KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DEĞİŞTİRİLMİŞ ATEŞ BÖCEĞİ ALGORİTMASI VE VERİ YOĞUNLUĞU KÜMELEMESİNE UYGULANMASI

DOKTORA TEZİ

Aref YELGHI

ŞUBAT 2018
TRABZON

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DEĞİŞTİRİLMİŞ ATEŞ BÖCEĞİ ALGORİTMASI VE VERİ YOĞUNLUĞU KÜMELEMESİNE UYGULANMASI

Aref YELGHI

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünce
"DOKTOR (BİLGİSAYAR MÜHENDİSLİĞİ)"
Unvanı Verilmesi İçin Kabul Edilen Tezdir.

Tezin Enstitüye Verildiği Tarih : 23 / 01 /2018
Tezin Savunma Tarihi : 16 / 02 /2018

Tez Danışmanı : Prof. Dr. Cemal KÖSE

Trabzon 2018

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bilgisayar Mühendisliği Anabilim Dalında
Aref YELGHİ Tarafından hazırlanan

DEĞİŞTİRİLMİŞ ATEŞ BÖCEĞİ ALGORİTMASI VE VERİ YOĞUNLUĞU KÜMELEMESİNE UYGULANMASI

başlıklı bu çalışma, Enstitü Yönetim Kurulunun 30 /01 / 2018 gün ve 1738 sayılı kararıyla oluşturulan jüri tarafından yapılan sınavda
DOKTORA TEZİ
olarak kabul edilmiştir.

Jüri Üyeleri

Başkan : Prof. Dr. Derviş KARABOĞA .....

Üye : Prof. Dr. Cemal KÖSE .....

Üye : Prof. Dr. Temel KAYIKÇIOĞLU .....

Üye : Doç. Dr. Hasan BULUT .....

Üye : Yrd. Doç. Dr. Bekir DİZDAROĞLU .....

Prof. Dr. Sadettin KORKMAZ
Enstitü Müdürü

ÖNSÖZ

Sürü zekasından esinlenilen optimizasyon teknikleri son yıllarda giderek daha popüler hale gelmiştir. Bu tekniklerin geleneksel tekniklere göre yararlılarından en önemlileri sağlamlık ve esnekliktir. Bu özellikler, sürü zekasını giderek daha karmaşık problemlerle uğraşan algoritmalar için başarılı bir tasarım paradigması haline getirmiştir.

Son çalışmalarda, algoritmalar bölgesel minimumda takılıp kaldığı anda küresel minimumu nasıl bulunacağı söz konusu olmuştur. Ayrıca, bölgesel ve küresel arama arasında bir denge sağlama konusunda yeni geliştirilmiş algoritmalar sunulmuştur. Bu tez çalışması kapsamında, popüler olan ateş böceği algoritması gelgit kuvveti ile geliştirilmiş ve bölgesel-küresel aramada dengeyi sağlamak amacıyla yeni üst sezgisel algoritma önerilmiştir. Önerilen algoritma veri setlerinin üzerinde verilerinin yoğunluğuna dayalı kümeleme yapmak amacı ile kullanılmıştır. Algoritma kullanarak gelişi güzel veri setleri, küme sayısı ve örtüşme gibi problemlerin ortadan kaldırılması hedeflenmiş ve yeni kavramlar kazandırılmıştır.

Doktora çalışmamda danışmanlığımı üstlenerek bana çalışmalarımı yürütmemde her zaman sınırsız destek veren hocam Prof. Dr. Cemal KÖSE' ye, çalışma sürecinde değerli görüş ve katkılarını esirgemeyen sayın Prof. Dr. Temel Kayıkçıoğlu'na ve Yrd. Doç. Dr. Bekir Dizdaroğlu'na teşekkürü bir borç bilirim. Doktora tezimde yer alan arkadaşım Saeid AGAHİAN 'a ve Doktora tez süresi boyunca maddi desteği olan Yurtdışı Türkler ve Akraba Topluluklar Başkanlığına da sonsuz minnettarlığımı sunarım. Ayrıca ilgiyi ve desteği üzerimden eksik etmeyen sevgili anneme çok teşekkür ederim.

Aref YELGHI

Trabzon 2018

TEZ ETİK BEYANNAMESİ

Doktora Tezi olarak sunduğum "DEĞİŞTİRİLMİŞ ATEŞ BÖCEĞİ ALGORİTMASI VE VERİ YOĞUNLUĞU KÜMELEMESİNE UYGULANMASI" başlıklı bu çalışmayı baştan sona kadar danışmanım Prof. Dr. Cemal Köse'nin sorumluluğunda tamamladığımı, verileri/örnekleri kendim topladığımı, deneyleri/analizleri ilgili laboratuvarlarda yaptığımı/yaptırdığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 16/02/2018

Aref YELGHİ

İÇİNDEKİLER

Sayfa No

Doktora Tezi

ÖZET

GELİŞTİRİLMİŞ ATEŞBOCEĞİ ALGORİTMASI VE VERİ YOĞUNLUĞU KÜMELEMESİNE UYGULANMASI

Aref YELGHI

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Prof. Dr. Cemal KÖSE
2018, 124 Sayfa

Son yıllarda, Ateş böceği algoritması, algoritmalar bölgesel minimuma takıldığından küresel minimumu nasıl bulunacağını oldukça ele almamışlar. Ayrıca, bölgesel ve küresel arama arasında bir dengeyi kuramamışlar. Bu çalışmada, Gelgit Kuvveti formülü ile Ateş böceği algoritmayı geliştirilmiştir. Önerilen algoritma (FAtidal), optimizasyon alanında yeni bir strateji getirmektedir. Özelliği olan bölgesel arama Gelgit Kuvveti kullanılarak ve işlev uygunluğuyla ilgili bölgesel ve küresel arama aralarında bir denge kurularak uygulanmaktadır. Deney sonuçlarının karşılaştırılması için Plate shaped, Steep Ridges, Unimodal ve Multimodal fonksiyonları kullanılmıştır. FAtidal algoritması diğer mevcut geliştirilmiş Ateş böceği algoritmalarından daha iyi performans göstermektedir. Geliştirilmiş Ateş böceği algoritması ile veri setlerin kümelenmesi için yeni bir strateji önerilmiştir. Kümeleme veri nesnelerini gruplara ayıran bir prosedürdür. Birçok algoritma aynı anda morfoloji, örtüşme ve küme sayısının üstesinden gelememişler ve ayrıca son yıllarda iyi uygulamalardan birisi olan yoğunluk perspektifi kümeleme alanında kullanılmamıştır. Bu çalışmada, iki parametrenin başlatılmasıyla yeni bir bulanık ve DBSCAN'e dayalı kümeleme algoritması (AFD) önerilmiştir ve sonra iki parametre bağlı olmadan optimizasyon algoritmadan önerilen Ateş böceği algoritması veri yoğunluğa dayalı(FAtidal-DBSCAN) kümeleme problemine uygulanmıştır. Denemelerde, Önerilen algoritma son zamanlardaki geliştirilen kümeleme algoritmalarından daha iyi performans sergilemektedir.

Anahtar Kelimeler: Ateş böceği algoritma, Gelgit kuvveti, Sürü zekâsı, Küresel minimum, Kümeleme, Veri madenciliği, Bulanık mantığı.

PhD Thesis

SUMMARY

IMPROVED FIREFLY ALGORITHM AND APPLAY TO CLUSTERING BASED ON DENSITY

Aref YELGHİ

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Computer Engineering Graduate Program
Supervisor: Prof. Dr. Cemal KÖSE
2018, 124 Pages

The Firefly algorithm is a population-based optimization algorithm. It has become popular in the field of optimization and has been applied to engineering practices. Recent works have failed to address how to find the global minimum, because their algorithm was trapped in the local minimum. Also, they were not able to provide a balance between exploration and exploitation. In this work, the Tidal Force formula has been applied to modify the Firefly algorithm. The proposed algorithm FAtidal brings a strategy into the optimization field. It is applied by using exploitation (Tidal Force) and keeping a balance between the exploration and exploitation on function suitability. Plate shaped, Steep Ridges, Unimodal and Multimodal Benchmark functions were used to compare experimental results. The study findings indicate that the Tidal Force Firefly algorithm outperforms the other existing modified Firefly algorithms. Another section of thesis proposes a strategy for clustering of the dataset with improved firefly algorithm. It is a procedure that partition data objects into the groups. Many algorithms could not overcome morphology, overlap and number of clusters problems at the same time. Clustering based on density is one of the best methods for those problems. This study proposed AFD algorithm based on Fuzzy and DBSCAN which works with the initialization of two parameters and FAtidal_DBSCAN algorithm proposed to reduce the sensitive paramters problems. In the experiments, It is demonstrated the proposed algorithms outperforms the other recently developed clustering algorithms.

Key Words: Firefly algorithm, Tidal force, Optimization, Swarm intelligence, Global minimum, Clustering algorithm, Data mining, Fuzzy logic

ŞEKİLLER DİZİNİ

Sayfa No

TABLOLAR DİZİNİ

Sayfa No

SAMBOLLER DİZİNİ

IP	Integer Program
MIP	Mixed İnteger Programming Problems
LP	Linear Programming
NLP	Non Linear Programming
TSP	Travel Salesman Problem
NP	Non Deterministic Polynomial
P	Deterministic Polynomial
GA	Genetic Algorithm
ABC	Artificial Bee Colony
ICA	Imperialist Competitive Algorithm
BA	Bee Algorithm
CA	Cultural Algorithm
ACO	Ant Colony Optimization
BBO	Biogeography-Based Optimization
HSI	Habitat Suitability Index
SIVs	Suitability Index Variables
ICA	Imperialist Competitive Algorithm
FA	Firefly Algorithm
LFA	Light intensity difference Firefly Algorithm
ODFA	Opposition and Dimensional based FA
OBL	Opposition Based Learning
FEN	Function Evalution Number

1. GENEL BİLGİLER

1.1. Giriş

Mühendislik tasarımından iş planlamasına ve internetin rotasından tatil planlamasına sayıl optimizasyonun her yerde olduğunu söylemek abartılı değildir. Hemen hemen tüm bu faaliyetlerde, belirli hedeflere ulaşmaya çalışmakta ya da kâr, kalite ve zaman gibi bir şeyi optimize etmekte kullanılmaktadır. Gerçek dünya uygulamalarında kaynaklar olarak, zaman ve para daima sınırlıdır. Değerli kaynakları en iyi şekilde kullanmak için çeşitli kısıtlamalar altında çözümler bulmak zorundayız. Matematiksel optimizasyon veya programlama, bunu benzer tasarım planlama problemlerinde kullanılmaktadır. Verilen kısıtlamalar altında, istenen faktörleri en üst düzeye çıkararak ve istenmeyen faktörleri en aza indirgeyerek en uygun maliyetli veya en yüksek erişilebilir performansa sahip bir alternatif bulmaktır. Buna karşılık, maksimizasyon, kârda en yüksek sonucu elde etme çalışması denilir yani en az kârlı alternatifi seçmesi demektir. Bir optimizasyonun tasarımında, tasarım amacı basitçe üretim kârını en aza indirgemek veya üretim verimliliğini en üst düzeye çıkarmaktır. Optimizasyon uygulamasında tam bilgi eksikliği ve hangi bilgilerin mevcut olduğunu değerlendirmek için zaman eksikliği ile sınırlamaktadır. Günümüzde bilgisayar simülasyonları çözümü için çeşitli verimli arama algoritmalarıyla bu tür optimizasyon problemleri vazgeçilmez bir araç haline gelmektedir[1] ve destekli tasarım faaliyetlerinin bir parçası olmuştur. Günümüzde yaygın olarak kullanılan iki farklı optimizasyon algoritması türü vardır. Birden fazla matematik işlem gerektiren çok değişkenli matematiksel modelleri, klasik metotlardan yararlanabiliriz. Ama değişkenler arttıkça ve problem modelleri karmaşıksa klasik metotlar başa çıkmamış olabilirler. Başka ifade ile Bir optimizasyon sorununun doğru şekilde formüle edildikten sonra bazı çözüm yöntemleriyle optimal çözümleri bulmak ve en uygun çözümü aramak avcılık hazinesi gibidir. Bir tepenin içinde gizli bir hazine aramaya çalışma amacımız ve zaman sınırlı ise, avcı kör olduğunu varsayalım ve herhangi bir rehberlik olmadan arama çok kütü bir arama göstermektedir. Başka bir taraftan, hazinenin bilinen bir bölgesi en yüksek zirvesinde bulunduğunu söylenirse, doğrudan en dik tırmanışını yapılmaktadır ve en yüksek zirveye

ulaşmaktadır ve bu senaryo klasik tepeye tırmanma[1] denilir. Çoğu durumda, aramamız bu ikisinin arasındadır, körü körüne değiliz ve nereleri arayacağımızı bilmiyoruz [1].

Optimizasyon teknikleri ilk başlarda klasik ve deterministik yöntemlere sınıflandırılmıştır [2]. Deterministik yöntemler ile optimal çözümleri elde ederken bunların en uygun cevabı garanti sağlamaktadır. Bu kategori da ayır ve bağla[2] algoritması, dinamik programlama, Bayes arama algoritmaları[3] ve ardışık yaklaşma yöntemleri[4] içermektedir. Sezgisel algoritmalar makul bir sürede iyi bir çözüm bulmada etkilidir. Üst sezgisel, Sezgiselin bir alt dalı olarak rehberlik mekanizması gibi davranan bir algoritma sınıfı oluşturur. Onlar bir probleme veya alana göre özgü değiller ve herhangi bir optimizasyon problemine uygulanabilir. Üst sezgisel terimi Glover tarafından tanıtılmıştır [3].

Mühendislik tasarım problemlerinde, klasik optimizasyon tekniklerini kullanmak yerine üst sezgisel tercih edilmektedir. Gündemimizde iki çeşitli algoritmalar söz konusudur.

1) Deterministik Algoritmalar. Bir çözümü diğerine taşımak için belirli kurallar kullanıyorlar. Bu algoritmalar bir sürü takım için kullanılıyor ve pek çok mühendislik tasarım problemi için başarılı bir şekilde uygulanıyor.

Deterministinin özellikleri:

- Aynı giriş göz önüne alındığında, her koşulmasında aynı çıkışı üretir.
- Aynı giriş göz önüne alındığında, her çalıştırıldığında aynı miktarda zaman, bellek ve kaynak alınmaktadır.
- Deterministik bir bilgisayarla çok terimli zamanında çözülebilen karmaşıklık sınıfı P'nin problemleri çözebilmekte, yalnızca deterministik olmayan bir bilgisayar kullanarak çokterimli zamanında çözülebilen karmaşıklık sınıfı NP' nın sorunlarına başa çıkmamıştır.

2) Olasılıksal Algoritmalar: Olasılıksal algoritmalar doğasında olasılık kurallara dayanmaktadır. Deterministik algoritmaların sahip olmadığı bazı özellikler nedeniyle popülerlik kazandırmıştır. Modellerin bazı doğal rastlantısalılar vardır. Aynı parametre değerle set ile ve başlangıç koşullar ile farklı bir topluluğa çözüme yol açmıştır.

Bu sınıflara dayalı klasik ve modern sezgisel algoritmaları bu bölümde açıklanacaktır.

Tezin organizasyonu aşağıda belirtiği gibidir:

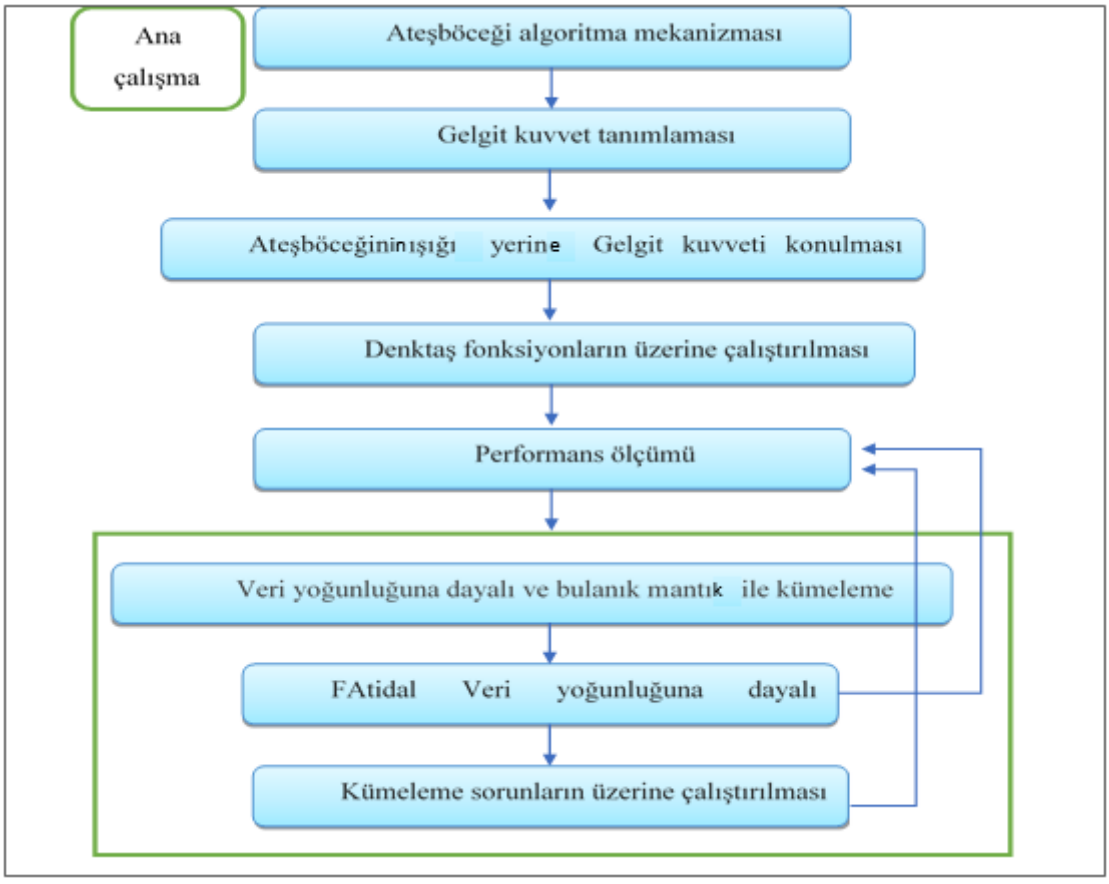
---

[1] classical hill-climbing
[2] Branch -and- bound
[3] Bayesian search algorithms
[4] successive approximation methods

Bu bölümde optimizasyon problemleri, kısaca optimizasyon bileşenlerin tanımını yapılmıştır. İkinci bölümde modern sezgiseli kapsayan yapılan çalışmalar ve ayrıca kümeleme problemi de örnek olay çalışma amacı ile bahis edilmiştir. Üçüncü bulgular bölümde gelgit kuvveti ve Fatidal algoritması açıklanmıştır.

Dördüncü tartışma bölümde önerilen algoritma kıyaslama fonksiyonların üzerinde deneme yapıldı ve elde ettiği bulgularla sonuçları kıyaslayarak tartışılmış ve ayrıca önerilen algoritma veri setlerin üzerinde kümeleme amacıyla iki fazda yapılmıştır. Birinci AFD algoritması basitçe veri yoğunluğuna dayalı otomat kümeleme, ikincisi ise Fatidal_DBSCAN algoritması veri yoğunluğuna dayalı önerilen optimizasyon algoritmasıdır.

Son bölümde ise elde ettiğimiz katkılar ve geleceğe önerilen planlardan bahsedilmiştir. Tezde yapılan çalışma ve kıyaslamanın genel akış Şekil 1'de gösterilmektedir.



Şekil 1.1. Yapılan çalışmanın ve kıyaslama bölümünün genel akışı

1.2. Optimum Tasarım Problemin Formülasyonu

Bu bölümde problem tasarımı, matematiksel anlatımı, değişken tasarımını ve problem için bir fonksiyon ve sınırlama tanıtımı yapılmıştır.

1.2.1. Problem Tanımı

Genel olarak, bir problemin doğru tanım formülasyonu çözülmesi gereken toplam çabanın yaklaşık yüzde 50'sidir. Bu nedenle, tasarım optimizasyon problemlerini formüle etmek için iyi tanımlanmış prosedürü takip etmek önemlidir. Bu metinde genel olarak çeşitli ön analizlerin tamamlandığı ve bir konsept veya alt problemin detaylı bir tasarımın yapılmasını hedeflenmektedir. Optimum çözümü oldukça formülasyonu bağlıdır ve bir tasarım optimizasyon iyi olması için önemi vurgulanmaktadır. Çoğu tasarımda optimizasyon problemleri, aşağıdaki prosedürü kullanmaktadır [4]:

1. Proje / problem açıklaması.
2. Veri ve bilgi toplama.
3. Tasarımın değişkenlerinin tanımlanması.
4. Optimize edilecek kriterin tanımlanması.
5. Sınırlamaların tanımlanması

Problemi tasarlamak için matematiksel bir formülasyonu geliştirmek için beş adım sunulmuştur.

1.2.1.1. Problem Açıklaması

Formülasyon süreci, problem için tanımlayıcı bir ifade geliştirerek başlar. Proje veya problem, genellikle projenin sahibi / sponsoru tarafından yapılmaktadır. Projenin genel hedeflerini ve karşılayacağı gereken şartları açıklanması gerekmektedir.

1.2.1.2. Veri Bilgi Toplama

Problemin matematiksel bir formülasyonu geliştirmek için, malzeme özellikleri, performans gereksinimleri, kaynak sınırları, ham madde maliyeti ve diğer ilgili bilgileri toplamamız gerekir.

1.2.1.3. Tasarım Değişkenlerinin Tanımlanması

Formülasyon sürecindeki bir sonraki adım, adı verilen tasarım değişkenleri sistemi tanımlayan bir değişken seti tanımlamaktadır. Genel olarak bunlara herhangi bir değer atabilmek için serbesttirler. Değişkenler de farklı değerler farklı tasarımlar üretilir. Tasarım değişkenleri mümkün oldukça birbirinden bağımsız olmalıdır. Eğer bağımlıysa değerleri bağımsız olarak belirlenemez. Bağımsız tasarım değişkenlerinin sayısı, problemin tasarım serbestlik derecesi belirlenir. Bazı problemler için, sistem aynı şekilde tanımlamak için farklı değişken setleri tanımlanabilir. Problem formülasyonu seçilen sete bağlı olacaktır. Tasarım değişkenlerine sayısal değerler verildiğinde, sistemin tasarımına ulaşmış oluyoruz. Bu tasarımın tüm şartlarını yerine getirip getirmediği başka bir sorudur.

Değişken tasarımı aşağıdaki gibi özetlenmiştir.

- Tasarım değişkenleri mümkün oldukça birbirinden bağımsız olmalıdır. Eğer öyle değilse, aralarında bazı eşitlik sınırlamalar olmalıdır. Aksi takdirde, problemin eşitlik sınırlaması varsa, tasarımın değişkenlerine bağımlıdırlar.
- Tasarım problemi olarak mümkün oldukça çok sayıda bağımsız parametreleri tasarım değişken şeklinde çıkarılmalı.
- Tasarımdan sonra her değişkene sayısal bir değer verilmelidir [4].

1.2.1.4. Optimize Edilecek Kriterin Tanımlanması

Bir sistem için uygulanabilecek birçok tasarım yapılabilir ve bazıları diğerlerinden daha verimli göstermektedir. Farklı tasarımları birbirleriyle karşılaştırmak için bir kritere sahiptirler. Tasarım problemde sayısal değeri elde edebilmek için, yani tasarımda x değişken vektörünün olması gereken bir sayısal fonksiyonun yapılması gerekir. Böyle bir ölçüt, genellikle optimum tasarım problemler için amaçlı fonksiyon olarak adlandırılır ve problemin gereklerine bağlı

olarak maksimize edilmesi veya en aza indirgenmesi gerekmektedir. En aza indirmesi gereken bir ölçüt, genellikle bu metinde kullanılan terim olan mühendislik literatüründe maliyet fonksiyonu olarak adlandırılır. Bir objektif fonksiyonun, tasarım probleminin değişkenlerine doğrudan veya dolaylı olarak etkilemesi gerektiği önemlidir; Aksi halde, anlamlı bir amaç fonksiyon değildir.

1.2.1.5. Sınırlamaların Tanımlanması

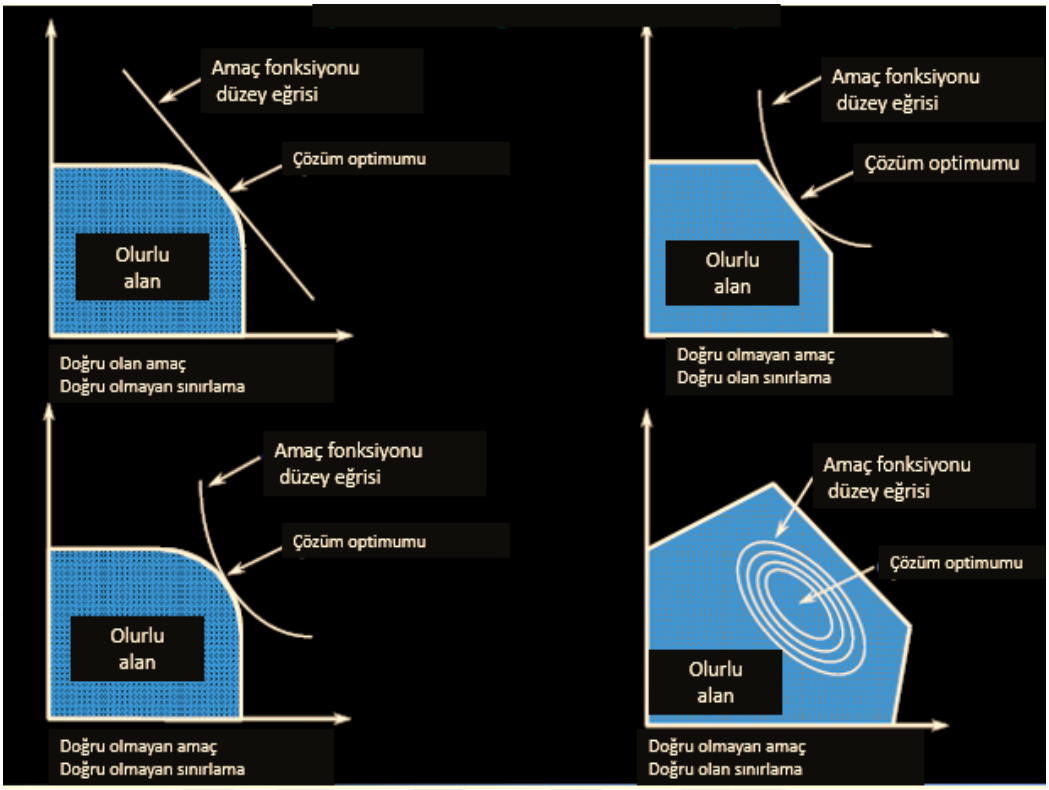
Bir tasarım üzerine yerleştirilen tüm kısıtlamalar sınırlama olarak adlandırılır. Formülasyon sürecindeki son adım, tüm sınırlamalar için ifadeler geliştirmektir. Sınırlamalarla ilgili birkaç kavram ve terim aşağıdaki paragraflarda açıklanmaktadır.

1.2.1.5.1. Doğrusal Olan ve Olmayan Sınırlamalar

Doğrusal olan ve doğrusal olmayan birçok sınırlamalı fonksiyonlar tasarım değişkenlerinde bulunmaktadır. Birinci dereceden terimlere doğrusal sınırlama denir. Doğrusal programlama problemleri sadece lineer/doğrusal sınırlama ve amaç fonksiyonlarına sahiptir. Daha genel problemlerin doğrusal olmayan maliyeti ve sınırlama fonksiyonları mevcuttur. Bunlara doğrusal ve doğrusal olmayan programlama problemleri denir.

1.2.1.5.2. Olurlu Tasarım

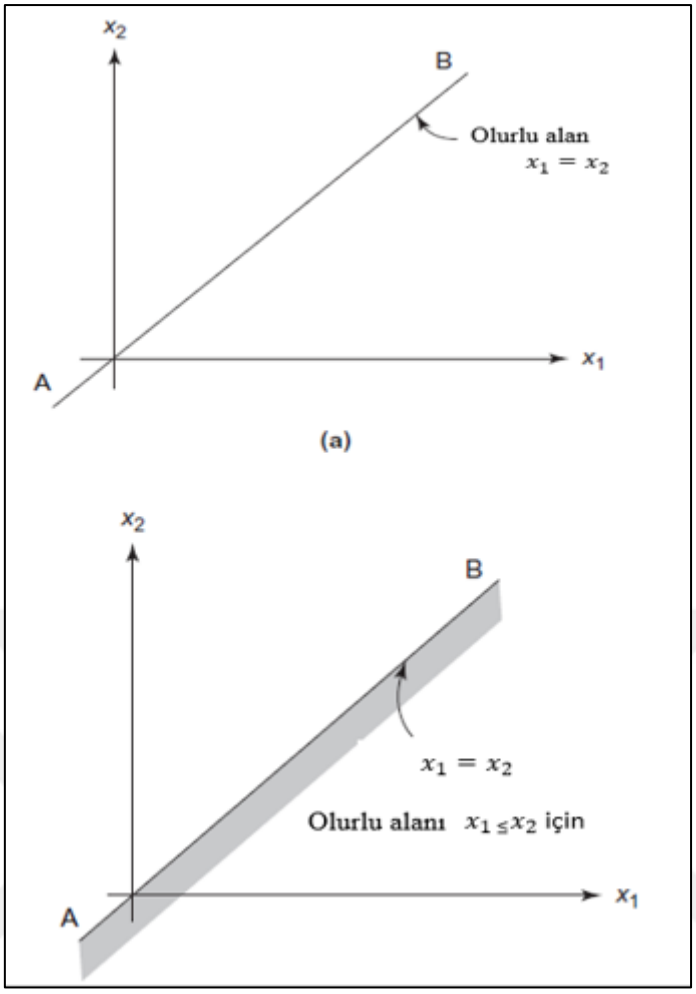
Bir sistemin tasarımı, tasarım değişkenlerine verilen bir dizi sayısal değerlerdir (yani, x belirli bir tasarım değişken vektörü). Bu tasarım manasız olsa da (örneğin, negatif yarıçapı) veya fonksiyon açısından yetersiz olsa da yine de bir tasarım olarak adlandırılabilir. Açıkçası, bazı tasarımlar kullanışlıdır ve bazıları da değildir. Tüm gereksinimleri karşılayan bir tasarım, uygulanabilir bir tasarım (kabul edilebilir veya elverişli) olarak adlandırılır. Olursuz (kabul edilemez) bir tasarım bir veya daha fazla gereksinimi karşılayamıyor demektir. Şekil 1.2'de olurlu alanı ve doğrusal olan ve olmayan sınırlama ve amaç fonksiyonu gösterilmiştir.



Şekil 1.2. Olurlu alan, doğrusal olan v doğrusal olmayan sınırlama ve amaç fonksiyonu [5].

1.2.1.5.3. Eşitlik ve Eşitsizlik Sınırlama

Tasarım problemlerinde hangi gereksinimlerin eşitlik olarak formüle edilmesi gerektiğini ve hangilerinin eşitsizlik olarak tanımlanacağını belirlemek için dikkatli incelenmelidir. Eşitli sınırlamalara göre olurlu bölgenin ifadesi eşitsizlik sınırlamalara göre daha çok kullanılmaktadır. Şekil 1.3'te bu sınırları gösterilmiştir.



Şekil 1.3. Eşitlik(a) ve Eşitsizlik(b) Sınırlamaların olurlu alanları [4].

1.2.1.5.4. Örtük Sınırlama

Örtük sınırlama: Tasarım değişkenlerinin en küçük ve en büyük izin verilen değerleridir ve daha karmaşık değişkenlerde tasarım değişkenlerin tarafından dolaylı olarak etkilenebilir [4].

1.3. Problemin Matematiksel Anlatımı

Matematiksel olarak, çoğu optimizasyon problemini genel formda aşağıdaki gibi gösterilmektedir.

$$\underset{x\in R^n}{\text{Min}} \; f_i(x), \quad (i = 1 \dots M), \tag{1.1}$$

$$h_j(x)=0, \quad (j = 1 \dots J), \tag{1.2}$$

$$g_j(x)=0, \quad (k = 1 \dots K), \tag{1.3}$$

Burada $f_i(x)$, $h_j(x)$ ve $g_j(x)$ tasarım vektörünün fonksiyonlarıdır.

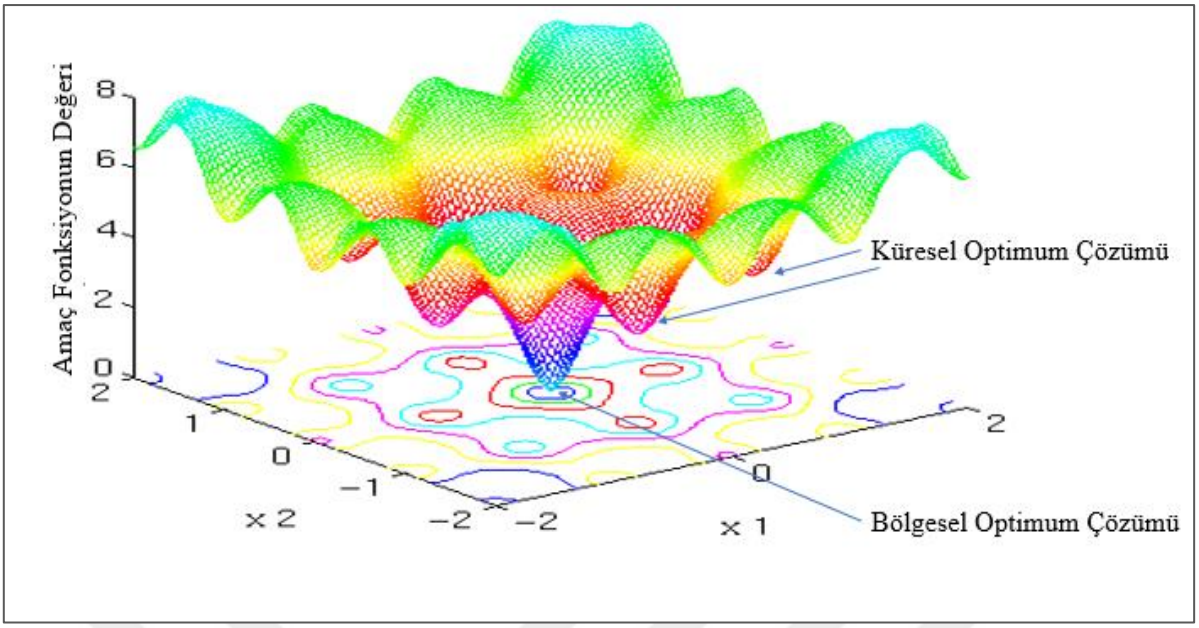
$$x=(x_1,x_2,\dots,x_n)^T \tag{1.4}$$

Burada, x'in x_i bileşenlerine tasarım veya karar değişkenleri denir ve gerçel sürekli ve ayrık veya bu ikisinin de karışımını içermektedir.

$f_i(x)$ fonksiyonu (i=1…M), amaç fonksiyonlarına basitçe maliyet fonksiyonları diyebiliriz ve $M = 1$ durumunda sadece tek hedefi amaçlanmaktadır. Karar değişkenleri tarafından kapsanan alan, $\mathbb{R}^n$ tasarım uzayı veya araştırma uzayı denir, amaç fonksiyon tarafından oluşturulan alan, çözüm uzayı veya cevap uzayı denir. h_j için eşittir ve g_j için eşitsizlikler sınırlama olarak adlandırılır. Dikkat edilmesi gereken husus eşitsizlikleri bu şekilde yazabiliriz > 0 ve hedefleri ise bir maksimizasyon problemi olarak da formüle edebiliriz. Benzer şekilde, optimizasyon J + K sınırlama cinsinden sınıflandırabiliriz [1].

$$\left\{\begin{array}{cc} J=K=0 & \text{sınırlamasız problemi} \\ K=0\,,\, J\geq 1 & \text{eşitlik sınırlamalı problemi} \\ K\geq 1,\, J=0 & \text{eşitsizlik sınırlamalı problemi} \end{array}\right\} \tag{1.5}$$

Şekil 1.4'te gösterildiği gibi çözüm uzayı gösterilmektedir. Klasik algoritmalar bölgesel optimum bulmada müsaittirler ama küresel bölgede optimuma ulaşmasına başa çıkamamışlar ve bundan dolayı popülasyona dayalı optimizasyon algoritmalar ortaya çıkmıştır. Bu tez kapsamında küresel optimum ulaşmak için var olan sorunları çözmek ve gidermek hedeflenmektedir. Bir problemi çözmek için tanımlanan amaç fonksiyonu küresel optimuma ulaşması tasarlanmasıdır.



Şekil 1.4. Çözüm Uzayı [5].

1.4. Problemlerin Üzerine Karar Verme

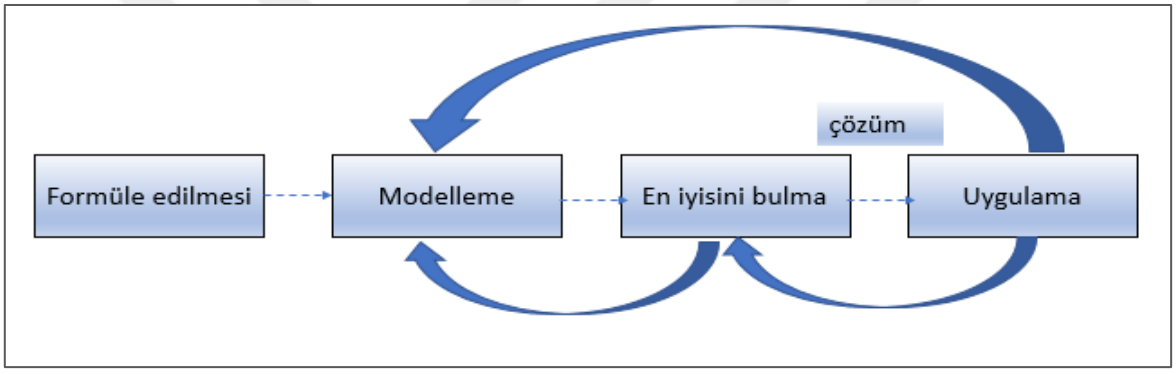
Bilim adamları, mühendisler ve yöneticiler olarak daima karar almak zorundadırlar. Dünyamız gittikçe daha karmaşık ve rekabetçi bir hale gelmekte, mantıklı ve en uygun şekilde karar verme önemlidir. Karar verme prosedürü aşağıdaki aşamalardan oluşur.

Problemi Formüle Edilmesi: Bu ilk adımda, karar problemi belirlenmiştir. Ardından, problemin ilk ifadesi yapılır. Bu Formülasyon kesin olmayabilir. İç ve dış faktörler ile amaç problemlerin ana hatları belirtilmiştir. Birçok karar verici, problemin formüle edilmesine müdahale edebilir.

- Problemin Modellemesinin Yapılması: Bu önemli aşamada, problem için soyut bir matematiksel model oluşturulması beklenilmektedir. Bu problemi iyi çalışılan optimizasyon modele değiştirmesi istenmektedir. Genellikle, çözmekte olduğumuz modeller gerçeğin basitleştirilmesidir. Yaklaşımlar içerirler ve bazen matematiksel bir modelde temsil ettikleri karmaşık süreçleri atlarlar.
- Problemin Eniyilemesi: Problemin modellendikten sonra, çözme prosedürü problem için "iyi" bir çözüm üretir. Çözüm en uygun veya en uyguna yakın olabilir. Orijinal olarak hazırlanmış gerçek hayat problemi çözülmesi değil, problemin soyut modelin çözülmesine dikkat edilmelidir. Bu nedenle, elde edilen çözüm performansları, modelin

doğru olduğunu nitelemektedir. Algoritma tasarımcılar, benzer problemler üzerinde popular olan algoritmaları tekrardan kullanabilir veya bu özel uygulamaların bilgilerini alınarak algoritmalar ile entegre edebilir.

- Problem Uygulaması: Elde edilen çözüm, pratik olarak karar verici tarafından test edilir ve "kabul edilebilir" ise uygulanır. Uygulanacak çözümde bazı pratik bilgiler getirilebilir. Çözüm kabul edilemez ise, model ve / veya optimizasyon algoritması geliştirilmeli ve karar verme süreci tekrar edilmelidir. Klasik karar verme suresi: Uygulamada bu süreç, kabul edilebilir bir çözüm bulunana kadar optimizasyon modeli veya algoritmayı iyileştirmek için tekrarlanabilir. Yazılım mühendisliğinde yaşam döngüleri gibi optimizasyon modellerinin ve algoritmalarının yaşam döngüsü doğrusal, spiral veya basamaklı olabilir. Şekil 1.5 klasik süreci göstermektedir [2].



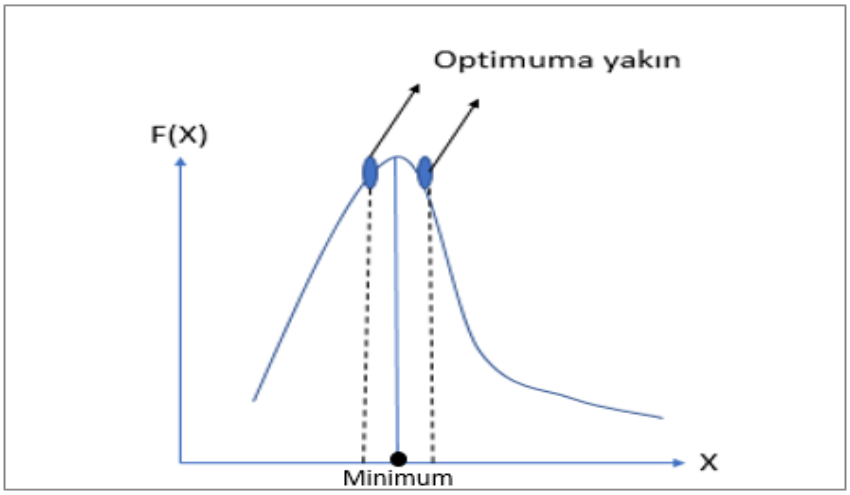
Şekil 1.5. Klasik karar süreci [2].

1.4.1. Optimizasyonun Matematiksel Anlatımı

$A \subset R^n$ Bir küme ve bir fonksiyon $f:A \rightarrow R$ olarak düşünürüz. Bu bölümde f Fonksiyonun (bölgesel veya küresel) maksimum ve / veya minimuma eriştiği A'daki noktaları belirlenmesi için hedeflenmektedir.

- Bölgesel Maksimum: $A \subset R^n$ varsayalım ve $A, \mathcal{R}$ arasında bir f fonksiyonu düşünün: x^*'n yakın çevresinde açık bir yuvarlık varsa, $B_\varepsilon(x^*)$, öyle ki $f(x^*) \geq f(x)$ her $x \in B_\varepsilon(x^*) \cap A$ ise $x^* \in A$ noktası bölgesel maksimum olarak söylenir. Yani x^*'n çevresindeki başka büyük bir değere sahip olan bir nokta bulunmamaktadır.

- Küresel Maksimum: $A \subset R^n$ varsayalım ve $A, \mathcal{R}$ arasında bir f fonksiyonu düşünün: eğer x^* için $f(x^*) \geq f(x)$ her $x \in A$ şartları sağlıyorsa $x^* \in A$ noktası küresel maksimum olarak söylenir. Şurada iki konu söz konusudur:
  (a) Bir çözüm varlığı: Weierstrass Teoremi
  (b) Çözümün tekliği: Sıkı Yarı-içbükeylik
- En yakın iyisini bulma: global komşu minimum değeri global minimuma çok yakın ama daha düşük bir değere sahiptir. Bazı mühendislik problemlerinde problemin doğuştan karmaşıklığı ya da sorunu çözmek için kullanılan yöntemden dolayı mutlak küresel optimum elde etmek son derece zor ya da bazen imkansızdır. Ya da küresel optimumu sağlanması hesaplama açısından engelleyici olabilir. Birçok gerçek dünya probleminin çözülmesinde optimum yakınında tatminkardır. Yakın optimumun küresel optimuma yakınlığı optimizasyon problemi çözülmekte ve analistin kararına bağlıdır. Şekil 1.6'da Maksimizasyon probleminde bir yakın optimum kavramı gösterilmektedir.



Şekil 1.6. Optimuma yakın tek boyutlu bir maksimum problemi

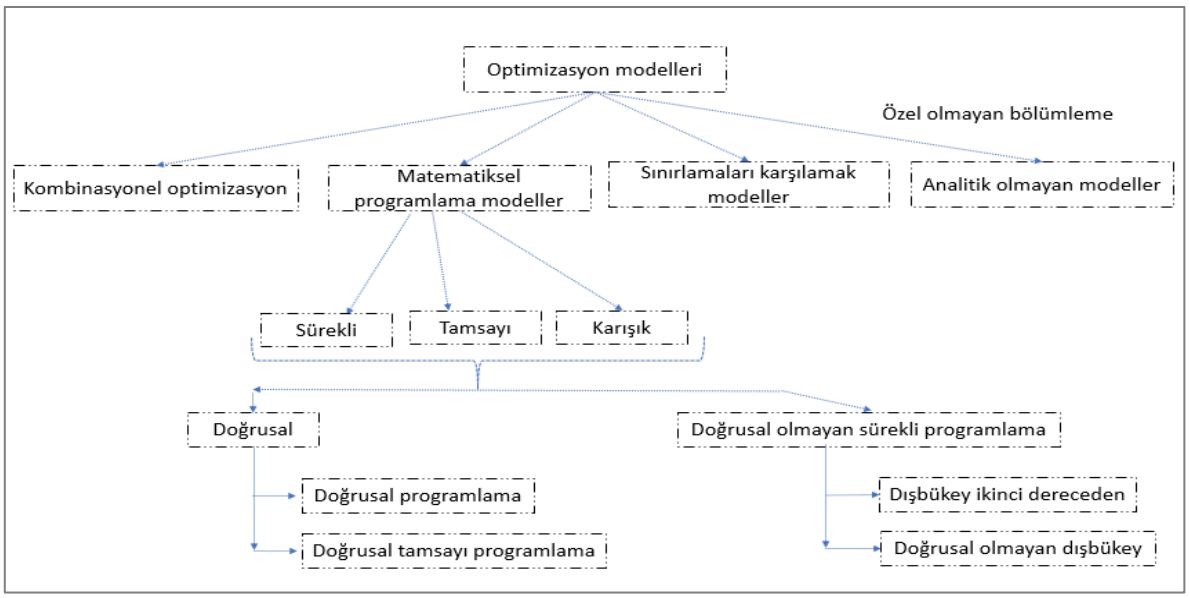
- Simülasyon: Bir optimizasyon probleminin her karar değişkeni, bir objektif fonksiyon değerini tanımlar. Amaç fonksiyonu tahmin edebilecek haline getirilmesi gereken durum değişkenlerinin değerlendirme süreci ve herhangi bir karar değişkeniyle kısıtlamalar simülasyon olarak bilinir. Simülasyon modeli, karar değişkenlerini girdi olarak alır ve sistemin durum değişkenlerini simüle eder. Bazen simülasyon modeli bir veya daha fazla basit matematiksel fonksiyon ve denklemlerden oluşur. Bununla birlikte, çoğu gerçek dünya ve mühendislik problemlerde, fiziksel süreçlere

yaklaştırılan denklem sistemleri ve çeşitli formülleri çözen karmaşık prosedürler simülasyon modelleri gerekir. Simülasyon, bu nedenle, gerçek dünya süreci ya da sistemin zaman işleyişinin hesaplama taklididir[6].

