmasında nesnel değerlendirmelerin yapılmasını zorunlu kılar. Aynı zamanda böyle bir durum, farklı alt-kümelerin kalite testlerinin gerçekleştirilmesinde birden çok değerlendirme fonksiyonunun sonuçlarının bir arada incelenmesinin önemini de vurgular. Bu amaçla tüm ağlarda gerçekleştirilen testlerin sonuçlarının bir arada değerlendirildiği sonuçlar EK-1, EK-2 ve EK-3'teki şekillerde sunulmuştur. İlk şekilde (EK-1), tüm amaç fonksiyonları için test ağlarına göre elde edilen en yüksek (*mak.*) değerlendirme sonuçlarının ortalamasını ifade eden $\langle mak \rangle$ değerlerinin karşılaştırılması gösterilmiştir. 2. şekilde (EK-2) de benzer karşılaştırma, $\langle ort \rangle$ değerlerine göre yapılmıştır. EK-3'teki şekilde ise Çizelge 4.18'deki bilgilere göre tüm amaç fonksiyonları için ortalama değerlerin ($\langle \text{mak} \rangle, \langle \text{min} \rangle, \langle \text{ort} \rangle$ ve $\langle \text{std} \rangle$) tümünün birden hesaba alındığı en son başarı sıralamaları karşılaştırmalı olarak sunulmuştur.

Çizelge 4.18. $\langle mak \rangle$, $\langle min \rangle$, $\langle ort \rangle$ ve $\langle std \rangle$ değerlerine göre fonksiyonların ortalama başarı sıralamaları

	E_{MI}	E_{NMI}	E_{RI}	E_{ARI}	E_{JI}	E_P	Ortalama	Sıralama
F_M	4.75	2.5	2	1.75	3.25	3.5	2.9583	1
F_C	6.5	6	7.5	4	3	2.5	4.9167	4
F_{ID}	6.5	8.25	7.5	9.5	8.75	9	8.2500	9 [a]
F_{GD}	3.75	4.75	4.75	8	8.75	8	6.3333	8
F_{CR}	8.25	6.75	8.5	3.75	2.75	1.75	5.2917	6
F_{NC}	7.5	9.5	7	8.75	6.75	10	8.2500	9 [b]
F_{FS}	1.75	3.25	3	5.5	6.25	6	4.2917	3
F_{ODF}	9	7.75	9	5	3.75	2.25	6.1250	7
F_S	3	3.75	3.75	6.5	7.25	7	5.2083	5
F_{Sr}	4	2.5	2	2.25	4.5	5	3.3750	2

a ve *b* konumları aynı ortalama başarı sıralamasına sahip iki amaç fonksiyonunu temsil eder.

Çizelge 4.18'de her bir değerlendirme ölçütüne göre amaç fonksiyonları için elde edilen ortalama başarı sıralamaları ile bu ortalama değerlere göre 1'den 9'a kadar atanan başarı sıralamaları gösterilmiştir. Tablodaki *'Ortalama'* sütununda her bir amaç fonksiyonu için nihai ortalama sıralamalar verilmiştir. Örneğin; F_M amaç fonksiyonu ile elde edilen ortalama başarı, 2.9583'tür. Bu değer, F_M'nin $\langle mak \rangle$, $\langle min \rangle ort \rangle$ ve $\langle std \rangle$ değerlerine göre elde edilmiş olan başarı sıralamasını gösterir. F_M fonksiyonu için

E_{MI} ölçütüne göre Çizelge 4.17'de parantez içinde gösterilen sıralamalar ile ortalama değerler sırasıyla; 0.98818 (6), 0.88483 (5), 0.92978 (6) ve 0.03239 (2)'dir. Parantez içindeki sayıların ortalaması 4.75'tir ve bu değer, F_M için E_{MI}'ya göre ortalama başarı değerini temsil eder. Aynı şekilde F_M fonksiyonu için diğer ölçütlere göre de ortalama başarı değerleri belirlenir. Buna göre modülerlik amaç fonksiyonunun değerlendirme ölçütlerine göre tüm başarı değerleri şunlardır: 4.75, 2.5, 2, 1.75, 3.25 ve 3.5'tur. Bunların ortalama değeri ise 2.9583'tür. Bu şekilde tüm amaç fonksiyonları için ortalama başarı değerleri hesaplanır. Daha sonra da düşükten yükseğe doğru bu fonksiyonlar temsili olarak sıralanırlar. Sonuç olarak, en düşük (*en uygun*) başarı değerine sahip F_M fonksiyonu, en uygun amaç fonksiyonu olarak etiketlenir. Tüm amaç fonksiyonları için belirlenen en son başarı sıralamaları ise şu şekilde olur: F_M, F_{Sr}, F_{FS}, F_C, F_S, F_{CR}, F_{ODF}, F_{GD}, F_{ID}, F_{NC}. Çizelge 4.18'de son iki fonksiyonun (F_{ID} ve F_{NC}) ortalama başarı değerleri aynıdır. Bu yüzden ikisinin de başarı sıralaması 9 olarak alınmıştır. Tablodaki başarı değerleri ile Yukarıda verilen tüm çizelgelerdeki değerlendirme sonuçları dikkatlice incelendiğinde, F_M fonksiyonunun her ne kadar başarı sıralaması 1 olsa da genel olarak en iyi sonuçların sadece bu amaç fonksiyonuyla elde edilmediği anlaşılmıştır. Hatta bazı durumlarda modülerlik amaç fonksiyonuyla diğer fonksiyonlara göre daha kötü sonuçlar elde edilmiştir. Ayrıca başarı sıralaması 2 olan F_{Sr} fonksiyonu, F_M'e yakın sonuçlara ulaşmıştır. Tüm bu sonuçlara göre ağ modüllerinin tespiti probleminde algoritmanın uygunluk kriteri olarak, modülerlik amaç fonksiyonunun seçilmesi yanında F_{Sr}, F_{FS}, F_C ve F_S gibi uygun fonksiyonların da bazı durumlarda tercih edilebileceği anlaşılmıştır.

Bölüm 4.1.1'de karşılaştırma sonuçları sunulan ve test sonuçları değerlendirilen 8 adet metasezgisel algoritmadan en başarılı olanı *3pHybrid* algoritmasıdır. Ayrıca Bölüm 4.1.2'de ayrıntılı olarak analiz edilen 10 farklı amaç fonksiyonu, *3pHybrid* algoritması için uygunluk fonksiyonları olarak kullanılmıştır. Bu bölümde verilen tüm deney sonuçları incelendiğinde, modülerlik fonksiyonunun bir sonraki karşılaştırma testlerinde kullanılmasının uygun olduğu anlaşılabilir. Ayrıca en popüler kalite fonksiyonun da modülerlik fonksiyonu olduğu, *Fortunato* tarafından belirtilmiştir (Fortunato 2010). Bu yüzden topluluk tespitinde en çok kullanılan amaç fonksiyonunun modülerlik kriteri olması da karşılaştırmada bu fonksiyonun seçilmesinin diğer bir temel sebebidir. Böylece uygunluk kriteri olarak, F_M modülerlik amaç fonksiyonunu kullanan *3pHybrid* algoritması aşağıda sunulan gerçek dünya ağlarında test edilmiştir.

Elde edilen sonuçlar, literatürde daha öncesinden önerilmiş bazı algoritmaların sonuçları ile karşılaştırılmıştır. Bu sonuçlar Çizelge 4.19'da sunulmuştur. Burada her bir algoritmanın tüm çalıştırmalar sonunda elde ettiği en iyi modülerlik skorları verilmiştir. Sayıların kesirli kısımları dört ondalık basamağa yuvarlanmıştır. Ayrıca tabloda bazı sonuçların kesirli kısımları 3 haneli olarak gösterilmiştir. Seçilen birkaç algoritmanın sonucu ilgili çalışmalarda bu şekilde verildiğinden dolayı karşılaştırma tutarlılığı açısından tüm sonuçlar benzer şekilde gösterilmiştir. Böylece Çizelge 4.19'da hem '1' ile etiketlenen ağlardaki sonuçların tümünde hem de '2' ile etiketlenen algoritmaların sonuçlarında modülerlik skorlarının kesirli kısımları üç ondalık basamağa yuvarlanmıştır. Bir diğer durumda, *karate* ağındaki test sonuçları için hem * ile belirtilen 0.420 skorları hem de 0.4198 ile gösterilen skorlar koyu renkle vurgulanmıştır. Çünkü her iki sayının da yuvarlama sonrası üç ondalık basamak değerleri aynı sonucu (*0.420*) gösterir. Buna benzer durumlar *c.elegans* ve *email* ağlarının sonuçları için de geçerlidir. Örneğin bu ağlardaki testler için sırasıyla; *iMod* algoritmasının 0.453 ve 0.580 olan değerleri ile *3pHybrid* algoritmasının 0.4525 ve 0.5749 olan değerleri benzer şekilde koyu renklerle vurgulanmıştır. Diğer algoritmalar için aynı işlemler tekrarlanmıştır. Çizelge 4.19'daki algoritmalarının temsili isimleri ile referansları şöyledir: CC–GA (Said ve ark 2018), CLPA–GNR (Chin ve Ratnavelu 2016), CM–Net (Zalik KR ve Zalik B 2017), CNM (Clauset ve ark 2004), FU (*veya Louvain*) (Blondel ve ark 2008), FWA (Guendouz ve ark 2017), GA–Net (Pizzuti 2008), HC (*Hierarchical Clustering*) (Xu ve ark 2007), iMod (Xu ve ark 2010), Infomap (Rosvall ve Bergstrom 2008), LSA–LPA (Barber ve Clark 2009, Wu ve Pan 2015), MA–Net (Naeni ve ark 2015), Meme–Net (Gong ve ark 2011), MMCD (Wu ve Pan 2015), MOA (Wu ve Pan 2015), MOCD (Shi ve ark 2012), MOGA–Net (Pizzuti 2012), OptMod (Xu ve ark 2007), RN (Ronhovde ve Nussinov 2010). Ayrıca ağlarla ilgili düğüm/bağlantı sayıları ve referans numaraları çizelgenin altında verilmiştir. Tüm test ağlarındaki düğüm ve bağlantı sayıları, ağların hem ağırlıksız hem de yönsüz olarak ayarlanması sonrası belirlenmiştir. Bunlara ek olarak, döngülü düğümler de bu ağlardan çıkarılmıştır.

Çizelge 4.19'da, 13 gerçek dünya ağı üzerinde testler gerçekleştiren *3pHybrid* algoritması ile birlikte toplam 21 adet algoritma seçilmiştir. Genellikle, verilen algoritmalardan her biri ağ topolojisinin farklı bir özelliğine göre çözüm önermektedir. Ağlarda gerçekleştirilen testler sonunda elde edilen en iyi modülerlik skorları koyu renkle vurgulanmıştır. *c.elegans* ağında en iyi skorlara 0.453 ile *iMod* ve 0.4525

(yuvarlama sonunda *0.453*) ile *3pHybrid* ulaşmıştır. *dolphins* ağındaki en iyi sonuç *3pHybrid* algoritmasına aittir ve bu sonuç 0.5294'tür. Diğer algoritmaların bu ağdaki testlerde elde ettikleri skorlar birbirlerine oldukça yakındır. Üçüncü ağda—*e.coli*'de skorların kesirli kısımları üç hane olarak ifade edilmiştir ve yedi tane algoritmanın test sonucu verilmiştir. Bunlardan *3pHybrid* ile yakın zamanda önerilmiş *CC–GA*'nın sonuçları birbirlerine yakındır (*0.787—0.786*). Sonraki en iyi sonuç *iMod* ile elde edilmiştir. *football* test ağında kaydedilen en yüksek modülerlik skoru 0.6058'dir. Bu sonuca tezde önerilen algoritma ile *CM–Net* algoritması birlikte ulaşmıştır. *jazz* ağında ise en iyi sonucu sadece *3pHybrid* algoritması vermiştir. *karate* ağında düğüm/bağlantı sayılarının az olmasından dolayı çoğu algoritma en iyi sonuca ulaşmıştır. Bu sonuç 0.4198'dir. *lesmis* gerçek dünya ağındaki deney sonuçlarına göre *3pHybrid* algoritması en iyi sonuç olan 0.5667 skoruna ulaşmıştır. Bir sonraki en yüksek skor ise 0.5600'dür ve bu skora *iMod*, *MMCD*, *MOCD* ve *OptMod* algoritmaları ulaşmıştır. Çizelge 4.19'daki *netscience* ve *power* ağlarındaki testlerde en iyi sonuçları *FU* algoritması elde etmiştir. Bu algoritma, bazı çalışmalarda *Louvain* ismi ile de temsil edilmektedir. *netscience* ve *power* ağları için en iyi skorlar sırasıyla 0.9592 ve 0.935'tir. *netscience* için ikinci en iyi değer 0.9581'tir ve bu değere *Infomap* ve *3pHybrid* algoritmaları ulaşmıştır. *power* ağı için ise sonraki en yüksek skor 0.934'tür ve bu skora *3pHybrid* ile birlikte *CNM* algoritması da ulaşmıştır. *polbooks* ağındaki testlerde *CM–Net*, *MMCD*, *MOCD* ve *3pHybrid* algoritmaları en iyi sonuçları vermiştir.

p53 protein etkileşim ağındaki testlerin sonuçlarıyla ilgili sadece 5 algoritmanın skorları tabloda sunulmuştur. Bunlardan *iMod*, *OptMod* ve *3pHybrid* algoritmaları en yüksek skora—0.535 ulaşmışlardır. En kötü sonucu (*0.458*) ise hiyerarşik kümeleme yaklaşımı sunan *HC* algoritması elde etmiştir. *words* ağında gerçekleştirilen denemeler sonunda 0.3121 modülerlik skoru ile en yüksek değeri, önerilen *3pHybrid* algoritması elde etmiştir. Sonuçlara göre sonraki en yüksek değere (*0.3082*) *MMCD* algoritması ulaşmıştır. En kötü sonuç ise *Infomap* algoritmasına aittir.

Ağ topluluklarının tespitiyle ilgili Bölüm 4.1'deki tüm sonuçlara göre *3pHybrid* algoritmasının elde edilen modülerlik skorları genellikle 0.3 – 0.7 arasındadır. Bu konuda *Newman* tarafından sunulan çalışmada (Newman 2004 (a)), modülerlik değerlerinin *0.3* ile *0.7* arasında olması durumunda toplulukların normal anlamlılık düzeyini sağladığı ve bunların güçlü-anlamlı (*significant community structure*) toplulukları ifade ettiği belirtilmiştir (Steinhaeuser ve Chawla 2008). Çizelge 4.19'daki karşılaştırmalı test sonuçları incelendiğinde, 13 ağdan ikisi hariç diğerlerinde *3pHybrid*

algoritması ya tek başına ya da birkaç algoritma ile en yüksek modülerlik skorlarına ulaşmıştır. Önerilen algoritmanın deneylerdeki başarısı sebebiyle sonraki problemde kanserle ilişkili testlerde kullanılan biyolojik ağlardaki ağ modüllerinin tespitinde bu algoritma kullanılmıştır. Aynı şekilde bu bölümde verilen tüm sonuçlara göre ortalama değerlendirme başarıları açısından en iyi sıradaki F_M—modülerlik fonksiyonu, algoritmanın uygunluk kriteri olarak seçilmiştir.

Çizelge 4.19. *3pHybrid* algoritması ile diğer algoritmaların en iyi modülerlik skorları açısından kıyaslanması

	c.elegans	dolphins	email	e.coli[1]	football	jazz	karate	lesmis	netscience	polbooks	power[1]	p53[1]	words
CC–GA[2]	—	0.529	—	0.786	0.594	0.444	**0.420***	—	0.958	0.527	—	—	—
CLPA–GNR[2]	—	0.513	0.520	0.749	0.601	0.282	0.303	—	—	0.514	0.888	—	—
CM–Net	—	0.5285	—	—	**0.6058**	0.4451	**0.4198**	—	—	**0.5272**	—	—	—
CNM	0.4061	0.4955	0.5130	—	0.5772	0.4389	0.3807	0.5006	0.9555	0.5020	0.934	0.521	0.2953
FU	0.4320	0.5188	0.5720	—	0.6043	0.4451	0.4188	0.5583	**0.9592**	0.5268	**0.935**	—	0.2886
FWA	—	0.5093	—	—	0.6046	—	**0.4198**	—	—	0.5271	—	—	—
GATHB[2]	—	0.522	—	—	0.551	—	0.402	—	—	0.518	—	—	—
GA–Net	0.3082	0.4950	0.3122	—	0.5798	0.4049	**0.4198**	0.5402	—	0.5076	—	—	0.2199
HC	—	0.5084	—	—	—	—	**0.4198**	0.5000	—	—	—	0.458	—
iMod[2]	**0.453***	0.529	**0.580***	0.781	—	0.445	**0.420***	0.560	—	—	—	**0.535**	—
Infomap	0.4194	0.5230	0.5340	0.707	0.6005	0.4423	0.4020	0.5513	0.9581	0.5268	0.818	—	0.0333
LSA–LPA	0.3993	0.5107	0.5170	0.771	0.6031	0.4448	**0.4198**	0.5357	0.8691	0.5185	0.627	—	0.2782
MA–Net[2]	—	0.529	—	—	0.605	0.445	**0.420***	—	—	0.527	—	—	—
Meme–Net	—	0.5164	—	—	0.6031	0.4386	0.4020	0.5397	—	0.5248	—	—	0.1198
MMCD	0.4415	0.5285	**0.5749**	—	0.6046	0.4451	**0.4198**	0.5600	0.9394	**0.5272**	0.827	—	0.3082
MOA	0.3825	0.5285	0.5219	—	0.6044	0.4430	**0.4198**	0.5537	0.9394	0.5271	0.680	—	0.2526
MOCD	0.3931	0.5285	0.4771	—	0.6021	0.4374	**0.4198**	0.5600	0.9410	**0.5272**	0.738	—	0.2810
MOGA–Net	0.2728	0.5253	0.3475	—	0.5960	0.2952	0.4156	0.5476	—	0.5267	—	—	0.1746
OptMod	—	0.5285	—	—	—	—	**0.4198**	0.5600	—	—	—	**0.535**	—
RN[2]	—	0.379	—	0.771	0.601	0.288	0.406	—	—	0.527	—	—	—
3pHybrid	**0.4525**	**0.5294**	**0.5749**	**0.787**	**0.6058**	**0.4452**	**0.4198**	**0.5667**	0.9581	**0.5272**	0.934	**0.535**	**0.3121**

'1' ile etiketlenen ağlardaki sonuçların tümünde ve **'2'** ile etiketlenen algoritmaların sonuçlarında modülerlik skorlarının kesirli kısımları üç haneye yuvarlanmıştır.

- Ağların düğüm—bağlantı sayıları ile referansları: $c.elegans\ (453—2025)$ (Jeong ve ark 2000, Duch ve Arenas 2005); $dolphins\ (62—159)$ (Lusseau ve ark 2003, dolphins 2017); $email\ (1133—5451)$ (Guimera ve ark 2003); $e.coli\ (418—519)$ (Shen-Orr ve ark 2002); $football\ (115—613)$ (Girvan ve Newman 2002); $jazz\ (198—2742)$ (Gleiser ve Danon 2003); $karate\ (34—78)$ (Zachary 1977); $lesmis\ (77—254)$ (Knuth 1993); $netscience\ (1589—2742)$ (Newman 2006 (b)); $polbooks\ (105—441)$ (polbooks 2017)(polbooks 2017); $power\ (4941—6594)$ (Watts ve Strogatz 1998); $p53\ (104—226)$ (Dartnell ve ark 2005); $words\ (112—425)$ (Newman 2006 (b)).

4.2. Kanserde Sağkalım Süresiyle İlişkili Alt Ağların Bulunması Deneyleri

Kanserle ilişkili *CNA*'lı genlerin sağkalım analizi ile tespit edilmesi için önerilen *DPT* ve *GAT* algoritmaları *C++* programlama dili ile kodlanmıştır. Ağ yapılarının çizgelerle modellenmesi ve ağlardaki genler arası ilişkilerin uygun bir şekilde analizi için *SNAP* kütüphaneleri tercih edilmiştir. *SNAP*, farklı programlama dillerinde mevcut olan ve özellikle büyük ağların analizi için önerilen kapsamlı kütüphaneler içerir (SNAP 2016). Bu tez çalışmasında, sunulan algoritmalar için *C++ SNAP* kütüphanesindeki uygun fonksiyonlardan yararlanılmıştır. Bu bölümde kullanılan gerçek dünya verilerinin ayrıntılı açıklamaları Bölüm 3.3'te yapılmıştır. Testlerde *DPT* algoritması $k = 4, 5, 6$ sayılarına ve $\varepsilon = 0.001$ parametre değerine göre farklı sayılarda çalıştırılmıştır. Buna göre bu algoritma yaklaşık olarak en uygun log-rank skorlarına ulaşmak için sırasıyla; $k = 4$ için 70 kez; $k = 5$ için 176 kez; $k = 6$ için 444 kez çalıştırılmıştır. Buradaki k değerlerine göre hesaplanan *DPT* algoritmasının uygun çalıştırılma sayıları Bölüm 3.5.1'in sonunda sunulan formüller ile belirlenmiştir. *GAT* algoritması ise farklı $k = 4, 5, 6, 7, 8, 9, 10$ değerleri için her bir kanser türüne göre *Cytoscape* (Shannon ve ark 2003, Cytoscape 2017) programı ve *STRING* (string-db 2018) yardımıyla oluşturulan biyolojik ağlarda 20 kez çalıştırılmıştır. *DPT*'nin sadece ilk üç k değerine göre çalıştırılmasının sebebi bu algoritmanın süre kısıtlamasıdır. Çünkü eğer bu algoritma diğer değerler için de test edilmek istenirse; $k = 7$ için 1125 kez; $k = 8$ için 2870 kez; $k = 9$ için 7371 kez ve $k = 10$ için 19032 kez çalıştırılmalıdır. Bu da gerekli test sürelerinin logaritmik olarak ciddi bir şekilde artması anlamına gelir.

Üç farklı k değeri için elde edilen *DPT* ve *GAT* algoritmalarının karşılaştırmalı sonuçları Bölüm 4.2.1.1'de açıklanmıştır. Ayrıca farklı hasta—gen sayılarına sahip ve rastsal olarak üretilmiş yapay test verileri aynı bölümde sunulmuştur. Rastsal olarak üretilen test verilerinin temsili etkileşim ağları *ER* modeli ile oluşturulmuştur. Tüm yapay verilerle ve ağlarla ilgili özellikler Çizelge 4.20'de sunulmuştur. Ağ verilerinin genel özellikleri yine bu bölümde verilmiştir. Böylece hem gerçek hem de yapay verilerdeki testlerle önerilen algoritmaların performansları karşılaştırılmıştır. Bu karşılaştırma, algoritmaların çalışma sürelerini ve nihai log-rank skorlarını kapsar.

Bölüm 4.2.1.1'de, rastsal olarak üretilen üç yapay test verisi (*S-1*, *S-2* ve *S-3*) ile *GBM* gerçek dünya verisinin $k = 4, 5, 6$ değerlerine göre kaydedilen test sonuçları

verilmiştir. Bölüm 4.2.1.2'de ise daha büyük ve daha karmaşık gen etkileşim ağlarına sahip klinik veri setlerinde—*LAML*, *HNSC*, *KIRC*, *STAD* gerçekleştirilen deneylerin $k = 4, 5, 6, 7, 8, 9, 10$ değerlerine göre sonuçları verilmiştir.

Son olarak, Bölüm 2.3'te açıklanan üçüncü probleme (P^3) göre dört karmaşık etkileşim ağının (*LAML-net*, *HNSC-net*, *KIRC-net* ve *STAD-net*) modül yapıları ortaya çıkartıldıktan sonra önceki bölümde elde edilmiş olan gen gruplarından aynı modüllerde (*en fazla üyelere sahip olan modüller*) bulunan gen listeleri kaydedilmiştir. Daha sonra bu listelerdeki genler, "*http://cbdm-01.zdv.uni-mainz.de/~jfontain/cms/?page_id=592*" isimli web aracı yardımıyla analiz edilmiş ve aynı modüllerdeki genlerin etkiledikleri biyolojik değişiklikler ya da hastalıklar Bölüm 4.2.2'deki tablolarda sunulmuştur.

4.2.1. Gen etkileşim ağlarından elde edilen test sonuçları

Bu bölüm, "*DPT ve GAT algoritmalarının karşılaştırmalı test sonuçları*" ve "*Diğer klinik test verileri üzerindeki deney sonuçları*" isimli 2 alt başlıkta incelenmiştir.

4.2.1.1. DPT ve GAT algoritmalarının karşılaştırmalı test sonuçları

Burada *DP* ve *GA* temelli iki farklı yaklaşımın karşılaştırmalı test sonuçları verilmiştir. Bu karşılaştırma, *DPT*'nin işlem süresi kısıtları sebebiyle sadece $k = 4, 5, 6$ değerlerindeki sonuçlara göre yapılmıştır. Çizelge 4.20'de hem *DPT* hem de *GAT* algoritmalarının performanslarının kıyaslamalı test edileceği yapay verilerin genel özellikleri sunulmuştur. Bu çizelgede, *ER* modeli kullanılarak üretilen 250, 500 ve 1000 düğümlü (*bunlar genleri temsil ederler*) ağlar ve bunlarla ilgili yine rastsal olarak oluşturulan hasta bilgileri verilmiştir. Burada yapay veriler; "*S-1*", "*S-2*" ve "*S-3*" ile isimlendirilmiştir. Benzer şekilde bu veriler için üretilen rastsal ağlar "*S-1-net*", "*S-2-net*" ve "*S-3-net*" ile gösterilmiştir.

