



**SOSYAL ÖRÜMCEK ALGORİTMASIYLA
SOSYAL AĞLARDA
DUYGU ANALİZİ**

Vahtettin Cem BAYDOĞAN

**Yüksek Lisans Tezi
Yazılım Mühendisliği Anabilim Dalı
Danışman: Doç. Dr. Bilal ALATAŞ
OCAK-2018**

**T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**SOSYAL ÖRÜMCEK ALGORİTMASIYLA SOSYAL AĞLARDA
DUYGU ANALİZİ**

YÜKSEK LİSANS TEZİ

Vahtettin Cem BAYDOĞAN

(162137120)

**Tezin Enstitüye Verildiği Tarih : 26.01.2018
Tezin Savunulduğu Tarih : 26.01.2018**

Tez Danışmanı: Doç. Dr. Bilal ALATAŞ (F.Ü.)

Diğer Jüri Üyeleri: Doç. Dr. Erkan TANYILDIZI (F.Ü.)

Yrd. Doç. Dr. Ulaş Baran BALOĞLU (M.Ü.)

OCAK-2018

ÖNSÖZ

Yüksek lisans tez çalışma süreci boyunca, değerli görüşleri ile beni yönlendiren, her konuda desteğini esirgemeyen, kıymetli tecrübelerinden faydalandığım danışman hocam Sayın Doç.Dr. Bilal ALATAŞ'a teşekkürü borç bilirim.

Sadece öğrenim hayatım değil, her koşulda fedakârlığını ve desteğini esirgemeyen her zaman yanımda olan değerli aileme, son olarak da sadece tez dönemim boyunca değil, her zaman yanımda olup, bana destek olan eşim Merve PARLAK BAYDOĞAN'a sonsuz teşekkürlerimi sunarım.

V. Cem BAYDOĞAN
Elazığ, 2018

İÇİNDEKİLER

ÖNSÖZ.....	III
İÇİNDEKİLER.....	IV
ÖZET.....	V
SUMMARY.....	VI
ŞEKİLLER LİSTESİ.....	VII
TABLolar LİSTESİ.....	VIII
KISALTMALAR.....	IX
SEMBOLLER LİSTESİ	X
1. GİRİŞ	1
1.1. Tezin Amacı.....	4
2. SOSYAL ÖRÜMCEK OPTİMİZASYON ALGORİTMASI (SSA)	5
2.1. Algoritmanın Biyolojik Yapısı.....	5
2.2. Matematiksel Model ve Optimizasyon Algoritması	6
3. GÜNCEL OPTİMİZASYON ALGORİTMALARI	12
3.1. Karınca Aslanı Optimizasyon Algoritması (ALO)	12
3.2. Güve Alevi Optimizasyon Algoritması (MFO)	16
3.3. Balina Optimizasyon Algoritması (WOA).....	19
3.3.1. Ağ Kuşatma Denklemi.....	20
3.3.2. Kabarcık Ağ Saldırı Yöntemi	21
4. DENEYLER VE SONUÇLARI	24
4.1. Kalite Testi Fonksiyonları.....	24
4.2. Değerlendirme ve Sonuç	31
5. SOSYAL AĞLARDA DUYGU ANALİZİ.....	32
5.1. Metin Sınıflandırma	32
5.2. Metin Sınıflandırma Teknikleri.....	34
5.2.1. Denetimli Makine Öğrenmesi	35
5.2.2. Denetimsiz Makine Öğrenmesi.....	37
6. DUYGU ANALİZİ UYGULAMA ÇALIŞMASI	38
6.1. Verilerin Toplanması	38
6.2. Makine Öğrenmesi Tekniklerinin Kullanılması.....	38
6.3. Sosyal Örümcek Algoritması (SSA)'nın Kullanılması	47
6.3.1. SSA'nın Başlangıç Parametrelerinin Ayarlanması	47
6.3.2. Uygunluk Fonksiyonunun Ayarlanması	48
6.4. Değerlendirme ve Sonuç	51
7. SONUÇ.....	58
KAYNAKÇA	59
ÖZGEÇMİŞ.....	64

ÖZET

Sürü zekâsı, böcek veya diğer hayvan sürülerinin toplu şekilde davranışlarının araştırıldığı bir çalışma alanıdır. Sürü zekâsına dayalı algoritmalar, karmaşık optimizasyon problemlerinin geniş bir yelpazede çözümlenmesi için ileri sürülmüştür. Bu çalışmada, Sosyal Örümcek Optimizasyon Algoritması (SSA) olarak adlandırılan yeni bir sürü zekâsı algoritması incelenmiştir. SSA örümceklerin işbirliği içerisindeki davranışlarının simülasyonuna dayanmaktadır. SSA'nın performansı, stabilitesi ve duyarlılığı Karınca Aslanı Optimizasyon Algoritması (ALO), Güve Alevi Optimizasyon Algoritması (MFO), Balina Optimizasyon Algoritması (WOA) ile karşılaştırılmıştır. Literatürde var olan kalite testi fonksiyonu kullanılarak algoritmaların başarısı incelenmiştir.

Bu çalışma kapsamında ayrıca veri paylaşımının en yoğun olduğu Twitter sosyal medya aracından elde edilen veriler üzerinde duygu analizi çalışması yürütülmüştür. Makine öğrenmesi yöntemleri kullanılarak yapılan duygu analizi çalışmasına ek olarak elde edilen veriler üzerinde literatürde ilk kez bir metasezgisel algoritmayla olan Sosyal Örümcek Algoritması kullanılarak duygu analizi yapılmıştır. Duygu analizi için kullanılan algoritmaların başarıları farklı metriklerle karşılaştırmalı olarak incelenmiştir.

Anahtar Kelimeler: Optimizasyon, Sürü Zekâsı, Metasezgisel Algoritmalar, Sosyal Örümcek Algoritması (SSA), Yapay Zekâ, Duygu Analizi, Sosyal Ağ Analizi

SUMMARY

SENTIMENT ANALYSIS IN SOCIAL NETWORKS WITH SOCIAL SPIDER OPTIMIZATION ALGORITHM

Swarm intelligence is a research area that investigates the behavior of insects or other animals herd with a collective manner of intelligence. Swarm intelligence algorithms are proposed for solving complex optimization problems in a wide range. In thesis study, a new algorithm for swarm intelligence called Social Spider Optimization (SSA) is analyzed. The SSA is based on the simulation of the spiders' behaviors in collaboration. The SSA performance, stability and sensitivity are compared with Ant Lion Optimization Algorithm (ALO), Moth Flame Optimization Algorithm (MFO), and Whale Optimization Algorithm (WOA). The success of the algorithm has been investigated using the existing benchmark functions in the literature.

Furthermore was a sentiment analysis study carried out on the data obtained from the Twitter social media tool where data sharing is the most intensive. In addition to the sentiment analysis with machine learning methods, Sentiment analysis was performed on the obtained data using Social Spider Algorithm with a metasequential algorithm for the first time in the literature. Finally the success of the algorithms used for sentiment analysis has been examined comparatively with different metrics.

Keywords: Optimization, Swarm Intelligence, Metaheuristic Algorithms, Social Spider Algorithm (SSA), Artificial Intelligence, Sentiment Analysis, Social Network Analysis

ŞEKİLLER LİSTESİ

Sayfa No

Şekil 2.1.	<i>Vibc_i</i> titreşimin oluşumu (a), <i>Vibb_i</i> titreşiminin oluşumu (b) ve <i>Vibf_i</i> titreşiminin oluşumu (c)	8
Şekil 2.2.	SSA'nın akış diyagramı	11
Şekil 3.1.	Yumuşak kumda karınca aslanı tarafından yapılan tuzaklar	12
Şekil 3.2.	Karınca aslanının avlanma davranışı	12
Şekil 3.3.	ALO'nun akış diyagramı	16
Şekil 3.4.	Güvelerin yakın ışık kaynağı etrafındaki hareketi	16
Şekil 3.5.	Kambur balinaların balon ağları ile besleme davranışı	20
Şekil 4.1.	Kalite testi fonksiyonu	26
Şekil 4.2.	ALO, MFO, WOA ve SSA'nun ortalama uygunluk değerleri için yakınsama grafikleri	29
Şekil 4.3.	ALO, MFO, WOA ve SSA'nun en iyi uygunluk değerleri için yakınsama grafikleri.....	31
Şekil 5.1.	Metin madenciliğinin fonksiyonları.....	33
Şekil 5.2.	Metin Madenciliği oluşturan temel adımlar.....	33
Şekil 5.3.	Makine Öğrenmesi Teknikleri	35
Şekil 6.1.	KNIME kullanılarak oluşturulmuş duygu analizi çalışması.....	39
Şekil 6.2.	“File Reader” düğüm çıktısı	40
Şekil 6.3.	“Preprocessing” düğümünün alt düğümleri	41
Şekil 6.4.	“N Chars Filter” yapılandırma ekranı	41
Şekil 6.5.	“Document Vector” ile verilerin sayısallaştırılması	43
Şekil 6.6.	Karar ağaçları yapılandırma ekranı.....	44
Şekil 6.7.	k en yakın komşu algoritması yapılandırma ekranı	45
Şekil 6.8.	Destek vektör makinaları yapılandırma ekranı	46
Şekil 6.9.	Naive Bayes yapılandırma ekranı	47

TABLÖLER LİSTESİ

	<u>Sayfa No</u>
Tablo 2.1. SSA'nın sözde kodu	10
Tablo 3.1. ALO'nun sözde kodu	15
Tablo 3.2. I, P, T fonksiyonları.....	17
Tablo 3.3. MFO'nun sözde kodu.....	19
Tablo 3.4. WOA'nın sözde kodu.....	23
Tablo 4.1. Kalite testi fonksiyonları	24
Tablo 4.2. ALO, MFO, WOA, SSA'nın karşılaştırılması	27
Tablo 6.1. Sosyal Örümcek Algoritmasının başlangıç parametreleri	47
Tablo 6.2. Doğruluk Analizi yöntemindeki durum uzayı.....	49
Tablo 6.3. KNIME Doküman matrisi.....	50
Tablo 6.4. Sosyal Örümcek Algoritması'yla oluşan temsili pozitif örümcek modeli örneği	50
Tablo 6.5. Sosyal Örümcek Algoritması'yla oluşan temsili negatif örümcek modeli örneği.....	50
Tablo 6.6. 1. deney seti için kullanılan algoritmaların karşılaştırılması.....	51
Tablo 6.7. 1. deney seti için algoritmaların başarımların sıralaması.....	52
Tablo 6.8. 2. deney seti için kullanılan algoritmaların karşılaştırılması.....	52
Tablo 6.9. 2. deney seti için algoritmaların başarımların sırası	52
Tablo 6.10. 3. deney seti için kullanılan algoritmaların karşılaştırılması	53
Tablo 6.11. 3. deney seti için algoritmaların başarımların sırası	53
Tablo 6.12. 4. deney seti için kullanılan algoritmaların karşılaştırılması.....	54
Tablo 6.13. 4. deney seti için algoritmaların başarımların sırası	54
Tablo 6.14. 5. deney seti için kullanılan algoritmaların karşılaştırılması.....	55
Tablo 6.15. 5. deney seti için algoritmaların başarımların sırası	55
Tablo 6.16. 6. deney seti için kullanılan algoritmaların karşılaştırılması	56
Tablo 6.17. 6. deney seti için algoritmaların başarımların sırası	56
Tablo 6.18. Tüm deney setleri için kullanılan algoritmaların aritmetik ortalaması.....	57
Tablo 6.19. Tüm deney setleri için kullanılan algoritmaların başarımların sırası	57

KISALTMALAR

ALO	: Karınca Aslanı Optimizasyon Algoritması
DN	: Doğru Negatif
DP	: Doğru Pozitif
k-NN	: k En Yakın Komşu Algoritması
MFO	: Güve Alevi Optimizasyon Algoritması
ML	: Makina Öğrenmesi
SSA	: Sosyal Örümcek Optimizasyon Algoritması
SVM	: Destek Vektör Makineleri
WOA	: Balina Optimizasyon Algoritması
YN	: Yanlış Negatif
YP	: Yanlış Pozitif

SEMBOLLER LİSTESİ

N	: Popülasyon Sayısı
N_f	: Dişi örümcek sayısı
N_m	: Erkek örümcek sayısı
V	: Örümcek ağındaki titreşim
d_{ij}	: i . ve j . örümcekler arasındaki öklid uzaklığı
i	: Maksimum iterasyon sayısı
M_{ant}	: Her bir karıncanın pozisyonunu tutan matris
m_i^{t+1}	: $t+1$. adım zamanında i . erken örümceğin pozisyonunu
r	: $[0, 1]$ aralığında rastgele bir sayı
f_i^{t+1}	: $t+1$. adım zamanında i . dişi örümceğin pozisyonu için standart
T	: Maksimum iterasyon sayısını
R_A^t	: t . iterasyonda rulet tekerleği tarafından seçilen karınca aslanı
S	: Spiral fonksiyon

1. GİRİŞ

Karınca kolonileri, balık sürüleri, arılar ve termit sürüleri gibi birçok hayvan veya böcek gruplarının doğada toplu halde göstermiş oldukları akıllı davranışlar araştırmacıların ilgisini çekmiştir. Böceklerin veya hayvanların toplu davranışı sürü davranışı olarak adlandırılır. Bireylerin karmaşık ve etkileşim yoluyla aralarında sürülerin toplu davranışını ele alan bu yapay zekâ dalı sürü zekâsı olarak adlandırılmaktadır. Bonabeau “Herhangi bir algoritma tanımlarken veya yaygın bir problemi çözme cihazı tasarlama girişimde bulunurken diğer hayvan toplumlarının ve sosyal böcek kolonilerinin toplu davranışlarından esinlenilir.” diyerek sürü zekâsını tanımlamıştır [1]. Sürü zekâsının ölçeklenebilirlik, hata toleransı, adaptasyon, hız, modülerlik, özerklik ve paralellik gibi birçok avantajı bulunur [2].

Sürü zekâsının anahtar bileşenleri öz örgütlenme ve iş gücü bölümüdür. Kendi kendini örgütleyen bir sistemde, kapalı birimlerin her biri ayrı ayrı yerel uyaranlara yanıt verir ve merkezi denetimden kaçınan iş gücünün ayrılması ile küresel görevi gerçekleştirmek için birlikte hareket ederler. Böylece tüm sistem iç ve dış değişikliklere uyum sağlamış olur. Çeşitli sürü algoritmaları doğada böcek veya hayvan gruplarının davranışlarını taklit ederek deterministik kurallarla veya rastgele kombinasyon oluşturulmasıyla geliştirilmiştir.

Parçacık Sürü Optimizasyonu (PSO) algoritması kuş akını ve balık eğitimi gibi sosyal davranışları [3], Yapay Arı Kolonisi (ABC) arı kolonilerinin kooperatif davranış şekli tekniğini [4], Bakteriye Toplayıcılık Optimizasyon Algoritması bakteri sosyal toplayıcı davranışını (BFOA) [5], Krill Sürüsü (KH) yöntemi krill bireylerinin sürü davranışlarının simülasyonunu [6], Firefly (FF) yöntemi ateşböceği böceklerin çiftleşme davranışını [7] ve Guguk Kuşu Optimizasyon Algoritması (COA) guguk kuşlarının yaşam tarzı gibi sosyal davranışları temelde esas alır [8].

PSO ve ABC karmaşık optimizasyon problemlerini çözmek için en popüler algoritmalar olmalarına rağmen, bu algoritmaların erken yakınsama ve yerel minimum noktayı aşmada zorluk çekmek gibi zayıflıkları bulunmaktadır [9]. Pozisyonların tek tek değiştirilmesinin sebebi operatörlerle ilişkilidir. Bu yüzden bu tür algoritmaların evrimi sırasında, bir sonraki iterasyon için her bir ajanın pozisyonu şimdiye kadar gördüğü en iyi parçacığın konumuna doğru bir çekicilik göstererek (PSO durumunda) veya diğer rastgele seçilmiş bireylere yönelik olarak güncellenir (ABC durumunda). Algoritma geliştikçe bu tür davranışlar tüm nüfusun en

iyi parçacık etrafında yoğunlaşmasına veya kontrolsüz bir şekilde uzaklaşmasına neden olur [10].

Sosyal böceklerin ilginç ve egzotik toplu davranışları araştırmacıların yıllarca dikkatini çekmiştir. Sosyal örümcekler, sosyal böcek temsilinin bir örneğidir. Örümcek türünün üyelerini karmaşık iş birliğine dayalı davranış kümesi ile sosyal örümcekleri muhafaza eder. Çoğu örümcekler kendi türünün diğer üyeleri karşı bile agresifken, sosyal örümcekler genellikle koloniler olarak adlandırılan uzun ömürlü topluluklar oluşturup gruplar halinde yaşamak için eğilim gösterirler [11]. Sosyal örümcek kolonisinde ki her üye kendi cinsiyetine göre örneğin avlanma, çiftleşme, ağ tasarımı ve sosyal etkileşim gibi çeşitli görevleri yürütür. Burada ağ, koloninin önemli bir parçasıdır. Çünkü ağ tüm bireyler için ortak bir yaşam alanı olarak kullanılır ve aynı zamanda bireyler arasında bir iletişim kanalı olarak da kullanılır. Dahası önemli bilgiler ağ üzerinden küçük titreşimler tarafından iletilir [12].

Bu çalışmada ilk önce yeni bir sürü zekâsına dayalı algoritma olan SSA detaylı bir şekilde incelenmiştir. SSA sosyal örümceklerin iş birliğine ve davranışlarına dayanmaktadır. Bu algoritmada bireylerin birbirleriyle iş birliği ve etkileşim içinde olduğu koloninin biyolojik yasaları esas alındığı görülmektedir. Algoritma, iki farklı arama ajanını (örümcekler) göz önüne alır: Dişi ve erkekler. Cinsiyete bağlı olarak, bir koloninin içindeki her bireyin farklı iş birliğine dayalı simülasyonu bir dizi evrimsel operatör tarafından ifade edilmektedir. Önerilen yaklaşımın mevcut sürü zekâsına dayalı algoritmalarından en temel farkı her bir bireyin cinsiyeti dikkate alınarak bir modellemenin yapılmış olmasıdır.

Bu çalışmaya adını veren duygu analizi işlemi çalışmanın ana temasını oluşturmaktadır. 2000 yılı başından bu yana, duygu analizi, doğal dil işleme ve sosyal ağ analistleri tarafından en aktif araştırma alanlarından biri haline gelmiştir. Ayrıca, bu konuyla ilgili olarak veri madenciliği, web madenciliği ve metin madenciliği konuları da geniş çapta incelenmektedir. Dahası, duygu analizi yönteminin, bir bütün olarak iş dünyasına ve topluma verdiği önem nedeniyle bilgisayar bilimlerinden yönetim bilimlerine ve sosyal bilimlerden ekonomi bilimine kadar birçok alana yayılmıştır.

Son yıllarda, duygu analizi konu alan endüstriyel faaliyetler de hızla gelişmektedir. Bu alanda çok sayıda yeni teşebbüs ortaya çıkmıştır. Birçok büyük şirket tesis içi hizmetlerinin kalitesini ölçmek için kendi duygu analizi sistemlerini geliştirerek iş dünyasında ve sosyal alanda farkındalık yaratma yolunu açmışlardır.

Gerçek dünyada, şirketler ve kuruluşlar her zaman ürün ve hizmetleri hakkında tüketici veya kamuoyunun düşünce ve yorumlarını almayı istemektedir. Bireysel tüketiciler ise, bir ürünün satın alınmadan önceki mevcut kullanıcılarının görüşlerini veya başkalarının siyasi seçimlerle ilgili bir oy vermeden önce siyasi adaylarla ilgili görüşlerini bilmek istemektedir. Kamuoyu ve tüketici görüşlerini edinmek, uzun zamandır pazarlama, halkla ilişkiler ve siyasi kampanya şirketleri için büyük bir iş yükü haline gelmiştir. Sosyal ağlar yardımıyla (örneğin eleştiriler, forum tartışmaları, bloglar, mikro bloglar, Twitter, yorumları ve sosyal paylaşım sitelerinde yayınlanan mesajlar), sosyal medyanın gücünün arması sonucu, bireyler ve organizasyonlar karar vermede bu medyalardaki içeriği dikkate alma zorunluluğu kaçınılmaz hale gelmiştir.

Gerçek hayatta kullanılan uygulamaların yanı sıra, duygu analizi alanında araştırma makaleleri de yayınlanmıştır. Örneğin, Leilei ve arkadaşları seçim sonuçlarını tahmin etmek için Twitter verilerini kullanmıştır [13]. Bernardo, Mahesh ve Venetis tarafından filmlerin gişe hasılat gelirlerini tahmin etmek için Twitter verileri, film görüşleri ve bloglar kullanılarak farklı duygu analizi çalışmaları yürütülmüştür. [14-15-16]. Özel yaptığı çalışmada web tabanlı anket arayüzü olan “Limesurvey” adlı yazılım aracını kullanarak bir anket çalışması yapmıştır. Bu çalışmada çalışanların Twitter kullanmalarının şirket profiline etkisi analiz edilmiştir. Anket linkini 10 farklı Twitter hesabı tarafından tweet ve retweet ederek iki bin Twitter kullanıcısına ulaştırmıştır. Elde ettiği verileri SPSS 15.0 yazılım aracılığıyla istatistik analizine tabi tutmuştur [17]. Nizam ve Akın sosyal medyada makine öğrenmesi yöntemlerinden denetimli öğrenme yaklaşımı kullanılarak duygu analizi çalışması yapmıştır. Bu çalışmayı yaparken bazı gıda firmalarının çeşitli ürünlerine ait yapılan yorumlardan oluşan veri setini Twitter üzerinde el yordamı ile elde etmiştir [18]. Oğuz ve ekibi grip benzeri hastalıkların sosyal medya aracılığıyla saptanabilirliğini araştırmak için Twitter iletilerini ve gazete sitelerini kullanmıştır. Sosyal medya için geliştirilmiş ücretsiz Topsy gerçek zamanlı arama motoru uygulaması kullanarak Twitter iletilerini elde etmiştir [19]. Yazan ve Üsküdarlı sosyal ağlar üzerinden deprem tespiti yapmak için Twitter tarafından geliştirilip ücretsiz olarak kullanıma sunulan Twitter arabirimini "Streaming API"yi veri elde etmek için kullanmıştır [20]. Sütçü ise sosyal medyanın gazeteleri nasıl etkilediğini yaptığı çalışmada incelemiştir. Bu çalışmayı yaparken otomatik veri setinin oluşturulması araştırmasının doğruluğunu azaltacağı düşünerek verilerini elle gazetelerin Twitter sayfalarından toplama yönetimine başvurmuştur [21].

1.1. Tezin Amacı

Yapılan bu çalışmada ilk olarak SSA üç farklı sezgisel algoritma (ALO, MFO, WOA) ile aynı şartlar altında karşılaştırılmıştır. Tek modlu ve çok modlu kalite testi fonksiyonları ile gerçekleştirilen bu testin sonuçları ve detayları ayrıntılı bir şekilde tablolar ve grafikler aracılığıyla raporlanmıştır. Test için seçilen algoritmaların biyolojik yapıları ve bu algoritmaların matematiksel modelleri ayrıca incelenmiştir. Çalışmanın diğer bölümünü literatürde ilk defa bir metasezgisel algoritma olan SSA kullanılarak duygu analizi çalışması oluşturmaktadır. Yapılan bu çalışmada Twitter verileri üzerinde SSA kullanılarak yeni bir model önerilmiştir. SSA'nın başarımı literatürde var olan makine öğrenmesi yöntemleriyle karşılaştırılarak sonuçlar ayrıntılı bir şekilde incelenmiştir. Önerilen yöntemin iyileştirilmesi adına çıkarımlarda bulunulmuştur.

Yapılan bu tez çalışmasının, duygu analizi çalışmasına optimizasyon bakış açısıyla yaklaşıldığı için literatürde referans bir çalışma olarak değerlendirileceği; metin madenciliği, sosyal ağ analistleri ve özellikle optimizasyon çalışmaları yürüten araştırmacıların dikkatini çekeceği öngörülmektedir.