- Weierstrass Teoremi: A bir R^n alt kümesi olduğu varsayalım ve f:A→R, A üzerinde sürekli fonksiyon olduğunu varsayalım o halde x_m ve x_M noktaları A'nın içinde $f(x_m)\leq f(x)\leq f(x_M)$ ve $(x\in A)$ bulunmaktadır. x_m Bölgesel minimumu ve x_M küresel minimumu olarak isimlendirir.
- Çözümün tekliği: A boş olmayan, kapalı ve dışbükey R^n'nın alt kümesi varsayalım. f:A→R, A üzerinde sürekli fonksiyon ve tam yarı içbükey (ya konveks dışbükey) olsa o halde $x^*\in A$, bu şart ile $x\neq x^*$ A'nın için de $f(x^*)>f(x)$ (ya da $f(x^*)<f(x)$) ve $(x\in A)$ bulunmaktadır. x^* Tekil küresel minimum isimlendirir. Daha fazla detaylar için [7-9] bulunmaktadır.

1.5. Modellerin Tanıtımı

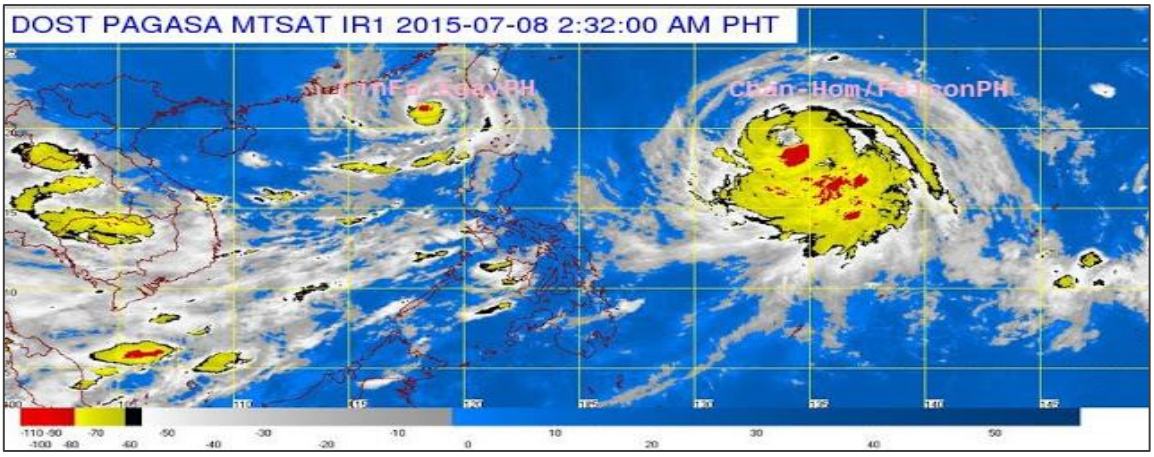
Optimizasyon modellerin farklı aileleri formüle etmesinde ve karar verme problemlerinde çözmesinde pratikte kullanılmaktadır. En başarılı modeller, matematiksel ve sınırlama programlamasına dayanmaktadır. Aşağıda yer alan Şekil 1.7'da farklı modelleri göstermektedir.



Şekil 1.7. Klasik optimizasyon modeller [2].

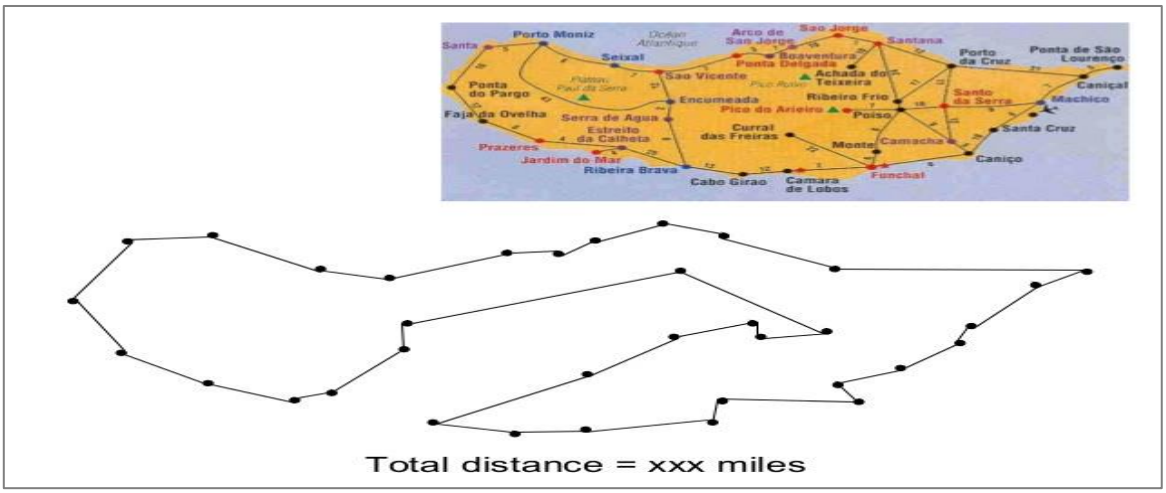
Matematiksel programlama modellerde yer alan çeşitli veri tipler bulunmaktadır. Bunlardan kısaca bahsedecek olursak optimizasyon algoritmaları açısından sürekli optimizasyon teorileri, ayrık(tamsayı) optimizasyondan daha gelişmiştir. Bununla birlikte, ayrık karar değişkenler ile modellenmesi gereken birçok gerçek hayat uygulaması bulunmaktadır. Sürekli modeller bu problemler için uygun değildir. Aslında, birçok pratik optimizasyon probleminde, kaynaklar bölünmez (makineler, insanlar vb.). Bir tam sayı program (IP) optimizasyon modelinde, karar değişkenleri ayrıktır. Bu problemlerin daha genel bir sınıfı, Kombinatorik optimizasyon problemleridir. Bu sınıf ayrık karar değişkenleri ve sonlu bir arama ile nitelendirilmiştir. Bununla birlikte amaç fonksiyonu ve sınırlamaları herhangi bir farklı modellemede bulunmaktadır. Kombinatorik optimizasyon problemlerinin popülaritesi, birçok gerçek dünya probleminde amaç fonksiyonu ve sınırlamaları (doğrusal olmayan, analitik olmayan, kara kutu, vb.) niteliğinde farklıdır, oysa araştırma alanları sonlu olarak tanımlanmıştır.

- Sürekli Programlama Optimizasyon Modeli: Sürekli optimizasyonda, değişkenler modelde gerçek sayıları almasına izin verilir.
- Örnek: Sürekli optimizasyon problemleri genelde şu algoritmalar ile çözülmektedir: problemin bir çözümüne yakınsayan ve tekrarlanarak tanımlanan değişkenlerin değer dizisi üretilmesidir. Bir şirket, kaynakların kullanılabilirliği konusunda verilen sınırlamalar (ekipman, iş gücü, hammadde), üretim maliyetleri ve talep tahmini göz önüne alındığında, kârını en yükseğe çıkarmak istemektedir.
- Örnek: Hava durumu tahmini edebilmek için önce birkaç saat önce atmosferin durumunu belirlemek için bir problemi çözmeliyiz. Bu problem, çeşitli yer ve zamanlarda alınan en uygun son meteorolojik gözlemler (sıcaklık, rüzgâr hızı, nem vb.) durumların bulunmasıyla yapılır. Modelde, atmosferin evrimini, atmosferin ön bilgi durumunu tanımlayan istatistiksel öğeleri ve atmosferin durumu ile gözlemler arasındaki tutarlılığı ölçen bir objeyi(amacı) tanımlayan diferansiyel denklemler içerir. (Şekil 1.8'de Örnek olarak PAGASA uydudan görünen hava durumu).



Şekil 1.8. Hava Durumun Tahmini [10].

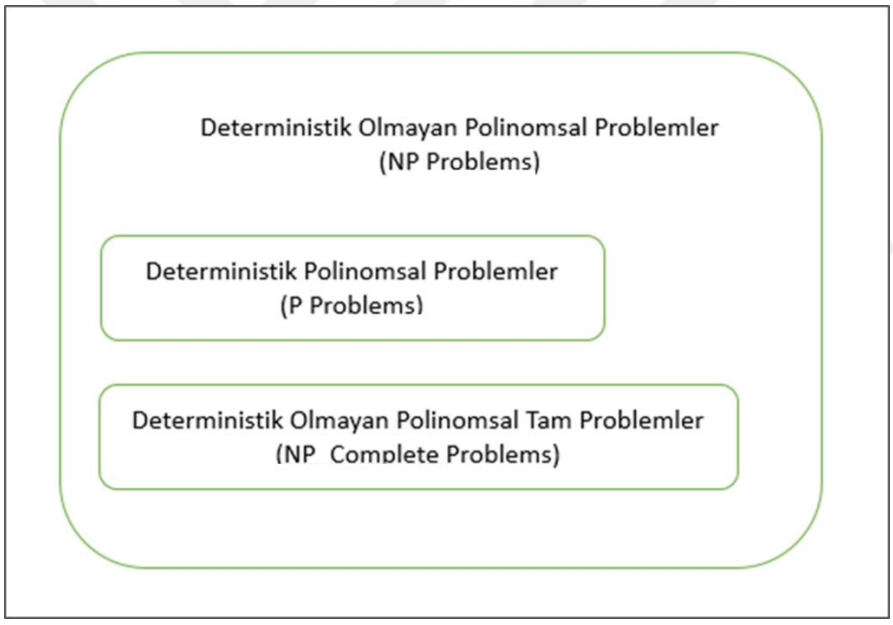
- Tamsayı Programlama Optimizasyon Modeli: Tamsayı programlama karar değişkenleri ayrık ailesindedir. Tamsayı programlama problemlerinin daha genel bir sınıfı, Kombinatorik optimizasyon problemleridir. Belki de en popüler kombinatorik optimizasyon problemi seyahat satıcı problemidir (TSP). Aşağıdaki şekilde formüle edilebilir. n Şehir ve mesafe matrisi $d_{n,n}$ farz edersek, burada her eleman $d_{i,j}$ de i,j şehirler arasındaki mesafeyi temsil eder ve amaç toplam mesafeyi en aza indiren bir tur bulmasıdır. Bir tur ile her şehri tam bir kez ziyaret eder. Hamilton döngüsü (Şekil 1.9'da). Arama uzayının büyüklüğü $n!$.Şehir sayısı ile ilgili çözüm sayısının kombinatorik patlamasını göstermektedir. Maalesef, mümkün olan tüm çözümleri kapsamlı bir şekilde orta ve büyük örnekler için numaralandırmak pratikli değildir.



Şekil 1.9. Gezgin Satıcı Problemi [11].

1.6. Karmaşıklık Teorisi

Problemleri çözmenin ne kadar zor olduğunu gösteren sınıflandırma teorisidir. Problemi çözmesi gereken adımların sayısı, problemin büyüklüğünün gücü ile sınırlı ise P-problemi (deterministik çokterimli zamanlı) sınıfına atanır. Belirleyici olmayan çözüm içeren ve çözümün adımlarının sayısını onaylamak için problemin büyüklüğü gücü ile sınırlı ise, bu şekil problemler NP-problemi (deterministik olmayan çokterimli zamanlı) sınıfına atanır. P-problemleri sınıfı, NP-problemlerin sınıfının bir alt dalıdır, ancak NP de olmayan problemler mevcuttur. NP sınıfına içeren problemlerin bazıları o kadar zordur ki bunlara çokterimli zaman bulmak çok sayıda çaba göstermekten sonuçsuz kalmıştır. Bu sınıfa NP-Complete adı verilir. Bir problemin NP-Complete (NPC) deterministik olmayan çokterimli tam sınıfında atınmışsa o problem için bugüne dek çokterimli zaman bulunmamış demektir [12]. (Şekil 1.10)



Şekil 1.10. Karmaşıklık problemlerin sınıfları

1.7. Optimizasyon Metotları

1.7.1. Klasik Metotlar

Klasik optimizasyon yöntemleri, sürekli ve türevlenebilir fonksiyonlarda optimum çözüm bulunmasında yararlıdır. Bu yöntemler analitik bir yöntemdir ve problemlerde optimum

noktaların bulunmasında diferansiyel denklem teknikler ile kullanılmaktadır. Bazı pratik problemlerde, sürekli ve / veya türevlenemeyen objektif fonksiyonları, klasik optimizasyon teknikleri, pratik uygulamalarda sınırlı kapsamlara sahiptir. Klasik metotlar analitik metotla ve çözüm yapısına dayalı metotlar sınıflanabilir.

1.7.1.1. Problem Açıklaması

Eğer Arama yönü gradyan iniş yönü ile seçilirse, $-\nabla f(x)$ karşılık gelen iteratif arama, gradyan iniş yöntemi olarak da adlandırılır (ayrıca en dik iniş veya Cauchy yöntemi olarak bilinir). Yüksek boyutlu bir araştırma uzayı olan optimizasyon probleminde, klasik algoritmalar optimum çözüme yol açamaz, çünkü araştırma uzayı, problemin büyüklüğüne göre katlanarak artar. Bu yüzden sezgisel ve modern sezgisel algoritmalara ihtiyaç duyulmuştur.

1.7.1.2. Gradyan İniş Metodu

Gradyan iniş yöntemi, bir fonksiyonun yerel minimum bulmak için kullanılan bir yöntemdir. Çözümün başlangıçtaki tahminiyle başlar ve o noktada fonksiyonun gardiyanını alınır.

- Teorem: A'nın simetrik ve pozitif belirli matris olduğunu, b bir vektör ve J(x) ikinci dereceden fonksiyon $J(x)=\frac{1}{2}x^T Ax-x^Tb$ varsayalım. Sonuçta $A\bar{x}=b \iff J(\bar{x}) < J(x)$ $\forall x \neq \bar{x}$. Bu Teorem, doğrusal bir sistemin çözümünü Ax = b'nin (spd) matris ile, ikinci dereceden J(x) fonksiyonu minimize yapabilir. Bunu için gradyan yöntemleri kullanılabilir. Bu sınıfın en açıklayıcı yöntemi gradyan iniş Yöntemidir.
- Çok değişkenli ikinci dereceden olan fonksiyon için kesin çizgi arama ile en dik iniş Metodu: x_0 Noktadan başlayarak, J(x) değerinin en dik iniş yönünü bulanarak ve J(x) değeri o yönde azalana kadar hareket ediyor. Bu noktada, en dik inişin yeni yönü bulunuyor ve tüm işlem tekrarlanıyor. Bir noktadaki fonksiyonun en dik iniş yönü, o noktadaki gradyanın ters yöndedir. Gradyan, verilen noktadan geçen bir kontur çizgisine diktir. (Şekil 1.11)
- Çok değişkenli ikinci dereceden fonksiyon gradyanı:

$J(x)=\frac{1}{2}x^T Ax-x^Tb \quad b=\frac{1}{2} \quad \sum_{i,j=}^{n} a_{i,j}x_ix_j - \sum_{i=1}^{n} b_ix_i$

$\frac{\partial J(x)}{\partial x_k} = \sum_{i=1}^{n} ak_ix_i-b_k x_k \Rightarrow grad(J)= Ax-b$

Y (x) gardiyanının ters yönü, eşittir: r=b-Ax Ax=b (1.6)

Kesin çizgi arama ikinci dereceden fonksiyon için:

$x_0 \in R^n$ noktayı ve $v \in R^n$ Vektörü verildiğini varsayalım. Sonra $x=x_0+\alpha v$, $\alpha \epsilon R^n$

Denklemi x_0 noktasından v yönünde geçen bir çizgiyi temsil eder. Problemdeki J(x) fonksiyonunun minimum değerini çizgi üzerinde bulmak. Yani bulunacak olan minimum fonksiyonu $f(\alpha)\equiv J(x_0+\alpha v)$ ve onun bir gerçel değişkeni a

$$f(\alpha)=J(x_0+\alpha v)=\frac{1}{2}J(x_0+\alpha v)^T A(x_0+\alpha v)-J(x_0+\alpha v)^T b$$

$$=\frac{1}{2}\left[x_0^T Ax_0+ \alpha x_0^T Av+\alpha v^T Ax_0+a^2 v^T Av\right]-x_0^T b-av^T b$$

$$=\frac{1}{2}\left[x_0^T Ax_0+2av^T Ax_0+a^2 v^T Av\right]-x_0^T b-av^T b \quad (1.7)$$

A'nın simetrisinden dolayı son eşitlik bu şekil olacak.

$$\frac{\partial f(a)}{\partial a}=v^T Ax_0+av^T \quad Av-v^T \quad b=0 \quad \Leftrightarrow a=\frac{v^T(b-Ax_0)}{v^T Av} \quad (1.8)$$

Gradyan iniş metodunun algoritması:

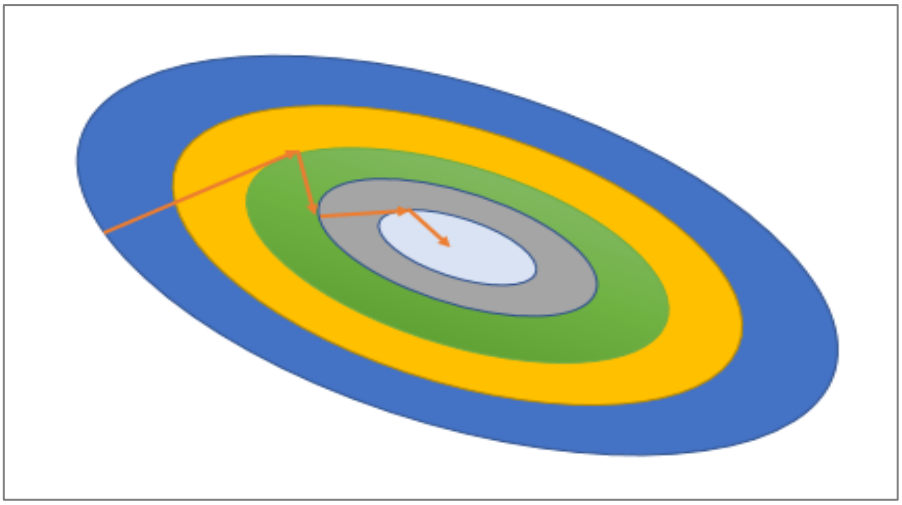
x^0 k=0,1, 2,…

1) $r^{(k)}$=b-A$x^{(k)}$

2) $a_k=\left(r^{(k)}\right)^T r^{(k)} / \left(r^{(k)}\right)^T A\, r^{(k)}$

3) $x^{(k+1)} = x^{(k)} + a_k r^{(k)}$

$\left\|r^{(k)}\right\| < \varepsilon$ (Seçilen küçük bir ε değere kadar).

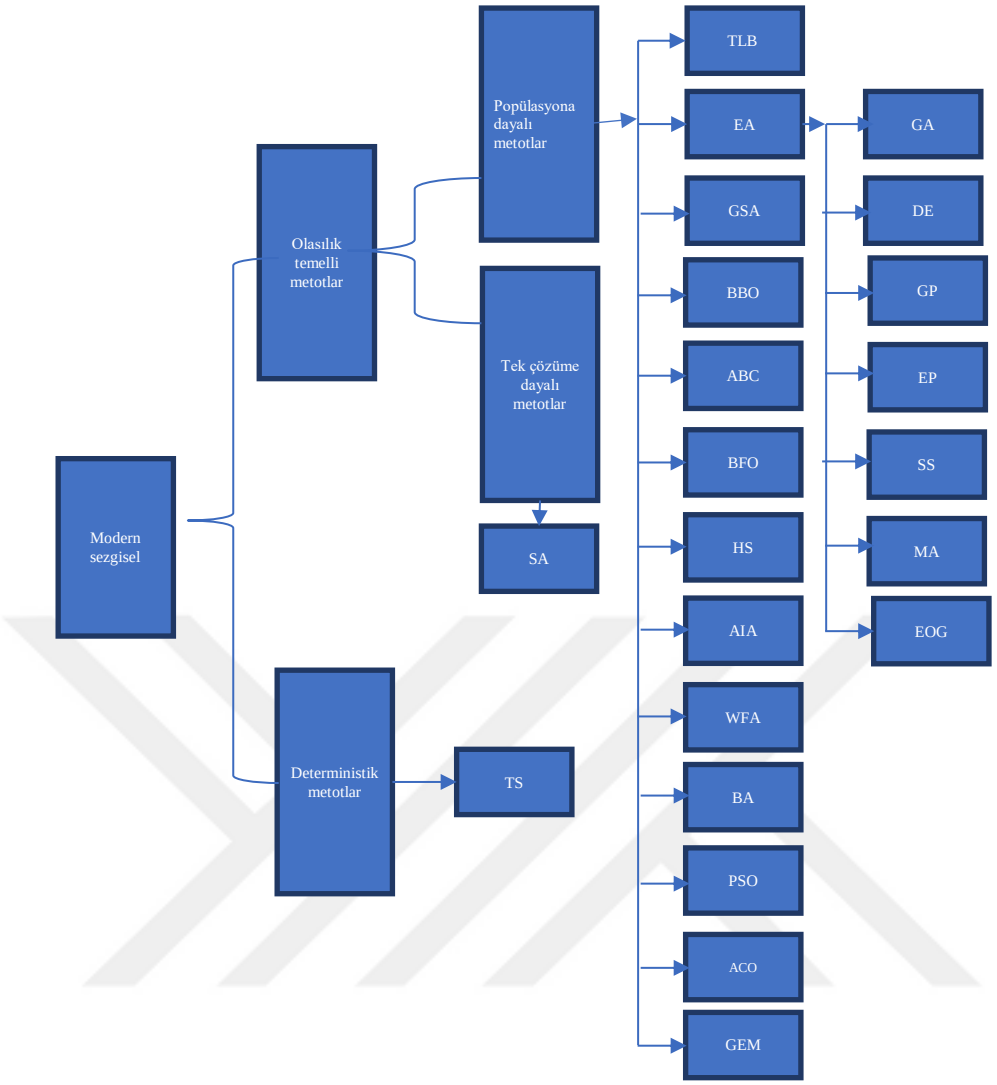


Şekil 1.11. Elips konturun üzerine gradyan metodu hareketi [13].

1.7.2. Modern Sezgisel Algoritmalar

Modern optimizasyon yöntemleri, bazen geleneksel olmayan optimizasyon yöntemleri olarak da adlandırılır ve son yıllarda karmaşık mühendislik optimizasyon problemlerini çözmek için güçlü ve popüler yöntemler olarak ortaya çıkmıştır. Bu yöntemler, genetik algoritmalar, benzetimli tavlama, Tabu arama, parçacık sürü optimizasyon, karınca koloni optimizasyon, sinir ağı tabanlı optimizasyonu ve bulanık optimizasyonu içermektedir. Genetik algoritmalar ise, doğal genetik ve doğal seleksiyon mekaniğine dayalı optimizasyon algoritmalarıdır.

Tabu arama, yörünge tabanlı sınıfa ait olan matematiksel bir optimizasyon metodudur, Aşağıdaki görüldüğü gibi deterministik grup atanmıştır. Tabu arama [14], ziyaret edilen çözümleri bir bellek yapıdan kullanarak bölgesel arama yönteminin performansını arttırır. Potansiyelli olan bir çözüm belirlendikten sonra, "tabu" (aynı sözcüğün farklı bir yazım türü olan "tabu" olarak işaretlenir) böylece algoritma gelecek aşamada bu çözümü ziyaret etmez. Benzetilmiş tavlama algoritması (SA), belli bir fonksiyonun küresel optimumuna iyi bir yakınsama bulunmaktadır ve geniş arama uzayında küresel optimizasyon problemi için genel bir olasılık metottur [15]. Tez kapsamında kısaca popülasyon çözümüne temelli metotlara odaklanılmıştır. Şekil 1.12'de Mevcut olan Modern sezgisel algoritmaları göstermektedir.



Şekil 1.12. Modern sezgisel metotları [16].

1.8. Evrimsel Algoritmalar

Evrimsel algoritmalar bir araç olarak aramada, optimizasyonda, makine öğrenmede ve tasarlanmış problem çözülmelerinde araştırmacıların arasında popular olmuştur. Bu algoritma gelişimi simülasyon yaparak karmaşık problemleri çözülmektedir. Bu kapsamda çok çeşitli algoritmalar bulanmaktadır. Tarihsel açıdan bakarsak genetik algoritmayı ve gelişim stratejisi iki temel olarak tanımlanmaktadır. Genetik algoritmayı Amerika'da olan John Holland ve öğrencisi tarafından geliştirilmiştir[17, 18]. Seleksiyon, reprodüksiyon ve mutasyon operatörler esas olarak yoğunlukla vurgulanmış ve bunları kalıtımsal yapı üzerine kod yaparak problemlerde en iyi maliyet değeri bulmaktadır. Burada reprodüksiyon operatörü mutasyon operatöre göre daha önemli rol oynamaktadır. Evrimsel stratejisi ise Almanyalı Ingo

Rechenberg ve öğrencisi Hans-Paul Schwefel tarafından geliştirilmiştir[17, 18]. Bu strateji mutasyon operatörü reprodüksiyon operatörüne göre daha önemli rol oynamaktadır. Genetik algoritması çok yerlerde örnek olarak arama, optimizasyon, makine öğrenme ve tasarımda kullanılmaktadır[17, 18]. Ama evrimsel stratejisi sadece optimizasyona odaklanmıştır [19, 20].

1.9. Evrimsel ve Modern Sezgisel Algoritmaların Tanımlaması

Modern/Üst sezgisel ve evrimsel algoritmalar problemden bağımsız tekniklerdir ve çok çeşitli problemlere uygulanabilir. Bir "algoritma", bir problemi çözmek için gerçekleştirilen bir dizi işlemi ifade eder. Algoritma, iteratif işlemli ulaşır ya da adımları tanımlanan bir yakınsama kriterine ulaşmasından oluşur. Her adım, basit işlemler açısından daha rafine detaylara kadar rafine edilebilir. Şekil 1.12'de Mevcut olan Modern sezgisel algoritmaları göstermektedir.

Üst sezgisel ve evrimsel algoritmalar başlangıç durumdan ve ilk veriden başlar. Bu algoritmaların amacı, optimizasyon problemlere karar değişkenleri uygun değer bulmaktır ve böylece objektif fonksiyon optimize edilmiştir. Her ne kadar üst sezgisel ve evrimsel arasında farklılıklar olsa da algoritmalar, hepsi başlangıç verilerini ve başlangıç durumu, iteratif, son durumu, karar değişkenleri, durum değişkenleri, simülasyon modeli, sınırlamalar, objektif fonksiyon ve uygunluk fonksiyonu gerekmektedir[6].

1.9.1. Başlangıç Durumu

Her bir üst sezgisel ve evrimsel algoritma, değişkenlerin başlangıç durumundan başlar. Bu başlangıç durumu önceden tanımlanmış, rastgele üretilmiş veya deterministik olarak formüllerden hesaplanır.

1.9.2. Döngü

Algoritmalar, bir çözüm aramayı için işlemleri tekrarlayarak gerçekleştirirler. Evrimsel ve üst sezgisel algoritmalar, döngülerini, optimizasyon problemleri üzerinde bir veya birkaç ilk çözümüyle başlatırlar. Ardından, yeni çözümler üretmek için sıralı işlemler gerçekleştirilir.

Yeni bir olumsal çözüm üretildiğinde bir tekrar adımı sona erer. Yeni üretilen çözümler, algoritmanın bir sonraki iterasyon için ilk çözümler olarak dikkate alınabilir.

1.9.3. Son Durumu

Seçilen sonlandırma kriterlerin karşıladıktan sonra, algoritma durur ve problemin en iyi veya nihai üretilmiş çözümlerini rapor eder.

Sonlandırma kriterleri çeşitli farklı şekillerde tanımlanmıştır:

(1) Tekrarlama sayısı,

(2) Ardışık iterasyon arasındaki çözüm değerinin iyileştirme eşiği

(3) Optimizasyon algoritmalarının çalışma süresi

1.9.4. Başlangıç veriler (Bilgi)

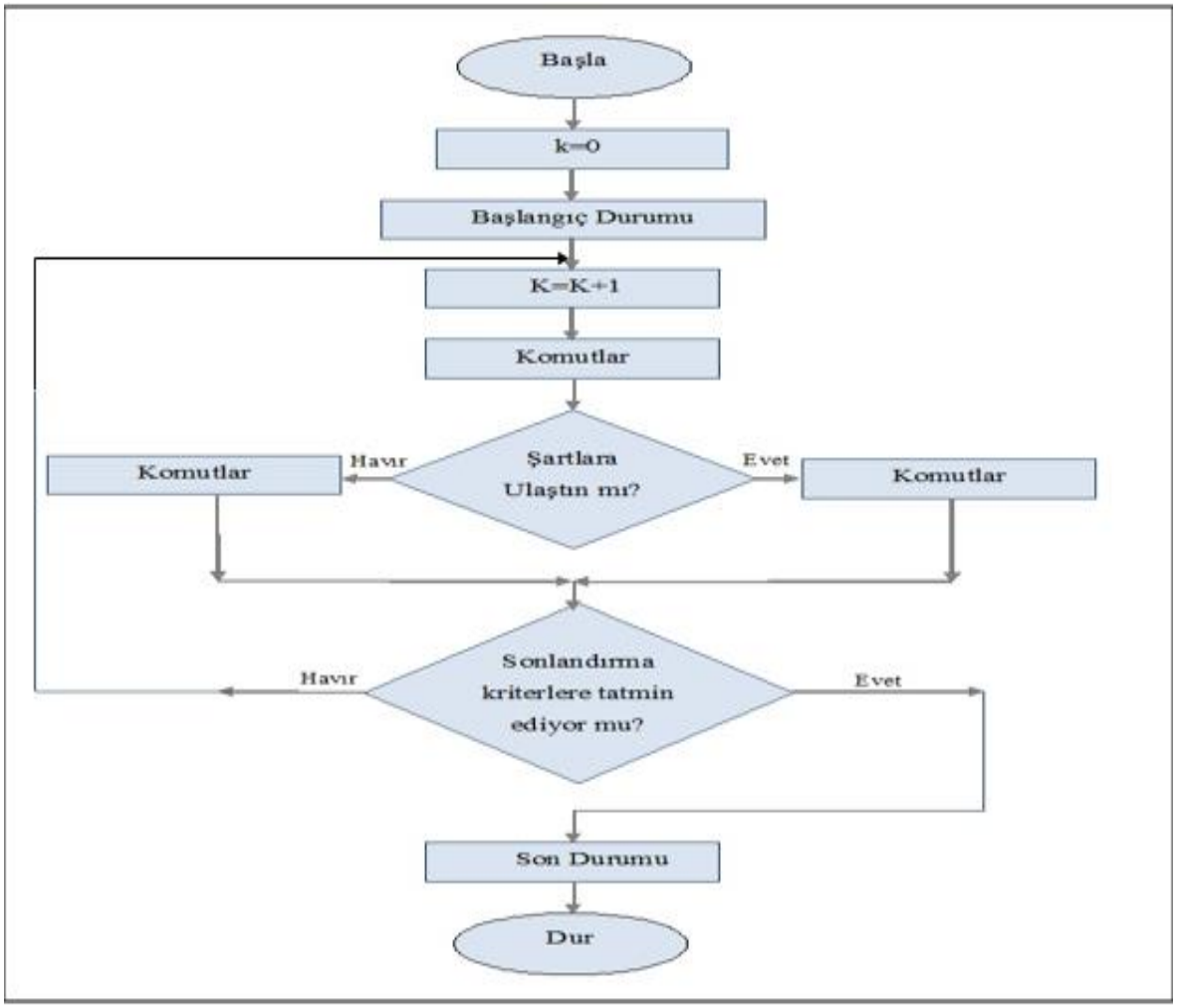
Başlangıç bilgileri, iki kategoriye ayrılır, simülasyon için gerekli olan optimizasyon problemi tanıtan veriler, çalıştırması ve kalibre edilmesi gereken algoritmanın parametreleridir**.**

2. YAPILAN ÇALIŞMALAR

2.1. Evrimsel, Sezgisel ve Modern Sezgisel Algoritmaların Tartışılması

Sezgisel algoritmalar, probleme bağlı tekniklerdir. Bu nedenle, probleme genellikle uyarlar ve bu problemin özelliklerinden tam olarak yararlanmaya çalışırlar. Bununla birlikte, genellikle çok açgözlü oldukları için, yerel bir optimumda tıkınabilir, bu nedenle optimum çözümü elde etmekte başarısız olurlar. Diğer yandan, modern sezgisel algoritmalar, problemden bağımsız tekniklerdir. Aslında, problemin özgüllüğünden faydalanmazlar, bu nedenle kara kutular olarak kullanılabilirler. Genel olarak açgözlü değiller. Aslında, çözümü geçici olarak bozulmasını kabul ederek çözüm uzayda daha kapsamlı bir şekilde keşfetmelerine imkânı sağlar.

Sezgisel algoritmalar probleme özgüdür. Bu kısıtlılığın üstesinden gelmek için modern sezgisel tasarlanmıştır ve aynı zamanda evrimsel algoritmalar bu tür modern sezgisel algoritmalara aittir. Evrimsel algoritmalar olasılığa dayalı rasgele değişkenler kullanarak iyi sonuçlar göstermektedir, yani algoritmalar her bir çalışmada farklı bir çözüm üreteceği anlamına gelmektedir (deterministik değildir). Şekil 2.1'de basit bir algoritmanın genel şeması gösterilmiştir.



Şekil 2.1. Basit bir algoritmanın genel şeması [6].

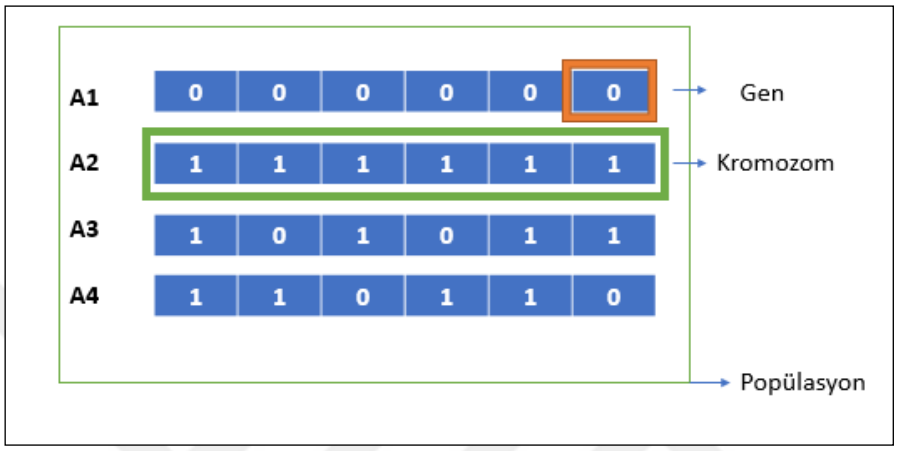
2.1.1. Genetik Algoritması

Genetik algoritma popülasyona dayalı, modern sezgisel uyarlı yöntemdir. Genetik algoritma 1970'lerin başında John Holland tarafından önerilmiştir [17, 18]. Doğal seleksiyon ve genetik mekaniğine dayanan Olasılıksal bir arama algoritmasıdır. Doğal seleksiyon süreci, bir popülasyondan en uygun bireylerin seçilmesiyle başlar. Ebeveynlerin özelliklerini miras alan yavrular üretirler ve bir sonraki kuşağa eklenirler. Yavrular ebeveynlerinden daha iyi uygunluk miktarı varsa hayatta kalmasına bir şansı daha olacaktır. Bu süreç tekrarlamaya devam ederek en sonunda, en uygun bireylere sahip bir nesil bulmaktadır [18]. Genetik algoritmada dört aşama dikkate alınır.

- Başlangıç popülasyonu

  Algoritma sürecinde popülasyon, bir dizi bireyler ile başlar. Her bireyin çözmek

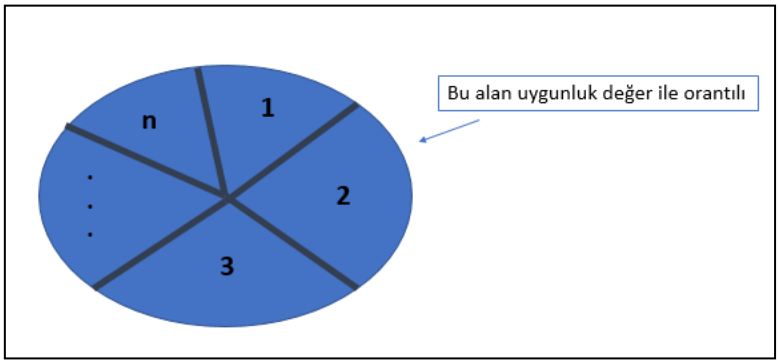
istediğiniz problem için bir çözümdür. Her bireyi, Genler olarak bilinen bir dizi parametre (değişken) tarafından nitelenmiştir. Genler, bir kromozomu (çözümü) oluşturmak için bir dizide birleştirilir. Genetik algoritmada, bir bireyin genleri alfabe cinsinden bir dizi kullanılarak gösterilir. Genellikle, ikili değerler kullanılır (1 ve 0 dizisi) yani genleri bir kromozomda kodladığımızı söylüyoruz. (Şekil 2.2)



Şekil 2.2. Popülasyon tanıtımı

- Seçilme

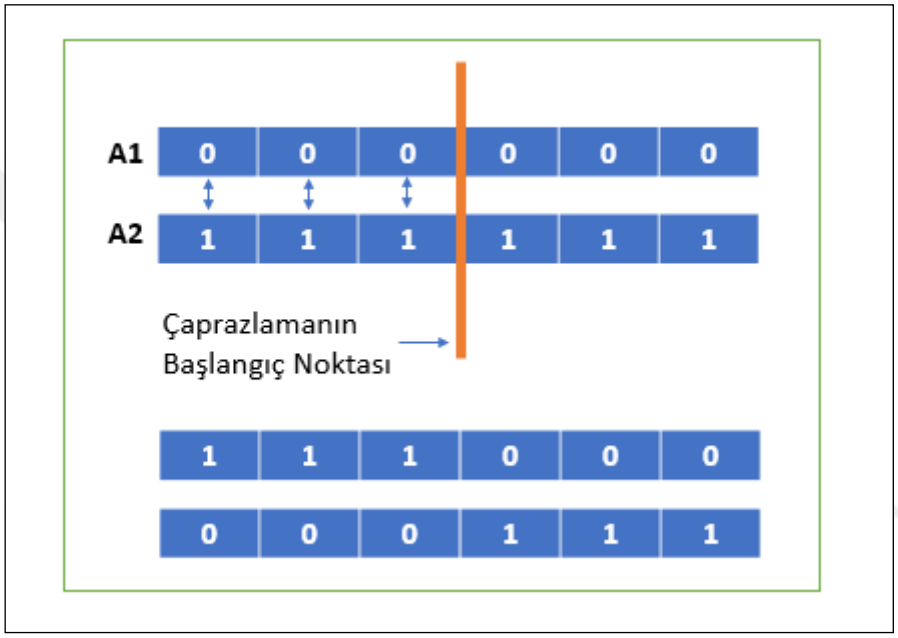
  Seçim aşamada, en uygun bireyleri seçmek ve genlerini bir sonraki kuşağa aktarmaktır. Bireylerin (ebeveyn), kendi uygunluk değerlerine göre seçilir. Yüksek değeri olan bireylerin röprodüksiyon için seçilme şansı daha fazladır ya da rulet tekerleği yöntemi ile uygunluk orantılı seçim yapılabilir. (Şekil 2.3)



Şekil 2.3. Seçilme şeması

- Çaprazlama

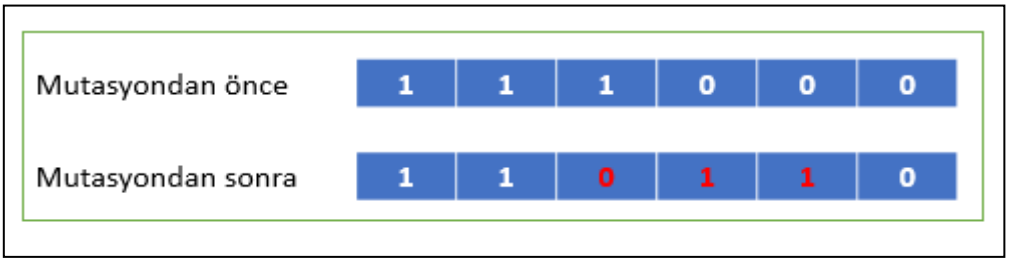
Çaprazlama, genetik bir algoritmada en önemli aşamadır. Eşleştirilecek olan çift Ebeveynlerin genlerinin arasında rasgele seçilen bir Çaprazlama başlangıç noktası vardır. Yavrular, çaprazlama noktasına gelene kadar ebeveynlerin genlerini kendi aralarında değiştirerek yeni yavrular üretmektedir. Yeni üretilen yavrular bir sonraki aşamada popülasyona eklenir. (Şekil 2.4)



Şekil 2.4. Çaprazlama şeması

- Mutasyon

Üretilen yavrular, bazı genleri düşük olasılıklı bir mutasyona tabi tutulabilir yani bazı bitler dizide değerleri ters çevrilebilir. (Şekil 2.5)

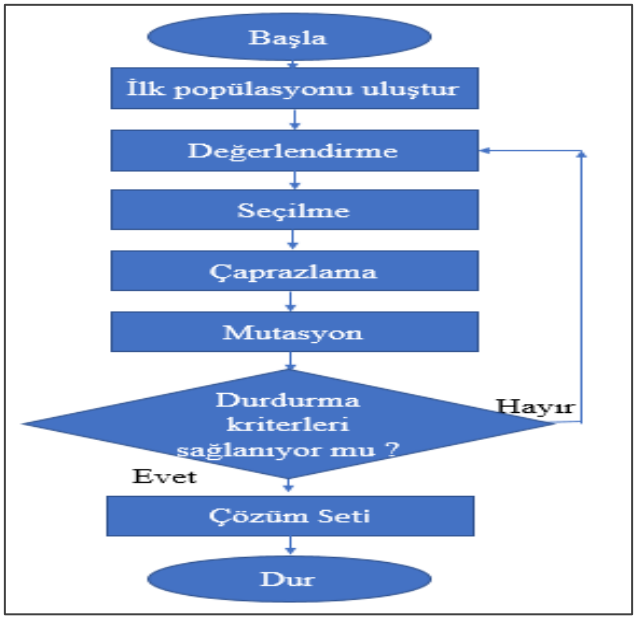


Şekil 2.5. Mutasyon şeması

Newton veya gradyan iniş gibi geleneksel sürekli optimizasyon yöntemleriyle karşılaştırıldığında, aşağıdaki önemli farklılıkları açıklanmıştır. GA, kodlanmış problemin parametrelerinin versiyonlarını kendi parametrelerin yerine kullanılmaktadır.

1. Tüm geleneksel yöntemler tek bir noktadan ararken, GA algoritması her zaman bütün popülasyonun noktalarının (dizeler) üzerinde çalışırlar ve bu genetik algoritmaların sağlamlığına çok şey katmaktadır. Küresel optimuma ulaşma şansını geliştirir ve ters durumda, yöresel optimumda tıkanma riskini azaltır.
2. Normal genetik algoritmalar problem hakkında herhangi bir yardımcı bilgi kullanmazlar.
   (türev gibi objektif fonksiyon değeri). Bu nedenle, herhangi bir sürekli veya ayrık optimizasyon problemlilerin üzerinde uygulanabilirler ve sadece yapılması gereken tek şey şudur (anlamlı bir kod çözme işlevi belirtmeleri).

Genetik algoritma, olasılıklı operatörleri kullanırken, geleneksel yöntemler Sürekli optimizasyon deterministik operatörlerini uygular. Daha spesifik olarak, gerçek nesilden yeni jenerasyonun hesaplanma şekli bazı rastgele bileşenlere sahiptir. (daha sonra bazı rasgele bileşenlerin neye benzediğini bazı örneklerle inceleyeceğiz) [21]. Şekil 2.6'da Genetik algoritmanın akış diyagramı göstermektedir.



Şekil 2.6. Genetik algoritmanın akış diyagramı

2.2. Sürü Zekâsı

Etkileşime giren çoklu ajanlar ile sürü zekâsı, oldukça genel bir kavramdır ve basit kurallara uyarak bilgi alışverişinde bulunmaktadır. Şaşırtıcı bir şekilde, bu kadar basit karmaşık sistemlerde, kendi kendini organize eden davranış da göstermektedir. Ajan etkileşimlerinin özellikleri doğadaki farklı ilham kaynaklarından elde edebilmesidir. Yazılacak algoritma prosedürü oldukça pratikte basit, esnek ve verimli olabilir. Sürü zekâsı, evrimsel hesaplama paradigmasının bir parçasıdır [22-25]. Tüm evrimsel algoritmaların arasında, sürü zekasına dayalı algoritmalar problemin esnek olarak çözülmesine hâkimdirler. Bu hâkimiyetin çok nedeni vardır ve üç bariz nedenler şunlardır:

(1) sürü zekâsı, birden fazla ajan topluğunu yani popülasyona kullanır, bu nedenle doğal sistemleri taklit etmek için iyi yolları sağlar.

(2) popülasyona dayalı yaklaşımlar, uygulamada paralellik ve vektörellik izin vermekte, bu nedenle uygulamada açıktır.

(3) sürü zekâsı tabanlı algoritmalar basit ve uygulanması kolay ve esnektir yine de yeterince etkindir. Sonuç olarak, bu algoritmalar, uygulamalarda nispeten geniş bir yelpazede problemler ile başa çıkabilirler.

2.2.1. Sürü Zekanın Genel Özellikleri

Bonabeau[26] tarafından Sürü Zekası şu şekilde tanımlanmış: toplumsal böcek kolonileri ve diğer hayvan topluluklarının ortak davranışlarından esinlenen, algoritmalar veya dağıtılmış problem çözme cihazları tasarlamasına çabasıdır.

- Sürün terimi, genel olarak etkileşim içindeki olan ajanlar veya bireyler topluluğudur.
- İki temel kavramlar kendi kendine organize olabilmek ve iş bölümü, sürünün akıllı davranış elde etmesi için gerekli ve yeterli özellikleridir.

Öz-örgütlenme dört özelliği bulunmaktadır:

1- Olumlu geribildirim: uygun yapıların oluşturulmasını teşvik etmesi istihdam ve güçlendirme gibi bazı karınca türlerinde iz bırakma ve takip etme, olumlu geri bildirim örneği olarak gösterilebilir.

2- Negatif geribildirim: pozitif geri beslemeye karşı dengelemek ve ortak örüntü kararlığına yardımcı olmak. Mevcut yiyecek arayan birey açısından oluşabilecek işbanın önüne geçmek için negatif bir geri bildirim mekanizması gereklidir.

3- Dalgalanmalar: rastlantısal yürüme, hatalar, rastgele görev değiştirme bireylerin arasında önemlidir. Rastlantısalım, yeni ortaya çıkan yapılar için çözümlerin keşfedilmesi için olanak tanınması genellikle önemlidir.
4- Çoklu etkileşimler: Sürünün içindeki ajanlar, diğer ajanlardan gelen bilgileri kullanır; böylece bilgiler ağ boyunca yayılır.

BÜ özelliklere ilaveten, iş bölümü olarak adlandırılan eşzamanlı görevleri özel ajanların tarafından yapılsa da önerilen ilkeler bulunmaktadır.

Millonas'a[27] göre, akıllı bir sürü oluşması için aşağıdaki ilkeleri yerine getirmelidir.

1. Sürü, basit uzayda ve zaman hesaplamalarını kolayca yapabilmelidir (yakınlık ilkesi).
2. Sürü, ortamlarda kalite faktörlerine cevap verebilmelidir (kalite ilkesi).
3. Sürü, faaliyetlerini aşırı dar kanallar boyunca gerçekleştirmemelidir (çeşitli tepki ilkesi).
4. Sürü, her dalgalanma üzerine davranış biçimini değiştirmemelidir (istikrar ilkesi).