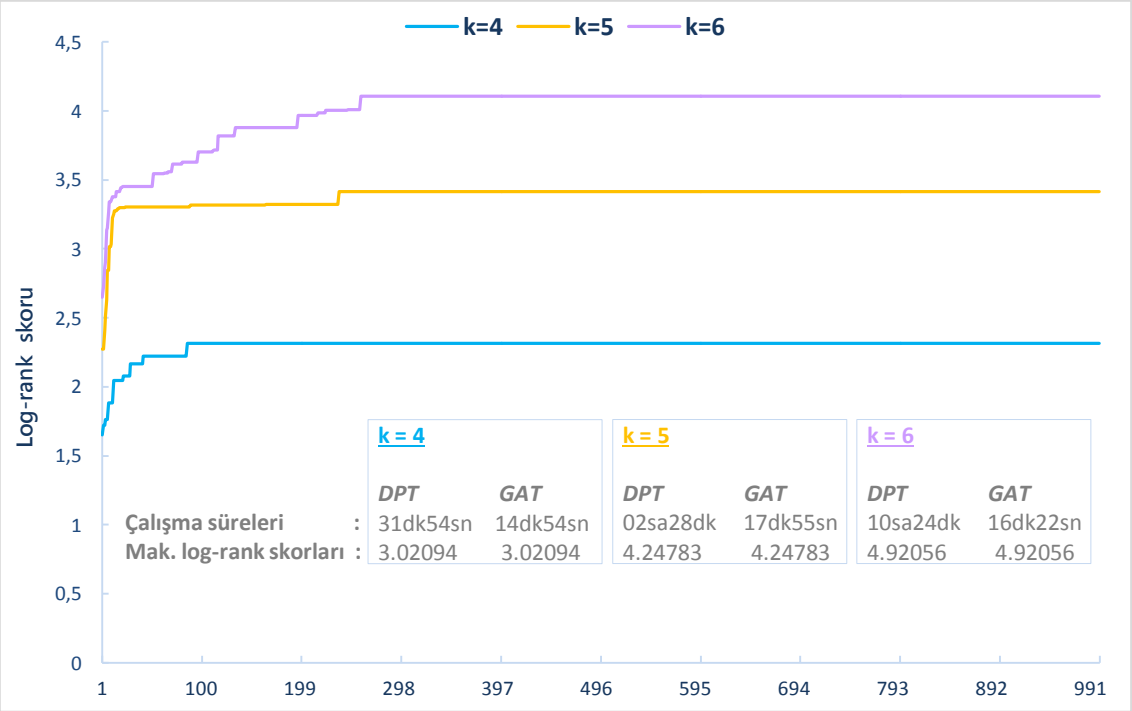
Çizelge 4.20'deki *ortalama* (*ort.*) *CNA sayısı*, hastalarda bulunan toplam *CNA* 'lı genlerin ortalamasını ifade eder. *Ort. sağkalım süresi* parametresi ise günlük bazda hastaların sağkalım sürelerinin ortalamasını verir. Sansürlü ve sansürsüz bilgiler ise Bölüm 2.2'deki açıklamalara göre hastaların sansür bilgisinin $c = 0$ olması durumu sansürlü; $c = 1$ olması durumu ise sansürsüz bilginin mevcut olduğunu gösterir. Böylece aşağıdaki tabloda verilen farklı özelliklere sahip üç adet giriş verisi dikkate alınarak; düşük k değerlerine göre karşılaştırmalı testler gerçekleştirilmiştir.

Çizelge 4.20. Rastsal olarak oluşturulan test verileri

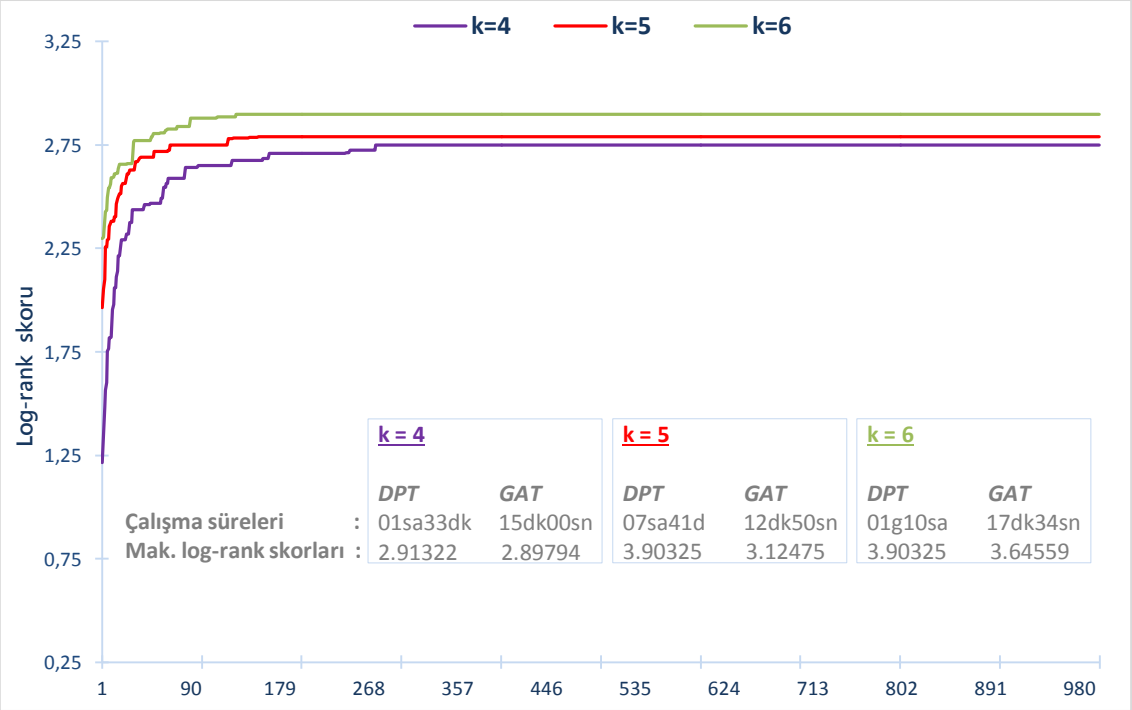
Yapay veri ismi	*Hasta sayısı*	*Gen sayısı*	*Bağlantı sayısı*	*Ort. CNA sayısı*	*Ort. sağkalım süresi*	*Sansürlü bilgi sayısı*	*Sansürsüz bilgi sayısı*
S-1	200	250	249	223.76	460.20	103	97
S-2	200	500	499	348.91	503.69	100	100
S-3	300	1000	999	649.16	498.09	153	147

Bu bölümde, Çizelge 4.20'de sunulan rastsal olarak üretilmiş verilerde ve *GBM* klinik verisinde $k = 4, 5, 6$ değerlerine göre testler gerçekleştirilmiştir. Gerçekleştirilen testlerin hem *DPT* hem de *GAT* algoritmalarına göre sonuçları sırasıyla; *S-1* için Şekil 4.15'te; *S-2* için Şekil 4.16'da; *S-3* için Şekil 4.17'de ve *GBM* için Şekil 4.18'de verilmiştir.

Aşağıdaki şekillerde algoritmaların test verileri üzerindeki çalışma süreleri ve elde edilen en iyi log-rank skorları, ilgili k değerlerine göre sonuçların verildiği renkli alanlarda gösterilmiştir. Çalışma sürelerinin gösterimi için günler, g; saatler, sa; dakikalar, dk; saniyeler, sn kısaltmaları ile temsil edilmiştir. Bu şekillerde, *GAT* algoritmasının 20 kez çalıştırılması sonucu elde edilen maksimum log-rank skorlarına göre her bir iterasyona göre değişen ortalama skorlar verilmiştir. Burada maksimum iterasyon sayısı 1000 olarak belirlenmiştir. Her bir k değerine göre kaydedilen sonuçlar farklı renklerle sunulmuştur. Örneğin; Şekil 4.15'teki $k = 4$ değerinin sonuçları açık mavi renkle vurgulanmıştır. Bu şekilde *DPT* algoritması ile 3.02094 skoru, 31 dakika 54 saniyede elde edilmişken; *GAT* algoritması ile aynı skora 14 dakika 54 saniyede ulaşılmıştır. $k = 5$ değerine göre sonuçlar, turuncu renkle vurgulanarak gösterilmiştir. Buna göre iki algoritma da 4.24783 skorunu elde etmişlerdir. Bu skora *DPT* ile 2 saat 28 dakikada; *GAT* ile 17 dakika 55 saniyede ulaşılmıştır. Son olarak, $k = 6$ değerinin sonuçları ise açık mor renkle temsil edilmiştir. Önceki skorlarda olduğu gibi yine iki algoritma aynı log-rank değerine (*4.92056*) sahiptir. *S-1* verisinde sadece 200 hasta olmasına ve *S-1-net* ağında sadece 250 düğüm ve 249 bağlantı olmasına rağmen; 6 bağlı gen grubunun tespitinde $(k = 6)$, *DPT* algoritması uygun sonuca 10 saat 24 dakikada ulaşmıştır. *GAT* ile sadece 16 dakika 22 saniyede aynı sonuç elde edilmiştir. Tüm k değerlerine göre *GAT* algoritmasının sonuçlarına dikkat edildiğinde, 300. iterasyondan önce en uygun sonuçlara erişilmiştir. Bu yüzden bu algoritmanın genellikle daha kısa sürede en iyi log-rank skorlarına ulaştığı anlaşılmıştır.



Şekil 4.15. S-1 veri seti için *DPT* ve *GAT* algoritmalarının karşılaştırmalı test sonuçları

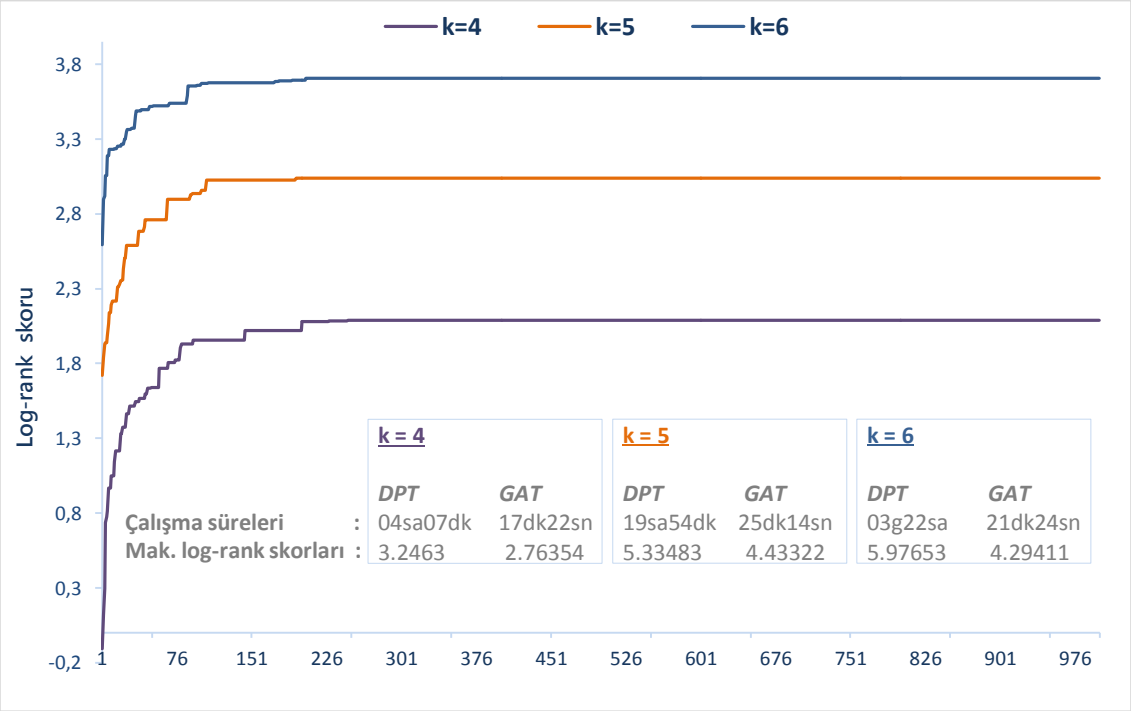


Şekil 4.16. $k = 4, 5, 6$ değerleri için algoritmaların S-2 test verindeki karşılaştırmalı test sonuçları

Şekil 4.16'da verilen *S-2* veri setindeki deneysel sonuçlar, $k = 4$ için mor renkte; $k = 5$ için kırmızı renkte; $k = 6$ için açık yeşil renkte gösterilmiştir. Bu test

verisinde $k = 4$ için *DPT*, 1 saat 33 dakikada 2.91322 skorunu; *GAT* ise 15 dakikada 2.89794 skorunu elde etmiştir. Şekil 4.16'da kırmızı renkle vurgulanan $k = 5$ sonucu için *DPT* ile 7 saat 41 dakikada 3.90325 değeri; *GAT* ile 12 dakika 50 saniyede 3.12475 değeri elde kaydedilmiştir. Bir diğer k değeri için 6 boyutlu alt-ağ tespitinde, *GAT* algoritması ile sadece 17 dakika 34 saniyede 3.64559 sonucuna ulaşılmışken; *DPT* ile yaklaşık olarak 34 saatte 3.90325 sonucuna ulaşılmıştır.

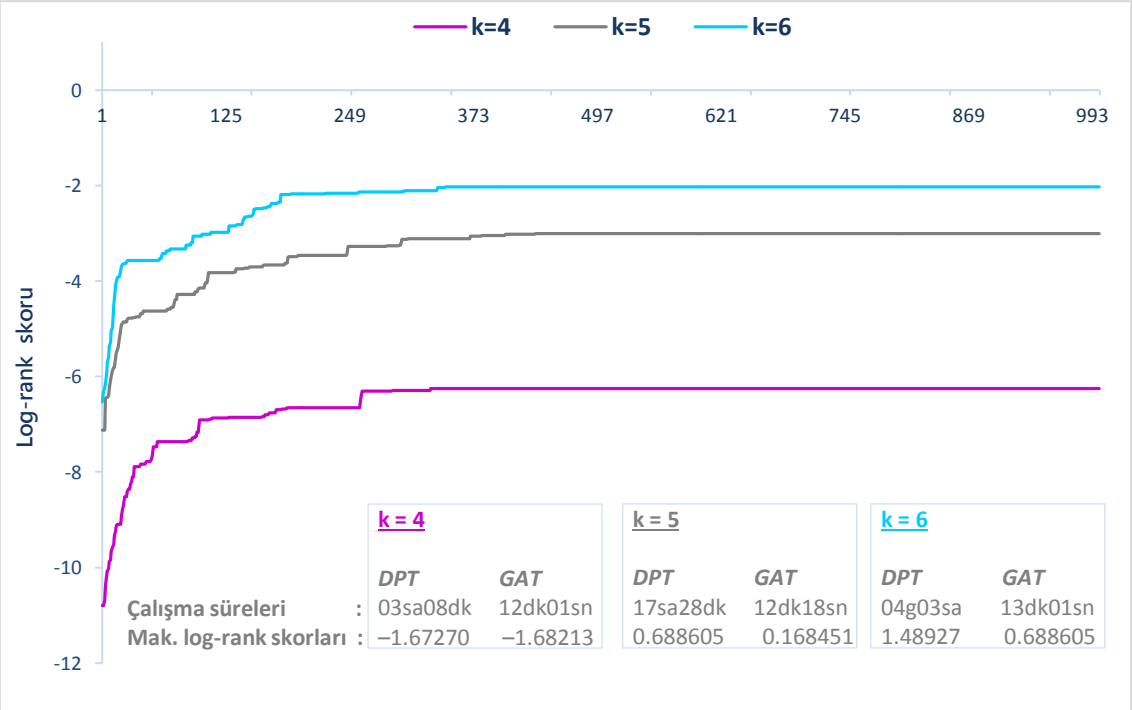
Şekil 4.17'de sunulan *S-3* yapay veri setinin tüm sonuçlarına göre *GAT* algoritması en son log-rank değerlerine yarım saatten önce ulaşmışken; *DPT* ile sonuçlar çok daha uzun sürelerde elde edilmiştir. Hatta $k = 6$ için *GAT* algoritması ile 4.29411 skoru yaklaşık olarak 22 dakikada elde edilirken; DPT algoritması ile 5.97653 değerine 3 gün 22 saatte ulaşılmıştır. Bu skor değeri, *GAT* algoritması ile ulaşılan değerden daha iyi olmasına rağmen çalışma süresi açısından *DPT* algoritmasının performansının çok daha kötü olduğu sonucuna varılmıştır.



Şekil 4.17. *GAT* ve *DPT* algoritmalarının S-3'te gerçekleştirilen deney sonuçları

Son olarak bu alt bölümde, *GBM* klinik kanser verisinde gerçekleştirilen testlerin sonuçları verilmiştir. Bu sonuçlar Şekil 4.18'de gösterilmiştir. Bu grafikte tüm k değerlerinin sonuçları farklı renklerle vurgulanmıştır. $k = 4$'e göre algoritmalara ait skorlar ve çalışma süreleri sırasıyla; *DPT* için -1.67270 ile 03sa08dk ve *GAT* için

-1.68213 ile 12dk01sn'dir. $k = 5$ ve $k = 6$ için *DPT* algoritması sırasıyla; 0.688605 ve 1.48927 skorlarını, 17 saat 28 dakika ve 4 gün 3 saat gibi uzun sürelerde ancak elde edebilmiştir. Buna karşılık *GAT* algoritması bu skorlara yakın değerleri çok daha kısa sürelerde elde etmiştir. Bu bölümdeki tüm test sonuçları incelendiğinde, her ne kadar *DPT* algoritması çoğu testlerde en iyi log-rank skoruna ulaşmış olsa da kıyaslamada süre kısıtı da bir diğer kriter olarak alındığında, bu algoritmanın buradaki test ağlarından daha karmaşık ağlarda işleme alınmasının oldukça zor olduğu anlaşılmıştır. Hatta bazı durumlarda (örneğin $k = 10$ gibi) neredeyse bu algoritmanın uygun değerlere ulaşabilmesi için aylarca çalıştırılması gerekir. Buna bir örnek olarak bir sonraki bölümde sunulan *LAML* ağındaki test sonuçları verilebilir. Bu veri setinde hem *GAT* hem de *DPT* algoritmaları ile $k = 4$ değerine göre testler gerçekleştirilmiştir. Buradaki klinik veri setinde 191 hasta, 10161 gen/protein ve 229867 adet bağlantı mevcuttur (ayrıntılı bilgi için Bölüm 3.3'e bakınız). *DPT* algoritmasının sadece $k = 4$ değerine göre sonuç alma süresi, en az 18 gün 8 saattir. *GAT* ise aynı log-rank değerine, ortalama olarak sadece 18 dakika 16 saniyede ulaşmıştır. Bu sonuçlar algoritmaların performansları arasındaki ciddi farklılıkları göstermiştir. Bu yüzden sonraki deneylerde *DPT* algoritması yerine sadece *GAT* algoritması kullanılmıştır. Buradaki sonuçlar, bu tür bir problemde *DP* ya da diğer sezgisel yaklaşımlar yerine *GA* gibi metasezgisel yaklaşımların daha uygulanabilir olduğunu göstermektedir.



Şekil 4.18. Farklı k değerlerine göre *GBM* klinik kanser verisinde yapılan testlerin sonuçları

4.2.1.2. Diğer klinik test verileri üzerindeki deney sonuçları

Bu bölümde *LAML*, *HNSC*, *KIRC* ve *STAD* klinik kanser veri setleri ile yapılan testlerin sonuçları verilmiştir. Bu verilerle ilgili ayrıntılı açıklamalar Bölüm 3.3'te yapılmıştır. Buradaki testler sadece *GAT* algoritması ile yapılmıştır. Burada farklı gen sayılarına sahip alt-ağların tespiti için gerçekleştirilen sonuçlar sunulmuştur. Bu alt-ağlardaki gen sayıları sırasıyla; $k = 4, 5, 6, 7, 8, 9, 10$ şeklindedir.

Aşağıda verilen tüm çizelgelerde, *GAT* algoritmasının 20 kez çalıştırılması sonunda elde edilen en iyi ve ortalama log-rank skorları sırasıyla; s^E ve s^O ile temsil edilmiştir. Ayrıca $\bar{t}$ ile bu algoritmanın k değerine göre ortalama çalışma süresi verilmiştir. Burada saatler, sa; dakikalar, dk; saniyeler, sn kısaltmaları ile temsil edilmiştir. Son olarak, ilgili k değerine göre belirlenen en iyi log-rank skorunu veren alt-ağdaki genlerin isimleri ise tablolardaki son sütunlarda verilmiştir. Elde edilen tüm genlerle/proteinlerle ilgili ayrıntılı bilgilere, *"https://string-db.org"* isimli web sitesi üzerinden ulaşılabilir. Bu web platformundaki *"minimum required interaction score"* parametresi, genler/proteinler arasındaki bağlantıların güven (*confidence*) değerini ifade eder. Burada verilen her bir alt-ağdaki genler kendi grubunda en az 0.05 güven değerine göre etkileşimdedir. Bu genlerin oluşturdukları bağlı gruplar, *STRING* web platformu

(Szklarczyk ve ark 2014, Szklarczyk ve ark 2016, string-db 2018) ile alt-ağ yapıları olarak modellenmiştir. Bu genler arası etkileşim bilgileri aynı web platformundaki *textmining*, *experiments*, *databases*, *co-expression*, *neighborhood*, *gene fusion* ve *co-occurrence* isimli aktif etkileşim kaynaklarından temin edilmiştir (string-db 2018).

Çizelge 4.21. LAML veri setinde gerçekleştirilen deneylerin sonuçları

LAML	s^E	s^O	$\bar{t}$	Elde edilen gen listeleri
$k = 4$	–0.565023	–0.642573	18dk16sn	*BCKDK, KAT8, PRSS8, STX1B*
$k = 5$	–0.624177	–0.637637	18dk52sn	*HDAC1, JUN, MYSM1, PLK3, TAL1*
$k = 6$	–0.673574	–0.713521	19dk21sn	*CDC42, EPHA2, EPS15, INPP5B, JUN, LCK*
$k = 7$	–0.673574	–0.734055	21dk04sn	*ARHGEF16, CDC42, EPHA2, INPP5B, JUN, PIK3R3, RSG1*
$k = 8$	–0.711387	–0.734820	18dk41sn	*CDC20, DMAP1, EIF3I, FBXL19, HDAC1, KAT8, PLK3, SRCAP*
$k = 9$	–0.570232	–0.628311	20dk15sn	*CDKN2A, CHL1, EPHA6, JUN, LCK, NLGN1, PTPN1, STX1B, VAPA*
$k = 10$	–0.545172	–0.609729	22dk01sn	*CDKN2B, EBNA1BP2, ENTPD3, HEATR1, ITGAX, JUN, PDE4D, PTPN1, UMPS, YES1*

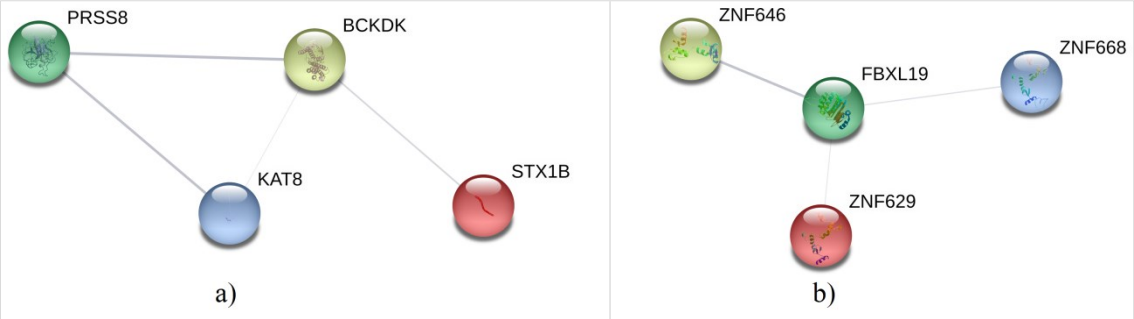
Çizelge 4.21'de tüm farklı k değerlerine göre elde edilen genlerin oluşturduğu etkileşim ağı EK-4'te sunulmuştur. Burada minimum 0.25 güven değeri ile bağlantılar oluşturulmuştur. Ağda toplam 36 adet gen bulunmaktadır ve bu genler *LAML* klinik kanser verisindeki testlerde sağkalım parametresi ile maksimum ilişkili *CNA*'lı genleri ifade eder. Ayrıca, bu genlerin tümünün birden dikkate alındığı analiz sonucu EK-5'te verilmiştir. "*http://cbdm-01.zdv.uni-mainz.de/~jfontain/cms/?page_id=592*" isimli web sitesindeki *GS2D* web aracı yardımıyla tüm genlerin analiz edilmesi sağlanmış ve 36 adet genden bazılarının etkili olduğu biyolojik değişikliklerin ya da hastalıkların listesi sunulmuştur. Bu sonuçlara göre tüm genlerin birtakım biyolojik süreçlerdeki etkilerinin anlaşılması amacıyla bunların birlikte değerlendirilmesinin önemi vurgulanmıştır. Diğer klinik kanser verileri için de aynı web sitesindeki analiz aracı ile ilişkili gen listeleri değerlendirilmiştir. Sırasıyla *LAML*, *HNSC*, *KIRC* ve *STAD* verileri için sonuçlar; EK-5, EK-7, EK-9, EK-11 isimli tablolarda gösterilmiştir. *GS2D* analiz aracı yardımıyla elde edilen genlerin birlikte etkili oldukları düşünülen/farz edilen hastalıklarla ilgili literatüre sunulmuş önemli çalışmalar (*atıflar*) bu tablolarda listeler halinde verilmiştir. Bu listelerin oluşturulmasında, *GS2D*'deki üç önemli parametrenin kullanıcılar tarafından belirlenen değerleri etkili olmaktadır. İlk parametre, bir gen için hastalıklarla ilgili

minimum sayıda alıntı (*min number of disease-related citations for a gene*) miktarını belirler. Bu değer tüm klinik veriler için 5 olarak belirlenmiştir. Eklerde verilen her bir genin üzerindeki sayılar bu genlerle ilgili atıf yapılan çalışmaların toplam sayısını verir. Bir sonraki parametre, bir gen seti için hastalıkla önemli ölçüde ilişkili olan en az sayıda gen (*for a gene set, min number of genes significantly associated with a disease*) miktarını belirler. Bu değer 5 olarak alınmıştır. Son parametre ise maksimum yanlış bulgu oranı (*max FDR—maximal false discovery rate*) değerini belirler. Tüm testler için bu parametrenin değeri 0.05'tir. Buradaki sayı varsayılan olarak alınmıştır.