2. SOSYAL ÖRÜMCEK OPTİMİZASYON ALGORİTMASI (SSA)

2.1. Algoritmanın Biyolojik Yapısı

Sosyal böcek topluluğu, bir dizi kısıtlamalar seti içinde kendiliğinden organize olan karmaşık bir iş birliği sistemidir. İş birlikçi gruplar içerisinde grup üyeleri arasındaki görev özelleştirmesine izin verme, kaynakların korunması ve çevresinin kullanılması gibi davranışlar gözlemlenmiştir [22]. Entegre edilmiş bir birim olarak sosyal böcek koloni işlevi, dağıtılmış tarzda işlem yapma yeteneğine sahip olmakla beraber aynı zamanda küresel projenin muazzam yapısını inşa etme görevini üstlenmektedir. Sosyal böcekler içinde küresel sınırlar, üyeler arasındaki iç etkileşimin bir sonucu olduğunu kabul etmek gerekir [23].

Örümceklerin birkaç türünün sosyal davranışının derecelenmesi ve sergilenmesi belgelenmiş durumdadır [24]. Örümceklerin davranışı, tek başına ya da sosyal örümcekler olmak üzere iki formda genelleştirilebilir [25]. Bu sınıflandırma onların sergilediği iş birliği seviyelerine göre olur [26]. Bir tarafta, tek başına örümcekler benzer türlerin diğer bireyleriyle iletişim kurarak kendi ağlarını sürdürürken ve oluştururken, diğer tarafta sosyal örümcekler diğer grup üyelerine uzaysal yakın ilişkilerle toplumsal ağ üzerine birlikte kalan koloniler oluşturur [27].

Bir sosyal örümcek kolonisi üyeler ve ticari ağlar olmak üzere iki temel bileşenden oluşur. Üyeler, erkekler ve dişiler olmak üzere iki farklı kategoriye bölünür. Sosyal örümceklerin en ilginç karakteristiği, baskın bir şekilde dişi tabanlı popülasyona sahip olmalarıdır. Bazı çalışmalar erkek örümceklerin sayısının zar zor toplam kolonilerin yüzde 30'una ulaştığını gösterir [25]. Cinsine bağlı olarak kolonideki her bir üye sosyal iletişim ve gelişim için ticari ağlar inşa etme ve sürdürme gibi farklı aktiviteler gösterirler [12]. Üyeler arası etkileşim hem doğrudan hem dolaylı yoldan gerçekleşir [28]. Doğrudan etkileşim, üreme gibi sıvıların değişimi ya da vücutların etkileşimini ima eder. Dolaylı etkileşim ise, ticari ağ haberleşmenin ortası olarak kullanılır ki bilgi iletimi önemlidir. Bu da her bir koloni üyesine uygundur [27]. Küçük titreşimler olarak kodlanan bilgi, üyeler arasında karşılıklı koordinasyon için önemli bir unsurdur [12]. Titreşimler komşu üyelerin karakteristikleri ve soyulmuş tuzakların büyüklükleri gibi bir kaç mesajı kodlamak için koloni üyeleri tarafından kullanılır. Bu titreşimlerin yoğunluğu, onların üretildiği örümceklerin mesafelerine ve ağırlığına bağlıdır.

Koloni seviyeleri, küresel tarzda iş birliği içerisindeki tüm koloni üyeleri arasında var olan karmaşıklığa rağmen, iç etkileşimin sonucu olarak üretilir [29]. Böyle iç etkileşimler kolonide her bir örümcek tarafından takip edilen basit davranışsal dizisi içerir. Davranışsal kurallar sosyal etkileşim (işbirlikçi davranış) ve çiftleşme olarak iki farklı sınıfa bölünür [30].

Bir sosyal böcek olarak, örümcekler diğer koloni üyeleriyle işbirlikçi etkileşimler icra ederler. Bu işbirlikçi davranışlar örümceğin cinsiyetine bağlıdır. Titreşimler onları tetikleyen üyelerin mesafesine ve ağırlığına bağlı olduğu için, güçlü titreşimler hem büyük örümcekler tarafından hem de daha yakın komşu üyeler tarafından üretilir. Ne kadar büyük örümcek o kadar iyi düşünülmüş koloni üyesi demektir. Belirlenmiş bir üye üzerine etkileşim veya nefretin nihai kararı, ağırlık durumuna göre alınır, bu karar diğer rastgele olay ve merak yeniden üretim periyodu gibi birkaç faktör tarafından etkilenir [27].

Çiftleşme sadece kolonilerin hayatta kalmasını değil aynı zamanda üyeler arasında bilgi değişimine izin veren önemli bir işlemdir. Sosyal örümcek kolonisinde çiftleşme dominant erkekler ve dişi üyeler tarafından icra edilir. Bu durumda, dominant erkek örümcek belli bir alan içinde bir veya daha fazla dişi üye konumlanmış ise, yavrular üretmek için tüm dişilerle çiftleşir [31, 32].

2.2. Matematiksel Model ve Optimizasyon Algoritması

Cuevas tarafından önerilen sosyal örümcek optimizasyon algoritması sosyal örümceklerin davranışlarına dayanmaktadır [33]. Arama alanı ortak ağ olarak kabul edilir ve bir örümceğin pozisyonu optimum çözümü temsil eder. Sosyal örümceklerin ilginç bir özelliği dişi eğilimli nüfustur. Erkek örümceklerin sayısı toplam koloni üyelerinin neredeyse %30'larına ulaşmaktadır. Dişiler tüm nüfusta (N) 65-90% aralığında rastgele N_f seçilmektedir. Bu nedenle N_f Denklem 2.1'de gösterildiği gibi hesaplanır:

$$N_f = [(0.9 - rand * 0.25) * N] \quad (2.1)$$

Burada $rand$ 0 ile 1 arasında rastgele bir sayıdır. Erkek örümcekler sayısı N_m ise Denklem 2.2'deki gibi bulunmaktadır:

$$N_m = N - N_f \quad (2.2)$$

Her örümcek kendi çözümünün uygunluk değerine göre bir ağırlık alır. Bu ağırlık değeri Denklem 2.3'te hesaplanmıştır:

$$w_i = \frac{fitness_i - worst}{best - worst} \quad (2.3)$$

Burada $fitness_i$ her bir örümceğin uygunluk değeridir ve $i=1,2,3,...,N$ olmak üzere tüm örümceklerin pozisyonları değerlendirerek Denklem 2.3 kullanılarak hesaplanmıştır. *worst* ve *best* değerleri ise sırasıyla tüm popülasyon taranarak en iyi ve en kötü değerler olarak bulunmuştur.

Ortak ağ (arama alanı) koloni üyeleri arasında bilgi iletimi için bir mekanizma olarak kullanılır. Bilgi, küçük titreşimler olarak kodlanmış olup ağırlıklara ve onları üreten örümcek ile arasındaki mesafeye bağlıdır. Bu ilişki Denklem 2.4'te gösterilmiştir:

$$V_{ij}=w_j * e^{-d^2_{ij}} \quad (2.4)$$

Burada d_{ij} , i . ve j . örümcekler arasındaki öklid uzaklığıdır. Üç özel ilişki düşünülebilir:

(1) Titreşim $Vibc_i$ Denklem 2.5'e göre hesaplanır ve c tarafından aktarılan bilgilerin bir sonucu olarak i . birey tarafından algılanan birey iki önemli özelliğe sahiptir. En yakın i . üyesi ile karşılaştırıldığında daha yüksek bir ağırlığa sahiptir ($w_c > w_i$). Bu durum Şekil 2.1. (a)' da gösterilmiştir.

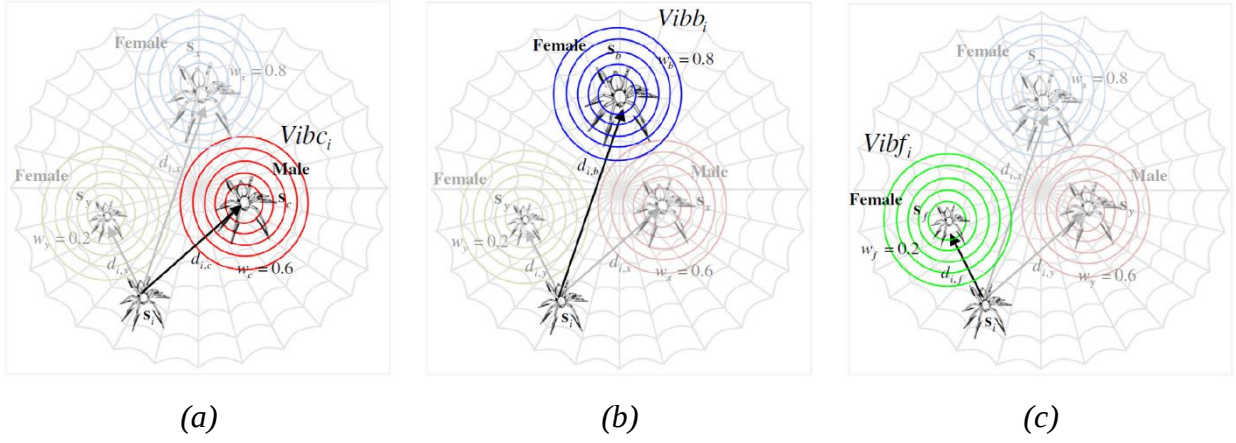
$$Vibc_i=w_c * e^{-d^2_{ij}} \quad (2.5)$$

(2) Titreşim $Vibb_i$ Denklem 2.6'ya göre hesaplanır ve b tarafından aktarılan bilgilerin bir sonucu olarak i . birey tarafından algılanan bireyin özelliği tüm nüfusun (N) en iyi uygunluk değerine sahip olmasıdır. Bu durum Şekil 2.1. (b)' de gösterilmiştir.

$$Vibb_i=w_b * e^{-d^2_{ij}} \quad (2.6)$$

(3) Titreşim $Vibf_i$ Denklem 2.7'e göre hesaplanır ve f tarafından aktarılan bilgilerin bir sonucu olarak i . birey tarafından algılanan birey i . bireye en yakın dışı bireydir. Bu durum Şekil 2.1. (c)' de gösterilmiştir.

$$Vibf_i=w_f * e^{-d^2_{ij}} \quad (2.7)$$



Şekil 2.1. $Vibc_i$ titreşimin oluşumu (a), $Vibb_i$ titreşiminin oluşumu (b) ve $Vibf_i$ titreşiminin oluşumu (c) [34]

Sosyal örümcekler cinsiyetlerine bağlı olarak diğer koloni üyeleriyle kooperatif etkileşim gerçekleştirir. Dişi örümceklerin ortak davranışını simüle etmek amacıyla, yeni bir operatör Eşitlik 2.8’de tanımlanmış. Çekme ya da itme hareketi ortak ağ üzerinden yayılan kendi titreşimlerine göre diğer örümcekler üzerine geliştirilir:

$$f_i^{t+1} = \begin{cases} f_i^{(t)} + \alpha V_{i,c}(s_c - f_i) + \beta V_{i,b}(s_b - f_i^{(t)}) + \\ + \gamma \left(\sigma - \frac{1}{2} \right) & \text{if } r_m < PF; \\ f_i^{(t)} - \alpha V_{i,c}(s_c - f_i^{(t)}) - \beta V_{i,b}(s_b - f_i^{(t)}) + \\ + \gamma \left(\sigma - \frac{1}{2} \right) & \text{if } r_m \geq PF, \end{cases} \quad (2.8)$$

Burada r_m , α , β , γ ve σ $[0,1]$ arasında tek tip rastgele sayılardır, f_i^{t+1} $t+1$. adım zamanında i . dişi örümceğin pozisyonu için standart olarak ifade edilir, s_c PF ’in bir anlık parametresini gösterir ve s_b sırasıyla tüm nüfusun en yüksek ağırlık ve en iyi i . yakın üyeyi temsil etmektedir.

Erkek örümcek nüfusu iki sınıfa ayrılır: baskın olan ve baskın olmayan. Baskın sınıf örümcek baskın olmayana kıyasla daha iyi uygunluk değerlerine sahiptir ve bu örümcekler ortak ağ merkezine en yakın kadın örümcek tarafından çekilir. Öte yandan, baskın olmayan erkek örümcekler, baskın erkekler tarafından israf edilen kaynaklardan yararlanmak için strateji olarak erkek nüfusunu C merkezinde yoğunlaştırma eğilimindedir. Bu durum Eşitlik 2.9’da gösterilmiştir:

$$m_i^{(t+1)} = \begin{cases} m_i^{(t)} + \alpha V_{i,f} (s_f - m_i^{(t)}) + \gamma \left(\sigma - \frac{1}{2} \right) & \text{if } w_{N_f+i} > w_{N_f+c}; \\ m_i^{(t)} + \alpha \left(\frac{\sum_{h=1}^{N_m} m_h w_{N_f+h}}{\sum_{h=1}^{N_m} w_{N_f+h}} \right) & \text{if } w_{N_f+i} \leq w_{N_f+c} \end{cases} \quad (2.9)$$

Burada s_f i . erkek örümceğe en yakın dişi örümceği temsil etmektedir ve m_i^{t+1} $t+1$. adım zamanında i . erken örümceğin pozisyonunu göstermektedir.

Çiftleşme sosyal örümcek kolonisinde baskın erkek ve dişi üyeler tarafından yapılır. Yarıçap olarak r dikkate alındığında (Denklem 2.10'a göre hesaplanır.) baskın erkek örümcek r içinde kadın üye bulursa çiftleşir ve yeni yavru oluşur.

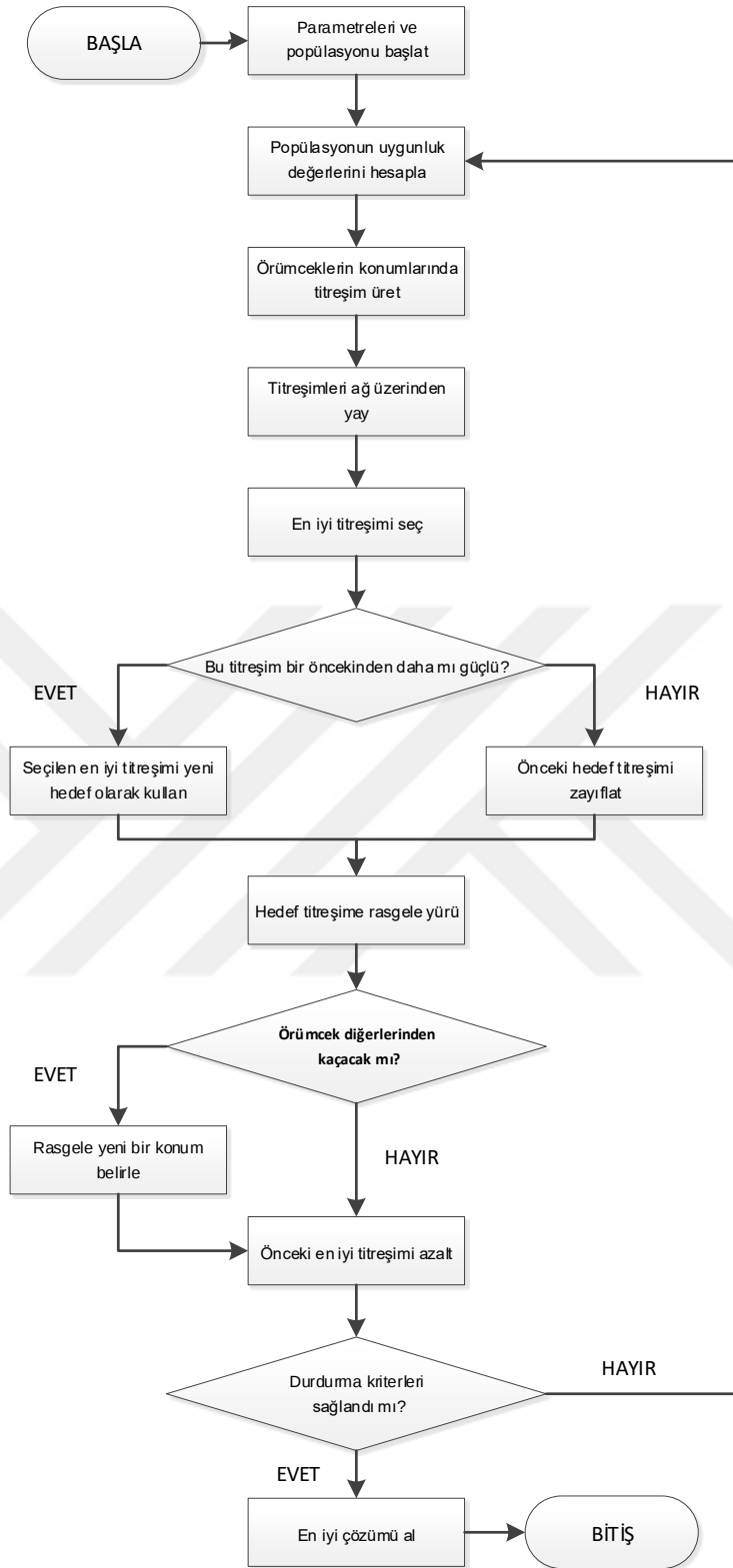
$$r = \frac{\sum_{j=1}^n p_j^{high} - p_j^{low}}{2n} \quad (2.10)$$

Burada n (optimize edilecek değişkenlerin sayısı) sorunun boyutu, p_j^{high} ve p_j^{low} sırayla üst ve alt sınırlar için standarttır. Yeni örümcek oluştuktan sonra koloninin en iyi örümceği ile karşılaştırılır. Yeni örümcek daha iyi ise, kötü örümcek yenisiyle değiştirilir. Sosyal örümcek algoritmasının genel yapısı Tablo 2.1'de gösterildiği gibidir.

Tablo 2.1. SSA'nın sözde kodu

```
prosedür SSA
SSA'nun parametre değerlerini ata
Örümcek popülasyonunu ve bunun için gerekli hafızayı oluştur
Her bir örümcek için  $v_{tar}$ 'i başlat
  while durdurma kriteri sağlayıncaya kadar çalıştır
    for each  $s$  in pop
       $s$ 'nin uygunluk değerini hesapla
       $s$ 'nin pozisyonu için  $a$  titreşimini üret
    end
    for each  $s$  in pop
      diğer örümcekler tarafından üretilen  $V$  titreşim değerini hesapla
       $V$ 'den en güçlü  $v_{best}$  titreşimi seç
      if  $v_{best}, v_{tar}$ 'dan daha büyük ise
         $v_{best}$ 'i  $v_{tar}$  olarak ata
      end
       $v_{tar}$ 'a doğru rastgele yürüyüş gerçekleştir.
       $[0,1)$  arasında  $a$  ve  $r$  rasgele sayıları üret
      if  $r < p_j$  ise
        örümceklerin pozisyonunu  $a$ 'ya ata
      end
       $v_{tar}$ 'ın yoğunluğunu azalt
    end
  end
end
```

Sosyal örümcek optimizasyon algoritmasının temel adımlarını ve bu algoritmanın genel akış yapısı Şekil 2.2'de gösterilmiştir.



Şekil 2.2. SSA'nın akış diyagramı

3. GÜNCEL OPTİMİZASYON ALGORİTMALARI

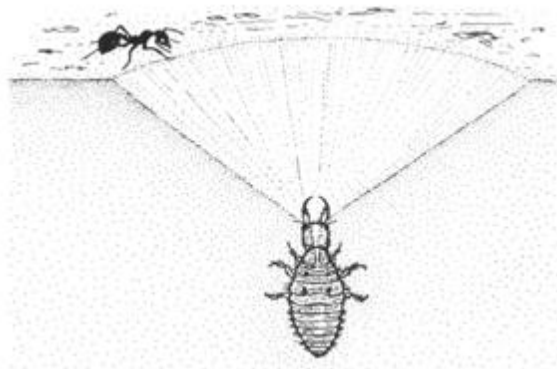
3.1. Karınca Aslanı Algoritması (ALO)

Karınca aslanları, Myrmelentidae familyasındaki böcek gruplarından biridir. Karınca aslanlarının yaşam döngüsünün iki ana safhası vardır: larva safhası ve yetişkin safhası. Larva karınca aslanı, Şekil 3.1’de görüldüğü gibi tuzak kurmak için iyi bir yer ararken kumda bıraktığı izler nedeniyle sık sık "doodlebug" olarak adlandırılır [35-36].

Avlanma süreci boyunca, karınca aslanı huni çukurlarını yumuşak kum içine alır ve daha sonra Şekil 3.2’de gösterildiği gibi çukurun tabanında sabırla bekler. Şekil 3.2’de kurban aşağı doğru kayarak karınca aslanı tarafından hızlıca yakalanır. Ya da, av tuzaktan kaçmaya yönelirse, karınca aslanı kurbanını çukurun dibine doğru kaydırmak için tuzağın kenarına kum atar. Larva çukurun altından bol kum atarak çukurun kenarlarını zayıflatır. Böylece tuzağın çökmesine ve avın karınca aslanına gelmesine neden olur [37-38].



Şekil 3.1. Yumuşak kumda karınca aslanı tarafından yapılan tuzaklar [36]



Şekil 3.2. Karınca aslanının avlanma davranışı [37]

Karınca aslanlarının ve karıncaların bu davranışlarından esinlenerek geliştirilen karınca aslanı optimizasyon algoritmasının (ALO) matematiksel modellenmesi kısmında birçok parametre vardır. Bu parametrelerden doğada gıdaları araştırırken karıncaların rastgele yürüyüşleri Denklem 3.1'deki gibi tanımlanabilir:

$$X(t) = [0, cumsum(2r(t_1) - 1), cumsum(2r(t_2) - 1), \dots, cumsum(2r(t_n) - 1)] \quad (3.1)$$

cumsum kümülatif toplamı hesaplar, n en fazla yineleme sayısını, t rastgele yürümenin basamağını gösterir ve $r(t)$ Denklem 3.2'de tanımlanan skolastik fonksiyondur:

$$r(t) = \begin{cases} 1 & \text{if } rand > 0.5 \\ 0 & \text{if } rand \leq 0.5 \end{cases} \quad (3.2)$$

Burada t rasgele yürümenin basamağını gösterir ve $rand$ $[0,1]$ aralığındaki uniform dağılımla üretilen rastgele bir sayıdır.

Karıncaların pozisyonu Denklem 3.3'te sunulmaktadır:

$$M_{ant} = \begin{bmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,d} \\ A_{2,1} & A_{2,2} & \dots & A_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n,1} & A_{n,2} & \dots & A_{n,d} \end{bmatrix} \quad (3.3)$$

Burada M_{ant} her bir karınca pozisyonu için matristir, $A_{i,j}$, j . karıncanın i . değişkeninin değerini, n , karıncanın sayısı ve d değişkenlerin sayısını sunar.

Her karıncanın uygunluk fonksiyonu Denklem 3.4'teki M_{OA} matrisinde saklanır:

$$M_{OA} = \begin{bmatrix} f([A_{1,1}, A_{1,2}, \dots, A_{1,d}]) \\ f([A_{2,1}, A_{2,2}, \dots, A_{2,d}]) \\ f([A_{n,1}, A_{n,2}, \dots, A_{n,d}]) \end{bmatrix} \quad (3.4)$$

Burada f , amaç fonksiyonudur.

$$M_{Antlion} = \begin{bmatrix} AL_{1,1} & AL_{1,2} & \dots & AL_{1,d} \\ AL_{2,1} & AL_{2,2} & \dots & AL_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ AL_{n,1} & AL_{n,2} & \dots & AL_{n,d} \end{bmatrix} \quad (3.5)$$

Denklem 3.5'de $M_{Antlion}$ herbir karınca aslanının matrisidir, $AL_{i,j}$, i . karınca aslanı değişkeninin güncel j . değeri, n karınca aslanı sayısı ve d 'de değişken sayısıdır.

Benzer şekilde, her karınca aslanının amaç fonksiyonunu Denklem 3.6'daki M_{OAL} matrisi kaydeder:

$$\begin{aligned}
& f([A_{1,1}, A_{1,2}, \dots, A_{1,d}]) \\
M_{OAL} &= f([A_{2,1}, A_{2,2}, \dots, A_{1,d}]) \\
& f([A_{n,1}, A_{n,2}, \dots, A_{n,d}])
\end{aligned} \tag{3.6}$$

Karncaların rasgele yürümelerini arama alanının içinde tutmak için Denklem 3.7 kullanılarak standartlaştırılır:

$$X_i^t = \frac{(X_i^t - a_i)(d_i - c_i^t)}{(d_i^t - a_i)} + c_i \tag{3.7}$$

Burada a_i i . değişkenin minimum rasgele yürümesi, c_i^t t . iterasyonda i . minimum değeri, d_i^t t . iterasyonda i . maksimum değeri temsil etmektedir.