Sürü, gerektiğinde davranış durumunu değiştirebilmelidir (uyarlana bilirlik ilkesi) [28]

Tez kapsamında, Önerilen algoritma sürü zekasının sınıfına aittir ve bu sınıfta olan algoritmalar özellikle önerilen algoritma ile karşılaştırılmış olanları detaylı şeklinde bu bölümde açıklanmıştır.

2.2.2. Parçacık Sürü Optimizasyonu

Kennedy (bir sosyal psikolog) ve Eberhart'ın (bir elektrik mühendisi) parçacık sürüleri hakkındaki ilk fikirleri hesaplamalı zekâ üretimde tüm bireysel bilişsel yeteneklerden ziyade basit sosyal etkileşimin analoglarını kullanmayı amaçlamışlardır[29-31]. Parçacık kelimesi, örneğin, bir kolonideki arı veya bir sürüdeki kuşu ifade eder. Sürüdeki her birey veya parçacık, kendi zekasını veya grup zekasını kullanarak dağıtılan bir biçimde davranıyor. Bu nedenle eğer bir parçacık gıdaya giden iyi yol bulursa, derhal sürülerin geri kalanı ve hatta sürüdeki uzak olanları da iyi yol izleyebilir. Sürü zekaya dayalı optimizasyon yöntemler, davranışsal ilham kaynağı denir ve ona karşı Genetik algoritma evrimsel algoritmaya dayalı yöntemdir. PSO ve GA aralarındaki farklar GA ayrık problemlerin üzerinde uygundur ama PSO sürekli problemlerin üzerinde daha iyi performans göstermektedir. Çok değişkenli optimizasyon bağlamında, sürünün belirten büyüklüğü, her bir parçacık başlangıçta rasgele konumu ile çok boyutlu tasarım alanında yerleşmektedir. Her parçacığın iki özelliği varsayılır: Birisi konum diğeri ise hızdır. Her parçacık tasarım alanında gezinir ve bulduğu en iyi konumu (gıda kaynağı

veya amaç fonksiyon değeri açısından) hatırlar. Parçacıklar birbirlerine bilgi veya iyi konumlar birbirleriyle haberleşirler ve iyi konumlarda alınan bilgilere dayanarak kendi konumlarını ve hızlarını ayarlamaktadır.

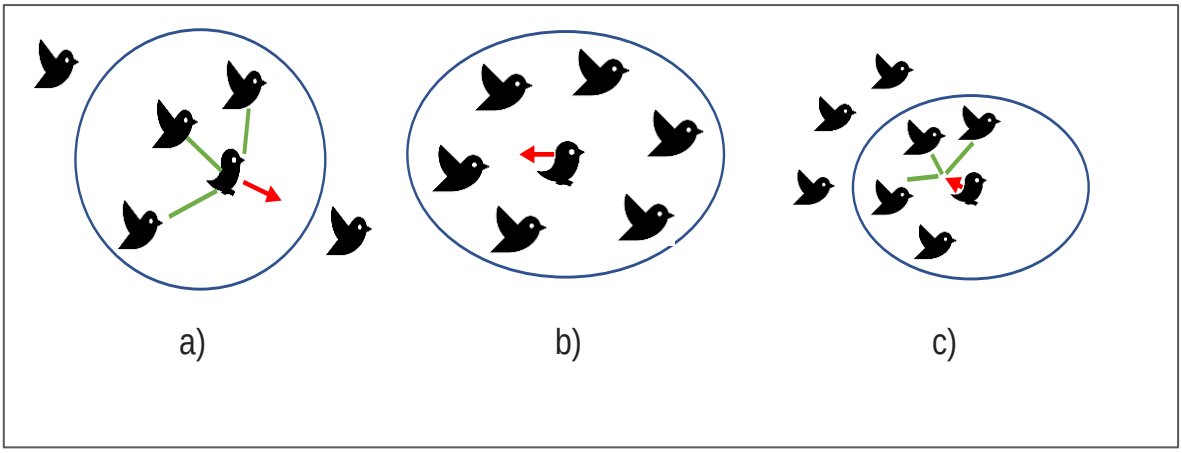
Örnek olarak, bir sürüde kuşların davranışını göz önünde bulundurun (Şekil 2.7). Her kuşun sınırlı bir zekâyı sahip olmasına rağmen, aşağıdaki basit kuralları izler:

1. Diğer kuşlara çok yaklaşmamaya çalışır.
2. Diğer kuşların ortalama yönüne yönelir.
3. Diğer kuşlar arasındaki Geniş boşlukları olmayan "ortalama konumu" bulmaya çalışırlar[31].

Şekil 2.7. Kuşların sürü zekâsı [32].

Böylece sürünün davranışı üç basit kombinasyona dayanmaktadır.
faktörler:

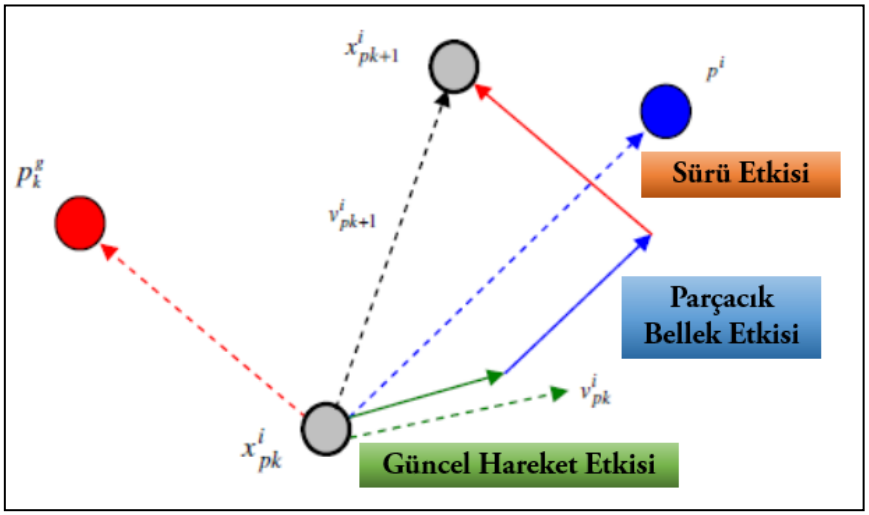
1. (Kohezyon) birlikte yapışma.
2. (Ayırma) çok yaklaşmayın.
3. (Uyuşma) sürünün genel başlığını izleyin.



Şekil 2.8. Sürünün davranış şeması a)ayrılma b)uyuşma c) kohezyon [31].

PSO aşağıdaki temel modeli dayanarak geliştirilmiştir:

1. Bir kuş bir hedefi veya gıdayı (veya amaç fonksiyonun maksimumunu) bulduğu zaman, bilgileri diğer tüm kuşlara derhal iletir.
2. Diğer tüm kuşlar hedefe veya yiyeceğe (amaç fonksiyonun maksimum yönüne) çekilmektedir ve bu çekilmesi işlem sırası doğrudan olmayacaktır.
3. Her kuşun kendi bağımsız düşüncesinin yanı sıra geçmiş hafızasında da bir bileşen vardır. Böylece, model tasarım alanındaki amaç fonksiyonunun maksimum değeri için rastgele bir aramayı taklit eder. Şekil 2.8'de, birçok kez tekrarlamadan sonra, kuşlar hedefe (amaç fonksiyonun maksimum yönüne) gideceklerdir[29-31].



Şekil 2.9. PSO konsept davranışı [33].

Gördüğünüz gibi şekil 2.9'da her parçacık konumu, şimdiki konumu, şimdiki hızı, şimdiki konum ve Pbest'ın arasındaki farkı ve şimdiki konum ve Gbest'ın arasındaki farkına hareket etmektedir. Şekil 2.10, 2.11'de PSO'nun akış diyagramı ve formül tanıtımı sıra ile göstermektedir [34].

Pbest: Parçacıkların en iyi konumu (şimdiki iteratif)

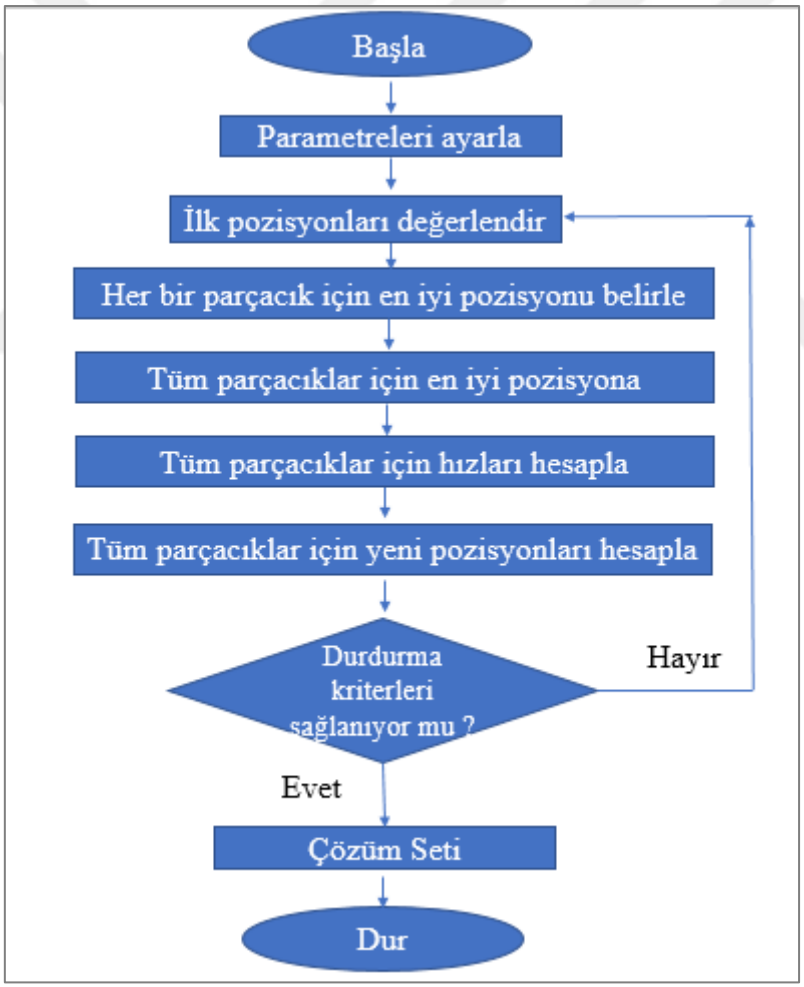
Gbest: Sürünün en iyi konumu (başlangıçtan şu ana kadar gelen iteratif)

P: Parçacığın konumu

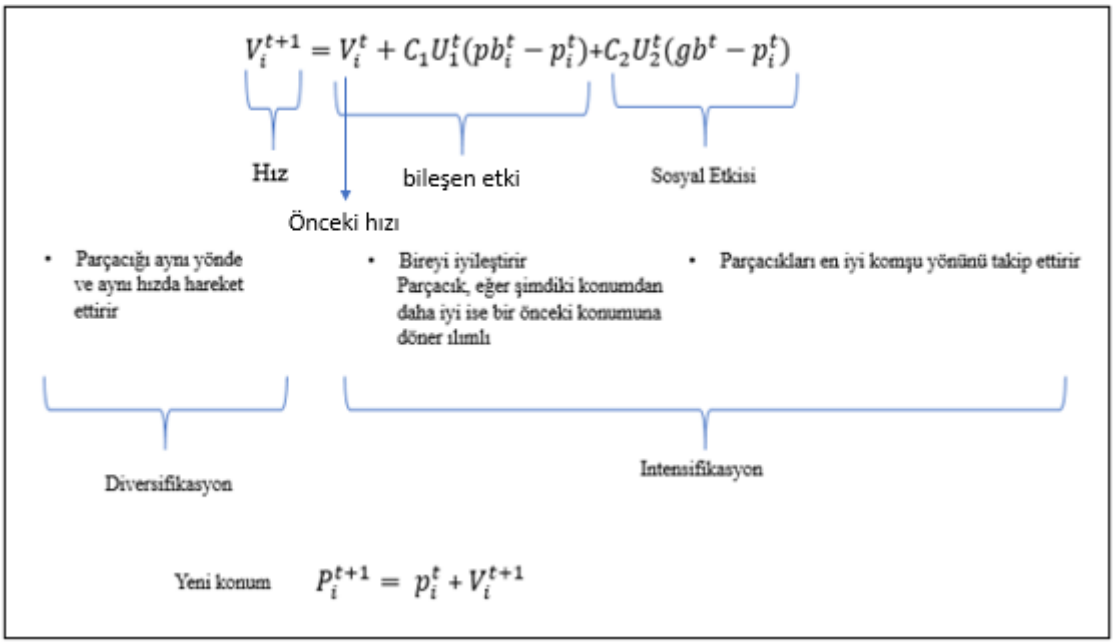
V: Parçacığın hızı

C_1: bölgesel bilginin ağırlığı

C_2: Küresel bilginin ağırlığı



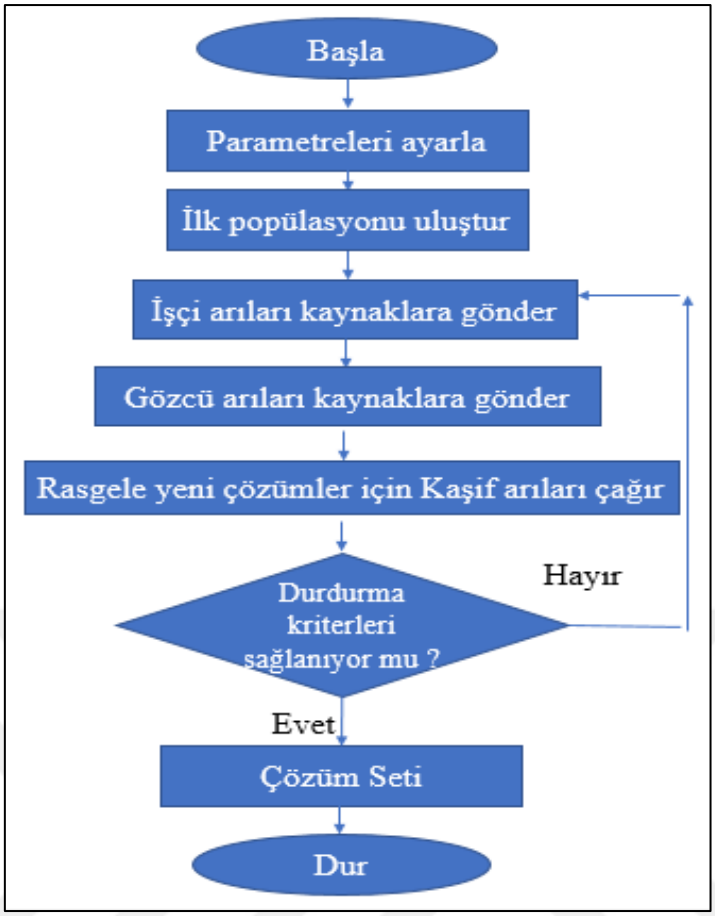
Şekil 2.10. Parçacık sürü optimizasyonun akış diyagramı



Şekil 2.11. Parçacık sürü optimizasyonun formül tanıtımı

2.2.3. Yapay Arı Koloni Algoritması

Bal arı sürülerinin kolektif zekâsı ortaya çıkmasına yol açan yem seçiminin minimal modeli üç temel bileşenden oluşur. Başka bir deyişle bir koloni de üç grup arı mevcuttur: Görevli arılar, gözcü arılar ve kâşif arılarıdır [28]. Önerilen modelde kolinin yarısı Görevli arıya ait ve diğer yarısı ise gözcü arıya aittir. Bir Görevli arı her bir nektar kaynağa işlem yapmaktadır. Açıkçası nektar kaynağın sayısı Görevli arıların sayısına eşittir[35]. ABC optimizasyon metodunun genel algoritma yapısı Şekil 2.12 göstermektedir.

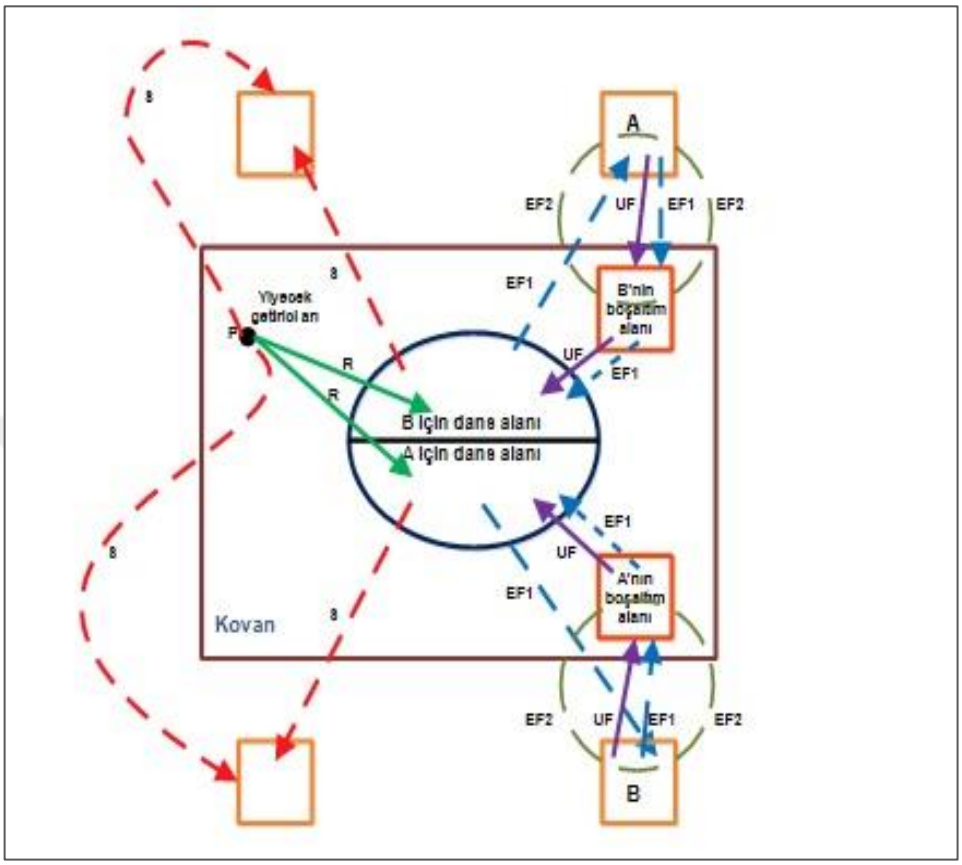


Şekil 2.12 Yapay arı koloni algoritmasının akış diyagramı

Başlatma aşamasında, gıda kaynakların popülasyonu (Çözümler), yapay kâşif arılar tarafından başlatılır ve kontrol parametreleri ayarlanır. İşçi arı fazında yapay olarak çalışan arılar, hafızasında gıda kaynağının komşuluğunda daha fazla nektar bulunan yeni besin kaynakları ararlar. Onlar kaynak komşu bulurlar ve daha sonra onun uygunluğunu değerlendirirler. Yeni gıda kaynağını ürettikten sonra uygunluğu hesaplanır ve ebeveynler ile arasında açgözlü seçilim yapılır. Bundan sonra, Görevli arılar, dans alanında dans ederek kovana bekleyen gözcü arılar ile besin kaynak bilgilerini paylaşırlar. Gözcü arılar aşamasında, yapay çalışmak ile Görevli arıların verdiği bilgiye bağlı olarak olasılıksal yiyecek kaynaklarını seçerler. Bu amaçla, rulet tekerleği seçme yöntemi gibi, uygunluğa dayalı bir seçim tekniği kullanılabilir. Açgözlü arı için bir gıda kaynağı olasılıksal olarak seçildikten sonra, komşu kaynağı belirlenir ve uygunluk değeri hesaplanır. İşçi arıların aşamasında olduğu gibi iki kaynak arasında aç gözlü seçilim yapılır.

Kâşif arıların aşamasında, önceden belirlenmiş döngü sayısına göre çözümleri iyileştirilemeyen Görevli arılar tarafından çözümleri terk etmelerini ve kâşif arılar araya

girmesidir. Özet olarak söylersek şekil 1.25'te gösterildiği üzere arıların davranışlarını model olarak gösterilmiş ve ABC'nin yukarıdaki Akış diyagramı da bu modelden kabaca alınmış.



Şekil 2.13. Yapay arı koloni algoritmasının arıların hareketleri [36].

Şekil 2.13'te gösterildiği üzere kovanın çevresinde olan yeni besin kaynakları, kâşif arılar tarafından rasgele arama ile bulunacaklar. Bulunan besin kaynaklar arıların tarafından nektarı taşıyarak kovana götürmektedirler. Nektar aktarıldıktan sonra üç olasılık bulunacaktır. Bunlar; dans ile aldığı bilgileri paylaşmak, hiç bilgi paylaşmadan geri dönmek, yeniden kâşif arı olarak besin kaynağı terk etmesi ile başlamaktır. Kovanda bulunan gözcü arılar da dansa göre davranışlarını yansıtmaktadır.

ABC algoritmayı 4 ana aşamada formülleri kapsayan, aşağıda açıklanmıştır. İlk aşamadaki başlangıçta besin kaynağı rastgele üretilmesidir. l_i ve u_i amaç fonksiyonun çözüm uzayının alt ve üst sınırı tanıtmaktadır.

$$x_m=l_i+rand(0,1)*(u_i-l_i) \quad (2.1)$$

İkinci aşamada Görevli arının besin kaynağı komşuyu haspalar. Rastgele seçilmiş parametre, x_k rastgele seçilmiş besin kaynağı, $\emptyset_{mi}$ -1,1 arası rastgele sayı

.

$$v_{mi}=x_{mi}+\emptyset_{mi}\ (x_{mi}-x_{ki}) \tag{2.2}$$

Uygunluk değeri aşağıdaki formül ile hesaplanır ondan sonra x_m ve v_m aralarında açgözlü seçilim yapılır.

$$fit_m(x_m)=\frac{1}{1+f_m(x_m)},f_m(x_m)>0 \quad ve \quad fit_m(x_m)=1+|f_m(x_m)|,\ f_m(x_m)<0 \tag{2.3}$$

Üçüncü aşamada besin kaynağın karlık miktarı tüm besin kaynakların karlıklık miktarlarına göre ihtimali hesaplanır. Besin kaynak komşusunu yeniden üretilmesi için Gözcü arı bu aşama da görevini ihtimale dayalı formül 2.4'du kullanmaktadır.

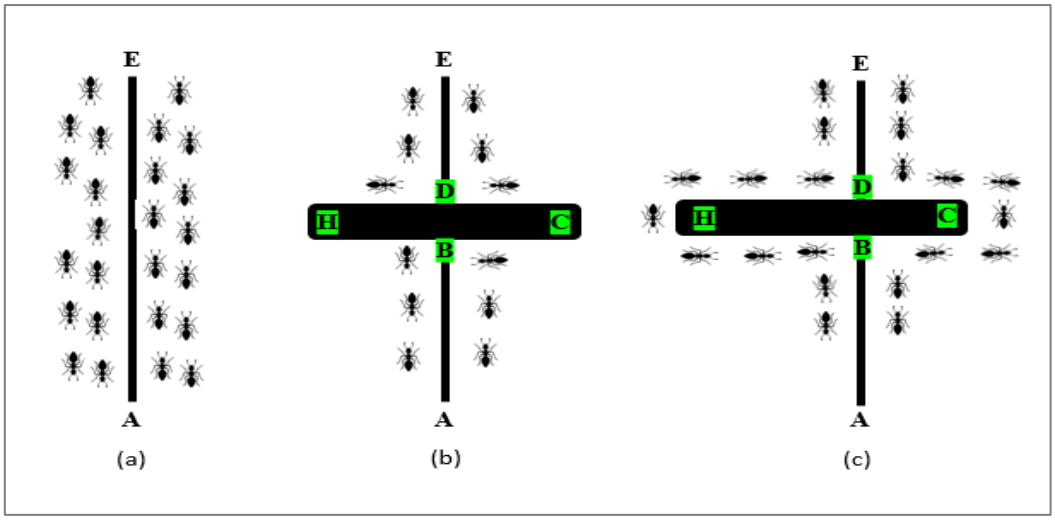
$$P_m=\frac{fit_m(x_m)}{\sum_{m=1}^{SN} fit_m(x_m)} \tag{2.4}$$

Kâşif arılar faydasız besin kaynakları kaldırıp yeni besin kaynağı üretmektedir.

$$x_m = l_i+rand\ (0,1)\ *(u_i - l_i) \tag{2.5}$$

2.2.4. Karınca Koloni Algoritması

Karınca koloni optimizasyonu kombinatorik optimizasyon problemleri için modern sezgisel bir yaklaşımdır[37]. Sosyal böceklerin, özellikle karıncaların arama davranışlarından esinlenilmiştir. Bu algoritma, göz önünde bulundurulan probleme çözüm üretmek için bir (parametremize) olasılık modelinin kullanılması ile nitelenmektedir.



Şekil 2.14. Karıncaların gerçek davranışları a) karıncalar yoldan geçerken b) Karıncaların yolunda bir engel ile karşılaşıyorlar c) karıncalar kısa yolu seçerler

ACO da karıncalar yemek bulma amacıyla ile yuvalardan çıkarak yemek noktalara hareket etmektedirler. Bu yolda kısa yollarını feromon izleri ile bakarak seçilmektedir. Bu işlem zaman sürecinde en fazla feromon miktarın bulunduğu yollarda diğer karıncalar tarafından seçilmektedir. Şekil 2.14'te görüldüğü üzere karıncaların yollarında bir engel ile karşılaşırlar. Önce rastgele iki yoldan birsini seçerler ve zaman gittikçe feromon miktarının fazla olduğu yolu seçerler[37] .

1. Geçiş kuralı

Bir yolun feromon miktarı fazla ise karınca tarafından seçilme ihtimalide miktarı da yüksektir.

- $\tau(i,j)$ *i* ve *j* noktaları arasındaki feromon miktarı.
- $\eta(i,j)$, *i* ve *j* noktaları arasındaki mesafenin tersi ($1/\delta(i,j)$).
- α ve β parametreler .

$$p_k(i,j)=\begin{cases} \dfrac{[\tau(i,j)]^{\alpha}.[\eta(i,j)]^{\beta}}{\sum_1^t [\tau(i,j)]^{\alpha}.[\eta(i,j)]^{\beta}} & j,u\in N_{k,i} \\ 0 & \text{aksı takdirde} \end{cases} \tag{2.6}$$

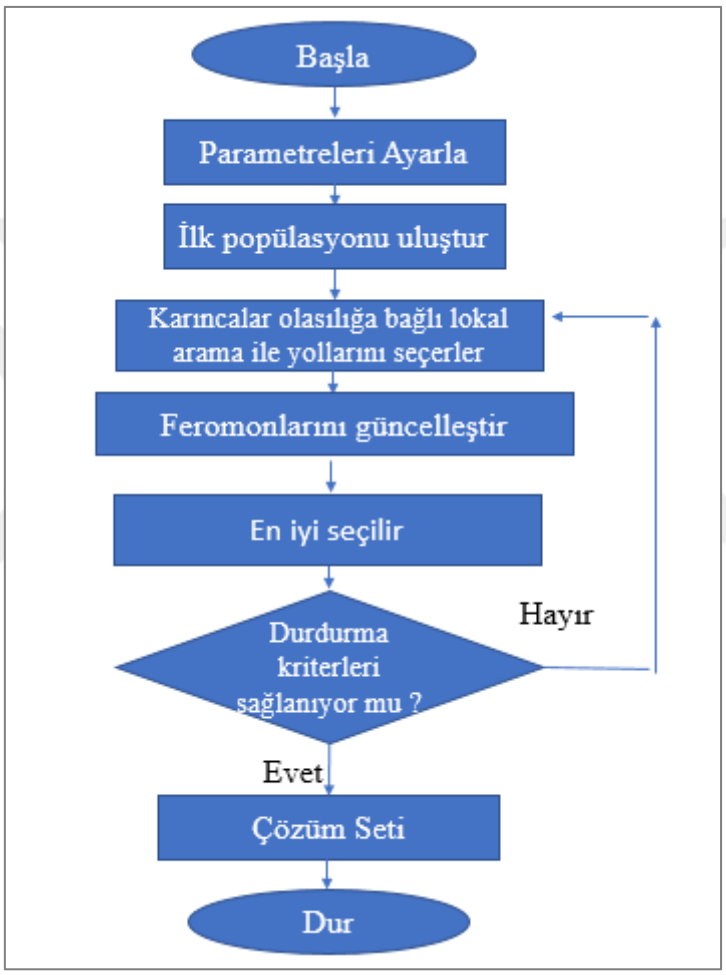
2. Güncelleme formülü

- α Feromon buharlaşma parametre $0<\alpha<1$
- φ Hareket ettiği noktalar.
- $J_{j,i}$ Lokal ve global ait toplam geçtiği tur uzunluğudur.

$$\tau(i,j) \leftarrow (1-\alpha).\tau(i,j) + \sum_{k=1}^{m} \Delta\tau_k(i,j) \quad (2.7)$$

$$\Delta\tau_k(i,j) \begin{cases} \frac{1}{J_{j,i}} & (i,j) \in \varphi \\ 0 & \text{aksı takdirde} \end{cases} \quad (2.8)$$

ACO'nun Akış diyagramı şekil 2.15'de göstermektedir.



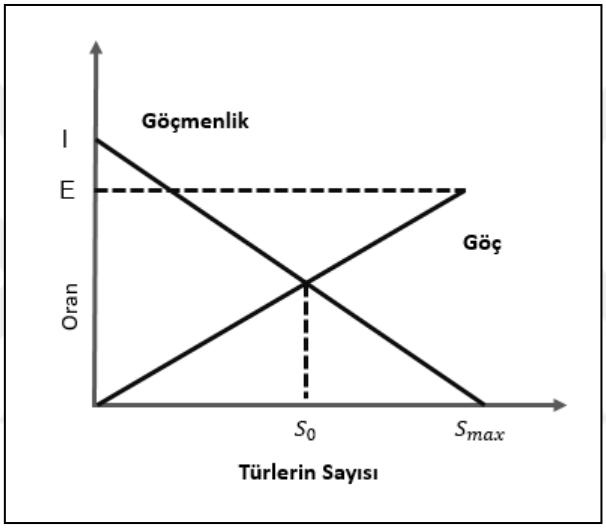
Şekil 2.15. ACO'nun Akış Diyagramı

2.2.5. Biyocağrafya Tabanlı Optimizasyon

Simon tarafından yeni modern sezgisel algoritmayı doğada yaşayan türlerin dağıtımını önermiştir. Bu algoritma BBO algoritmasıdır. Popülasyona dayalı bir yöntemdir. Biyocoğrafya'nın matematiksel modelleri, türlerin bir adadan diğerine nasıl göç ettiğini, yeni türlerini nasıl ortaya çıktığını ve türlerin neslinin nasıl tükenmekte olduğunu açıklamaktadır.

Burada "ada" terimi kelimenin tam anlamıyla değil, tanımlayıcı olarak kullanılmaktadır. Yani, bir ada, diğer yaşam alanlarından coğrafi olarak izole edilmiş herhangi bir yaşam alanıdır. Daha genel bir terim olan Habitatta (yaşam alanı) kullanılır. Biyolojik türler için konut olarak iyi uyumu bulunan coğrafi alanların yüksek habitat uygunluk indeksi (HSI) olduğu söylenir. (SIVs) Yaşanabilirlik, yağış, topoğrafya, bitki örtüsü çeşitliliği, sıcaklık vb.
Özelliklerle ilgilidir. Şekil 2.16'da gösterildiği gibi yaşam alanı uygunluğu geliştikçe tür sayıları artar. Bu prosedür de iki işlem söz konusudur.

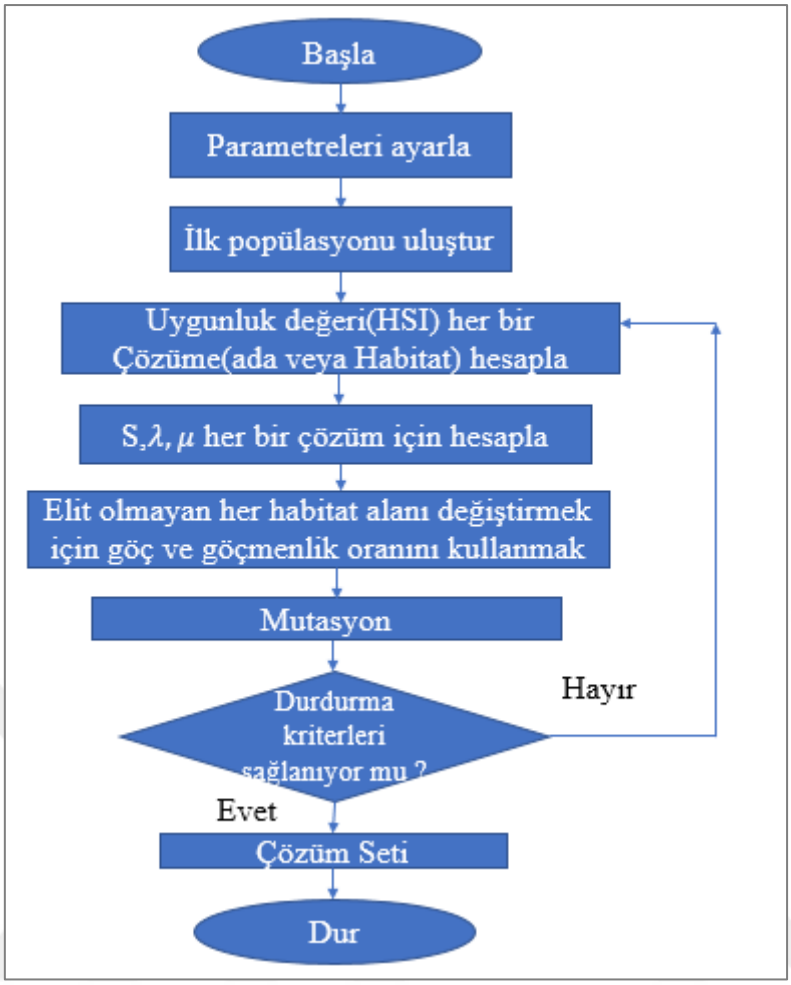
- Göç artar (daha fazla hayvan yaşamsal ortamdan uzaklaşır).
- Göçmenlik azalır (daha az tür yaşam alana girer).



Şekil 2.16. Tek yaşam da türlerin bolluk modeli

E: maksimum göç oranı
μ: Göç oranı
I: maksimum göçmenlik oranı
λ: Göçmenlik oranı
S_0: Türlerin denge sayısı
S: Burada habitata ait olan türlerdir.

Şekil 2.17'de akış diyagramı göstermektedir. Gördüğünüz gibi olasılıksal işlem habitatlarını üzerinde yapılmaktadır ve ilgili değişkenleri güncellemektedir. Diğer bir adımda mutasyon işlemi farklılık davranışı üretmektedir ve çözüme esnek amacıyla kullanılmıştır.



Şekil 2.17. BBO'nun akış diyagramı

Habitatın türlerin ihtimalini tam olarak düşünürsek. Zaman zaman $(t+\Delta t)$ aşağıdaki gibi değişir.

$$p_s(t+\Delta t)=p_s(t)\left(1-\lambda_s\Delta t-\mu_s\Delta t\right)+p_{s-1}\lambda_{s-1}\Delta t+p_{s+1}\mu_{s+1}\Delta t \quad (2.9)$$

Başka bir deyişle eğer bir habitatın S türeleri şu zamanda $(t\pm\Delta t)$ aşağıdaki şartların birisini yapılmaktadır.

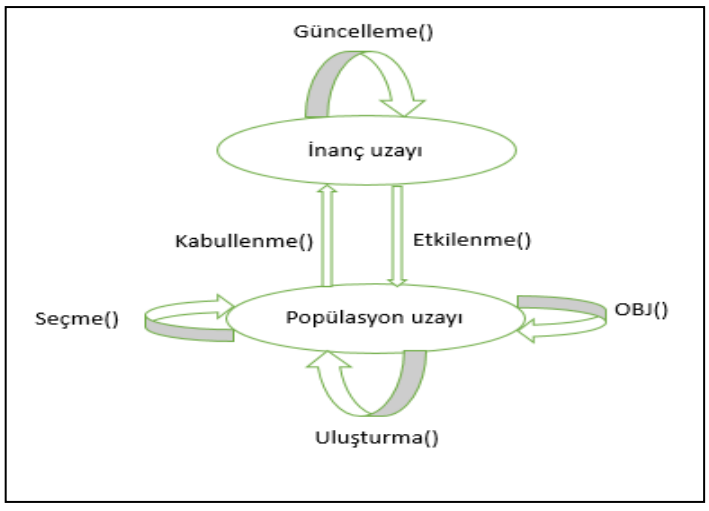
1) S türleri t zamanda varsa ve göçmenlik yoktur ya göç şu zamanlarda t ve $(t\pm\Delta t)$ aralarında yapılır.
2) (S-1) türleri t zamanda varsa, bir tür göçmenlik yapar.
3) (S+1) türleri t zamanda varsa, bir tür göç eder.

2.2.6. Kültürel Algoritma

Kültürel Algoritması, Reynold tarafından[38], toplumsal evrimi ve ajan temelli toplumlarda öğrenmeyi modelleyen bir araç olarak önermiştir. Kültürel Algoritması, evrimsel ajan popülasyonunun deneyimleri çeşitli sembolik bilgi formlarından oluşan bir inanç alanına entegre yaparak oluşur. 5 çeşitli genel bilgi sürekli kültürel alanında kullanmaktadır. Bilgi kaynakların arasında normatif bilgi, mekânsal bilgi (topografı), zamansal (tarihi), alan bilgisi ve örnek bilgi bulunmaktadır. Kültürel Algoritmanın ikili bir miras sistemi olarak iki temel bileşeni vardır: Popülasyon uzayı ve inanç uzayı.

Her nesilde Popülasyon uzayındaki bireyler ilk olarak bir amaç fonksiyonu obje() ile değerlendirilir. Kabul etme fonksiyonu kabullenmek (), bundan sonra hangi kişilerin Beyan Uzayı güncellemesine izin verileceğini belirlemek için kullanılır. Kabul etme fonksiyonu bireylerin aralarından hangi bireyin İnanç Alanını güncellemesine izin verilmesi için kullanılır. Seçilen bireylerin deneyimleri daha sonra güncelleme fonksiyon güncellemek () ile inanç uzayın içeriğine eklenir. Sonra, inanç uzaydaki bilgiler etkileme fonksiyon etkilenmek () ile popülasyonun gelecek nesil için bireylerin seçimine izin verirler. Bu, ikili kalıtım fikrini desteklemektedir; çünkü, popülasyon ve inanç uzayı birbirinden geri bildirimlere dayalı olarak her adımda güncellenmektedir. CA, önceden tanımlanmış belirli kriterlerin yerine getirilene kadar bu işlemi her nesil için tekrarlar.

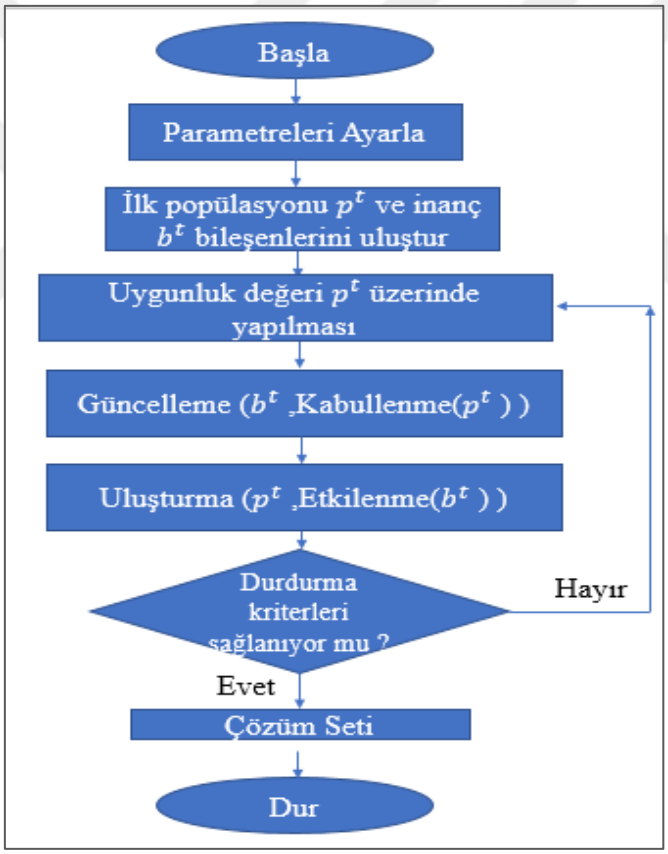
Bu şekilde, popülasyon ve inanç uzayın bileşenleri, benzer şekilde insan kültürünün evrimi için birbirleriyle etkileşime geçerler ve birbirlerini desteklerler. (Şekil 2.18, 2.19)



Şekil 2.18. CA'nın çerçevesi

Kültürel bilginin beş temel kategorisi, Bilişsel Bilim ve Göstergebilimdeki literatüre dayanılarak tanımlanmıştır:

1. Normatif bilgi: Bireyleri belirlenen aralığa atanması.
2. Durumsal Bilgi: Bireyleri "örneklere doğru hareket ettirmeye" yönlendirir.
3. Topoğrafik Bilgi: Bireyleri arama alanındaki en iyi performans gösteren hücrelere doğru yönlendirir.
4. Alan Bilgisi: Araştırmaya rehberlik etmek için problem alanı hakkındaki bilgileri kullanır.
5. Tarihsel veya Zamansal Bilgi: arama sürecini izler ve aramadaki önemli olayları kaydeder ve sonrada bireylerin üzerinde doğru seçim yönünde kullanır.



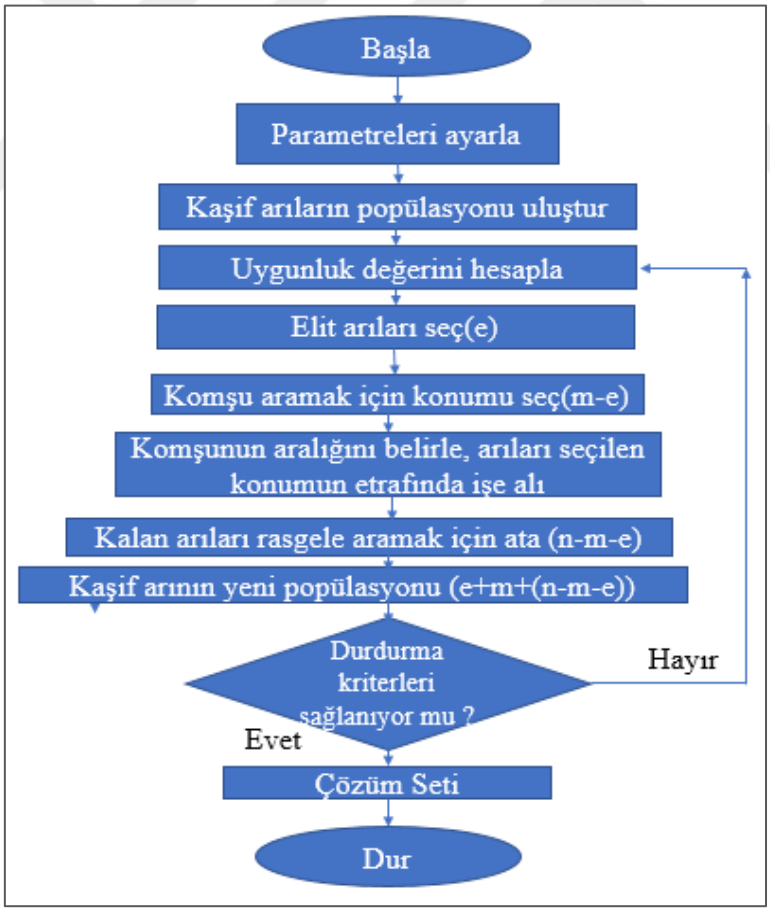
Şekil 2.19. CA'nın akış diyagramı

2.2.7. Arı algoritması

Bal arıları kolonisi büyük ve dağınık canavar olarak açıklanabilir; kendilerine uzak mesafelerde (10 km'den fazla) ve çok yönlerde dağılmışlar ve aynı anda çok sayıda gıda

kaynağından yararlanmaktadır diğer algoritma ona benzer işlemler yapılmaktadır[39]. Koloni, ancak avcılarını iyi alanlara yerleştirerek başarılı olabilir. Prensip olarak, çiçek parçalar fazla miktarda nektar veya çiçektozu sağlamakta, daha az enerji kullanımı ile toplanması kolaydır ve bu işlem için daha fazla arı olmalıdır, aksı takdirde daha az nektar veya çiçektozu içeren parçalar daha az arı alınacaktır. Bal arıların Algoritması arıların doğal avlanma davranışlarından esinlenerek uygun bir çözüm bulmaktadır. Algoritma, bir dizi parametrenin başlatılmasıyla başlar:

- n, Kaşif arıların sayısı
- e, Elit arıların sayısı
- m, n noktadan seçilen bölgeden hariç sayısı
- Elit bölgelerde Görevli arıların sayısı, seçilen diğer alanlardaki (m-e) Görevli arıların sayısı
- Durdurma kriterleri



Şekil 2.20. Arı algoritmanın akış diyagramı

Arı algoritmasında en önemli bileşenlerin birisi Komşu analizidir. İzleme işlemi komşu araştırması olarak kullanılabilir. Prensip olarak, bir kâşif arısı iyi bir alan bulursa (iyi bir çözüm), konumuna daha fazla arı davet eder. Ardından, bu arılar o kaynağa uçarlar, nektar parçasını alır ve tekrar kovanlarına dönerler. Kaliteye bağlı olarak kaynak bazı arılar tarafından reklam edilebilir. Önerilen arı algoritmasında bu davranış komşu araştırması olarak kullanılmıştır. Her bir avlanma konumunda (veya kömüşü konumda) sadece bir arı seçilir. Bu arının o alanla ilgili en iyi çözüm bilgilerine sahip olması gerekir. Böylece algoritma, önceki çözümlerle ilgili bazı çözümler üretebilir. Arıların önceden tanımlanmış komşu aralığı rastgele dağılıma dayalıdır. (Şekil 2.20)

2.2.8. Harmoni Arama

Harmoni arama (HS) ilk olarak Zong Woo Geem ve arkadaşların Tarafından 2001'de geliştirilmiştir[40]. Harmoni araması, müzik tabanlı modern sezgisel optimizasyon algoritmasıdır. Buradaki müziğin hedefi, mükemmel bir uyum ahengi gözleminden geçirerek esinlenmiştir. Müzikteki uyum, bir optimizasyon sürecinde optimumumu bulması ile benzemektedir. Burada arama süreci bir caz müzisyeninin doğaçlama süreciyle karşılaştırılabiliriz. Bir yandan, mükemmel sevindiren harmoni ses güzelduyu standardı tarafından belirlenir. Bir müzisyen her zaman mükemmel bir uyum arayışında bir parça müzik üretmeyi düşünüyor. Öte yandan, bir optimizasyon problemin optimal çözümü, verilen hedefler altında ve sınırlamalar ile en iyi çözüm olmalıdır. Her iki süreç en iyi veya en uygun sonucu üretmeyi amaçlamaktadır. Bir müzik aletinin güzelduyu kalitesi temel olarak basamak (frekans), ses tonu (ses kalitesi) ve genlik (ses yüksekliği) tarafından belirlenir. Ses tonu büyük oranda harmoni içerik ile belirlenir. Yani ses sinyalinin dalga biçimleri veya modülasyonları ile belirlenir. Bununla birlikte, üretebildiği harmoniler büyük oranda belirli bir enstrümanın aralığı veya frekans aralığına bağlı olacaktır.

Harmoni araması daha ayrıntılı olarak açıklamak için, önce doğaçlama sürecini müzisyen tarafından idealleştirme sürecini açıklayacağız.