Yukarıda kısa tanımları verilen parametreler hakkında ayrıntılı ek bilgilere, "*http://cbdm-01.zdv.uni-mainz.de/~jfontain/cms/?page_id=605*" isimli web sitesinden ulaşılabilir. Verilen eklerde hastalık (*disease*), gen sayısı (*genes count*), genlerin yüzdesi (*genes percentage*), kat değişimi (*fold change*), p-değeri (*p-value*), yanlış bulgu oranı (*FDR*) ve gen sembolleri (*gene symbols*) bilgileri bulunmaktadır. Ayrıca, eklerde verilen genlerle ilgili sayılar (***** *ifadesiyle*), ismi geçen hastalıkla ilgili *PubMed* (*https://www.ncbi.nlm.nih.gov/pubmed*) biyomedikal literatür veri tabanında yayımlanan toplam çalışma sayısını gösterir. Bu çalışmaların hangileri olduğu bilgisine dijital halde sunulan doktora tezindeki erişilebilir bağlantı adreslerinden (genler üzerindeki link bağlantıları—*hyperlink*) ulaşılabilir. Bu link bilgileri her bir gen ile ilişkilendirilmiştir. Genlere özgü *ID* numaraları (*Entrez Gene IDs*) ile bunların önemli ölçüde ilişkili olduğu her bir çalışmanın (örn. *makale*) *PubMed* veri tabanında kullanılan özel *ID* numaraları (*PMIDs*) gibi önemli deney çıktılarına tez çalışması ile birlikte verilen ek dokümanlarda ulaşılabilir. Tez ile birlikte sunulan eklerin isimleri sırasıyla; *LAML*, *HNSC*, *KIRC* ve *STAD* klinik verilerinden elde edilen gen-hastalık ilişkilerini gösteren "*EK-5-doküman*", "*EK-7-doküman*", "*EK-9-doküman*", "*EK-11-doküman*" ile ifade edilmiştir. Bu dokümanlarda verilen gen sembolleri, gen *ID*'leri ve bunlarla ilgili atıf yapılan çalışmaların *ID* numaraları arasına kolay ayırt edilebilmeleri için düz işaret sembolü ('|') konulmuştur. Bu şekilde istenilen gen veya atıf bilgilerine *PubMed* veri tabanından ulaşılabilir. Ayrıca EKLER bölümde verilen hastalıklar/genetik değişiklikler hakkında *"MeSH Descriptor Data 2018"* kütüphanesindeki tanımlayıcı bilgilere, ilgili tablolarda verilen hastalık ismiyle ilişkilendirilmiş web bağlantı linklerden erişilebilir.

LAML veri setindeki deneyler için Çizelge 4.21'deki sonuçlara ek olarak, karşılaştırma amacıyla sadece $k = 4$'e göre hem *DPT* hem de *GAT* algoritmaları ile gerçekleştirilen denemelerin sonuçları aşağıda verilmiştir. Buna göre hem *GAT* hem de *DPT* ile en iyi log-rank skoru "-0.565023" elde edilmiştir. Bu skora göre *DPT* ile elde

edilen "*FBXL19, ZNF629, ZNF646, ZNF668*" genleri ve *GAT* ile kaydedilen "*BCKDK, KAT8, PRSS8, STX1B*" genleri en uygun alt-ağları temsil ederler. Her iki algoritmaya göre *STRING* veri tabanı kullanılarak elde edilen gen ağları Şekil 4.19'da sunulmuştur. Burada genler arasındaki bağlantıların kalınlığı etkileşim skorunun (*güven değeri*) yüksekliğine göre belirlenir. İki gen arasındaki yüksek skor daha kalın çizgi ile temsil edilmiştir. Bunun aksine, düşük güven değerlerine sahip bağlantılar daha ince çizgilerle vurgulanmıştır. Şekil 4.19'daki her iki alt-ağ da minimum 0.15 güven değerine göre oluşturulmuştur (string-db 2018).



Şekil 4.19. $k = 4$ değerine göre; **a)** GAT ile — **b)** DPT ile elde edilen alt gen ağları (string-db 2018)

LAML veri setinde gerçekleştirilen $k = 4$ testleri için algoritmaların çalışma süreleri kontrol edildiğinde; log-rank skorları açısından aynı değer elde edilmiş olsa da çalışma süreleri hesaba katıldığında, *GAT* algoritmasının performans açısından daha avantajlı olduğu gözlemlenmiştir. Çünkü aynı skora *DPT* algoritması ile en az 18 gün 8 saatte ulaşılmışken; *GAT* algoritması ile sadece 18 dakika 16 saniyede ulaşılmıştır. Daha büyük ve daha karmaşık ağlar için *DPT*'nin kullanımı performans ve süre kısıtları sebebiyle oldukça zordur. Bunun aksine bu tür büyük ve karmaşık ağlarda, *GAT* algoritması ile genellikle uygun çözümlere çok daha kısa sürelerde ulaşılabilmektedir. Bu yüzden bu çalışmada, gen/protein ve bağlantı sayısı olarak çok fazla karmaşık olan biyolojik ağlardaki anlamlı alt-ağların tespitinde sadece *GAT* algoritması kullanılmıştır.

Çizelge 4.22. HNSC klinik verisi için elde edilen test sonuçları

HNSC	s^E	s^O	$\bar{t}$	Elde edilen gen listeleri
$k = 4$	2.13728	0.922587	14dk01sn	*BCL2, BLK, CDKN2A, NSUN3*
$k = 5$	2.13728	1.35164	19dk09sn	*BCL2, BLK, CDKN2A, NSUN3, TSLP*
$k = 6$	2.93832	1.71436	19dk31sn	*BLK, CDKN2A, E2F4, IRAK2, KRAS, PHLPP1*
$k = 7$	3.4033	2.17749	22dk45sn	*AR, ATXN1, CDK5, CDKN2B, GAPDH, HMGB1, IKBKB*
$k = 8$	2.93832	1.90087	27dk12sn	*ATXN7, CDKN2A, CREBBP, GATA4, KRAS, SMAD2, TADA3, TAF6*
$k = 9$	3.4033	1.87798	25dk32sn	*ATXN1, CDK5, CDKN2B, GAPDH, HMGB1, IKBKB, MLL5, SUPT5H, YWHAE*
$k = 10$	3.4033	2.58629	25dk57sn	*BMS1, CDKN2B, CHEK2, ELP3, HIST1H3C, HIST1H4A, ITK, MPP6, NOP2, SMC1A*

Çizelge 4.22'de *HNSC* kanser veri setinde gerçekleştirilen testlerde farklı *k* değerlerine göre elde edilen skorlar ile *GAT* algoritmasının çalışma süreleri verilmiştir. Bu çizelgede gösterilen genlerin oluşturduğu etkileşim ağı ise EK-6'da verilmiştir. Bu alt-ağdaki genler arası bağlantıların en düşük güven değeri 0.4 olarak seçilmiştir. Bu değer, ağdaki tüm genlerin genellikle güçlü bağlantılara sahip olduğunu gösterir. Alt-ağdaki genlerin sayısı 34'tür ve bu genlere göre oluşturulan gen-hastalık ilişkisi tablosu EK-7'de gösterilmiştir. Bu tablo, hastalıklarla önemli ölçüde ilişkili olan az beş adet gene göre oluşturulmuştur. *HNSC* verisinin EK-7'de de sunulan ve *GS2D* web analiz aracı yardımıyla elde edilen gen-hastalık ilişki analizi sonuçlarına, "*EK-7-doküman*" isimli dosyadan ulaşılabilir. Bu sonuçlara göre *LAML* veri setinden daha fazla ilişkili gen listesi elde edilmiştir. Her bir hastalık sonucu için en az 5 hasta ile ilgili atıflar tabloda verilmiştir. Hatta *HNSC* verisi için "Genetic Predisposition to Disease" isimli *"hastalığa genetik yatkınlık"* durumu için 14 genin birden etkili olduğu sonucu, EK-7'deki tablodan anlaşılmıştır. *LAML* veri seti için bu sayı 7'dir.

Bir diğer klinik kanser verisi olan *KIRC* ile ilgili deney sonuçları Çizelge 4.23'te sunulmuştur. Burada 7 farklı *k* değerine göre 20 kez çalıştırılan *GAT* algoritmasının en iyi ve ortalama log-rank skorları ile ortalama çalışma süresi verilmiştir. Deney sonuçlarına göre toplam 42 adet *CNA*'lı genin sağkalım parametresi ile maksimum ilişkili olduğu gözlemlenmiştir. "*https://string-db.org*" web adresindeki çoklu protein aracı yardımıyla tüm genlerin/proteinlerin oluşturduğu etkileşim ağı, EK-8'de verildiği gibi oluşturulmuştur. Bu ağda genler arası bağlantıların en az güven değeri 0.2'dir. Yine

bu genlerin hastalıklarla ya da diğer biyolojik değişikliklerle olan ilişki analizi sonucu, EK-9'da gösterilmiştir.

Çizelge 4.23. KIRC veri seti ile yapılan deneylerin sonuçları

KIRC	s^E	s^O	$\bar{t}$	Elde edilen gen listeleri
$k = 4$	2.28798	1.63717	41dk16sn	*ATRX, FLNB, ITGA2, TLN1*
$k = 5$	2.19862	1.97063	48dk38sn	*ACTR3B, DCTN2, DNAI1, FLNB, KLC1*
$k = 6$	2.46178	2.04212	46dk22sn	*ACO1, DLD, H6PD, PDHB, PYGL, UBC*
$k = 7$	2.54742	2.17746	48dk07sn	*ANAPC4, COL4A5, FLNB, ITGA2, ITGB5, MAD2L2, VCP*
$k = 8$	2.69316	2.34385	51dk28sn	*ACO1, DCLRE1A, FBXO6, NOS3, PDHB, PIAS1, UBC, UBE2G2*
$k = 9$	2.28055	2.18961	54dk47sn	*ACTA1, AKT1, CD72, CDK2, IMPDH2, PTPRC, RBX1, TLR7, YME1L1*
$k = 10$	2.45687	2.30339	01sa03dk	*ACTRT2, CEBPB, MAPK11, MAPKAPK3, RHOC, SDHC, SMAD3, TAF1L, TAF9, UBC*

Son olarak, *STAD* veri setinde gerçekleştirilen testlerle ile ilgili elde edilen alt-ağda toplam 48 gen bulunmaktadır. Bu deney verisine göre kaydedilen test sonuçları Çizelge 4.24'te verilmiştir. Tüm genlerin *STRING* yardımıyla görselleştirilmiş hali ise EK-10 sunulmuştur. Bu gen ağındaki tüm düğümlerin birbirleriyle etkileşimlerinin yoğunluğu (*toplam bağlantı sayısı gibi*), önceki ağlara göre ortalama olarak daha azdır. Ancak minimum güven değeri *LAML* ve *KIRC* veri setlerindekinden daha yüksektir. *STAD* verisi ile ilgili elde edilen genler arası minimum güven değeri 0.275'tir. Burada elde edilen tüm genlerin bazı hastalıklara minimum 5'li etkilerinin analiz edildiği test sonuçları ise EK-11'de verilmiştir. Ayrıca bu ekte verilen genler ve atfı yapılan ilgili hastalıkların PubMed veri tabanındaki ID numaraları ile diğer bilgileri "*EK-11-doküman*" isimli *Microsoft Excel* dosyasında sunulmuştur. Bu veri setindeki testlerde diğer veri setlerine oranla daha fazla ilişki gen elde edilmiş olmasına rağmen bu genlerin sadece üç adet hastalıkla ya da biyolojik değişiklikle ilişkili oldukları anlaşılmıştır. Bu sonuçlardan yola çıkarak, kanser türlerine göre elde edilen sağkalım ile yoğun ilişkili *CNA*'lı genlerin tespit edilmesinde, bazı kanser türlerinin sonuçlarının daha anlamlı olduğu sonucuna varılmıştır.

Çizelge 4.24. Farklı k değerleri için STAD klinik veri seti ile gerçekleştirilen testlerin sonuçları

STAD	s^E	s^O	$\bar{t}$	Elde edilen gen listeleri
$k = 4$	–5.9898	–6.59314	42dk05sn	*ACTA2, CCT8, PTEN, VCL*
$k = 5$	–4.5059	–5.19135	46dk02sn	*CALR, FTCD, GOLPH3, HM13, TAC1*
$k = 6$	–3.70775	–4.63196	53dk11sn	*COL18A1, PIAS4, TECPR1, TGM2, TRAPPC8, TRIO*
$k = 7$	–4.16116	–4.24811	52dk22sn	*CDK13, PSMD12, RNF40, SMURF1, SUMO3, TMEM189, USP14*
$k = 8$	–3.44722	–4.2548	01sa07dk	*ATP5I, ATP5J2, ATP5O, C14orf2, COX4I2, MRPL27, NDUFS4, NDUFS7*
$k = 9$	–3.97231	–4.38007	56dk46sn	*ASIP, CTNNA1, MC5R, MLLT4, PVR, RASSF8, SAMSN1, YWHAG, YWHAH*
$k = 10$	–3.87913	–4.86286	01sa16dk	*ACTBL2, ATP5A1, CCT5, CORO1A, HSPA13, RPL7L1, SHMT1, SMURF1, TNNC2, TNNT1*

4.2.2. Kanserle ilişkili CNA'lı genlerin diğer biyolojik etkilerinin analizi

Bu bölümde hem ağ modüllerinin ortaya çıkarılması problemi (P^1) hem de sağkalım analizi problemi (P^2) birlikte ele alınmıştır. Bu yeni problem Bölüm 2.3'te, P^3 ile temsil edilmiştir. Bir önceki bölümde sırasıyla; *LAML*, *HNSC*, *KIRC* ve *STAD* klinik kanser verileri ile yapılan sağkalım analizleri sonucu Çizelge 4.21, Çizelge 4.22, Çizelge 4.23 ve Çizelge 4.24'te farklı k değerlerine göre test sonuçları verilmiştir. Buradaki test sonuçlarına göre her bir klinik deney verisinin tüm k değerleri kendi içlerinde birleştirilerek sırasıyla; *LAML-net* ağından 36; *HNSC-net* ağından 34; *KIRC-net* ağından 42 ve *STAD-net* ağından 48 adet genin ilgili kanser türlerinde sağkalım parametresi ile maksimum ilişkide oldukları sonucuna varılmıştır. Bunlarla ilgili deneysel sonuçlar tez çalışmasının EKLER bölümünde sunulmuş olup; test sonuçlarının ayrıntılı olarak sunulduğu diğer açıklayıcı bilgilere tezle birlikte verilen ek dosyalardan ulaşılabilir. Bu başlık altında dört farklı biyolojik ağın en uygun modül yapıları ile ilgili deneysel sonuçlar verilmiştir. Çizelge 4.25'te en uygun modülerlik skorları ile kaydedilen toplam ağ modülleri sayıları gösterilmiştir. Bu çizelgede gen/düğüm sayısı N ile ifade edilirken; bağlantı sayısı M ile temsil edilmiştir.

Çizelge 4.25. Dört klinik test verisinin etkileşim ağlarındaki en anlamlı ağ modülleri

Ağ ismi	***N***	***M***	***En yüksek modülerlik skoru***	***Ağ modülleri sayısı***
LAML-net	*10161*	*229867*	0.481352	13
HNSC-net	*13828*	*425037*	0.477511	17
KIRC-net	*14161*	*447977*	0.431194	18
STAD-net	*10911*	*210245*	0.482870	20

Çizelge 4.25'te dört karmaşık biyolojik ağ (*LAML-net*, *HNSC-net*, *KIRC-net* ve *STAD-net*) üzerindeki deneylerden elde edilen skorların 0.3 ile 0.7 arasında olduğu ve bu yüzden de her bir ağ üzerindeki deneylerle elde edilen ağ modüllerinin normal anlamlılık koşulunu (Newman 2004 (a)) sağladığı anlaşılmıştır. Bu yüzden tez çalışmasında elde edilen *LAML—modül*, *HNSC—modül*, *KIRC—modül* ve *STAD—modül* isimli ağ modül yapılarının güçlü-anlamlı topluluk yapılarını temsil ettikleri gözlemlenmiştir. Ayrıca her bir biyolojik ağdaki genlerin ait oldukları modüllerin *ID* numaraları ile diğer bilgiler, "*EK-modüller-doküman*" isimli dosyada sunulmuştur.

Çizelge 4.21'de *LAML* veri setinde sağkalım parametresi ile en yoğun ilişkili *CNA*'lı genlerden aynı modüllerde bulunan ve yoğun etkileşimlere sahip oldukları farz edilen genler, *LAML—modül* listesini oluşturmuştur. Bu listede 17 adet gen vardır. Bunların görselleştirilmiş hali EK-12'de sunulmuştur. Burada verilen genler arası bağlantı skoru (*güven değeri*) 0.28'dir. EK-13'teki tablonun altında bu genlerin isimleri verilmiştir. Bunlardan en az 3 genin bazı hastalıklarla veya bozukluklar/değişikliklerle ilişkili oldukları varsayılarak; EK-13'teki tabloda bu hastalıklar verilmiştir. Ayrıca "*EK-13-doküman*" isimli dosyada da bu hastalıklar ve genlerle ilgili analiz sonuçları sunulmuştur. Burada *LAML—modül*'deki gen gruplarının hastalıklarla ortak ilişkilerinin değerlendirilebilmesi amacıyla "*http://cbdm.uni-mainz.de/geneset2diseases*" isimli web adresinden ulaşılan "*GS2D*" web analiz modülü kullanılmıştır (Fontaine ve Andrade-Navarro 2016, GS2D 2018).

HNSC-net ağındaki modülerlik skoru 0.477511'tir ve bu değere göre ağ yapısı 17 adet modüle ayrılmıştır. *HNSC* veri setinde sağkalım ile maksimum ilişkili olduğu düşünülen 34 adet genden aynı modülde bulunan düğüm/gen sayısı 21'dir. Bu 21 adet genin hepsinin bulunduğu modülün *ID* numarası 4'tür. Hem bu modüldeki üyeler hem de *HNSC-net* ağındaki 13828 adet genin ait oldukları modül bilgileri, "*EK-modüller-doküman*" isimli dosyada verilmiştir. Seçilen 17 genin ait olduğu grup, *HNSC—modül* ile isimlendirilmiştir. Bunların oluşturduğu etkileşim ağı, "*https://string-db.org*" web

adresinde görselleştirilmiş ve EK-14'te verilmiştir. Ayrıca "*HNSC—modül*" ağındaki genlerin en az üçünün etkili oldukları hastalık listesi EK-15'te gösterilmiştir. Bu tablo verilen 2 listeden oluşan toplam 12 hastalık ya da benzeri biyolojik eğilim içermektedir. Tezle birlikte verilen "*EK-15-doküman*" dosyası, seçilen genlerin ilişkili oldukları varsayımıyla atıf yapılan çalışmaların *PMIDs* numaralarını ve diğer analiz sonuçlarını içermektedir. Bu çıktının analizin sonuçları diğer kanser türündeki verilerin çıktılarının analiz sonuçlarından farklıdır. Çünkü burada en az üç genin birlikte etkili oldukları görülen hastalıklar ile ilişkileri sorgulandığında, bu genlerin kombinasyonlarının daha fazla hastalıkta görüldüğü anlaşılmıştır. Ayrıca bu genlerin aynı modülde bulunmaları ve ağda en az 0.4 güven değeriyle bağlı olmaları anlamlı bir sonuç olarak görülebilir.

Üçüncü veri setindeki (*KIRC*) denemeler sonunda, "*KIRC-net*" ağının farklı boyutlarda olan 18 adet modüle ayrılabileceği sonucuna varılmıştır. Bu ağdaki toplam 14161 genden sağkalım ile maksimum ilişkili olan 42 adet gen için aynı modülde olanları "*KIRC—modül*" alt-ağını oluşturmuştur. Bu ağdaki düğümler topluluğu, 3. modülden seçilen 14 geni ifade eder. Bu genlerin birbirleri ile oluşturdukları etkileşim ağının görüntüsü EK-16'da sunulmuştur. Ağdaki bağlantıların güven değeri bir önceki ağdaki gibi minimum 0.4'tür. Seçilen genlerin isimleri ise EK-17'deki tablonun altında gösterilmiştir. EK-17'deki tablo ile bu genlerin en az 3 tanesinin etkili oldukları hastalıkların listesi sunulmuştur. Genlerin ilişkili oldukları varsayılan hastalıklarla ilgili atıf bilgileri daha ayrıntılı olarak "*EK-17-doküman*" isimli dosyada verilmiştir.

Son olarak, *STAD* klinik kanser verisinde elde edilen sağkalım ile maksimum ilişkili tüm genler farklı k değerlerine göre Çizelge 4.24'te gösterilmiştir. Bu genlerin hepsinin birleştirildiği ağda toplam 48 adet gen bulunmaktadır. Bu genlerin etkileşim görüntüsü EK-10'daki şekilde verilmiştir. Bu genlerden aynı modülde olanların oluşturdukları "*STAD—modül*" alt-ağı ise toplam 21 adet gen içermektedir. Bu genlerin tamamı ilk modülden alınmıştır ve en anlamlı hastalık ilişkilerine sahip olan genler bunlardır. Bu alt-ağdaki genlerin *STRING* aracılığıyla oluşturdukları gen-gen etkileşim ağı EK-18'de gösterilmiştir. Bu ağdaki genler en az 0.2 güven değeri ile birbirlerine bağlıdırlar. Bu değer, alt-ağdaki bağlantıların kuvvetli olmadığı gösterir. Ayrıca bu bilgiye ek olarak, *STAD—modül*'de 21 adet gen bulunmasına rağmen en az üç genin birden etkilediği hastalık sayısı sadece 3'tür. Buna yakın bir sonuç *LAML—modül*'de de elde edilmiştir. Ancak *LAML—modül*'de sadece 17 gen bulunmasına rağmen; 21 adet gene sahip olan *STAD—modül*'deki analiz sonuçları *LAML—modül*'deki sonuçlardan daha az anlamlıdır. Buradan da çeşitli kanser türleri ile yapılan deneylerin anlamlılık

açısından bazı farklılıklar içerebildiği sonucuna varılmıştır. *STAD—modül*'e göre elde edilen analiz sonuçları, EK-19'da verilmiştir. Bu sonuçların daha ayrıntılı hali ise tezle birlikte verilen "*EK-19-doküman*" isimli dosyada sunulmuştur. *STAD-net* ağındaki tüm genler 0.482870 modülerlik skoru ile 20 farklı modülde gruplandırılmıştır. Genlerin ait olduğu modül *ID* numaraları, gen/protein isimleri ve tez çalışmasında kullanılan gen *ID* numaraları, "*EK-modüller-doküman*" dosyasındaki "*STAD-net_20_modül_10911_gen*" isimli alt sayfada verilmiştir. Diğer kanser türlerindeki veriler için de elde edilen modül bilgileri, bu kanser türlerinin deneysel sonuçlarının verildiği ek dosyalarda yine aynı kanser türünün temsil edildiği isimlerle sunulmuştur.

5. SONUÇLAR VE ÖNERİLER

Farklı gerçek dünya sistemlerinden temin edilmiş olan biyolojik, ekolojik, sosyal ve tıbbi ağlar gibi karmaşık sistemler mevcut dinamik mekanizmaların ve keşfedilecek anlamlı bilgilerin bulunduğu gerçek dünya ağlarını temsil eder. Ağ bilim dalının şekillenmesine katkı sağlayan bu yapıların analizi için farklı matematiksel yaklaşımlar önerilmektedir. Bu yaklaşımların uygulanması sonucu sunulan yöntem, teknik ve algoritmaların çoğunda ağ yapıları çizgelerle temsil edilirler. Böylece gerçek ağlarda, basit tekniklerle ortaya çıkarılamayan önemli ve anlamlı bilgiler temel olarak çizge teorisi ve bilgisayar bilimleri iş birliği ile geliştirilen birtakım yöntemlerle tespit edilebilmektedir. Bu amaçla bu çalışmada, gerçek ağ sistemlerinde mevcut olan önemli bilgilerin tanımlanması için kullanılan iki farklı ağ analiz problemi ele alınmıştır. İlk problem, ağ modül tespiti problemidir. Burada istatistiksel analizlere dayalı anlamlı ağ modüllerinin tespiti için iki temel konuya odaklanılmıştır. Bunlardan ilkinde klasik ve sezgisel yaklaşımlarla doğru bir şekilde tanımlanamayan ağ modül yapılarının başarılı bir şekilde tespiti için farklı mekanizmalara sahip sekiz adet metasezgisel optimizasyon algoritması önerilmiştir. Bu açıdan tez çalışması ağ modül tespitinde farklı çözümler sunan algoritmaların analizlerini içermektedir. Analizlerde başarı kriteri olarak "*modülerlik*" ölçütü referans alınmıştır. İlgili deneysel çalışmalarda, en iyi sonuçlara ulaşan *3pHybrid* algoritması aynı zamanda literatürde de genellikle en uygun sonuçlara ulaşan algoritmalara benzer veya daha iyi çözümler sunmaktadır. Bu amaçla yapılan testlerle *3pHybrid* algoritmasının sonuçları literatürde mevcut olan 20 farklı algoritmanın sonuçları ile karşılaştırılmıştır. Burada karşılaştırma, 13 farklı gerçek dünya ağında yapılmıştır. Bu tezde önerilen en başarılı algoritma—*3pHybrid*, tüm ağlardaki sonuçlara göre bunlardan 11'inde maksimum modülerlik skorlarına ulaşmıştır. Bu açıdan bakıldığında, önerilen yöntemlerle ağ modül tespitinin başarılı bir şekilde gerçekleştirilmesi tez çalışmasının bu konudaki önemini vurgulamaktadır. Bu tezdeki bir diğer önemli çalışmada, iyi bilinen 10 farklı amaç fonksiyonundan en iyi sonuçları veren fonksiyonun tespitine odaklanılmıştır. Bu aşamadaki deneylerde; biyoloji, gen, protein, radar, sosyal ve tıbbi alanlardan temin edilen gerçek dünya ağları kullanılmıştır. Bu ağlar, referans topluluk yapılarına sahiptir. Böylece test işlemine tabi tutulan amaç fonksiyonları ile elde edilen ağ modüllerinin kaliteleri altı farklı küme değerlendirme ölçütü ile analiz edilmiştir. Genel başarı sıralamaları dikkate alındığında, genellikle en başarılı küme değerlendirme sonuçlarına modülerlik fonksiyonu ile ulaşılmış olsa da

bazı ağlardaki testlerde modülerlik fonksiyonu ile kaydedilen ağ modüllerinin daha az kaliteli olduğu anlaşılmıştır. Böylece modül yapıları tanımlanacak olan gerçek dünya ağlarında uygunluk ölçütü olarak *surprise*, *fitness score*, *conductance* ve *significance* gibi fonksiyonların kullanılmasının da uygun olacağı sonucuna varılmıştır. Bu yüzden ağ modüllerinin tanımlanması aşamasında uygun amaç fonksiyonunun seçilmesinin önemi, doktora tez çalışması kapsamında gerçekleştirilen tüm testlerin analiz sonuçları ve değerlendirmeleri dikkate alınarak gösterilmiştir.