Karınca aslanının tuzak çukurlarında karncaların yakalanmasının matematiksel modellenmesi Denklem 3.8 ve 3.9'da verilmiştir:

$$c_i^t = \text{Anatlion}_j^t + c^t \tag{3.8}$$

$$d_i^t = \text{Anatlion}_j^t + d^t \tag{3.9}$$

Burada c_i^t tüm değişkenlerin t . iterasyonunda minimum değerini, d_i^t tüm değişkenlerin t . iterasyonunda maksimum değerini ve Anatlion_j^t karınca aslanının t . iterasyonun j . seçilen pozisyonunu ifade eder.

Karınca aslanının avlanma yetenekleri rulet tekerleği seçimi ile modellenmiştir. Tuzağa düşmüş karınca, karınca aslanına doğru nasıl kayacağını açıklayan matematiksel model Denklem 3.10 ve Denklem 3.11'de şu şekilde verilir:

$$c^t = \frac{c^t}{I} \tag{3.10}$$

$$d^t = \frac{d^t}{I} \tag{3.11}$$

Burada hesaplanan I oranı Denklem 3.12'deki gibidir:

$$I = 10^w \cdot \frac{t}{T} \tag{3.12}$$

Burada t şimdiki iterasyonu, T maksimum iterasyon sayısını, w geçerli olan iterasyona bağlı olan sabittir ve Denklem 3.13'te gösterilmiştir.

$$w = \begin{cases} 2 & \text{if } t > 0.1T \\ 3 & \text{if } t > 0.5T \\ 4 & \text{if } t > 0.75T \\ 5 & \text{if } t > 0.9T \\ 6 & \text{if } t > 0.95T \end{cases} \tag{3.13}$$

Son olarak en iyi sonuç şu şekilde seçilir: Her iterasyonda en iyi karınca aslanı sonuç olarak kabul edilir. Bu, her karıncanın seçilen karınca aslanının çevresinde rastgele yürümesi anlamına gelir ve Denklem 3.14'te ifade edilmiştir:

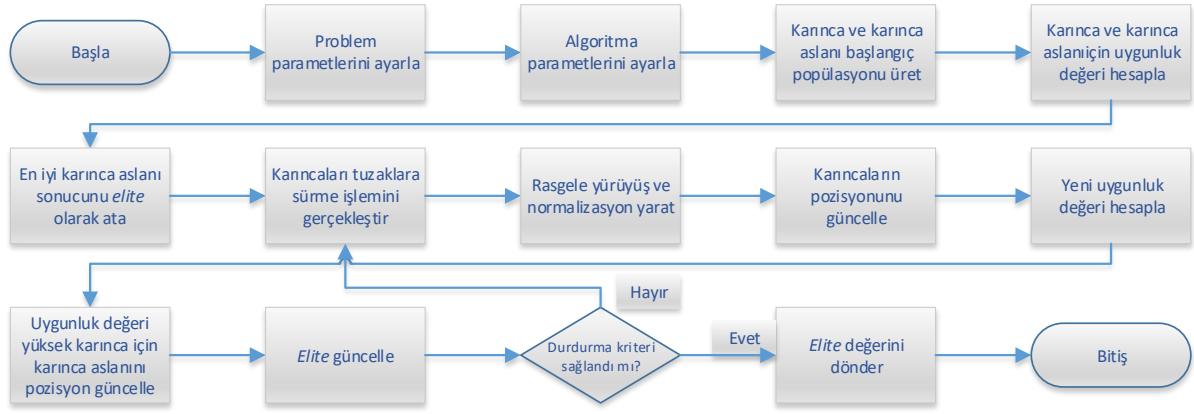
$$Ant_i^t = \frac{(R_A^t + R_E^t)}{2} \quad (3.14)$$

Burada R_A^t t. iterasyonda rulet tekerleği tarafından seçilen karınca aslanının rasgele yürümesi ve R_E^t t. iterasyonda en iyi sonucun rasgele yürümesi olarak ifade edilir. Karınca aslanı algoritmasının genel adımları Tablo 3.1'de gösterildiği gibidir.

Tablo 3.1. ALO'nun sözde kodu

prosedür ALO <i>Karıncaların ve karınca aslanlarının ilk popülasyonunu rasgele başlat</i> <i>Karıncaların ve karınca aslanlarının uygunluğunu hesapla</i> <i>En iyi karınca aslanlarını bul ve onu elite olarak ata (belirlenen optimum)</i> while <i>durdurma kriteri sağlanıncaya kadar çalıştır</i> for <i>her karınca için</i> <i>Roulet tekerleği seçimi ile bir karınca aslanı seç</i> <i>Denklem 3.10 ve Denklem 3.11'i kullanarak c ve d'yi güncelle</i> <i>Denklem 3.1 ve Denklem 3.7'i kullanarak rasgele yürüyüş ve normalizasyon</i> <i>yarat</i> <i>Denklem 3.14'ü kullanarak karıncaların pozisyonunu güncelle</i> end <i>Tüm karıncaları uygunluk değerini hesapla</i> <i>Bir karınca aslanını daha uygun olan karınca ile değiştirin</i> <i>elite değerinden daha iyi ise elite değerini güncelle</i> end return elite
--

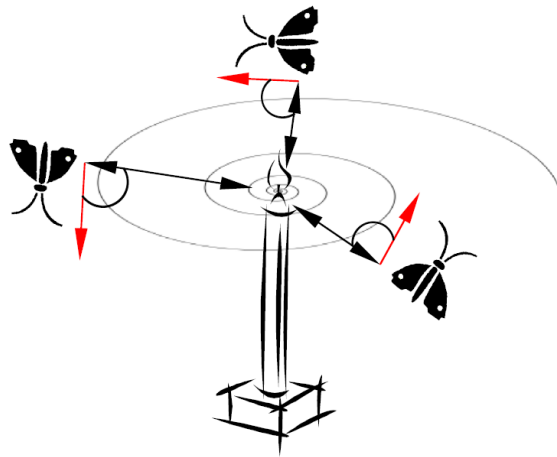
Karınca aslanı optimizasyon algoritmasının temel adımlarını ve bu algoritmanın genel akış yapısı Şekil 3.3'te gösterilmiştir.



Şekil 3.3. ALO'nun akış diyagramı

3.2. Güve Alevi Optimizasyon Algoritması (MFO)

Seyedali Mirjalili tarafından önerilen Güve alevi optimizasyon algoritması, güvelerin geceleri özel navigasyon davranışlarının simülasyonuna dayanan yeni bir meta sezgisel optimizasyon metodudur. Güveler navigasyon için “enine yönlendirme” olarak adlandırılan bir mekanizma kullanılmaktadır. Bu yöntemde, bir güve aya göre sabit bir açı tutarak uçar; Bu da, düz bir yolda uzun mesafeli seyahat etmek için çok etkili bir mekanizmadır. Çünkü ay, güveden çok uzaktadır. Bu mekanizma, güvelerin gece boyunca düz çizgi boyunca uçtuğunu garanti eder. Fakat biz genellikle güvelerin ışıklar etrafında spiral uçtuğunu gözlemliyoruz. Bunun nedeni, güveler yapay ışıklarla kandırılır ve bu tür davranışlar gösterirler. Bu ışıklar aşırı derecede yakın ışık kaynakları olduğu için, Şekil 3.4’te görüldüğü gibi ışık kaynağına olan sabit açının korunması, güvelerin spiral uçuş yolu izlemesine neden olur [40-41].



Şekil 3.4. Güvelerin yakın ışık kaynağı etrafındaki hareketi [39]

MFO algoritmasının matematiksel modelinde, güveler kümesi bir M matriste temsil edilir. Bütün güveler için, uygunluk değerlerini saklamaya yarayan bir OM dizisi vardır.

Algoritmada ikinci önemli bileşen alevlerdir. F matrisi güve matrisine benzer bir matris düşünülebilir. Ayrıca alevler için uygunluk değerlerini depolamak için OF dizisi olduğu varsayılmaktadır.

MFO algoritması, optimizasyon problemlerinin küresel optimalini yaklaşık olarak belirleyen ve Denklem 3.15'teki gibi tanımlanan üçlü bir fonksiyondur:

$$M = (I, P, T) \quad (3.15)$$

I , güvelerin rastgele bir popülasyonu ve buna karşılık gelen uygunluk değerlerini oluşturan işlevdir. Bu fonksiyonun matematiksel modeli Denklem 3.16'daki gibidir:

$$I : \phi \rightarrow \{M, OM\} \quad (3.16)$$

P fonksiyonu temel fonksiyondur. Bu fonksiyon güvelerin arama alanında hareket ettirilmesini sağlar. Bu fonksiyon Denklem 3.17'deki gibi, M matrisini alır ve güncellenmiş olanını sonunda döndürür:

$$P : M \rightarrow M \quad (3.17)$$

T fonksiyonu Denklem 3.18'de gösterildiği gibi sonlandırma ölçütü yerine getirildiğinde *true* değerini, kriterinin karşılanmaması durumunda ise *false* değerini döndürür:

$$T : M \rightarrow \{true, false\} \quad (3.18)$$

I , P ve T fonksiyonları ile MFO algoritmasının genel yapısı Tablo 3.2'deki gibi oluşturulmuştur.

Tablo 3.2. I , P , T fonksiyonları

$M = I();$ while $T(M)$ false'a eşit ise $M = P(M);$ End

Başlatma işleminden sonra, P işlevi *true* döndürene kadar yinelemeli olarak çalıştırılır. Güvelerin davranışını matematiksel olarak taklit etmek uğruna, her güvenin konumu bir alevi dikkate alarak Denklem 3.19'u kullanarak güncellenir:

$$M_i = S(M_i, F_j) \quad (3.19)$$

Denklem 3.19'da M_i i . güveyi, F_j j . Alevi ve S ise spiral fonksiyonu gösterir.

Burada aşağıdaki koşullara tabi kalınırsa her tür spiral kullanılabilir:

- (1) Spiral'in başlangıç noktası, güveden başlamalıdır.

(2) Spiral'in son noktası alevin pozisyonu olmalıdır.

(3) Spiral aralığındaki dalgalanma, arama alanını aşmamalıdır.

Bu koşulları göz önünde bulundurarak, MFO algoritması için Denklem 3.20'deki gibi bir logaritmik spiral tanımlanmaktadır:

$$S(M_i, F_j) = D_i \cdot e^{b \cdot t} \cdot \cos(2\pi t) + F_j. \quad (3.20)$$

Burada D , j . Alev için i . güvenin mesafesini belirtir, b logaritmik spiralın şeklini tanımlamak için sabittir ve t ise $[-1,1]$ arasında rastgele bir sayıdır.

D Denklem 3.21'deki gibi hesaplanır.

$$D_i = |F_j - M_i|. \quad (3.21)$$

Denklem 3.21'de M_i , i . güveyi, F_j , j . Alevi ve D_i ise j . Alev için i . güvenin mesafesini belirtir

Denklem 3.20, güvelerin spiral uçuş yolunu tarif eder. Bu denklemden, bir güvenin bir sonraki konumu bir aleve göre tanımlanır. Spiral denklemdeki t parametresi, güvenin bir sonraki konumunun alevle ne kadar yakın olması gerektiğini tanımlar.

MFO algoritmasının yerel optimuma takılıp kalmasını önemek için Denklem 3.22 kullanılmıştır.

$$flame_no = round\left(N - I * \frac{N - 1}{T}\right), \quad (3.22)$$

Denklem 3.22'de I anlık iterasyon sayısını, N maksimum güve sayısını ve T maksimum iterasyon sayısını gösterir.

Alev sayısının kademeli olarak azalması, arama alanının araştırılması ve sömürülmesiyle dengelenir. MFO algoritmasının genel adımları Tablo 3.3 ile açıklanabilir.

Tablo 3.3. MFO'nun sözde kodu

```
prosedür MFO
Güve popülasyonunu oluştur
Denklem 2.22'i kullanarak flame_no güncelle
OM = Uygunluk_Fonksiyonu(M);
if iterasyon == 1
    F = sort(M);
    OF = sort(OM);
else
    F = sort(Mt-1, Mt);
    OF = sort(Mt-1, Mt);
end
for i = 1: n
    for j = 1: d
        r ve t'yi güncelle
        Denklem 3.21'i kullanarak D'yi hesapla
        Denklem 3.19 ve Denklem 3.20'i kullanarak Mi,j'i güncelle
    end
end
```

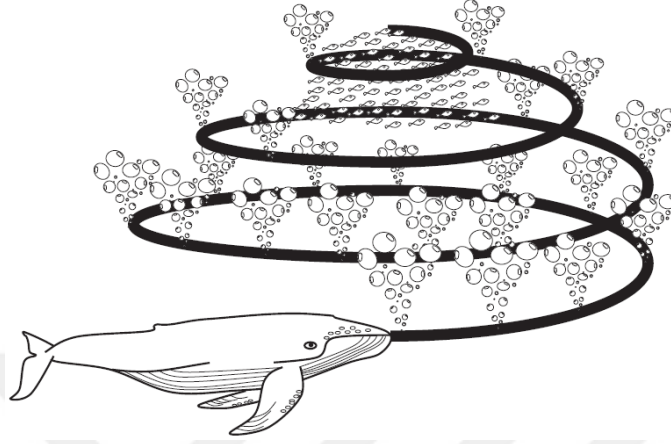
Tablo 3.3'te gösterildiği gibi MFO algoritması sonlandıktan sonra en iyi güve optimum değeri olarak döndürülür.

3.3. Balina Optimizasyon Algoritması (WOA)

Balinalar garip yaratıklardır. Dünyadaki en büyük memeliler olarak kabul edilirler. Yetişkinliği tamamlamış balina 25-30 metre uzunluğuna ve 170-180 ton ağırlığa kadar büyüyebilir. Bu dev memelinin katil, Minke, kuzey, kambur, sağ, oluklu ve mavi olmak üzere 7 farklı ana çeşidi vardır. Balinalar çoğunlukla yırtıcı olarak düşünülür. Okyanus yüzeyinden nefes alma ihtiyaçlarını karşıladıkları için hiç uyuya kalmazlar. Aslında sadece beynin bir tarafı uyur. Balinaların ilginç yanı, duygularıyla son derece akıllı hayvanlar olarak kabul edilmeleridir. Beyinlerindeki hücreler insanlardaki yargı, duyguları ve sosyal davranışlarını belirleyen hücrelerdir. Bir başka deyişle “spindle” hücreleri bizi diğer canlılardan farklı kılar. Balinaların akıllı olmalarının temel sebebi yetişkin insandan 2 kat daha fazla hücreye sahip olmalarıdır. Balinalar insanlar gibi düşünebilir, öğrenebilir, yargılayabilir, iletişim kurabilir ve hatta duygusal olabilirler. Ancak tabi ki insanlardan çok daha düşük zekâ seviyeleri vardır. Balinaların (çoğunlukla katil balinalar) çenelerini daha iyi geliştirebildikleri gözlemlenmiştir [42].

Bir diğer ilginç nokta da balinaların toplumsal davranışlarıdır. Yalnız veya grup halinde yaşarlar. Fakat çoğunlukla gruplar halinde gözlenmektedir. Türlerinden bazıları (örneğin katil

balinalar), tüm yaşam süresi boyunca aile ile de yaşayabilir. Balina türlerinin içindeki en büyük balina türü kambur balinalardır (*Megaptera Novaeangliae*). Yetişkin bir kambur balina neredeyse bir okul otobüsü boyutundadır. En sevdikleri av, kril ve küçük balık sürüleridir. Şekil 3.5'te bu hayvan ve avı gösterilmiştir.



Şekil 3.5. Kambur balinaların balon ağı ile besleme davranışı [42]

Kambur balinalarla ilgili en ilginç şey özel avlanma yöntemidir. Bu beslenme davranışına kabarcık ağ besleme yöntemi denir [43]. Kambur balinalar yüzeye yakın krilleri veya küçük balıkları avlamayı tercih eder. Bu yemleme işlemini Şekil 3.5'te gösterildiği gibi çemberlerden oluşan kendine özgü kabarcıklar oluşturarak yaptığı gözlemlenmiştir. 2011'den önce bu davranış sadece yüzey üzerinden gözlemlere dayanılarak incelenmiştir. Bununla birlikte, Goldbogen ve arkadaşları, bu davranışı “tag sensörleri” kullanarak araştırmışlardır [44]. Onlar kabarcık ile ilişkili iki manevra buldular ve onları “upward-spirals” (yukarı doğru spiral) ve “double-loops” (çift döngüler) olarak adlandırdılar. İlk manevrada kambur balinalar 12 metre aşağıya dalarak yem etrafında spiral şekilde kabarcıklar oluşturmaya başlar ve yüzeye doğru yüzer. Daha sonraki manevra üç farklı aşamayı içerir: Mercan döngüsü, lob tırnağı ve yakalama halkası. Bu davranışlarla ilgili ayrıntılı bilgi [44]'de bulunabilir.

Kabarcık ağ beslemesi, yalnızca kambur balinalarda gözlemlenebilen benzersiz bir davranış olduğunu belirtmek gerekir. Spiral kabarcık ağı ile besleme manevrası, optimizasyonu gerçekleştirmek için matematiksel olarak modellenmiştir [45].

3.3.1. Av Kuşatma Denklemi

Kambur balinaları avın yerini tanıyabilir ve onları kuşatabilir. Kambur balina, avı kuşatır (küçük balıklar) ve daha sonra başlangıçtan maksimum yineleme sayısına kadar yineleme

sayısının artması boyunca optimum çözüme doğru konumunu günceller. Bu işlem Denklem 3.23 ve Denklem 3.24'te belirtilmiştir.

$$\vec{D} = |C.\vec{X}^*(t) - X(t)| \quad (3.23)$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A}.\vec{D} \quad (3.24)$$

Burada $\vec{A}$ ve $\vec{D}$ iş birlikçi vektörlerdir. t anlık iterasyonu, $\vec{X}^*(t)$ geçen iterasyon boyunca optimum çözüm vektörünü, $X(t)$ ise pozisyon vektörünü ifade eder.

$\vec{A}$ ve $\vec{D}$ vektörleri sırasıyla Denklem 3.25 ve Denklem 3.26'daki gibi hesaplanır:

$$\vec{A} = 2\vec{a} * r - \vec{a} \quad (3.25)$$

$$\vec{C} = 2 * r \quad (3.26)$$

Burada $\vec{a}$ yineleme boyunca doğrusal olarak 2'den 0'a düşen bir değişkendir ve r [0,1] arasında rastgele bir sayıdır.

3.3.2. Kabarcık Ağı Saldırı Yöntemi

Kambur balinalarının kabarcık ağı davranışına ilişkin matematiksel denklem için, iki yöntem şu şekilde modellenmiştir:

(1) Küçülen Kuşatma Mekanizması:

Bu teknik, $\vec{a}$ değerinin 2'den 0'a doğru lineer olarak azaltılması ile uygulanır.

(2) Spiral Konum Güncelleme:

Kambur balina ve av arasındaki konum güncelleme için aşağıdaki gibi sarmal şeklinde hareket olan matematiksel spiral denklemi Denklem 3.27'deki gibidir:

$$\vec{X}(t+1) = \vec{D}' * e^{bt} * \cos(2\pi l) + \vec{X}^*(t) \quad (3.27)$$

Burada: l , [-1, 1] arasında rastgele bir sayıdır. b sabiti logaritmik şekli tanımlar,

$\vec{D} = |\vec{X}^*(t) - X(t)|$ ise i. balina ile kurban arasındaki mesafeyi başka bir deyişle şimdiye kadar en iyi çözümü ifade eder.

Kambur balinalarının aynı anda küçülen bir daire içerisinde ve aynı anda spiral biçimli bir yol boyunca av çevresinde yüzdüğü bilinmektedir. Bu eşzamanlı davranışı modellemek için, optimizasyon sırasında balinaların konumunu güncellemek için küçülen kuşatma mekanizması veya spiral konum güncelleme modeli arasında seçim yapmak için %50 olasılığının olduğunu varsayılmaktadır. Bu durum Denklem 3.28'de belirtilmiştir.

$$\vec{X}(t+1) = \begin{cases} \vec{X} * (t) - \vec{A} \cdot \vec{D} & \text{if } p < 0.5 \\ \vec{D}' * e^{bt} * \cos(2\pi l) + \vec{X} * (t) & \text{if } p \geq 0.5 \end{cases} \quad (3.28)$$

Burada p , $[0,1]$ arasında rastgele bir sayıdır.

(3) Av Arama:

$\vec{A}$ vektörü, av için arama yapmakta kullanılabilir. $\vec{A}$ vektörü ayrıca, 1 değerinden büyük veya -1 değerinden daha küçük değerleri alır. Arama iki şartla yapılmaktadır.

Bu şartlar Denklem 3.29 ve Denklem 3.30'da belirtildiği gibidir.

$$\vec{D} = |C \cdot \overrightarrow{X_{rand}} - \vec{X}| \quad (3.29)$$

$$\vec{X}(t+1) = \overrightarrow{X_{rand}} - \vec{A} \cdot \vec{D} \quad (3.30)$$

Son olarak şu koşulları takip eder:

$|\vec{A}| > 1$ global optimumu bulmak için WOA algoritmasına keşif uygulayarak yerel optimumu önler.

$|\vec{A}| < 1$ geçerli arama ajanının konumunu güncellemek için en iyi çözüm seçilmiştir.

Balina optimizasyon algoritmasının matematiksel modellemesinin ardından algoritmaya ait genel yapı Tablo 3.4'te sunulmuştur.

Tablo 3.4. WOA'nın sözde kodu

prosedür WOA

Balinalar popülasyonunu başlat $X_i (i = 1, 2, \dots, n)$

Tüm arama ajanlarının uygunluğunu hesapla

X^ = en iyi arama ajanı olarak ata*

while $t < \text{iterasyon sayısı}$

for *herbir ajan için*

a, A, C, I ve p 'yi güncelle

if1 ($p < 0.5$)

if2 ($|A| < 1$)

Denklem 3.23'u kullanarak şu andaki arama ajanının pozisyonunu güncelle

else if2 ($|A| \geq 1$)

rasgele bir a ajanını seç (X_{rand})

Denklem 3.30'u kullanarak şu andaki arama ajanının pozisyonunu güncelle

end

else if1 ($p \geq 0.5$)

Denklem 3.30'u kullanarak şu andaki arama ajanının pozisyonunu güncelle

end

end

Herhangi bir arama ajanının arama alanının ötesine geçip gelmediğini kontrol edin ve değiştir

Her bir arama ajanının uygunluğunu hesapla

Daha iyi bir çözüm varsa X^ 'i güncelle*

$t = t + 1$;

end

return X^*

4. DENEYLER VE SONUÇLARI

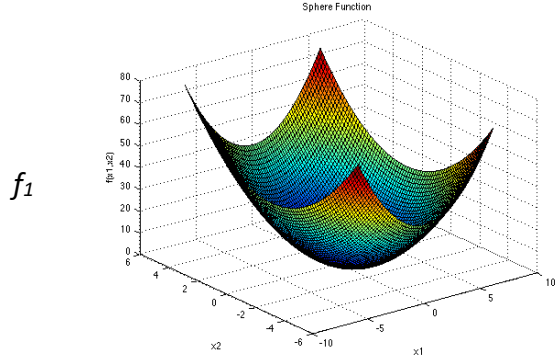
4.1. Kalite Testi Fonksiyonları

Bu çalışmada biyolojik yapıları ve matematiksel modelleri incelenen WOA, SSA, ALO ve MFO algoritmaları kalite testi fonksiyonları ile eşit şartlar altında karşılaştırılmıştır. Algoritmaların başarısını ölçmek için birçok kalite testi fonksiyonu bulunmaktadır. Seçilen kalite testi fonksiyonları ve özellikleri Tablo 4.1'deki gibidir. Şekil 4.1'de kullanılan kalite testi fonksiyonların grafikleri verilmiştir [46-47-48-49].