Bir müzisyen doğaçlamaya başladığında, üç olası seçeneği vardır:

(1) hafızasından herhangi bir ünlü müziğin parçasını çalması (uyum içinde bir dizi sahalar)

(2) bilinen bir parçayı benzer bir şekilde çalması (böylece ziftin biraz ayarlanması).

(3) yeni veya rastgele noktaları oluşturması.

Başka deyişle eğer doğaçlama sürecini optimizasyon çözüm probleme çevirirsek aşağıdaki Tablo 2.1'de gibi olacaktır.

Tablo 2.1. Doğaçlama sürecinin optimum çözümüne karşılığı

Doğaçlama Sureci	Optimum Çözümü
Müzik Örneği	Karar Değişkeni
Basamak Aralığı	Değişken Aralığı
Harmoni	Çözüm Vektörü
Güzelduyu	Amaç Fonksiyonu
Deneme	İterasyon
Deneyim	Matris hafızası

Harmoni algoritmasında 3 bileşen önemli rol oynamaktadır.

1- Harmoni hafızasının kullanılması önemlidir, çünkü genetik algoritmasındaki gibi en uygun birey seçimine benziyor ve bu en iyi harmonilerin yeni harmoni hafızasına taşınmasını sağlayacaktır. Bu hafızaya daha etkili bir şekilde kullanmak için, harmoni hafızasını kabul edilerek veya dikkat oranı genellikle $r_{accept} \in [0,1]$ olarak atanır.

2- İkinci bileşen basamak ayarlaması, bir basamak bant genişliği b_{range}ve bir basamak ayarlama oranı r_{pa} ile belirlenir. Müzikte aralık ayarı frekansları değiştirmek anlamına gelse de Harmoni arama algoritmasında biraz daha farklı bir çözüm üretmeye karşılık gelir.

$$x_{new}=x_{old}+b_{range}*\varepsilon \quad (2.10)$$

Burada x_{old}, harmoni hafızasından gelen mevcut basamak veya çözüm. Yeni basamak ayarlama yaptıktan sonra yeni basamak x_{new}. Yeni bir çözüm, aslında eski kaliteli çözümün etrafında olan basamağı biraz küçük rastgele bir miktar ile hafifçe değiştirebiliriz. Buradaki e, aralıktaki[1,-1] rasgele sayı üreticisidir.

Basamak ayarı genetik algoritmalardaki mutasyon operatörüne benzer (burada genetik algoritmayı ana modern sezgisel algoritması olduğu için algoritmaları tanıtmasına katkı sağlıyor). Ayar derecesini kontrol etmek için bir basamak ayar oranı (r_{pa}) atamakla yapılmaktadır.

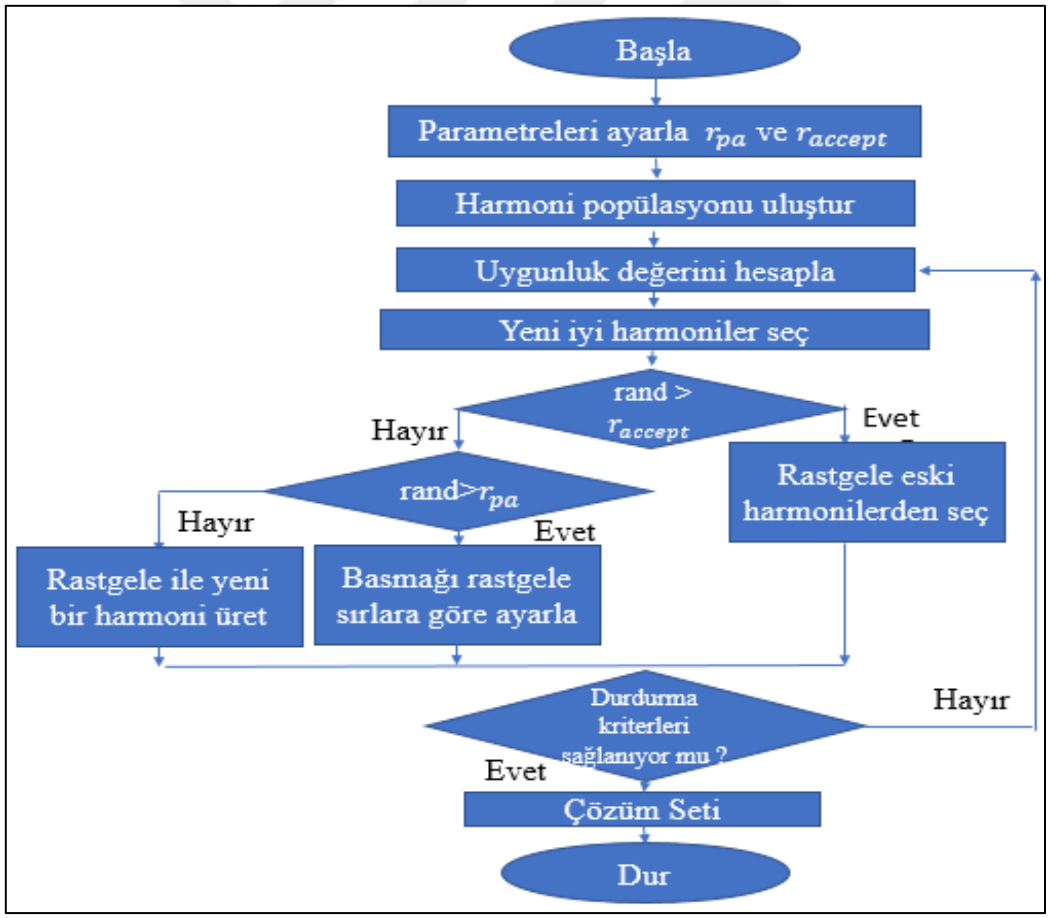
3- Üçüncü bileşen rastgeledir, çözüm çeşitliliğini arttırmak için kullanır. Basamak ayarlama işlevine benzer bir role sahip olsa da belirlenen yerel basamak ayarı yerel aramaya karşılık gelir. Rastgele dağıtımın kullanılması, sistemin küresel optimumunu bulmak için çeşitli farklı çözümleri keşfetmesine neden olabilir. Rasgele olasılığı şu şekilde olabilir:

$$P_{random}=1-r_{accept} \quad (2.11)$$

Basamak ayarın gerçek olasılığı:

$$P_{pitch}=r_{accept}*r_{pa} \quad (2.12)$$

HS Algoritmanın akış diyagramı Şekil 2.21 göstermektedir.



Şekil 2.21. HS algoritmanın akış diyagramı

2.2.9. Yerçekimi Arama Algoritması

Yerçekimi Arama Algoritması (GSA) Rashedi ve arkadaşların tarafından önerilmiştir. yer çekimi ve kütle etkileşim yasalarına dayanan bir optimizasyon algoritmasıdır. Bu algoritma Newton yerçekimi kanununa dayalıdır. Evrendeki her parçacık kuvvetle diğer parçacıkları çeker, kütlerin çarpımı ile doğru orantılı ve aralarındaki mesafenin karesi ile ters orantılıdır. Newton kuralı aşağıdaki formülde açıklanmıştır. (Şekil 2.22)

$$F = \frac{G * M_1 * M_2}{R^2} \tag{2.13}$$

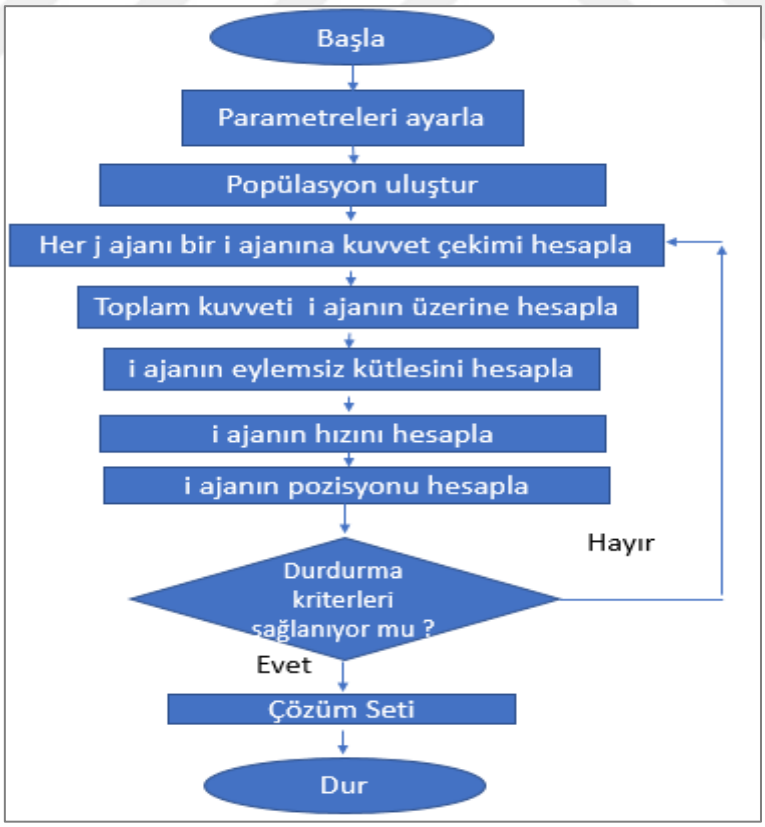
G: Her döngüde belirlenen bir değerden başlanıp indirmektedir

M_1: Birinci nesnenin kütlesi

M_2: İkinci nesnenin kütlesi

R: İki nesnenin aralarındaki mesafe

F: Yerçekimin kuvvetinin büyüklüğün



Şekil 2.22. GSA Algoritma 'nın Akış Diyagramı

2.2.10. Emperyalist Yarışmacı Algoritması

ICA[41], insan toplumundaki imparatorluklar arasındaki rekabet sürecini taklit eder. Genetik algoritmadaki kromozomlar gibi rastgele N boyutlu popülasyonlar ile oluşturulmuş ve bu popülasyonlara ülkeler denilir. Her bir ülkenin maliyeti hesaplanır. Daha sonra ülkeler emperyaliste ve koloniye ayrılıyor. Emperyalistler popülasyondaki en iyi ülkeler ve koloniler diğer kalan ülkelerdir. Sonra koloniler rasgele emperyalistlere dağıtılır. Bir emperyalistin elde ettiği koloni sayısı, onun gücü ile orantılıdır.

Burada, her emperyalistin gücü, maliyetine bağlı olarak hesaplanır ve normaliz edilir. Daha büyük güç değeri sahip olan emperyalist daha iyidir. Bir emperyalist ve koloni bir imparatorluktan oluşur, böylece birçok imparatorluk ile problemi çözmeye başlatır.

Sonra, her imparatorluk grubun içinde, koloniler önceden belirlenen kurala göre emperyalist konuma hareket ederler. Bu işleme "asimilasyon" denilir ve bu koloniler gerçek bir toplumda uyguladıkları asimilasyon sürecini taklit eder. Bu arada, bazı koloniler rasgele seçilir ve rastgele üretilen yeni ülkeler yerine alır. Bu işleme "devrim" denilir, tıpkı genetik algoritmadaki mutasyon operatörü gibi, gerçek bir toplumda kolooninin sosyal-politik özellikleri meydana gelen ani değişikliği taklit eder. Asimilasyon ve devrim sürecinde, bir koloni emperyalistten daha iyi olursa, koloni ve emperyalist yerlerini değiştirebileceklerdir.

İmparatorluklar arasındaki rekabet davranış ICA'nın ana işlemidir. ICA bu aşamada, tüm imparatorlar diğerlerin kolonilerini işgal etmeye çalışırlar. Öncelikle, her imparatorluğun (2.14) ve (2.15) formüllerine göre toplam maliyeti hesaplanır ve normaliz edilir. N imparatorluğunda $T.C._n$ toplam maliyeti ve $N.T.C._n$ normaliz olan toplam maliyetidir. ξ Pozitif bir değerdir.

$$T.C._n = \text{uygunluk hesaplanması}(\text{imparatorluk}_n) + \xi.\text{mean}\{\text{uygunluk hesaplanması}(\text{imparatorluk kolonileri})\} \quad (2.14)$$

$$N.T.C._n = T.C._n - \max_i\{T.C._n\} \quad (2.15)$$

Bu değer, kolonilerin imparatorluğun toplam maliyetini belirlenmesine rol almaktadır. Daha sonra, en zayıf imparatorluğun en zayıf kolonisi seçilir. Diğer imparatorluklar rekabet yoluyla elde etmeye çaba gösterirler. Her bir imparatorluğun başarı olasılığı formül 18'deki gibi hesaplanır. P vektörü formül 19'deki gibi olacaktır. P ile aynı büyüklüğe sahip olan R vektörün elemanları düzgün şekilde sayıları rastgele dağıtılmış (formülü 20).

$$P_{pn}= \left| \frac{N.T.C._n}{\sum_{i=1}^{N_{imp}} N.T.C._i} \right| \quad (2.16)$$

$$P= \left[p_{p_1}, p_{p_2}, p_{p_3}, \ldots, p_{p_{N_{imp}}} \right] \quad (2.17)$$

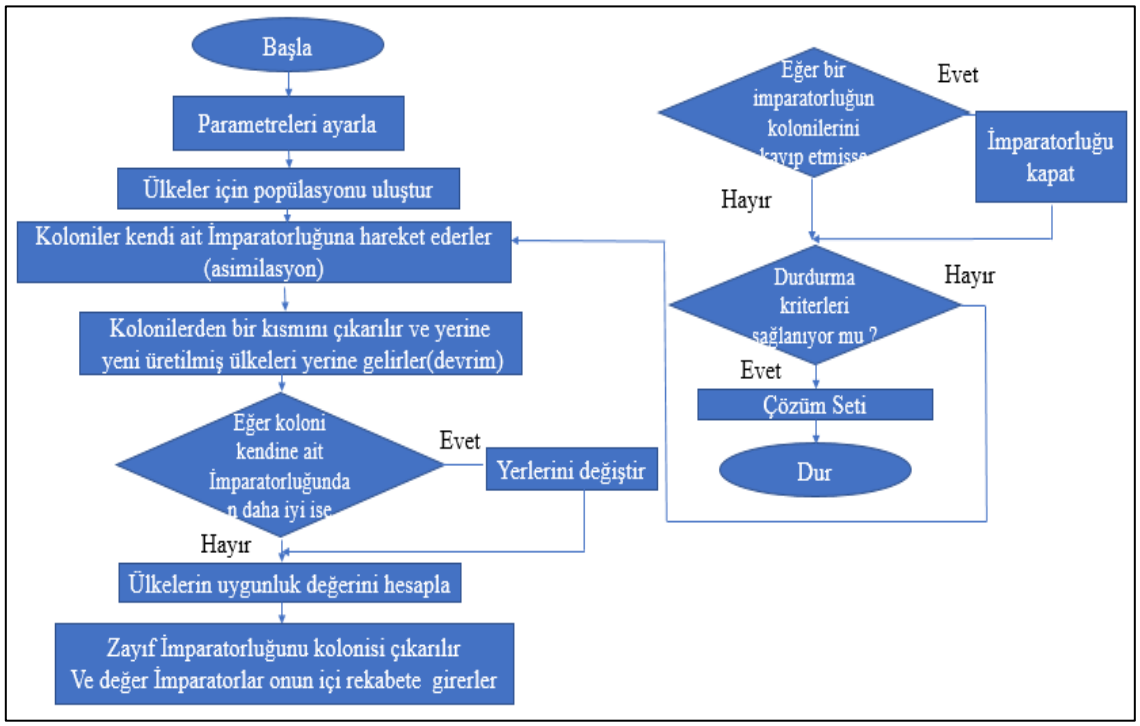
$$R= \left[r_1, r_2, r_3, \ldots, r_{N_{imp}} \right], \; r_i \sim \cup(0,1) \text{ ve } \; 1 \leq i \leq N_{imp} \quad (2.18)$$

Sonra D vektörü, formül 21'deki gibi R'den P çıkarır. D'deki ilgili indeksi en maksimum çıkaran imparatorluğu sonunda bahsedilen koloni elde edecektir.

$$D=P-R= \left[D_1, D_2, D_3, \ldots, D_{N_{imp}} \right] =$$

$$\left[p_{p_1} - r_1, p_{p_2} - r_2, p_{p_3} - r_3, \ldots, p_{p_{N_{imp}}} - r_{N_{imp}} \right] \quad (2.19)$$

Sadece bir imparatorluk kalmış ise veya önceden belirlenmiş maksimum yineleme sayısına ulaşırsa genellikle algoritmanın sonlandırma koşulu olarak kullanılır. Zayıf imparatorluğu kolonilerini kaybeder ve imparatorluğu çöker güçlü imparatorluk hepsine sahip olur. ICA'nın akış diyagramı şekil 2.23'de göstermektedir.



Şekil 2.23. İCA'nın akış diyagramı

2.2.11. Ateş Böceği Algoritması ve Literatür

Ateş böceği (FA) algoritması Yang [1] tarafından önerildi. Ateş böceklerinin yanıp sönen ışıklarından ilham alınarak oluşturuldu ve optimizasyonda kullanıldı. Bu algoritma, anlamak ve uygulama için optimizasyonda basit bir algoritmadır ve PSO algoritma sınıfına aittir. FA üç kuraldan oluşur.

1. Ateşböceklerin cinsiyetine bakılmaksızın, bir ateşböceği diğer tüm ateşböcekleri cezbedebilmektedir.

2.Çekicilik parlaklığa dayanır. Daha az olan ateşböceğin parlağı, fazla olan ateşböceklerin parlaklığına hareket etmektedir. Ateşböceklerin ışık yoğunluğu mesafe ile ters orantılıdır.

I: Ateşböceklerin ışık yoğunluğu

B: Çekici faktörü

r: Mesafe ile ters orantılıdır.

y: Işık emme katsayısı

B_0: Çekicilik sabit değeri

a: Rastgele parametresi

3. Burada, uygunluk değeri ateşböceğin yanıp sönen ışıklar için tanımlayabiliriz.

Her iki ateşböceğin arasındaki mesafe x_i ve x_j aşağıdaki formülü kullanılır.

$$r_{ij} = \sqrt{\sum_{m=1}^{D}(x_{i,m} - x_{j,m})^2} \tag{2.20}$$

$x_{i,m}$ Burada i'nin $ateşböceği$ ve m'nın ise uzaysal koordinat boyutudur. Genel uygulamada çekicilik fonksiyonu monoton olarak azaltılabilir ve aşağıdaki formülde gösterilir:

$$\beta(r) = \beta_0 e^{-yr^2} \qquad (m \geq 1) \tag{2.21}$$

Bir Ateşböceği yüksek çekiciliğinden dolayı diğer Ateş böcekler onun yönüne doğru hareket etmektedir. Hareket formülü aşağıdaki şekilde tanımlanır:

$$x_i^{new} = x_i^{old} + \beta_0 e^{-yr_{ij}^2}\left(x_i^{old} - x_j\right) + \alpha\left(rand - \frac{1}{2}\right) \tag{2.22}$$

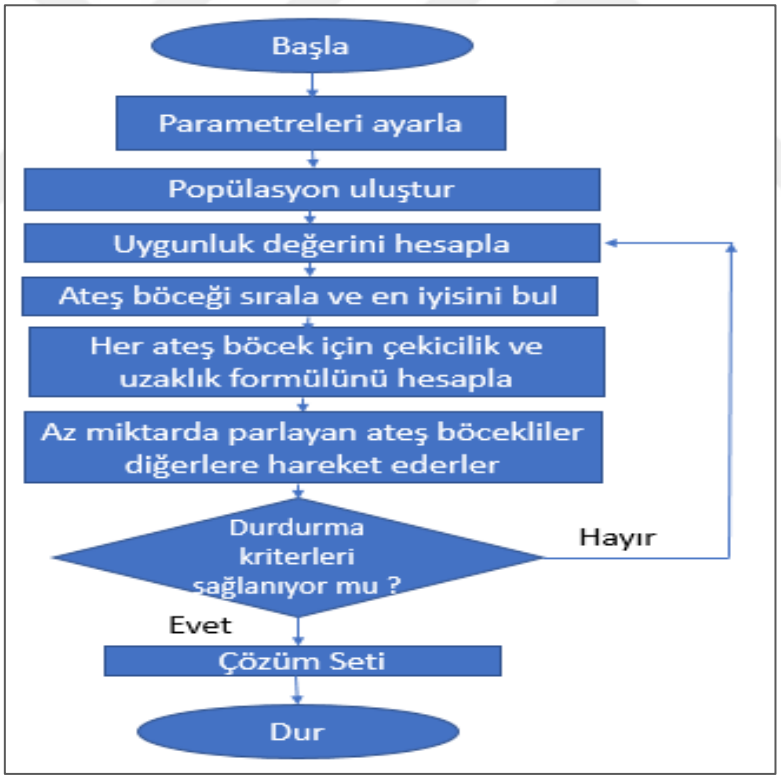
Ateş böceği algoritmasının akış diyagramı şekil 1.35'te gösterilmektedir. İlk terimin bir Ateş böceğinin mevcut konumunu göstermektedir. İkincisi ise Ateş böceği tarafından diğer Ateş böceği ışık yoğunluğuna göre çekicilik hesaplanmasıdır. Üçüncü terimde ise Ateş böceği daha parlak olanların olmama durumuna göre rasgele hareket etmektedir. Burada üniform dağılım ile hareketler varsayılmaktadır. α Parametresi, algoritmanın her döngüsünde indirgenmekte ve bu şekilde tanımlanmaktadır.

t: döngü sayısı

dam:bastırma değeri genelde =0.99

a_t=a_{t-1}*dam (2.23)

Önerilen algoritma FA algoritmasından yararlanılarak ilerdeki bölümde detaylı FA ve FATidal'ın felsefesi açıklanacaktır.



Şekil 2.24. FA'nın akış diyagramı

2.2.11.1. Geliştirilmiş Ateş Böceği Algoritmalar

Doğrusal olmayan tasarım problemlerini çözülmesi için geliştirilmiş veya değiştirilmiş Ateş böcek (FA) algoritmalar diğer mevcut olan Ateş böceği algoritmaya göre daha iyi performans göstermiştir. Levy-flight FA algoritması, FA'nın üç ideal kurallar ile L'evy flight özellikleriyle birleştirilmiştir.

$$x_i^{new} = x_i^{old} + \beta_0 e^{-\gamma r_{ij}^2}\left(x_i^{old} - x_j\right) + \alpha\left(rand - \frac{1}{2}\right) \oplus \text{L}'\text{evy} \quad (2.24)$$

Birinci ve ikinci terimler FA aittirler ve Üçüncü terim ise L'evy flight ve α ile rastgele parametresi olarak tanımlanmıştır. Lévy Flight, Lévy hareketi olarak da adlandırılır, sabit giderleri, Gauss dışı rastgele süreci bir sınıfı teşkil etmektedir. Sabit artışlar ile Lévy'nin kararlı dağılımına göre dağıtılan bir rastgele yürüyüştür [1, 42, 43]. LFA olarak adlandırılan bir başka geliştirilmiş Ateş böcek algoritması, Wang ve arkadaşların tarafından ışık yoğunluğunu değiştirerek yeni katkılar bulunmuştur. Işık yoğunluğunun değişimi her durumlarda kendiliğinden uyarlanabilir ve uygulamada herhangi bir probleme göre ayarlanabilir. Işık emme katsayısı ve rastlantısallık katsayısı ile yakınsama problemleri önlemişler ve büyük araştırma uzaylarında ise iyi çalışması yeni mutasyon önerilmiştir[44]. Verma ve arkadaşların tarafından geliştirilen ODFA algoritması, Ateş böceğe dayalı boyutsal davranışı ve başlangıçta Ateş böceklerin karşıtlık olarak hareket etmesi yeni zamansal problemi ve küresel minimum problemlere yol açmıştır. Karşıtlık temelli metodoloji, Çözüm adaylar olan her Ateş böceği zıt sayıdaki konumlarının başlatılması ile FA'nın yakınsama oranını iyileştirmiş. Aynı zamanda tüm araştırma uzayında da verimli bir şekilde aranmasını sağlamıştır. Boyutsal temelli metodolojide her ateşböceğinin pozisyonunu farklı boyutlarda hesaplanması optimum bulmasına neden olmuştur[45]. Diğer bir çalışmada NaFA[46] algoritması komşu kavram ile ateşböceklerini tanımlanmış. FA'nın metoduna göre, her Ateş böceği diğer tüm ateşböceklerin daha ışık yoğunluğuna sahip olanlar tarafından çekilebilir. Bununla birlikte, çok fazla çekimlerde, arama işlemi sırasında salınım ve yüksek hesaplama zaman karmaşıklığı problemleri ortaya çıkmıştır. Bu çalışmada her ateş böcek tüm ateş böceklerin yerine sadece belirlenmiş komşu sayısı ile komşu olan ateş böceği ile çekişme işlemini yapmaktadır. MFA algoritması Fister ve arkadaşların tarafından önerilmiştir. Bu algoritma lokal sezgisel arama ve Ateş böcek algoritmayı karıştırarak üçlü grafik renklendirme problemleri üzerine çalışmaktadır [47]. MSA_FFA algoritması, yeni seçim popülasyonunu tanımlanmış. Alfa, çekilme ve emme

katsayısı gibi üç parametreyi umut verici bölgelerde aramalarının kendi kendine çalışmalarını mümkün kılmış. Ayrıca bölgesel arama ve küresel aramalar da aralarındaki dengeleri sağlanmıştır[48]. MSA_FFA algoritmayı sürekli optimizasyon problemleri geliştirilmiş ve NMSA_FFA algoritma adlı olarak adlandırılmış. Metodun en önemli özellikleri: için kontrol parametrelerinin probleme bağlı seçimi kendi kendine adapte olması sağlanmış, bölgesel arama ve küresel arama arasında yeterli dengeyi sağlayan basit bir popülasyon modeli sunmuş ayrıca Luus-Jaakola rastgele lokal araştırma çalışmasında da kullanmıştır[49]. Kullanılan kurallardan birisi olan, bir ateşböceği daha parlak olan bir Ateş böceğe çekilir. Daha parlak olan bir Ateş böceği kendi kendine çekilme işlemi için Ateş böceği rastgele hareket eder. Rastgele hareket etmesi en iyi yönü rastgele üretilir olmasıdır ve ateşböceğin parlağını arttırır [50]. Diğer çalışmada ikili kodlanmış olan Ateş böceği, yazılımın ve donanımın hata tanımasına paralel ve dağıtılmış sistemlerde kullanılmış. Bu metotta, araştırma uzayında araştırmayı hızlandırmak ile ve probleme özgü bilgiyi ile uygulanamaz çözümleri uyarlanabilir bir ışık emme katsayısını sunulmuştur [51]. Bu çalışma, bir karışım modeli önermiştir. Öğrenme özdevinir tanımlayarak Ateş böceğin davranışını adapte yaparak ve genetik algoritmayı kullanarak global araştırmayı geliştirebilmesi için, yeni çözümler üretmiştir. Özet olarak söylemek gerekirse Önerilen algoritma üç kategoriden oluşmuştur:1- LA-FF (Learning Automata-Firefly) algoritma parametreleri adaptı şeklinde önermiştir. 2- HGA-FF (Hybrid Genetic ve Firefly) algoritması küresel araştırmayı geliştirmek için Ateş böceğin aldığı bilgileri iki popülasyonda ortak halinde kullanılmıştır. 3- Her Ateş böcek için Gauss dağılımı ve yönlendirilmiş bir hareket uygulanılmıştır[52]. Diğer bir çalışmada, Ateş böceğin hareketini dengeye tutmak için, etrafında daha iyi bir çözüm bulunmamak sızın, ateşböceğin hareketini en iyi yolu yönlendirmek amacı ile yeni bir hareket davranışı önermiştir. Yakınsama hızını arttırmasının yanı sıra, her tekrarda en iyi olan küresel değeri taşımak için Gauss dağılımı kullanılmıştır. Başlangıçta Alpha parametresi büyük değeri taşınarak gittikçe küçülmeye başlar yani küresel arama ile bölgesel aramaya geçmesine sebep olmuştur [53]. Analiz yapılmış olan ateş böcek algoritmasının üzerinde kaotik davranış özelliği sahip olmakta genellikle farklı çözümler üretmek için etkili bir karıştırma tekniği olarak kullanılabilir ve çekicilik değeri çok fazla ise kaos davranışı ortaya çıkabilir [54, 55]. Ateşböceğin algoritması formülünü çok kipli için geliştirilmiş ve GA ve PSO algoritmalara göre daha iyi performans göstermiştir[56].Diğer makalede sınırlamalı sürekli optimizasyon problemler için yeni algoritma önerilmiştir. Önerilen teknik, Ateş böceklerinin sosyal davranışlarından ve ışıldama iletişim fenomenden esinlenilmiştir[57].

2.2.11.2. Ateş Böceği Algoritmasının Uygulanması

Palit ve arkadaşların çalışmalarında, algoritmanın knapsack chiper'ın kripto analiz için bir ikili Ateş böceği Algoritması (FA) sunulmuştur. Uygulanan algoritma, Ateş böceklerinin yerlerini oluşturan ve 4 özellik karakterize edilmiştir. Bunlar arasında ışık yoğunluğu, mesafeler, çekicilik ve konum güncelleme, amaç fonksiyonu bulunmaktadır. Geliştiren ikili Ateş böceği şifrelenmiş metinleri düz metine deşifre yapılmasına yeni katkı sağlamış ve diğer mevcut olan bu kapsamda Genetik Algoritmaya (GA) göre daha iyi performans göstermiştir[58]. Ateşböceklerinin yanıp sönme davranışına dayanan ikili gerçel kodlanmış olan Ateş böcek algoritması (BRCFF), ağ ve güvenilirlik sınırlı Unit Commitment (UC) problemleri çözüm önerilmiştir. UC problemi ilk aşamasında, güvenilirlik sınırlı UC, ağ güvenlik sınırlamalarında tatmin etmeksizin, önerilen ikili kodlu ateşböceği algoritması kullanılarak çözmüştür. İkinci aşamada ise, birleştirilmiş ve güvenli bir optimal güç akışını (OPF) elde edilebilmesi için gerçek kodlanmış ateşböceği algoritmasını kullanmış [59]. Bu çalışmada, son zamanlarda geliştirilen Ateş böceği Algoritması kullanılarak Ekonomik Dispeç (ED) problemlerinin optimal çözümüne yönelik yeni bir yaklaşım sunulmuştur. ED problemi, eşitlik ve eşitsizlik sınırlamaları ile dışbükey olmayan amaç fonksiyona sahiptir[60]. Vektör nicelemenin kod çizelgesini(VQ) oluşturmak için sayısal imge sıkıştırma alanında başlangıçta Ateş böceği algoritmanın LGB metodu kullanarak vektör niceleme metodunu geliştirilmiştir[61]. Diğer bir çalışmada, Ateş böceği algoritması karelerin hatası toplaması ile amaç fonksiyon olarak tanımlanmış ve verilerin üzerinde kümeleme yapılmıştır [62].

Çoğu zaman, yüzey yaklaşımı verilen veri nokta seti, genellikle parametrik formda tanımlanan çok terimli ailesindedir. Veri noktaları için uygun parametrik değerlerin belirlenmesine yol açmıştır. Gerçek dünya ayarlarında, veri noktaları genellikle düzensiz olarak örneklenir ve gürültü ölçümünde maruz kalırlar. Burada Ateş böceği algoritması uygulanarak çok terimli Bezier yüzeylerin parametre problemi için kullanılmıştır[63]. ABC algoritmasını portföy seçim problemi için uygulanmıştır. Yakınsama süreci yavaş olması bir dezavantajı ortaya çıkmıştır. Bunun önüne geçmek için Ateş böceği algoritması ve ABC ile karışım yapmıştır[64]. Optimum alıcı yerleşim problemi çözmek için Ateş böceği algoritmasını ayrık modelleme yaprak ve iç-içe kümeleme stratejisini kullanarak yeni çözüm metodunu geliştirmiştir[65]. Tez kapsamında önerilen algoritma, kümeleme problemlerin üzerine uygulanmıştır ve yeni bulgular elde edilmiştir.

2.3. Kümeleme

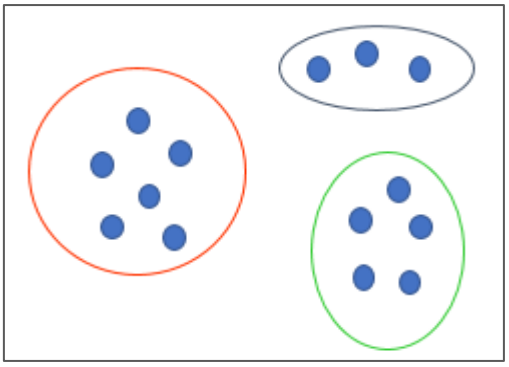
Kümeleme, veri madenciliğinde kullanılan tekniklerden biridir. Kümelemenin ana amacı benzer nesneleri (veri/nokta) aynı gruba koymaktır, ancak aynı grupta benzemeyen veriler bulunmamalıdır. Kümeleme, veri madenciliği, belge erişimi, imge tanıma ve nesne tanımadır [66, 67].

2.3.1. Tanımlar ve Gösterimler

Bir x nesne (veri/nokta) tek bir veriye ifade eder ve kümeleme algoritmalarda kullanılır. Genellikle bir d vektörün ölçümünden oluşur: x'in nesnesi bireysel sayıl bileşenleri olan x_i özellik veya öznitelik denilir. d nesnenin boyutudur x={x_1,…x_d}. Bir nesne seti $R = \{x_1, \dots x_n\}$ olmak üzere ve i'nin nesnesi R de bu şekilde x_i=$(x_{i,1},\dots x_{i,d})$ ifade edilmektedir. Çoğu durumda, kümelenmeye hazır olan nesne seti, n×d nesne matrisi olarak gösterilir. Bir sınıf, özellik uzayında dağılımı olan bir nesne kaynağı olarak görülebilir ve sınıfı özgü bir olasılık yoğunluk tarafından yönetilir. Aşağıdaki kavramlar kümeleme teknikleri açıklanmıştır.

- Kümeleme teknikleri, nesneleri gruplamaya çalışır; böylece elde edilen sınıflar, nesne setinde temsil edilen farklı nesne oluşturma süreçlerini yansıtır.
- Sert kümeleme teknikleri her bir nesne x_i bir sınıfa ait l_i olmak üzere sınıflarını tanımlamaktadır. Bütün etiketler her bir nesne seti için l={l_1,…l_n}, l_i∈{1,…,k} olmakla ve k burada kümenin sayısıdır.
- Bulanık kümeleme teknikleri her bir x_i nesnesi bir kesirli üyelik dercesine $f_{i,j}$ aittir. Burada j mevcut olan kümedir.
- Mesafe ölçüsü bir metriktir ve nesnelerin benzerliklerini ölçmek için özellik uzayda kullanılır[66]. Örnek olarak Öklid mesafesi, Manhattan mesafesi vb. söylenebilir.

Veri madenciliğinde kümeleme denetlenmeyen bir metottur: herhangi veri noktaların kümelere ait olduğu hiçbir bilgi almadan kümeleme metotları sunulmuştur. Şekil 2.5'te basit bir kümeleme gösterilmiştir.



Şekil 2.25. Basit kümeleme

2.3.2. Kümeleme Teknikleri

Bu bölümde, en iyi bilinen kümeleme metotları açıklanmaktadır ve çok yakın çalışmalarda bulunan teknikleri detaylı olarak tartışılmaktadır. Mevcut olan metotlar tezin kapsamı çok fazla içermeyen metotları kısaca açıklanmıştır.

2.3.2.1. Model Tabanlı Kümeleme

Grupları tanımlayan klasik kümelemenin aksine nesneleri model tabanlı kümeleme yöntemleri ile karakteristik tanımlama yapar ve her grup bir kavram veya kümeyi temsil eder.

2.3.2.2. Şebeke Tabanlı Kümeleme

Bu yöntemler, belirlenen uzayda, sonlu sayıda hücre oluşturur ve kümelemeyi tablo yapıda tüm işlemlerini yapmaktadır.

2.3.2.3. Hiyerarşik Metotları

Bu yöntemler, örüntüleri yukarıdan aşağıya ya da aşağıdan yukarıya doğru kümeleme yapılandırır[68-71].

Yığıştırma (Agglomerative):

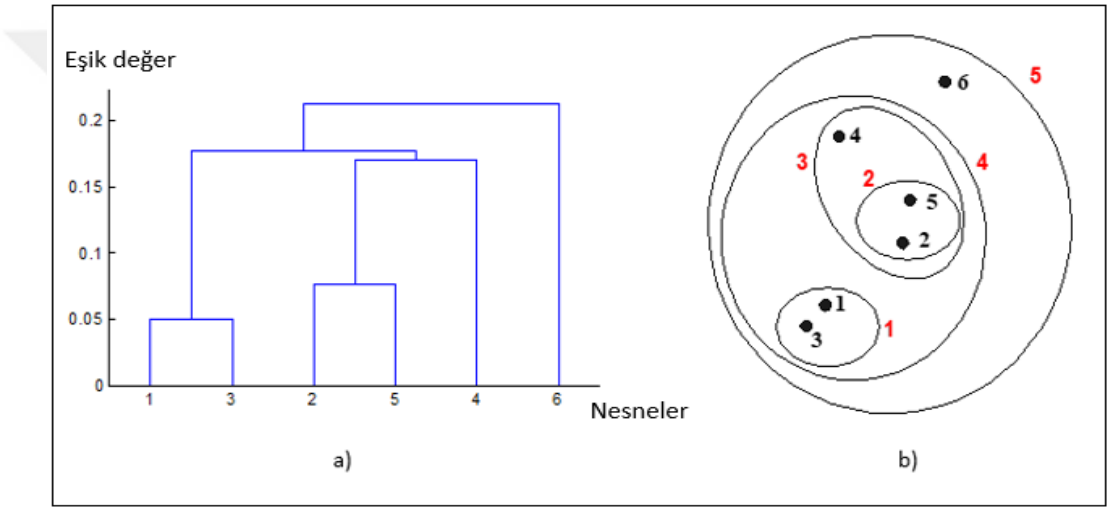
- Nesneleri (veya noktaları) tek tek küme halinde başlatın.

- Her adımda, bir küme (veya k kümeler) kalana kadar en yakın mesafede olan kümeyi birleştirin.

Bölüştürme (Divisive):

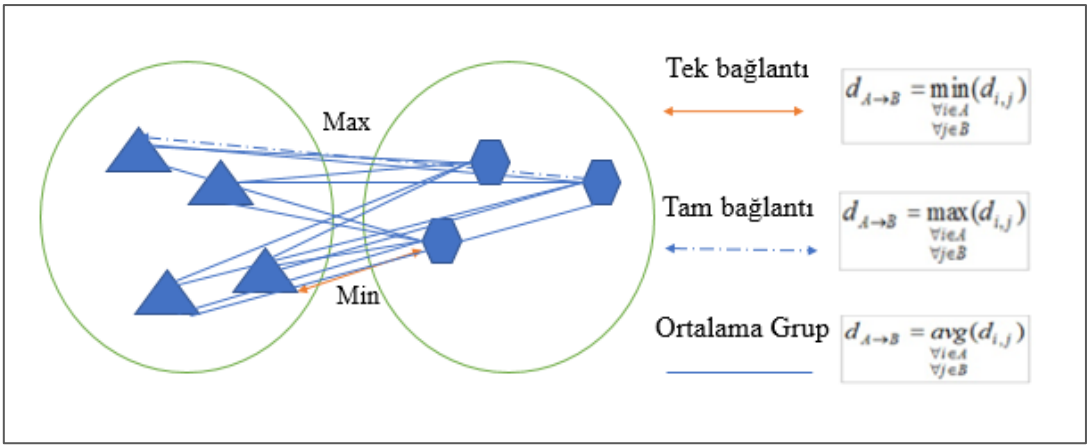
- Tek, bütün nesneleri (veya noktaları) bir küme olarak başlayın.
- Her adımda, her küme bir nesne (veya k kümeleri) içerene kadar bir küme ayırın.

Hiyerarşik yöntemlerin sonucu, nesnelerin iç içe gruplamasını ve gruplamaların değiştiği benzerlik seviyelerini temsil eden bir dendogramdır. Veri nesnelerinin kümelenmesi için, dendogram da benzerlik seviyesi kesilerek elde edilir. (Şekil 2.26)



Şekil 2.26. Hiyerarşik metottu a)kümelemenin dendogramı b)kümeleme[66].

Hiyerarşik kümeleme yöntemleri, benzerlik ölçümün hesaplanma biçimine göre daha da bölünebilir. Bu tekniklerde, üç farklı Tek Bağlantı, Tam Bağlantı ve Ortalama bağlantı mevcuttur. Şekil (2.27)



Şekil 2.27. Hiyerarşik kümeleme benzerlik ölçüsüne dayalı kriterler

Avantajlar ve dezavantajlarına incelersek şöyle aşağıdaki gibi açıklanacaktır:

Avantajlar:

1- Dendogram, görselleştirmek için çok faydalı olması.
2- Kümelerin arasında Hiyerarşik bağlantıyı sağlanması.
3- Kümeleri iç içe görsellik yapabilmesi.

Dezavantajlar:

1- Kümeler için seviye belirlenmesi çok zor.
2- Sonuçlar başlangıç parametrelere çok bağlı
3- Aykırı verilerin üzerinde çok hassas olması.
4- Bir veri bir kümeye atanırsa sonradan başka bir kümeye arınmaması.
5- Büyük verilerin üzerinde zaman problemi

BIRCH[68], CURE[69], ROCK[70] ve CHAMELEON[72] Ağaç metoduna dayalı keyfi şekilli kümeler gerçekleştirirler. Bununla birlikte, zaman karmaşıklığının üstesinden gelememekle, diğerlerine göre daha iyi sonuçlar göstermişlerdir. Karmaşıklık yapısının üstesinden gelmek için, hiyerarşik kümeleme algoritmasına dayanan bazı algoritmalar önerilmiştir. Bunlardan birisi hiyerarşik temelli otomatik versiyonudur [73]. Leaders-subleaders algoritma veri akışındaki zaman serileri açısından hiyerarşiklerin önemini ortaya koymuştur. [74] Öklid mesafe hesaplanması yoğun veri kümesindeki yoğunluk problemin varlığını açıklar Bunlarla başa çıkabilmek için, grafik modeli temelli olan SNN algoritmayı önerilmiştir. Algoritma, her veri noktasının en yakın komşularını bulup daha sonra komşu sayısına göre noktalar çiftleri arasındaki benzerliği yeniden tanımlar [75].

2.3.2.4. Bölme Dayalı Kümeleme

K-ortalama kümelemesi [76], gözetimsiz öğrenme türüdür ve etiketlenmemiş nesnelerin üzerinde (yani, nesnelerin kategorilerde tanımlanmamış durum) işlem görülür. Bu algoritmanın amacı, K değişkeni temsil eden grup sayısıyla nesnelerin gruplarını bulması amaçlanmaktadır. K-ortalama algoritması keskin bir algoritma olarak tanımlanır. Yani her nesne sadece bir kümeye ait olabilir. Bu metot her grubun nesneleri benzerliklerinin minimum olmasına ve gruplar arasının maksimum olmasını hedeflemektedir. Bu işlem belirlenmiş iteratif sayısına kadar her bir grubun uygun bir şekilde yani maksimum (grupların ayırışları) ve minimum (grupların içindekileri) sağlanması ile kümeye atanır. (Şekil 2.28)

Algoritma temel olarak 4 aşamadan oluşur:

1. Rastgele atarak Küme merkezlerinin belirlenmesi.
2. Merkeze yakın olan nesneyi mesafelerine göre gruplanması.
3. Yapılan gruplara göre yeni merkezlerin belirlenmesi.
4. Belirlenmiş kriterlere ulaşana kadar 2. ve 3. adımların tekrarlanması.

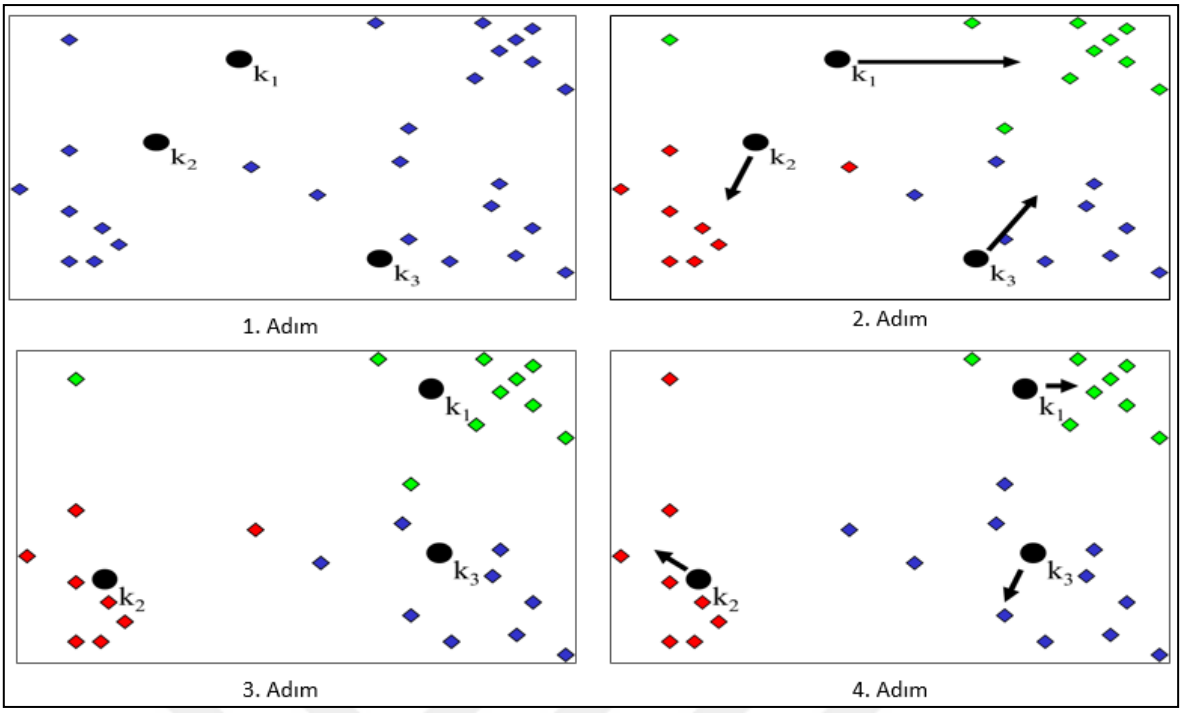
Avantajlar ve dezavantajlarına incelenirsek şöyle aşağıdaki gibi açıklanacak:

Avantajlar:

1- Veriler her aşamada farklı kümelerde bulunma şansları olması.
2- Programın basitçe yazılması.
3- Hızlı ve küre şeklinde olan verilerin üzerinde iyi çalışması.

Dezavantajlar:

1- K değeri belirlenmesinin çok zor olması.
2- Küresel kümeleme için çalışmaması.
3- Farklı başlangıçların farklı sonuçları göstermesi.
4- Farklı boyutta ve farklı yoğunlukta iyi çalışmaması.



Şekil 2.28. K-ortalama kümelemesi çalışırken alınan adımlar [76].

K_mediod [77], Pam[78] ve CLARA[79] da gürültü hassasiyetini indirgenmesini sağlayan bölümleme metoduna dayanmaktadır, ancak algoritmalar stratejik olarak keyfi şekilli kümelemelere değinmeyi başaramamıştır.