Bu tez çalışması ile daha öncesinden bu alana uygulanmamış farklı metasezgisel yaklaşımların ağ modül tespitindeki sonuçları, performans ve başarı kriterleri açısından değerlendirilmiştir. Burada deneysel sonuçların yanında iki farklı istatistiksel test ile hem algoritmalar başarı durumlarına göre sıralanmış hem de algoritmaların sonuçları arasındaki farklar istatistiksel olarak analiz edilmiştir. Böylece farklı mekanizmalara sahip algoritmaların bu problemde sundukları çözümler ayrıntılı olarak incelenmiştir.

Bu çalışmanın diğer bir önemi ise klinik verilerin ve ilişkili biyolojik ağların birlikte analizlerinin yapılarak bilgi çıkarımında bulunulmasıdır. Bununla ilgili bölümde kanserde sağkalım süresi ile maksimum ilişkili olan alt gen gruplarının farklı k değerlerine göre seçilmesi amaçlanmıştır. Bu analizlerle sağkalım parametresine en yüksek etkide bulunulduğu düşünülen alt-ağların tespiti gerçekleştirilmektedir. Burada *CNA* veya *CNA*'lar içeren genlerin özellikle kanser hastalıklarında önemli olabileceği varsayımından yola çıkılarak yeni bir problem üzerinde durulmuştur. Bu konuda dinamik programlama ve genetik algoritma temelli önerilen iki yöntemle (*DPT* ve *GAT*) maksimum $k = 10$ üyeye sahip bağlı alt-ağların ortaya çıkarılmasına çalışılmıştır. Çalışma sürelerinin yaklaşık olarak logaritmik şekilde arttığı *DPT* yöntemi bu kısıtlamalar sebebiyle daha büyük ve daha karmaşık biyolojik ağlarda kullanılamamıştır. Bu yüzden bu yöntemle sadece *GAT* algoritmasının sonuçları doğrulanmıştır. İlgili deneylerde uygunluk kriteri olarak log-rank istatistik ölçütü kullanılmıştır. Burada tüm testler sonunda kaydedilen genlerden bazılarının ilişkili oldukları hastalıkların ya da biyolojik değişikliklerin neler olduğu ilgili bölümde tablolarla ve eklerle sunulmuştur. Aynı zamanda elde edilen alt-ağlarla ilgili gen-hastalık ilişkisinin değerlendirildiği sonuçlar da bu bölümlerde verilmiştir.

Bu tez çalışmasında odaklanılan diğer bir problemle *STRING* veri tabanı ve *Cytoscape* aracı yardımıyla temin edilen gen-gen etkileşim ağlarının ağ modül yapıları tespit edilmiş ve bu modül yapılarına göre sağkalım analizi ile kaydedilen tüm genler birlikte değerlendirilmiştir. Böylece hem aynı ağ modülünde yer alan hem de ilgili

kanser türünde hastaların yaşam sürelerine ciddi etkisi olan genler tespit edilmiştir. Bu genlerin görselleştirilmiş ağ yapıları deneysel çalışmalarda verilmiştir. Gen gruplarının tespitinden sonra elde edilen bu genlerin diğer bazı hastalıklara etkilerinin birlikte analiz edilebilmesi için "*gene set to diseases—GS2D*" (Fontaine ve Andrade-Navarro 2016, GS2D 2018) isimli web aracından yararlanılmıştır. Bu analiz işleminden sonra kaydedilen tüm gen-hastalık ilişki çizelgeleri hem ekler bölümünde hem de ilgili ek dosyalarda verilmiştir.

Bundan sonraki süreçlerde, buradaki test sonuçlarına göre en etkili ağ modül yapılarını sunan ve en rekabetçi olan *3pHybrid* algoritması da dahil olmak üzere önerilen yöntemlerin tümü için çalışma sürelerinin ayrıntılı bir şekilde analizi yapılarak; daha kısa sürede daha uygun sonuçlara ulaşan farklı hibrit yaklaşımlar geliştirilebilir. Bu amaçla tezdeki metasezgisel yöntemler için çalışma süreleri açısından iyileştirmeler yapılabilir veya başarılı yöntemler performans iyileştirmeleri açısından paralel veya dağıtık sistemlere uyarlanabilir. Bu yöntemlerin örtüşen ağ topluluklarının/modüllerinin tespiti problemine uygulanması veya tezde odaklanılan amaç fonksiyonlarından uygun olanlarının çok amaçlı modül tespiti problemine uygulanması gibi diğer konulara da odaklanılabilir. Daha sonra bu algoritmaların parametreleri için probleme özgü analizler yapılarak; yöntemlerin performansları uygun parametre değerlerine göre geliştirilebilir. Sonraki aşamalarda, bu çalışmadaki tüm test sonuçları ışığında genel olarak uygun ağ modül yapıları sunan amaç fonksiyonlarının kullandıkları ağ topolojik özelliklerinden ve temel aldıkları teorik matematiksel yaklaşımlarından yola çıkılarak bunlara benzer amaç fonksiyonlarının geliştirilmesine odaklanılabilir. Ağ modül tespiti probleminde önerilen yöntemler ve başarılı olarak etiketlenen amaç fonksiyonları eğer yönlü ve ağırlıklı ağlara uygulanmamışsa, bu tür ağlara uygulanabilir. Ayrıca kanser türlerinde sağkalım analizleri sonucu farklı ağ analiz yaklaşımları ile kaydedilen tüm testlerin sonuçları ileri süreçlerde anlamlı biyolojik çıkarımların yapılabilmesi ve sonraki çalışmalara öncülük edebilmesi amacıyla tez çalışmasının ekinde verilmiş olup; bu alanda çalışan kişilerin kullanımına sunulmuştur. Bunlara ek olarak, sadece kanser hastalıklarında değil; diğer önemli biyolojik bozuklukların ya da değişikliklerin de benzer şekilde analizleri gerçekleştirilebilir. Bunun için diğer önemli hastalıklarla ilgili spesifik genlerin durumlarına göre bu hastalığa sahip kişileri bazı özelliklere göre ayırt eden yeni uygunluk fonksiyonları geliştirilebilir. Böylece özellikle medikal çalışmalarda uygun klinik verilere sahip bilim insanlarıyla veya biyoinformatik alanında çalışan uzmanlarla ortak çalışmalar gerçekleştirilebilir ve yeni uygulamalar geliştirilebilir.

KAYNAKLAR

Afsariardchi N, 2013. Community Detection in Dynamic Networks, McGill University, Department of Electrical and Computer Engineering, McGill University Libraries.

Ajwad R, 2017. Identification of significantly mutated subnetworks in the breast cancer genome, M.Sc., Computer Science, University of Manitoba.

Akgüneş N, 2013. Graf parametreleri ve cebirsel yapılara grafsal yaklaşımlar, Selçuk Üniversitesi Fen Bilimleri Enstitüsü.

Albert R, 2005. Scale-free networks in cell biology, Journal of Cell Science, v. 118, i. 21, pp. 4947-4957, ISSN: 0021-9533.

Albert R, Barabasi AL, 2002. Statistical mechanics of complex networks. Rev Mod Phys, 74, 1, 47-97.

Aldecoa R, Marin I, 2011. Deciphering Network Community Structure by Surprise. Plos One, 6, 9.

Alon N, Yuster R, Zwick U. Color-coding: a new method for finding simple paths, cycles and other small subgraphs within large graphs. Proceedings of the twenty-sixth annual ACM symposium on Theory of computing, 326-35.

Amiri B, Hossain L, Crawford JW, Wigand RT, 2013. Community detection in complex networks: Multi–objective enhanced firefly algorithm. Knowl-Based Syst, 46, 1-11.

Arınık N, Orman GK, İncel ÖD. Mobil İletişim Verisi Kullanarak Topluluk Bulma Konusuna Genel Bakış ve Yöntemler. Akademik Bilişim '15, 1033, 4 - 6 Şubat 2015, Eskişehir, Turkey.

Arnaboldi V, Conti M, La Gala M, Passarella A, Pezzoni F, 2016. Ego network structure in online social networks and its impact on information diffusion. Comput Commun, 76, 26-41.

Atay Y, Aslan M, Kodaz H, 2017 (a). A swarm intelligence-based hybrid approach for identifying network modules, Journal of Computational Science, Online erişim başlangıç tarihi: 18 October 2017.

Atay Y, Koc I, Babaoglu I, Kodaz H, 2017 (b). Community detection from biological and social networks: A comparative analysis of metaheuristic algorithms. Applied Soft Computing, 50, 194-211.

Atay Y, Koc I, Beskirli M, 2017 (c). Detection of Cohesive Subgroups in Social Networks using Invasive Weed Optimization Algorithm. The Eurasia Proceedings of Educational & Social Sciences, 7, 221-6.

Atay Y, Kodaz H. Modularity-Based Graph Clustering Using Harmony Search Algorithm. Advanced Computer Science Applications and Technologies (ACSAT), 2015 4th International Conference on, 109-14.

Atay Y, Kodaz H, 2016. A New Adaptive Genetic Algorithm for Community Structure Detection. Proc Adapt Learn Opt, 5, 43-55.

Babers R, Hassanien AE, 2017. A nature-inspired metaheuristic cuckoo search algorithm for community detection in social networks. International Journal of Service Science, Management, Engineering, and Technology (IJSSMET), 8, 1, 50-62.

Bachmaier C, Brandes, U., and Schreiber, F., 2013. Biological networks. In: Handbook of graph drawing and visualization. Eds: Tamassia R. USA: CRC Press., p. 621 - 51.

Bader GD, Hogue CW, 2003. An automated method for finding molecular complexes in large protein interaction networks. Bmc Bioinformatics, 4.

Barber MJ, Clark JW, 2009. Detecting network communities by propagating labels under constraints. Phys Rev E, 80, 2.

Bass JIF, Sahni N, Shrestha S, Garcia-Gonzalez A, Mori A, Bhat N, Yi S, Hill DE, Vidal M, Walhout AJM, 2015. Human Gene-Centered Transcription Factor Networks for Enhancers and Disease Variants. Cell, 161, 3, 661-73.

Bennett L, 2013. Community structure detection in complex biological networks, King's College London (University of London).

Bennett L, Kittas A, Muirhead G, Papageorgiou LG, Tsoka S, 2015. Detection of Composite Communities in Multiplex Biological Networks. Sci Rep-Uk, 5.

Bennett L, Liu S, Papageorgiou LG, Tsoka S, 2012 (a). Detection of Disjoint and Overlapping Modules in Weighted Complex Networks. Adv Complex Syst, 15, 5.

Bennett L, Liu SS, Papageorgiou LG, Tsoka S, 2012 (b). A Mathematical Programming Approach to Community Structure Detection in Complex Networks. Comput-Aided Chem En, 30, 1387-91.

Bennett L, Lysenko A, Papageorgiou LG, Urban M, Hammond-Kosack K, Rawlings C, Saqi M, Tsoka S. Detection of multi-clustered genes and community structure for the plant pathogenic fungus Fusarium graminearum. Proceedings of the 10th international conference on Computational Methods in Systems Biology, 69-86.

Beroukhim R, Mermel CH, Porter D, Wei G, Raychaudhuri S, Donovan J, Barretina J, Boehm JS, Dobson J, Urashima M, Mc Henry KT, Pinchback RM, Ligon AH, Cho YJ, Haery L, Greulich H, Reich M, Winckler W, Lawrence MS, Weir BA, Tanaka KE, Chiang DY, Bass AJ, Loo A, Hoffman C, Prensner J, Liefeld T, Gao Q, Yecies D, Signoretti S, Maher E, Kaye FJ, Sasaki H, Tepper JE, Fletcher JA, Tabernero J, Baselga J, Tsao MS, Demichelis F, Rubin MA, Janne PA, Daly MJ, Nucera C, Levine RL, Ebert BL, Gabriel S, Rustgi AK, Antonescu CR, Ladanyi M, Letai A, Garraway LA, Loda M, Beer DG, True LD, Okamoto A, Pomeroy SL, Singer S, Golub TR, Lander ES, Getz G, Sellers WR, Meyerson M, 2010. The landscape of somatic copy-number alteration across human cancers. Nature, 463, 7283, 899-905.

Biggs N, 1993. Algebraic graph theory, Cambridge university press, ISSN: 0521458978.

Biggs N, Lloyd EK, Wilson RJ, 1976. Graph Theory, 1736-1936, Oxford University Press, ISSN: 0198539169.

Blake A, Zisserman A, 1987. Visual reconstruction, MIT Press, p. 225, ISSN: 0-262-02271-0.

Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E, 2008. Fast unfolding of communities in large networks. J Stat Mech-Theory E.

Boccaletti S, Latora V, Moreno Y, Chavez M, Hwang D-U, 2006. Complex networks: Structure and dynamics. Physics reports, 424, 4-5, 175-308.

Bollobas B, 2013. Modern graph theory, Springer Science & Business Media, v. 184, ISSN: 1461206197.

Bollobas B, 1981. The diameter of random graphs. Transactions of the American Mathematical Society, 267, 1, 41-52.

Botta F, del Genio CI, 2016. Finding network communities using modularity density, Journal of Statis. Mechanics-Theory and Experiment.

Brandes U, Delling D, Gaertler M, Gorke R, Hoefer M, Nikoloski Z, Wagner D, 2008. On modularity clustering. Ieee T Knowl Data En, 20, 2, 172-88.

Brandes U, Delling D, Gaertler M, Görke R, Hoefer M, Nikoloski Z, Wagner D, 2006. Maximizing modularity is hard. arXiv preprint physics/0608255.

Butcher EC, Berg EL, Kunkel EJ, 2004. Systems biology in drug discovery. Nat Biotechnol, 22, 10, 1253-9.

Cai Q, Gong M, Ma L, Jiao L, 2015. A Novel Clonal Selection Algorithm for Community Detection in Complex Networks. Computational Intelligence, 31, 3, 442-64.

Cai Q, Ma LJ, Gong MG, Tian DY, 2016. A survey on network community detection based on evolutionary computation. Int J Bio-Inspir Com, 8, 2, 84-98.

Cao B, Luo J, Liang C, Wang S, Song D, 2015. Moepga: A novel method to detect protein complexes in yeast protein–protein interaction networks based on multiobjective evolutionary programming genetic algorithm. Computational biology and chemistry, 58, 173-81.

cBioPortal, 2016. Klinik kanser veri setleri, *http://www.cbioportal.org/data_sets.jsp,* [En ziyaret Tarihi: 02.12.2016].

Cerami E, 2011. Identifying driver networks in cancer, Weill Medical College of Cornell University, ISSN: 1267307986.

Cerami E, Gao J, Dogrusoz U, Gross BE, Sumer SO, Aksoy BA, 2012. The cBio Cancer Genomics Portal: An Open Platform for Exploring Multidimensional Cancer Genomics Data (vol 2, pg 401, 2012), Cancer Discovery, 2 (10), pp. 960.

CGARN, 2013. Cancer Genome Atlas Research Network, Comprehensive molecular characterization of clear cell renal cell carcinoma. Nature, 499, 7456, 43.

Chakraborty T, Srinivasan S, Ganguly N, Mukherjee A, Bhowmick S. On the permanence of vertices in network communities. Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 1396-405.

Chen M, 2015. Discovering community structure by optimizing community quality metrics, Rensselaer Polytechnic Institute.

Chin JH, Ratnavelu K, 2016. Detecting Community Structure by Using a Constrained Label Propagation Algorithm. Plos One, 11, 5.

Chin L, Meyerson M, Aldape K, Bigner D, Mikkelsen T, VandenBerg S, Kahn A, Penny R, Ferguson ML, Gerhard DS, Getz G, Brennan C, Taylor BS, Winckler W, Park P, Ladanyi M, Hoadley KA, Verhaak RGW, Hayes DN, Spellman PT, Absher D, Weir BA, Ding L, Wheeler D, Lawrence MS, Cibulskis K, Mardis E, Zhang JH, Wilson RK, Donehower L, Wheeler DA, Purdom E, Wallis J, Laird PW, Herman JG, Schuebel KE, Weisenberger DJ, Baylin SB, Schultz N, Yao J, Wiedemeyer R, Weinstein J, Sander C, Gibbs RA, Gray J, Kucherlapati R, Lander ES, Myers RM, Perou CM, McLendon R, Friedman A, Van Meir EG, Brat DJ, Mastrogianakis GM, Olson JJ, Lehman N, Yung WKA, Bogler O, Berger M, Prados M, Muzny D, Morgan M, Scherer S, Sabo A, Nazareth L, Lewis L, Hall O, Zhu YM, Ren YR, Alvi O, Yao JQ, Hawes A, Jhangiani S, Fowler G, San Lucas A, Kovar C, Cree A, Dinh H, Santibanez J, Joshi V, Gonzalez-Garay ML, Miller CA, Milosavljevic A, Sougnez C, Fennell T, Mahan S, Wilkinson J, Ziaugra L, Onofrio R, Bloom T, Nicol R, Ardlie K, Baldwin J, Gabriel S, Fulton RS, McLellan MD, Larson DE, Shi XQ, Abbott R, Fulton L, Chen K, Koboldt DC, Wendl MC, Meyer R, Tang YZ, Lin L, Osborne JR, Dunford-Shore BH, Miner TL, Delehaunty K, Markovic C, Swift G, Courtney W, Pohl C, Abbott S, Hawkins A, Leong S, Haipek C, Schmidt H, Wiechert M, Vickery T, Scott S, Dooling DJ, Chinwalla A, Weinstock GM, O'Kelly M, Robinson J, Alexe G, Beroukhim R, Carter S, Chiang D, Gould J, Gupta S, Korn J, Mermel C, Mesirov J, Monti S, Nguyen H, Parkin M, Reich M, Stransky N, Garraway L, Golub T, Protopopov A, Perna I, Aronson S, Sathiamoorthy N, Ren G, Kim H, Kong SK, Xiao YH, Kohane IS, Seidman J, Cope L, Pan F, Van Den Berg D, Van Neste L, Yi JM, Li JZ, Southwick A, Brady S, Aggarwal A, Chung T, Sherlock G, Brooks JD, Jakkula LR, Lapuk AV, Marr H, Dorton S, Choi YG, Han J, Ray A, Wang V, Durinck S, Robinson M, Wang NJ, Vranizan K, Peng V, Van Name E, Fontenay GV, Ngai J, Conboy JG, Parvin B, Feiler HS, Speed TP, Socci ND, Olshen A, Lash A, Reva B, Antipin Y, Stukalov A, Gross B, Cerami E, Wang WQ, Qin LX, Seshan VE, Villafania L, Cavatore M, Borsu L, Viale A, Gerald W, Topal MD, Qi Y, Balu S, Shi Y, Wu G, Bittner M, Shelton T, Lenkiewicz E, Morris S, Beasley D, Sanders S, Sfeir R, Chen J, Nassau D, Feng L, Hickey E, Schaefer C, Madhavan S, Buetow K, Barker A, Vockley J, Compton C, Vaught J, Fielding P, Collins F, Good P, Guyer M, Ozenberger B, Peterson J, Thomson E, Network CGAR, Sites TS, Ctr GS, Ctr CGC, Teams P, 2008. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. Nature, 455, 7216, 1061-8.

Clauset A, Newman MEJ, Moore C, 2004. Finding community structure in very large networks. Phys Rev E, 70, 6.

Cog A, Dumitrescu D, Hirsbrunner B, 2007. Community detection in complex networks using collaborative evolutionary algorithms. Advances in Artificial Life, Proceedings, 4648, 886-+.

Collins FS, Morgan M, Patrinos A, 2003. The Human Genome Project: lessons from large-scale biology. Science, 300, 5617, 286-90.

Collins FS, Patrinos A, Jordan E, Chakravarti A, Gesteland R, Walters L, Fearon E, Hartwelt L, Langley CH, Mathies RA, Olson M, Pawson AJ, Pollard T, Williamson A, Wold B, Buetow K, Branscomb E, Capecchi M, Church G, Garner H, Gibbs RA, Hawkins T, Hodgson K, Knotek M, Meisler M, Rubin GM, Smith LM, Smith RF, Westerfield M, Clayton EW, Fisher NL, Lerman CE, McInerney JD, Nebo W, Press N, Valle D, Grp D, Grp N, 1998. New goals for the US Human Genome Project: 1998-2003. Science, 282, 5389, 682-9.

Cover TM, Thomas JA, 2012. Elements of information theory, John Wiley & Sons, ISSN: 1118585771.

Creusefond J, Largillier T, Peyronnet S. On the evaluation potential of quality functions in community detection for different contexts. International Conference and School on Network Sci., 111-25.

Cui L, 2016. Community detection in social networks, PhD Thesis, Software Engineering, University of Texas - Dallas (UTD), http://hdl.handle.net/10735.1/5191.

Cytoscape, 2017. http://www.cytoscape.org/, En son erişim tarihi: 09.09.2017.

Danon L, Diaz-Guilera A, Duch J, Arenas A, 2005. Comparing community structure identification, Journal of Statistical Mechanics-Theory and Experiment, ISSN: 1742-5468.

Dartnell L, Simeonidis E, Hubank M, Tsoka S, Bogle IDL, Papageorgiou LG, 2005. Robustness of the p53 network and biological hackers. Febs Lett, 579, 14, 3037-42.

de Castro LN, Von Zuben FJ, 2002. Learning and optimization using the clonal selection principle. Ieee T Evolut Comput, 6, 3, 239-51.

Del Ser J, Lobo JL, Villar-Rodriguez E, Bilbao MN, Perfecto C, 2016. Community Detection in Graphs based on Surprise Maximization using Firefly Heuristics. Ieee C Evol Computat, 2233-9.

Demir HS, 2015. A comparison of focus measures for autofocus systems in IR cameras (Termal Kameralarda Otomatik Odaklama icin Fokus Metriklerinin Karsilastirilmasi), 23nd Signal Processing and Communications Applications Conference (SIU), 16-19 May 2015, Malatya, Turkey.

Deng K, Zhang JP, Yang J, 2015. An Efficient Multi-Objective Community Detection Algorithm in Complex Networks. Teh Vjesn, 22, 2, 319-28.

dolphins, 2017. ağ veri seti, *http://konect.uni-koblenz.de/networks/dolphins*, [Ziyaret Tarihi: 18.06.2017].

Dorigo M, 1992. Optimization, learning and natural algorithms. PhD Thesis, Politecnico di Milano.

Dorigo M, Gambardella LM, 1997. Ant colony system: a cooperative learning approach to the traveling salesman problem. Ieee T Evolut Comput, 1, 1, 53-66.

Dorigo M, Maniezzo V, Colorni A, 1996. Ant system: Optimization by a colony of cooperating agents. Ieee T Syst Man Cy B, 26, 1, 29-41.

Dorigo M, Maniezzo V, Colorni A, Maniezzo V, 1991. Positive feedback as a search strategy.

Duch J, Arenas A, 2005. Community detection in complex networks using extremal optimization. Phys Rev E, 72, 2.

Erdos P ve Renyi A, 1959, On random graphs I, Publ. Math. Debrecen, 6, 290-7.

Eusuff M, Lansey K, Pasha F, 2006. Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. Eng Optimiz, 38, 2, 129-54.

Eusuff MM, Lansey KE, 2003. Optimization of water distribution network design using the Shuffled Frog Leaping Algorithm. J Water Res Pl-Asce, 129, 3, 210-25.

Flake GW, Lawrence S, Giles CL. Efficient identification of web communities. Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, 150-60.

Folino F, Pizzuti C. A multiobjective and evolutionary clustering method for dynamic networks. Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on, 256-63.

Folino F, Pizzuti C, 2014. An Evolutionary Multiobjective Approach for Community Discovery in Dynamic Networks. Ieee T Knowl Data En, 26, 8, 1838-52.

Fontaine JF, Andrade-Navarro MA, 2016. Gene Set to Diseases (GS2D): Disease Enrichment Analysis on Human Gene Sets with Literature Data. Genomics and Computational Biology, 2, 1, 33.

Ford LR, Fulkerson DR, 1956. Maximal flow through a network. Canadian journal of Mathematics, 8, 3, 399-404.

Fortunato S, 2010. Community detection in graphs. Phys Rep, 486, 3-5, 75-174.

Fortunato S, Barthelemy M, 2007. Resolution limit in community detection. P Natl Acad Sci USA, 104, 1, 36-41.

Fortunato S, Hric D, 2016. Community detection in networks: A user guide. Phys Rep, 659, 1-44.

Friedman M, 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the american statistical association, 32, 200, 675-701.

Fu YH, Huang CY, Sun CT, 2017. A community detection algorithm using network topologies and rule-based hierarchical arc-merging strategies. Plos One, 12, 11.

GS2D, 2018. Data mining tools: Gene set to Diseases, http://cbdm-01.zdv.uni-mainz.de/~jfontain/cms/?page_id=592, Erişim Tarihi: 13.01.2018.

Gach O, Hao J-K. A memetic algorithm for community detection in complex networks. International Conference on Parallel Problem Solving from Nature, 327-36.

Gao JJ, Aksoy BA, Dogrusoz U, Dresdner G, Gross B, Sumer SO, Sun YC, Jacobsen A, Sinha R, Larsson E, Cerami E, Sander C, Schultz N, 2013. Integrative Analysis of Complex Cancer Genomics and Clinical Profiles Using the cBioPortal. Sci Signal, 6, 269.

GBM, 2016. Glioblastoma (TCGA, Nature 2008), TCGA Glioblastoma project, *http://www.cbioportal.org/study?id=gbm_tcga_pub#summary*, [Ziyaret Tarihi: 12.12.2016].

Ghiassian SD, Menche J, Barabasi AL, 2015. A DIseAse MOdule Detection (DIAMOnD) Algorithm Derived from a Systematic Analysis of Connectivity Patterns of Disease Proteins in the Human Interactome. Plos Comput Biol, 11, 4.

Gibbons JD, Chakraborti S, 2011. Nonparametric statistical inference, Springer, International encyclopedia of statistical science, pp. 977-979.

Girvan M, Newman MEJ, 2002. Community structure in social and biological networks. P Natl Acad Sci USA, 99, 12, 7821-6.

Gleiser PM, Danon L, 2003. Community structure in jazz. Adv Complex Syst, 6, 4, 565-73.