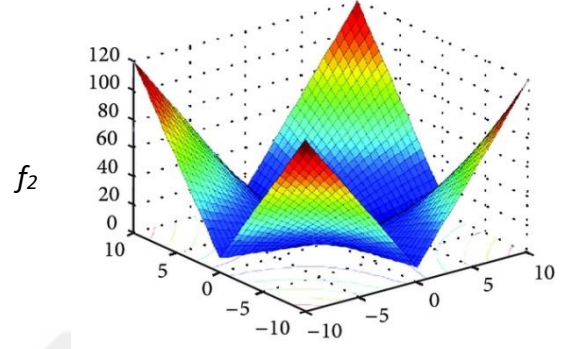
Tablo 4.1. Kalite testi fonksiyonları

Fonksiyonlar	Alt ve Üst Sınırlar	Min	D	Tip	Denklem
f_1 -Sphere	[-5.12, 5.12]	0	30	UM	$f_1 = \sum_{i=1}^n x_i^2$
f_2 -Schwefel2.22	[-10, 10]	0	30	UM	$f_2 = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $
f_3 -Rosenbrock	[-30, 30]	0	30	UM	$f_3 = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i) + (x_i - 1)^2]$
f_4 -Powell	[-4, 5]	0	24	UM	$f_4 = \sum_{i=1}^{n/4} \left[(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 \right] + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4$
f_5 -Trid6	$[-D^2, D^2]$	-50	6	UM	$f_{10} = \sum_{i=1}^n (x_i - 1)^2 - \sum_{i=2}^n x_i x_{i-1}$
f_6 -Beale	[-4.5, 4.5]	0	5	UM	$f_{12} = (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_1^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2$
f_7 -Schwefel2.26	[-500, 500]	-12569.5	30	MM	$f_{14} = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$
f_8 -Rastrigin	[-5.12, 5.12]	0	30	MM	$f_{15} = \sum_{i=1}^n [x_i - 10 \cos(2\pi x_i) + 10]$
f_9 -Bohachevsky1	[-100, 100]	0	30	MM	$f_{16} = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7$
f_{10} -Ackley	[-32, 32]	0	30	MM	$f_{17} = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + e$
f_{11} -Griewank	[-600, 600]	0	30	MM	$f_{18} = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$
f_{12} -Penalized1	[-50, 50]	0	30	MM	$f_{19} = \frac{\pi}{n} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{1}{4}(x_i + 1)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$

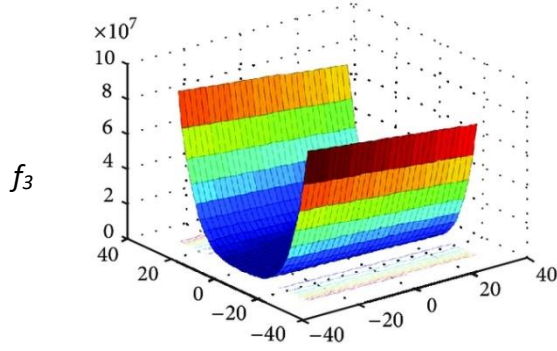
Tablo 4.1’de belirtilen her bir fonksiyon için WOA, SSA ALO ve MFO algoritmaları 30 kez çalıştırılmıştır. Bu çalıştırmanın sonucunda dört algoritmanın en iyi, en kötü, ortalama, ortanca ve standart sapma gibi değerleri Tablo 4.2 ile sunulmuştur.



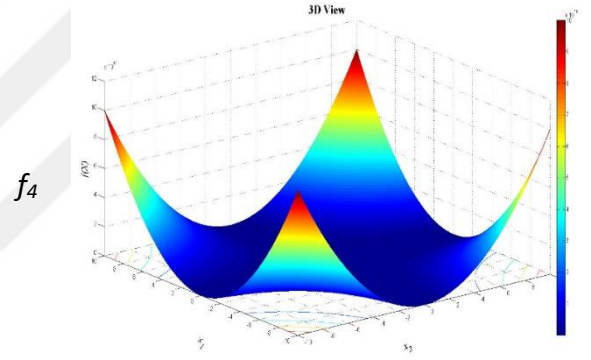
(a)



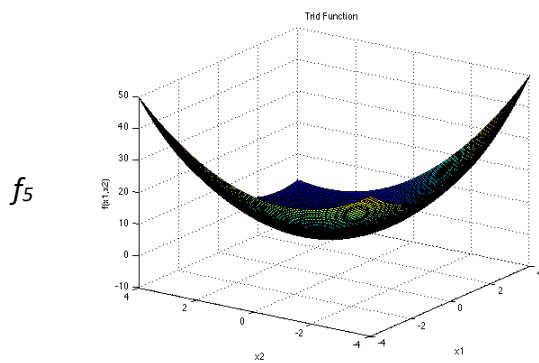
(b)



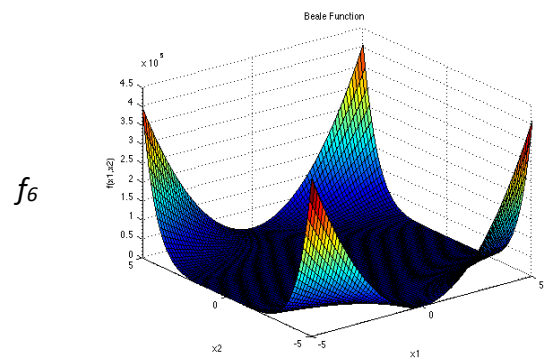
(c)



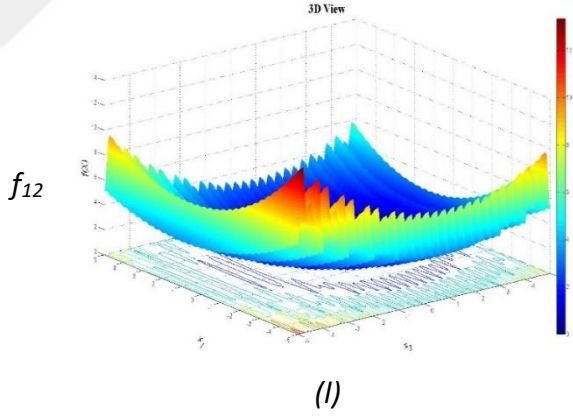
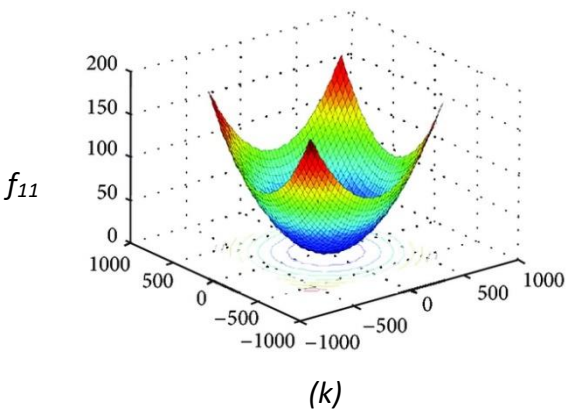
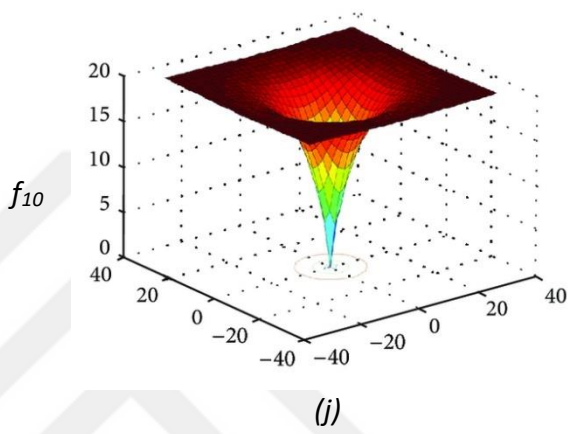
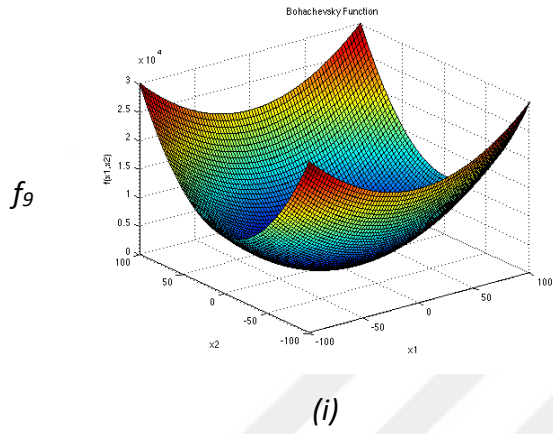
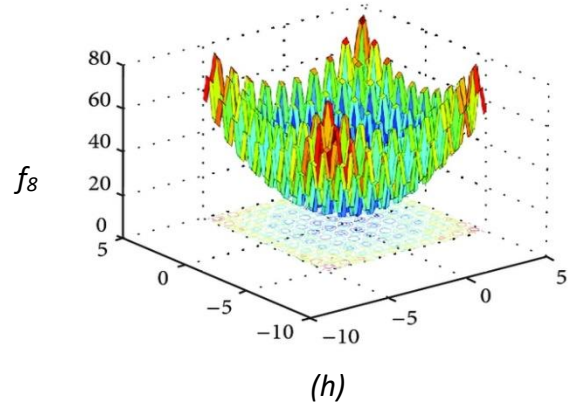
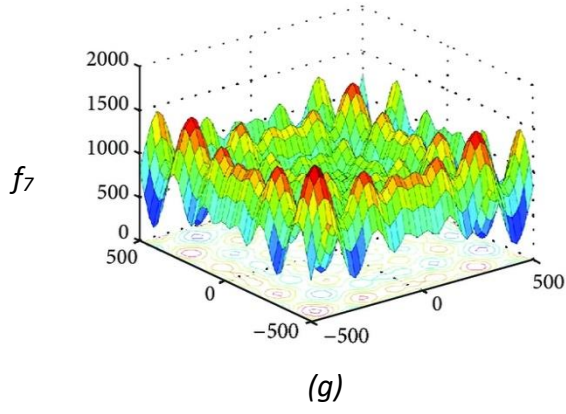
(d)



(e)



(f)

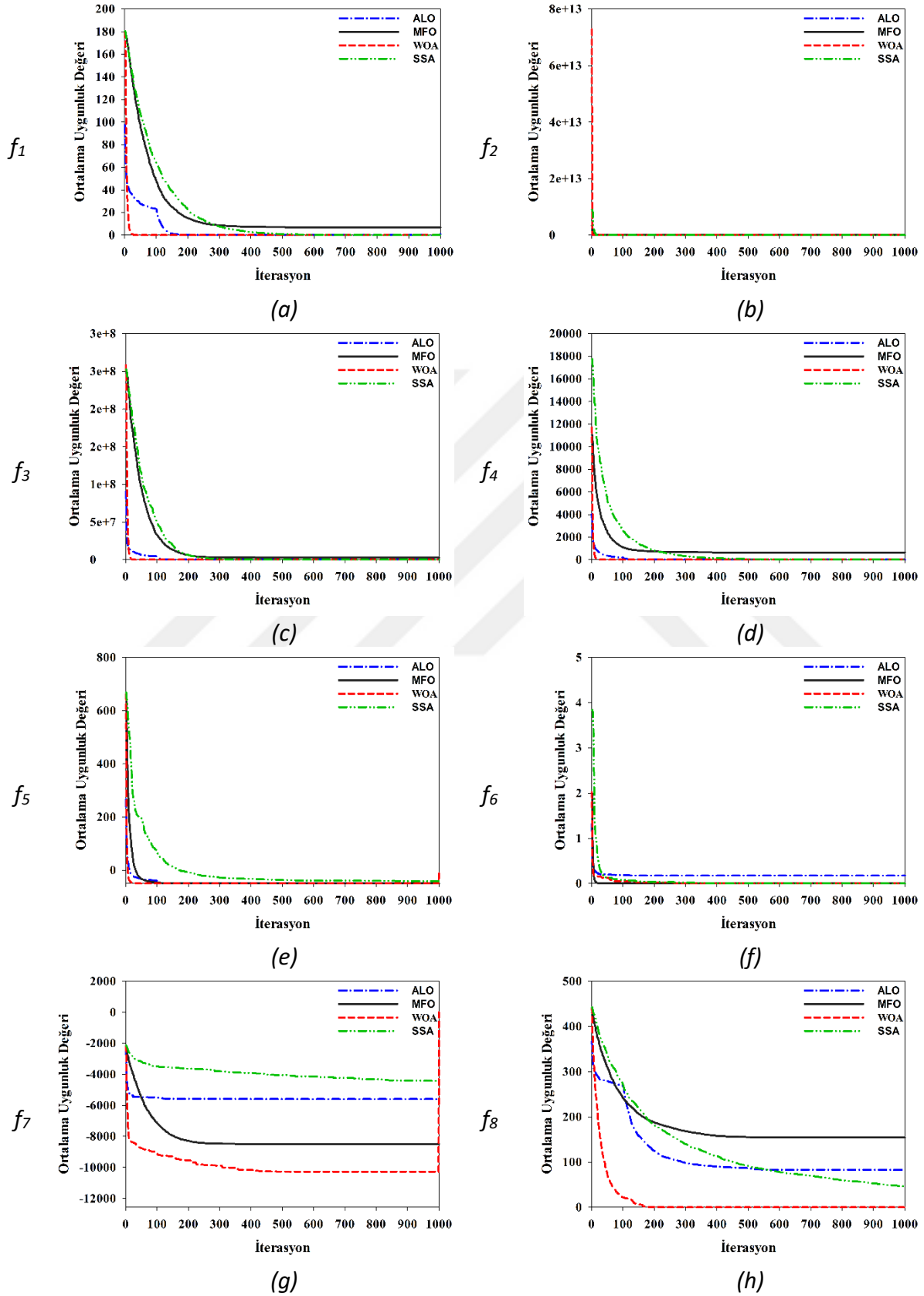


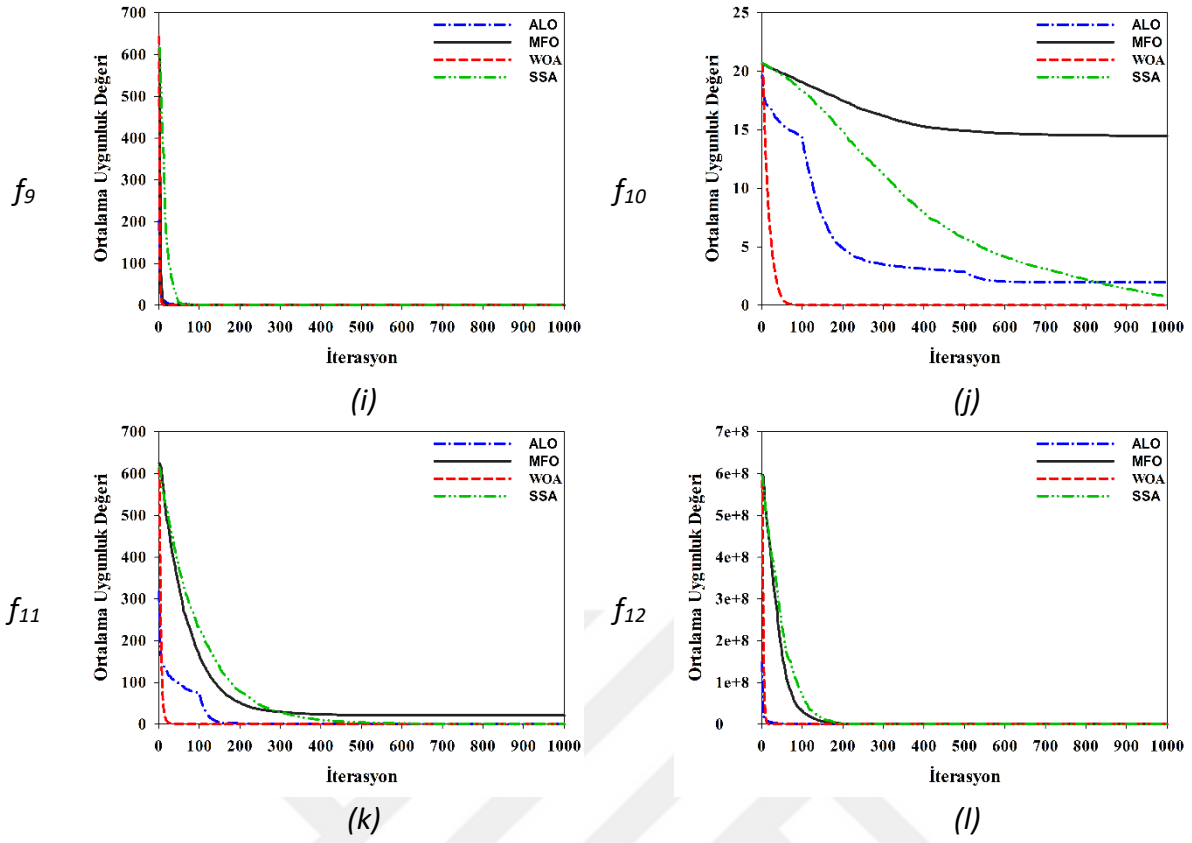
Şekil 4.1. Kalite testi fonksiyonları

Tablo 4.2. ALO, MFO, WOA, SSA'nın karşılaştırılması

onksiyonlar	Algoritmalar	En İyi Değer	En Kötü Değer	Ortalama (Mean)	Ortanca (Median)	Standart Sapma
f_1	ALO	2.13E-009	1.07E-007	3.20E-008	1.98E-008	2.73E-008
	MFO	2.66E-008	5.24E+001	6.99E+000	5.54E-007	1.52E+001
	WOA	7.45E-173	8.04E-152	2.77E-153	1.28E-161	1.46E-152
	SSA	1.41E-003	1.06E-002	5.51E-003	5.26E-003	2.08E-003
f_2	ALO	2.49E-001	1.32E+002	3.64E+001	7.99E+000	4.87E+001
	MFO	2.93E-004	9.00E+001	3.20E+001	3.00E+001	2.20E+001
	WOA	3.26E-116	1.80E-102	7.28E-104	1.33E-108	3.30E-103
	SSA	2.34E-001	1.79E+000	4.95E-001	3.99E-001	2.88E-001
f_3	ALO	1.92E+001	1.47E+003	2.86E+002	9.34E+001	4.19E+002
	MFO	1.02E+000	8.00E+007	2.68E+006	2.02E+002	1.46E+007
	WOA	2.65E+001	2.87E+001	2.71E+001	2.69E+001	4.86E-001
	SSA	2.95E+002	9.63E+002	5.09E+002	4.75E+002	1.49E+002
f_4	ALO	7.60E-002	2.17E+000	7.43E-001	6.26E-001	4.78E-001
	MFO	8.75E-003	3.74E+003	6.40E+002	1.47E+002	1.01E+003
	WOA	1.02E-015	1.01E-005	1.46E-006	8.55E-008	2.43E-006
	SSA	1.43E+000	8.47E+000	4.03E+000	3.81E+000	1.85E+000
f_5	ALO	-5.00E+001	-5.00E+001	-5.00E+001	-5.00E+001	1.24E-011
	MFO	-5.00E+001	-5.00E+001	-5.00E+001	-5.00E+001	9.36E-014
	WOA	-5.00E+001	-4.99E+001	-4.99E+001	-4.99E+001	7.47E-005
	SSA	-4.77E+001	-3.14E+001	-4.10E+001	-4.17E+001	4.56E+000
f_6	ALO	3.28E-017	7.62E-001	1.77E-001	7.05E-015	3.27E-001
	MFO	0.00E+000	1.03E-018	3.97E-020	0.00E+000	1.90E-019
	WOA	8.57E-016	2.33E-010	2.39E-011	1.97E-012	5.55E-011
	SSA	4.29E-008	1.26E-003	1.36E-004	3.30E-005	2.54E-004
f_7	ALO	-8.83E+003	-5.41E+003	-5.58E+003	-5.48E+003	6.16E+002
	MFO	-1.01E+004	-6.86E+003	-8.48E+003	-8.42E+003	7.96E+002
	WOA	-1.25E+004	-8.43E+003	-1.02E+004	-9.15E+003	1.61E+003
	SSA	-7.80E+003	-3.31E+003	-4.48E+003	-4.24E+003	1.06E+003
f_8	ALO	2.68E+001	1.38E+002	8.27E+001	7.86E+001	2.82E+001
	MFO	7.86E+001	2.46E+002	1.54E+002	1.55E+002	4.29E+001
	WOA	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
	SSA	3.52E+001	5.78E+001	4.66E+001	4.89E+001	6.30E+000
f_9	ALO	7.43E-014	3.37E-010	4.70E-011	3.06E-011	6.36E-011
	MFO	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
	WOA	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
	SSA	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
f_{10}	ALO	6.67E-004	4.46E+000	1.97E+000	2.01E+000	8.76E-001
	MFO	1.09E-002	1.99E+001	1.44E+001	1.90E+001	7.86E+000
	WOA	8.88E-016	7.99E-015	5.15E-015	4.44E-015	2.86E-015
	SSA	3.11E-001	1.15E+000	7.43E-001	7.17E-001	2.03E-001
f_{11}	ALO	2.27E-004	4.71E-002	1.33E-002	8.42E-003	1.36E-002
	MFO	1.01E-004	1.80E+002	2.11E+001	2.34E-002	4.56E+001
	WOA	0.00E+000	1.06E-001	3.54E-003	0.00E+000	1.94E-002
	SSA	7.41E-001	1.03E+000	9.73E-001	1.00E+000	7.27E-002
f_{12}	ALO	4.02E+000	1.18E+001	7.92E+000	7.69E+000	1.62E+000
	MFO	8.77E-003	3.74E+000	8.03E-001	4.70E-001	8.66E-001
	WOA	3.04E-003	1.43E-001	2.48E-002	1.44E-002	2.97E-002
	SSA	1.37E-001	3.40E-001	2.05E-001	1.98E-001	4.89E-002

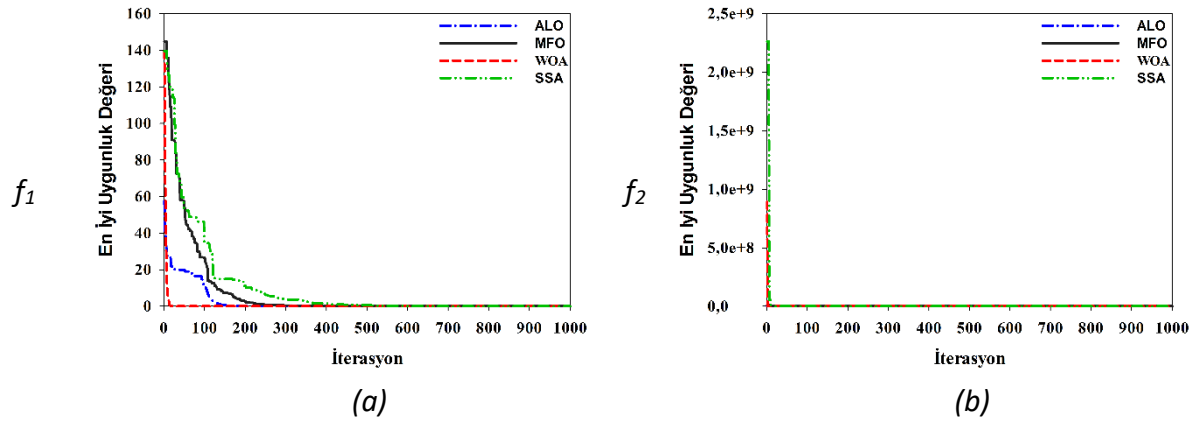
Her bir fonksiyon için 30 kez çalıştırılan algoritmaların ortalama uygunluk değeri için yakınsama grafikleri Şekil 4.1’de gösterilmiştir.