2.3.2.5. Yoğunluğa dayalı kümeleme ve Geliştirilmiş Olanlar

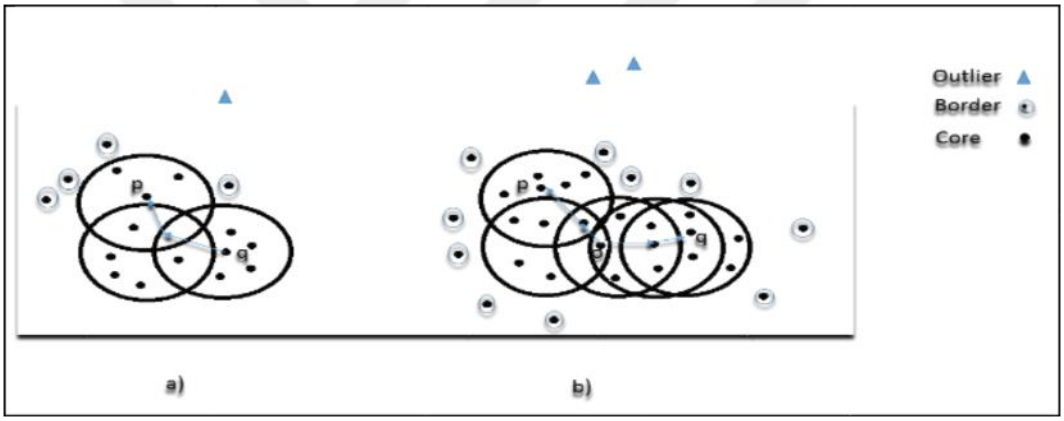
DBSCAN algoritması Ester ve arkadaşların tarafından önerilen algoritma örüntülerin/nesnelerin komşuları ile uzaklığı hesaplayarak ve hesaplanan bölgede eşik değerden daha fazla nesne bulunursa nesneleri bir kümeye atarlar ve böylece her bir nesne üzerinde bir defa hesaplamakta devam ederler. OPTICS algoritması, Ankerst ve arkadaşları tarafından önerilmiş ve DBSCAN algoritmasının dezavantajlarına başa çıkması için yeni bir yol açmışlar. DBSCAN algoritmasının iki parametre olan Epsilon (bulunduğu noktanın yarıçapı) ve MinPoints (minimum nokta değeri) değerlerine bağımlığını hafifletmek için verileri Epsilon değerini grafiksel bir ifade ile kümele yapılmasını sağlar.

DBSCAN algoritması gürültü verileri ayırt etmesi ve veri setin üzerinde şekilbilgisi yapabilmesi kümeleme yeteneği sahiptir. İki parametre Epsilon (bulunduğu noktanın yarıçapı) ve MinPoints (minimum noktanın değeri) kullanıcı tarafından tanımlanır. DBSCAN kavramsal olarak aşağıdaki gibi tanımlanabilir. Komşular, farklı mesafe fonksiyonlarıyla tanımlanır. İki

nokta p ve q dist(p, q) mesafesi olmak üzere bir p'nin Epsilonu {q∈D‖dist(p,q)≤Epsilon} tanımlanmıştır. Çekirdek nesnesi veya noktası, Epsilon 'unun en az bir minimum nokta sayısı (MinPoints) içermesi ile tanımlanır (Core Point). Burada küme seti$C = \{c_1 \dots c_k\}$ ve k küme sayısıdır. Aykırı nokta/nesne, D veri setindeki herhangi bir c_i kümeye ait olmayan ve çekirdek nesnesinden uzakta olan nesneyi aykırı nokta=$\{p \in D \parallel \forall_i : p \notin D\}$ söylenir (Outlier Point). Eğer çekirdek bir nesne ve aykırı bir nokta değilse ve aynı zamanda bir nokta çekirdek nokta tarafından yoğunluk ile erişilebilir ise o nokta sınırda kalan noktadır (Border Point). (Şekil 2.29)

P nesnenin q'nın Epsilon-komşusu içerisindeyken ve q da bir çekirdek nesnesi olursa o nesne doğrudan yoğunluğa erişilebilen bir nesnedir.

- $P \in N_{Eps}(q)$ ($N_{Eps}(q)$, q'nın epsilon komşusu)
- $|N_{Eps}(q)| > MinPts$ (Çekirdek nesnesi olmanın şartları)



Şekil 2.29. Yoğunluğa dayalı kümeleme tanıtımı (a) p nesnesi q nesnesinden Yoğunluk ile erişilebilir bir nesnedir, (b) p ve q nesneleri yoğunluğa bağlı nesnelerdir.

Bir p nesnesi q nesnesinden yoğunluğa dayalı erişebilirse aşağıdaki gibi ifade edilir. Doğrudan yoğunluğa erişilebilen bu şekilde olacaktır. P=$\{p_1,\dots p_n\}$,p_1=q ve p_n = p olmak üzere $p_i + 1$ nesnesi p_i nesnesinden Epsilon ve MinPointsa bağlı olarak Doğrudan yoğunluğa erişilebilir. $1 \le i \le n, p_i \in D$. Bu şartların sağlanması gerekiyor Şekil 1.40(a) (Maximality).

P ve q aralarında o ∈ D bir nesne varsa ve aynı zamanda Epsilon ve MinPoints miktarlarına göre yoğunluğa bağlı ise Şekil 1.40(b) de gösterildiği gibi yoğunluğa bağlı duruma girerler (Connectivity)[76].

Avantajlar ve dezavantajları incelerise şöyle aşağıdaki gibi açıklanır:

Avantajlar:

1- Küme sayısına ihtiyaç duymadan çalışması.
2- İki parametre ile çalışması.
3- Gürültü ve aykırı verilere hassas olması.
4- Gelişigüzel kümelemelerin üzerine çalışılması.

Dezavantajlar:

1- Verilerin sıra ile işlem görmesi.
2- Parametre değerleri bazen çok zor belirlenmesi.
3- Çok zaman harcaması.

Diğer teknikler olan model tabanlı kümeleme, şebeke tabanlı kümeleme ve yumuşak hesaplama kümelemesi bulunmaktadır.

ST-DBSCAN[80], DBSCAN algoritmasını modifiye ederek, yoğunluk faktörünü kullanarak uzaysal ve zamansal verilerin üzerinde benzer yoğunluklara ve aynı zamanda gürültü verilerini bulmaya çalışmaktadır. GMM algoritması [81] çekirdek yoğunluğuna dayalı olarak çalışarak ve az sayıda bileşen içeren yoğunluk bölgesini tahmin eder ve sonuçta keyfi şekil kümelerini elde eder. GDD algoritması [82] Gauss çekirdeği ve Öklid uzaklığı ile birleştirerek önceden bilgi ve parametreler olmaksızın yeni bir kümeleme yöntemi sunulmuştur.

Keyfi şekil kümeleme elde etmek amacı ile yoğunluğa dayanan ROUGH-DBSCAN algoritması [83] zaman problemin üstesinden gelmek için, algoritma, yüksek yoğunluklu verilerin seçiminde, çekirdek lideri kullanılmış. Bir diğer kümeleme türü bulanık kümeleme temeli olan FCM [84] algoritmasıdır. FN-DBSCAN[85] DBSCAN algoritmasına dayalı olarak, keskin komşu ilişkisinin yerine Bulanık komşu ilişkilerin kavramı kullanılarak tanıtıldı. FN-DBSCAN algoritması DBSCAN'dan daha sağlam sonuçlar verilmiştir. DENCFURE[86] Bir veri seti hakkında önceden bilgi sahibi olmayan benzer yoğun kümeleri elde etmek için veri noktaları arasındaki bulanık yakınlık ilişkileri kullanır. Spektral kümeleme (SC) [87] algoritması, yoğunluk tabanlı kümeleme algoritmalarından biridir, grafik yöntemiyle kümelerin üzerinde veri kümeleme yapılmıştır. Algoritma, yapısal olmayan verilerin üzerinde iyi performanslar göstermiştir.

LOF algoritması[88] Veri kümesindeki her veri için ikili kavramından derece kavramına (derece formülü) dönüştürdükten sonra, yakın verileri (kümedeki normal veriler) ve uzakta olan verileri (aykırı veriler) ayrıt etmek için derece formülünü kullanılmıştır.

2.3.2.1. Yumuşak hesaplama kümeleme

Bu yöntemler, genelde yapay sınır ağlar, sezgisel ve modern sezgisel algoritmalar ile kümelemenin problemlerine üzerinde yeni katkılar bulunmuştur [89]. Optimizasyon açısından bakıldığında, kümeleme problemleri bir tür NP-hard problemi olarak düşünülebilir[90].

Bu tür algoritmalar, için en uygun çözümü bulmayı çalışmaktadırlar. Kümelemede yerel optimumu tıkınmasının riskini azaltırlar yani algoritma en uygun mevcut olan kümeleri bulmaya çaba gösterir. Bölümleme kümeleme stratejisini kullanan modern sezgisel algoritmalar ABC [91], ANT[92],PSO[93],GSA[94] ve FA[95] iyi sonuçları göstermişlerdir. Bu algoritmalar çoğu yeni amaç fonksiyonu sunmaktadır ve ayrıca bunlara dayalı geliştirilmiş olan algoritmalar mevcuttur. Kümeleme diğer alanlarda da söz konusudur. Genetik algoritmayı ve yapay sinir ağların tekniklerinden birisini karıştırarak kablosuz sensor ağlarda kullanılmıştır[96]. Diğer çalışmalarda örnek olarak piyasa segmenti[97], Büyük veri[98], internet[97, 99] kümeleme teknikleri kullanılmıştır.

Tez kapsamında önerilen modern sezgisel algoritmayı (FAtidal) ve veri yoğunluğuna dayalı DBSCAN algoritma ile yeni bir algoritma geliştirilmiştir. Kümelemede bulunan problemler ve mevcut olan algoritmaların dezavantajlarını ele alınarak yeni bir kümeleme algoritmayı önermesine sebep olmuştur. Bu çalışmada, küme sayısı, verilerin örtüşme ve keyfi şekil olan veri grupların probleminin üzerine odaklanmıştır. İlerdeki bölümde problemler detaylı şeklinde açıklanacaktır.

2.3.3. Ölçütler

Rand Ölçümü, iki veri kümelemesi arasındaki benzerliğin bir ölçüsüdür. Matematiksel açıdan, Rand ölçümü doğrulukla ilgilidir ve sınıf etiketleri kullanılmadığında bile geçerlidir. 0 değer sonucu en kütü benzerliği ve 1 değer sonucu en iyi benzerliği sunmaktadır.

$$\mathrm{RI} = \frac{a+b}{\binom{n}{2}} \tag{2.25}$$

- Formüldeki a değişkeni, bir çift örneğin aynı kümeye ait olma sayısını, iki farklı kümeleme sonucuna bağlılığını belirtir.

- Formüldeki b değişkeni, bir çift örneğin aynı kümeye ait olmama sayısını, iki farklı kümeleme sonucuna bağlılığını belirtir.
- n Sırasız çiftler.

Diğer küme sayısına bağımsız olan indeks Adjusted Rand İndex aşağıdaki formülde açıklanmıştır. 1 değer sonucu en iyi benzerliği sunmaktadır ve negatif değer ise az benzerliğe sahiptir.

$$\text{ARI}= \frac{\Sigma_{ij}\binom{n_{i,j}}{n}- \left[\Sigma_i\binom{a_i}{2}\ \Sigma_i\binom{b_j}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\Sigma_i\binom{a_i}{2}\ \Sigma_i\binom{b_j}{2}\right]-\left[\Sigma_i\binom{a_i}{2}\ \Sigma_i\binom{b_j}{2}\right]/\binom{n}{2}} \tag{2.26}$$

Rand ölçümü küme sayısına ve öğe sayısına bağlıdır ama Adjusted Rand ölçümü null hipotez olarak genelleştirilmiş hipergeometrik dağılım varsayılmıştır. Sonuçta küme sayısı bağımsız hesaplama işlemini sunmaktadır [100].

Diğer ölçüm F_ measure aşağıdaki iki kavramın harmonik ortalamasını hesaplanır.

$$\text{kesinlik (percision)}=\frac{\{\text{İlgili getirimi}\}\cap\{\text{bütün veri çıkarımı}\}}{\{\text{Bütün veri çıkarımı}\}} \tag{2.27}$$

$$\text{Hassasiyet(Recall)}=\frac{\{\text{İlgili getirimi}\}\cap\{\text{bütün veri çıkarımı}\}}{\{\text{ilgili veri çıkarımı}\}} \tag{2.28}$$

$$F_1\,\text{Skoru}=2\frac{\text{Kesinlik. Hassasiyet}}{\text{Kesinlik+Hassaiyet}}=2\frac{pr}{p+1} \tag{2.29}$$

3. BULGULAR

3.1. Gelgit

Hindistan'daki Ahmedabad da yapılan kazılara dayanarak, gelgit havuzları eskiden gemileri tamir etmek amacı ile kullanıldığı bulunmuştur. Bulunan havuz eski çağlardan beri insanoğlunun dikkatini çeken bu fenomene M.Ö. 2450 yılları dayanıyor. Orta çağdan sonra, üç gelgit teorisi bulunmaktadır. İlk teori, dünyanın güneşteki yıllık dolaşımının yanı sıra eksen boyunca günlük dönüşüyle denizlerde hareket yaratacağını Galileo tarafından sunuldu. Her yerde deniz yatağının şekli ve geometrisi nedeniyle değişiklikleri gösterirse gelgitler meydana gelmiş demektir. İkinci teori ise, Fransız filozofu Descartes ile ilişkili olup, ayın çevresindeki alanın, eterden dolu olduğunu düşünüyormuş. Ay dünyayı dolaştığında bu nesne sıkıştırır ve eter de bu baskıyı denize bırakır, böylece gelgit ortaya çıkacaktır. Kepler üçüncü teoriyi sundu. O, okyanus sularının yükselme nedeni ayın çekim kuvveti olduğunu söyledi. Bu çekim kuvveti okyanus sularına uygulanan dünyanın çekim kuvveti ile dengelenmektedir.

Güneşin merkezi olma fikri ve Güneş sistemin gezegenlerinin her birinin rotasyonu güneşin etrafında bulunurken bu fikir daha kuvvetli yer almaktadır. Kepler'in görüşleri daha yaygın olarak kabul edilmiştir. Bununla birlikte, bu görüşler iki defa gelgitin bir ay gününde (bazı bölgelerde) ortaya çıkma gerçeğini ifade edememiştir. Fakat Newton kanunu çıkmasıyla birlikte bu soruya da cevap vermiştir. Her iki nesne birbirinin aralarındaki mesafenin karesi ile orantılı olan bir kuvvetle birbirini çekmektedir. Bu kuvvet vücuttaki kütle ile orantılıdır.

3.1.1. Gelgitin Oluşma Mekanizması

Gökyüzündeki cisimlerin gelgit akışı, özellikle ayın ve güneşin çekim kuvveti etkisinin sonucudur. Bu kuvvetler su yüzeyi yüksekliğini arttırır ve bu artış yatay gelgit akışlarına (Tidal stream) sebep olur. Dünya kendi etrafında dönerken, suyun yüzeyindeki yükseklik değişiklikleri dünyanın her iki noktasında da ortaya çıkar. Su yüzeyin yükselişi gökyüzünün cisimlerin çekim kuvveti neden olmuştur. Bu görüngüye met (High Tide) denilir ve Su yüzeyin düşüşüne cezir (Low Tide) denilir. Medcezir su yüzeyinin yüksekliği ve düşüklüğü aralarındaki farklılığa (Tidal Range) denilir. Gökyüzü cisimlerinden (ay ve güneş) Newton'un kütle çekim

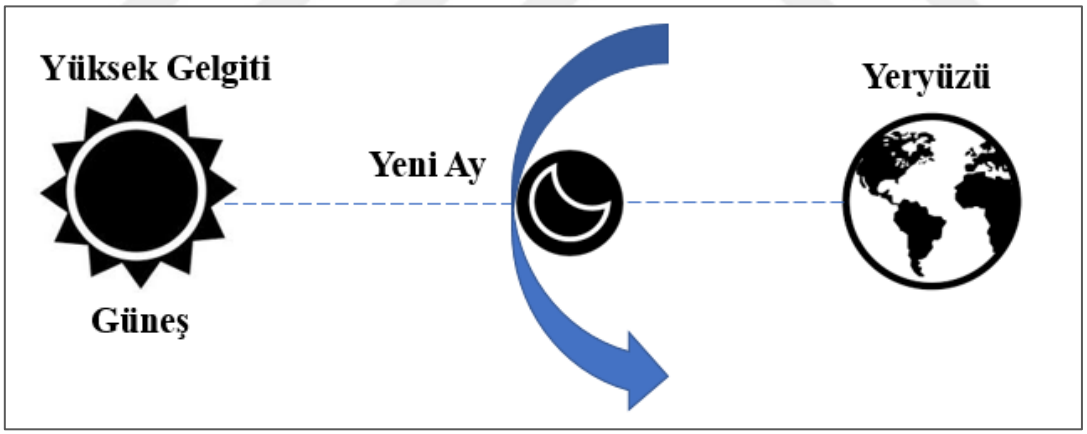
kuvvetin kurallara göre aya ait çekim kuvveti Güneşin çekim kuvvetinden çok daha yüksektir. Çoğunlukla gelgit oluşumuna katkıda bulunduğu sebep bu kuraldır.

Gelgitin zamanlaması günden güne değişir ve bunun nedeni Ay'ın devresi 24 saat düzenli çalışmamasıdır. Bunun yerine, ay her 24 saat ve 50 dakika içinde dünyayı dolaşır. Bu süre zarfında, gelgit iki kez meydana gelir ve sonuçta döngüsü 12 saat ve 25 dakikadan kısa bir süresi olur. Güneşin çekim kuvvetinin dalgası her 12 saatte bir defa ortaya çıkar ve onun aralığı çok düşüktür.

3.1.2. Gelgit Üzerindeki Ayın ve Güneşin Etkisini Birleştirmesi

3.1.2.1. Yüksek Gelgiti

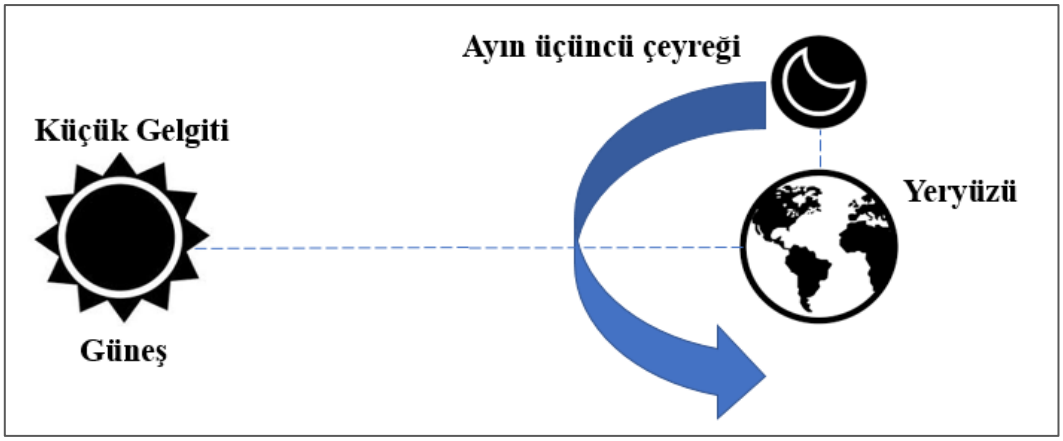
Yüksek gelgiti, olabildiğince Ayın, Güneş'in ve Dünya'nın birbirlerine doğru yönünde konumlaması ile ortaya çıkar. Bu durumda en çok gelgiti veya tam gelgit görünebilir. Başka bir değişle tam met en yüksek değeri ve tam cezir ise en küçük değeri sahip olacaktır. (Şekil 3.1)



Şekil 3.1. Yüksek Gelgitinin Durumu

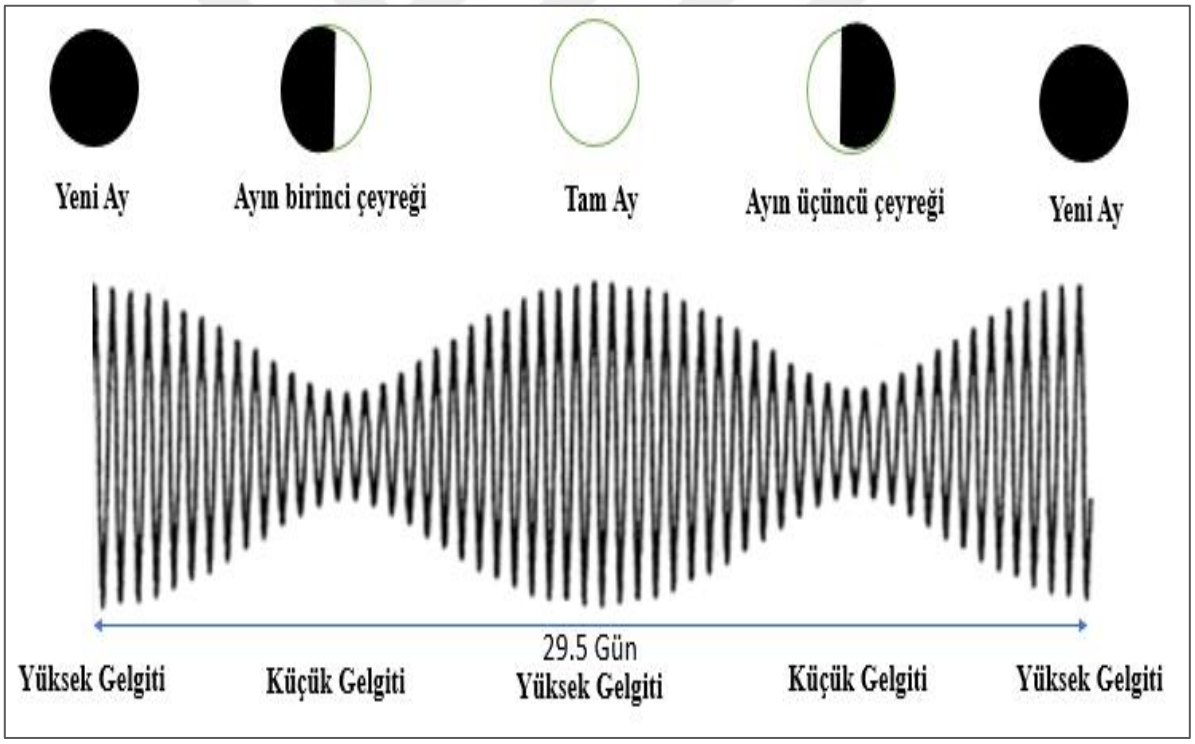
3.1.2.2. Küçük Gelgiti

Küçük gelgit, Güneş, Ay ve Dünya birlikte dikey bir açıyla oluştuğunda meydana gelir. Bu durumda tam met en yüksek değeri ve tam cezir ise en küçük değeri sahip olmayacaktır. (Şekil 3.2).



Şekil 3.2. Küçük Gelgitinin Durumu

Hesaplamalara göre, bir ay boyunca yüksek gelgit iki defa görülmektedir (Şekil 3.3).



Şekil 3.3. Kameri aya göre Gelgit aralığı değişiklikleri diyagramı

3.1.3. Ayın ve Güneşin Kuvvetlerinin Gelgit Oranı

Newton kütle çekim kuvveti kanuna göre, her zaman iki kütle birbirlerine çekim kuvveti uygularsa değer gücü iki kütle arası direkt bağıntısı olmakta ve iki kütlerin aralarındaki

uzaklığın karesi ile ters bağıntısıyla değer tanımlanmıştır. Dünya sürekli dönüyor ve hareket ediyor. Bir şekilde ayın ve güneşin çekim kuvvetin arasında kalıyor. Her zaman birbirlerine güç kuvveti uygulamaktadırlar. İki cisim birbirlerini çekim kuvvet ile çekerken ve aynı zamanda kendi merkez kütlesi dönmekte olduğundan birbirlerinden uzaklaşmaya başlarlar ve bu güce merkezkaç kuvveti denilir. Bu iki kuvvet, her biri kütlelinin merkezinde dengeyi kurmaktadır. Ancak, bu kuvvetler yeryüzünde dengeyi kuramıyorlar ve yeryüzündeki gelgitin ortaya çıkması ana nedeni olmaktadır.

Aşağıdaki ifadeye göre, güneşin yeryüzüne uyguladığı ayın kuvvetine göre 177 kat fazladır.

$$\frac{\text{Kürenin ve ayın güç kuvveti}}{\text{Kürenin ve güneşin güç kuvveti}} = \frac{\left(\frac{M/E}{D_m^2}\right)}{\left(\frac{S/E}{D_2^2}\right)} = \frac{1}{177} \tag{3.1}$$

Gelgitin oluşumunda, daha önce bahis etiğimiz şekilde ay güneşe göre daha etkilidir. Çünkü gelgitin oluşumunda F_a, F_b kuvvet değerlerin arasındaki farklılık (ayın kuvvetine göre) ve F_a', F_b' kuvvet değerlerin arasındaki farklılık (güneşin kuvvetine göre). Bu kuvvetlerin oranı aşağıdaki formülden elde edilir.

$$\frac{F'_a}{F_a} = \frac{F'_b}{F_b} = \frac{SD_m^3}{MD_s^3} = \frac{\frac{S}{E}}{\frac{M}{E}} \times \left(\frac{D_M}{D_S}\right)^3 = \frac{330000}{0.0123} \times \left(\frac{3.84\times10^5}{1.5\times10^8}\right)^3 = 0.46 \tag{3.2}$$

$$F_{a,b} = -\frac{2GMr}{D_m^2} \quad F'_{a,b} = -\frac{2GSr}{D_m^2}$$

$D_m = 3.84 \times 10^5 (km)$ Ayın kütle merkezi dünya arasındaki uzaklığı

$D_s = 1.5 \times 10^8 (km)$ Güneşin kütle merkezi dünya arasındaki uzaklığı

$\frac{S}{E} = 330000 (kg)$ Güneş ve dünya kütlesinin oranı

$\frac{M}{E} = 0.0123 (kg)$ Ayın ve dünya kütlesinin oranı

$G = 6.67 \times 10^{-11} (Nm^2/kg^2)$ Sabit evrensel yerçekimin ivmesi

$r = 6.37 \times 10^3 (km)$ Dünyanın yarıçapı

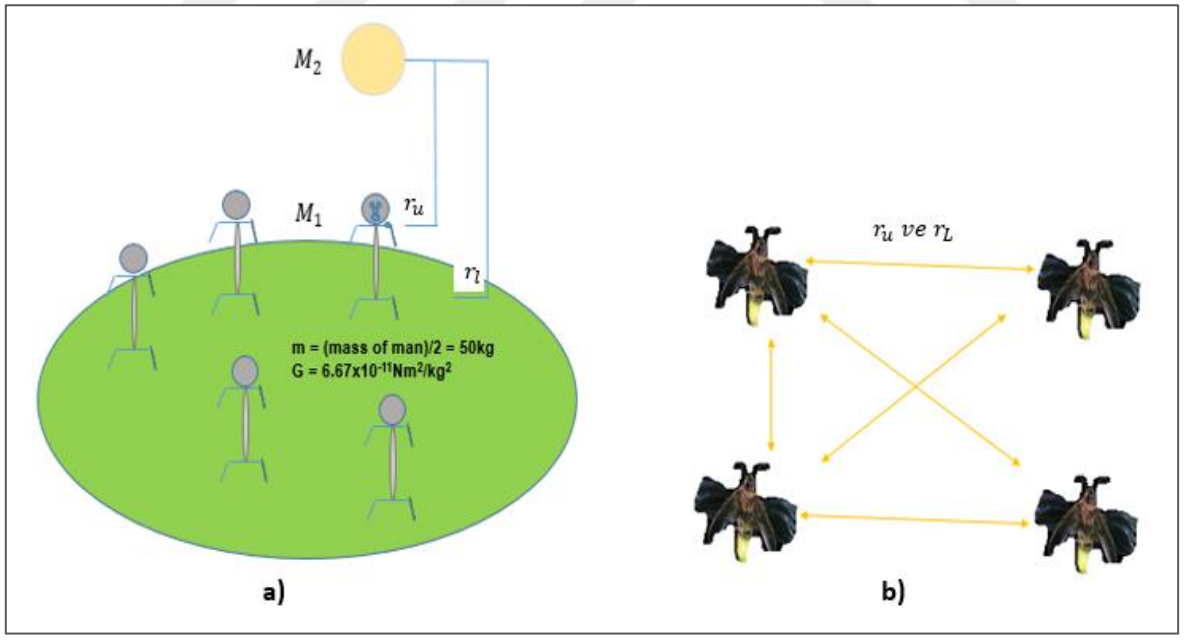
Dünya, güneş ve ayın kütlesi sıra ile $6\times 10^{24}, 1.963 \times 10^{30}$ ve $7.342 \times 10^{22} (kg)$ olmaktadır. Yukarıda belirtildiği gibi, gelgitin oluşumunda, Güneş tarafından uygulanan kuvveti gelgitin oluşumunda, aydan uygulanan kuvvetini dünya üzerinde %46'sı oluşmaktadır. Ayın ve güneşin

gelgit oranı için gösterilen ilişk-i göz önüne alındığında kütlenin ağırlığından daha çok uzaklık önemli etkenlerden birisidir[101-103].

3.2. Gelgitin Tanımı ve Uygulaması

Göksel mekanik alanında "Gelgit Kuvveti" ifadesi, bir cismin (ör. Su Gelgiti) ikinci bir kütlenin (ör. Dünya) ve üçüncü kütlenin (ör. Ay) çekim kuvveti etkisine bağlıdır[101]. Geçen bölümde bahsedilen, Newton'un evrensel çekim gücü kanunu, iki cismin birbirini çekmesi ile tanımlar ve büyüklüğü de kütlelerin çarpılmasıyla elde edilir ve cisimler arasındaki uzaklığın karesi ile ters orantılıdır. Gelgit Kuvvetler, yerçekimi kuvvetin bir sınıfıdır ve bunlar arasında sadece bir fark vardır. Yerçekimi çekme kuvveti, kütlelerin arasındaki uzaklığın karesi ile ters orantılıdır, ancak Gelgitte uzaklığın küpü ile ters orantılıdır[101-103].

Şekil 3.4'te (a) kısmında görüldüğü gibi, (M_2 = Ay) dünyaya yakın olduğu zaman gelgit olayı ortaya çıkmasına neden olur, burada küçük gelgit, yüksek gelgit ve güneşin etkisini göz ardı ediyoruz. Şekil 3.4 (b) gösterildiği gibi ateşböceklerinin parlak yoğunluğunun yerine Şekil 3.4 (a) gösterilen ayın ve dünyanın birbirlerini çekmesi ile Gelgit Kuvvetini kullandık.



Şekil 3.4. Önerilen model a) Gelgiti olayı b) Ateş böceğin davranışı

F: gelgit kuvveti, G: çekim kuvvet sabit değeri, M_1: Yeryüzü kütlesi ve M_2: ay kütlesi, r: uzaklık, Δd: ayın yarıçapı

$$F_{tidal}=\frac{GM_1M_2}{r^2}-\frac{GM_1M_2}{(r+\Delta d)^2} \tag{3.3}$$

$$F_{tidal}=\frac{GM_1M_2(r+\Delta d)^2-GM_1M_2r^2}{r^2(R+\Delta d)^2} \tag{3.4}$$

Düzenledikten ve basitleştirdikten sonra:

r_u: Yakın mesafe F_{near}: Yakın mesafedeki çekim kuvvetin büyüklüğü

r_l: Uzak mesafe F_{far}: Uzak mesafedeki çekim kuvvetin büyüklüğü

$M_1 = 0.1$ Küre Kütlesi

$M_2 = 0.1$ Ay Kütlesi

G = 0.667 Evrensel yerçekimi ivmesi

$$F_{tidal}=\frac{2GM_1M_2}{r^3}\equiv F_{near}-F_{far}\equiv(G*M_1*M_2/r_u^2)-(G*M_1*M_2/r_l^2) \tag{3.5}$$

$$r_l \neq r_u$$

Genellikle Gelgit Kuvveti formül 3.5 tanımlanır ve bu formülü yeni formül olarak Ateş böceğinin parlak yoğunluğun yerine kullanıldı. Ateş böceklerin çekiliş formülü yerine gelgit kuvvetin formülü kullanılmıştır. Burada her ateşböceğin arasında gelgit kuvveti hesaplanarak ve aralarındaki en iyi amaç fonksiyonu kalite değerine göre çekilmektedir. Bir sonraki bölümde ayrıntılı olarak açıklanacaktır.

3.3. Fatidal Algoritması

Ateş Böceği Algoritması çerçevesinde gelgit kuvveti kullanarak algoritma koşarken daha az nesil üretim ile küresel minimum değeri elde edebilmektedir. Birinci bölümde anlatıldığı gibi, r mesafesi $I \propto 1/r^2$ olarak azalırsa, ışık kaynağından çıkan ışık yoğunluğu artmaktadır. Burada ışık yoğunluğun yerine gelgit kuvveti kullanılmıştır.

$$A=\begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{bmatrix}$$

A: Ateş böceğin popülasyonu

m: maksimum satır ya da maksimum ateşböceği sayısı

n: maksimum sütun ya da maksimum ateşböceğin değişken sayısı

$x_{m,n}$: Değişkenlerin ışık yoğunlukları

d: Bir vektörün değişken sayısı(boyut sayısı)

Ateş böceği $x = (x_{11}, \dots, x_{1d})$ olmak üzere ve n Ateş böcek (bireysel) sayısıdır. Her bir döngü de Ateş böcekleri ikili şeklinde hesaplama yaparlar. Eğer çiftin birisi diğerine göre değeri düşükse her boyutta yeni hareket etmede bulunmalıdır.

Gelgit Kuvvetlerinin daha az boyutunu hesaplamak için, çiftli ateşböceklerinin her n değişkende m sütunları arasındaki (Ör, x_{11},x_{21}) mesafe hesaplanır, öte yandan daha çok boyutlu problem ile karşılaşırsak, zaman problemin önüne geçmek için ateş böceğin her boyutuna bakmadan konumladığı yerden uzaklığı hesaplanır (Ör, x_1,x_2).

Önerilen stratejide kaç iz düşümden uluşmaktadır:

3.3.1. Ateş Böceklerin Gelgit İzdüşümü

Önerilen algoritmada az boyut ve çok boyutta farklı bakış ve izdüşümü yapılmıştır. Az boyutta Ateşböceğin vektör değişkenlerini baz alarak iz düşümü yapılmış ve çok boyutta ise Ateşböceğin kendisini (vektör) baz alınmıştır. İzdüşümü tanımı aşağıda açıklanmıştır.

X_{max}: Maximum aralık değeri

X_{min}: Minimum aralık değeri

Ateş böceği Gelgit küvetine izdüşümü

En= (M_1*M_2* G) / ((1/ (1+(X_{max}- X_{min}))) ^2) Maximum Değeri

St= (M_1*M_2* G) / (1^2) Minimum Değeri

DiffGelgit=En-St gelgit kuvvet hesaplaması (3.6)

3.3.2. İki Ateş Böceğin Uzaklık İzdüşümü

Gelgit parametrelerine göre Ateş böceklerin uzaklık hesaplanmaktadır. N: Ateş böceğin toplam sayısı, d: Toplam değişken sayı

i=1..N, j=1..N, k=1..d i ≠j,

$FADis=\sqrt{(X_{max}-X_{min})^2}$ (Ateş böceğin minimum ve maksimum aralığın uzaklığı)

$RDis_{i,j}=\sqrt{\sum_{k=1}^{d}(X_{i,k}-X_{j,k})^2}$ (Çok boyutta ateş böceklerin uzaklığı)

$RDis_{i,j,k}=\sqrt{(X_{i,k}-X_{j,k})^2}$ (Az boyutta ateş böceklerin uzaklığı) (3.7)

İzdüşümün Minimum ve Maksimum değeri: $S_{min}=0, S_{max}=DiffGelgit$

İki Ateş böceğin uzaklık gelgiti izdüşümün hesaplanması:

$TIDis=(RDis_{i,j}-S_{min})/(FADis-S_{min})*(S_{max}-S_{min})+S_{min}$ (3.8)

$r_i=0$, $r_j=TIDis$

3.3.3. İki Ateş Böceğin Gelgit Olayın Hesaplanması:

Burada Gelgit formülünü burada hesaplanarak Ateşböceğin çekiliş değeri elde etmektedir.

$r_i=\frac{1}{1+r_i}$, $r_j=\frac{1}{1+r_j}$

$r_l=Min\ (r_i,r_j)$

$r_u=Max\ (r_i,r_j)$ (3.9)

Şekil 3.4 (a) Gösterildiği gibi insan eli (r_u) ve ayağı (r_l) olmaktadır.

$F_{far}=M_1*M_2*G/r_l^2$

$F_{near}=M_1*M_2*G/r_u^2$

$Ftidal=|F_{near}-F_{far}|$

$Ftidaldis=TIDis-Ftidal$ uzaydan aldığı gerilim değeri (3.10)

C: sabit değeri, 0.1 ve 0.4 aralığa bulunabilir. $B_{i,j}$: gelgit küvetin çekiliş değeri

$B_{i,j}=(Ftidaldis-S_{min})/(TIDis-S_{min})*(C-(S_{min}))+(S_{min})$ parametreli

$$B_{i,j}=(Ftidal - S_{min}) /(TIDis - S_{min}) *(diskol- (S_{min})) + (S_{min}) \text{ parametre sız} \quad (3.11)$$

Özetçe söylenirse:

1. Önerilen algoritmada Ateş böceğin çerçevesini kullanıldı ve ışık kavramını göz ardı edilmiştir.
2. İki Ateş böceğin aralarındaki etkisini gelgit kuvveti ile hesaplanmaktadır.
3. Hesaplanmış olan gelgit kuvvetin gerilimi adım büyüklük değeri olarak tanımlanır ve bir Ateş böceğin amaç fonksiyonun kalitesi az olan diğer fazlan olana hareket miktarıdır.

Özet olarak, Ateş Böceklerinin yanıp sönmesinin yerine, Gelgitin kuvveti amaç fonksiyon olarak formüle edilebilir. Ateşböceği hareket yönü için, elde edilen gelgit miktarlar ile cismi gerilmeye yönelik ve koordine etmek için daha az boyut ve çok boyutta rastgele monoton dağılımı kullanılmıştır. Ateşböceği hareket hızı M_1, M_2ve G ile kontrol edilir ve ayrıca uygunluk değere çarpılabilir. Önerilen stratejide bunlar sadece sabit değerdir. Küresel arama/ farklılaştırmak için algoritma mümkün olan tüm bölgelerde minimum değeri aramayı deniyor; yani Ateş böceği her aşamada normal dağılım ve genel Alpha operatörü kombinasyonuyla en iyi Ateş böceğe doğru ilerliyor demektir. Ateşböceği algoritması ve varyantları dahil olmak üzere, çok boyutlu problemlerde, iyi farklılaştırmaya yönelik yeni bir Alfa operatörü tanımlamışlar. Ama önerilen algoritma da genel Alfa operatörü, çok boyutlu problemlerde yeni operatörü ihtiyaç duyulmadan kullanılmaktadır.

t: döngü sayısı

a: sabit değerdir ve 0.2 ve 0.9 arası olarak tanımlanabilir ve her döngüde indirgemektedir.

w: bastırma değer 0.9 ve 0.99 arası olarak tanımlanabilir.

$$a_t=a_{t-1}*w \quad (3.12)$$

Küresel arama, yakınsama oranını azaltma eğilimindedir. Alpha ayrıca bastırma miktar ile kullanılmaktadır. En uygun değer bastırmak için 0.99 seçilebilir. Algoritmada Genel olarak, bölgesel arama ve küresel arama ile aralarında bir dengeyi kurmak için, aşağıdaki gösterildiği gibi ikiye bölünür.

$x_i=(x_{i1},x_{i2},\ldots, x_{id})$ d boyut sayısı, x ateş böceği

$$\begin{pmatrix} \text{ikili Ateş Böceklerin çalıştırılması } d>10 \\ \text{ikili Ateş Böceklerin değişkenlerinin çalıştırılması } d\leq 10 \end{pmatrix} \quad (3.13)$$

$$x_{ij}=x_i+B_{i,j}*(x_i - x_j) +a_t*(\text{rand}-\frac{1}{2}) \quad (3.14)$$

Önerilen algoritmada ateşböceğin hareketi (3.14) formüle ve (3.13) şartlarına göre dayanmaktadır. 3.13 deki şart, Ateş böceğin Algoritmasının zamanını düşürmek için yapılmıştır. Gelgit Kuvvetinin özelliklerinden biri, mesafeye bağlı olması ve M_1 ve M_2 miktarını Alfanın bastırma ile uyarak bölgesel aramayı daha fazla yoğunlaşmak için yapmaktadır. Gelgit Kuvvetinin diğer bir özelliği, ateşböceğin stratejisine uyum sağlamasıdır. İkili Ateş böceklerin değişkenlerini çalıştırma stratejisinde başlangıça dayanan rastgele sayıları en iyi şeklilde yönlendirmek için Opposition based learning (OBL) yöntemi kullanılmıştır. Bu yöntem Tizhoosh [104] tarafından tanımlanmış. $x \in [m,n]$'de bulunan x gerçek sayı olarak varsayalım. Karşıt sayı şu şekildedir:

$$\tilde{x}=m+n-x \quad (3.15)$$

D-boyutlu uzayda formül şu şekilde olabilir:

$$\tilde{x}_i=m_i+n_i-x_i \quad i=1\ldots,D. \quad (3.16)$$

```
1- Amaç Fonksiyonu f(x), x=(x_1, ..., x_d)^T
2- Başlangıçta ki Ateş Böcek popülasyonunu oluştur x_i(i = 1,2, ..., n)
3- Ateşböceklerinin popülasyon karşıt yönünü hesaplayın x̃_i (2.13, 2.14)
4- x̃_i ve x Amaç fonksiyonunu hesaplayın.
5- x̃_i ve x  Popülasyon sayısına göre açgözlü seç.
6-      G, m_1, m_2, Alpha, MaxGeneration, nVar tanımla.
7-      While (t<MaxGeneration)
8-          For i=1:n
9-            For j=1:n
10-               If (I_i ≠ I_j)
11-                 If (I_i < I_j), i Ateş böceği j'ye hareket eder; end if
12-                    For k=1: nVar
13-                    Ön hesaplama (3.6-13)
14-                    Yeni çözüm üretilmesi (3.14) ve güncelleme işlemi
15-                    End for k
16-               end if
17-            end for j
18-          end for i
19-          Ateş böcekleri sırala ve en iyisini g_* ata.
20-     End While
```

Şekil 3.5 Fatidal algoritması

Opposition based learning yöntemde f (.) Amaç fonksiyonu, x aday çözümü ve $\tilde{x}$ aday karşıt çözümü d boyutlu uzayda olmaktadırlar. Eğer f($\tilde{x}$)≥ f(x), x yerine $\tilde{x}$ geçecektir. Şekil 3.5'te algoritmanın sözde kodu.

4. İRDELEME

4.1. Giriş

Modifiye yapılan Ateş böceği algoritmalar ve ABC algoritması az ve çok boyutlarda Önerilen FAtidal algoritma ile karşılaştırdı. Sonuçta denemelerde istatistiksel olarak değerlendirilir. Bununla birlikte, diğer sınıflandırılmış algoritmalar (sürü zekâsı ve evrimsel algoritmalar) için, sadece iki boyutta performansları gösterilmiş ve istatistiksel testi yapılmamış, çünkü tez kapsamında bahis ettiğimiz algoritmalar farklı sınıfa ait olduğundan (farklı çerçeveye sahip olan) dolayı değerlendirme yapılmamıştır. Örnek olarak BBO algoritması biyo-ilham sınıfından ve HS algoritması fizik ve kimya sınıfından sınıflandırılırmış [105].

4.1.1. Başlangıç Ayarları

Tüm karşılaştırılan algoritmaların en uygun şekilde parametreleri ayarlandı. Her denemede, 30 defa koşturulmuştur. Her bir koşmada Amaç fonksiyonu değerlendirme sayısına (FEN) kadar gerçekleştirilmiş. Bu deneyde, iki boyut için FEN 100.000, on ve yüz boyut için ise 200.000 ve 300.000 olarak ayarlandı. Popülasyon büyüklüğü sırası ile 20, 40 ve 60 olarak belirlenmiştir.

- nPop, popülasyon büyüklüğüne işaret eder.
- n, kelimelerin baş harfi genellikle sayıyı belirtir.
- Co, Katsayıya karşılık gelir.
- Mu, Mutasyona temsil eder.
- GA: Mu probability pm = 0.05, crossover probability pc = 0.8, gene length = 10.
- ABC: nOnlooker=nPop*1/2, Abandonment Limit, Parameter=round(0.5*nVar*nPop) Abandonment Limit Parameter, colony size=20, Acceleration Coefficient Upper Boun=1.
- BBO: KeepRate=0.2, nKeep=round(KeepRate*nPop), nNew=nPop-nKeep, alpha=0.9, pMutation=0.1, sigma=0.02*(VarMax-VarMin).
- BEE: Number of Selected Sites =round(0.5*nScoutBee), nSelected Elite Sites =round(0.4*nSelectedSite), nRecruited Bees for Selected Sites= round(0.5*nScoutBee), nRecruited Bees for Elite Sites=nEliteSiteBee=2*nSelectedSiteBee, Neighborhood Radius=0.1*(VarMax-VarMin), rdamp=0.99.

- CU: Acceptance Ratio =0.35, Number of Accepted =round(pAccept*nPop), Accepted Individuals, alpha=0.25, beta=0.5.
- HS: Harmony Memory Size =20, nNewHarmonies =20, Harmony Memory Consideration Rate =0.5, Pitch Adjustment Rate=0.1, Fret Width (Bandwidth)=0.02*(VarMax-VarMin), Fret Width Damp Ratio=0.995.
- ICA: Number of (Empires/Imperialists) =10, Selection Pressure=1, Assimilation Coefficient =2, Revolution Probability =0.1, Revolution Rate =0.05, Colonies Mean Cost Coefficient Share Settings =0.1.
- FA: $\beta 0 = \gamma = 1$, $\alpha = 0.02$, a_damp=0.97.
- LFA: $\gamma = 1$, $\gamma 0 = 4$, $\eta 1 = 0.3$, $\eta 2 = 0.1$.
- MFA: $\alpha = 0.5$, betamin=0.2, $\gamma 0 = 1$.
- NMSA_FFA: $\alpha = 0.02$, $\gamma 0 = 1$, betamin=0.2.
- NaFA: $\alpha = 0.5$, $\beta 0 = \gamma = 1$.
- Fatidal: Uniform MuRange =0.05*(VarMax-VarMin), $\alpha = 0.9$, a_damp=0.99, m1=0.1, m2=0.1, G=0.667.

4.1.2. Kıyaslama Fonksiyonlar

Tablo 4.1 (1.Grup) [106] ve Tablo 4.2 (2.Grup - CEC 2005), farklı özelliklere sahip olan fonksiyonlar deneme için 21 Kıyaslama fonksiyonu dikkate alınmıştır.

Tablo 4.1 Dört fonksiyon sınıfı Many Local Minima, Bowl Shaped, Plate Shaped ve Steep Ridges ve Tablo 4.2 de ise Unimodal, Multimodal and expanded functions bulunmaktadır.

Tablo 4.1. Kıyaslama Fonksiyonu [106].