Glover F, 1977. Heuristics for integer programming using surrogate constraints. Decision sciences, 8, 1, 156-66.

Goldberg DE, Holland JH, 1988. Genetic algorithms and machine learning. Machine learning, 3, 2, 95-9.

Gong MG, Fu B, Jiao LC, Du HF, 2011. Memetic algorithm for community detection in networks. Phys Rev E, 84, 5.

Gong MG, Ma LJ, Zhang QF, Jiao LC, 2012. Community detection in networks by using multiobjective evolutionary algorithm with decomposition. Physica A, 391, 15, 4050-60.

Guendouz M, Amine A, Hamou RM, 2017. A discrete modified fireworks algorithm for community detection in complex networks. Appl Intell, 46, 2, 373-85.

Guimera R, Danon L, Diaz-Guilera A, Giralt F, Arenas A, 2003. Self-similar community structure in a network of human interactions. Phys Rev E, 68, 6.

Gunes I, 2006 (a). Community Detection using Agents in Complex Networks, Boğaziçi University.

Gunes I, Bingol H, 2006 (b). Community detection in complex networks using agents. arXiv preprint cs/0610129.

Hafez AI, Zawbaa HM, Hassanien AE, Fahmy AA, 2014. Networks Community Detection Using Artificial Bee Colony Swarm Optimization. Adv Intell Syst, 303, 229-39.

Hall G, 2015. Pearson's correlation coefficient. other words, 1, 9.

Hansen T, Vandin F, 2016. Finding Mutated Subnetworks Associated with Survival in Cancer. arXiv preprint arXiv:1604.02467.

Harenberg S, Bello G, Gjeltema L, Ranshous S, Harlalka J, Seay R, Padmanabhan K, Samatova N, 2014. Community detection in large-scale networks: a survey and empirical evaluation. Wiley Interdisciplinary Reviews: Computational Statistics, 6, 6, 426-39.

Harrington D, 2005. Linear rank tests in survival analysis. Encyclopedia of biostatistics.

HE Dong-Xiao ZX, WANG Zuo, ZHOU Chun-Guang, WANG Zhe, JIN Di, 2010. Community Mining in Complex Networks --- Clustering Combination Based Genetic Algorithm. Acta Automatica Sinica, 36, 8, 1160-70.

He Y, Wang JH, Wang L, Chen ZJ, Yan CG, Yang H, Tang HH, Zhu CZ, Gong QY, Zang YF, Evans AC, 2009. Uncovering Intrinsic Modular Organization of Spontaneous Brain Activity in Humans. Plos One, 4, 4.

helico, 2017. DIP veri tabanı protein-protein etkileşim ağı verisi (Helicobacter Pylori PPI ağı), *http://cosinproject.eu/extra/data/proteins/,* [Ziyaret Tarihi: 15.03.2017].

HNSC, 2016. *veri seti,* Head and Neck Squamous Cell Carcinoma (TCGA, Nature 2015), *http://www.cbioportal.org/study?id=hnsc_tcga_pub#summary*, [Ziyaret Tarihi: 23.12.2016].

Hofree M, Shen JP, Carter H, Gross A, Ideker T, 2013. Network-based stratification of tumor mutations. Nat Methods, 10, 11, 1108-15.

Holland JH, 1992. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, MIT press, ISSN: 0262581116.

Hollander M, Wolfe DA, 1999. Nonparametric statistical methods, Wiley Series (11 Ocak 1999).

Honghao C, Zuren F, Zhigang R. Community detection using ant colony optimization. Evolutionary Computation (CEC), 2013 IEEE Congress on, 3072-8.

Hu YQ, Chen HB, Zhang P, Li MH, Di ZR, Fan Y, 2008. Comparative definition of community and corresponding identifying algorithm. Phys Rev E, 78, 2.

Huang C-L, 2014. A module based approach for identifying driver genes and expanding pathways from integrated biological networks, Boston University.

Hubert L, Arabie P, 1985. Comparing Partitions. J Classif, 2, 2-3, 193-218.

Hutchings S, 2011. The behaviour of modularity-optimizing community detection algorithms, University of Oxford.

Jeong H, Tombor B, Albert R, Oltvai ZN, Barabasi AL, 2000. The large-scale organization of metabolic networks. Nature, 407, 6804, 651-4.

Jeong HH, Leem S, Wee K, Sohn KA, 2015. Integrative network analysis for survival-associated gene-gene interactions across multiple genomic profiles in ovarian cancer. J Ovarian Res, 8.

Jia G, Cai Z, Musolesi M, Wang Y, Tennant DA, Weber RJ, Heath JK, He S. Community Detection in Social and Biological Networks Using Differential Evolution.

Jin D, Liu J, Yang B, He D-X, Liu D-Y, 2011. Genetic algorithm with local search for community detection in large-scale complex networks. Acta Automatica Sinica, 37, 7, 873-82.

Kakışım A, 2013. Şüpheli Ağlarda Topluluk Tanıma, Yüksek Lisans Tezi, Bilgisayar Mühendisliği Anabilim Dalı, Gebze Teknik Üniversitesi.

Kannan R, Vempala S, Vetta A, 2004. On clusterings: Good, bad and spectral. J Acm, 51, 3, 497-515.

Karaboga D, Basturk B, 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Global Optim, 39, 3, 459-71.

Karasoy D, 2008. Yaşam Verilerinin Meta-Analizi. SDÜ Fen Dergisi, 3, 2.

Karasoy D, Tilki B, 2013. Yaşam eğrilerini karşılaştırmak için kullanılan skor ve ağırlıklı testler: Sayısal örnekler. İstatistikçiler Dergisi: İstatistik ve Aktüerya, 6, 1.

karate, 2017. Mark Newman, ağ verisi, Zachary's Karate Club (1977), *http://www-personal.umich.edu/~mejn/netdata/*, [Ziyaret Tarihi: 11.09.2017].

Kaur S, Singh S, Kaushal S, Sangaiah AK, 2016. Comparative Analysis of Quality Metrics for Community Detection in Social Networks Using Genetic Algorithm. Neural Netw World, 26, 6, 625-41.

Kazancı G, 2003. Tıp Terminolojisi. Türk Tabipleri Birliği (TTB) Sürekli Tıp Eğitimi Dergisi (STED), 6 (12), 222-6.

Kennedy J, Eberhart R, 1995. Particle swarm optimization. 1995 Ieee International Conference on Neural Networks Proceedings, Vols 1-6, 1942-8.

Khan BS, Niazi MA, 2017. Network Community Detection: A Review and Visual Survey. arXiv preprint arXiv:1708.00977.

Kim S, 2014. Community Detection in Directed Networks and its Application to Analysis of Social Networks, The Ohio State University, ISSN: 1321485158.

Kim YA, Cho DY, Przytycka TM, 2016. Understanding Genotype-Phenotype Effects in Cancer via Network Approaches. Plos Comput Biol, 12, 3.

KIRC, 2016. *veri seti,* Kidney Renal Clear Cell Carcinoma (TCGA, Nature 2013), NCI. PubMed, *http://www.cbioportal.org/study?id=kirc_tcga_pub#summary*, [Ziyaret Tarihi: 28.12.2016].

Klimt B, Yang YM, 2004. The Enron corpus: A new dataset for email classification research. Machine Learning: Ecml 2004, Proceedings, 3201, 217-26.

Knuth DE, 1993. The Stanford Graphbase - a Platform for Combinatorial Algorithms. Proceedings of the Fourth Annual Acm-Siam Symposium on Discrete Algorithms, 41-3.

Koç İ, 2016. Sınıflandırma problemlerinde meta-sezgisel optimizasyon yöntemlerinin özellik seçimi ve ayrıklaştırma amacıyla kullanımı, Selçuk Üniversitesi Fen Bilimleri Enstitüsü.

Labatut V, Balasque J-M, 2012. Detection and interpretation of communities in complex networks: Practical methods and application. In: Computational Social Networks. Eds: Springer, p. 81-113.

LAML, 2016. *veri seti,* Acute Myeloid Leukemia (TCGA, NEJM 2013), TCGA, NCI. PubMed, *http://www.cbioportal.org/study?id=laml_tcga_pub#summary*, [Ziyaret Tarihi: 19.12.2016].

Lancichinetti A, Fortunato S, 2014. Community detection algorithms: A comparative analysis (vol 80, 056117, 2009). Phys Rev E, 89, 4.

Lancichinetti A, Fortunato S, Kertesz J, 2009. Detecting the overlapping and hierarchical community structure in complex networks. New J Phys, 11.

Lancichinetti A, Fortunato S, Radicchi F, 2008. Benchmark graphs for testing community detection algorithms. Phys Rev E, 78, 4.

Lawrence MS, Sougnez C, Lichtenstein L, Cibulskisl K, Lander E, Gabriel SB, Getz G, Ally A, Balasundaram M, Birol I, Bowlby R, Brooks D, Butterfield YSN, Carlsen R, Cheng D, Chu A, Dhalla N, Guin R, Holt RA, Jones SJM, Lee D, Li HYI, Marra MA, Mayo M, Moore RA, Mungall AJ, Robertson AG, Schein JE, Sipahimalan P, Tam A, Thiessen N, Wong T, Protopopov A, Santoso N, Lee S, Parfenov M, Zhang JH, Mahadeshwar HS, Tang JB, Ren XJ, Seth S, Haseley P, Zeng D, Yang LX, Xu AW, Song XZ, Pantazi A, Bristow CA, Hadjipanayis A, Seidman J, Chin L, Park PJ, Kucherlapati R, Akbani R, Casasent T, Liu WB, Le Y, Mille G, Motter T, Weinstein J, Diao LX, Wang J, Fan YH, Lie JZ, Wang K, Auman JT, Balu S, Bodenheimer T, Buda E, Hayes DN, Hoadley KA, Hoyle AP, Jefferys SR, Jones CD, Kimes PK, Liu YF, Marron JS, Meng SW, Mieczkowski PA, Mose LE, Parker JS, Perou CM, Prins JF, Roach J, Shi Y, Simons JV, Singh D, Soloway MG, Tan DH, Veluvolu U, Walter V, Waring S, Wilkerson MD, Wu JY, Zhao N, Cherniack AD, Hammerman PS, Tward AD, Pedamallu CS, Saksena G, Jung J, Ojesina AI, Carter SL, Zack TI, Schumacher SE, Beroukhim R, Freeman SS, Meyerson M, Cho J, Chin L, Getz G, Noble MS, DiCara D, Zhang H, Heiman DI, Gehlenborg N, Voet D, Lin P, Frazer S, Stojanov P, Liu YC, Zou LH, Kim J, Sougnez C, Gabriel SB, Lawrence MS, Muzny D, Doddapaneni H, Kovar C, Reid J, Morton D, Han Y, Hale W, Chao H, Chang K, Drummond JA, Gibbs RA, Kakkar N, Wheeler D, Xi L, Ciriello G, Ladanyi M, Lee W, Ramirez R, Sander C, Shen R, Sinhal R, Weinhold N, Taylor BS, Aksoy BA, Dresdner G, Gao JJ, Gross B, Jacobsen A, Reva B, Schultz N, Sumer SO, Sun YC, Chan TA, Morris LG, Stuart J, Benz S, Ng S, Benz C, Yau C, Baylin SB, Cope L, Danilova L, Herman JG, Bootwalla M, Maglinte DT, Larid PW, Triche T, Weisenberger DJ, Van den Berg DJ, Agrawal N, Bishop J, Boutros PC, Bruce JP, Byers LA, Califano J, Carey TE, Chen Z, Cheng H, Chiosea SI, Cohen E, Diergaarde B, Egloff AM, El-Naggar AK, Ferris RL, Frederick MJ, Grandis JR, Guo Y, Haddad RI, Hammerman PS, Harris T, Hayes DN, Hui ABY, Lee JJ, Lippman SM, Liu FF, McHugh JB, Myers J, Ng PKS, Perez-Ordonez B, Pickering CR, Prystowsky M, Romkes M, Saleh AD, Sartor MA, Seethala R, Seiwert TY, Si H, Tward AD, Van Waes C, Waggott DM, Wiznerowicz M, Yarbrough WG, Zhang JX, Zuo ZX, Burnett K, Crain D, Gardner J, Lau K, Mallery D, Morris S, Lauskis JP, Penny R, Shelton C, Shelton T, Sherman M, Yena P, Black AD, Bowen J, Frick J, Gastier-Foster JM, Harper HA, Leraas K, Lichtenberg TM, Ramirez NC, Wise L, Zmuda E, Baboud J, Jensen MA, Kahn AB, Pihl TD, Pot DA, Srinivasan D, Walton JS, Wan YH, Burton RA, Davidsen T, Demchok JA, Eley G, Ferguson ML, Shaw KRM, Ozenberger BA, Sheth M, Sofia HJ, Tarnuzzer R, Wang ZN, Yang LM, Zenklusen JC, Saller C, Tarvi K, Chen C, Bollag R, Weinberger P, Golusinski W, Golusinski P, Ibbs M, Korski K, Mackiewicz A, Suchorska W, Szybiak B, Wiznerowicz M, Burnett K, Curley E, Gardner J, Mallery D, Penny R, Shelton T, Yena P, Beard C, Mitchell C, Sandusky G, Agrawal N, Ahn J, Bishop J, Califano J, Khan Z, Bruce JP, Hui ABY, Irish J, Liu FF, Perez-Ordonez B, Waldron J, Boutros PC, Waggott DM, Myers J, William WN, Lippman SM, Egea S, Gomez-Fernandez C, Herbert L, Bradford CR, Carey TE, Chepeha DB, Haddad AS, Jones TR, Komarck CM, Malakh

M, McHugh JB, Moyer JS, Nguyen A, Peterson LA, Prince ME, Rozek LS, Sartor MA, Taylor EG, Walline HM, Wolf GT, Boice L, Chera BS, Funkhouser WK, Gulley ML, Hackman TG, Hayes DN, Hayward MC, Huang M, Rathmell WK, Salazar AH, Shockley WW, Shores CG, Thorne L, Weissler MC, Wrenn S, Zanation AM, Chiosea SI, Diergaarde B, Egloff AM, Ferris RL, Romkes M, Seethala R, Brown BT, Guo Y, Pham M, Yarbrough WG, Network CGA, 2015. Comprehensive genomic characterization of head and neck squamous cell carcinomas. Nature, 517, 7536, 576-82.

Lecca P, Re A, 2015. Detecting modules in biological networks by edge weight clustering and entropy significance. Front Genet, 6.

Lee J, Lee J, 2013. Hidden information revealed by optimal community structure from a protein-complex bipartite network improves protein function prediction. Plos One, 8, 4, e60372.

Leskovec J, Lang KJ, Mahoney M. Empirical comparison of algorithms for network community detection. Proceedings of the 19th international conference on World wide web, 631-40.

Leskovec J, Mcauley JJ. Learning to discover social circles in ego networks. Advances in neural information processing systems, 539-47.

Leung A, Bader GD, Reimand J, 2014. HyperModules: identifying clinically and phenotypically significant network modules with disease mutations for biomarker discovery. Bioinformatics, 30, 15, 2230-2.

Ley TJ, Miller C, Ding L, Raphael BJ, Mungall AJ, Robertson AG, Hoadley K, Triche TJ, Laird PW, Baty JD, Fulton LL, Fulton R, Heath SE, Kalicki-Veizer J, Kandoth C, Klco JM, Koboldt DC, Kanchi KL, Kulkarni S, Lamprecht TL, Larson DE, Lin L, Lu C, McLellan MD, McMichael JF, Payton J, Schmidt H, Spencer DH, Tomasson MH, Wallis JW, Wartman LD, Watson MA, Welch J, Wendl MC, Ally A, Balasundaram M, Birol I, Butterfield Y, Chiu R, Chu A, Chuah E, Chun HJ, Corbett R, Dhalla N, Guin R, He A, Hirst C, Hirst M, Holt RA, Jones S, Karsan A, Lee D, Li HI, Marra MA, Mayo M, Moore RA, Mungall K, Parker J, Pleasance E, Plettner P, Schein J, Stoll D, Swanson L, Tam A, Thiessen N, Varhol R, Wye N, Zhao YJ, Gabriel S, Getz G, Sougnez C, Zou LH, Leiserson MDM, Vandin F, Wu HT, Applebaum F, Baylin SB, Akbani R, Broom BM, Chen K, Motter TC, Nguyen K, Weinstein JN, Zhang NZ, Ferguson ML, Adams C, Black A, Bowen J, Gastier-Foster J, Grossman T, Lichten-Berg T, Wise L, Davidsen T, Demchok JA, Shaw KRM, Sheth M, Sofia HJ, Yang LM, Downing JR, Eley G, Alonso S, Ayala B, Baboud J, Backus M, Barletta SP, Berton DL, Chu AL, Girshik S, Jensen MA, Kahn A, Kothiyal P, Nicholls MC, Pihl TD, Pot DA, Raman R, Sanbhadti RN, Snyder EE, Srinivasan D, Walton JS, Wan YH, Wang ZN, Issa JPJ, Le Beau M, Carroll M, Kantarjian H, Kornblau S, Bootwalla MS, Lai PH, Shen H, Van den Berg DJ, Weisenberger DJ, Link DC, Walter MJ, Ozenberger BA, Mardis ER, Westervelt P, Graubert TA, DiPersio JF, Wilson RK, Network CGAR, 2013. Genomic and Epigenomic Landscapes of Adult De Novo Acute Myeloid Leukemia. New Engl J Med, 368, 22, 2059-74.

Li JW, Song YL, 2013 (a). Community detection in complex networks using extended compact genetic algorithm. Soft Comput, 17, 6, 925-37.

Li K, Chen FJ, 2013 (b). Detecting Core Protein Complexes based on Link Patterns, IEEE International Conference on Bioinformatics and Biomedicine-BIBM, ISSN: 2156-1125.

Li M, Tang Y, Wu XH, Wang JX, Wu FX, Pan Y, 2016 (a). C-DEVA: Detection, evaluation, visualization and annotation of clusters from biological networks. Biosystems, 150, 78-86.

Li S, Lou H, Jiang W, Tang J, 2015. Detecting community structure via synchronous label propagation. Neurocomputing, 151, 1063-75.

Li SM, Armstrong CM, Bertin N, Ge H, Milstein S, Boxem M, Vidalain PO, Han JDJ, Chesneau A, Hao T, Goldberg DS, Li N, Martinez M, Rual JF, Lamesch P, Xu L, Tewari M, Wong SL, Zhang LV, Berriz GF, Jacotot L, Vaglio P, Reboul J, Hirozane-Kishikawa T, Li QR, Gabel HW, Elewa A, Baumgartner B, Rose DJ, Yu HY, Bosak S, Sequerra R, Fraser A, Mango SE, Saxton WM, Strome S, van den Heuvel S, Piano F, Vandenhaute J, Sardet C, Gerstein M, Doucette-Stamm L, Gunsalus KC, Harper JW, Cusick ME, Roth FP, Hill DE, Vidal M, 2004. A map of the interactome network of the metazoan C-elegans. Science, 303, 5657, 540-3.

Li Y, Liu G, Lao SY, 2013 (c). A genetic algorithm for community detection in complex networks. J Cent South Univ, 20, 5, 1269-76.

Li YH, Wang JQ, Wang XJ, Zhao YL, Lu XH, Liu DL, 2017. Community Detection Based on Differential Evolution Using Social Spider Optimization. Symmetry-Basel, 9, 9.

Li ZP, Zhang SH, Wang RS, Zhang XS, Chen LN, 2008. Quantitative function for community detection. Phys Rev E, 77, 3.

Li ZT, Liu J, 2016 (b). A multi-agent genetic algorithm for community detection in complex networks. Physica A, 449, 336-47.

Li ZX, He LL, Li YR, 2016 (c). A novel multiobjective particle swarm optimization algorithm for signed network community detection. Appl Intell, 44, 3, 621-33.

Liu HX, Ji JZ, Yang CC, Lv JW, Zhang XZ, 2014. Ant colony clustering approach combined with multilevel framework for functional module detection in large-scale PPI networks. 2014 Ieee/Wic/Acm International Joint Conferences on Web Intelligence (Wi) and Intelligent Agent Technologies (Iat), Vol 3, 17-23.

Lusseau D, Schneider K, Boisseau OJ, Haase P, Slooten E, Dawson SM, 2003. The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations - Can geographic isolation explain this unique trait? Behav Ecol Sociobiol, 54, 4, 396-405.

Ma XK, Gao L, 2012. Biological network analysis: insights into structure and functions. Brief Funct Genomics, 11, 6, 434-42.

Mantel N, 1966. Evaluation of survival data and two new rank order statistics arising in its consideration. Cancer Chemother. Rep., 50, 163-70.

Marcus SE, Moy M, Coffman T, 2007. Social network analysis. Mining graph data, 443-68.

Marti R, Laguna M, Glover F, 2006. Principles of scatter search. European Journal of Operational Research, 169, 2, 359-72.

Matula DW, Shahrokhi F, 1990. Sparsest Cuts and Bottlenecks in Graphs. Discrete Appl Math, 27, 1-2, 113-23.

Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, Alon U, 2002. Network motifs: Simple building blocks of complex networks. Science, 298, 5594, 824-7.

Mirjalili S, Lewis A, 2013. S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization. Swarm Evol Comput, 9, 1-14.

Manual for the Community Detection Toolbox v.0.91, 2014. Erişim tarihi 11.12.2017. Erişim adresi, http://users.auth.gr/~kehagiat/Software/index.htm.

Moon S, Lee JG, Kang M, Choy M, Lee JW, 2016. Parallel community detection on large graphs with MapReduce and GraphChi. Data Knowl Eng, 104, 17-31.

Moradian Zadeh P, 2017. Social Network Analysis using Cultural Algorithms and its Variants.

Naeni LM, Berretta R, Moscato P, 2015. MA-Net: A Reliable Memetic Algorithm for Community Detection by Modularity Optimization. Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, Vol 1, 311-23.

Narayanan T, 2013. Community Detection in Biological Networks, Ph.D., Electrical Engineering (Intelligent Systems,Roboticsand Control), University of California, San Diego, https://escholarship.org/uc/item/88w7n5wt, ProQuest ID: Narayanan_ucsd_0033D_13310.

Newman M, 2010. Networks: an introduction, Oxford university press, ISSN: 0199206651.

Newman ME, 2004 (a). Fast algorithm for detecting community structure in networks. Phys Rev E, 69, 6, 066133.

Newman ME, Girvan M, 2004 (b). Finding and evaluating community structure in networks. Phys Rev E, 69, 2, 026113.

Newman MEJ, 2001. The structure of scientific collaboration networks, Proceedings of the National Academy of Sciences of the United States of America, v. 98, i. 2, pp. 404-409, ISSN: 0027-8424.

Newman MEJ, 2003. The structure and function of complex networks, Siam Review, v. 45, i. 2, pp. 167-256, ISSN: 0036-1445.

Newman MEJ, 2006 (a). Modularity and community structure in networks, Proceedings of the National Academy of Sciences of the United States of America, v. 103, i. 23, pp. 8577-8582.

Newman MEJ, 2006 (b). Finding community structure in networks using the eigenvectors of matrices. Phys Rev E, 74, 3.

Newman MEJ, 2013. Community detection and graph partitioning. Epl-Europhys Lett, 103, 2.

Nicolini C, Bifone A, 2016. Modular structure of brain functional networks: breaking the resolution limit by Surprise. Sci Rep-Uk, 6.

Omar AY, 2017. Community Detection in Social Networks, Master, Fırat University.

Orman GK, Labatut V, Cherifi H, 2013. Towards realistic artificial benchmark for community detection algorithms evaluation. International Journal of Web Based Communities, 9, 3, 349-70.

Orman K, 2014. Contribution to the interpretation of evolving communities in complex networks: Application to the study of social interactions, INSA de Lyon.

Özturk K, 2014. Community detection in social networks, Msc. Thesis. Graduate School of Natural and Applied Sciences, Middle East Technical University.

Palla G, Derenyi I, Farkas I, Vicsek T, 2005. Uncovering the overlapping community structure of complex networks in nature and society. Nature, 435, 7043, 814-8.

Papadopoulos S, Kompatsiaris Y, Vakali A, Spyridonos P, 2012. Community detection in Social Media Performance and application considerations. Data Min Knowl Disc, 24, 3, 515-54.

Pardalos PM, Xue J, 1994. The Maximum Clique Problem. J Global Optim, 4, 3, 301-28.

Park YJ, Song MS. A genetic algorithm for clustering problems. 3rd Annual Conference on Genetic Programming, 568–75, Paris, France.

Patel VN, Gokulrangan G, Chowdhury SA, Chen YW, Sloan AE, Koyuturk M, Barnholtz-Sloan J, Chance MR, 2013. Network Signatures of Survival in Glioblastoma Multiforme. Plos Comput Biol, 9, 9.

Pereira-Leal JB, Enright AJ, Ouzounis CA, 2004. Detection of functional modules from protein interaction networks. Proteins, 54, 1, 49-57.

Peto R, Peto J, 1972. Asymptotically efficient rank invariant test procedures. Journal of the Royal Statistical Society. Series A (General), 185-207.

Pizzuti C, 2008. GA-Net: A Genetic Algorithm for Community Detection in Social Networks. Parallel Problem Solving from Nature - Ppsn X, Proceedings, 5199, 1081-90.

Pizzuti C, 2009. A Multi-Objective Genetic Algorithm for Community Detection in Networks. Proc Int C Tools Art, 379-86.

Pizzuti C, 2012. A Multiobjective Genetic Algorithm to Find Communities in Complex Networks, IEEE Transactions on Evolutionary Computation, v. 16, i. 3, pp. 418-430, ISSN: 1089-778x.

polbooks, 2017. (Books about US Politics), *http://www-personal.umich.edu/~mejn/netdata/*, [Ziyaret Tarihi: 25.06.2017].

Porter MA, Onnela J-P, Mucha PJ, 2009. Communities in networks. Notices of the AMS, 56, 9, 1082-97.

Pothen A, Simon HD, Liou K-P, 1990. Partitioning sparse matrices with eigenvectors of graphs. SIAM journal on matrix analysis and applications, 11, 3, 430-52.

Radicchi F, Castellano C, Cecconi F, Loreto V, Parisi D, 2004. Defining and identifying communities in networks. P Natl Acad Sci USA, 101, 9, 2658-63.

Rand WM, 1971. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association, 66, 336, 846-50.

Rashedi E, Nezamabadi-Pour H, Saryazdi S, 2009. GSA: A Gravitational Search Algorithm. Inform Sciences, 179, 13, 2232-48.