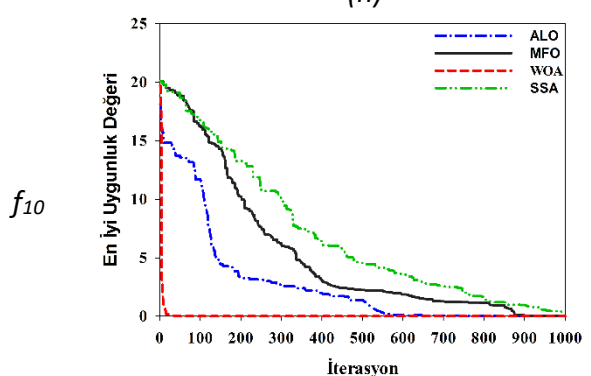
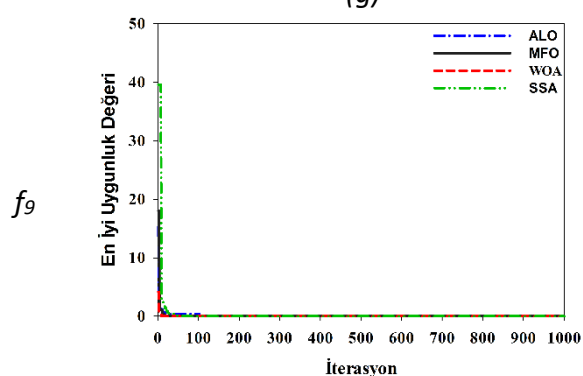
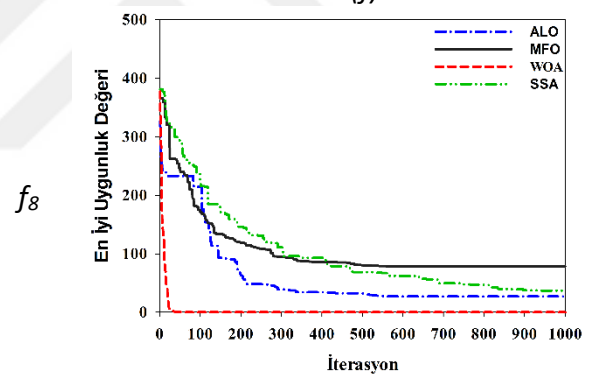
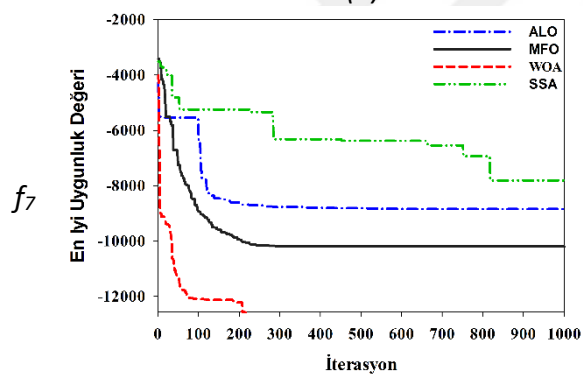
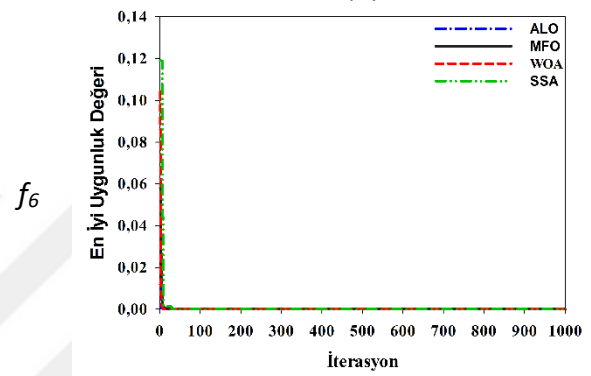
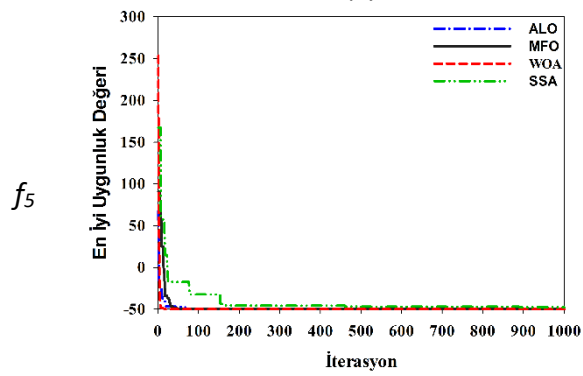
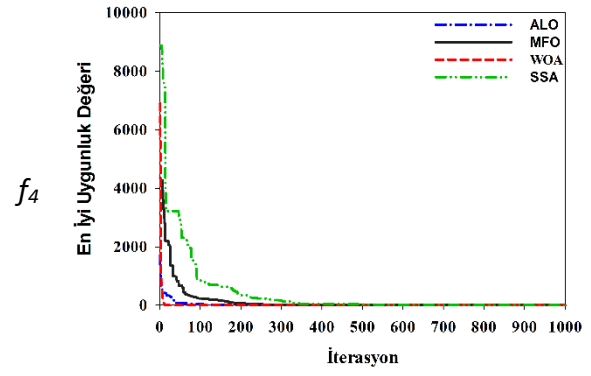
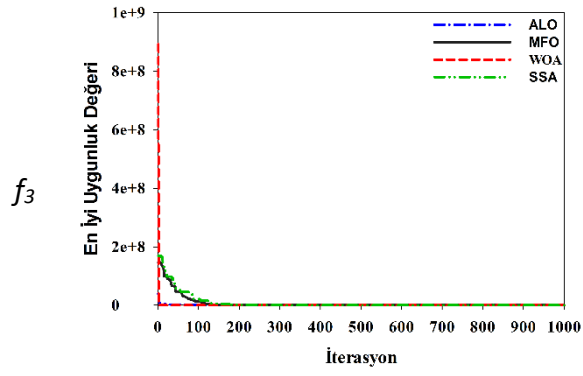


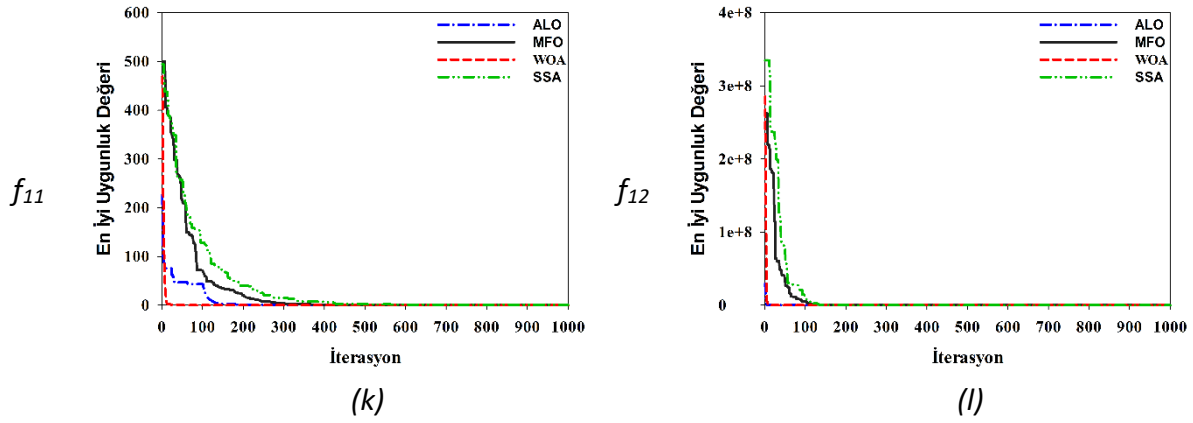


Şekil 4.2. ALO, MFO, WOA ve SSA'nun ortalama uygunluk değeri için yakınsama grafikleri

Her bir fonksiyon için 30 kez çalıştırılan algoritmaların en iyi uygunluk değeri için yakınsama grafikleri Şekil 4.2'de gösterilmiştir.







Şekil 4.3. ALO, MFO, WOA ve SSA'nun en iyi uygunluk değerleri için yakınsama grafikleri

4.2. Değerlendirme ve Sonuç

Bu kısımda Sosyal Örümcek Optimizasyonu (SSA) olarak adlandırılan yeni bir sürü zekası optimizasyon algoritması incelenmiştir. SSA, örümceklerin işbirliği içerisindeki davranışlarının simülasyonuna dayanmaktadır. Dişi ve erkek örümceklerin bulunduğu sosyal ağda birbirleriyle olan iletişimi ve çiftleşme mekanizması üzerinde durulmuştur. Cinsiyete bağlı olarak, bir koloninin içindeki her bireyin farklı iş birliğine dayalı simülasyonu bir dizi evrimsel operatörler tarafından yapılmaktadır. Önerilen yaklaşımın mevcut sürü zekâsına dayalı algoritmalarından en temel farkı her bir bireyin cinsiyeti dikkate alınarak modellenmiş olmasıdır.

WOA, SSA, ALO ve MFO algoritmalarının biyolojik yapısı detaylı olarak incelenmiştir. Bu algoritmalar için çıkarılan matematiksel modellerin temsil biçimleri göz önünde bulundurularak algoritmaların genel işleyişi sözde kod kullanılarak verilmiştir.

Bu çalışmada on iki kalite testi fonksiyonu kullanılarak SSA'nın performansı, stabilitesi ve duyarlılığı farklı optimizasyon algoritmaları (ALO, MFO, WOA) ile karşılaştırılmıştır. Elde edilen sonuçlar ışığında SSA'nın kovaryansının ALO ve WOA'sına göre düşük olduğu ancak, MFO algoritmasıyla başa baş olduğunu söyleyebiliriz. Ayrıca, SSA'nın yakınsama hızı MFO algoritmasından çok daha başarılıdır. WOA'nın kovaryansının ve yakınsama hızının daha iyi olduğu çalışma sonuçlarıyla sabittir. Bu çalışmadan çıkarılacak en önemli sonuç tüm algoritmalar için parametre değerleri iyi ayarlanırsa daha dikkat çekici sonuçlar elde edilebilir.

5. SOSYAL AĞLARDA DUYGU ANALİZİ

Başkalarının ne düşündüğü çoğumuz için her zaman karar verme sürecinde önemli bir bilgi parçası olmuştur. İnternetin farkındalığı yaygınlaşmadan çok önce, birçoğumuz arkadaşlarımıza bir oto tamircisi tavsiye etmelerini, yerel seçimlerde kime oy vermeyi planladıklarını, meslektaşlarımızın iş başvuru için iş sahipleri ile ilgili referans mektuplarının talebini ya da hangi bulaşık makinesini almayı önerdiklerini sorup bilgi alışverişi yapırdı. Ancak bugün internet teknolojilerinin gelişimi hem kişisel olarak tanımadıklarımızdan hem de tanınmış profesyonel eleştirmenlerden, sahip oldukları görüş ve deneyimlerini keşfetme fırsatını sağlamış oldu. Öyle ki yapılan çalışmalar her geçen gün daha fazla insanın, yabancılar için görüşlerini internet üzerinden sunmaya başladığını göstermektedir [50].

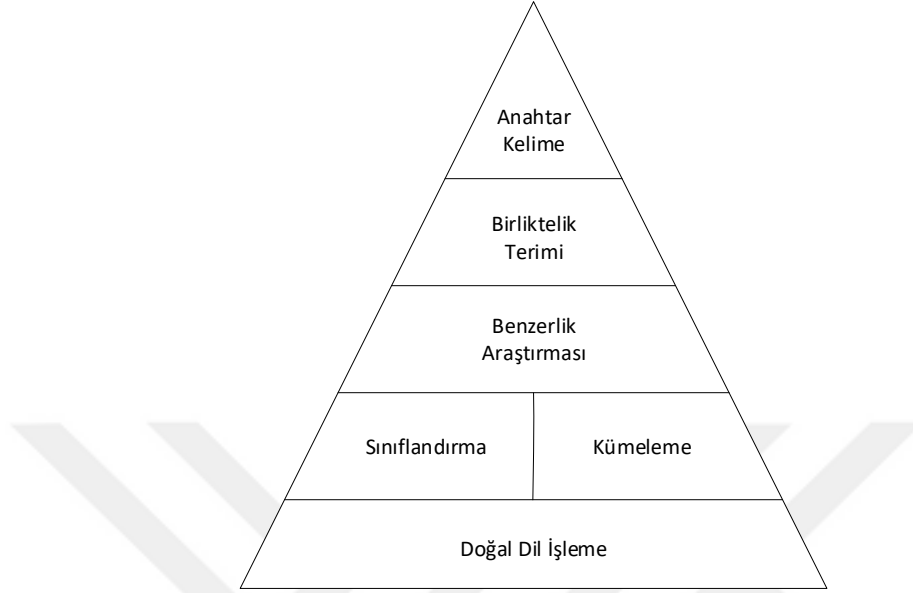
Görüşler neredeyse bütün insan faaliyetlerinin merkezinde yer alır ve bizim davranışlarımızın önemli bir yansımasını oluşturur. Gerçekte olan inancımız, algılamalarımız ve yaptığımız seçimler, başkalarının dünyayı nasıl gördüğüne ve değerlendiklerine büyük ölçüde bağlıdır. Bu nedenle, karar vermemiz gerektiğinde başkalarının görüşlerini sıklıkla başvururuz. Bu sadece bireyler için geçerli değildir. Organizasyonlar için de geçerlidir.

Görüşler, ölçümler, değerlendirmeler, tutumlar ve yorumlar gibi düşünceler ve bunlarla ilgili kavramlar, duygu analizi ve fikir madenciliğinin çalışma alanlarını oluşturur. Çalışma alanlarının hızla büyümesi, web'deki forumların, tartışma ve tanışma sayfalarının, blogların, mikroblogların ve diğer sosyal medya araçlarının insanlar arasında kullanımının artmasını sağlamıştır. İnternet kullanımının artması ile birlikte iletişim altyapısı köklü bir değişikliğe uğramıştır. Bilişim teknolojilerinin gelişmesiyle ortaya çıkan sosyal paylaşım siteleri, insan hayatında önemli bir yere sahip olmuştur. Dünya üzerinde kullanılan en popüler sosyal paylaşım siteleri arasında; Facebook, Twitter, Instagram, Youtube, Google Play, Vine gibi web siteleri ve uygulamaları, blog, mikro blog, sosyal ağ ve sosyal imleme servisleri bulunmaktadır. Tüm bu servisler bir araya gelerek “Sosyal Medya” yapısını ortaya çıkarır [51].

5.1. Metin Sınıflandırma

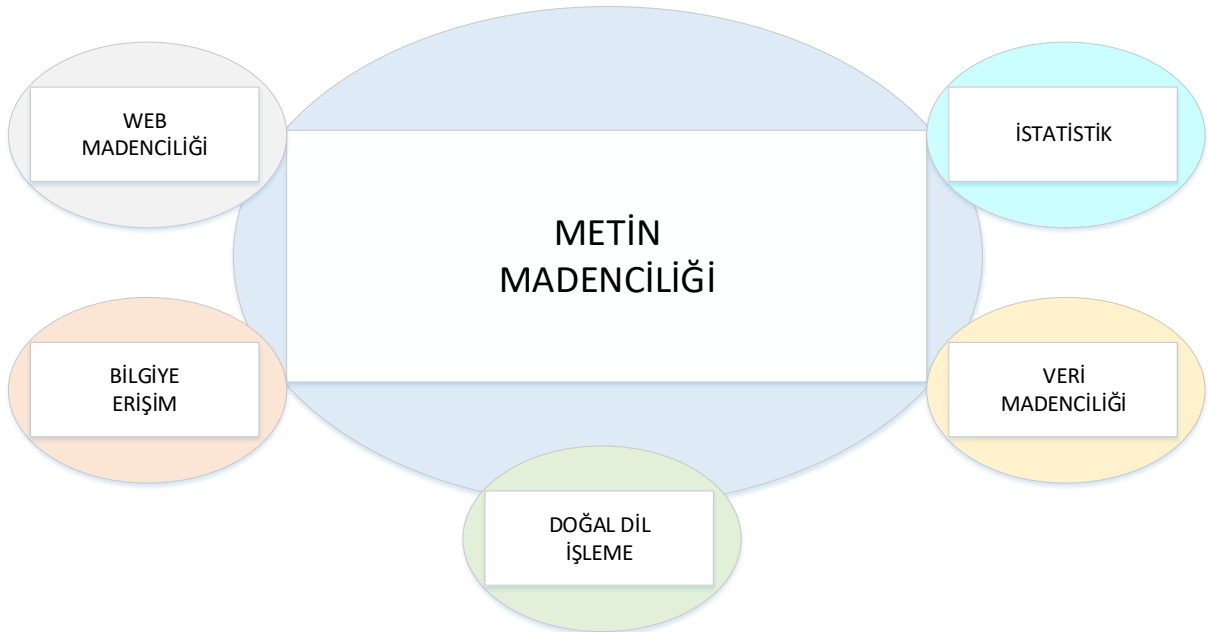
Metin veya belge sınıflandırması, metin ve belgeleri bir veya daha fazla önceden tanımlanmış sınıf veya kategoriye atama işlemidir. Duygu analizi çalışmasında verilen bir metinden anlamlı bilgiler edinme yolculuğuna atılan ilk adım metni anlayabilmektir. Dilin nasıl yapılandırıldığını ve belirli metin sözdizimi kalıplarını anlamak önemlidir. Ancak tek

başına bu geniş miktardaki metin verilerinden maksimum faydalanmak isteyen işletmeler ve kuruluşlar için kullanılması yeterli değildir. Metin madenciliği fonksiyonlarını Şekil 5.1 ile özetleyebiliriz.



Şekil 5.1. Metin madenciliğinin fonksiyonları [52]

Dil işleme konusundaki bilgi, analitik ve makine öğrenmesi (ML) kavramlarıyla birleşince, metin verilerini kullanabilen ve işletmelerde fayda sağlayan gerçek dünyadaki pratik problemleri çözmeye yardımcı olan sistemlerin oluşturulmasına olanak sağlanmış olur. Şekil 5.2’de Metin Madenciliği oluşturan temel adımlar listelenmiştir.



Şekil 5.2. Metin Madenciliği oluşturan temel adımlar [52]

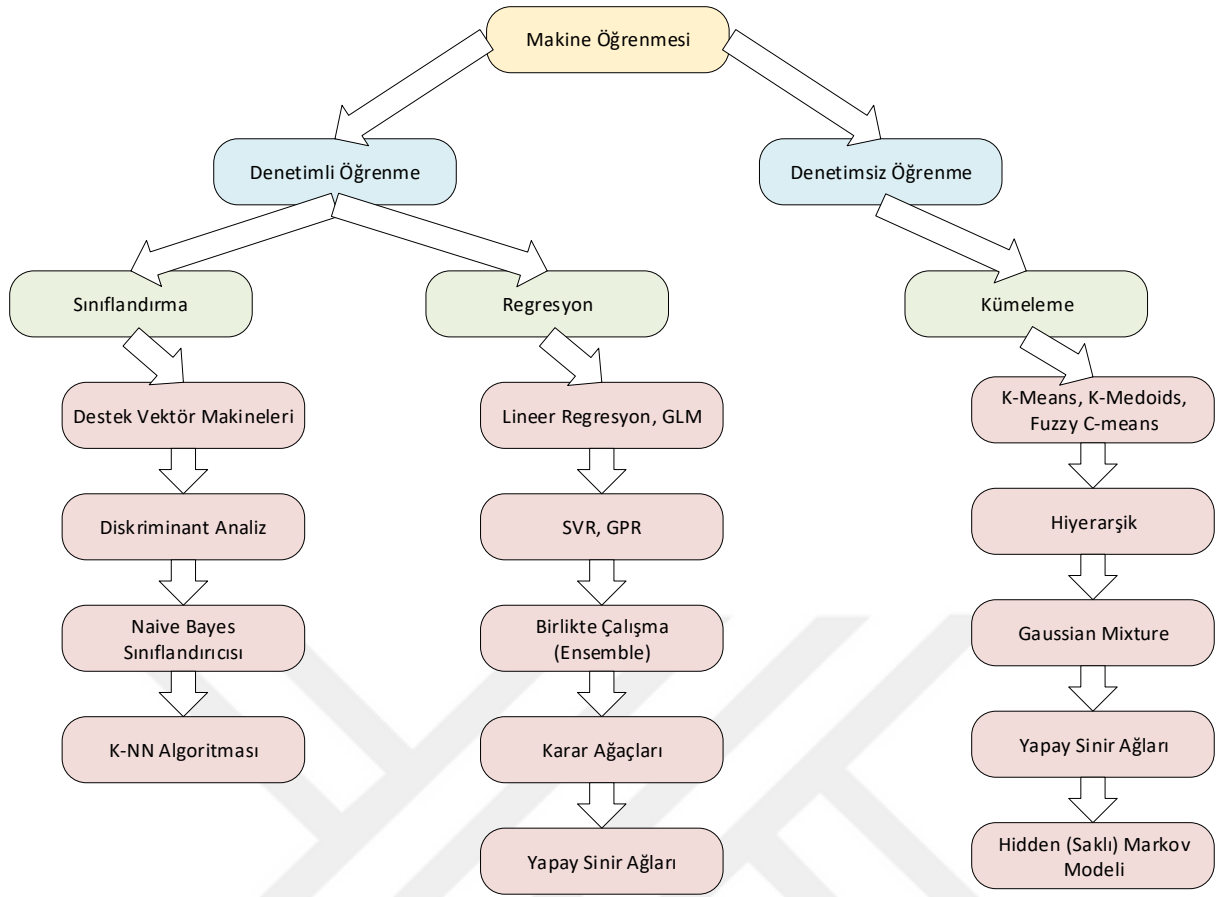
ML'nin denetimli öğrenme, denetimsiz öğrenme, pekiştirmeli öğrenme ve hatta yakın zamanda derin öğrenmeyi içine alan çok yönlü yapısı bulunmaktadır. Bu kavramların her biri, metin verisi üzerinde fikir kurabilen ve çok fazla elle denetime ihtiyaç duymayan kendi kendine öğrenme sistemleri kurmak için kullanılan birçok teknik ve algoritmayı içerir. ML'nin en önemli özelliği, bir model eğitildikten sonra, yararlı bilgiler ve istenen sonuçları görmeye başlamak için bunu yeni ve önceden görülmemiş veriler üzerinde doğrudan kullanabiliriz.

5.2. Metin Sınıflandırma Teknikleri

Metin sınıflandırmasını tanımlamadan önce, metin verisinin kapsamını ve sınıflamayla ne demek istediğimizi anlamamız gerekir. Burada yer alan metinsel veriler, bloglardan, Web'den yada herhangi bir yerden edinilebilen cümlecik, cümle veya metin paragrafları içeren herhangi bir belge olabilir. Sözcük belgesi, yazı, kaydedilmiş konuşma, çizimler, sunum biçiminde olabilecek düşüncelerin ya da olayların somut temsilinin bir biçimi olarak tanımlanabilir. Metin sınıflandırması belirtildiği gibi birçok yolla yapılabilir [53]. Metin sınıflandırmasını otomatikleştirmek için birkaç ML tekniği ve kavramından faydalanabiliriz. Duygu analizi probleminin çözümünde esas olarak iki tür ML tekniği vardır. Bunlar;

- Denetimli makine öğrenimi
- Denetimsiz makine öğrenimi

Ancak bu çalışmada sınıflandırma denetimli öğrenme yaklaşımına odaklanarak gerçekleştirilecektir. Sınıflandırma algoritmaları kullanımı yalnızca metinlerle sınırlı değildir. Bilim, sağlık, hava tahmini, görüntü işleme ve teknoloji de dahil olmak üzere diğer mühendislik alanlarında da oldukça sık kullanılmaktadır. Şekil 5.3'te makine öğrenmesi algoritmalarının sınıflandırılması gösterilmiştir.



Şekil 5.3. Makine Öğrenmesi Teknikleri

5.2.1. Denetimli Makine Öğrenmesi

Denetimli öğrenme, eğitim verileri olarak bilinen ön etiketli veri örnekleri üzerinde eğitilen belirli ML teknikleri veya algoritmaları ifade eder. Özellikler veya nitelikler, özellik çıkarımı kullanarak veriden çıkarılır ve her veri noktası için kendi kümesine ve karşılık gelen sınıf etiketine ait özelliklere sahip olacaktır. Algoritma, eğitim verilerinden her bir sınıf türü için çeşitli kalıpları öğrenir. Bu süreç tamamlandıktan sonra, eğitilmiş bir model oluşmaktadır. Özelliklerini modele aktardıktan sonra bu model, gelecekteki test veri örnekleri için sınıf tahmininde kullanılabilir. Böylece makine aslında, önceki eğitim verileri örneklerine dayanarak yeni sınıfın öngörülmemeyen veri örnekleri için nasıl tahmin yürüteceğini öğrenmiş olacaktır. Denetimli öğrenme algoritmalarının iki ana türü vardır. Bunlar;

Sınıflandırma: Tahmini sonuçlar ayrı ayrı kategorilerden meydana geldiği durumlarda denetimli öğrenme süreci sınıflandırma olarak adlandırılır. Dolayısıyla sonuç değişkeni bu durumda kategorik bir değişkendir. Haber kategorileri veya film türleri buna örnek verilebilir.