Fonksiyon/ Özellikler	Denklem	x_i aralığı	Küresel minimum
Many Local Minima/Multimodal			
Schaffer	$f_1(x)=\frac{\sin^2\sqrt{\sum_{i=1}^{D} x_i^2}-0.5}{\left[1+0.001\left(\sum_{i=1}^{D} x^2\right)\right]^2}+0.5$	[-100, 100]	0
Ackley/ nearly flat outer region and a large hole at the center	$f_2(x)=20-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right)-\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right)+e$	[-32.768, 32.768]	0
Schawefel	$f_3(x)=418.9829d-\sum_{i=1}^{d} x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	[-500, 500]	0

Tablo 4.1'in devamı

Langerman/ unevenly distributed local minima(A and C for 2 dimensions was available in this study)	$f_4(x)=-\sum_{i=1}^{m} c_i \exp\left(-\frac{1}{\pi}\sum_{j=1}^{d}(x_j-A_{ij})^2\right) *\cos\left(\pi\sum_{j=1}^{d}(x_j-A_{ij})^2\right)zz$	[0, 10]	-5.162125
Levy	$f_5(x)=\sin^2(\pi w_1)-\sum_{i=1}^{d-1}(w_i-1)^2\left[1+10\sin^2(\pi w_1+1)\right]+(w_d-1)^2\left[1+\sin^2(2\pi w_d)\right]$, where $w_i=1+\frac{x_i-1}{4}$, for all i=1,…,d	[-10,10]	0
Rastrigin/ locations of the minima are regularly distributed	$f_6(x)=\sum_{i=1}^{D}\left[x_i^2-10\cos(2\pi x_i)\right]$	[-5.12,5.12]	0
Bowl Shaped			
Sphere/has d local minima except for the global one, continuous, convex	$f_7(x)=\sum_{i=1}^{D} x_i^2$	[-5.12,5.12]	0
Perm Function 0, d, Beta	$f_8(x)=\sum_{i=1}^{D}\left(\sum_{j=1}^{d}(j+\beta)\left(x_j^i-\frac{1}{j^i}\right)\right)^2$	[-d, d]	0
Rotated Hyper-Ellipsoid/continuous, convex	$f_9(x)=\sum_{i=1}^{d}\sum_{j=1}^{i} x_j^2$	[-65.536, 65.536]	0
Sum of Different Powers	$f_{10}(x)=\sum_{i=1}^{d}\lvert x_i\rvert^{i+1}$	[-1, 1]	0

Tablo 4.1'in devamı

Sum Squares/ continuous, convex	$f_{11}(x)=\sum_{i=1}^{d} ix_i^2$	[-10, 10]	0
Plate Shaped			
Zakharov/ has no local minima except the global one	$f_{12}(x)=\sum_{i=1}^{d} x_i^2+\left(\sum_{i=1}^{d} 0.5ix_i\right)^2+\left(\sum_{i=1}^{d} 0.5ix_i\right)^4$	[-5, 10]	0
Dixon-Price	$f_{13}(x)=(x_1-1)^2+\sum_{i=2}^{d} i\left(2x_i^2-x_{i-1}\right)^2$	[-10, 10]	0
Rosenbrock / global minimum lies in a narrow, parabolic Valley	$f_{14}(x)=\sum_{i=1}^{D-1}\left[100\left(x_{i+1}-x_i^2\right)^2+(x_i-1)^2\right]$	[-5, 10]	0
Steep Ridges			
Michalewicz/ has d! local minima	$f_{15}(x)=-\sum_{i=1}^{d} sin(x_i)\, sin^{2m}\left(\frac{ix_i^2}{\pi}\right)$	[0, π]	-1.8013(d=2) -9.6601(d=10) -99.6201 (d=100)

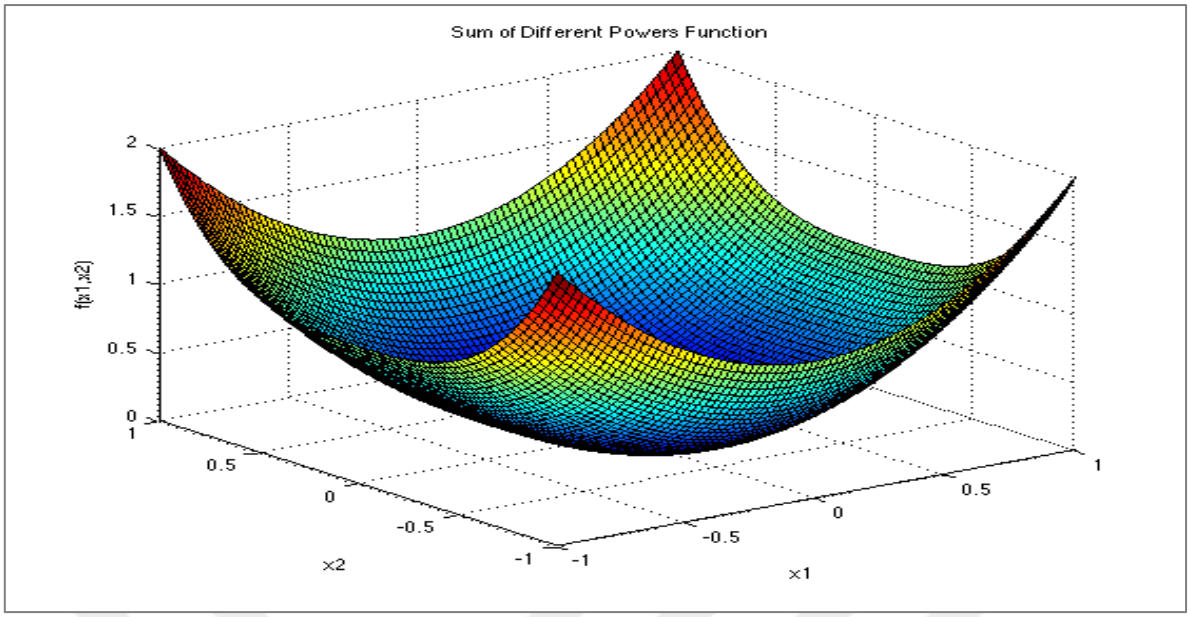
Table 4.2. CEC Kıyaslama Fonksiyonlar

Fonksiyon/ Özellikler	Denklem	x_i aralığı	Küresel minimum
Unimodal			
Shifted Sphere Function/ separable, scalable	$f_{16}(x)=\sum_{i=1}^{d} z_i^2+f_{bias_1}$, $z=x-o, x=[x_1,x_2,\ldots,x_D]$	$[-100,100]$	-450
Shifted Schwefel's Problem 1.2/non-separable, scalable	$f_{17}(x)=\sum_{i}^{d}\left(\sum_{j=1}^{i} z_j\right)+f_{bias_1}$, $z=x-o, x=[x_1,x_2,\ldots,x_D]$	$[-100,100]$	-450
Multimodal			

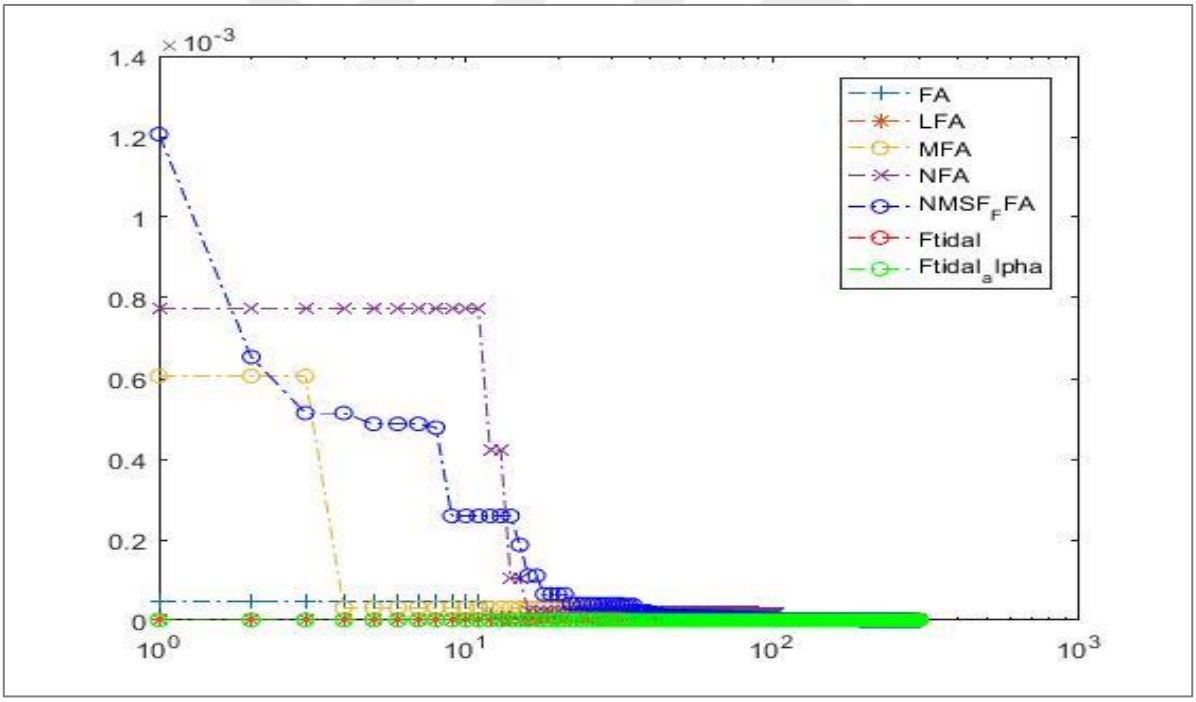
Tablo 4.2'in devamı

Shifted Rosenbrock/ non-separable, scalable	$f_{18}(x)=\sum_{i=1}^{d-1}(100(z_i^2-z_{i+1})^2+(z_i-1)^2)+f_{bias_1}$, $z=(x-o)+1, x=[x_1,x_2,\ldots$	$[-100,100]$	390
Shifted Rotated Griewank/ non-separable, scalable and no bound for variable x	$f_{19}(x)=\sum_{i=1}^{d}\frac{z_i^2}{4000}-\prod_{i=1}^{d}\cos(\frac{z_i}{\sqrt{i}})+f_{bias_1}$, $z=(x-o)*M, x=[x_1,x_2,\ldots,x_D]$	[0, 600]	-180
Shifted Rotated Ackley non-separable, scalable and global optimum on the optimum	$f_{20}(x)=-20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d}z_i^2}\right)-\exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi z_i)+20+e\right)+f_{bias_1}$, $z=(x-o)*M, x=[x_1,x_2,\ldots,x_D]$	[-32,32]	-140
Expanded Functions			
Expanded Rotated Extended Schaffer/ non-separable, scalable	$f_{21}(x,y)=0.5\frac{\left(\sin^2\left(\sqrt{x^2+y^2}\right)-0.5\right)}{\left(1+0.001(x^2+y^2)\right)^2}$	$[-100,100]$	-300

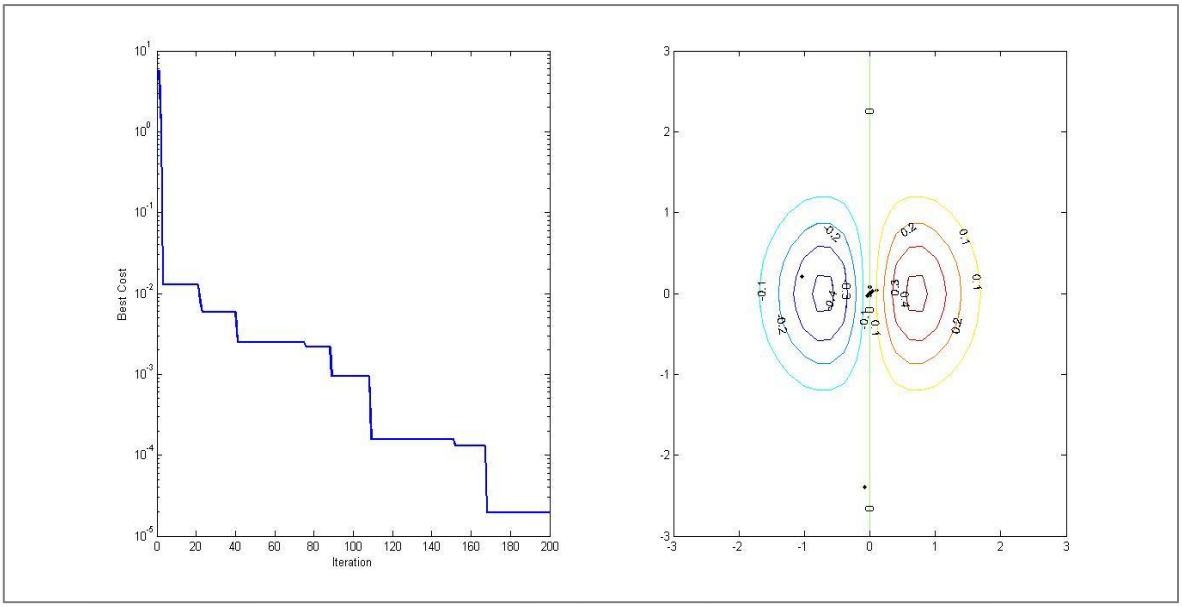
Şekil 4.1 gösterildiği gibi örnek olarak Sum of Different Power Fonksiyonun üzerine algoritmaları çalıştırılmış ve Şekil 4.2 ise iki boyutta algoritmaların karşılaştırması verilmiştir. Şekil 4.3 (a) Sphere fonksiyonu üzerinde Fatidal'ın gelişimi ve Şekil 4.3 (a) ise eşyükselti uzayı gösterilmektedir.



Şekil 4.1. Sum of different Power fonksiyonu [106].



Şekil 4.2. Sum of different power Fonksiyonun üzerine iki boyutta yakınsama eğrimi



Şekil 4.3. Sphere fonksiyonun üzerinde Fatidal çalışması a) Hareketin diverjansı
b) Eş yükselti uzayı

4.1.3. Algoritmaların Karşılaştırması

Önerilen algoritmayı uygun şeklinde ispatlamak amacı ile algoritmaların yakınsaklığını, tutarlığını ve gürbüzlüğünü karşılaştırmalarını istatistiksel olarak sonuçlarla gösterilmiştir, ilaveten Wilcoxon Signed Rank testi de yapılmıştır.

Testte, p'nin signifikans değeri olmak üzere ve h'nin, tanımlanmış olan hipotezin reddedilip reddedilmediğine dayanan mantıksal değeridir. Bu metot çiftli bir yöntem olarak kullanılır. Başka bir değişle bu yöntem örnek olarak iki algoritmanın elde ettiği sonuçların üzerinde test etmektedir. Test, a = 0.05 (%95) düzeyinde olan hipotez ile yapılır

H_0:$\mu_1 = \mu_2$ İki veri setler statiksel olarak farklı değiller.

H_1: $\mu_1 \neq \mu_2$ İki veri setler statiksel olarak farklıdırlar.

Karşılaştıran algoritmalar, kendi ayarlana göre 30 kez koşturulmuş ve Wilcoxon Signed Rank testi ile karşılaştırılmış ve sonuçları Tablolar 4.5, 4.6, 4.9, 4.10, 4.13 ve 4.14'de gösterilmektedir. Bold olan değerleri önerilen algoritma diğer algoritmadan daha iyi performans gösterdiğini ve altı çizili değerleri ise önemli bir fark olmadığını gösterir.

Tablolar 4.3, 4.7, 4.11 ve 4.4, 4.8, 4.12, sıra ile Tablolar 4.1 ve 4.2'nin kıyaslama fonksiyonuna karşılık gelir. Tablo 4.3, 4.4 ve 4.7, 4.8 ve 4.11, 4.12 sıra ile 2, 10 ve 100 boyutlu problem uzayında elde edilen değerleri göstermektedir. Tablolarda gösterildiği gibi, bold olan değerler, dikkate alınan algoritmalar tarafından elde edilen en iyi veya en minimum değerlerdir.

Tablodaki istatistiksel verileri tanımlamak için 4 terim bulunmaktadır. m, M, A ve V sıra ile Maksimum, Minimum, Ortalama ve Sapma değerine karşılık gelmektedir.

Tablo 4.3, Birinci deneyde (2 boyutlu) FA ve FAtidal diğerlerine göre daha iyi yakınsama sahip olduğunu göstermektedir. Fakat istatistiksel testte, FAtidal'in FA'dan daha dayanaklı olduğunu ispatlamış. Tablo 4.4'de ise NAFA daha iyi bir yakınsama ve istatistiksel test en iyi performansı göstermektedir (Tablo 4.3, 4.4, 4.5, 4.6).

Tablo 4.7, 4.8, On boyut için yapılan ikinci deneyde FAtidal minimum değeri ve istatistiksel testlerin sonucu diğerlerine göre daha iyi performans göstermektedir (Tablo 4.7, 4.8, 4.9, 4.10).

Tablo 4.11, yüz boyut için üçüncü deneyde, FAtidal ve NAFA algoritmaları en iyi minimum değeri elde etmişler ancak istatistiksel testte sadece MFA, FAtidal‘den daha iyi sonuç göstermektedir (Tablo 4.11, 4.12, 4.13, 4.14).

ABC algoritması ile karşılaştırıldığında, önerilen algoritma en iyi sonuç vermektedir. (Tablo 4.15-4.20 Bolt olan değerleri Fatidal tarafından domine olmuş demektir.). Genel olarak, tüm deneylerde önerilen algoritma (2-, 10- ve 100- boyut uzaylarda), tüm fonksiyonlarda küresel minimum bulunmasında ve dayanaklı davranış özelliğini göstermektedir.

Tablo 4.3. İki boyutta birinci grup fonksiyonlara bağlı karşılaştırma

Algorithms	Many Local Minima						bowl_shaped					Plate_shaped			Steep Ridges
	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15
BBO(m)	0.00E+00	2.22E-14	2.55E-05	-5.16E+00	6.43E-32	0.00E+00	1.22E-31	9.81E-16	2.75E-31	3.35E-34	8.28E-30	6.82E-27	1.04E-16	9.53E-10	-1.80E+00
BBO(M)	0.00E+00	4.00E-10	2.37E+02	-1.60E+00	1.16E+00	3.98E+00	7.12E-22	2.50E-09	1.05E-20	3.63E-18	3.64E-21	1.22E-20	1.23E-13	2.06E+00	-1.21E+00
BBO(A)	0.00E+00	1.78E-11	4.74E+01	-4.19E+00	3.87E-02	1.23E+00	2.58E-23	2.56E-10	4.12E-22	1.22E-19	2.40E-22	1.13E-21	1.93E-14	8.85E-02	-1.78E+00
BBO(V)	0.00E+00	5.27E-21	4.45E+03	1.38E+00	4.49E-02	7.29E-01	1.68E-44	2.89E-19	3.66E-42	4.40E-37	6.82E-43	7.77E-42	7.92E-28	1.48E-01	1.15E-02
BEE(m)	4.35E-14	7.73E-07	2.55E-05	-5.16E+00	8.07E-14	1.02E-12	1.50E-14	3.85E-08	1.01E-13	1.35E-18	9.98E-15	1.49E-13	1.34E-12	1.99E-05	-1.80E+00
BEE(M)	1.47E-03	3.53E-05	2.55E-05	-5.16E+00	2.54E-12	1.63E-09	3.11E-12	2.73E-07	4.78E-10	2.00E-14	9.05E-12	1.59E-11	1.65E-09	3.46E-02	-1.80E+00
BEE(A)	2.87E-04	1.16E-05	2.55E-05	-5.16E+00	7.34E-13	3.03E-10	5.71E-13	1.39E-07	8.90E-11	2.21E-15	2.37E-12	1.77E-12	3.65E-10	6.97E-03	-1.80E+00
BEE(V)	2.02E-07	6.60E-11	9.64E-19	8.32E-23	5.10E-25	1.98E-19	5.75E-25	4.50E-15	8.87E-21	1.73E-29	5.51E-24	8.75E-24	1.65E-19	8.88E-05	2.75E-24
CU(m)	0.00E+00	8.88E-16	2.55E-05	-5.16E+00	1.50E-32	0.00E+00	1.34E-119	5.16E-17	5.08E-123	1.21E-52	3.57E-222	1.15E-121	3.70E-32	3.85E-05	-1.80E+00
CU(M)	2.14E-02	5.94E+00	3.36E+02	-2.21E+00	2.37E-06	3.98E+00	4.81E-06	3.80E-01	4.58E-07	9.45E-12	1.05E-06	2.45E-05	1.27E-02	1.18E+01	-1.00E+00
CU(A)	1.20E-03	2.42E-01	1.14E+02	-4.30E+00	8.29E-08	7.96E-01	1.61E-07	4.33E-02	1.54E-08	4.03E-13	3.93E-08	8.25E-07	7.21E-04	1.22E+00	-1.77E+00
CU(V)	1.91E-05	1.22E+00	9.99E+03	1.40E+00	1.87E-13	7.10E-01	7.72E-13	8.65E-03	7.00E-15	3.02E-24	3.67E-14	2.00E-11	7.49E-06	9.46E+00	2.14E-02
GA(m)	0.00E+00	5.37E-11	2.55E-05	-5.16E+00	4.26E-28	0.00E+00	1.30E-24	2.35E-14	1.64E-21	7.64E-29	1.21E-27	1.60E-19	1.38E-23	1.54E-07	-1.80E+00
GA(M)	2.64E-03	4.71E-03	1.18E+02	-2.95E+00	1.95E-07	1.23E-05	1.45E-07	1.32E-02	1.10E-05	7.92E-09	9.10E-08	9.17E-07	1.13E-04	6.91E-01	-1.80E+00
GA(A)	3.86E-04	4.17E-04	3.55E+01	-4.29E+00	1.07E-08	1.13E-06	7.00E-09	2.81E-03	8.15E-07	3.57E-10	1.02E-08	6.18E-08	1.67E-05	1.32E-01	-1.80E+00
GA(V)	5.26E-07	1.05E-06	3.05E+03	1.05E+00	1.33E-15	7.38E-12	7.17E-16	1.55E-05	7.41E-12	2.22E-18	5.49E-16	2.74E-14	5.72E-10	2.51E-02	9.44E-16
HS(m)	0.00E+00	9.33E-14	2.55E-05	-5.16E+00	3.39E-29	0.00E+00	6.23E-29	3.70E-22	2.49E-26	3.03E-33	6.77E-29	3.82E-28	1.30E-26	5.31E-06	-1.80E+00
HS(M)	5.33E-03	5.07E-12	2.55E-05	-5.16E+00	1.18E-25	0.00E+00	3.00E-26	1.85E-03	7.27E-24	2.97E-28	1.31E-25	1.32E-25	1.35E-23	1.36E-01	-1.80E+00
HS(A)	7.57E-04	1.09E-12	2.55E-05	-5.16E+00	1.10E-26	0.00E+00	5.08E-27	8.43E-05	1.02E-24	4.62E-29	2.35E-26	2.55E-26	4.23E-24	1.17E-02	-1.80E+00
HS(V)	1.86E-06	9.09E-25	0.00E+00	3.26E-30	6.46E-52	0.00E+00	4.29E-53	1.24E-07	1.91E-48	4.50E-57	7.33E-52	1.07E-51	1.51E-47	6.27E-04	4.59E-31
ICA(m)	0.00E+00	8.88E-16	2.55E-05	-5.16E+00	1.50E-32	0.00E+00	2.55E-60	7.89E-31	1.52E-54	1.24E-66	3.64E-55	4.09E-51	3.70E-32	6.15E-15	-1.80E+00
ICA(M)	0.00E+00	4.44E-15	1.18E+02	-3.00E+00	1.50E-32	0.00E+00	5.22E-45	3.48E-23	4.87E-45	3.20E-52	1.15E-41	7.89E-40	5.03E-30	2.11E-07	-1.80E+00
ICA(A)	0.00E+00	1.24E-15	4.74E+01	-4.15E+00	1.50E-32	0.00E+00	4.57E-46	2.93E-24	1.83E-46	1.07E-53	3.85E-43	2.97E-41	3.98E-31	1.55E-08	-1.80E+00
ICA(V)	0.00E+00	1.18E-30	3.48E+03	1.20E+00	7.75E-96	0.00E+00	1.71E-90	6.61E-47	7.90E-91	3.42E-105	4.42E-84	2.08E-80	9.23E-61	2.41E-15	4.59E-31
FA(m)	6.54E-03	8.88E-16	1.72E-01	-5.16E+00	1.50E-32	0.00E+00	7.18E-130	0.00E+00	7.91E-126	2.08E-22	4.53E-129	3.35E-129	3.70E-32	0.00E+00	-1.80E+00
FA(M)	2.39E-01	4.14E+00	1.21E+02	-3.20E+00	1.50E-32	1.99E+00	1.65E-127	7.89E-31	9.55E+00	4.23E-14	2.17E-127	2.05E-127	4.93E-32	7.01E-02	-1.80E+00
FA(A)	7.37E-02	1.40E+00	1.79E+01	-5.05E+00	1.50E-32	5.15E-01	2.61E-128	1.31E-31	2.13E+00	1.53E-15	9.90E-128	7.07E-128	3.86E-32	4.87E-03	-1.80E+00
FA(V)	2.88E-03	2.25E+00	7.01E+02	1.48E-01	7.75E-96	4.15E-01	1.17E-255	8.94E-62	6.22E+00	5.94E-29	3.50E-255	2.88E-255	1.82E-65	2.46E-04	4.59E-31
LFA(m)	8.71E-06	6.36E-09	8.34E+01	-5.16E+00	4.97E-09	3.31E-09	8.92E-19	3.72E-06	8.74E-21	4.27E-11	1.05E-14	1.90E-12	8.87E-31	5.31E-04	-1.80E+00
LFA(M)	2.96E-01	9.02E+00	4.83E+02	-1.08E+00	9.28E-01	3.98E+00	3.41E-01	4.30E+01	1.97E+02	1.27E-02	2.80E+00	4.01E+00	3.24E+00	8.88E+00	-1.00E+00
LFA(A)	6.43E-02	1.39E+00	2.71E+02	-2.71E+00	9.41E-02	1.29E+00	4.30E-02	2.58E+00	1.21E+01	6.42E-04	1.55E-01	4.95E-01	1.85E-01	1.34E+00	-1.61E+00
LFA(V)	6.94E-03	4.35E+00	1.32E+04	1.42E+00	3.28E-02	1.17E+00	5.86E-03	7.56E+01	1.37E+03	5.86E-06	2.71E-01	1.09E+00	3.59E-01	4.13E+00	6.89E-02
MFA(m)	1.45E-12	2.00E-05	2.55E-05	-5.16E+00	3.79E-13	1.64E-10	4.58E-13	4.32E-11	8.76E-10	7.20E-17	2.02E-11	7.48E-13	3.05E-11	6.30E-03	-1.78E+00
MFA(M)	4.03E-11	2.83E-04	1.18E+02	-3.00E+00	2.08E-10	7.56E-08	2.08E-10	6.44E-09	5.14E-08	4.05E-14	9.71E-10	1.45E-09	2.06E-09	1.84E+01	-9.96E-01
MFA(A)	1.52E-11	1.31E-04	2.37E+01	-5.09E+00	6.38E-11	1.67E-08	5.82E-11	1.00E-09	1.64E-08	8.14E-15	3.04E-10	2.60E-10	8.38E-10	2.83E+00	-1.52E+00
MFA(V)	1.10E-22	4.52E-09	2.32E+03	1.56E-01	2.62E-21	3.98E-16	2.34E-21	1.41E-18	2.05E-16	9.55E-29	6.06E-20	1.01E-19	3.13E-19	1.80E+01	8.07E-02
MSA_FFA(	0.00E+00	8.88E-16	5.93E+02	-5.16E+00	7.13E-09	0.00E+00	3.63E-55	5.90E-06	2.61E-53	5.26E-58	1.70E-55	1.09E-50	2.48E-07	6.34E-05	-1.93E+00
MSA_FFA(	1.36E-02	1.48E+00	6.61E+02	-5.16E+00	1.64E-06	9.95E-01	4.67E-04	6.76E-03	1.25E-01	1.92E-06	6.60E-03	7.12E-03	9.70E-05	5.99E-02	-1.80E+00
MSA_FFA(	3.11E-03	5.97E-01	6.19E+02	-5.16E+00	2.48E-07	1.06E-01	3.04E-05	2.08E-03	7.05E-03	1.33E-07	4.39E-04	2.45E-04	2.12E-05	9.01E-03	-1.81E+00
MSA_FFA(	1.37E-05	3.80E-01	5.77E+02	1.09E-10	1.63E-13	6.15E-02	1.05E-08	4.67E-06	5.91E-04	1.83E-13	1.77E-06	1.69E-06	5.83E-10	1.44E-04	1.08E-03
NaFA(m)	3.62E-13	2.97E-05	2.55E-05	-5.16E+00	7.13E-13	5.76E-11	2.69E-12	8.43E-11	4.06E-10	2.19E-17	8.87E-13	2.19E-12	1.71E-11	3.24E-11	-1.80E+00
NaFA(M)	3.27E-11	2.30E-04	1.18E+02	-3.00E+00	1.66E-10	2.51E-08	8.98E-11	2.24E-09	2.95E-08	1.20E-14	6.70E-10	2.71E-10	2.66E-09	4.71E-09	-1.80E+00
NaFA(A)	8.59E-12	1.00E-04	7.90E+00	-5.02E+00	4.31E-11	6.17E-09	2.17E-11	6.40E-10	6.64E-09	4.02E-15	1.27E-10	9.14E-11	4.59E-10	9.73E-10	-1.80E+00
NaFA(V)	4.77E-23	2.24E-09	9.03E+02	3.01E-01	2.02E-21	3.93E-17	5.63E-22	2.29E-19	4.17E-17	1.10E-29	2.01E-20	5.27E-21	2.55E-19	9.54E-19	6.19E-21
FAtidal(m)	0.00E+00	8.88E-16	2.55E-05	-5.16E+00	1.50E-32	0.00E+00	2.27E-54	0.00E+00	4.18E-51	3.39E-28	5.99E-53	1.58E-53	3.70E-32	0.00E+00	-1.80E+00
FAtidal(M)	0.00E+00	8.88E-16	4.57E+02	-2.42E+00	1.50E-32	3.98E+00	6.47E-52	0.00E+00	1.70E-49	1.60E-17	4.47E-51	4.02E-51	3.70E-32	2.27E-01	-1.21E+00
FAtidal(A)	0.00E+00	8.88E-16	1.21E+02	-5.01E+00	1.50E-32	1.13E+00	2.27E-52	0.00E+00	5.32E-50	1.03E-18	1.14E-51	8.22E-52	3.70E-32	7.56E-03	-1.76E+00
FAtidal(V)	0.00E+00	1.01E-62	1.57E+04	3.62E-01	7.75E-96	1.01E+00	3.97E-104	0.00E+00	1.61E-99	8.71E-36	8.18E-103	6.63E-103	2.79E-94	1.72E-03	2.22E-02

Tablo 4.4. İki boyutta ikinci grup fonksiyonlara bağlı karşılaştırma

Algorithm	Unimodal Functions		Multimodal Functions			Expanded Function
	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}	f_{21}
FA(m)	-4.50E+02	-4.50E+02	3.90E+02	-1.41E+02	-1.17E+02	-3.00E+02
FA(M)	-3.92E+02	-4.34E+02	4.79E+02	-7.78E+01	-1.15E+02	-2.77E+02
FA(A)	-4.44E+02	-4.46E+02	3.96E+02	-1.28E+02	-1.16E+02	-2.97E+02
FA(V)	1.26E+02	2.10E+01	2.86E+02	2.18E+02	5.09E-01	3.51E+01
LFA(m)	-4.50E+02	-4.50E+02	3.90E+02	-1.37E+02	-1.17E+02	-3.00E+02
LFA(M)	-8.08E+01	-3.76E+02	1.41E+03	6.45E+01	-1.09E+02	9.70E+02
LFA(A)	-4.05E+02	-4.34E+02	5.50E+02	-5.64E+01	-1.13E+02	-1.85E+02
LFA(V)	1.09E+04	6.02E+02	6.96E+04	2.68E+03	5.09E+00	9.58E+04
MFA(m)	-4.50E+02	-4.50E+02	3.90E+02	-1.41E+02	-1.17E+02	-3.00E+02
MFA(M)	-4.50E+02	-4.50E+02	3.90E+02	-1.40E+02	-1.17E+02	-3.00E+02
MFA(A)	-4.50E+02	-4.50E+02	3.90E+02	-1.40E+02	-1.17E+02	-3.00E+02
MFA(V)	3.26E-16	4.14E-16	4.66E-16	4.00E-01	7.25E-18	2.84E-16
NMSA_FFA(m)	-4.50E+02	-4.50E+02	3.90E+02	-1.41E+02	-1.17E+02	**-3.00E+02**
NMSA_FFA(M)	-4.50E+02	-4.50E+02	3.90E+02	-1.40E+02	-1.17E+02	-3.00E+02
NMSA_FFA(A)	-4.50E+02	-4.50E+02	3.90E+02	-1.40E+02	-1.17E+02	-3.00E+02
NMSA_FFA(V)	6.96E-17	1.28E-16	7.34E-17	4.50E-01	2.25E-18	7.85E-17
NaFA(m)	-4.50E+02	-4.50E+02	3.90E+02	-1.41E+02	-1.40E+02	-3.00E+02
NaFA(M)	-4.50E+02	-4.50E+02	4.03E+02	-1.40E+02	-1.30E+02	-3.00E+02
NaFA(A)	-4.50E+02	-4.50E+02	3.92E+02	-1.40E+02	-1.36E+02	-3.00E+02
NaFA(V)	3.72E-07	3.08E-07	8.17E+00	1.78E-01	8.48E+00	9.75E-07
FAtidal(m)	-4.50E+02	-4.50E+02	3.90E+02	-1.41E+02	-1.17E+02	-3.00E+02
FAtidal(M)	-4.50E+02	-4.50E+02	3.90E+02	-1.41E+02	-1.17E+02	-3.00E+02
FAtidal(A)	-4.50E+02	-4.50E+02	3.90E+02	-1.41E+02	-1.17E+02	-3.00E+02
FAtidal(V)	4.56E-23	4.98E-23	3.04E-23	8.36E-28	7.52E-27	1.13E-22

Tablo 4.5. İki boyutta birinci grup fonksiyonlara bağlı wilcoxon signed rank testi

Function	FA(p)	(h)	LFA(p)	(h)	MFA(p)	(h)	NMSA_FFA(P)	(h)	NaFA(p)	(h)
f1	1.73E-06	1	1.73E-06	1	1.73E-06	1	8.30E-06	1	1.73E-06	1
f2	2.42E-06	1	1.73E-06	1	1.73E-06	1	2.70E-05	1	1.73E-06	1
f3	4.68E-03	1	2.22E-04	1	1.38E-03	1	1.73E-06	1	5.87E-05	1
f4	6.33E-01	0	1.73E-06	1	4.65E-03	1	3.59E-04	1	7.50E-02	0
f5	1.00E+00	0	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
f6	4.24E-03	1	3.71E-01	0	2.60E-05	1	4.54E-05	1	2.60E-05	1
f7	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.13E-05	1	1.73E-06	1
f8	6.25E-02	0	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
f9	2.35E-06	1	1.73E-06	1	1.73E-06	1	2.60E-05	1	1.73E-06	1
f10	4.11E-03	1	1.73E-06	1	1.73E-06	1	6.73E-01	0	1.73E-06	1
f11	1.73E-06	1	1.73E-06	1	1.73E-06	1	5.75E-06	1	1.73E-06	1
f12	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
f13	1.25E-01	**0**	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
f14	2.15E-02	1	1.73E-06	1	1.73E-06	1	3.11E-05	1	3.11E-05	1
f15	7.80E-02	0	1.89E-04	1	1.36E-04	1	1.65E-01	0	5.00E-01	0

Tablo 4.6. İki boyutta ikinci grup fonksiyonlara bağlı wilcoxon signed rank testi

Function	FA(p)	(h)	LFA(p)	(h)	MFA(p)	(h)	NMSA_FFA(P)	(h)	NaFA(p)	(h)
f16	1.22E-05	1	3.77E-06	1	5.00E-01	0	1.73E-06	1	1.00E+00	0
f17	8.27E-06	1	5.58E-06	1	1.56E-02	1	1.73E-06	1	1.00E+00	0
f18	2.56E-06	1	2.55E-06	1	6.25E-02	0	1.73E-06	1	1.00E+00	0
f19	1.72E-06	1	1.73E-06	1	1.78E-06	1	4.18E-07	1	3.62E-05	1
f20	2.70E-05	1	1.73E-06	1	1.00E+00	0	1.73E-06	1	1.00E+00	0
f21	8.83E-05	1	5.58E-06	1	5.00E-01	0	1.73E-06	1	1.00E+00	0

Tablo 4.7. On boyutta birinci grup fonksiyonlara bağlı karşılaştırma

Algorithms	Many Local Minima						bowl shaped				plate shaped			Steep Ridges
	f1	f2	f3	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15
FA(m)	2.58E-14	1.96E-03	4.74E+02	1.12E-07	9.95E-01	4.97E-08	6.39E-06	6.35E-05	4.78E-11	1.35E-06	2.54E-07	1.80E-05	2.01E+00	-9.58E+00
FA(M)	1.04E-10	3.63E-03	1.64E+03	3.87E-07	2.39E+01	1.93E-07	1.70E+00	2.17E-04	1.35E-09	3.97E-06	1.22E-06	6.67E-01	9.22E+00	-7.34E+00
FA(A)	1.99E-11	2.80E-03	1.02E+03	2.49E-07	7.79E+00	1.20E-07	3.76E-01	1.15E-04	4.69E-10	2.49E-06	6.21E-07	6.44E-01	3.79E+00	-8.66E+00
FA(V)	4.69E-22	1.75E-07	8.89E+04	5.54E-15	2.57E+01	1.69E-15	3.87E-01	1.56E-09	9.88E-20	5.36E-13	5.33E-14	1.48E-02	2.90E+00	3.70E-01
LFA(m)	6.66E-16	1.16E+01	2.80E+02	4.39E+00	5.94E+01	1.97E+00	1.53E+02	3.64E+03	1.69E-03	8.49E+01	2.04E+01	1.68E+02	1.97E+03	-5.33E+00
LFA(M)	3.40E-02	1.87E+01	3.06E+03	3.37E+01	1.10E+02	2.79E+01	1.41E+05	2.51E+04	1.86E-01	4.96E+02	2.94E+03	5.40E+04	1.15E+05	-3.09E+00
LFA(A)	1.63E-03	1.62E+01	2.42E+03	1.54E+01	8.55E+01	1.75E+01	1.41E+04	1.21E+04	5.23E-02	2.98E+02	1.58E+02	1.68E+04	3.15E+04	-4.13E+00
LFA(V)	4.06E-05	3.36E+00	2.80E+05	5.04E+01	1.50E+02	4.30E+01	7.33E+08	2.75E+07	2.75E-03	1.58E+04	2.77E+05	1.86E+08	6.10E+08	3.13E-01
MFA(m)	3.57E-14	3.46E-03	2.07E+03	6.99E-07	8.95E+00	3.79E-07	1.44E-04	5.54E-04	7.07E-10	1.76E-05	2.87E-06	6.67E-01	1.42E+01	-1.68E+01
MFA(M)	5.06E-11	5.75E-03	4.34E+03	2.72E-06	4.38E+01	8.77E-07	1.05E+02	3.97E-03	9.85E-09	1.22E-04	9.65E-06	8.43E-01	1.93E+01	-1.26E+01
MFA(A)	1.69E-11	4.43E-03	2.99E+03	1.50E-06	2.44E+01	6.66E-07	1.49E+01	1.66E-03	3.01E-09	3.81E-05	6.10E-06	6.81E-01	1.71E+01	-1.49E+01
MFA(V)	1.94E-22	3.40E-07	2.39E+05	2.04E-13	8.76E+01	1.59E-14	7.02E+02	7.40E-07	3.22E-18	3.68E-10	2.84E-12	1.96E-03	1.69E+00	1.44E+00
NMSA_FFA(m)	2.66E-15	1.33E+00	3.51E+03	1.10E+01	4.98E+00	7.42E-19	1.33E-02	1.68E-17	7.16E-23	3.62E-18	1.30E-07	6.67E-01	6.31E+00	-9.80E+00
NMSA_FFA(M)	1.47E-08	2.00E+01	3.57E+03	4.42E+01	5.47E+01	1.93E-03	2.70E+01	1.01E-02	4.32E-08	1.11E-02	5.26E-04	6.82E-01	8.88E+01	-6.50E+00
NMSA_FFA(A)	3.36E-09	1.66E+01	3.52E+03	2.10E+01	2.89E+01	1.99E-04	4.26E+00	8.59E-04	2.90E-09	8.67E-04	7.44E-05	6.68E-01	1.32E+01	-9.02E+00
NMSA_FFA(V)	1.70E-17	4.94E+01	1.92E+02	8.43E+01	1.42E+02	1.85E-07	6.16E+01	4.73E-06	1.12E-16	7.66E-06	1.34E-08	1.06E-05	3.55E+02	5.01E-01
NaFA(m)	1.11E-14	1.59E-03	5.72E+02	5.37E-08	4.97E+00	3.11E-08	5.64E-04	3.05E-05	2.19E-09	6.43E-07	9.75E-08	6.43E-01	5.79E-01	-9.39E+00
NaFA(M)	1.44E-11	8.40E-01	1.69E+03	1.56E-07	2.29E+01	8.39E-08	5.75E+00	6.40E-05	2.75E-08	1.67E-06	3.08E-07	6.67E-01	2.48E+00	-6.66E+00
NaFA(A)	3.69E-12	2.99E-02	9.88E+02	1.15E-07	1.14E+01	5.77E-08	1.17E+00	4.64E-05	1.31E-08	1.19E-06	2.20E-07	6.66E-01	1.48E+00	-7.88E+00
NaFA(V)	1.14E-23	2.34E-02	8.78E+04	6.15E-16	1.32E+01	2.17E-16	2.67E+00	1.05E-10	3.85E-17	6.68E-14	2.74E-15	1.87E-05	1.70E-01	4.30E-01
Fatidal(m)	0.00E+00	4.44E-15	2.25E+03	1.26E-31	2.98E+00	9.66E-44	7.68E-03	4.19E-41	2.32E-09	1.13E-42	3.45E-43	6.67E-01	9.57E-01	-9.28E+00
Fatidal(M)	0.00E+00	1.51E-14	3.37E+03	1.50E-30	1.59E+01	3.00E-43	1.03E+01	2.45E-40	1.97E-07	6.39E-42	1.37E-42	6.67E-01	7.72E+01	-7.68E+00
Fatidal(A)	0.00E+00	7.40E-15	2.86E+03	7.24E-31	7.20E+00	2.05E-43	1.05E+00	1.66E-40	6.44E-08	3.83E-42	8.99E-43	6.67E-01	9.93E+00	-8.80E+00
Fatidal(V)	0.00E+00	4.42E-30	1.00E+05	1.44E-61	1.01E+01	3.02E-87	4.80E+00	2.01E-81	2.42E-15	1.35E-84	5.89E-86	3.19E-31	2.52E+02	1.71E-01

Tablo 4.8. On boyutta ikinci grup fonksiyonlara bağlı karşılaştırma

Algorithms	f16	f17	f18	f19	f20	f21
FA(m)	-4.50E+02	-4.50E+02	3.90E+02	1.60E+04	8.36E+03	-3.00E+02
FA(M)	-4.50E+02	-4.50E+02	3.90E+02	1.60E+04	8.36E+03	-3.00E+02
FA(A)	-4.50E+02	-4.50E+02	3.90E+02	1.60E+04	8.36E+03	-3.00E+02
FA(V)	2.47E-10	2.08E-10	1.99E-10	2.32E-02	4.93E-03	1.05E-10
LFA(m)	3.90E+03	4.69E+03	5.68E+03	2.82E+04	1.41E+04	3.18E+03
LFA(M)	1.94E+04	2.62E+04	2.23E+04	6.33E+05	2.27E+04	2.12E+04
LFA(A)	1.21E+04	1.25E+04	1.38E+04	3.52E+05	1.82E+04	1.31E+04
LFA(V)	1.06E+07	2.78E+07	1.61E+07	4.61E+10	5.29E+06	2.74E+07
MFA(m)	-4.50E+02	-4.50E+02	3.90E+02	1.60E+04	8.36E+03	-3.00E+02
MFA(M)	-4.50E+02	-4.50E+02	3.90E+02	1.60E+04	8.36E+03	-3.00E+02
MFA(A)	-4.50E+02	-4.50E+02	3.90E+02	1.60E+04	8.36E+03	-3.00E+02
MFA(V)	1.58E-10	2.77E-10	2.84E-10	1.87E-02	1.12E-03	1.14E-10
NMSA_FFA(m)	-4.50E+02	-4.50E+02	3.90E+02	1.60E+04	8.36E+03	-3.00E+02
NMSA_FFA(M)	-4.50E+02	-4.50E+02	3.90E+02	1.60E+04	8.36E+03	-3.00E+02
NMSA_FFA(A)	-4.50E+02	-4.50E+02	3.90E+02	1.60E+04	8.36E+03	-3.00E+02
NMSA_FFA(V)	4.35E-11	3.59E-11	2.71E-11	1.20E-09	1.90E-13	2.91E-11
NaFA(m)	-4.50E+02	-4.50E+02	3.90E+02	1.09E+03	-1.20E+02	2.95E+02
NaFA(M)	-4.50E+02	-4.50E+02	3.90E+02	1.09E+03	-1.20E+02	3.17E+02
NaFA(A)	-4.50E+02	-4.50E+02	3.90E+02	1.09E+03	-1.20E+02	3.12E+02
NaFA(V)	1.64E-05	2.96E-05	1.81E-05	2.14E-25	4.19E-03	4.47E+01
Fatidal(m)	-4.50E+02	-4.50E+02	3.90E+02	-1.80E+02	-1.40E+02	-3.00E+02
Fatidal(M)	-4.50E+02	-4.50E+02	3.90E+02	-1.80E+02	-1.40E+02	-3.00E+02
Fatidal(A)	-4.50E+02	-4.50E+02	3.90E+02	-1.80E+02	-1.40E+02	-3.00E+02
Fatidal(V)	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00

Tablo 4.9. On boyutta birinci grup fonksiyonlara bağlı wilcoxon signed rank testi

Function	FA(p)	(h)	LFA(p)	(h)	MFA(p)	(h)	NMSA_FFA(P)	(h)	NaFA(p)	(h)
f1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
f2	1.73E-06	1	1.73E-06	1	1.73E-06	1	3.35E-07	1	1.73E-06	1
f3	1.73E-06	1	1.60E-04	1	2.10E-01	0	1.73E-06	1	1.73E-06	1
f5	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.55E-06	1	1.73E-06	1
f6	1.00E+00	0	1.73E-06	1	2.55E-06	1	1.92E-06	1	2.92E-04	1
f7	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
f8	2.29E-01	0	1.73E-06	1	3.16E-03	1	2.25E-01	0	5.58E-01	0
f9	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
f10	1.73E-06	1	1.73E-06	1	1.92E-06	1	1.73E-06	1	6.98E-06	1

Tablo 4.9'ın devamı

f11	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
f12	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
f13	1.00E+00	0	1.73E-06	1	6.25E-02	0	3.13E-02	1	1.00E+00	0
f14	2.22E-04	1	1.73E-06	1	3.59E-04	1	2.61E-04	1	1.73E-06	1
f15	4.59E-01	0	1.73E-06	1	1.73E-06	1	1.57E-02	1	8.18E-05	1

Tablo 4.10. On boyutta ikinci grup fonksiyonlara bağlı wilcoxon signed rank testi

Functions	FA(p)	(h)	LFA(p)	(h)	MFA(p)	(h)	NMSA_FFA(P)	(h)	NaFA(p)	(h)
f16	1.00E+00	0	1.73E-06	1	1.00E+00	0	1.00E+00	0	1.00E+00	0
f17	1.00E+00	0	1.73E-06	1	1.00E+00	0	1.00E+00	0	1.00E+00	0
f18	1.00E+00	0	1.73E-06	1	1.00E+00	0	1.00E+00	0	1.00E+00	0
f19	4.32E-08	1	1.64E-06	1	4.32E-08	1	4.32E-08	1	4.32E-08	1
f20	4.32E-08	1	1.73E-06	1	4.32E-08	1	4.32E-08	1	4.32E-08	1
f21	1.00E+00	0	1.73E-06	1	1.00E+00	0	1.00E+00	0	1.43E-06	1