Rashedi E, Nezamabadi-pour H, Saryazdi S, 2010. BGSA: binary gravitational search algorithm. Nat Comput, 9, 3, 727-45.

Reimand J, Bader GD, 2013. Systematic analysis of somatic mutations in phosphorylation signaling predicts novel cancer drivers. Mol Syst Biol, 9.

Rhrissorrakrai K, Gunsalus KC, 2011. MINE: Module Identification in Networks, BMC Bioinformatics, v. 12, ISSN: 1471-2105.

Ronhovde P, Nussinov Z, 2009. Multiresolution community detection for megascale networks by information-based replica correlations. Phys Rev E, 80, 1.

Ronhovde P, Nussinov Z, 2010. Local resolution-limit-free Potts model for community detection. Phys Rev E, 81, 4.

Rosvall M, Bergstrom CT, 2007. An information-theoretic framework for resolving community structure in complex networks. P Natl Acad Sci USA, 104, 18, 7327-31.

Rosvall M, Bergstrom CT, 2008. Maps of random walks on complex networks reveal community structure. P Natl Acad Sci USA, 105, 4, 1118-23.

Sadi S, 2010 (a). Community Detection in Social Networks using Parallel Clique-Finding Ants, M.Sc., Istanbul Technical University.

Sadi S, Oguducu SG, Uyar S, 2010 (b). An Efficient Community Detection Method using Parallel Clique-Finding Ants, IEEE Congress on Evolutionary Computation (Cec-2010).

Said A, Abbasi RA, Maqbool O, Daud A, Aljohani NR, 2018. CC-GA: A clustering coefficient based genetic algorithm for detecting communities in social networks. Appl Soft Comput, 63, 59-70.

Sastry K, Goldberg DE, Kendall G, 2014. Genetic algorithms. In: Search methodologies. Eds: Springer, p. 93-117.

Sertkaya D, Ata N, Sözer MT, 2005. Yaşam çözümlemesinde zamana bağlı açıklayıcı değişkenli Cox regresyon modeli. Ankara Üniversitesi Tıp Fakültesi Mecmuası, 58(04).

Shannon CE, 1997. The mathematical theory of communication (Reprinted). M D Comput, 14, 4, 306-17.

Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T, 2003. Cytoscape: A software environment for integrated models of biomolecular interaction networks. Genome Res, 13, 11, 2498-504.

Shen-Orr SS, Milo R, Mangan S, Alon U, 2002. Network motifs in the transcriptional regulation network of Escherichia coli. Nat Genet, 31, 1, 64-8.

Shen R, 2014. Graph mining and module detection in protein-protein interaction networks, State University of New York at Albany, ISSN: 1303932962.

Shi C, Yan ZY, Cai YN, Wu B, 2012. Multi-objective community detection in complex networks. Applied Soft Computing, 12, 2, 850-9.

Shi C, Yan ZY, Wang Y, Cai YA, Wu B, 2010. A Genetic Algorithm for Detecting Communities in Large-Scale Complex Networks. Adv Complex Syst, 13, 1, 3-17.

Shi JB, Malik J, 1997. Normalized cuts and image segmentation, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (1997), Proceedings, pp. 731-737, ISSN: 1063-6919.

Shi JB, Malik J, 2000. Normalized cuts and image segmentation. Ieee T Pattern Anal, 22, 8, 888-905.

Shi Y, Eberhart RC. Empirical study of particle swarm optimization. Evolutionary computation, 1999. CEC 99. Proceedings of the 1999 congress on, 1945-50.

Shi ZW, Liu Y, Liang JJ, 2009. PSO-based Community Detection in Complex Networks. 2009 Second International Symposium on Knowledge Acquisition and Modeling: Kam 2009, Vol 3, 114-+.

SLNDC-sosyal ego ağları, Stanford Large Network Dataset Collection, ego-Facebook/ego-Twitter/ego-Gplus, *http://snap.stanford.edu/data/index.html,* [Ziyaret Tarihi: 21.04.2017].

SNAP, 2016. "Stanford Network Analysis Project, C++", *https://snap.stanford.edu/snap/index.html*, [Ziyaret Tarihi: 05.12.2016].

Sporns O, Betzel RF, 2016. Modular Brain Networks. Annu Rev Psychol, 67, 613-40.

STAD, 2016. *veri seti,* Stomach Adenocarcinoma (TCGA, Provisional), TCGA-*cBioPortal* (geçici veri temini tarih: 29.12.2016), *http://www.cbioportal.org/study?id=stad_tcga#summary*, Ziyaret Tarihi: 04.12.2016].

Steinhaeuser K, Chawla NV, 2008. Community detection in a large real-world social network. Social Computing, Behavioral Modeling and Prediction, 168-75.

Steinhaeuser K, Chawla NV, 2010. Identifying and evaluating community structure in complex networks. Pattern Recogn Lett, 31, 5, 413-21.

STRING, 2017. (En sık görülen 20 insan kanser genleri) The 20 most frequently mutated human cancer genes, Example 2, https://string-db.org/, En son erişim tarihi: 16 Ekim 2017.

string-db, 2018. STRING veri tabanı—gen/protein etkileşim ağları, https://string-db.org, En son erişim tarihi: 15 Ocak 2018.

Stutzle T, Hoos HH, 2000. MAX-MIN Ant System. Future Gener Comp Sy, 16, 8, 889-914.

Su G, Kuchinsky A, Morris JH, States DJ, Meng F, 2010. GLay: community structure analysis of biological networks. Bioinformatics, 26, 24, 3135-7.

Subelj L, Bajec M, 2014. Group detection in complex networks: An algorithm and comparison of the state of the art. Physica A, 397, 144-56.

Sun HL, Huang JB, Zhang X, Liu J, Wang D, Liu HL, Zou JH, Song QB, 2014. IncOrder: Incremental density-based community detection in dynamic networks. Knowl-Based Syst, 72, 1-12.

Sun Y, 2009. Optimization problems in complex networks, University of Houston, ProQuest Dissertations Publishing.

Szklarczyk D, Franceschini A, Wyder S, Forslund K, Heller D, Huerta-Cepas J, Simonovic M, Roth A, Santos A, Tsafou KP, 2014. STRING v10: protein–protein interaction networks, integrated over the tree of life. Nucleic acids research, 43, D1, D447-D52.

Szklarczyk D, Morris JH, Cook H, Kuhn M, Wyder S, Simonovic M, Santos A, Doncheva NT, Roth A, Bork P, 2016. The STRING database in 2017: quality-controlled protein–protein association networks, made broadly accessible. Nucleic acids research, gkw937.

Tan Y, Zhu YC, 2010. Fireworks Algorithm for Optimization, Advances in Swarm Intelligence, Pt 1, Proceedings, v. 6145, p. 355, Lecture Notes in Computer Science, ISSN: 0302-9743.

Tasgin M, 2005. Community detection model using genetic algorithm in complex networks and its application in real-life networks. Graduate Program in Computer Engineering.

Tasgin M, Herdagdelen A, Bingol H, 2007. Community detection in complex networks using genetic algorithms. arXiv preprint arXiv:0711.0491.

TCGA, 2016. Kanser Genom Atlası, The Cancer Genome Atlas (TCGA) - Cancer Types, *https://tcga-data.nci.nih.gov/docs/publications/tcga/*, [Ziyaret Tarihi: 12.09.2016].

TCGA, geçici-2016. (Geçici veri setleri) Broad Inst. Firehose, *http://www.cbioportal.org/data_sets.jsp,* [Ziyaret Tarihi: 04.12.2016].

Tong AHY, Evangelista M, Parsons AB, Xu H, Bader GD, Page N, Robinson M, Raghibizadeh S, Hogue CWV, Bussey H, Andrews B, Tyers M, Boone C, 2001. Systematic genetic analysis with ordered arrays of yeast deletion mutants. Science, 294, 5550, 2364-8.

Traag VA, Krings G, Van Dooren P, 2013. Significant Scales in Community Structure. Sci Rep-Uk, 3.

Vandin F, 2016. Sunulan çalışma: Finding Mutated Subnetworks Associated with Survival in Cancer, Sunumu yapan: Fabio Vandin (University of Padova), Computational Cancer Biology, The Simons Institute, *https://simons.berkeley.edu/talks/fabio-vandin-02-02-2016,* [Ziyaret Tarihi: 23.12.2016].

Vandin F, 2017. Computational Methods for Characterizing Cancer Mutational Heterogeneity. Front Genet, 8, 83.

Vandin F, Clay P, Upfal E, Raphael BJ, 2012. Discovery of Mutated Subnetworks Associated with Clinical Data in Cancer. Biocomput-Pac Sym, 55-66.

Vandin F, Papoutsaki A, Raphael BJ, Upfal E, 2015. Accurate Computation of Survival Statistics in Genome-Wide Studies. Plos Comput Biol, 11, 5.

Vinh NX, Epps J, Bailey J, 2010. Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. J. Mach. Learn. Res., 11, 2837-54.

von Luxburg U, 2007. A tutorial on spectral clustering. Stat Comput, 17, 4, 395-416.

Wagner S, Wagner D, 2007. Comparing clusterings: an overview, Universitat Karlsruhe, Fakultat für Informatik Karlsruhe.

Wang LR, Hopcroft J, 2010. Community Structure in Large Complex Networks. Theory and Applications of Models of Computation, Proceedings, 6108, 455-66.

Wang Y, 2015. Module Identification for Biological Networks, PhD thesis (12.08.2015), Texas A & M University, Online: https://oaktrust.library.tamu.edu/handle/1969.1/155541, Erişim tarihi: 11.08.2017.

Watts DJ, Strogatz SH, 1998. Collective dynamics of 'small-world' networks. Nature, 393, 6684, 440-2.

Wei YC, Cheng CK, 1989. Towards Efficient Hierarchical Designs by Ratio Cut Partitioning. 1989 Ieee International Conference on Computer-Aided Design, 298-301.

White S, Smyth P, 2005. A Spectral Clustering Approach To Finding Communities in Graphs. Siam Proc S, 274-85.

Wu GM, Stein L, 2012. A network module-based method for identifying cancer prognostic signatures. Genome Biol, 13, 12.

Wu P, Pan L, 2015. Multi-Objective Community Detection Based on Memetic Algorithm. Plos One 10/5.

Xu G, Bennett L, Papageorgiou LG, Tsoka S, 2010. Module detection in complex networks using integer optimisation. Algorithm Mol Biol, 5.

Xu G, Tsoka S, Papageorgiou LG, 2007. Finding community structures in complex networks using mixed integer optimisation. Eur Phys J B, 60, 2, 231-9.

Yan Y, Qiu SZ, Jin ZX, Gong SH, Bai Y, Lu JW, Yu TW, 2017. Detecting subnetwork-level dynamic correlations. Bioinformatics, 33, 2, 256-65.

Yang J, McAuley J, Leskovec J, 2013. Community Detection in Networks with Node Attributes. 2013 Ieee 13th International Conference on Data Mining (Icdm), 1151-6.

Yang JW, Leskovec J, 2012 (a). Defining and Evaluating Network Communities based on Ground-truth, 12th IEEE International Conference on Data Mining (Icdm 2012), pp. 745-754, ISSN: 1550-4786.

Yang XS, 2010. A New Metaheuristic Bat-Inspired Algorithm. Stud Comput Intell, 284, 65-74.

Yang Z, Hao T, Dikmen O, Chen X, Oja E. Clustering by nonnegative matrix factorization using graph random walk. Advances in Neural Information Processing Systems, 1079-87.

Yayla ME, 2013. Yaşam Analizleri ve Cox Regresyon Modeli, Erişim Tarihi: 19 Haziran 2017, www.academia.edu, (Kasım 2013).

Yu Y, Liu J, Feng N, Song B, Zheng ZY, 2017. Combining sequence and Gene Ontology for protein module detection in the Weighted Network. J Theor Biol, 412, 107-12.

Yuan XG, Zhang JY, Yang LY, Zhang SL, Chen BD, Geng YJ, Wang Y, 2012. TAGCNA: A Method to Identify Significant Consensus Events of Copy Number Alterations in Cancer. Plos One, 7, 7.

Zachary WW, 1977. An information flow model for conflict and fission in small groups. Journal of anthropological research, 33, 4, 452-73.

Zalik KR ve Zalik B, 2017. Multi-objective evolutionary algorithm using problem-specific genetic operators for community detection in networks. Neural Computing and Applications, 1-14.

Zaremotlagh S, Hezarkhani A, Sadeghi M, 2016. Detecting homogenous clusters using whole-rock chemical compositions and REE patterns: A graph-based geochemical approach. Journal of Geochemical Exploration, 170, 94-106.

Zheng XH, Wu LT, Ye SZ, Chen RQ, 2017. Simplified Swarm Optimization-Based Function Module Detection in Protein-Protein Interaction Networks. Appl Sci-Basel, 7, 4.

Zhou X, Liu Y, Li B, 2016. A multi-objective discrete cuckoo search algorithm with local search for community detection in complex networks. Mod Phys Lett B, 30, 07, 1650080.

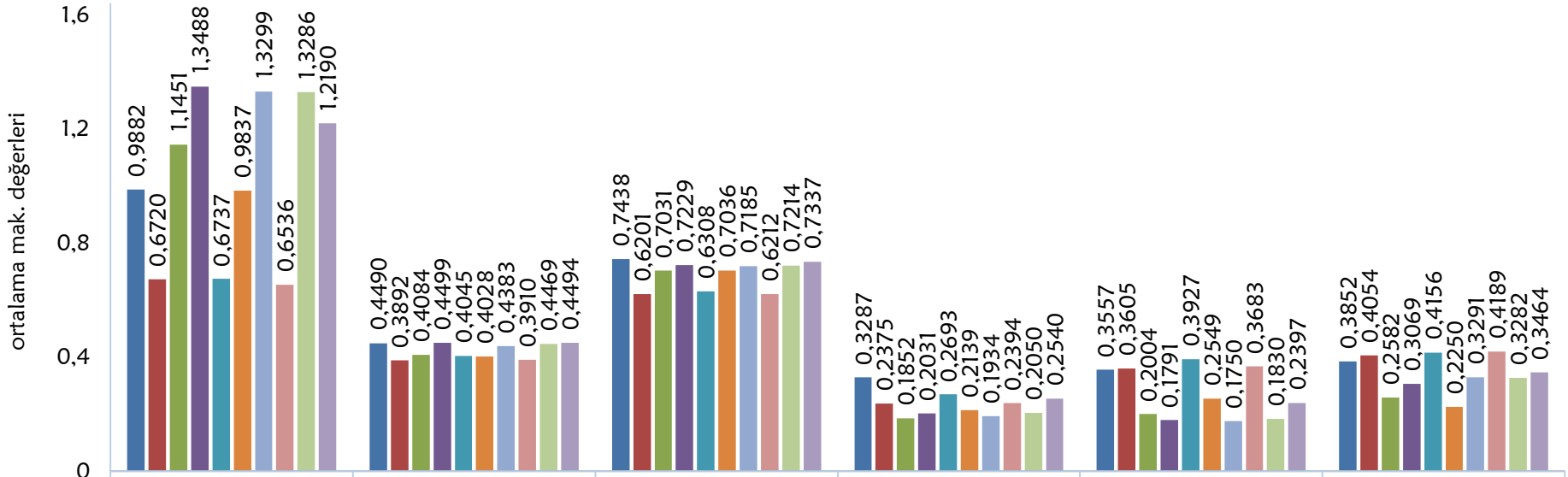
Zhou X, Liu YH, Li B, Li H, 2017. A multiobjective discrete cuckoo search algorithm for community detection in dynamic networks. Soft Comput, 21, 22, 6641-52.

Zhu XW, Gerstein M, Snyder M, 2007. Getting connected: analysis and principles of biological networks. Gene Dev, 21, 9, 1010-24.

Zırhlıoğlu G, Kara K, 2004. Yaşam Analizi Yöntemleri Kullanılarak Ana Arı Yetiştiriciliği İle İlgili Bazı Parametrelerin Tahmini. Yüzüncü Yıl Üniversitesi Tarım Bilimleri Dergisi, 14, 1, 7-15.

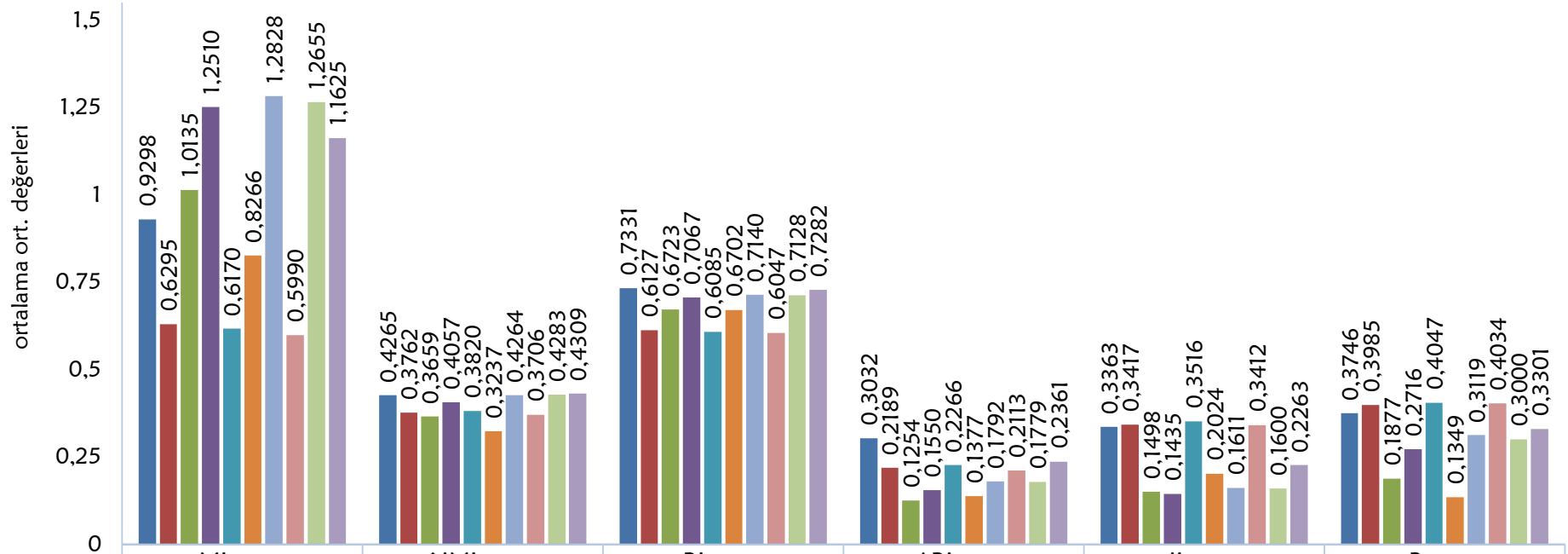
EKLER

EK-1 10 adet amaç fonksiyonu için hesaplanan en iyi skorların ortalamaları ($\langle mak \rangle$ *değerleri*)—tüm ağlardaki sonuçlara göre



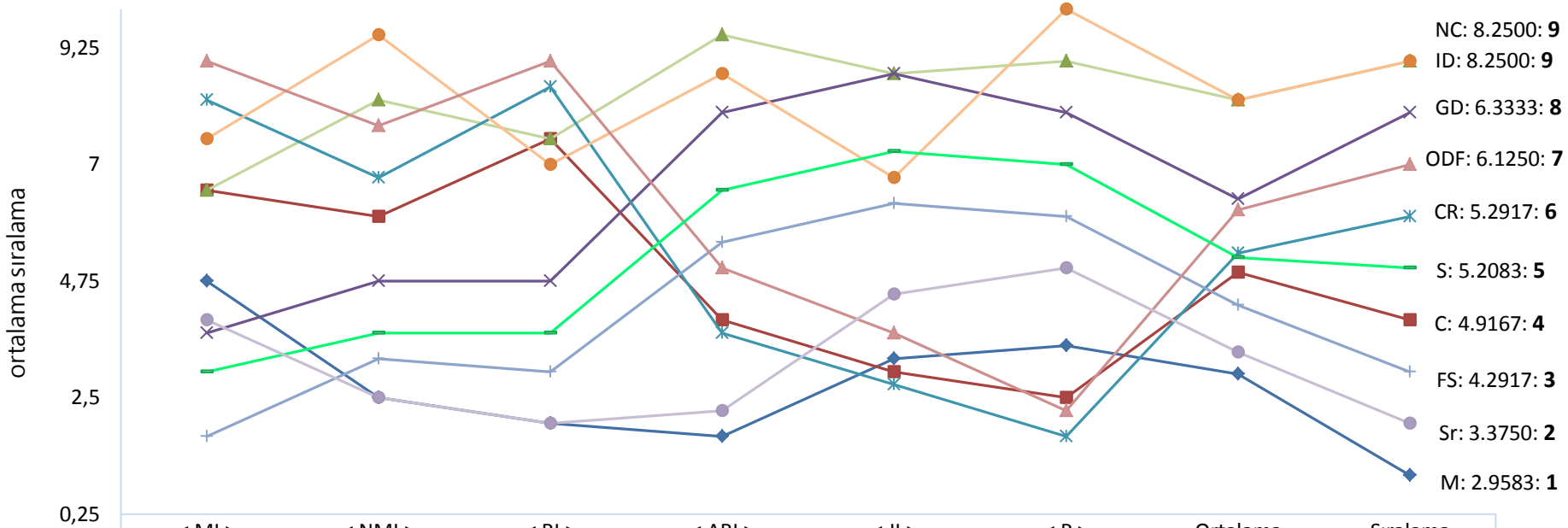
	MI—mak.	NMI—mak.	RI—mak.	ARI—mak.	JI—mak.	P—mak.
M	0,9882	0,4490	0,7438	0,3287	0,3557	0,3852
C	0,6720	0,3892	0,6201	0,2375	0,3605	0,4054
ID	1,1451	0,4084	0,7031	0,1852	0,2004	0,2582
GD	1,3488	0,4499	0,7229	0,2031	0,1791	0,3069
CR	0,6737	0,4045	0,6308	0,2693	0,3927	0,4156
NC	0,9837	0,4028	0,7036	0,2139	0,2549	0,2250
FS	1,3299	0,4383	0,7185	0,1934	0,1750	0,3291
ODF	0,6536	0,3910	0,6212	0,2394	0,3683	0,4189
S	1,3286	0,4469	0,7214	0,2050	0,1830	0,3282
Sr	1,2190	0,4494	0,7337	0,2540	0,2397	0,3464

EK-2 Farklı amaç fonksiyonları için tüm ağlardaki sonuçlara göre elde edilen ortalama $\langle ort \rangle$ değerler



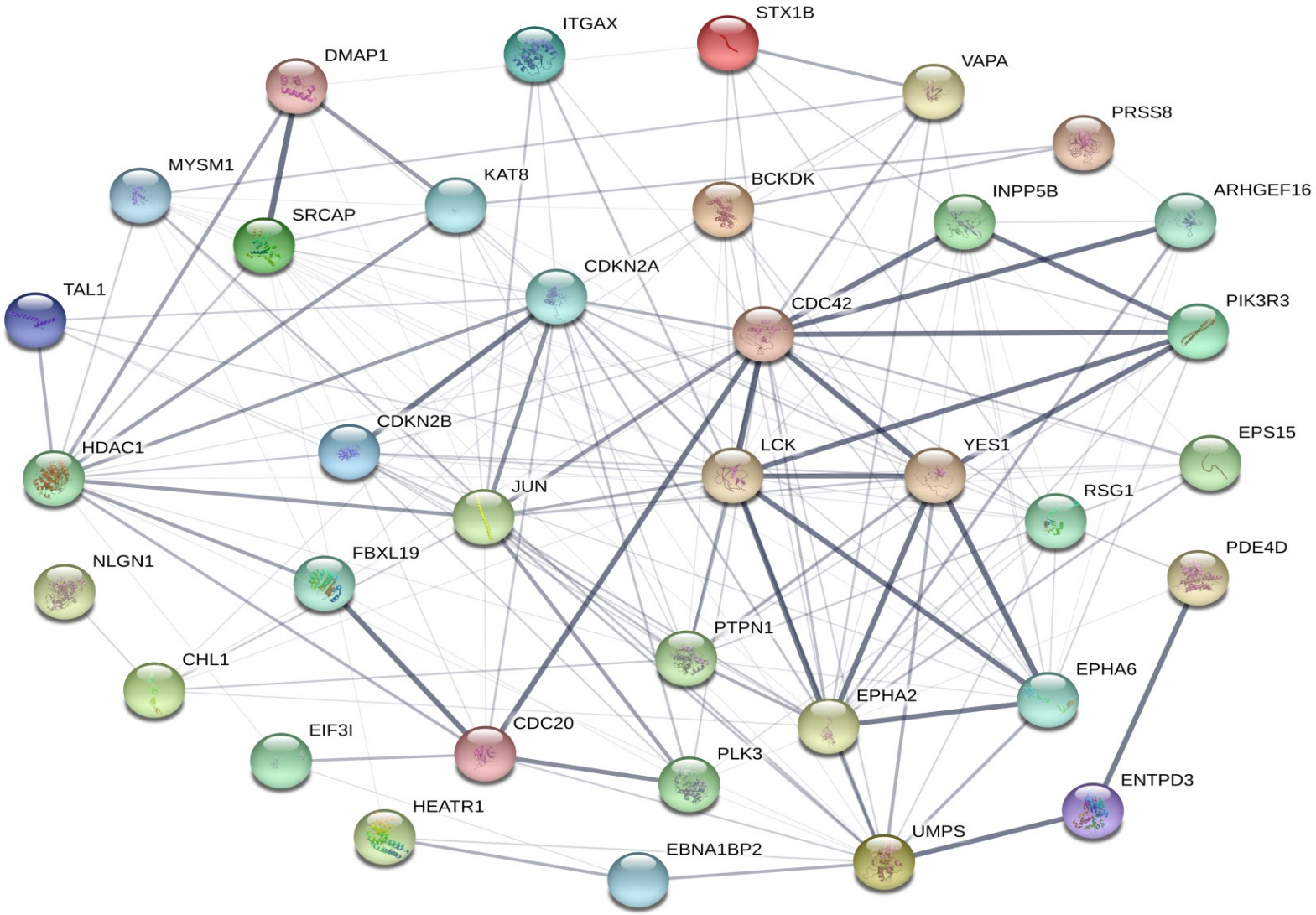
	MI—ort.	NMI—ort.	RI—ort.	ARI—ort.	JI—ort.	P—ort.
M	0,9298	0,4265	0,7331	0,3032	0,3363	0,3746
C	0,6295	0,3762	0,6127	0,2189	0,3417	0,3985
ID	1,0135	0,3659	0,6723	0,1254	0,1498	0,1877
GD	1,2510	0,4057	0,7067	0,1550	0,1435	0,2716
CR	0,6170	0,3820	0,6085	0,2266	0,3516	0,4047
NC	0,8266	0,3237	0,6702	0,1377	0,2024	0,1349
FS	1,2828	0,4264	0,7140	0,1792	0,1611	0,3119
ODF	0,5990	0,3706	0,6047	0,2113	0,3412	0,4034
S	1,2655	0,4283	0,7128	0,1779	0,1600	0,3000
Sr	1,1625	0,4309	0,7282	0,2361	0,2263	0,3301