Sınıflandırma tabanlı olarak duygu analizi çalışmasında kullanılan algoritmaları şu şekilde sıralayabiliriz;

- Destek vektör makinaları, veri ve metin madenciliği gibi sınıflandırma tabanlı çalışmalarında kullanılan Cortes tarafından önerilen bir denetimli makine öğrenmesi algoritmasıdır. İki ya da daha fazla sınıflı durum uzayında sınıfları birbirinden ayıracak düzlemin nasıl çizileceğini belirler. Bu çalışma kapsamında iki sınıflı veri seti kullanılmıştır. Çizilen bu düzlem iki sınıfa da maksimum uzaklıkta olması gerekmektedir [54].
- Naive Bayes öğrenme, sınıflandırma tabanlı problemlerin çözümü için önerilen istatistiksel bir denetimli makine öğrenmesi algoritmasıdır. Eğitim verileri kullanılarak algoritmanın eğitilmesi işleminden sonra test verilerinin olasılıksal olarak belli sınıflardan hangisine daha yakın olduğuna karar veren bir mekanizmaya sahiptir [55].
- k en yakın komşu algoritması, sınıflandırma tabanlı problemlerin çözümü için önerilen bir denetimli makine öğrenmesi algoritmasıdır. Test verisinin sınıflandırılması işleminde en yakın k tane eğitim verisine bakılarak hangi sınıfa daha çok benziyorsa o sınıfa ait olduğunun tespit edilmesi işlemi olarak tanımlanmaktadır [56]. k sayısı tek sayı olarak seçilmesi sınıf tahminin yapılmasında kolaylık sağlamaktadır. Bu çalışmada k değeri '3' olarak seçilmiştir.

Regresyon: Tahmin etmek istediğimiz sonuç sürekli sayısal bir değişken olduğunda, denetimli öğrenme algoritmaları regresyon olarak bilinir. Konut fiyatları veya insanların ağırlıkları buna örnek olabilir.

- Karar ağaçları, regresyon tabanlı sınıflandırma problemlerin çözümü için önerilen bir denetimli makine öğrenmesi algoritmasıdır. Veri ve metin madenciliğinde belirli bir amaca ulaşmak adına strateji türetmek için yaygın olarak kullanılan bir yöntem olmasına rağmen, aynı zamanda makine öğrenmesi çalışmalarında da yaygın olarak kullanılmaktadır. Karar ağaçlarıyla eğitim verileri kullanılarak çıkarılan özelliklerin birbirlerine göre hangisinin daha önemli olduğuna dayanarak oluşturulan bir ağaç ve karar modeli bulunmaktadır. Test verilerinin bu ağaç ve karar modeli üzerinde dolaşmasıyla sınıf ataması yapılması işlemi olarak özetlenebilir [57].

Özellikle duygu analizi çalışmasında sayısal verilerden çok kategorik değişkenlerle ilgilenildiği için yürütülen bu tez çalışmasında sınıflandırılma temelli problem çözme yaklaşımı benimsenmiştir.

5.2.2. Denetimsiz Makine Öğrenmesi

Denetimsiz öğrenme, bir model oluşturmak için önceden etiketli eğitim verileri örnekleri gerektirmeyen belirli ML tekniklerini veya algoritmaları ifade eder. Genellikle, çözmeye çalıştığımız soruna bağlı olarak, metin veya sayısal olabilen bir veri koleksiyonu vardır. Özellik çıkarımı olarak bilinen bir işlemi kullanarak veri noktalarının her birinden özellikler üretilmekte ve her veri noktası için özellik kümesini algoritmaya tanıtılması sağlanmaktadır. Veriden konu bütünlüğüne dayalı belgeleri kümeleme veya özetleme gibi teknikler kullanarak benzer veri noktalarını bir araya getirmeye çalışmak ve anlamlı kalıplar üretmek kullanılan denetimsiz makine öğrenme algoritmaların genel özellikleri arasında yer almaktadır. Belge kümeleme olarak da adlandırılan bu yöntem, metin belgelerinin sınıflandırılmasında son derece yararlıdır. Burada, dokümanları yalnızca özelliklerine, benzerliklerine ve niteliklerine dayalı olarak önceden etiketlenmiş veriler üzerinde herhangi bir model oluşturmaksızın gruplar halinde kümelendirme işlemi yapılır. Denetimsiz öğrenme ile doğru ya da yanlış cevap yoktur. Sadece makine öğrenme algoritmasını çalıştıran ve hangi kalıpların ve sonuçların meydana geldiğini gösteren bir durum oluşturmaktadır [58].

Kümeleme: Benzer özelliklere sahip nesneleri bir araya getirmek ya da benzer özelliklere sahip olmayan nesneleri ayırtmak ve bu nesneleri gruplandırmak için kullanılan bir yöntemdir. Kümeleme, makine öğrenmesi ve veri madenciliğinde kullanılan yaygın bir tekniktir. Ayrıca kümeleme, nesnelerin istatistiksel veri analizi ve genellemesi için kullanılabilir.

6. DUYGU ANALİZİ UYGULAMA ÇALIŞMASI

Bu tez çalışmasında Twitter verileri üzerinde duygu analizi çalışması yapılması amaçlanmıştır. Bu çalışmanın ilk adımı olarak verilerin toplanması işlemi gerçekleştirilmesi gerekmektedir. Sosyal medya üzerinde çalışma yapan araştırmacıların en fazla zaman ve güç kaybettikleri aşama, verilerin toplanması ve veri setlerinin oluşturulması olarak bilinmektedir. Sosyal ağlarda birçok farklı çeşitte veri toplanma olanağı bulunmaktadır. Literatüre baktığımız zaman verilerin toplanmasına yönelik birçok araç ve yöntem karşımıza çıkmaktadır. Bunlar arasında en çok kullanılanlar arasında özel olarak tasarlanmış API'ler, web crawling, web scraping işlemleri, scriptler, botlar vb. yer almaktadır. Bu çalışmada Twitter verileri baz alınarak veri setlerinin anlamlı ve düzenli bir şekilde elde edilmesini ve duygu analizi çalışmasının yürütülmesi amaçlanmıştır.

6.1. Verilerin Toplanması

Önerilen sistemde kullanılan veri setinin ayarlanması için ilk önce TweetSharp isimli Twitter Api'si incelenmiştir. Kullanılan bu Api kullanıcısına belirli kısıtlamalar getirdiği ve gerekli sayıda Twitter verilerini makul sürede sunmadığı için uygulamada kullanılması uygun görülmemiştir. Bunun yerine Sentiment140 isimli, Stanford Üniversitesi'nde Bilgisayar Bilimleri lisansüstü öğrencileri Alec Go, Richa Bhayani ve Lei Huang tarafından oluşturulmuş etiketli Twitter veri seti kullanılmıştır. Bu veri setinde yaklaşık 1.6 milyon pozitif ve negatif etiketine sahip Twitter verisi bulunmaktadır. Bu veriler toplanıp sınıflandırma işlemi yapılırken her bir verinin içinde geçen emojilerden faydalanılmıştır. Örneğin, "😊" mutluluk ifade eden bir emoji olduğu için pozitif etiketine sahip olduğu kabul edilmiştir. Aynı şekilde "☹" üzüntü ifadesini içeren verinin negatif etiketiyle sınıflandırılması yapılmıştır. Yürütülen tez çalışmasında 12000 adet Twitter verisi kullanılmıştır. 6 farklı veri seti oluşturulmuş ve bunların her birinde bulunan pozitif ve negatif etiketine sahip Twitter verisi sayısı her bir set için belirtilmiştir.

6.2. Makine Öğrenmesi Tekniklerinin Kullanılması

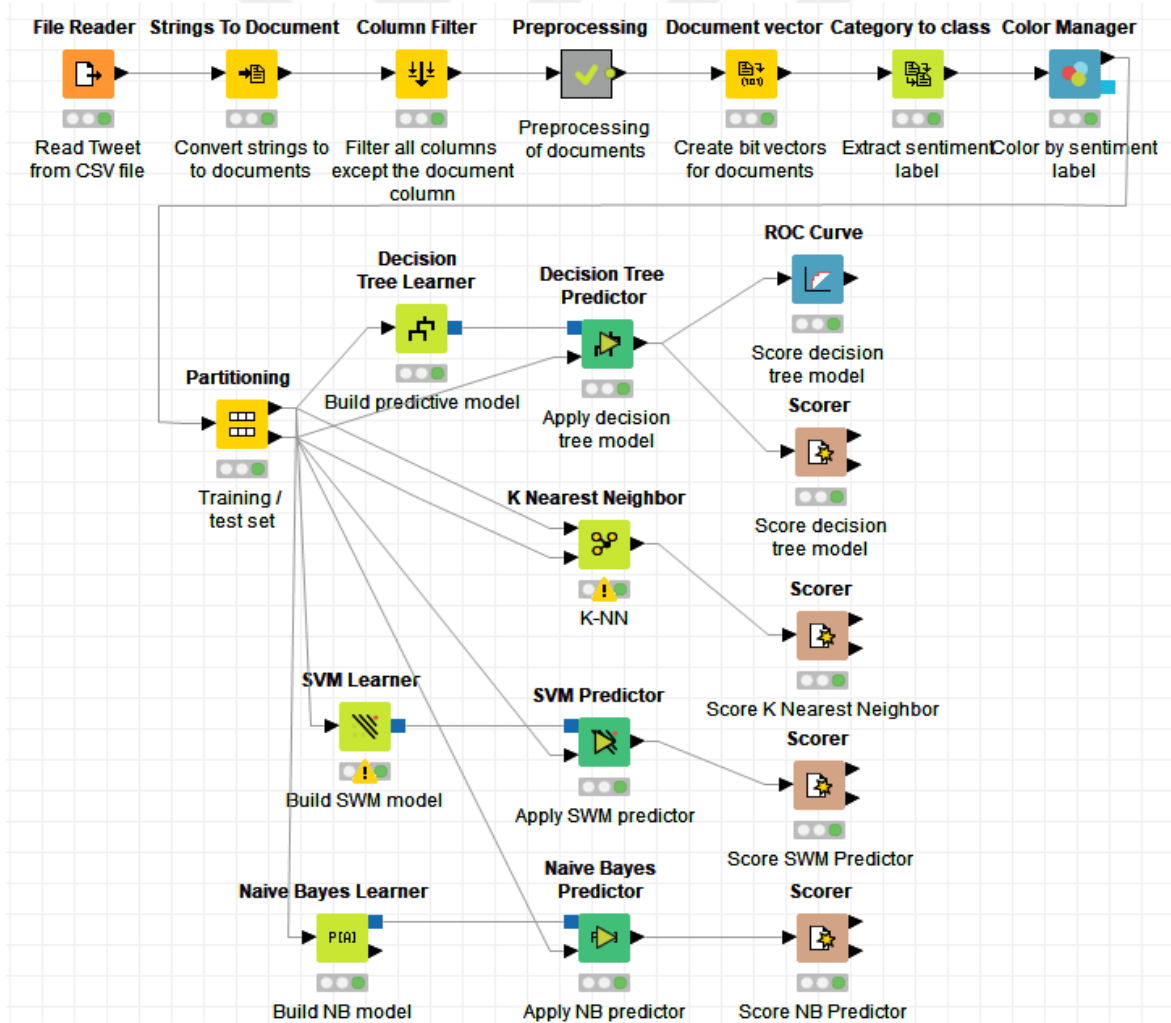
Çalışmasının bu bölümünde KNIME isimli veri ve metin madenciliği aracı kullanılmıştır. KNIME veri odaklı yenilikler öne süren açık kaynak kodlu bir yazılımdır. KNIME, metin işleme özelliği, sayesinde metinsel verileri okumak, işlemek ve sayısal verilere (belge ve terim vektörleri) dönüştürmek üzere tasarlanmış ve geliştirilmiştir. KNIME sayesinde kelime frekansları hesaplanabilir, anahtar kelimeler çıkarılabilir ve belgeler görselleştirilebilir.

Dokümanları kümelemek veya sınıflandırmak için bazı KNIME düğümleri kullanılmaktadır. Sayısal vektörler kullanılan bu düğümler sayesinde oluşturulur.

KNIME aracı kullanılarak duygu analizi çalışması yürütmek için temelde altı farklı adımın gerçekleştirilmesi gerekir. Bu adımlar şunlardır:

- Giriş/Çıkış
- Zenginleştirme
- Önişleme
- Frekanslar
- Dönüşüm
- Görselleştirme

Tüm bu adımları içeren KNIME programıyla oluşturulmuş sistem Şekil 6.1’de gösterildiği gibidir.



Şekil 6.1. KNIME kullanılarak oluşturulmuş duygu analizi çalışması

Şekil 6.1’de görüldüğü gibi her bir düğüm belirli bir görevi üstlenmektedir. Bunlardan ilki “File Reader” dosya okuma düğümü, .csv uzantılı dosyadan hazırlanmış etiketli veri setini okuma işlemini yapmaktadır. Şekil 6.2’de .csv uzantılı dosyadan okunan veriler listelenmektedir. “Text” sütununda Twitter verisi “Sentiment” sütununda ise ait olduğu sınıf etiketi yer almaktadır.

File Table - 0:289 - File Reader (Read Tweet)

File

Table "pos_neg_dataset.csv" - Rows: 2000 Spec - Columns: 2 Properties Flow Variables

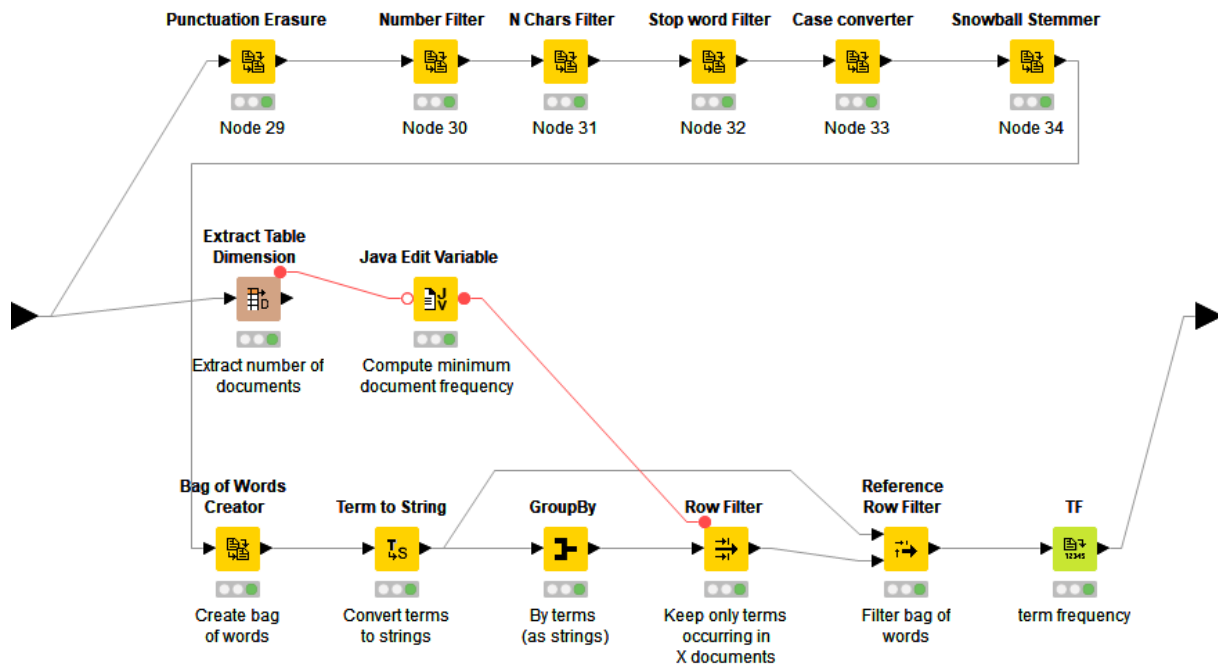
Row ID	S Text	S Sentiment
Row408	i hope everyone is at church so i can just go to my concert in peace on wed. dont have to worry about rushing back to church to visit	NEG
Row409	- i just spilt the last of my jordans cereal on the floor iâ€™m distraught http://tumblr.com/xxw2329lh	NEG
Row410	I miss my family and friends.	NEG
Row411	I see I missed the nectar sucking dirty talk	NEG
Row412	I want a freaking puppy now! A cute little black yorkshire terrier.	NEG
Row413	- I want irissa back http://tumblr.com/xpn1wrj48	NEG
Row414	I'm glad diversity got through...but what about stavros!	NEG
Row415	I'm here to tell ya ... very yummy. Decided to go on a hunt for blender drinks ... know any good ones?	POS
Row416	I'm out of Nutella	NEG
Row417	is it rily tht serious? I'm abt to go thru my fonebook...	NEG
Row418	is listenin to my new fav band more amor	POS
Row419	it is on environmental studies and driving me mental	NEG
Row420	I've always seen bits & amp	NEG
Row421	Just another day in paradise.	POS
Row422	Just found a bone in my chicken nugget Not cool	NEG
Row423	- Like I said, my back still fucking hurts and iâ€™m going to complain about it like no oneâ€™s business.... http://tumblr.com/x6n25amd5	NEG
Row424	love you! <	POS
Row425	my aunties dog is getting put down hes my dogs brother aswell	NEG
Row426	- my head feels like its going to explode - mmm, what a horrendous image... the mess!	NEG
Row427	my head is killing meee	NEG
Row428	- My name is Amy. IiçYsm only three My eyes are swollen I cannot see, I must be stupid, I must be bad What... http://tumblr.com/xxfj1uain0	NEG
Row429	- my picture with Kris Karmada is gone forever, its not in my comments, on my mysapce or on my... http://tumblr.com/xzg1wy4ij	NEG
Row430	Â«Confirmed: Missing Air France Flight 447 has Crashed in the AtlanticÂ»: http://bit.ly/BxdVL	NEG
Row431	about 1/4 of my myspace comments are gone. cause someone hacked onto wills account. and kept abusing others so he had to delete it.	NEG
Row432	about school..... it will all be over soon can't wait to celebrate STRES till then	NEG
Row433	abt missing Air France plane. Follow here for best air news source: http://tinyurl.com/m2p6yk	NEG
Row434	AFI are not at reading til sunday this sucks !!!!!!!!	NEG
Row435	after a crap night was looking forward to being cheered up, but no. Never mind.	NEG
Row436	ah the postman didn't have my package. What am I goon to do for the rest of the day....	NEG
Row437	ahh school on monday!!!! not ready	POS
Row438	ahhh. Drizzy. Mr. Rogers, you are indeed the best.	POS
Row439	Airport Express doesn't rock enough to allow wireless scanning with the Canon pixma MX310, but otherwise it's wireless goodness.	NEG
Row440	Alistair's friends are all in the kitchen and I'm in my pjs...I JUST WANT A CUP OF TEA, DAMMIT!	NEG
Row441	all day.. no play.	NEG
Row442	all dressed up and no bicyclist for Monday nite ride n Henderson KY http://bit.ly/10H4ne	NEG
Row443	all i want is my ma in my arms that's all i want is you in my arms	NEG
Row444	all my friends are gone haley n katie @ camp Paris @ moms Landin n Ayonna @ aunts n moms im so bored	NEG
Row445	all nighter. yeeeeeeeeeeeeessshhh what a nightmare up coming 2 weeks. cannot wait to graduate!	NEG

Şekil 6.2. “File Reader” düğüm çıktısı

Kullanılan diğer düğüm “Strings To Document”, girilen her verinin doküman olarak kaydedilmesini sağlamaktadır.

“Column Filter” düğümü eğer birden fazla sütun varsa ve kullanılmıyorsa bunların aktif ya da pasif olmasını ayarlamak için kullanılmaktadır.

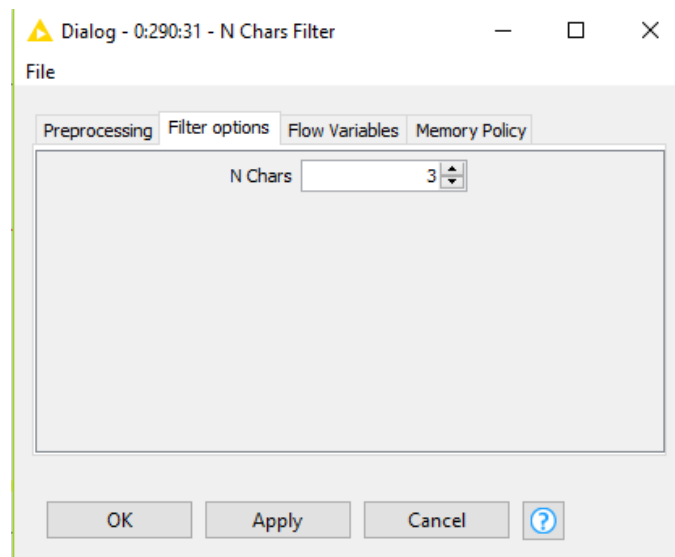
“Preprocessing” düğümü birden çok düğümün bir araya gelmesiyle oluşmuştur. Bu düğümde veri temizlemeden terim frekansına kadar birçok önışlem aşaması gerçekleştirilmektedir. İlgili tüm işlemlere ait KNIME program parçacığı Şekil 6.3’te gösterildiği gibidir.



Şekil 6.3. “Preprocessing” düğümünün alt düğümleri

“Punctuation Erasure” düğümü noktalama işaretleri kaldırma işlemini yapmaktadır.

“N Chars Filter” düğümüne ait yapılandırma ekranı Şekil 6.4’te gösterildiği gibidir.



“N Chars Filter” düğümü girilen bir N değerinden daha az harfe sahip söz öbeklerini silme işlemini yapar. Bu düğüm için bizim belirlediğimiz N değeri 3 olup, 3’ten az harfe sahip ifadeler silinecektir.

“Case Converter” düğümüyle verinin içindeki tüm karakterlerin büyük ya da küçük harf dönüşümü yapılmıştır. Tüm veriler aynı formda küçük harf olacak şekilde hazırlanmıştır.

“Preprocessing” düğümüne ait alt düğümlerinin içerisinde terim frekans değerinin ayarlanması için “Java Edit Variable” düğümü kullanılmıştır. Terim frekans değeri 20 olarak ayarlanmıştır. Bu işlem için “Java Edit Variable” düğümünün yapılandırılması “out_MinNumberOfDocuments = (int)((double)v_NumberRows / 100.0d);” şekilde tanımlanmıştır.

“Bag of Words Creator” düğümü ile tüm veri setinde geçen her bir sözcüğü köküne ayırma işlemi yapılmıştır.

“Group By” düğümü ile tekrar eden köklerin sayısı çıkarılmıştır.

“Row Filter” düğümü ile 20’den az frekansa sahip sözcükler tespit edilip filtrelenmiştir.

“TF” düğümü ile her bir kökün terim sıklığı değeri hesaplanmıştır.

“Document Vector” düğümü metin verilerini sayısal hale dönüştürme işlemini gerçekleştirme görevine sahiptir. Kullanılan birinci deney seti için Şekil 6.5’te gösterildiği gibi 43 tane kök 20 frekans değerinden yüksek olduğu için filtrelemeden başarıyla kurtulmuştur. 42 adet kök sözcük artık özellik seçimi için kullanacağımız anahtar dizin haline gelmiş olmaktadır.

“Category to Class” düğümü sayısal hale dönüştürülen her bir Twitter verisinin ait olduğu sınıfı etiketlemektedir.

“Color Manager” düğümü ait olunan sıfır etiketine göre renklendirme işlemi yapıp eğer etiket pozitifse yeşil, negatifse kırmızı renk ayarlaması yapmaktadır.

“Partitioning” düğümü mevcut Twitter verilerini eğitim ve test verisi olacak şekilde iki kısma ayırmaktadır. Bu çalışma kapsamında tüm deney setleri için Twitter verilerinin %72’si eğitim, kalan %28’i ise test için kullanılmıştır.