Tablo 4.11. Yüz boyutta birinci grup fonksiyonlara bağlı karşılaştırma

Algorithms	Many Local Minima					bowl_shaped				plate shaped			Steep Ridges
	f1	f2	f3	f5	f6	f8	f9	f10	f11	f12	f13	f14	f15
FA(m)	3.69E-13	1.44E-02	2.32E-02	8.57E-05	1.66E-04	3.64E-05	1.41E+00	1.02E-09	1.89E-01	8.93E+01	7.89E-01	9.60E+01	-5.98E+01
FA(M)	3.05E-11	1.42E-01	1.85E+05	2.14E+04	2.49E+02	1.21E+02	9.65E+01	1.14E-08	1.16E+01	2.07E+02	1.76E+01	9.73E+02	-3.90E+01
FA(A)	1.01E-11	2.43E-02	2.28E+04	1.50E+03	1.46E+02	4.05E+00	2.16E+01	6.61E-09	2.59E+00	1.56E+02	4.39E+00	3.24E+02	-5.31E+01
FA(V)	7.84E-23	4.98E-04	9.80E+08	2.94E+07	3.57E+03	4.91E+02	4.89E+02	9.56E-18	6.87E+00	1.25E+03	1.67E+01	1.10E+05	2.64E+01
LFA(m)	8.75E-14	1.51E+01	9.54E-05	1.47E+02	3.98E+02	1.41E+01	4.66E+05	2.81E-03	1.51E+03	3.98E+02	5.82E+03	2.70E+04	-3.05E+01
LFA(M)	6.65E-04	2.14E+04	3.77E+04	4.06E+02	1.23E+03	3.07E+02	2.26E+06	2.15E-01	5.04E+04	1.58E+08	3.12E+09	7.69E+06	-2.40E+01
LFA(A)	3.05E-05	4.56E+03	2.41E+04	2.62E+02	1.07E+03	2.25E+02	1.67E+06	3.54E-02	3.75E+04	5.28E+06	1.07E+08	1.24E+06	-2.70E+01
LFA(V)	1.48E-08	7.04E+07	2.58E+08	4.96E+03	3.69E+04	2.90E+03	1.90E+11	1.67E-03	1.64E+08	8.35E+14	3.23E+17	3.61E+12	2.44E+00
MFA(m)	3.29E-14	3.30E-07	2.02E+01	9.64E-05	1.06E+02	3.69E-05	1.39E+00	1.70E-09	5.32E-02	9.62E+01	7.58E-01	9.22E+01	-6.29E+01
MFA(M)	9.56E-11	3.99E-07	2.01E+01	2.01E+02	1.99E+02	6.84E-05	9.02E+01	1.80E-08	8.25E+00	2.40E+02	1.66E+01	9.87E+01	0.00E+00
MFA(A)	1.71E-11	4.73E-07	2.01E+01	3.50E+01	1.47E+02	5.12E-05	2.11E+01	6.85E-09	2.32E+00	1.62E+02	4.15E+00	9.70E+01	-5.29E+01
MFA(V)	4.73E-22	9.17E-07	2.01E+01	4.42E+03	5.30E+02	6.97E-11	5.26E+02	1.61E-17	5.26E+00	1.47E+03	1.71E+01	1.48E+00	1.22E+02
NMSA_FFA(m)	2.49E-10	1.60E+01	1.99E+01	1.18E+01	2.32E+02	2.06E-02	3.51E+00	5.79E-10	2.98E+00	6.69E+03	3.13E+00	9.74E+01	-5.04E+01
NMSA_FFA(M)	3.10E-06	3.63E+04	3.63E+04	9.81E+02	5.75E+02	1.20E-01	1.63E+02	2.51E-06	2.12E+02	3.27E+04	3.56E+02	4.68E+02	0.00E+00
NMSA_FFA(A)	1.46E-07	8.49E+03	1.10E+04	2.65E+02	3.73E+02	5.40E-02	2.43E+01	2.12E-07	3.41E+01	1.47E+04	4.95E+01	1.56E+02	-3.56E+01
NMSA_FFA(V)	3.15E-13	2.44E+08	2.85E+08	7.13E+04	6.57E+03	5.51E-04	1.29E+03	2.38E-13	2.31E+03	2.50E+07	8.24E+03	6.55E+03	8.56E+01
NaFA(m)	1.24E-13	3.10E-03	4.64E+00	4.35E-07	3.12E+01	2.08E-07	5.26E-02	5.08E-04	1.89E-08	1.01E-05	1.45E-06	6.44E-01	9.81E+00
NaFA(M)	1.74E-11	2.89E+03	2.01E+04	5.27E+01	1.92E+03	4.31E-07	3.29E+01	8.39E-04	6.20E-07	1.92E-05	2.98E-06	7.98E-01	1.71E+01
NaFA(A)	4.58E-12	5.38E+02	2.78E+03	1.10E+01	5.08E+02	3.38E-07	6.02E+00	6.59E-04	9.05E-08	1.51E-05	2.34E-06	6.71E-01	1.35E+01
NaFA(V)	1.70E-23	1.01E+06	1.13E+07	3.51E+02	2.99E+05	2.95E-15	7.66E+01	8.52E-09	1.09E-14	5.40E-12	1.17E-13	6.00E-04	1.70E+00
FAtidal(m)	0.00E+00	2.93E-14	3.60E+04	2.06E-12	8.95E+01	8.56E-42	1.86E+01	2.16E-08	1.25E+00	2.89E+02	7.96E-01	9.60E+01	-7.76E+01
FAtidal(M)	0.00E+00	4.35E-14	3.89E+04	4.46E+00	2.16E+02	1.14E-41	1.61E+03	1.26E-06	3.69E+01	6.23E+02	4.97E+01	3.12E+02	-5.92E+01
FAtidal(A)	0.00E+00	3.42E-14	3.75E+04	4.50E-01	1.36E+02	9.78E-42	5.20E+02	3.34E-07	1.17E+01	4.54E+02	8.57E+00	1.24E+02	-6.99E+01
FAtidal(V)	0.00E+00	2.48E-29	4.35E+05	8.31E-01	6.59E+02	6.25E-85	2.20E+05	1.01E-13	7.39E+01	8.16E+03	1.04E+02	2.83E+03	2.27E+01

Tablo 4.12. Yüz boyutta birinci grup fonksiyonlara bağlı karşılaştırma

Algorithms	f16	f17	f18	f19	f20	f21
FA(m)	-4.50E+02	-4.50E+02	3.90E+02	1.01E+04	8.08E+03	-3.00E+02
FA(M)	-4.49E+02	-4.48E+02	4.91E+02	1.34E+05	8.21E+06	-3.00E+02
FA(A)	-4.50E+02	-4.50E+02	4.33E+02	4.29E+04	9.64E+05	-3.00E+02
FA(V)	3.38E-02	1.18E-01	2.50E+03	3.05E+09	6.08E+12	5.23E-04
LFA(m)	2.79E+04	3.88E+03	2.82E+05	1.10E+04	2.55E+05	2.69E+04
LFA(M)	3.56E+05	3.01E+06	2.16E+11	5.71E+06	2.67E+07	3.11E+06
LFA(A)	2.99E+05	3.50E+05	8.22E+10	7.46E+05	1.71E+06	4.81E+05
LFA(V)	2.98E+09	2.64E+11	9.45E+21	3.22E+12	2.34E+13	6.73E+11
MFA(m)	-4.50E+02	-4.50E+02	3.90E+02	1.01E+04	8.15E+04	-3.00E+02
MFA(M)	-4.50E+02	-4.49E+02	4.90E+02	1.34E+05	8.60E+06	-3.00E+02
MFA(A)	-4.50E+02	-4.50E+02	4.50E+02	3.07E+04	4.16E+05	-3.00E+02
MFA(V)	6.58E-04	6.74E-02	2.44E+03	2.19E+09	2.43E+12	2.89E-04
NMSA_FFA(m)	-4.49E+02	-4.49E+02	3.91E+02	2.60E+04	8.10E+04	-2.99E+02
NMSA_FFA(M)	-4.48E+02	-4.48E+02	1.05E+03	2.96E+06	9.29E+05	-2.98E+02
NMSA_FFA(A)	-4.49E+02	-4.48E+02	6.49E+02	5.22E+05	1.73E+05	-2.98E+02
NMSA_FFA(V)	6.42E-02	4.63E-02	8.06E+04	6.78E+11	6.45E+10	1.87E-02
NaFA(m)	-4.50E+02	-4.50E+02	3.90E+02	1.02E+04	8.21E+04	-3.00E+02
NaFA(M)	-4.50E+02	-4.50E+02	7.40E+04	1.36E+05	8.59E+05	-3.00E+02
NaFA(A)	-4.50E+02	-4.50E+02	4.06E+03	3.96E+04	2.83E+05	-3.00E+02
NaFA(V)	3.72E-03	3.26E-03	2.00E+08	2.91E+09	1.14E+11	1.22E-03
FAtidal(m)	-4.50E+02	-4.50E+02	4.87E+02	5.98E+03	-1.20E+02	-2.55E+02
FAtidal(M)	-4.50E+02	-4.50E+02	8.17E+05	9.06E+03	-1.19E+02	-2.52E+02
FAtidal(A)	-4.50E+02	-4.50E+02	2.08E+05	7.08E+03	-1.19E+02	-2.53E+02
FAtidal(V)	0.00E+00	8.48E-05	5.09E+10	5.14E+05	5.26E-03	4.33E-01

Tablo 4.13. Yüz boyutta birinci grup fonksiyonlara bağlı wilcoxon signed rank testi

Function	FA(p)	(h)	LFA(p)	(h)	MFA(p)	(h)	NMSA_FFA(P)	(h)	NaFA(p)	(h)
f1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
f2	1.73E-06	1	1.72E-06	1	1.73E-06	1	1.16E-06	1	1.73E-06	1
f3	3.10E-05	1	1.56E-05	1	1.73E-06	1	2.01E-06	1	1.73E-06	1
f5	4.28E-01	0	1.73E-06	1	3.31E-04	1	1.73E-06	1	3.31E-04	1
f6	9.98E-02	0	1.73E-06	1	8.99E-03	1	1.73E-06	1	8.99E-03	1
f8	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
f9	2.13E-06	1	1.73E-06	1	1.73E-06	1	2.88E-06	1	1.73E-06	1
f10	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.96E-02	1	1.73E-06	1
f11	1.13E-05	1	1.73E-06	1	1.73E-06	1	5.45E-02	0	1.73E-06	1
f12	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
f13	5.71E-02	0	1.73E-06	1	1.73E-06	1	2.96E-03	1	1.73E-06	1
f14	1.58E-02	1	1.73E-06	1	1.72E-06	1	2.50E-02	1	1.72E-06	1
f15	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1

Tablo 4.14. Yüz boyutta ikinci grup fonksiyonlara bağlı wilcoxon signed rank testi

Functions	FA(p)	(h)	LFA(p)	(h)	MFA(p)	(h)	NMSA_FFA(P)	(h)	NaFA(p)	(h)
f16	5.00E-01	0	1.71E-06	1	1.00E+00	0	1.01E-07	1	1.00E+00	0
f17	1.00E+00	0	1.73E-06	1	5.00E-01	0	2.57E-07	1	1.00E+00	0
f18	1.92E-06	1	6.32E-05	1	1.92E-06	1	1.92E-06	1	1.92E-06	1
f19	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
f20	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
f21	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1

Tablo 4.15. ABC ve Fatidal ile iki boyutta birinci grup fonksiyonlara bağlı karşılaştırması ve wilcoxon signed rank testi

ABC	Many Local Minima						bowl_shaped					Plate_shaped			Steep Ridges
	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15
(min)	2.65E-07	2.69E-04	5.97E-04	-5.16E+00	1.41E-08	1.20E-03	1.26E-09	3.84E-06	2.66E-06	2.21E-12	4.57E-08	6.11E-08	2.25E-06	1.10E-04	-1.80E+00
(max)	3.42E-04	2.40E-02	1.70E+00	-5.15E+00	7.68E-07	1.84E-02	4.08E-07	7.72E-04	4.66E-04	5.64E-10	5.23E-06	7.37E-06	2.62E-04	6.87E-03	-1.80E+00
(Average)	5.21E-05	1.12E-02	3.23E-01	-5.16E+00	2.16E-07	6.93E-03	1.20E-07	1.79E-04	1.22E-04	7.95E-11	1.21E-06	2.36E-06	5.95E-05	1.10E-03	-1.80E+00
(variance)	7.04E-09	3.75E-05	1.56E-01	3.86E-06	3.84E-14	2.72E-05	9.95E-15	3.08E-08	1.41E-08	1.21E-20	1.18E-12	5.30E-12	4.02E-09	1.66E-06	2.09E-13
(P)	1.73E-06	1.73E-06	1.48E-03	6.25E-01	1.73E-06	2.60E-05	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	3.11E-05	5.00E-01
(h)	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0

Tablo 4.16. ABC ve Fatidal ile iki boyutta ikinci grup fonksiyonlara bağlı karşılaştırması ve wilcoxon signed rank testi

ABC	f16	f17	f18	f19	f20	f21
(min)	-4.50E+02	-4.50E+02	3.90E+02	1.37E+03	6.37E+02	-3.00E+02
(Max)	-4.50E+02	-4.50E+02	3.90E+02	1.37E+03	6.37E+02	-3.00E+02
(Average)	-4.50E+02	-4.50E+02	3.90E+02	1.37E+03	6.37E+02	-3.00E+02
(variance)	5.10E-09	3.05E-09	1.10E-09	1.34E-15	1.20E-25	2.25E-09
(P)	1.00E+00	1.00E+00	1.00E+00	4.32E-08	4.32E-08	1.00E+00
(h)	0	0	0	1	1	0

Tez kapsamında, algoritmaların zaman karmaşıklığı formel ve formel sız değerlendirilmiştir.

G_{max} : Maksimum jenerasyon, N^2: Popülasyon büyüklüğü

f : Fonksiyon, α : Bastırma formülü

Gösterildiği gibi, NaFA algoritması diğer algoritmaların koşma sürelerine göre daha randımanlı olmaktadır. Fatidal, FA, MFA ve LFA, ikinci grupta benzer zaman karmaşıklığına sahiptir. Çünkü bir seferde Luus Jaakola ve Opposition-based learning (OBL), sadece bir defa başlangıçta gerçekleştirilir. (Tablo 4.21). F1 fonksiyonunun üzerinde 2, 10 ve 100 boyutlar için tüm algoritmaları 30 defa çalıştırıldı ve ortalaması hesaplanmıştır (Tablo 4.22 ve Şekil 4.4). Formelsiz karşılaştırma da önerilen algoritma boyut arttıkça önerilen algoritma daha iyi performans göstermektedir.

Tüm deneyler, Intel® Core ™ (TM i5-4210u CPU 1.70 GHz ve 2.40 GHz'de 8.00 GB bellek ve Bilgisayar modeli Toshiba Skull candy uygulanmıştır. Önerilen algoritma ve türevleri MATLAB R2016a'da kodlanmıştır ve işletim sistemi de Windows 10'u kullanılmıştır.

Tablo 4.17. ABC ve Fatidal ile on boyutta birinci grup fonksiyonlara bağlı karşılaştırması ve wilcoxon signed rank testi

ABC	Many Local Minima						bowl shaped				plate shaped			Steep Ridges
	f1	f2	f3	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15
(min)	0.00E+00	3.11E-05	8.28E+02	1.95E-10	1.38E+01	9.81E-13	8.30E-01	1.50E-08	1.63E-21	2.67E-11	1.22E+00	6.67E-01	6.71E-02	-8.38E+00
(Max)	0.00E+00	3.69E-04	1.32E+03	3.67E-09	2.58E+01	1.04E-11	1.75E+01	4.03E-07	1.72E-18	8.27E-10	4.15E+00	6.67E-01	5.49E+00	-7.43E+00
(Average)	0.00E+00	7.83E-05	1.11E+03	1.41E-09	2.07E+01	4.22E-12	7.55E+00	1.18E-07	3.11E-19	2.79E-10	2.77E+00	6.67E-01	3.47E+00	-7.83E+00
(Variance)	0.00E+00	3.89E-09	1.66E+04	8.04E-19	7.51E+00	6.57E-24	1.47E+01	8.63E-15	1.94E-37	3.57E-20	6.68E-01	1.96E-10	2.91E+00	9.17E-02
(P)	1.00E+00	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	4.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.00E+00	1.73E-06	2.60E-06
(h)	0	1	1	1	1	1	1	1	1	1	1	0	1	1

Tablo 4.18. ABC ve Fatidal ile on boyutta ikinci grup fonksiyonlara bağlı karşılaştırması ve wilcoxon signed rank testi

ABC	f16	f17	f18	f19	f20	f21
(min)	-4.50E+02	-4.50E+02	3.90E+02	1.60E+04	8.36E+03	-3.00E+02
(Max)	-4.50E+02	-4.50E+02	3.90E+02	1.60E+04	8.36E+03	-3.00E+02
(Average)	-4.50E+02	-4.50E+02	3.90E+02	1.60E+04	8.36E+03	-3.00E+02
(variance)	9.23E-18	1.04E-17	4.97E-18	4.64E-15	1.05E-18	9.89E-18
(P)	1.00E+00	1.00E+00	1.00E+00	4.32E-08	4.32E-08	1.00E+00
(h)	0	0	0	1	1	0

4.2. Sonuçlar

Bu çalışmada gelgit olayı ile alarak Ateş böceği algoritması birleştirilmiştir. Az ve çok boyutlarda problemlerin üzerinde esneklik sağlanmıştır. Yapılan katkılardan birisi, basit Alfa operatörü ile ve küresel ve bölgesel arama aralarındaki dengeyi kurmakla algoritma geliştirilmiştir. FAtidal, dahili bastırma operatörler ve uygunluk değeri ile değişkenlerini (örneğin, m1, m2) kullanarak yeni bir bakış açısı getirmiştir. Formülde emme katsayısı olmamakla minimum değeri bulmaktadır. Diğer algoritmalara göre Küresel optimumda Wilcoxon Signed Rank testi ile değerlendirildi ve en iyi sonucu gösterilmiştir. Önerilen algoritma NP_hard problemlerine uygulanabilir ve ayrıca diğer mevcut popülasyon tabanlı algoritmalara ile adapte edilebilir.

Tablo 4.19. ABC ve Fatidal ile yüz boyutta birinci grup fonksiyonlara bağlı karşılaştırması ve wilcoxon signed rank testi

ABC	Many Local Minima					bowl_shaped				Plate_shaped			Steep Ridges
	f1	f2	f3	f5	f6	f8	f9	f10	f11	f12	f13	f14	f15
(min)	0.00E+00	1.59E+01	2.76E+04	3.54E+02	1.09E+03	1.03E+02	5.01E+05	5.32E-01	1.17E+04	1.39E+03	2.81E+06	4.33E+05	-2.47E+01
(Max)	0.00E+00	1.72E+01	3.24E+04	5.18E+02	1.23E+03	1.38E+02	6.43E+05	1.58E+00	1.45E+04	1.62E+03	4.87E+06	4.12E+07	-2.18E+01
(Average)	0.00E+00	1.67E+01	3.12E+04	4.51E+02	1.17E+03	1.20E+02	5.83E+05	9.78E-01	1.30E+04	1.49E+03	4.05E+06	2.27E+06	-2.29E+01
(variance)	0.00E+00	8.45E-02	1.14E+06	1.85E+03	1.17E+03	6.98E+01	9.94E+08	8.31E-02	5.58E+05	3.55E+03	1.94E+11	5.52E+13	4.40E-01
(P)	1.00E+00	1.65E-06	1.71E-06	1.73E-06	1.73E-06	1.72E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.732E-06
(h)	0	1	1	1	1	1	1	1	1	1	1	1	1

Tablo 4.20. ABC ve Fatidal ile yüz boyutta ikinci grup fonksiyonlara bağlı karşılaştırması ve wilcoxon signed rank testi

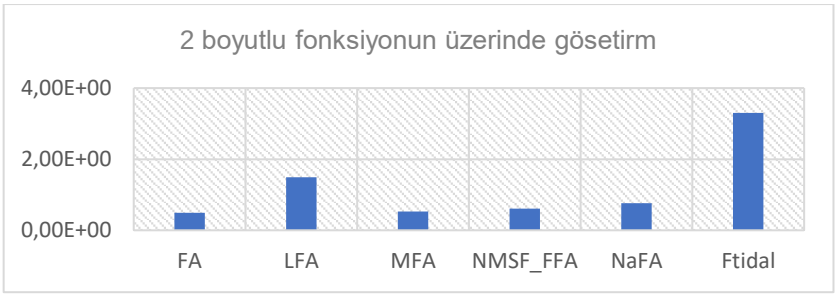
ABC	f16	f17	f18	f19	f20	f21
(min)	1.08E+04	1.16E+04	1.25E+04	1.35E+05	8.15E+04	1.22E+04
(Max)	1.61E+04	1.73E+04	1.69E+04	1.37E+05	8.15E+04	1.66E+04
(Average)	1.35E+04	1.40E+04	1.50E+04	1.36E+05	8.15E+04	1.38E+04
(variance)	1.68E+06	1.41E+06	9.21E+05	1.54E+05	3.55E+01	1.01E+06
(P)	1.72E-06	1.72E-06	9.32E-06	1.73E-06	1.73E-06	1.73E-06
(h)	1	1	1	1	1	1

Tablo 4.21. Formel şeklinde zaman karmaşıklığı

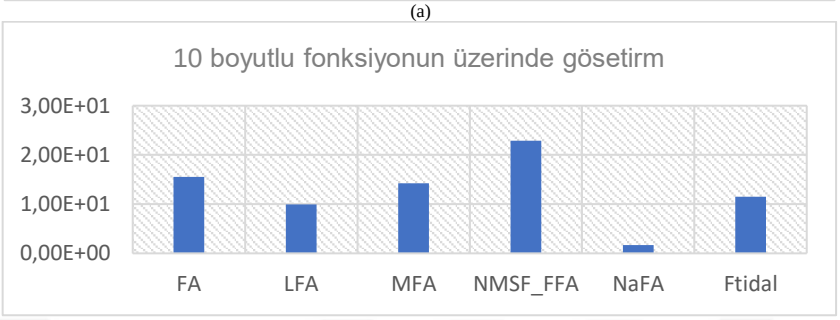
Algoritmalar	Zaman Karmaşıklığı
FA	$O(G_{max}.N^2.f.a)$
LFA	$O(G_{max}.N^2.f.a)$
MFA	$O(G_{max}.N^2.f.a)$
NaFA	$O(G_{max}.N.k.f.a)$ *K* komşu sayısı olmak üzere $\frac{N-1}{2}$ den az olacak
NMSF_FFA	$O(G_{max}.N^2.f.a+o+L)$ L : Luus_jaakola başlangıçta kullanılan fonksiyon
FATIDAL	$O(G_{max}.N^2.f.a+o)$ O: Opposition-based learning başlangıçta kullanılan fonksiyon

Tablo 4.22. Fonksiyon F1 üzerinde zaman karmaşıklığı

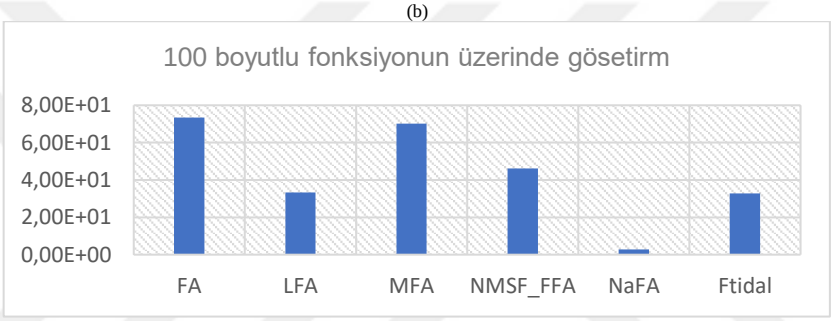
Algoritmalar	2 (boyutlu)	10 (boyutlu)	100 (boyutlu)
FA	4.95E-01	1.55E+01	7.34E+01
LFA	1.49E+00	9.96E+00	3.34E+01
MFA	5.28E-01	1.42E+01	7.02E+01
NMSF_FFA	6.08E-01	2.29E+01	4.62E+01
NaFA	7.60E-01	1.70E+00	2.75E+00
FAtidal	3.30E+00	1.15E+01	3.29E+01



(a)



(b)



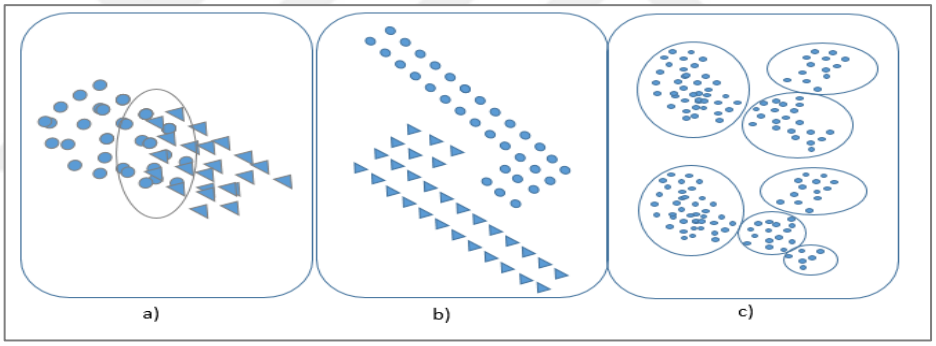
(c)

Şekil 4.4. Zaman karmaşıklığı karşılaştırması: a) 2 boyutlu b)10 boyutlu c)100 boyutlu

4.3. Kümelemeye Kapsayan Yapılan Tartışma

4.3.1. Giriş

Tez kapsamında DBSCAN ve Fuzzy metotların özelliklerini kullanarak Örtüşme, morfolojik ve Küme sayısı problemlerinin çözülmesi için iki kümeleme algoritması önerilmiştir (Şekil 4.5). Birinci önerilen algoritma, AFD algoritması, veri yoğunluğuna dayalı kümeleme problemi çözmek için yeni yollar açmıştır. Daha sonra önerilen algoritma ise (AFD) ve bu alanda mevcut olan algoritmaların lokal minimumda takılması ve parametrelerin seçilme hassasiyetine önlemek için ikinci önerilen algoritma Fatidal-DBSCAN'dır. Bu bölümde Fatidal algoritması ile veri setlerin üzerinde veri yoğunluğuna dayalı kümeleme yapılmasını hedeflenmiştir ve yeni bir kümeleme modelini sunulmuştur. Çalışmada önerilen modern sezgisel algoritma uygulamasının yansıra yeni bir bakış açısı ortaya koymuştur.



Şekil 4.5. Kümeleme Problemler a) Örtüşme b) Morfoloji
c) Küme sayısı (Ör. 3 ya 4)

4.3.2. AFD algoritması

Öncelikle, algoritma formül 4.2 [85] ile verilerin yüksek yoğunluk bölgeyi seçerek ve daha sonra tüm verileri DBSCAN tarafından tanımlanan parametrelerle soldan sağa doğru veri uzayını tarar. Veri komşunun seçilme şartı 3.1'de açıklanmıştır.

$$F(x;\varepsilon_1)=\{y\in D|N_x(y)\geq\varepsilon_1\} \qquad x\in D \tag{4.1}$$

$$N_x(Y)=\exp\left(-\left(k.\frac{d(x.y)}{d^{max}}\right)^2\right) \tag{4.2}$$

k: katsayı genelde 5'i seçilir

d^{max} : En uzak mesafe

d(x.y): İki nesnenin uzaklığı

$N_x(y)$: x'in komşu sayısı

ε_1: Birinci yarıçapı(Epsilon) kullanıcı tarafından belirlenir

İlk aşamada, çekirdek verileri bulmakla sonraki aşamada ise alt kümelerin kategorize edilmesine yol açan Eps1 ve Min parametreleri kullanarak kümelemeyi gerçekleştirmesi sağlanır. İkincisi, alt kümeler birleşme ve ayırma için, FindEps2 fonksiyona gönderilir. Fonksiyon, ikinci epsilon (Eps2) ve epsilonun pozisyonu (Pos) kurallara dayalı iki parametreyi otomatik olarak üretir. Parametrelerin üretilmesi için, bir veri setin uzayındaki alt kümelerin konumunun (Kartezyen koordinat sistemi) düzenlenmesi ile yapılır. Tüm konum durumlar koordinat sistemde alt kümelerin merkezini tanımlar. Dört durum tanım 1'de ADminimum, ADmaximum, ADmean ve Diffmean adları olan açıklanmıştır. Birleştirme ve ayırmaya değinen dört duruma dayanarak yapılır. Bulanık niceliği (formül 4.1) ve Bulanık Komşu kavramını (formül 4.2) kullanarak sonra veri yoğunluğuna dayalı taramakla algoritmamız problemlere yeni bir çözüm ortaya koymuştur. Çözülecek uzayın ölçüsünü kısaltmak için formül 4.3'ü[85] kullanılarak normal verilere dönüştürülür.

$$X_{ij}=\frac{x_{ij}-X_j^{min}}{\left(X_j^{max}-X_j^{min}\right)}\ ,\ j=1,m \tag{4.3}$$

$$X_j^{min}=\min_{i=1,n} x_{i,j} \text{ and } x_j^{max} \max_{i=1,n} x_{i,j}\ ,j=1,m$$

Iris veri setinde, 3 sınıfta tanımlanan sepal uzunluğu, sepal genişliği, yaprak uzunluğu, yaprak genişliği, petal uzunluğu ve petal genişliği gibi nitelikler içermektedir ve 150 örneğe sahiptir. Önerilen algoritmamız tarafından kümelenmiş olan Iris veri seti Şekil 4.7 göstermektedir.

4.3.2.1. Tanımlama

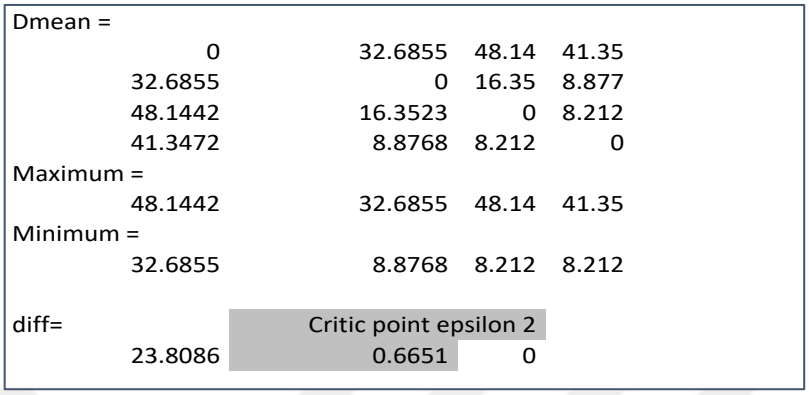
Tanım 1: Konum durumlar aşağıdaki gibi tanımlanmış ve Şekil 3.2'de Iris veri setin üzerinde Örnek olarak gösterilmiştir (şekil 3.3).

• Dmean (ADmean): Alt kümelerin merkezin arasındaki uzaklık değeri.

• Maxsimum (ADmaximum): Dmean'daki her bir sütunun maksimum değeri.

• Minimum (ADminimum): Dmean'daki her bir sütunun minimum değeri.

• Diff (Diffmean): Her bir sütun arasındaki Minimum Fark değeri.



Şekil 4.6. İris veri setin üzerinde önerilen algoritmada 7 ve 8. Satır ile ilgili hesaplama

Tanım 2: Findpeaks (Pos): Bir vektörün yerel maksimum ve minimum doruklar değeri olarak döndürür. Örnek olarak yerel maksimum doruk, iki komşu değerden daha yüksek bir değerdir. (Şekil 4.7, at (e) findpeak=3 ve (f) findpeak=2)

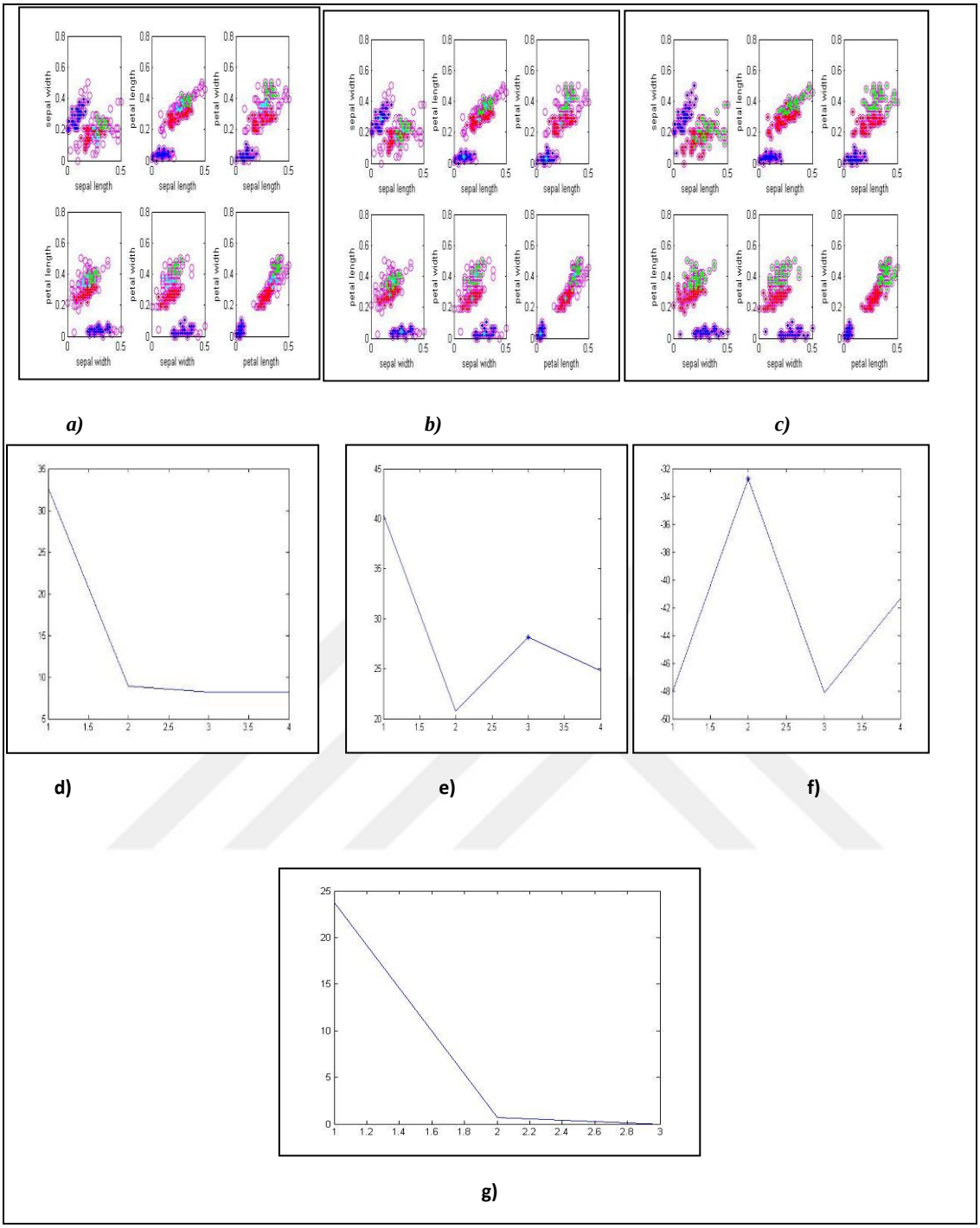
Tanım 3: Binary separation: Alt kümelerin konumları birbirlerine karşılık gelen durumlardır. Her ikisinin de Minimum, Maximum ve Average gibi ikili kodları döndürür. Soldan sağa konumlanan alt kümeler, eğer küçükse 0, büyükse 1 atanmaktadır. Şekil 4.7 (d, f ve e).

BMA=Minimum=101 Şekil 4.7 (d) Maximum duruğu(peak) alınmakla

BMI =Maximum=111 Şekil 4.7 (f) Minimum duruğu(peak) alınmakla

BME =Means=101 Şekil 4.7 (e) Maximum duruğu(peak) alınmakla

Önerilen algoritma (Şekil 4.8) aşağıda sunulmuş ve 11 satır yazılmıştır ve bunları üç aşamaya ayrılabilir. İlk aşamada 1-5 satır arası yazılmıştır, (bkz. Şekil 4.7(a)). 6-9 satırlar ise ikinci aşamayı belirtir (bkz. Şekil 4.7 (b)). İki kural şekil 4.9, 4.10'da gösterilmiştir. Üçüncü aşamada 10-11 satıra ait hiçbir kümeye ait olmayan verileri yakın kümeye verilmesi işlemi yapılır. Değinen satırda elde edilen Eps2 değerlerinin alt kümeleri birleştirip ayırma yapılmaktadır.



Şekil 4.7. Stepler a) step1, b) step2, c) step3, d) bminimum, e) ADmean, f) bmaximum g) diffmean. İris veri setin üzerinde metrik ölçüsü F =0.9600, RandIndx = 0.9495, AdjRandIndx =0.8857. (*) Dorukları göstermek için.

1. İki parametre Eps1 ve MinPoint tanımlanır ve N tane veri örneği ve d ise veri setin boyutu/özelliği olmaktadır.
2. Verilerin üzerinde formül (3.3)'e göre normaliz yapılması ve verilerin aralarındaki uzaklığın hesaplanması.
3. Normalize edilmiş komşu derecesi formül (3.2)'e göre hesaplanması.
4. DBSCAN tekniği ile Çekirdek verileri ve Düğüm(normal) verileri elde edilmesi (Eps1, MinPoint'leri (formül 3.1 ve 3.2)) sonradan alt kümeleri üretmesi).
5. Tüm komşular (Çekirdek verileri ve Düğüm(normal) verileri) seed'in içine ata, sonradan seed'de bulunan verileri soldan sağa formül 3.1 göre sınıflandır ve sürekli yeni bulunan verilerin üzerinde bu işlemi tekrarla.
6. Bütün elde edilen alt kümelerin ortalama değerlerini hesapladıktan sonra kümelerin ortalama uzaklığını hesapla.
7. Fonksiyon findEps2 (Dmean, sub-clasters) çağır. Bu aşama Eps2(critic point) ve pozisyonu Eps2(pos) bulmaya çalışmaktadır. Kural A ve 1,2 ve 3 uncu tanımına bakınız (Şekil 3.7).
8. Çıkarılan pozisyonlar alt kümeleri birleşme ve ayırma amacı ile fonksiyonu çağır. Kural B(Şekil 3.8).
9. Tüm alt kümeler bir Eps2 kümesine dayalı olarak ayrılır ve birleştirilir, sonra yeni alt kümesini oluşturulur.
10. Tüm alt kümeleri elde ederek kalan düğümleri sınıflandır.
11. Sonlandır.

Şekil 4.8. AFD algoritması

Kural A, beş parçaya bölünür, ilk bölümde setleri tanımlar ve geri kalan bölümler, algoritma çalışırken meydana gelen alt kurallardır. Alt kural 0 tamamen ayrılabilir kümeleri kontrol etmek için tanımlanmıştır. Farklı koşulların altında Eps2 miktarını elde etmek için alt kural 1-4 uygulamaktadır (Şekil 4.9). Kural B ise alt kümelerin ayrılmasına veya birleştirilmesi gerçekleşmektedir (Şekil 4.10).

Rule A for finding Eps2 and pos

The definition sets

Let be D is a database of N points in d dimensional space R^d

Let be mean matrix D, distance between clusters(C), which are gained with Eps1 and MinPoints in line 4 algorithm AFD

Let be ADminimum is minimum of Dmean for each sub cluster ADminimum=$\{MI_1, MI_2, \ldots, MI_n\}$ see bminimum

Let be ADmaximum is maximum of Dmean for each sub cluster ADmaximum= $\{MA_1, MA_2, \ldots, MA_n\}$ see bmaximum

Let be ADmean is average of ADmaximum and ADminimum for each sub cluster ADmean=$\{ME_1, ME_2, \ldots, ME_n\}$

Let be Diffmean is subtracted the ordered member of ADmean .Diffmean=$\{DM_1, DM_2, \ldots, DM_{n-1}\}=\{MI_1 - MI_2, MI_2 - MI_3, \ldots, MI_{n-1} - MI_n\}$ see

$bMaximum \in Findpeak(ADmaximum), bMinimum \in Findpeak(ADminimum), bmean \in$ Findpeak (ADmean).

Position Number=Pos $\in \{bMaximum \cup bMinimum \cup bmean\}$.

sub rule 0

$BMI = BMA = BME = \{0,1\}$.

C=subclusters, N=member of sub cluster numbers.

$BMI = \{S \mid S \in N \wedge MI_s < MI_{s+1} \in BMI \wedge MI_s < MI_{s+1} = 1 \vee MI_s < MI_{s+1} = 0\}$

$BMA = \{S \mid S \in N \wedge MA_s < MA_{s+1} \in BMA \wedge MA_s < MA_{s+1} = 1 \vee MA_s < MA_{s+1} = 0\}$

$BME = \{S \mid S \in N \wedge ME_s < ME_{s+1} \in BME \wedge ME_s < ME_{s+1} = 1 \vee ME_s < ME_{s+1} = 0\}$

Let be output of binary seperation =BMI, BMA and BME. See definition 2

Intersect_ Binary seperation= {intersect_binary| $(BMI \cap BMA \cap BME)$}.

Saperate_subcluster=($\forall (p,q) \in C)((s,d) \in N \wedge \ q_s \neq p_d$)

Saperate_cluster={pos=-1,eps1=-1| $\exists Saperat\_subcluster \wedge (BMI = BMA = BME)$ }. (checking all sub clusters are seperable and return pos=-1, eps1=-1).

sub rule 1

Pos1={ $pos1 \mid pos1 \in N \wedge bMaximum \cap bMinimum$ }

Defrule1=$(\exists pos1 \in pos)(DM_1 \in Diffmean \mid pos1 \wedge$ bMinumum<pos1 $\wedge \ DM_1 = 0)$

Rule1=pos={ bMinumum =bMinumum -1|$\exists$ Defrule1}

Eps2= diffminumum(bMinumum)

Defrule2=$(\exists pos1 \in pos)(DM_1 \in Diffmean \mid pos1 \wedge$ bMaximum<pos1 $\wedge \ DM_1 = 0)$

Rule2=pos={ bMaximum =bMaximum -1|$\exists$ Defrule1}

Eps2= Diffmean (bMaximum)

Defrule1_2={pos, eps2|$\exists Defrule2 \vee \exists Defrule2$}

sub rule 2

Defrule3={$bMinumum \mid pos1 \in pos \wedge (\exists bMaximum \wedge \exists bMinumum)$}

Rule3_1=pos={bMinumum| $\exists$Defrule3$\wedge \exists$(Diffmean(1) > Diffmean ($bMinumum$)) }

Rule3_2=pos={bMinumum= bMinumum -1| $\exists$Defrule3$\wedge$(Diffmean(1) > Diffmean ($bMinumum$)) }

Eps2= Diffmean (bMinumum).

Defrule4={ $bMaximum \mid pos1 \in pos \wedge (\exists bMaximum \wedge \exists bMinumum)$}

Rule4-1=pos={ $bMaximum$ | $\exists$Defrule4$\wedge$(Diffmean(1) > Diffmean ($bMinumum$)) }

Rule4_2=pos={bMinumum= bMinumum -1| $\exists$Defrule4$\wedge$(Diffmean(1) > Diffmean ($bMinumum$)) }

Eps2= Diffmean ($bMaximum$).

sub rule 3

Lst_Unintersect=($bMaximumb \cap Minumum$)

$bMaximum = bMinumum =$ Lst_Unintersect

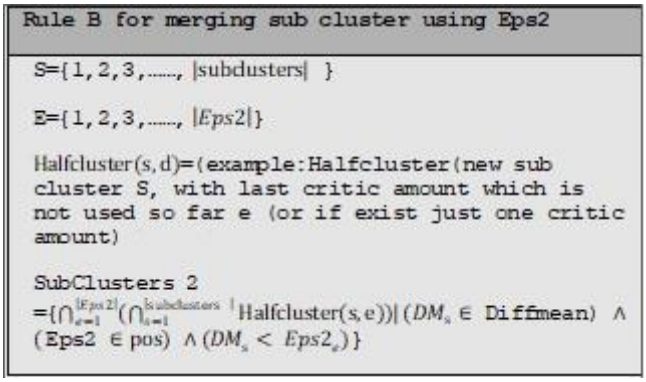
Defrule5={$bMinumum = bMinumum - 1 \mid pos1 \in pos \wedge \exists (bMaximum \cap bMinumum) \wedge Diffmean(1) = 0 \vee Diffmean(1) > Diffmean(bMinumum))$} .

Defrule6={$bMaximum = bMaximum - 1 \mid pos1 \in pos \wedge \exists (bMaximum \cap bMinumum) \wedge Diffmean(1) = 0 \vee Diffmean(1) > Diffmean(bMaximum))$} .

Rule5_6=pos=(bMinumum $\cup$ $bMaximum$).

Eps2= $Diffmean$ (bMinumum $\cup$ $bMaximum$).

Şekil 4.9. Kural A



Şekil 4.10. Kural B

İris veri setin üzerinde önerilen algoritma çalıştırılmış ve Şekil 4.8 (a,b ve c)'de 3 aşama gösterilmiştir. Başlangıçta Eps1=0.53, MinPts=5 olmak üzere alt kümelerin konum durumu belirlenir sonra ayırmak ve birleştirmek için karar verilir. Kural A'ya göre Şekil 4.9 (d, e, f ve c) göstermiş olan alt kümelerin konum durumu ve elde edilen Eps2=0.6651 ve pos=2 göstermektedir (Şekil 4.6). Eps2 miktarı (Eps2=Eps2-0.0002=0.6649) yaptıktan sonra kural B ile ayırma ve birleşme işlemi yapılmaktadır (Şekil 4.10). Örnek olarak DM_s<0.6649 (kritik nokta) küçük ise yeni bir alt küme oluşturur ya da ayırma işlemi yapılır.

4.3.2.2. Karşılaştırma Sonuçları

Algoritmaları karşılaştırmak için üç gerçek ve beş sentetik veri seti üzerinde çalışmalar yapıldı. UCI Makine Öğrenme deposundan alınmış olan gerçek veri setleri, örtüşme problemi test etmek için Wine, Glass ve Iris veri setleri kullanılmıştır (Şekil 4.11). Sentetik veri setleri ise morfoloji/ gelişi güzel problemi kullanılmıştır (Şekil 4.12). Burada, algoritmalar 30 kez çalıştırılmış ve elde edilmiş olan ortalama değerlerini bilinen DBSCAN algoritması ile karşılaştırılmıştır. Kümelenmiş veri setleri doğruluk bakımından algoritmaların performansını ölçmek için Rand Index, Adjusted Rand Index ve F- measure gibi bilinen ölçütleri kullanılmıştır. Tablo 4.23'te görüldüğü gibi, indekslerin maksimum değerleri kalınlaştırılmış ve Tablo 4.24, ayarlanmış olan parametreleri göstermektedir. AFD algoritması, Glass veri set üzerinde F-measure ölçümü dışında diğer ölçütlerde DBSCAN'a göre daha iyi sonuca sahip olduğunu göstermektedir. Doğru sayıda kümelemeyi ulaşmış olan önerilen algoritma ve DBSCAN algoritması Tablo 4.25'te gösterilmektedir. ADF algoritması görüldüğü gibi, Glass veri seti dışında küme sayısını tüm veri setlerde doğru şeklinde elde etmiştir.