EK-3 Amaç fonksiyonlarına göre ortalama değerlerin (⟨mak⟩, ⟨min⟩, ⟨ort⟩ ve ⟨std⟩) tümünün birden hesaba alındığı <u>en son</u> başarı sıralamaları



	< MI >	< NMI >	< RI >	< ARI >	< JI >	< P >	Ortalama	Sıralama
M	4,75	2,5	2	1,75	3,25	3,5	2,9583	1
C	6,5	6	7,5	4	3	2,5	4,9167	4
ID	6,5	8,25	7,5	9,5	8,75	9	8,2500	9
GD	3,75	4,75	4,75	8	8,75	8	6,3333	8
CR	8,25	6,75	8,5	3,75	2,75	1,75	5,2917	6
NC	7,5	9,5	7	8,75	6,75	10	8,2500	9
FS	1,75	3,25	3	5,5	6,25	6	4,2917	3
ODF	9	7,75	9	5	3,75	2,25	6,1250	7
S	3	3,75	3,75	6,5	7,25	7	5,2083	5
Sr	4	2,5	2	2,25	4,5	5	3,3750	2

EK-4 *LAML* klinik kanser veri setinde gerçekleştirilen deneyler sonunda elde edilen sağkalım ile ilişkili birleştirilmiş gen etkileşim ağı



EK-5 *LAML* verisindeki testlerde *LAML-net* ağından temin edilen genler için hastalık ilişki tablosu

Hastalık listesi	Gen sayısı	Genlerin yüzdesi	Kat değişimi	P-değeri	FDR	Gen sembolleri *
Stomach Neoplasms	5	0.15	1.52	2.269e-01	4.538e-01	CDKN2A[47], EPHA2[7], HDAC1[10], JUN[13], UMPS[5]
Cell Transformation, Neoplastic	7	0.21	1.60	1.388e-01	5.552e-01	CDC42[14], CDKN2A[72], EPHA2[6], HDAC1[12], JUN[22], LCK[5], PTPN1[6]
Neoplasm Invasiveness	7	0.21	1.30	2.881e-01	3.841e-01	CDC42[25], CDKN2A[50], EPHA2[18], HDAC1[18], JUN[33], PTPN1[6], PIK3R3[5]
Genetic Predisposition to Disease	7	0.21	0.36	1.000e+00	1.000e+00	#CDKN2A[200], #CDKN2B[123], ITGAX[6], PDE4D[52], PTPN1[16], CHL1[8], NLGN1[6]

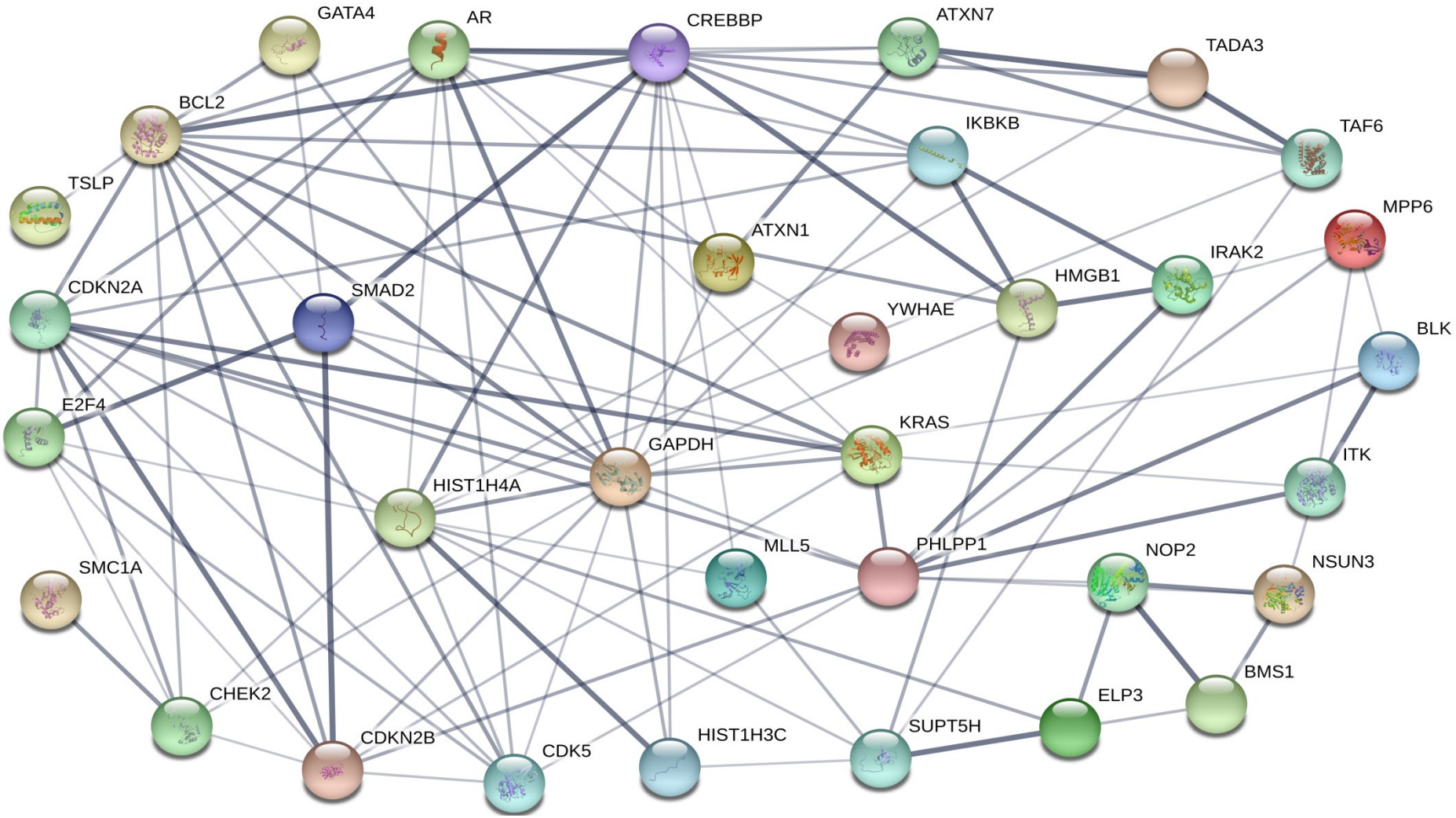
***** Her bir gen üzerindeki sayı, hastalıkla ilgili *PubMed* veri tabanında yayımlanmış toplam çalışma sayısını gösterir.

ile belirtilen genlerle ilgili *PubMed* iletişim link (*hyperlink*) adresleri çok uzun olduğu için atıf yapılan çalışmaların *PMIDs* numaralarına, EK-5-doküman'da erişilebilir.

Genler: *ARHGEF16, BCKDK, CDC20, CDC42, CDKN2A, CDKN2B, CHL1, DMAP1, EBNA1BP2, EIF3I, ENTPD3, EPHA2, EPHA6, EPS15, FBXL19, HDAC1, HEATR1, INPP5B, ITGAX, JUN, KAT8, LCK, MYSM1, NLGN1, PDE4D, PIK3R3, PLK3, PRSS8, PTPN1, RSG1, SRCAP, STX1B, TAL1, UMPS, VAPA, YES1*

Kaynak: *http://cbdm-01.zdv.uni-mainz.de/~jfontain/cms/?page_id=592*

EK-6 *HNSC* klinik verisindeki testler sonucunda tüm *k* değerlerine göre oluşturulan gen etkileşim ağının görüntüsü



EK-7 *HNSC* veri setindeki deneyler sonucunda *HNSC-net* gerçek dünya ağına göre elde edilen *CNA*'lı genlerin hastalıklarla ilişkisi

Hastalık listesi	Gen sayısı	Genlerin yüzdesi	Kat değişimi	P-değeri	FDR	Gen sembolleri *
Endometrial Neoplasms	5	0.15	4.65	4.013e-03	3.210e-02	AR[15], BCL2[16], CDKN2A[20], KRAS[26], CHEK2[5]
Carcinoma	6	0.18	2.43	3.324e-02	2.327e-01	AR[35], BCL2[27], CDKN2A[43], GATA4[5], KRAS[46], CHEK2[9]
Colorectal Neoplasms	6	0.18	1.56	1.813e-01	6.346e-01	BCL2[31], CDKN2A[51], E2F4[6], #KRAS[200], SMAD2[11], CHEK2[27]
Cell Transformation, Neoplastic	6	0.18	1.37	2.684e-01	6.263e-01	BCL2[27], CDKN2A[72], CREBBP[14], IKBKB[12], #KRAS[97], SMAD2[9]
Prostatic Neoplasms	6	0.18	1.21	3.756e-01	6.573e-01	#AR[200], BCL2[45], CREBBP[12], GAPDH[8], CHEK2[16], PHLPP1[5]
Neoplasm Invasiveness	6	0.18	1.11	4.590e-01	6.426e-01	AR[39], BCL2[30], CDKN2A[50], HMGB1[25], KRAS[45], SMAD2[16]
Breast Neoplasms	7	0.21	0.79	8.168e-01	9.529e-01	#AR[110], #BCL2[122], CREBBP[22], E2F4[9], IKBKB[12], SMAD2[21], #CHEK2[139]
Genetic Predisposition to Disease	14	0.42	0.71	9.853e-01	9.853e-01	#AR[163], BCL2[45], BLK[38], CDK5[9], #CDKN2A[200], #CDKN2B[123], CREBBP[11], GATA4[19], IKBKB[10], #KRAS[121], SMAD2[10], YWHAE[9], #CHEK2[133], TSLP[25]

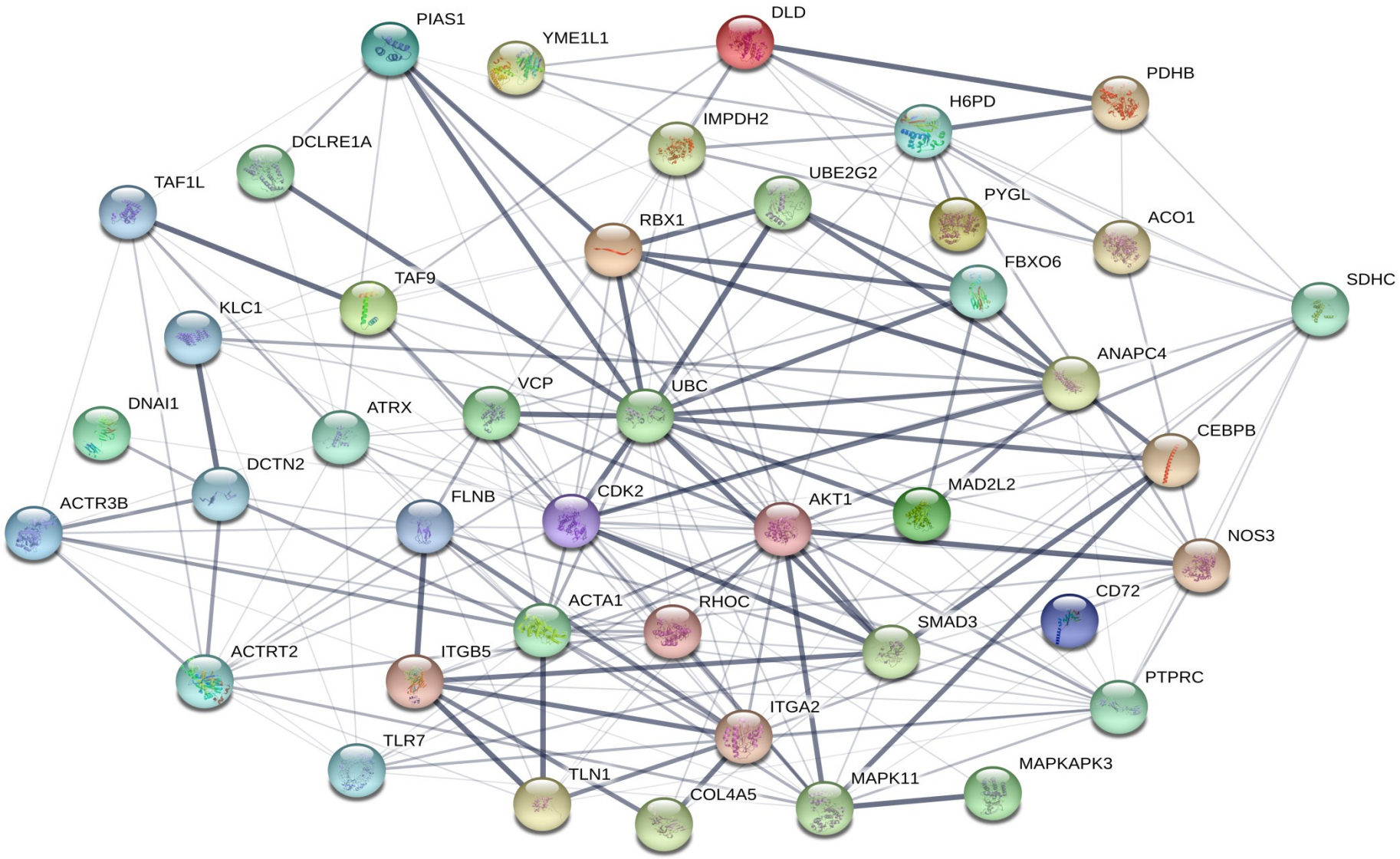
***** Her bir gen üzerindeki sayı, hastalıkla ilgili *PubMed* veri tabanında yayımlanmış toplam çalışma sayısını gösterir.

ile belirtilen genlerle ilgili *PubMed* iletişim link (*hyperlink*) adresleri çok uzun olduğu için atıf yapılan çalışmaların *PMIDs* numaralarına, EK-7-doküman'da erişilebilir.

Genler: *AR, ATXN1, ATXN7, BCL2, BLK, BMS1, CDK5, CDKN2A, CDKN2B, CHEK2, CREBBP, E2F4, ELP3, GAPDH, GATA4, HIST1H3C, HIST1H4A, HMGB1, IKBKB, IRAK2, ITK, KRAS, MLL5, MPP6, NOP2, NSUN3, PHLPP1, SMAD2, SMC1A, SUPT5H, TADA3, TAF6, TSLP, YWHAE*

Kaynak: *http://cbdm-01.zdv.uni-mainz.de/~jfontain/cms/?page_id=592*

EK-8 *KIRC* klinik veri setinde $k = 4, 5, 6, 7, 8, 9, 10$ değerlerine göre gerçekleştirilen deneylerle elde edilen alt-ağ



EK-9 *KIRC* deney verisindeki testlere göre kaydedilen en son *CNA*'lı genler ile bazı hastalıkların ya da biyolojik değişikliklerin ilişki tablosu

Hastalık listesi	Gen sayısı	Genlerin yüzdesi	Kat değişimi	P-değeri	FDR	Gen sembolleri *
Liver Neoplasms	5	0.13	0.90	6.661e-01	9.991e-01	AKT1[66], RHOC[7], CDK2[12], CEBPB[12], TLN1[5]
Cell Transformation, Neoplastic	6	0.16	1.19	3.918e-01	1.175e+00	AKT1[54], CDK2[15], CEBPB[7], ITGA2[7], SMAD3[19], UBC[6]
Carcinoma, Hepatocellular	6	0.16	1.18	4.006e-01	8.012e-01	AKT1[56], RHOC[7], CDK2[12], CEBPB[13], SMAD3[12], TLN1[5]
Breast Neoplasms	7	0.18	0.69	9.141e-01	1.097e+00	#AKT1[200], RHOC[22], CDK2[34], CEBPB[20], SMAD3[27], UBC[13], RBX1[10]
Neoplasm Invasiveness	8	0.21	1.29	2.746e-01	1.648e+00	#AKT1[101], RHOC[35], CEBPB[9], ITGA2[14], ITGB5[5], SMAD3[20], TLN1[8], RBX1[5]
Genetic Predisposition to Disease	13	0.34	0.57	9.996e-01	9.996e-01	ACO1[6], AKT1[54], CDK2[16], DLD[6], ITGA2[50], KLC1[10], SMAD3[33], #NOS3[200], PTPRC[18], SDHC[14], VCP[21], H6PD[5], TLR7[34]

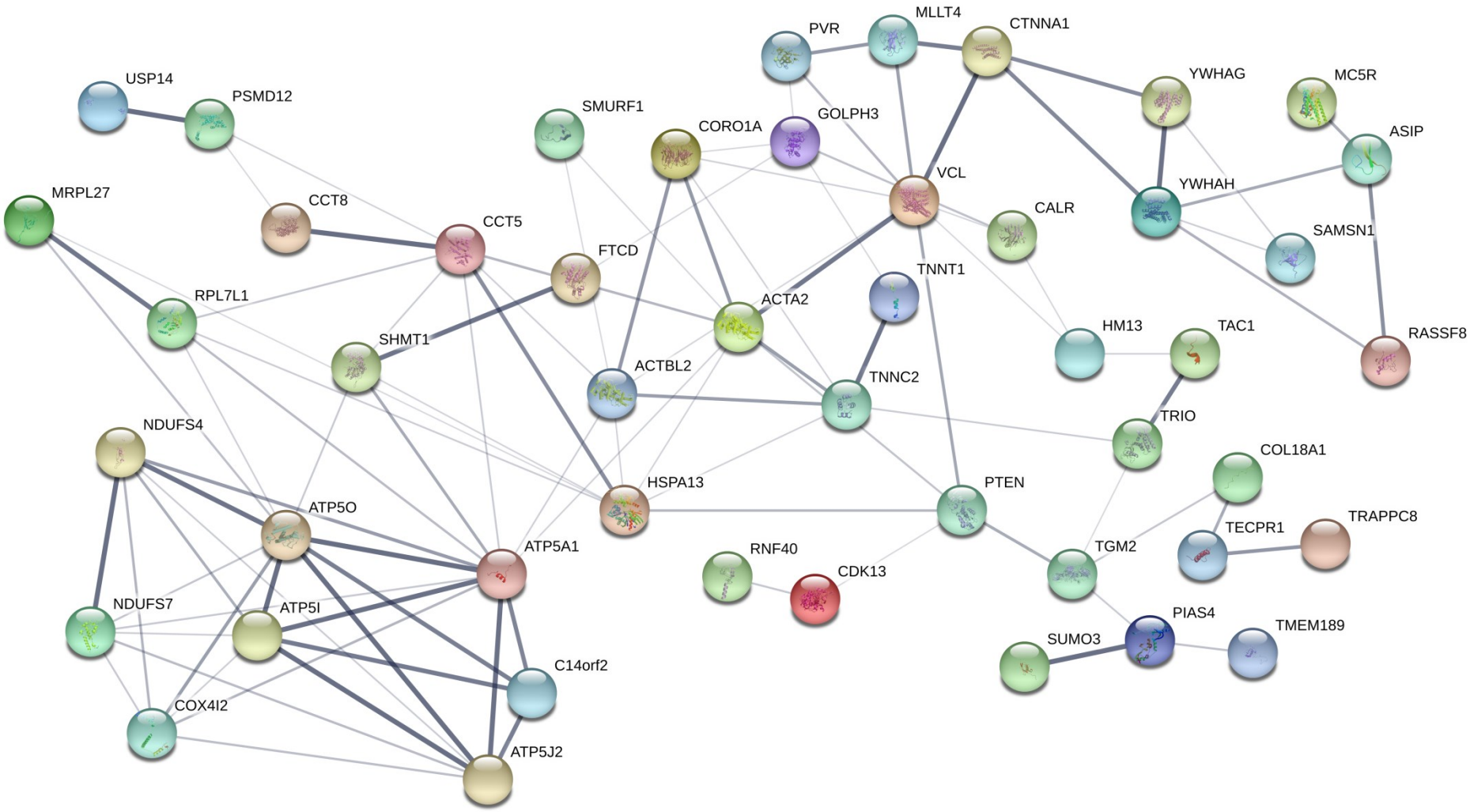
***** Her bir gen üzerindeki sayı, hastalıkla ilgili *PubMed* veri tabanında yayımlanmış toplam çalışma sayısını gösterir.

ile belirtilen genlerle ilgili *PubMed* iletişim link (*hyperlink*) adresleri çok uzun olduğu için atıf yapılan çalışmaların *PMIDs* numaralarına, EK-9-doküman'da erişilebilir.

Genler: *ACO1, ACTA1, ACTR3B, ACTRT2, AKT1, ANAPC4, ATRX, CD72, CDK2, CEBPB, COL4A5, DCLRE1A, DCTN2, DLD, DNAI1, FBXO6, FLNB, H6PD, IMPDH2, ITGA2, ITGB5, KLC1, MAD2L2, MAPK11, MAPKAPK3, NOS3, PDHB, PIAS1, PTPRC, PYGL, RBX1, RHOC, SDHC, SMAD3, TAF1L, TAF9, TLN1, TLR7, UBC, UBE2G2, VCP, YME1L1*

Kaynak: *http://cbdm-01.zdv.uni-mainz.de/~jfontain/cms/?page_id=592*

EK-10 *STAD* veri seti ile yapılan testlerin sonucunda elde edilen tüm genlere göre oluşturulan bağlı gen-etkileşim ağının temsili görüntüsü



EK-11 *STAD* veri setindeki testlere göre elde edilen tüm genlerin (*48 adet*) ilgili hastalıklarla ilişkisi

Hastalık listesi	Gen sayısı	Genlerin yüzdesi	Kat değişimi	P-değeri	FDR	Gen sembolleri *
Neoplasm Invasiveness	5	0.14	0.85	7.231e-01	2.169e+00	CTNNA1[6], #PTEN[102], TGM2[16], TRIO[5], GOLPH3[6]
Breast Neoplasms	7	0.19	0.73	8.823e-01	1.323e+00	CTNNA1[10], #PTEN[152], SHMT1[11], TGM2[22], YWHAG[6], YWHAH[5], SMURF1[6]
Genetic Predisposition to Disease	8	0.22	0.37	1.000e+00	1.000e+00	ACTA2[15], ASIP[9], NDUFS4[5], PTEN[79], SHMT1[38], TAC1[13], YWHAH[5], COL18A1[12]

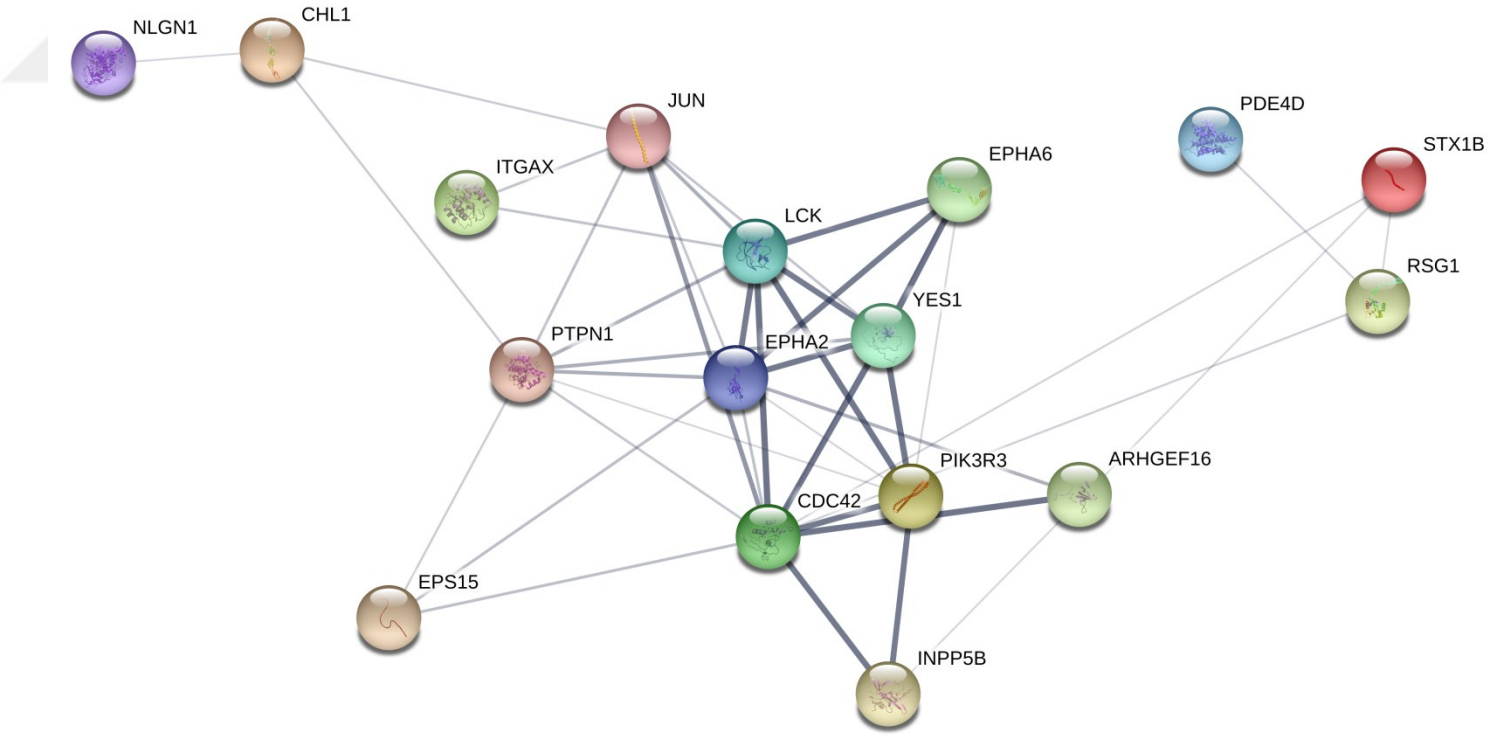
***** Her bir gen üzerindeki sayı, hastalıkla ilgili *PubMed* veri tabanında yayımlanmış toplam çalışma sayısını gösterir.

ile belirtilen genle ilgili *PubMed* iletişim link (*hyperlink*) adresleri çok uzun olduğu için atıf yapılan çalışmaların *PMIDs* numaralarına, EK-11-doküman'da erişilebilir.