Table with Colors - 0:276 - Color Manager (Color by sentiment)

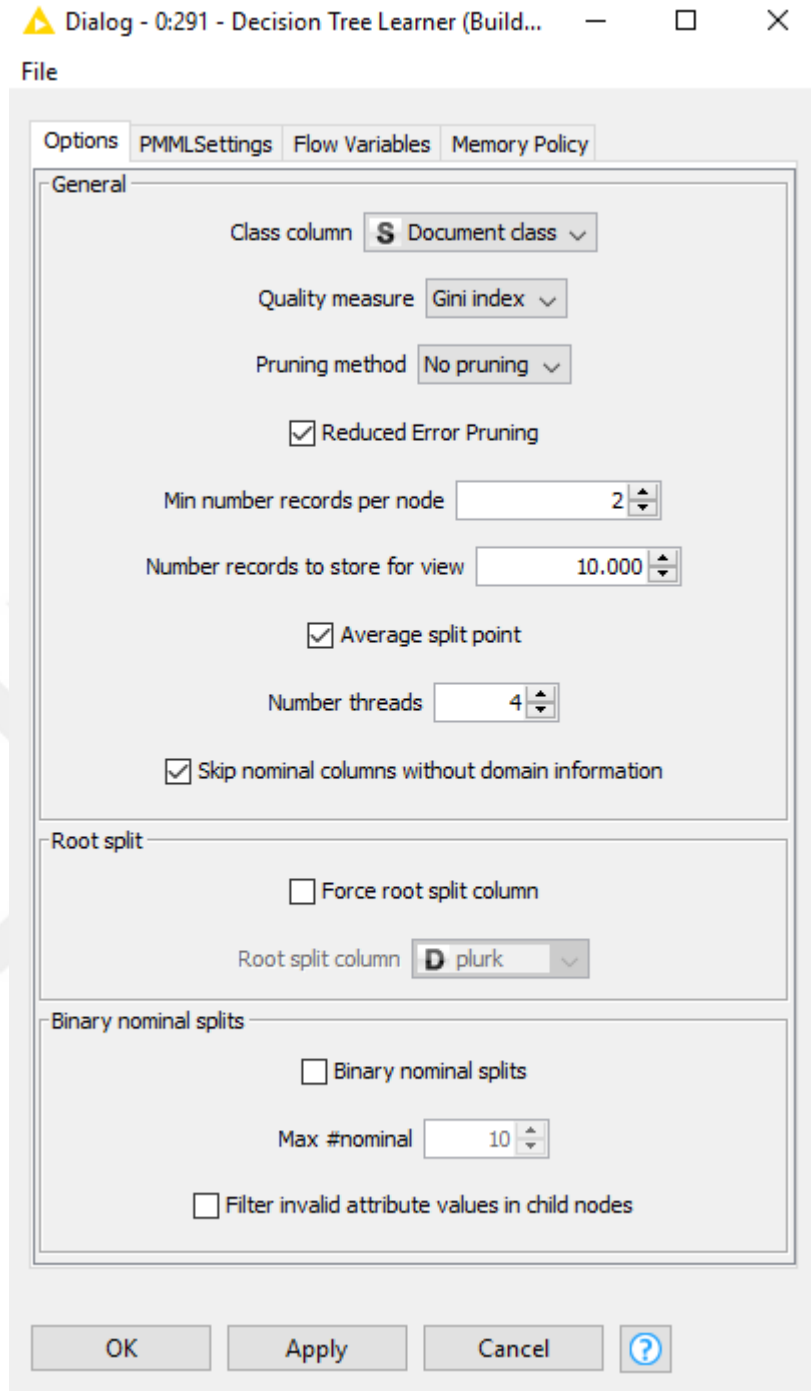
File

Table "default" - Rows: 1075 Spec - Columns: 44 Properties Flow Variables

Row ID	D bad	D sad	D feel	D sleep	D peop	D plea	D hope	D gonna	D guess	D weekend	D wait	D days	D dont	D lol	D happy	D tr
Row 1891	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1892	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1893	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Row 1894	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1895	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1896	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1897	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1898	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Row 1899	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Row 1900	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Row 1901	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1902	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1903	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1904	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1905	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1906	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Row 1907	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Row 1908	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1909	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1910	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1911	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Row 1912	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1913	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1914	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1915	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0
Row 1916	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Row 1917	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
Row 1918	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1919	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Row 1920	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1921	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Row 1922	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1923	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1924	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Row 1925	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1926	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1927	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Row 1928	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Row 1929	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1930	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1931	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Row 1932	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
Row 1933	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1934	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1935	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Row 1936	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1937	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Row 1938	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Row 1939	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1940	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Row 1941	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1942	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Row 1943	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Row 1944	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

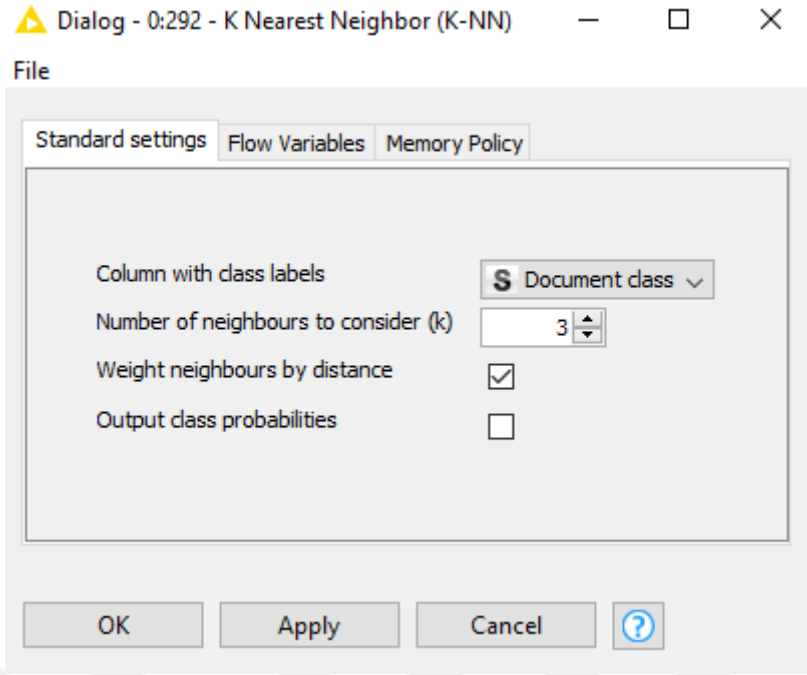
Şekil 6.5. “Document Vector” ile verilerin sayısallaştırılması

Karar ağaçları, Destek vektör makinaları, k en yakın komşu algoritması ve Naive Bayes algoritmalarını kullanmak için gerekli düğümler seçilmiştir. Seçilen bu düğümlerdeki algoritmaların parametreleri yapılandırma ekranlarında ayarlanmaktadır. Tüm deney setlerinde parametreler sabit tutulmuştur. Şekil 6.6’da karar ağaçları düğümü için yapılandırma ekranı yer almaktadır. Şekil 6.6’da “Quality Measure” seçeceği “Gini index” olarak ayarlanmıştır. Karar ağaçlarında kullanılan budama işlemine yer verilmeyerek “Pruning method” seçeneği “No pruning” olarak ayarlanmıştır. Karar ağaçları için kullanılan düğümün diğer var olan yapılandırma ayarları varsayılan olarak sabit bırakılmıştır.



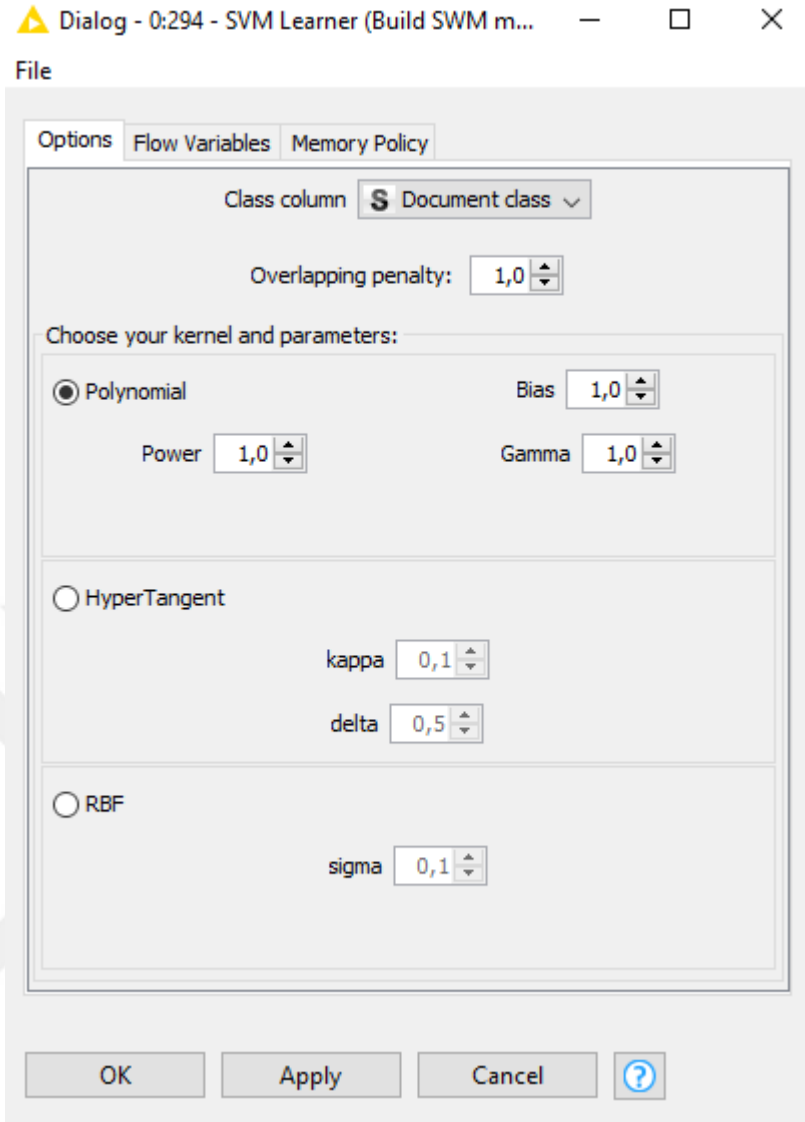
Şekil 6.6. Karar ağaçları yapılandırma ekranı

Şekil 6.7’de k en yakın komşu algoritması için kullanılan düğümün yapılandırma ekranı yer almaktadır. Bu düğümde k değeri ‘3’ olarak seçilmiştir. Mesafeye göre komşuların ağırlıklandırılması işlemi için “Weight neighbours by distance” kutucuğu aktif edilmiştir. k en yakın komşu algoritması için kullanılan düğümün diğer var olan yapılandırma ayarları varsayılan olarak sabit bırakılmıştır.



Şekil 6.7. k en yakın komşu algoritması yapılandırma ekranı

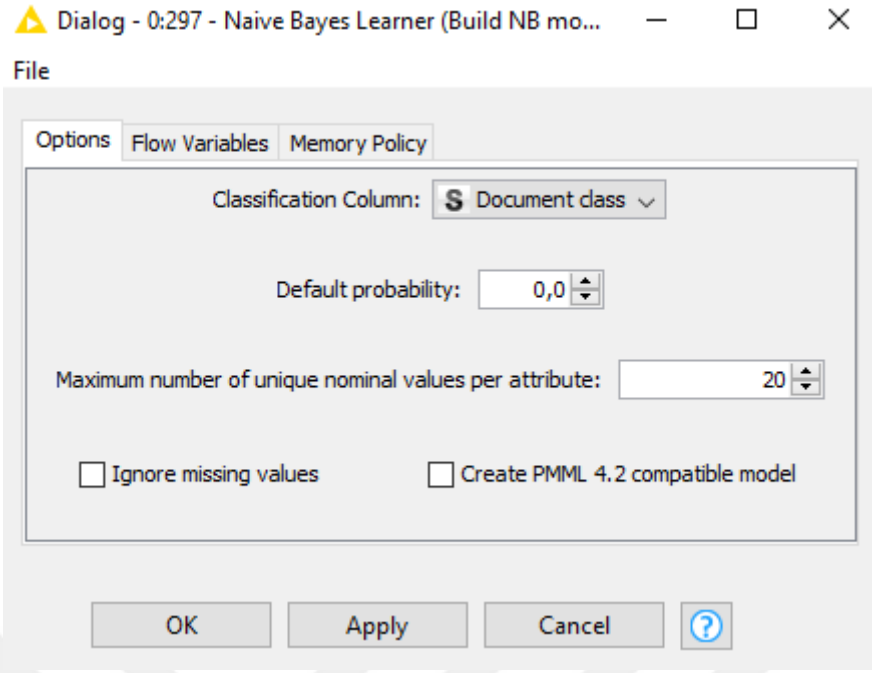
Şekil 6.8’de destek vektör makinaları için kullanılan düğümün yapılandırma ekranı yer almaktadır. “Polynomial” kutucuğu aktif edilerek, “Bias” değeri ‘1.0’, “Power” değeri ‘1.0’ ve “Gamma” değeri ‘1.0’ olarak atanmıştır. Destek vektör makinaları için kullanılan düğümün diğer var olan yapılandırma ayarları varsayılan olarak sabit bırakılmıştır.



Şekil 6.8. Destek vektör makinaları yapılandırma ekranı

Şekil 6.9’da Naive Bayes algoritması için kullanılan düğümün yapılandırma ekranı yer almaktadır. “Maximum number of unique nominal values per attribute” değeri ‘20’ olarak atanarak, özellik başına benzersiz nominal değerlerin maksimum sayısı ayarlanmıştır. Naive Bayes algoritması için kullanılan düğümün diğer var olan yapılandırma ayarları varsayılan olarak sabit bırakılmıştır.

Karar ağaçları, destek vektör makinaları, k en yakın komşu algoritması ve Naive Bayes algoritmalarının parametreleri tüm deney setleri için sabit tutulmuştur. Bu algoritmaların duygu analizi üzerindeki başarımları SSA algoritması tabanlı geliştirilen modellerle karşılaştırılmıştır.



Şekil 6.9. Naive Bayes yapılandırma ekranı

6.3. Sosyal Örümcek Algoritması (SSA)'nın Kullanılması

Sosyal Örümcek Algoritması'nın biyolojik yapısı ve matematiksel modeli Bölüm 2'de ayrıntılı bir şekilde açıklanmıştır. Bu bölümde KNIME yardımcı aracı kullanılarak oluşturulan doküman vektör sayesinde Sosyal Örümcek Algoritması'nın probleme uygun olarak modelleme çalışması yapılmıştır.

6.3.1. SSA'nın Başlangıç Parametrelerinin Ayarlanması

Sosyal örümcek algoritmanın birinci deney seti için başlangıç parametreleri Tablo 6.1'de gösterildiği gibidir.

Tablo 6.1. Sosyal Örümcek Algoritmasının başlangıç parametreleri

Boyut :	42 (her bir veri seti için boyut değişkenin değeri KNIME ile elde edilen özellik sayısı kadardır.)
Alt Sınır :	0
Üst Sınır :	1
Maksimum İterasyon Sayısı :	1000
Popülasyon Sayısı :	30
Deney Sayısı :	30

Her bir veri seti için toplam deney sayısı 30 olarak belirlenmiştir. Her bir deney için maksimum iterasyon sayısı 1000 olarak belirlenmiştir. 30 deneyin sonunda en iyi deney sonucu çözüm olarak kabul edilmiştir. Duygu analizi çalışması için SSA’da bulunan örümcek (popülasyon) sayısı 30 olarak belirlenmiştir. Her bir örümceğe 0 ile 1 arasında ağırlıklar verilmiştir. Bir sonraki iterasyon sonunda örümceklerin ağırlıkları güncellenmektedir. Her bir iterasyonda en iyi uygunluğa sahip örümcek var olan iterasyon için en iyi çözüm olarak kabul edilmiştir. Boyut olarak 42 seçilmesinin sebebi KNIME aracının birinci veri seti için 42 özellik belirleyerek doküman vektörünü oluşturmasıdır. Her bir veri seti için sadece boyut sayısı özellik sayısına göre değişiklik göstermektedir. Diğer parametreler sabit kalmaktadır.

6.3.2. Uygunluk Fonksiyonun Ayarlanması

Uygunluk fonksiyonu her bir veri setine ait eğitim verileri üzerinde negatif ve pozitif veriler için ayrı ayrı en iyi uygunluk değerine sahip modeli elde etmek için tasarlanmıştır. Negatif verileri en iyi temsil edecek model ve pozitif verileri en iyi temsil edecek model oluşturulduktan sonra test verileri üzerinde doğrulama işlemi yapılmıştır. Uygunluk değeri Denklem 6.1 ile hesaplanmıştır.

$$\begin{aligned} \text{uygunluk} = & a_1(\text{Doğruluk}) + a_2(\text{Hassasiyet}) \\ & + a_3(\text{Duyarlılık}) + a_4(\text{Seçicilik}) \end{aligned} \quad (6.1)$$

a_1 , a_2 , a_3 ve a_4 ağırlıkları rasgele verilen ve toplamı 1 olan sayılardır. Uygunluk fonksiyonunda eğer doğruluk değerinin yüksek olmasını istiyorsak a_1 değerinin, hassasiyet değerinin yüksek olmasını istiyorsak a_2 değerinin, duyarlılık değerinin yüksek olmasını istiyorsak a_3 ve seçicilik değerinin yüksek olmasını istiyorsak a_4 değerinin yüksek olarak seçilmesi beklenir. Bu çalışma kapsamında tüm veri setleri için $a_1=0.4$, $a_2=0.2$, $a_3=0.2$ ve $a_4=0.2$ olacak şekilde ayarlanmıştır. *DP*, gerçek durumun ve tahmin sonucunun pozitif olduğu durumdur. *DN*, gerçek durumun ve tahmin sonucunun negatif olduğu durumu ifade eder. *YP*, gerçek durumun negatif tahmin sonucunun pozitif olduğu durumu gösterir. *YN*, gerçek durumun pozitif tahmin sonucunun negatif olduğu durumu temsil eder. Doğruluk analizi için uygulanan bu yöntem daha ayrıntılı olarak Tablo 6.2’de gösterilmiştir.

Tablo 6.2. Doğruluk Analizi yöntemindeki durum uzayı

Tahmin Sonucu	Gerçek Durum	
	Pozitif	Negatif
Pozitif	Doğru Pozitif (<i>DP</i>)	Yanlış Pozitif (<i>YP</i>)
Negatif	Yanlış Negatif (<i>YN</i>)	Doğru Negatif (<i>DN</i>)

Sosyal Örümcek Algoritması kullanılarak oluşturulmuş modellemenin doğruluğu için karmaşıklık matrisi kullanılmıştır. Doğruluk, duyarlılık, hassasiyet ve seçicilik değerleri Denklem 6.2’de gösterildiği gibi hesaplanmıştır.

$$\begin{aligned}
\text{Doğruluk} &= \left(\frac{(DP + DN)}{(DP + YP + YN + DN)} \right) \\
\text{Duyarlılık} &= \left(\frac{(DP)}{(DP + YN)} \right) \\
\text{Hassasiyet} &= \left(\frac{(DP)}{(DP + YP)} \right) \\
\text{Seçicilik} &= \left(\frac{(DN)}{(YP + DN)} \right)
\end{aligned} \tag{6.2}$$

Elde edilen doğruluk, duyarlılık, hassasiyet ve seçicilik değerleri her iterasyonda uygunluk fonksiyonunda yerine koyularak en iyi uygunluğa sahip model elde edilmeye çalışılmıştır. Pozitif ve negatif verilerini temsil eden en iyi uygunluğa sahip modeller test verileri üzerinde doğrulama işlemi için kullanılmıştır. Her bir test verisi hem negatif hem de pozitif modelle Jaccard benzerliği kullanılarak karşılaştırılmıştır. Hangi sınıf etiketine sahip modele daha çok benziyorsa o Twitter test verisi benzerliği yüksek olan sınıfa ait olduğu kabul edilmiştir. Test verileri üzerinde doğruluk, duyarlılık, hassasiyet ve seçicilik değerleri hesaplanarak modelin başarımı hesaplanmıştır. Bu işlemler 6 farklı veri seti üzerinde ayrı ayrı yapılmıştır.

Uygunluk fonksiyonun oluşturulması aşamasında MATLAB aracından faydalanılmıştır. İlk önce algoritmanın çalıştırılması ile birinci veri seti için üretilen 42 boyutlu 0 ile 1 arasında sayılardan oluşmuş dizi *round* fonksiyonu ile 0 ve 1’lerden oluşan bit dizisi formatına dönüştürülmüştür. Birinci deney setin için 2000 satır 42 sütundan oluşan Tablo 6.3 KNIME

yöntem ile hangi verinin pozitif ya da negatif olduğuna dair tahmini etiketlendirme yapılmıştır. Yapılan tahmini etiketlendirme işleminin ardından doğruluk analizi yapılmış, gerçek sınıf etiketi ve tahmini sınıf etiketi karşılaştırılmıştır. Bu işlem hem pozitif hem de negatif verileri temsil edecek en iyi modeli bulmak için ayrı ayrı yapılmıştır.

6.4. Değerlendirme ve Sonuç

Duygu analizi çalışmasında Twitter verileri üzerinde makine öğrenmesi teknikleri ve SSA'sı kullanılarak bir sınıflandırma çalışması gerçekleştirilmiştir. Kullanılan yöntemlerinin başarımlar ölçütleri karmaşıklık matrisi yardımıyla hesaplanmıştır. 6 farklı deney seti için başarımlar ölçütleri ayrıntılı bir şekilde incelenmiştir.

1. deney seti için kullanılan algoritmalarla hesaplanan başarımlar ölçütlerinin istatistiksel değerleri Tablo 6.6'da gösterilmiştir.

Tablo 6.6. 1. deney seti için kullanılan algoritmaların karşılaştırılması

		Başarımlar Ölçütleri			
		Doğruluk	Duyarlılık	Hassasiyet	Seçicilik
Kullanılan Algoritmalar	Karar Ağaçları	0.631	0.617	0.604	0.644
	k-NN (k=3)	0.591	0.22	0.705	0.919
	SVM	0.621	0.418	0.648	0.8
	Naive Bayes	0.608	0.66	0.571	0.562
	SSA	0.652	0.681	0.589	0.636

1. deney seti için toplam 2000 tane Twitter verisi kullanılmıştır. Kullanılan bu 2000 Twitter verisinin 1000 tanesi negatif etikete diğer 1000 tanesi ise pozitif etikete sahiptir. KNIME yardımcı aracında her bir kelimenin geçme sıklığı tüm veri setleri için ortak olarak en az 20 olacak şekilde ayarlanmıştır. Eğer herhangi bir Twitter verisi içinde belirlenen frekans değerinin üstünde bir kelime yoksa o Twitter verisi 0 matrisinden oluşacaktır. Sıfır matrisinden oluşacak olan bu veri KNIME aracı ile geliştirilen uygulama çalışmasındaki “Row Filter” düğümüyle silinecektir. Herhangi bir Twitter verisinin doküman vektörünün oluşması için içerisinde en az 1 adet belirlenen frekans değerinin üstünde kelime geçmiş olması gerekmektedir. 1. deney setinde 42 özellik çıkarılmıştır. Buna bağlı olarak SSA'nın boyut başlangıç parametresi 42 olarak ayarlanmıştır. 1075 adet doküman vektörü oluşturulmuştur. Oluşturulan bu 1075 adet doküman vektörünün %72'si eğitim, %28'i ise test

için kullanılmıştır. Tüm algoritmalar ve deney setleri için bu değer sabit tutulmuştur. 1. deney seti için kullanılan algoritmaların başarımları Tablo 6.7’de gösterildiği gibidir.

Tablo 6.7. 1. deney seti için algoritmaların başarımlarının sıralaması

Doğruluk	SSA > Karar Ağaçları > SVM > Naive Bayes > k-NN
Duyarlılık	SSA > Naive Bayes > Karar Ağaçları > SVM > k-NN
Hassasiyet	k-NN > SVM > Karar Ağaçları > Naive Bayes > SSA
Seçicilik	k-NN > SVM > Karar Ağaçları > SSA > Naive Bayes

1. deney seti için en iyi doğruluk değeri SSA, en iyi duyarlılık değeri SSA, hassasiyeti değeri en yüksek olan k-NN ve en yüksek seçicilik değeri k-NN algoritmaları tarafından elde edilmiştir.

2. deney seti için kullanılan algoritmalarla hesaplanan başarımların ölçütlerinin istatistiksel değerleri Tablo 6.8’de gösterilmiştir.

Tablo 6.8. 2. deney seti için kullanılan algoritmaların karşılaştırılması

		Başarımların Ölçütleri			
		Doğruluk	Duyarlılık	Hassasiyet	Seçicilik
Kullanılan Algoritmalar	Karar Ağaçları	0.776	0.883	0.825	0.482
	k-NN (k=3)	0.754	0.974	0.759	0.145
	SVM	0.786	0.883	0.835	0.518
	Naive Bayes	0.741	0.757	0.874	0.699
	SSA	0.713	0.826	0.734	0.561

2. deney seti için toplam 2000 tane Twitter verisi kullanılmıştır. Kullanılan bu 2000 Twitter verisinin 1499 tanesi negatif etikete diğer 501 tanesi ise pozitif etikete sahiptir. 2. deney setinde 41 özellik çıkarılmıştır. Buna bağlı olarak SSA’nın boyut başlangıç parametresi 41 olarak ayarlanmıştır. 1116 adet doküman vektörü oluşturulmuştur. 2. deney seti için kullanılan algoritmaların başarımları Tablo 6.9’da gösterildiği gibidir.