Örnek sayısı: 150
Özellik Sayısı: 4
Özellik bilgisi:
1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm

Sınıf bilgisi:
1. Iris Setosa
2. Iris Versicolour
3. Iris Virginica

Örnek sayısı:178
Özellik Sayısı:13
Özellik bilgisi:
1. Alcohol
2. Malic acid
3. Ash
4. Alcalinity of ash
5. Magnesium
6. Total phenols
7. Flavanoids
8. Nonflavanoid phenols
9. Proanthocyanins
10. Color intensity
11. Hue
12. OD280/OD315 of diluted wines
13. Proline

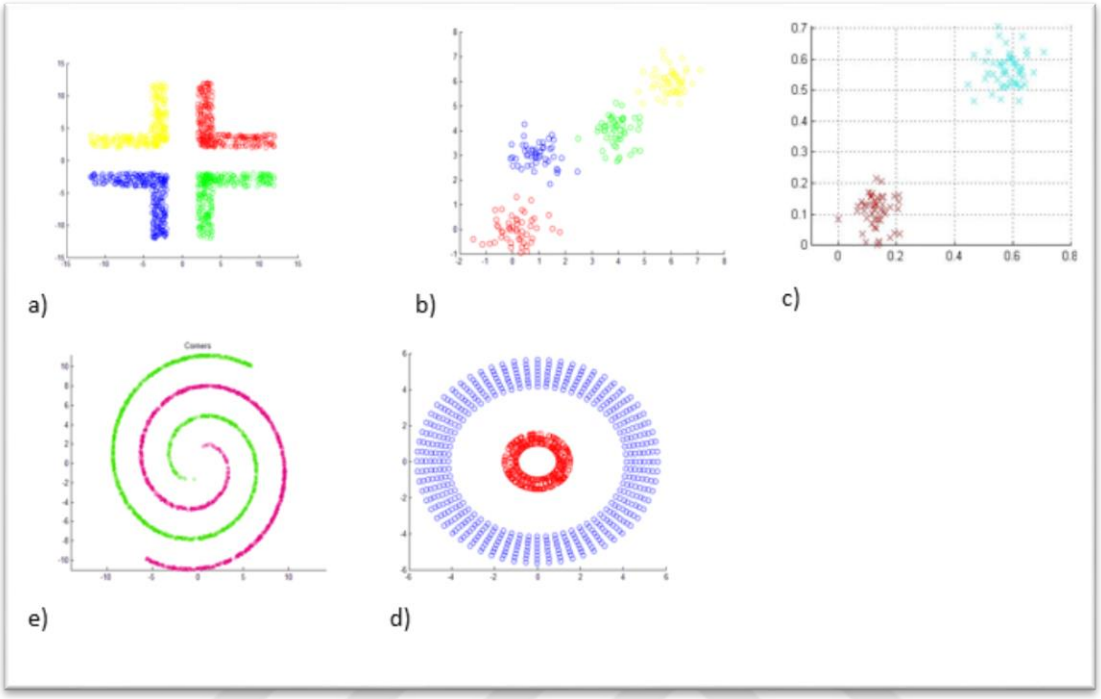
Sınıf sayısı:3

Örnek sayısı: 214
Özellik Sayısı: 10 **Özellik bilgisi:**
1. Id number: 1 to 214
2. RI: refractive index
3. Na: Sodium (unit measurement: weight percent in corresponding oxide, as are attributes 4-10)
4. Mg: Magnesium 5. Al: Aluminum
6. Si: Silicon 7. K: Potassium
8. Ca: Calcium 9. Ba: Barium
10. Fe: Iron **Sınıf bilgisi:**
1.building_windows_float_processed
2. building_windows_non_float_processed
3. vehicle_windows_float_processed
4. vehicle_windows_non_float_processed (none in this database)
5. Containers 6. tableware 7. headlamps

Şekil 4.11 UCI deposundan kullanılan veri setler

Önerilen algoritma İris veri seti üzerinde yoğunluk tabanlı algoritmalar LOF ve Spektral kümeleme gerçekleştirilmiştir (Şekil 4.14, 4.15). Gördüğünüz gibi, LOF ve Spektral kümeleme

algoritmalar Iris veri seti üzerinde önerilen algoritmaya göre daha iyi sonuçları göstermemiştir. Her bir renk, pembe renk dışında bir kümeye işaret etmektir.



Şekil 4.12. Sentetik veri seti. a) Köşede olan kümeler, b) Dört ayrı küme, c) İki ayrı küme, d) İki spiral küme, e) iki küme içi içe

Tablo 4.23. Algoritmaların karşılaştırması

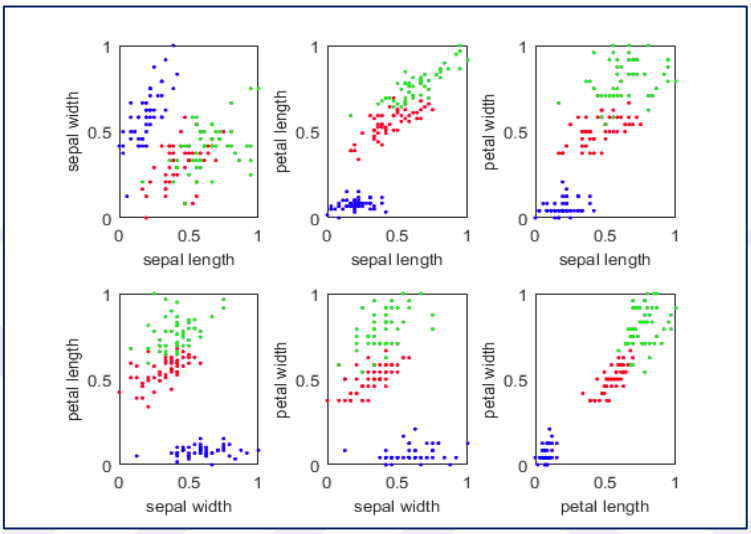
Name	Index	wine	glass	iris	cluster in cluster	corner separate	4 saperat cluster	2 saperate cluster	two spirals
ADF	RandIndx	**9.28E-01**	**7.96E-01**	**9.42E-01**	**1**	**1**	**1**	**1**	**1**
	AdjRandIndx	**8.39E-01**	**4.87E-01**	**8.68E-01**	**1**	**1**	**1**	**1**	**1**
	F	**9.48E-01**	4.66E-01	**9.53E-01**	**1**	**1**	**1**	**1**	**1**
DBSCAN	RandIndx	5.81E-01	6.55E-01	7.92E-01	**1**	**1**	5.87E-01	**1**	**1**
	AdjRandIndx	2.43E-01	3.03E-01	5.85E-01	**1**	**1**	3.36E-01	**1**	**1**
	F	4.47E-01	**5.70E-01**	5.78E-01	**1**	**1**	2.58E-01	**1**	**1**

Tablo 4.24. Parametreleri ayarlanması

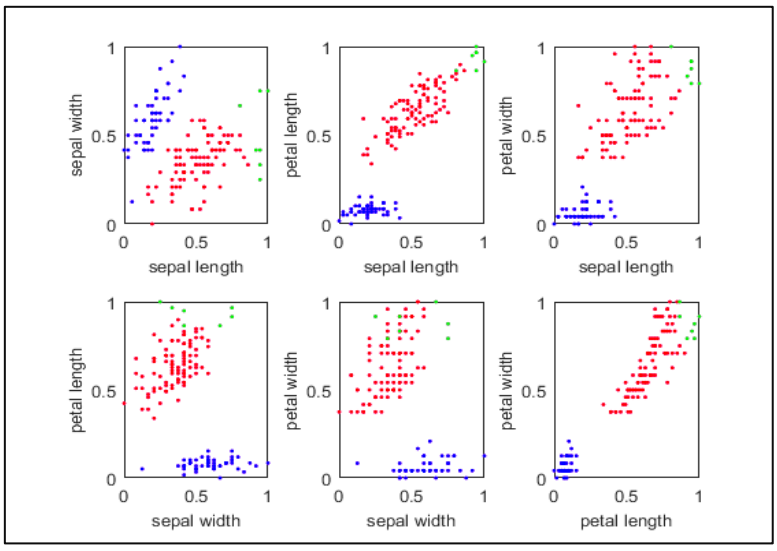
Name	Wine	Glass	iris	cluster in cluster	corner separate	4 saperat cluster	2 saperate cluster	two spiral
AFD	eps1=0.19 minpoint=4	eps1=0.60 minpoint=3	eps1=0.60 minpoint=3	eps1=0.7 minpoint=1	eps1=0.8 minpoint=3	eps1=0.7 minpoint=1	eps1=0.2 minpoint=5	eps1=0.7 minpoint =10
DBSCAN	minpoint =4, eps= 0.5	minpoint =5 eps = 0.5,	minpoint =5 eps= 0,8950	minpoint =1 eps= 0.001	minpoint =4 eps= 0.8950	minpoint =4 eps= 0.9	minpoint =3 eps= 0.5	minpoint =3 eps= 0.5

Tablo 4.25. Küme sayısı

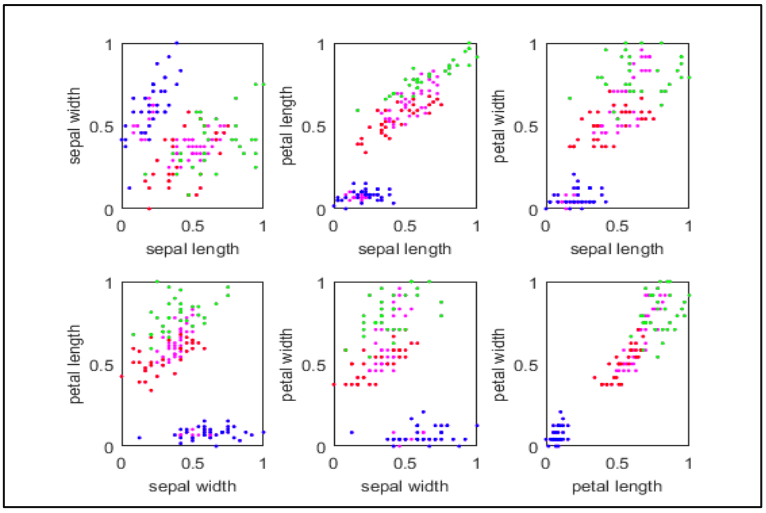
Name	wine(3)	glass(6)	iris(3)	cluster in cluster(2)	corner separate(4)	Four saperat cluster(4)	Two saperate cluster(2)	two spiral(2)
ADF(proposed algorithm)	3	4	3	2	4	4	2	2
DBSCAN	4	7	2	2	2	2	2	2



Şekil 4.13. Iris veri seti



Şekil 4.14. Spektral kümeleme Sigma 2 ile İris Veri seti üzerinde



Şekil 4.15. LOF kümeleme Minpts=10 ve Eps=1 ile İris veri seti üzerinde

4.3.2.3. Sonuç ve Tartışma

Önerilen algoritmanın özelliğini göstermek için gerçek ve sentetik veri setlerin üzerine koşturulmuştur. AFD algoritması çalışırken alt kümelerin ayrılması ve birleştirilmesi arasında karar vermek için veri setlerin üzerinde EPS2 ve Pos adlı olan iki parametre üretildi. Deneylerin sonuçları, UCI deposundan (Wine, Glass ve İris) veri setleri ve sentetik veri setleri (Şekil 4.11 ve 4.12) gerçek veri kümesine uygulanan AFD algoritmasının diğer algoritmalara göre daha iyi sonuçları göstermiştir. AFD algoritması küçük boyutta daha iyi sonuçları elde etmekte, ama diğer algoritmalardan daha fazla zaman almaktadır. Buradaki, Eps1 ve MinPoints parametreleri çok hassas ve dikkatli değer atanması gerekiyor. Veri setleri çok karmaşık yapıya sahip ise algoritmanın kuralları yetersiz olacaktır. Genellikle, başlatma parametreleri yüksek yoğunluğu gözlemleyerek gerçekleştirilir, yani miktarlarını tahmin etmemiz gerekir. Özet olarak algoritmanın parametrelere hassasiyeti ve yetersiz kuralları yeterince yumuşak kümeleme ile yapılmaktadır. Gelecek bölümde, önerilen modern sezgisel yöntem ile bahsedilen AFD problemlerini göz önüne alarak yeni bir kümeleme algoritması yazılmaktadır.

4.3.3. Fatidal Algoritmasının Uygulanması

Daha önce bahsedildiği gibi modern sezgisel algoritma genelde küresel minimumu bulmada uygulanmaktadır. AFD algoritması bulanık ve DBSCAN algoritmalarının

özelliklerini taşıyarak yeni bir kümelemeye yol açmıştır, ancak bu algoritma aşağıdaki dezavantajlara sahiptir.

4. Algoritma kurallara veya veri setindeki küme düzenine bağlı olarak çalışmakta.
5. Algoritma parametre değerlerine çok hassas ve farklı sonuçlar elde edilebilmektedir.

Klasik metotlar genelde bölgesel minimumda takılmaya müsaittirler yani daha iyi sonuçlar varken iyi sonucu sunarlar. Önerdiğimiz Fatidal algoritması DBSCAN ve AFD'ının özelliklerini taşıyarak yeni bir farklı uygulamaya yol açmıştır.

4.3.3.1. Tanımlar

Fatidal algoritması DBSCAN tanımı Epsilon (bulunduğu verinin/noktanın yarıçapısı) ve MinPoints (minimum nokta değeri) ve AFD deki Epsilon2 ve pozisyon parametrelerini simülasyon yaparak veri yoğunluğuna dayalı kümeleme yapılmıştır.

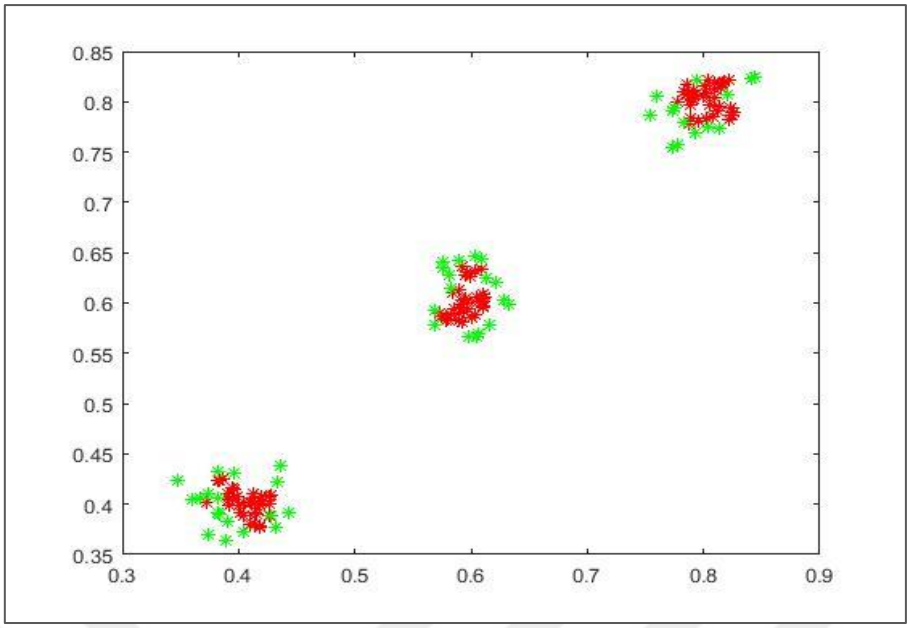
4.3.3.1.1. Küresel Veri Yoğunluk Seviye Tanımı

Epsilon2 ve pozisyon üretmek için X, N Elemanlı bir vektör ise, Findchangepts fonksiyonu $X'in$ ortalamasının en fazla değiştiği diziyi döndürür. Yani $X'i$, her bölgenin (kare) hatanın toplamını en aza indirmekle X $(1: ipt-1)$ ve X $(ipt: N)$ olmak üzere iki bölgeye dönüştürür.

Bu fonksiyonu kullanmak için öncelikle veri yoğunluk formülünü 4.4 kullanarak [107] sonradan verileri az yoğunluktan çok yoğunluğa sıralama yapılmıştır. Örnek olarak ekil 4.16 gösterildiği gibi Findchangepts fonksiyonu kullandık sonuçta iki seviye ayırmış olan veri setleri mavi (1. seviye) ve kırmızı (2.seviye) renk ile belirlenmiş.

$$l_{density}(p)=\frac{1}{l}\sum_{i=1}^{l} d_i \qquad (4.4)$$

- $l_{density}(p)$ Bir noktanın yoğunluk hesaplanması.
- l MinPoint ya da komşuluk sayısı.
- d Uzaklık mesafesi.



Şekil 4.16. Veri yoğunluğuna dayalı küresel seviye belirlenmesi

4.3.3.1.2. Uygunluk Fonksiyonu

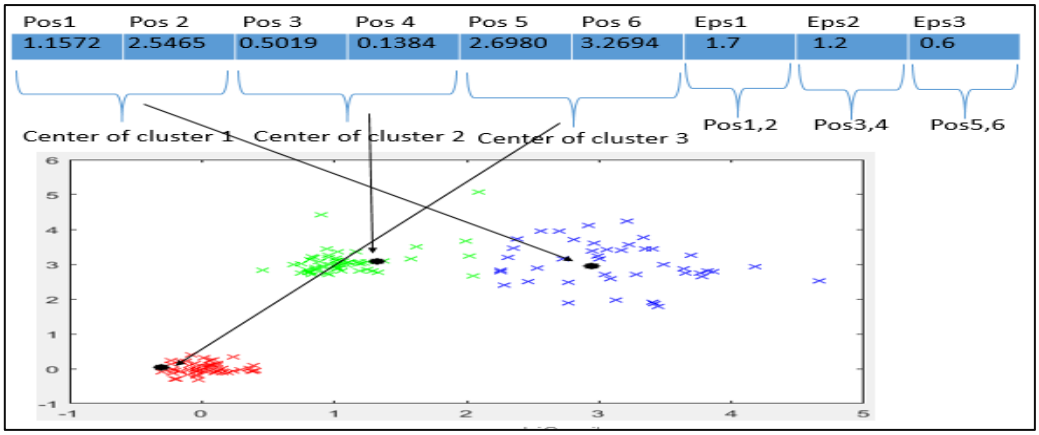
Uygunluk fonksiyonu veri yoğunluğuna dayalı ve seviye tanımlama ile önerilmiştir.

$$\text{Uygunluk\_F}=\sum_{k=1}^{C_k}\left(\sum_{i=1}^{NC_i}\text{density}_{l(p_i)}\Big/\left(\sum_{i=1}^{N}\text{density}_{l(P)}+\left(\sum_{i=1}^{NC_i}\text{Level}(p_i)\,\middle|\,p_i=2\right)\times 2\right)\right) \quad (4.5)$$

- C: bulunan küme sayısı.
- NC: Her kümenin verilerin sayısı.

4.3.3.2. Fatidal_DBSCAN Algoritması

Her bir vektör veya her bir Ateşböceği kümelemek amacıyla aşağıdaki gibi Şekil 4.17 tanımlanmıştır. Vektördeki değişken sayısı maksimum ne kadar kümeleme yapılması ve veri setinin kaç boyutlu olmasına bağılı olarak tanımlanmıştır. Örnek olarak Şekil 4.17'de 9 değişken bulunmaktadır. İkili üç tane DBSCAN'nın merkez konum başlanması bulunmaktadır ve diğer üç değişken her bir pozisyona yarıçapı olarak tanımlanmıştır. İkişer konumlar veri setin boyutuna bağlıdırlar ve 3 ikişer konumda maksimum beklenti küme sayısına işarettir. Diğer durumda küme sayısı 2 ya da 1 olabilir ve bu işlem önceden belirlen tesadüf şeklinde yapılmaktadır. Elde edilen sayılara göre belirli değişkenler etkisiz şekilde çalışabilir.



Şekil 4.17. Vektör ve küme ilişkisi

Şekil 4.18 Algoritmada seviye miktarı 2 olarak varsayılmıştır. Fonksiyon değerlendirme sayısı 100 ve popülasyon ise 20 olarak verilmiştir.

1- Amaç Fonksiyonu f(x), x=$(x_1, \dots, x_d)^T$
2- Başlangıçtaki Ateş Böcek popülasyonunu oluştur $x_i (i = 1,2, \dots, n)$
3- Veri seti yükle
4- Veri yoğunluğuna dayalı seviye belirle
5- G, m_1, m_2, Alpha, MaxGeneration, nVar tanımla.
6- While (t<MaxGeneration)
7- For i=1:n
8- For j=1:n
9- If ($I_i \neq I_j$)
10- If ($I_i < I_j$), i Ateş böceği j'ye hareket eder; end if
11- For k=1:NC
12- DBSCAN(dataset) // her defa bir küme bulması
13- End
14- Amaç Fonksiyonu hesapla (4.5)
15- Ön hesaplama (3.6-13)
16- Yeni çözüm üretilmesi ve güncelleme işlemi (3.14)
17- end if
18- end for j
19- end for i
20- Ateş böcekleri sırala ve en iyisini g_* ata.
21- End While

Şekil 4.18. Fatidal_DBSCAN Algoritması

4.3.3.3. Karşılaştırma Sonuçları

Diğer mevcut olan PSO_kmeans algoritması önerilen algoritma ile karşılaştırdı ve parametre değerlerine uygun şekilde verilmiştir. Sonuçta görüldüğü gibi koyu olan değerler diğerine göre iyi sonuçlar sunmuştur. Ancak burada daha örtüşme problemi sahip olan 3 veri setin üzerinde yapılmıştır. Diğer 5 suni veri setlerde çok basitçe bahsedebiliriz Fatidal_DBSCAN algoritması DBSCAN metodundan yararlanmaktadır. DBSCAN'nın bir özelliği faklı şekil verileri başarı ile kümeleme yapılmasıdır o yüzden önerilen algoritma beş suni veri setlerin üzerinde başarı ile kümelenmiştir. Ama diğer algoritma Kmeans'e dayalı bir algoritmadır. Bu algoritma merkezi ve yarıçapına bağlı olarak kümeleme yapmaktadır. Bu algoritma morfolojiye sahip olan veri setleri Şekil 4.12 e, d başarısız sonuçlar göstermektedir. Ayrıca küme sayısında Tablo 4.26 görüldüğü gibi kırmızı olan başarısız sonuçları göstermektedir.

Tablo 4.26. Önerilen algoritmanın karşılaştırması

Aloritma	Index	Wine	Glass	Iris
Fatidal_DBSCAN	*Küme sayısı*	**3**	**2**	**3**
	RandIndx	8.55E-01	7.01E-01	8.29E-01
	AdjRandIndx	6.72E-01	2.20E-01	6.14E-01
	F	7.92E-01	1.70E-01	8.18E-01
PSO_Kmeans	*Küme sayısı*	**2**	**2**	**2**
	RandIndx	6.52E-01	7.05E-01	8.81E-01
	AdjRandIndx	2.32E-01	1.92E-01	7.68E-01
	F	5.07E-01	1.62E-01	9.01E-01

5. SONUÇLAR

Gelgit olayından ilham alan bu çalışma Ateş böceği algoritması ile birleştirildi. Az ve çok boyutlu problemlerin üzerine uygun şekilde monte edilerek esneklik sağlanmaktadır. Yapılan katkılardan birisi, basit Alfa operatörü kullanılmakta ve küresel ve bölgesel arama aralarındaki dengeyi sağlamakla algoritma geliştirilmiştir. FAtidal, dahili bastırma operatörler ile değişkenlerini (örneğin, m1, m2) kullanarak yeni bir bakış açısı getirmiştir. Formülde emme katsayısı olmamakla minimum değeri bulmaktadır. Karşılaştırılmış algoritmalara göre Küresel optimum için Wilcoxon Signed Rank testte en iyi sonuçlar göstermektedir. Karşılaştırılan çalışmalar, önerilen algoritmanın diğerlerine göre oldukça rekabetçi olduğunu göstermektedir. Önerilen Algoritmayı (Fatidal) yeni alanlarda uygulamak için veri madenciliğinde kümeleme alanına odaklanmıştır. Veri madencilikte verilerin üzerinde anlam, trend ve keşif gibi hedefleri bulması bir teknik sürecidir. Kümeleme örüntülerini bir birler ile benzeyenlerin gruplanması veri madencilikte kullanılan bir tekniktir. Çalışmada kullanılan kümeleme tekniklerden birisi veri yoğunluğuna dayalı bir tekniktir. Üçüncü bölümde önerilen algoritma (AFD algoritması) veri yoğunluğuna dayalı küçük bir çabada bir teknik sunmuştur. Sonunda algoritmaları karşılaştırmak için üç gerçek ve dört sentetik veri seti üzerinde gerçekleştirildi. UCI Makine Öğrenme veri deposundan elde edilmiş olan gerçek veri setleri, örtüşme problemini test etmek için Wine, Glass ve Iris veri setleri kullanılmıştır. Sentetik veri setleri ise morfoloji/ gelişi güzel problem kullanılmıştır. Burada, algoritmalar 30 kez çalıştırılmış ve elde edilmiş olan ortalama değerleri bilinen DBSCAN, algoritma ile karşılaştırılmıştır. Kümelenmiş veri setleri doğruluk bakımından algoritmaların performansını ölçmek için Rand Index, Adjusted Rand Index ve F-measure gibi bilinen ölçümleri kullanmıştır. AFD algoritması, Glass veri set üzerinde F-ölçümü dışında diğer ölçümlerde DBSCAN'a göre daha sağlam sonuca sahip olduğunu göstermektedir. Doğru sayıda kümelemeye ulaşmış olan önerilen algoritma ve DBSCAN algoritması, Glass veri seti dışında küme sayısını tüm veri setlerde doğru şeklinde elde edilmiştir. AFD algoritması bulanık ve DBSCAN algoritmaların özelliklerini taşıyarak yeni bir kümelemeye yol açmıştır, ancak bu algoritma aşağıdaki dezavantajlara sahiptir. 1- Algoritma kurallara veya veri setteki küme düzenini bağlı olarak çalışmakta. 2- Algoritma parametrelerin miktarlarına çok hassas ve farklı sonuçlar elde edilmesi. Klasik metotlar genelde bölgesel minimumda takılmaya müsaittirler, yani daha iyi sonuçlar varken iyi sonuçlar bulabilirler.

Minimuma takılmayı ve parametre hassasiyetini gidermek için yeni Fatidal_DBSCAN algoritması önerilmiştir.

Diğer mevcut olan PSO_kmeans algoritması önerilen algoritma (Fatidal_DBSCAN) ile karşılaştırdı. Sonuçta algoritma farklı şekillerde ve örtüşme durumlarında iyi sonuçlar göstermiştir.

6. ÖNERİLER

FAtidal algoritma mevcut popülasyon tabanlı algoritmalar ile adapte edilebilir ve tüm uygulanmış olan sürü zekâsı ve evrimsel algoritmalar kendi performansını sergileyebilir. Yeni konu olan çok amaçlı optimizasyon problemi de basitçe modellenebilir. Diğer ifade ile çalışmada tek amaçlı olarak problemlerin üzerinde odaklanmış, ancak gerçek karmaşık dünyada birden fazla amaç fonksiyonu bulmaktadır ve tek bir cevabın yerine bir küme olan bastırılmamış cevabı bulmaktadır. FAtidal algoritması veri madenciliği, görüntü işlemi, sensor ağlarda, diagnostik sistemlerde, özellik seçimi, gezgin satıcı problemi , semantik web servisi, iş çizelgemde vb. Kullanılmaktadır.

7. KAYNAKLAR

1. Yang, X.S., Nature-inspired metaheuristic algorithms,Second Edition, Luniver press, United Kingdom, 2010.

2. Talbi, E.-G., Metaheuristics: From Design to Implementation, Wiley Publishing, New Jersey, 2009.

3. Glover, F., Future paths for integer programming and links to artificial intelligence, Computers & Operations Research, 13, 5 (1986) 533-549.

4. Arora, J., Introduction to Optimum Design, Second Edition, Elsevier Science, San Diego, 2004.

5. www.slideplayer.com/slide/10786238/,slideplayer, 15 Kasım 2017 .

6. Bozorg-Haddad, O., Solgi, M. ve LoÃ¡iciga, H.A., Meta-heuristic and Evolutionary Algorithms for Engineering Optimization,First Edition, Wiley Publishing, New Jersey, 2017.

7. Bartle, R.G., The Elements of Real Analysis, Second Edition, John Wiley & Sons, Incorporated, 1982.

8. Apostol, T.M., Mathematical Analysis, Second Edition, Addison-Wesley, 1974.

9. Takayama, A., Mathematical economics, First Edition, Dryden Press, South-Western, 1974.

10. Santiago, T., www.currenteventsdigest.blogspot.com.tr. 14 Ekim 2017.

11. Simon, G., www.slideshare.net/kaalnath/tsp-43384571. 21 Kasım 2017.

12. Weisstein, E. W., www.mathworld.wolfram.com/NP-Problem.html, 17 Temmuz 2017.

13. Čertíková, M., www.marian.fsik.cvut.cz/~certik/NM/Amns.pdf.19 Eylül 2017.

14. Glover, F. ve Laguna, M., Tabu Search, Kluwer Academic Publishers, Massachusetts, 1997.

15. Kirkpatrick, S., Gelatt, C.D. ve Vecchi, M.P., Optimization by Simulated Annealing, Science, 220, 4598 (1983) 671-680.

16. Kurada, R.R., Pavan, D.K.K. ve Rao, D.A., A preliminary survey on optimized multiobjective metaheuristic methods for data clustering using evolutionary approaches, İnternational Journal of Computer Science & Information Technology (IJCSIT), 5, 5 (2013) 57-57.

17. Holland, J.H., Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence, MIT Press, Michigan, 1992.

18. Goldberg, D.E., Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Longman Publishing Co., Michigan, 1989.

19. Schwefel, H.P., Evolution and Optimum Seeking: First Edition, Wiley-Interscience , New York., 1993.

20. Schwefel, H.P., Numerical Optimization of Computer Models, John Wiley & Sons, Chicheste, 1981.

21. Mukhopadhyay D.M., Balitanas M.O., Farkhod A., Jeon S.H. ve Bhattacharyya D., Genetic Algorithm: A Tutorial Review, International Journal of of Grid and Distributed Computing, 2, 3 (2009) 25-32.

22. Krause, J., Ruxton, G.D. ve Krause, S., Swarm intelligence in animals and humans, Trends in Ecology & Evolution, 25, 1 (2010) 28-34.

23. Gueron, S., Levin, S.A. ve Rubenstein, D.I., The Dynamics of Herds: From Individuals to Aggregations, Journal of Theoretical Biology, 182, 1 (1996) 85-98.

24. Anderson, C., Swarm Intelligence: From Natural to Artificial Systems. Eric Bonabeau , Marco Dorigo , Guy Theraulaz, The Quarterly Review of Biology, 76, 2 (2001) 268-269.

25. Parrish, J.K., Hamner, W.M. ve Prewitt, C.T., Animal Groups in Three Dimensions: How Species Aggregate, Introduction – From individuals to aggregations: Unifying properties, global framework, and the holy grails of congregation, J.K. Parrish ve W.M. Hamner editors, Cambridge University Press, Cambridge, 1997.

26. Bonabeau, E., Dorigo, M. ve Theraulaz, G., Swarm intelligence: from natural to artificial systems, Oxford university press, 1999.

27. Millonas, M. M., Swarms, phase transitions, and collective intelligence, Santa Fe Institue Studies In The Sciences of Complexity-Proceedings Volume, Addison-Wesley Publishing,Haziran 1994, Santa Fe, Bildiriler Kitabı , 417-417.

28. Karaboga, D., Gorkemli, B., Ozturk, C. ve Karaboga, N., A comprehensive survey: artificial bee colony (ABC) algorithm and applications, Artificial Intelligence Review, 42, 1 (2014) 21-57.

29. Kennedy, J. ve Eberhart, R., Particle swarm optimization, Proceedings of the. 1995 IEEE International Conference on Neural Networks, Kasım 1995, New Jersey, Bildiriler Kitabı , 1942-1948.

30. Rao, S.S., Engineering Optimization: Theory and Practice, New Age International, New Delhi, 1996.

31. Reynolds, C.W., Flocks, herds and schools: A distributed behavioral model, SIGGRAPH Comput. Graph., 21, 4 (1987) 25-34.

32. www.daneshjooyar.com. 10 Aralık 2017.

33. www.file.scirp.org/Html/9-7401422/f50596ac-6e22-4fb3-bfd6-8fec26a3e31b.jpg. 7 Mayıs 2017.

34. Hassan, R., Cohanim, B., de Weck, O. ve Venter, G., Structural Dynamics and Materials Conference, A Comparison of Particle Swarm Optimization and the Genetic Algorithm, American Institute of Aeronautics and Astronautics, Texas, 2005.

35. Karaboga, D., Akay, B. ve Ozturk, C., Modeling Decisions for Artificial Intelligence, Torra, V., Narukawa, Y. ve Y. Yoshida editors, Springer Berlin Heidelberg, Berlin, 2007.

36. Karaboga, D., Yapay Zeka Optimizasyon Algoritmaları, 2, Nobel Akademik Yayıncılık, Ankara, 2011.

37. Dorigo, M. ve Caro, G.D., New ideas in optimization, The ant colony optimization meta-heuristic, C. David, et al. editors, McGraw-Hill Ltd., United Kingdom, 1999.

38. Reynolds, R.G., An introduction to cultural algorithms, Proceedings of the third annual conference on evolutionary programming, Şubat 1994, San Diego, Bildiriler Kitabı,131-139.

39. Pham, D., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S. ve Zaidi, M., The bees algorithm, Manufacturing Engineering Centre, Cardiff University, Temmuz 2005, Bildiriler Kitabı, 1-57.

40. Geem, Z.W., Kim, J.H. ve Loganathan, G.V., A new heuristic optimization algorithm: Harmony search, Simulation, 76, 2 (2001) 60-68.

41. Atashpaz-Gargari, E. ve Lucas, C., Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, Evolutionary computation,Eylül 2007 , Singapore , Bildiriler Kitabı, 4661-4667.

42. Yang, X.S., Firefly algorithm, stochastic test functions and design optimisation, International Journal of Bio-Inspired Computation, 2, 2 (2010) 78-84.

43. Yang, X.S., Research ve Development in Intelligent Systems XXVI: Incorporating Applications and Innovations in Intelligent Systems XVII, M. Bramer, R. Ellis ve M. Petridis editors, Mart 2010, Springer London, London, 209-218.

44. Wang, X., Wang, Y. ve Wang, L., Improving fuzzy c-means clustering based on feature-weight learning, Pattern Recognition Letters, 25, 10 (2004) 1123-1132.

45. Verma, O.P., Aggarwal, D. ve Patodi, T., Opposition and dimensional based modified firefly algorithm, Expert Systems with Applications, 44 (2016) 168-176.

46. Wang, H., Wang, W.J., Zhou, X.Y., Sun, H., Zhao, J., Yu, X. ve Cui, Z.H., Firefly algorithm with neighborhood attraction, Information Sciences, 382 (2017) 374-387.

47. Fister Jr, I., Yang, X.-S., Fister, I. ve Brest, J., Memetic firefly algorithm for combinatorial optimization, Proceedings of the Fifth International Conference on Bioinspired Optimization Methods and their Applications, Mayıs 2012, Bohinj, Bildiriler Kitabı , 75-76.

48. Fister Jr, I., Yang, X.-S., Brest, J. ve Fister Jr, I., Memetic self-adaptive firefly algorithm, Swarm intelligence and bio-inspired computation: theory and applications, (2013) 73-102.

49. Galvez, A. ve Iglesias, A., New memetic self-adaptive firefly algorithm for continuous optimisation, International Journal of Bio-Inspired Computation, 8,5 (2016) 300-317.

50. Tilahun, S.L. ve Ong, H.C., Modified Firefly Algorithm, Journal of Applied Mathematics, 2012 (2012) 12.

51. Falcon, R., Almeida, M. ve Nayak, A., Fault identification with binary adaptive fireflies in parallel and distributed systems, 2011 IEEE Congress of Evolutionary Computation (CEC), Haziran 2011, New Orleans, 1359-1366.

52. Farahani, S.M., Abshouri, A.A., Nasiri, B. ve Meybodi, M., Some hybrid models to improve firefly algorithm performance, International Journal of Artificial Intelligence, 8, 12, (2012) 97-117.

53. Farahani, S.M., Abshouri, A., Nasiri, B. ve Meybodi, M., A Gaussian firefly algorithm, International Journal of Machine Learning and Computing, 1, 5 (2011) 448.

54. Yang, X.-S., Experimental Algorithms: Metaheuristic Optimization: Algorithm Analysis and Open Problems, Pardalos P.M. ve Rebennack S. editors, Springer Berlin Heidelberg, Berlin, 2011.

55. Yang, X.-S., Efficiency analysis of swarm intelligence and randomization techniques, Journal of Computational and Theoretical Nanoscience, 9, 2 (2012) 189-198.

56. Yang, X.-S., Firefly algorithms for multimodal optimization, Proceedings of the 5th international conference on Stochastic algorithms: foundations and applications, Ekim 2009, Sapporo, Bildiriler Kitabı, 169-178.

57. Lukasik, S., Firefly Algorithm for Continuous Constrained Optimization Tasks, Proceedings of the 1st International Conference on Computational Collective, Ekim 2009, Wroclaw, Bildiriler Kitabı, 97-106.

58. Palit, S., Sinha, S.N., Molla, M.A., Khanra, A. ve Kule, M., A cryptanalytic attack on the knapsack cryptosystem using binary Firefly algorithm, 2011 2nd International Conference on Computer and Communication Technology (ICCCT-2011), Eylül (2011), İndia, Bildiriler Kitabı, 428-432.

59. Chandrasekaran, K. ve Simon, S.P., Network and reliability constrained unit commitment problem using binary real coded firefly algorithm, International Journal of Electrical Power & Energy Systems, 43, 1 (2012) 921-932.

60. Yang, X.-S., Hosseini, S.S.S. ve Gandomi, A.H., Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect, Applied Soft Computing, 12, 3 (2012) 1180-1186.

61. Horng, M.-H., Vector quantization using the firefly algorithm for image compression, Expert Systems with Applications, 39, 1 (2012) 1078-1091.

62. Senthilnath, J., Omkar, S.N. ve Mani, V., Clustering using firefly algorithm: Performance study, Swarm and Evolutionary Computation, 1, 3 (2011) 164-171.

63. Gálvez, A. ve Iglesias, A., Firefly algorithm for polynomial Bézier surface parameterization, Journal of Applied Mathematics, 2013 (2013) 9.

64. Tuba, M. ve Bacanin, N., Artificial Bee Colony Algorithm Hybridized with Firefly Algorithm for Cardinality Constrained Mean-Variance Portfolio Selection Problem, Applied Mathematics & Information Sciences, 8, 6 (2014) 2831-2844.

65. Zhou, G.D., Yi, T.H. ve Li, H.N., Sensor Placement Optimization in Structural Health Monitoring Using Cluster-in-Cluster Firefly Algorithm, Advances in Structural Engineering, 17, 8 (2014) 1103-1115.

66. Jain, A.K., Murty, M.N. ve Flynn, P.J., Data clustering: A review, Acm Computing Surveys, 31, 3 (1999) 264-323.

67. Xu, H.X. ve Tian, Z., An Optimal Spectral Clustering Approach Based on Cauchy-Schwarz Divergence, Chinese Journal of Electronics, 18, 1 (2009) 105-108.

68. Zhang, T., Ramakrishnan, R. ve Livny, M., BIRCH: an efficient data clustering method for very large databases, SIGMOD Rec., 25, 2 (1996) 103-114.

69. Guha, S., Rastogi, R. ve Shim, K., CURE: an efficient clustering algorithm for large databases, Proceedings of the 1998 ACM SIGMOD international conference on Management of data., (1998) 73-84.

70. Guha, S., Rastogi, R. ve Shim, K., Rock: A robust clustering algorithm for categorical attributes, Information Systems, 25, 5 (2000) 345-366.

71. Karypis, G., Han, E.-H. ve Kumar, V., Chameleon: Hierarchical Clustering Using Dynamic Modeling, IEEE_Computer, 32, 8 (1999) 68-75.

72. Karypis, G., Han, E.-H. ve Kumar, V., Chameleon: Hierarchical Clustering Using Dynamic Modeling, IEEE_Computer, 32, 8 (1999) 68-75.

73. Boudaillier, E. ve Hebrail, G., Interactive interpretation of hierarchical clustering, Intelligent Data Analysis, 2, 1 (1998) 229-244.

74. Rodrigues, P.P., Gama, J. ve Pedroso, J., Hierarchical Clustering of Time-Series Data Streams, Ieee Transactions on Knowledge and Data Engineering, 20,5 (2008) 615-627.

75. Ertoz, L., Steinbach, M. ve Kumar, V., Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data, Proceedings of the Third Siam International Conference on Data Mining, 47 (2003) 47-58.

76. MacQueen, J.B., Kmeans some methods for classification and analysis of multivariate observations, 5th Berkeley Symposium on Mathematical Statistics and Probability, 1 (1967) 281–297.

77. Park, H.-S. ve Jun, C.-H., A simple and fast algorithm for K-medoids clustering, Expert Systems with Applications, 36, 2, Part 2 (2009) 3336-3341.

78. Kaufman.L, R.P., Partitioning around medoids (program pam).Finding groups in data: an introduction to cluster analysis, Wiley, New York, 1990.

79. Kaufman L, R.P., Finding groups in data: an introduction to cluster analysis, John Wiley & Sons, Antwerp, 2005.

80. Birant, D. ve Kut, A., ST-DBSCAN: An algorithm for clustering spatial–temporal data, Data & Knowledge Engineering, 60, 1 (2007) 208-221.

81. Rasmussen, C.E., The Infinite Gaussian Mixture Model, In Advances in Neural Information Processing Systems, Haziran 2000, MIT Press, Bildiriler Kitabı, 12, 554-560.

82. Güngör, E. ve Özmen, A., Distance and density based clustering algorithm using Gaussian kernel, Expert Systems with Applications, 69 (2017) 10-20.

83. Viswanath, P. ve Suresh Babu, V., Rough-DBSCAN: A fast hybrid density based clustering method for large data sets, Pattern Recognition Letters, 30, 16 (2009) 1477-1488.

84. Bezdek, J.C., Ehrlich, R. ve Full, W., Fcm - the Fuzzy C-Means Clustering-Algorithm, Computers & Geosciences, 10, 2 (1984) 191-203.

85. Nasibov, E.N. ve Ulutagay, G., Robustness of density-based clustering methods with various neighborhood relations, Fuzzy Sets and Systems, 160, 24 (2009) 3601-3615.

86. Himmelspach, L. ve Conrad, S., Density-based clustering using fuzzy proximity relations, 2011 Annual Meeting of the North American Fuzzy Information Processing Society, Mart 2011, Texas , , Bildiriler Kitabı, 1-6.

87. Chen, W.Y., Song, Y.Q., Bai, H.J., Lin, C.J. ve Chang, E.Y., Parallel Spectral Clustering in Distributed Systems, Ieee Transactions on Pattern Analysis and Machine Intelligence, 33, 3 (2011) 568-586.

88. Breunig, M.M., Kriegel, H.P., Ng, R.T. ve Sander, J., LOF: Identifying density-based local outliers, Sigmod Record, 29, 2 (2000) 93-104.

89. Nanda, S.J. ve Panda, G., A survey on nature inspired metaheuristic algorithms for partitional clustering, Swarm and Evolutionary Computation, 16,Supplement C (2014) 1-18.

90. Falkenauer, E., Genetic Algorithms and Grouping Problems, John Wiley & Sons, Inc. New York, 1998.

91. Karaboga, D. ve Ozturk, C., A novel clustering approach: Artificial Bee Colony (ABC) algorithm, Applied Soft Computing, 11, 1 (2011) 652-657.

92. Shelokar, P., Jayaraman, V.K. ve Kulkarni, B.D., An ant colony approach for clustering, Analytica Chimica Acta, 509, 2 (2004) 187-195.

93. Chen, C.-Y. ve Ye, F., Particle swarm optimization algorithm and its application to clustering analysis, 2012 Proceedings of 17th Conference on Electrical Power Distribution Networks, Mayıs 2012 , Tehran, , Bildiriler Kitabı, 789-794.

94. Han, X., Quan, L., Xiong, X., Almeter, M., Xiang, J. ve Lan, Y., A novel data clustering algorithm based on modified gravitational search algorithm, Engineering Applications of Artificial Intelligence, 61 (2017) 1-7.

95. Senthilnath, J., Omkar, S. ve Mani, V., Clustering using firefly algorithm: performance study, Swarm and Evolutionary Computation, 1, 3 (2011) 164-171.

96. Yelghi, A., Kalte, T. ve Yelghi, A., Combining Genetic with RDPTA Algorithm for the Prolonging of the Lifetime of Wireless Sensor Networks, Proceedings of 2015 International Conference on Image Processing, Production and Computer Science (ICIPCS'2015), Haziran 2015, İstanbul, 104-111.

97. Kuo, R.J., Ho, L.M. ve Hu, C.M., Integration of self-organizing feature map and K-means algorithm for market segmentation, Computers & Operations Research, 29, 11 (2002) 1475-1493.

98. Agrawal, R., Gehrke, J., Gunopulos, D. ve Raghavan, P., Automatic subspace clustering of high dimensional data, Data Mining and Knowledge Discovery, 11, 1 (2005) 5-33.

99. Wang, Y., Xiang, Y., Zhang, J., Zhou, W. ve Xie, B., Internet traffic clustering with side information, Journal of Computer and System Sciences, 80, 5 (2014) 1021-1036.

100. Wagner, S. ve Wagner, D., Comparing clusterings: an overview, Universität Karlsruhe, Fakultät für Informatik, 2007.

101. Avsiuk, I.N., On the tidal force, 3, Soviet Astronomy Letters, 1977.

102. Sumich, J.L., An Introduction to the Biology of Marine Life, Sixth Edition, 30-35, IA: Wm. C. Brown, Dubuque, 1996.

103. Thurman, H.V., Introductory Oceanography, seventh 252-276, NY: Macmillan, New York, 1994.

104. Tizhoosh, H.R., Opposition-Based Learning: A New Scheme for Machine Intelligence, International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), Kasım 2005, Vienna , Bildiriler Kitabı, 695-701.

105. Fister Jr, I., Yang, X.-S., Fister, I., Brest, J. ve Fister, D., A brief review of nature-inspired algorithms for optimization, Elektrotehniški vestnik, 80, 3 (2013) 1-7 .

106. Surjanovic, S., Bingham, D. Virtual Library of Simulation Experiments: Test Functions and Datasets, www.sfu.ca/~ssurjano, 20 Ocak 2017 .

107. Debnath, M., Tripathi, P.K. ve Elmasri, R., K-DBSCAN: Identifying Spatial Clusters with Differing Density Levels, 2015 International Workshop on Data Mining with Industrial Applications (DMIA), Eylül 2015, Asunción, Bildiriler Kitabı , 51-60.

ÖZGEÇMİŞ

1986 yılında İran da Golestan ilinde Gonbede Kavus şehrinde doğan Aref YELGHİ, tüm öğrenimini Golestan ilinde Bandar Türkmen şehrinde tamamladı. Bilgisayar bölümünde ve aynı zamanda okulda bulunduğu bölümlerde onur öğrencisi olarak ödüllendirilmiştir. 2002 yılında, Teknik ve Meslek öğrenim Gorgan merkezinde Bilgisayar Mühendislik Bölümü'nü ön lisansı kazandı ve 2004 yılında mezun oldu. Bilgisayar yazılımda (Uygulamalı Bilim ve Teknoloji) 2006-2008 Mirdamad Özel Üniversitesinde lisansını tamamladı. 2005 ve 2009 yılında askerlik görevini yapmıştır. 2010 yılında Bilgisayar Bilimleri (Bilgisayar sistemde) İslamik Azat üniversitesine Sari şehirinde başladı ve 2012 de bitirdi. 2011'den 2013'e kadar birkaç üniversitede öğretim görevlisi olarak görevini yapmıştır. 2013 yılında Bilgisayar Mühendisliği bölümünde başlanmış ve 2014-2015 yılında Linnaeus Üniversitesi'nde bir dönemlik Erasmus programına katılmıştır. Doktora çalışma süresince, Yurtdışı Türkler ve Akraba Topluluklar Başkanlık (YTB) tarafından maddi destek almıştır.

Yayımlar ve Konferanslar

Yelghi, A. ve Köse, C., A modified firefly algorithm for global minimum optimization, Applied Soft Computing, 62,Supplement C (2018) 29-44.

Yelghi, A., Kalte, T. ve Yelghi, A., Combining Genetic with RDPTA Algorithm for the Prolonging of the Lifetime of Wireless Sensor Networks 2015.