Genler: *ACTA2, ACTBL2, ASIP, ATP5A1, ATP5I, ATP5J2, ATP5O, C14orf2, CALR, CCT5, CCT8, CDK13, COL18A1, CORO1A, COX4I2, CTNNA1, FTCD, GOLPH3, HM13, HSPA13, MC5R, MLLT4, MRPL27, NDUFS4, NDUFS7, PIAS4, PSMD12, PTEN, PVR, RASSF8, RNF40, RPL7L1, SAMSN1, SHMT1, SMURF1, SUMO3, TAC1, TECPR1, TGM2, TMEM189, TNNC2, TNNT1, TRAPPC8, TRIO, USP14, VCL, YWHAG, YWHAH*

Kaynak: *http://cbdm-01.zdv.uni-mainz.de/~jfontain/cms/?page_id=592*

EK-12 "*LAML—modül*" alt-ağında bulunan genler ve bunlar arası bağlantılar



EK-13 "*LAML—modül*" verisindeki genlerden bazılarının dört farklı hastalıkla ya da biyolojik değişiklikle ilişkisini gösteren tablo

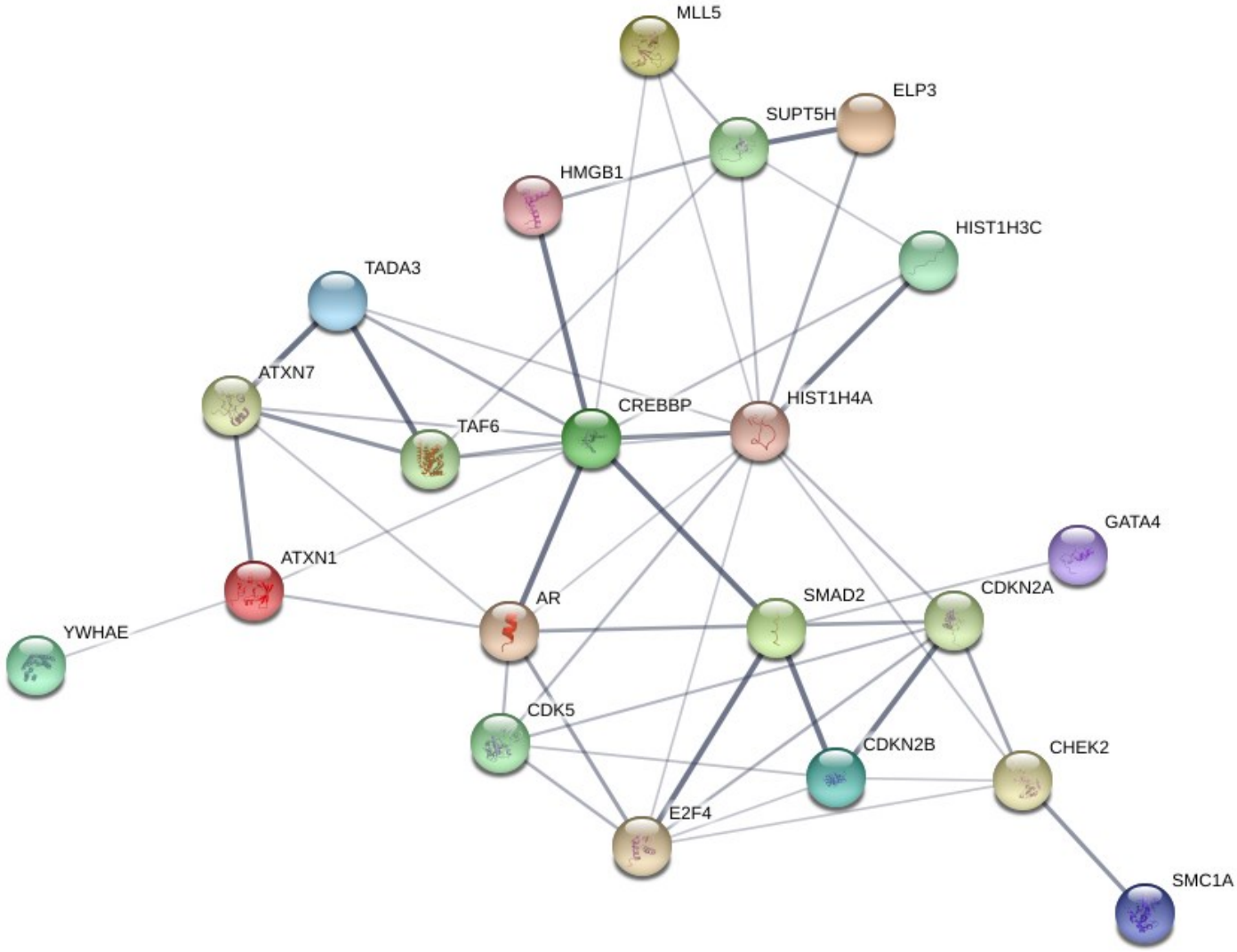
Hastalık listesi	Gen sayısı	Genlerin yüzdesi	Kat değişimi	P-değeri	FDR	Gen sembolleri *
Breast Neoplasms	3	0.19	0.70	8.418e-01	1.122e+00	CDC42[19], EPHA2[18], JUN[42]
Cell Transformation, Neoplastic	5	0.31	2.35	5.072e-02	2.029e-01	CDC42[14], EPHA2[6], JUN[22], LCK[5], PTPN1[6]
Neoplasm Invasiveness	5	0.31	1.91	1.061e-01	2.122e-01	CDC42[25], EPHA2[18], JUN[33], PTPN1[6], PIK3R3[5]
Genetic Predisposition to Disease	5	0.31	0.52	9.947e-01	9.947e-01	ITGAX[6], PDE4D[52], PTPN1[16], CHL1[8], NLGN1[6]

***** Her bir gen üzerindeki sayı, hastalıkla ilgili *PubMed* veri tabanında yayımlanmış toplam çalışma sayısını gösterir. Ek bilgilere, "EK-13-doküman" dosyasından ulaşılabilir.

Genler: *ARHGEF16, CDC42, CHL1, EPHA2, EPHA6, EPS15, INPP5B, ITGAX, JUN, LCK, NLGN1, PDE4D, PIK3R3, PTPN1, RSG1, STX1B, YES1*

Kaynak: *http://cbdm-01.zdv.uni-mainz.de/~jfontain/cms/?page_id=592*

EK-14 "*HNSC—modül*" alt-ağın temsili görüntüsü



EK-15 "*HNSC—modül*" alt-ağındaki genlerin farklı hastalıkla bağlantılarının olduğunu gösteren gen-hastalık listesi (**1**)

Hastalık listesi	Gen sayısı	Genlerin yüzdesi	Kat değişimi	P-değeri	FDR	Gen sembolleri *
Endometrial Neoplasms	3	0.15	4.60	2.597e-02	3.116e-01	AR[15], CDKN2A[20], CHEK2[5]
Osteosarcoma	3	0.15	4.24	3.213e-02	1.928e-01	CDKN2A[19], CREBBP[8], YWHAE[5]
Colonic Neoplasms	3	0.15	1.74	2.458e-01	5.899e-01	CREBBP[6], SMAD2[7], CHEK2[13]
Neoplasm Metastasis	3	0.15	1.34	3.898e-01	7.796e-01	AR[32], HMGB1[15], SMAD2[14]
Cell Transformation, Neoplastic	3	0.15	1.13	5.068e-01	7.602e-01	CDKN2A[72], CREBBP[14], SMAD2[9]
Carcinoma, Hepatocellular	3	0.15	1.12	5.133e-01	6.844e-01	CREBBP[7], SMAD2[10], YWHAE[6]
Prostatic Neoplasms	3	0.15	1.00	5.965e-01	7.158e-01	#AR[200], CREBBP[12], CHEK2[16]
Carcinoma	4	0.20	2.68	5.731e-02	2.292e-01	AR[35], CDKN2A[43], GATA4[5], CHEK2[9]
Colorectal Neoplasms	4	0.20	1.71	1.990e-01	5.970e-01	CDKN2A[51], E2F4[6], SMAD2[11], CHEK2[27]
Neoplasm Invasiveness	4	0.20	1.22	4.180e-01	7.166e-01	AR[39], CDKN2A[50], HMGB1[25], SMAD2[16]

***** Her bir gen üzerindeki sayı, hastalıkla ilgili *PubMed* veri tabanında yayımlanmış toplam çalışma sayısını gösterir.

ile belirtilen genle ilgili *PubMed* iletişim link (*hyperlink*) adresleri çok uzun olduğu için atıf yapılan çalışmaların *PMIDs* numaralarına, EK-15-doküman'da erişilebilir.

EK-15 "*HNSC—modül*" alt-ağındaki genlerin farklı hastalıkla bağlantılarının olduğunu gösteren gen-hastalık listesi (**2**)

Hastalık listesi	Gen sayısı	Genlerin yüzdesi	Kat değişimi	P-değeri	FDR	Gen sembolleri *
Breast Neoplasms	5	0.25	0.94	6.509e-01	7.101e-01	#AR[110], CREBBP[22], E2F4[9], SMAD2[21], #CHEK2[139]
Genetic Predisposition to Disease	9	0.45	0.75	9.399e-01	9.399e-01	#AR[163], CDK5[9], #CDKN2A[200], #CDKN2B[123], CREBBP[11], GATA4[19], SMAD2[10], YWHAE[9], #CHEK2[133]

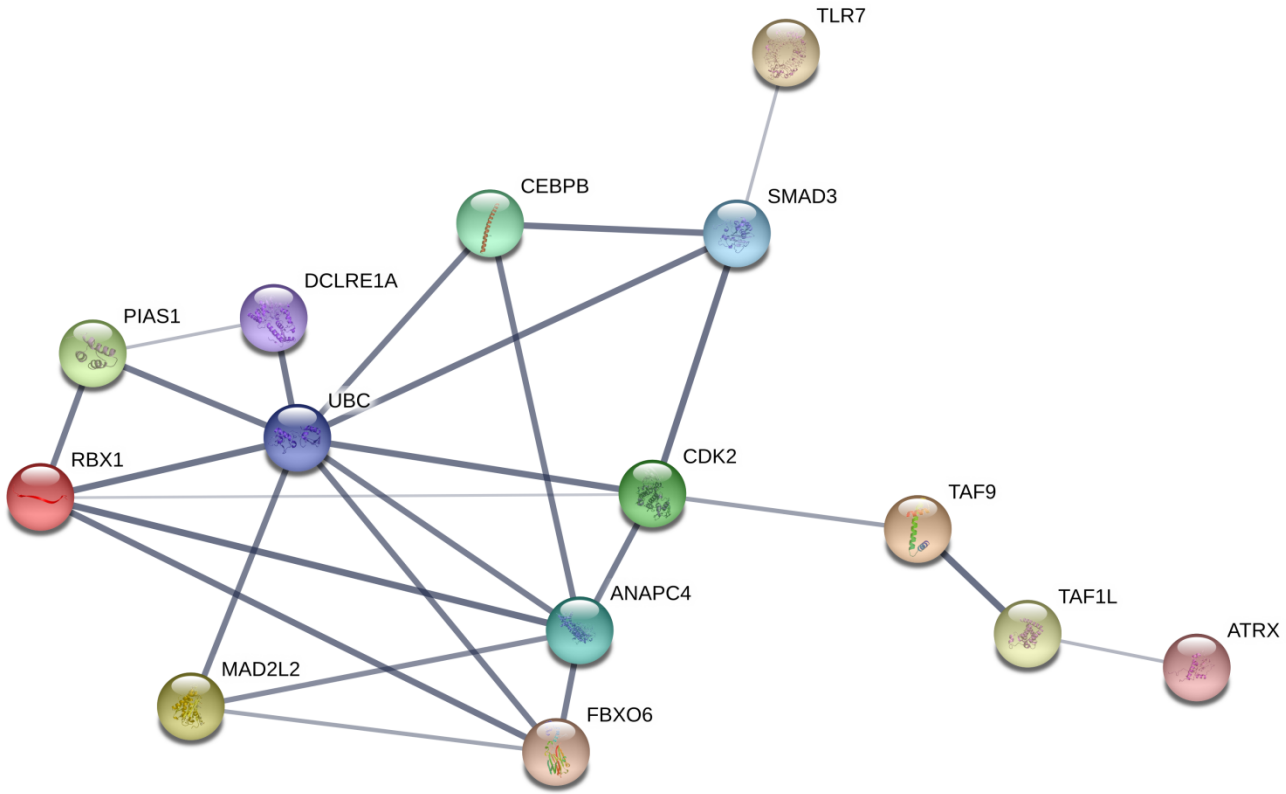
***** Her bir gen üzerindeki sayı, hastalıkla ilgili *PubMed* veri tabanında yayımlanmış toplam çalışma sayısını gösterir. Ek bilgilere, "EK-15-doküman" dosyasından ulaşılabilir.

ile belirtilen genlerle ilgili *PubMed* iletişim link (*hyperlink*) adresleri çok uzun olduğu için atıf yapılan çalışmaların *PMIDs* numaralarına, EK-15-doküman'da erişilebilir.

Genler: *AR, ATXN1, ATXN7, CDK5, CDKN2A, CDKN2B, CHEK2, CREBBP, E2F4, ELP3, GATA4, HIST1H3C, HIST1H4A, HMGB1, MLL5, SMAD2, SMC1A, SUPT5H, TADA3, TAF6, YWHAE*

Kaynak: *http://cbdm-01.zdv.uni-mainz.de/~jfontain/cms/?page_id=592*

EK-16 "*KIRC—modül*" için oluşturulan genler arası etkileşim ağı



EK-17 "*KIRC—modül*" ile ilgili genler ve hastalıklar arası ilişkileri ifade eden tablo

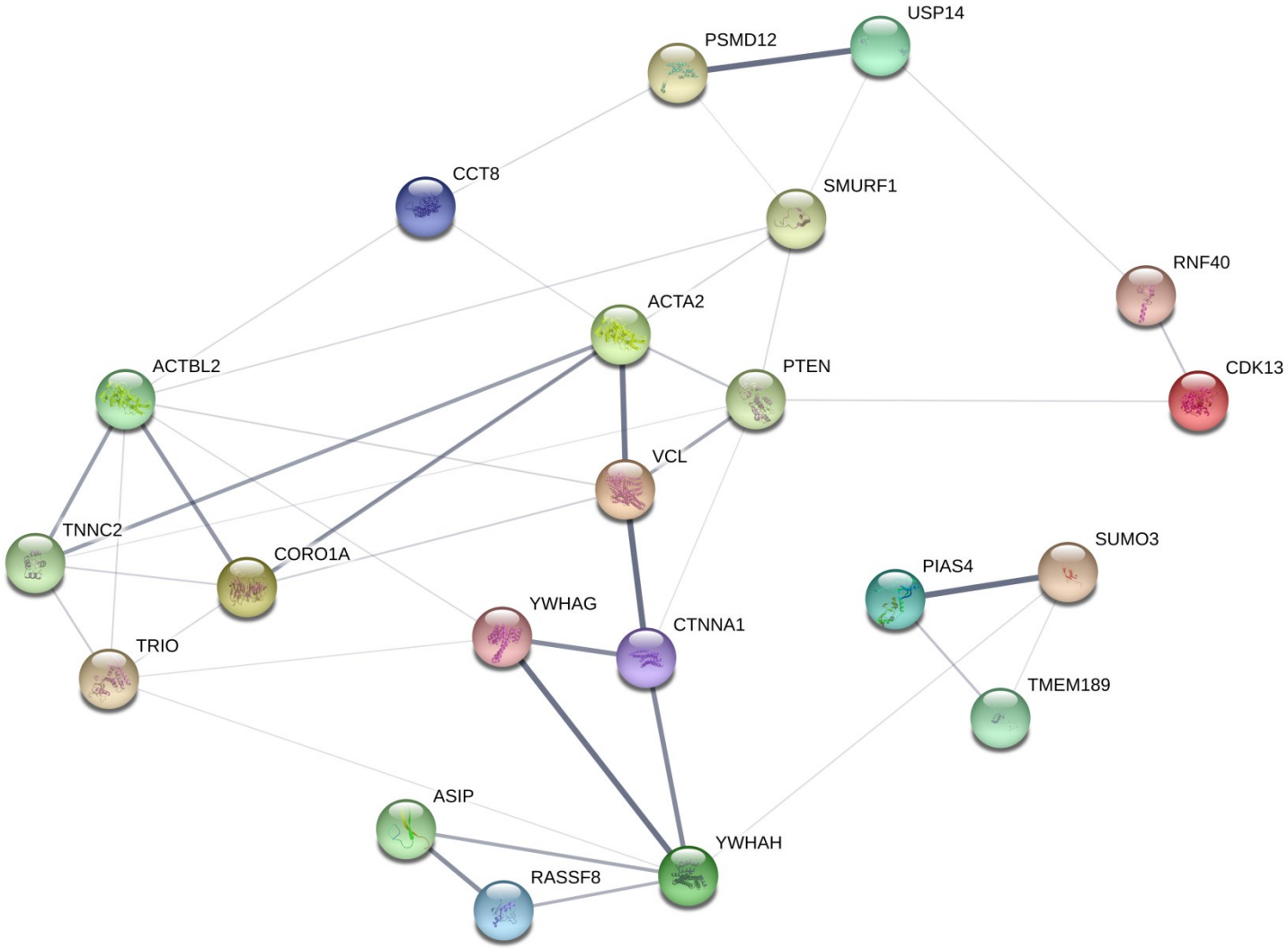
Hastalık listesi	Gen sayısı	Genlerin yüzdesi	Kat değişimi	P-değeri	FDR	Gen sembolleri *
Carcinoma, Hepatocellular	3	0.25	1.87	2.103e-01	3.155e-01	CDK2[12], CEBPB[13], SMAD3[12]
Neoplasm Invasiveness	3	0.25	1.53	3.112e-01	3.734e-01	CEBPB[9], SMAD3[20], RBX1[5]
Genetic Predisposition to Disease	3	0.25	0.42	9.970e-01	9.970e-01	CDK2[16], SMAD3[33], TLR7[34]
Inflammation	4	0.33	2.82	4.431e-02	2.659e-01	CEBPB[8], UBC[10], PIAS1[5], TLR7[18]
Cell Transformation, Neoplastic	4	0.33	2.51	6.355e-02	1.906e-01	CDK2[15], CEBPB[7], SMAD3[19], UBC[6]
Breast Neoplasms	5	0.42	1.56	1.942e-01	3.884e-01	CDK2[34], CEBPB[20], SMAD3[27], UBC[13], RBX1[10]

***** Her bir gen üzerindeki sayı, hastalıkla ilgili *PubMed* veri tabanında yayımlanmış toplam çalışma sayısını gösterir. Ek bilgilere, "EK-17-doküman" dosyasından ulaşılabilir.

Genler: *ANAPC4, ATRX, CDK2, CEBPB, DCLRE1A, FBXO6, MAD2L2, PIAS1, RBX1, SMAD3, TAF1L, TAF9, TLR7, UBC*

Kaynak: *http://cbdm-01.zdv.uni-mainz.de/~jfontain/cms/?page_id=592*

EK-18 "*STAD—modül*" kümesindeki genlerin oluşturduğu etkileşim ağının *STRING* yardımıyla elde edilen görüntüsü



EK-19 "*STAD—modül*" kümesindeki genlerin en az 3 tanesinin ilişkili olduğu hastalıkların listesi

Hastalık listesi	Gen sayısı	Genlerin yüzdesi	Kat değişimi	P-değeri	FDR	Gen sembolleri *
Neoplasm Invasiveness	3	0.17	1.02	5.833e-01	8.750e-01	CTNNA1[6], #PTEN[102], TRIO[5]
Genetic Predisposition to Disease	4	0.22	0.37	9.998e-01	9.998e-01	ACTA2[15], ASIP[9], PTEN[79], YWHAH[5]
Breast Neoplasms	5	0.28	1.04	5.473e-01	1.642e+00	CTNNA1[10], #PTEN[152], YWHAG[6], YWHAH[5], SMURF1[6]

***** Her bir gen üzerindeki sayı, hastalıkla ilgili *PubMed* veri tabanında yayımlanmış toplam çalışma sayısını gösterir. Ek bilgilere, "EK-19-doküman" dosyasından ulaşılabilir.

ile belirtilen genle ilgili *PubMed* iletişim link (*hyperlink*) adresleri çok uzun olduğu için atıf yapılan çalışmaların *PMIDs* numaralarına, EK-19-doküman'da erişilebilir.

Genler: *ACTA2, ACTBL2, ASIP, CCT8, CDK13, CORO1A, CTNNA1, MC5R, PIAS4, PSMD12, PTEN, RASSF8, RNF40, SMURF1, SUMO3, TMEM189, TNNC2, TRIO, USP14, VCL, YWHAG, YWHAH*

Kaynak: *http://cbdm-01.zdv.uni-mainz.de/~jfontain/cms/?page_id=592*

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Yılmaz Atay
Uyruğu : T.C.
Doğum Yeri ve Tarihi : Adana – 18.05.1987
Telefon : +90 5446781987
E-mail : *yilmazatay@selcuk.edu.tr*, *ylmzatay@gmail.com*

EĞİTİM

Derece		Adı, İlçe, İl	Yılı
Lise	:	Sunar Nuri Çomu Lisesi, Yüreğir, Adana, Türkiye	2004
Üniversite	:	S. Ü. Bilgisayar Mühendisliği, Selçuklu, Konya, Türkiye	2010
Yüksek Lisans	:	S. Ü. Bilgisayar Müh. A.B.D., Selçuklu, Konya, Türkiye	2012
Doktora (*araştırma*)	:	Florida Üni. CISE, Gainesville, Florida, ABD	2017
Doktora	:	S. Ü. Bilgisayar Müh. A.B.D., Selçuklu, Konya, Türkiye	2018

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2018	Osmaniye Korkut Ata Üniversitesi Bilgisayar Müh. Bölümü	Arş. Gör.
2011-2018	Selçuk Üniversitesi Bilgisayar Mühendisliği Bölümü	Arş. Gör.
2011	Osmaniye Korkut Ata Üniversitesi Bilgisayar Müh. Bölümü	Arş. Gör.

UZMANLIK ALANI

Bilgisayar bilimleri, biyoinformatik, yapay zeka, metasezgisel algoritmalar, ağ analizi ve çizge madenciliği

YABANCI DİLLER

İngilizce

SEÇİLEN YAYINLAR

Atay, Y. ve Kodaz, H., Optimization of job shop scheduling problems using modified clonal selection algorithm, Turkish Journal of Electrical Engineering & Computer Sciences, 2013. (SCI-Expanded/SCI/AHCI; Yüksek Lisans Tezinden)

Atay, Yilmaz ve Kodaz, Halife, A New Adaptive Genetic Algorithm for Community Structure Detection, Intelligent and Evolutionary Systems, Springer International Publishing, 2016. 43- 55. (Doktora Tezinden)

Atay, Y., Koc, I., Babaoglu, I., & Kodaz, H., Community detection from biological and social networks: A comparative analysis of metaheuristic algorithms, Applied Soft Computing—ASOC, 50, 194-211. (SCI-Expanded/SCI/AHCI; Doktora Tezinden)

Atay, Y., Aslan, M. ve Kodaz, H., A Swarm Intelligence-Based Hybrid Approach for Identifying Network Modules, Journal of Computational Science—JoCS. (SCI-Expanded/SCI/AHCI; Doktora Tezinden)

TEZ İLE İLGİLİ SEÇİLEN KONFERANS ÇALIŞMALARI

2014 Atay, Y. ve Kodaz, H., Biological sequence alignment using artificial immune system based algorithm, International Journal of Arts & Sciences (IJAS), 10-13 Haziran 2014, Anglo-American Üniversitesi (AAU), Prag, Çek Cumhuriyeti.

2015 Atay, Y. ve Kodaz, H., Network Motif Detection in PPI Networks and Effect of R Parameter on System Performance, ICAT'15, Antalya, Türkiye (IJAMEC).

2015 Atay, Y. ve Kodaz, H., Modularity-Based Graph Clustering using Harmony Search Algorithm, ACSAT (4th), Kuala Lumpur-Malezya {IEEE Xplore: 2016}

2017 Atay, Y., Aslan, M., ve Kodaz, H., Finding the Overlapping Modules in Weighted and Directed Networks, 3rd International Conference on Science, Ecology and Technology - ICONSETE'2017, Roma, İtalya.

2017 Atay, Yilmaz, Ismail Koc ve Mehmet Beskirli. "Detection of Cohesive Subgroups in Social Networks using Invasive Weed Optimization Algorithm", ICRES 2017: International Conference on Research in Education and Science, 18-21 Mayıs 2018, Kuşadası, Türkiye.

2017 Atay, Y., ve Kodaz, H., Identification of Meaningful Subnets from Clinical Cancer Datasets by the Proposed Dynamic Programming Method, International Advanced Researches and Engineering Congres, IAREC'17, Osmaniye, Türkiye.

BİLİMSEL VE ARAŞTIRMA PROJELERİ

2012—Selçuk Üniversitesi BAP, Yüksek Lisans Tez Desteği Projesi, Yapay Bağışıklık Sistemleri ile Atölye Tipi Çizelgeleme Problemlerinin Optimizasyonu, Proje Numarası: 12101020, Proje Danışmanı: Doç. Dr. Halife Kodaz.

2014—Türkiye Bilimsel ve Teknolojik Araştırma Kurumu (TÜBİTAK); 2211–C Yurt İçi Öncelikli Alanlar Doktora Burs Programı, Proje Numarası: 1649B031402383.

2016—TÜBİTAK-2214/A Yurt Dışı Doktora Sırası Araştırma Burs Programı, Florida Üniversitesi, CISE, Gainesville, Florida, ABD, Yurt Dışı Proje Danışmanı: Prof. Dr. Tamer Kahveci, Proje Numarası: 1059B141500230.

2018—Kanserli Hastaların Gen/Protein Etkileşim Ağlarından Anlamlı Alt Kümelerin Tespit Edilmesi, Selçuk Üniv. Bilimsel Araştırma Projesi, Proje No: 17401157.