Tablo 6.9. 2. deney seti için algoritmaların başarımlarının sıralaması

Doğruluk	SVM > Karar Ağaçları > k-NN > Naive Bayes > SSA
Duyarlılık	k-NN > SVM = Karar Ağaçları > SSA > Naive Bayes
Hassasiyet	Naive Bayes > SVM > Karar Ağaçları > k-NN > SSA
Seçicilik	Naive Bayes > SSA > SVM > Karar Ağaçları > k-NN

2. deney seti için en iyi doğruluk değeri SWM, en iyi duyarlılık değeri k-NN, hassasiyeti değeri en yüksek olan Naive Bayes ve en yüksek seçicilik değeri Naive Bayes algoritmaları tarafından elde edilmiştir.

3. deney seti için kullanılan algoritmalarla hesaplanan başarımların ölçütlerinin istatistiksel değerleri Tablo 6.10’da gösterilmiştir.

Tablo 6.10. 3. deney seti için kullanılan algoritmaların karşılaştırılması

		Başarım Ölçütleri			
		Doğruluk	Duyarlılık	Hassasiyet	Seçicilik
Kullanılan Algoritmalar	Karar Ağaçları	0.696	0.512	0.647	0.817
	k-NN (k=3)	0.623	0.14	0.60	0.939
	SVM	0.706	0.38	0.754	0.919
	Naive Bayes	0.706	0.636	0.626	0.751
	SSA	0.627	0.655	0.648	0.534

3. deney seti için toplam 2000 tane Twitter verisi kullanılmıştır. Kullanılan bu 2000 Twitter verisinin 1160 tanesi negatif etikete diğer 840 tanesi ise pozitif etikete sahiptir. 3. deney setinde 46 özellik çıkarılmıştır. Buna bağlı olarak SSA’nın boyut başlangıç parametresi 46 olarak ayarlanmıştır. 1161 adet doküman vektörü oluşturulmuştur. 3. deney seti için kullanılan algoritmaların başarımların sırası Tablo 6.11’de gösterildiği gibidir.

Tablo 6.11. 3. deney seti için algoritmaların başarımların sırası

Doğruluk	SVM = Naive Bayes > Karar Ağaçları > SSA > k-NN
Duyarlılık	SSA > Naive Bayes > Karar Ağaçları > SVM > k-NN
Hassasiyet	SVM > SSA > Karar Ağaçları > Naive Bayes > k-NN
Seçicilik	k-NN > SVM > Karar Ağaçları > Naive Bayes > SSA

3. deney seti için en iyi doğruluk değeri SVM, en iyi duyarlılık değeri SSA, hassasiyeti değeri en yüksek olan SVM ve en yüksek seçicilik değeri k-NN algoritmaları tarafından elde edilmiştir.

4. deney seti için kullanılan algoritmalarla hesaplanan başarımların ölçütlerinin istatistiksel değerleri Tablo 6.12’de gösterilmiştir.

Tablo 6.12. 4. deney seti için kullanılan algoritmaların karşılaştırılması

		Başarım Ölçütleri			
		Doğruluk	Duyarlılık	Hassasiyet	Seçicilik
Kullanılan Algoritmalar	Karar Ağaçları	0.689	0.656	0.708	0.722
	k-NN (k=3)	0.583	0.87	0.557	0.289
	SVM	0.723	0.594	0.809	0.856
	Naive Bayes	0.691	0.661	0.709	0.722
	SSA	0.693	0.451	0.826	0.732

4. deney seti için toplam 2000 tane Twitter verisi kullanılmıştır. Kullanılan bu 2000 Twitter verisinin 492 tanesi negatif etikete diğer 1058 tanesi ise pozitif etikete sahiptir. 4. deney setinde 45 özellik çıkarılmıştır. Buna bağlı olarak SSA'nın boyut başlangıç parametresi 45 olarak ayarlanmıştır. 1351 adet doküman vektörü oluşturulmuştur. 4. deney seti için kullanılan algoritmaların başarımları sırası Tablo 6.13'te gösterildiği gibidir.

Tablo 6.13. 4. deney seti için algoritmaların başarımları sırası

Doğruluk	SVM > SSA > Naive Bayes > Karar Ağaçları > k-NN
Duyarlılık	k-NN > Naive Bayes > Karar Ağaçları > SVM > SSA
Hassasiyet	SSA > SVM > Naive Bayes > Karar Ağaçları > k-NN
Seçicilik	SVM > SSA > Naive Bayes = Karar Ağaçları > k-NN

4. deney seti için en iyi doğruluk değeri SVM, en iyi duyarlılık değeri k-NN, hassasiyeti değeri en yüksek olan SSA ve en yüksek seçicilik değeri SVM algoritmaları tarafından elde edilmiştir.

5. deney seti için kullanılan algoritmalarla hesaplanan başarımları ölçütlerinin istatistiksel değerleri Tablo 6.14'te gösterilmiştir.

Tablo 6.14. 5. deney seti için kullanılan algoritmaların karşılaştırılması

		Başarım Ölçütleri			
		Doğruluk	Duyarlılık	Hassasiyet	Seçicilik
Kullanılan Algoritmalar	Karar Ağaçları	0.668	0.731	0.667	0.6
	k-NN (k=3)	0.588	0.827	0.573	0.327
	SVM	0.651	0.683	0.66	0.616
	Naive Bayes	0.668	0.673	0.686	0.663
	SSA	0.679	0.718	0.601	0.671

5. deney seti için toplam 2000 tane Twitter verisi kullanılmıştır. Kullanılan bu 2000 Twitter verisinin 495 tanesi negatif etikete diğer 1055 tanesi ise pozitif etikete sahiptir. 5. deney setinde 54 özellik çıkarılmıştır. Buna bağlı olarak SSA'nın boyut başlangıç parametresi 54 olarak ayarlanmıştır. 1418 adet doküman vektörü oluşturulmuştur. 5. deney seti için kullanılan algoritmaların başarımları sırası Tablo 6.15'te gösterildiği gibidir.

Tablo 6.15. 5. deney seti için algoritmaların başarımları sırası

Doğruluk	SSA > Naive Bayes = Karar Ağaçları > SVM > k-NN
Duyarlılık	k-NN > Karar Ağaçları > SSA > SVM > Naive Bayes
Hassasiyet	Naive Bayes > Karar Ağaçları > SVM > SSA > k-NN
Seçicilik	SSA > Naive Bayes > SVM > Karar Ağaçları > k-NN

5. deney seti için en iyi doğruluk değeri SSA, en iyi duyarlılık değeri k-NN, hassasiyeti değeri en yüksek olan Naive Bayes ve en yüksek seçicilik değeri SSA tarafından elde edilmiştir.

6. deney seti için kullanılan algoritmalarla hesaplanan başarımları ölçütlerinin istatistiksel değerleri Tablo 6.16'da gösterilmiştir.

Tablo 6.16. 6. deney seti için kullanılan algoritmaların karşılaştırılması

		Başarım Ölçütleri			
		Doğruluk	Duyarlılık	Hassasiyet	Seçicilik
Kullanılan Algoritmalar	Karar Ağaçları	0.843	0.806	0.801	0.868
	k-NN (k=3)	0.711	0.394	0.765	0.92
	SVM	0.86	0.848	0.809	0.868
	Naive Bayes	0.812	0.727	0.784	0.868
	SSA	0.751	0.735	0.785	0.729

6. deney seti için toplam 2000 tane Twitter verisi kullanılmıştır. Kullanılan bu 2000 Twitter verisinin 661 tanesi negatif etikete diğer 1339 tanesi ise pozitif etikete sahiptir. 6. deney setinde 51 özellik çıkarılmıştır. Buna bağlı olarak SSA'nın boyut başlangıç parametresi 51 olarak ayarlanmıştır. 1482 adet doküman vektörü oluşturulmuştur. 6. deney seti için kullanılan algoritmaların başarımları sırası Tablo 6.17'de gösterildiği gibidir.

Tablo 6.17. 6. deney seti için algoritmaların başarımları sırası

Doğruluk	SVM > Karar Ağaçları > Naive Bayes > k-NN > SSA
Duyarlılık	SVM > Karar Ağaçları > Naive Bayes > SSA > k-NN
Hassasiyet	SVM > Karar Ağaçları > Naive Bayes > SSA > k-NN
Seçicilik	k-NN > SVM = Karar Ağaçları = Naive Bayes > SSA

6. deney seti için en iyi doğruluk değeri SVM, en iyi duyarlılık değeri SVM, hassasiyeti değeri en yüksek olan SVM ve en yüksek seçicilik değeri k-NN algoritmaları tarafından elde edilmiştir. Doğruluk oranının en yüksek olduğu 6. deney setinde %86'lık bir başarı SVM algoritması tarafından sağlanmıştır.

Bu çalışmada doğruluk, duyarlılık, hassasiyet ve seçicilik değerlerine bağlı kalınarak Karar ağaçları, k-NN algoritması, SVM, Destek vektör makinaları ve SSA'nın göstermiş olduğu başarımları 6 farklı deney setinde ayrıntılı olarak incelenmiştir. 6 farklı veri setinde uygulanan 5 farklı algoritmanın başarımları ölçütlerinin aritmetik ortalaması alınarak hesaplanan değerler Tablo 6.18'de gösterilmiştir.

Tablo 6.18. Tüm deney setleri için kullanılan algoritmaların aritmetik ortalaması

		Başarım Ölçütleri			
		Doğruluk	Duyarlılık	Hassasiyet	Seçicilik
Kullanılan Algoritmalar	Karar Ağaçları	0.717	0.701	0.709	0.689
	k-NN (k=3)	0.641	0.571	0.66	0.59
	SVM	0.725	0.634	0.753	0.763
	Naive Bayes	0.704	0.686	0.709	0.711
	SSA	0.686	0.678	0.697	0.645

6 deney sonucunun aritmetik ortalama alınarak oluşturulan Tablo 6.17’de görüldüğü üzere en iyi doğruluk değeri SVM, en iyi duyarlılık değeri Karar ağaçları, hassasiyeti değeri en yüksek olan SVM ve en yüksek seçicilik değeri SVM algoritmaları tarafından elde edilmiştir. Tablo 6.19, 6 deney için kullanılan algoritmaların aritmetik ortalaması dikkate alınarak oluşturulmuştur.

Tablo 6.19. Tüm deney setleri için kullanılan algoritmaların başarımlarının sırası

Doğruluk	SVM > Karar Ağaçları > Naive Bayes > SSA > k-NN
Duyarlılık	Karar Ağaçları > Naive Bayes > SSA > SVM > k-NN
Hassasiyet	SVM > Karar Ağaçları = Naive Bayes > SSA > k-NN
Seçicilik	SVM > Naive Bayes > Karar Ağaçları > SSA > k-NN

6 deney seti için SSA ile oluşturulan model, 1. ve 5. deney setlerinde en iyi doğruluk değerine sahip olmuştur. 1. ve 3. deney setlerinde en iyi duyarlılık değerlerine sahiptir. 4. deney setinde en iyi hassasiyet değerine sahiptir. 5. deney setinde en iyi seçicilik değerine sahiptir. 6 deney seti için SSA’nın doğruluk oranının aritmetik ortalaması %69 olarak hesaplanmıştır. En iyi uygunluğa sahip SVM algoritmasının 6 deney için doğruluk oranının aritmetik ortalama %73 olarak hesaplanmıştır.

SSA’nın başlangıç parametreleri iyi ayarlanırsa, özellik seçimi işlemi uygulanırsa, çapraz doğrulama yapılsa ya da kullanılacak model sayısı artırılırsa SSA’nın duygu analizi çalışmalarında ki başarımlarının artması öngörülmektedir.

7. SONUÇ

Bu tez çalışmasında güncel metasezgisel algoritmaların biyolojik yapıları ve matematiksel modelleri incelenmiştir. Birbirleriyle girdikleri performans yarışında artıları ve eksileri değerlendirilmiştir. Tez çalışmasında kullanılan Sosyal Örümcek Algoritmasının başarısı benzetim çalışmalarıyla gözlemlenmiştir.

Duygu analizi bildiğimiz kadarıyla yapay zeka optimizasyon algoritmalarıyla hiç çalışılmamıştır. Bu çalışma sosyal ağlarda duygunun optimizasyon bakışı açısıyla da analiz edileceğini göstermesi bakımından önemlidir. Çalışma ile ayrıca sosyal örümcek algoritması için yeni bir uygulama alanı tanıtılmıştır.

Bilgisayar ve internet kullanımının artması internetten bilgi çıkarma yöntemlerinde ciddi bir artışa neden olmuştur. Bu tez çalışmasında, literatürde kullanılan bilgiye erişim ve yorumlama aşamaları detaylıca araştırılmıştır. Duygu analizi çalışması Twitter verileri üzerinden yürütülmüştür. Twitter verilerinin elde edilmesi, verilerin temizlenmesi, verilerin sayısal forma dönüştürülmesi, anlamlı sonuçlar çıkarılması ve yorumlanması işlemleri yapılmıştır. KNIME yazılımından yararlanılarak makine öğrenmesi algoritmaları kullanılmıştır. Duygu analizi çalışmalarında daha önce kullanılmayan yeni bir yöntem olarak metasezgisel algoritmalarından Sosyal Örümcek Algoritması yardımıyla duygu analizi çalışması için bir model oluşturulmuştur. Oluşturulan bu model ve makine öğrenmesi yöntemleri karşılaştırılarak sosyal ağlarda metasezgisel algoritmaların kullanılabileceği sonucuna varılmıştır.

Elde edilen sonuçların başarımını arttırmak amacıyla; farklı benzerlik ölçüleri kullanılabilir. Ayrıca her duygu sınıfı için tek bir çözüm/model yerine birden fazla model oluşturulabilir. Nitelik seçimi de entegre edilerek başarımın artırılabilmesi düşünülmektedir. Ayrıca bu tez kapsamında uyarlanan Sosyal Örümcek Algoritması'na sınıflandırıcı toplulukları yöntemi de entegre edilerek duygu sınıflandırma başarımı artırılabilir. Ayrıca problem çok amaçlı bir problem olarak modellenerek de başarılı sonuçlar elde edilebilir.

Gelecek çalışmalarda önerilen yöntemin iyileştirilerek, sosyal ağ analizindeki farklı problemlere çözüm arama algoritması olarak uyarlanması amaçlanmıştır. Uyarlanan bu yöntemin farklı yaklaşımlar eklenerek melez hale getirilmesiyle, farklı metrikler bakımından başarısının artırılması hedeflenmektedir.

KAYNAKLAR

- [1] **Bonabeau E., Dorigo, M., Theraulaz, G.,** 1999. Swarm intelligence: From natural to artificial systems, New York, NY, USA: Oxford University Press, Inc..
- [2] **Kassabalidis I., El-Sharkawi, M. A., II Marks, R. J., Arabshahi, P., Gray, A. A.** 2001. Swarm intelligence for routing in communication networks. Global Telecommunications Conference, GLOBECOM '01 (Vol. 6, pp. 3613–3617). IEEE.
- [3] **Kennedy J., Eberhart, R.** 1995. Particle swarm optimization. In Proceedings of the 1995 IEEE international conference on neural networks (Vol. 4, pp. 1942–1948).
- [4] **Karaboga D.,** 2005. An idea based on honey bee swarm for numerical optimization. Technical report-TR06. Engineering Faculty, Computer Engineering Department, Erciyes University.
- [5] **Passino K. M.** 2002. Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Systems Magazine, 22(3), 52–67.
- [6] **Hossein A., Hossein-Alavi, A.,** 2012. Krill herd: A new bio-inspired optimization algorithm. Communications in Nonlinear Science and Numerical Simulation, 17, 4831–4845.
- [7] **Yang X. S.,** 2010. Engineering optimization: An introduction with metaheuristic applications. John Wiley & Sons.
- [8] **Rajabioun R.,** 2011. Cuckoo optimization algorithm. Applied Soft Computing, 11, 5508–5518.
- [9] **Wang Y., Li, B., Weise, T., Wang, J., Yuan, B., Tian, Q.,** 2011. Self-adaptive learning based particle swarm optimization. Information Sciences, 181(20), 4515–4538.
- [10] **Wang H., Sun, H., Li, C., Rahnamayan, S., Jeng-shyang, P.,** 2013. Diversity enhanced particle swarm optimization with neighborhood. Information sciences, 223, 119–135.
- [11] **Gordon D.,** 2003. The organization of work in social insect colonies. Complexity, 8(1), 43–46.
- [12] **Yip C., Eric, K. S.,** 2008. Cooperative capture of large prey solves scaling challenge faced by spider societies. Proceedings of the National Academy of Sciences of the United States of America, 105(33), 11818–11822.

- [13] **Chen B., Leilei Z., Daniel K., Dongwon L.,** 2010. What is an opinion about? Exploring Political Standpoints Using Opinion Scoring Model. In Proceedings of AAAI Conference on Artificial Intelligence.
- [14] **Asur S., Bernardo A., Huberman,** 2010. Predicting the future with social media. Arxiv preprint arXiv:1003.5699.
- [15] **Joshi M., Dipanjan D., Kevin G., Noah A. S.,** 2010. Movie reviews and revenues: An experiment in text regression. in Proceedings of the North American Chapter of the Association for computational Linguistics Human Language Technologies Conference (NAACL).
- [16] **Sadikov E., Parameswaran A., Petros V.,** 2009. Blogs as predictors of movie success. In Proceedings of the Third International Conference on Weblogs and Social Media (ICWSM).
- [17] **Korap Ö.,** 2014. Çalışanların Twitter Kullanımının, Kurum İmajına Etkisi: Türkiye’de Bir Anket Çalışması Journal Of Yasar University, 2014 9(35) 6099-6260.
- [18] **Nizam H.,** 2016. Sosyal Medyada Makine Öğrenmesi ile Duygu Analizinde Dengeli ve Dengesiz Veri Setlerinin Performanslarının Karşılaştırılması. International Artificial Intelligence and Data Processing Symposium (IDAP'16).
- [19] **Bilge U., Bozkurt S., Oğuz Y. B., Özel D.,** 2011. Sosyal medya araçları Türkiye'deki grip benzeri hastalıkları saptayabilmek için kullanılabilir mi? XVI. Türkiye'de İnternet Konferansı. İzmir.
- [20] **Kıvanç Y.,** 2015. Sosyal Ağlar Üzerinden Deprem Tespiti. XVII. Akademik Bilişim Konferansı Boğaziçi Üniversitesi.
- [21] **Sütcü S., Bayrakçı S.,** 2014. Sosyal Medya Gazeteleri Nasıl Etkiliyor? Haberlerin Twitter’da Yayılması Üzerine Bir Araştırma, The Turkish Online Journal of Design, Art and Communication – TOJDAC April 2014 Volume 4 Issue 2: 40-52.
- [22] **Hölldobler B., Wilson, E. O.,** 1994. Journey to the ants: A story of scientific exploration. ISBN 0-674-48525-4.
- [23] **Hölldobler B., Wilson, E. O.,** 1990. The ants. 0-674-04075-9. Harvard University Press.
- [24] **Lubin T. B.,** 2007. The evolution of sociality in spiders. In H. J. Brockmann (Ed.), Advances in the study of behavior Vol. 37, pp. 83–145.
- [25] **Aviles L.,** 1986. Sex-ratio bias and possible group selection in the social spider *Anelosimus eximius*. The American Naturalist, 128(1), 1–12.

- [26] **Burgess J. W.**, 1982. Social spacing strategies in spiders. In P. N. Rovner (Ed.), Spider communication: Mechanisms and ecological significance (pp. 317–351). Princeton, NJ: Princeton University Press.
- [27] **Maxence S.**, 2010. Social organization of the colonial spider *Leucauge* sp. in the Neotropics: Vertical stratification within colonies. *The Journal of Arachnology*, 38, 446–451.
- [28] **Rayor E. C.**, 2011. Do social spiders cooperate in predator defense and foraging without a web? *Behavioral Ecology & Sociobiology*, 65(10), 1935–1945.
- [29] **Gove R., Hayworth M., Chhetri M., Rueppell O.**, 2009. Division of labour and social insect colony performance in relation to task and mating number under two alternative response threshold models. *Insectes Sociaux*, 56(3), 19–331.
- [30] **Ann L., Rypstra R. S.**, 1991. Prey size, prey perishability and group foraging in a social spider. *Oecologia*, 86(1), 25–30.
- [31] **James J.Q.**, 2013. A Social Spider Algorithm for Global Optimization Technical Report No. TR-2003-004, Dept. of Electrical & Electronic Engineering, The University of Hong Kong, Oct.
- [32] **Pereira L. A. M. D., Rodrigues, P. B. Ribeiro, J. P.**, 2014. Social-Spider Optimization-Based Artificial Neural Networks Training And Its Applications For Parkinson's Disease Identification.
- [33] **Cuevas E., Cienfuegos M., Zaldívar D., Pérez-Cisnero M.**, 2013. A swarm optimization Algorithm inspired in the behavior of the social-spider, *Expert Systems with Applications*, vol. 40, no. 16, pp. 6374–6384.
- [34] **Erik C., Miguel C., Daniel Z., Marco P.C.**, 2013. A swarm optimization algorithm inspired in the behavior of the social-spider.
- [35] **Mirjalili, S.**, 2015. The ant lion optimizer, *Advances in Engineering Software*, 83, pp. 80-98.
- [36] **URL-1**, <http://www.asknature.org/media/image/3305> Ask Nature, 20 Haziran 2017.
- [37] **URL-2**, <http://oldblockwriter.blogspot.com/2012/02/antlions-and-angle-of-rest.html> Old writer in the block. 13 Mayıs 2017.
- [38] **Petrović M.**, 2015. The ant lion optimization algorithm for flexible process planning, *Journal of Production Engineering*.
- [39] **Mirjalili S.**, 2015. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, *Knowledge-Based Systems* 89, page 228–249.

- [40] **Frank K.D., Rich C., Longcore T.**, 2006. Effects of artificial night lighting on moths, in: *Ecological Consequences of Artificial Night Lighting*, pp. 305–344.
- [41] **Zhiming L., Yongquan Z.**, 2016. Lévy-Flight Moth-Flame Algorithm for Function Optimization and Engineering Design Problems *Mathematical Problems in Engineering* Volume 2016, Article ID 1423930, 22 pages.
- [42] **Seyedali M., Andrew L.**, 2016. The Whale Optimization Algorithm. *Advances in Engineering Software*. May; 95:51-67.
- [43] **Watkins WA , Schevill WE.**, 1979. Aerial observation of feeding behavior in four baleen whales: *Eubalaena glacialis* , *Balaenoptera borealis* , *Megaptera novaeangliae*, and *Balaenoptera physalus* . *J Mammal*:155–63.
- [44] **Goldbogen J.A., Friedlaender A.S., Calambokidis J., Mckenna M.F., Simon M., Nowacek D.P.**, 2013. Integrative approaches to the study of baleen whale diving behavior, feeding performance, and foraging ecology. *BioScience*; 63:90–100.
- [45] **Indrajit N. Trivedi**, 2016. A Novel Adaptive Whale Optimization Algorithm for Global Optimization, *Indian Journal of Science and Technology*, Vol 9(38).
- [46] **URL-3**,
http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.htm. *Global Optimization Test Problems*, 25 Haziran 2016.
- [47] **Molga, M., Smutnicki, C.**, 2005. Test functions for optimization needs, Retrieved June 2013, from <http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf>.
- [48] **Adorio, E. P., Diliman, U. P.** 2005. MVF Multivariate Test Functions Library in C for Unconstrained Global Optimization Retrieved June 2013, from <http://http://www.geocities.ws/eadorio/mvf.pdf>.
- [49] **Back, T.** 1996. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press on Demand.
- [50] **John A. H.**, 2008. Online shopping. *Pew Internet & American Life Project Report*.
- [51] **Com S., Kelsey G.**, 2007. Online consumer-generated reviews have significant impact on offline purchase behavior. *Press Release*, November.
<http://www.comscore.com/press/release.asp?press=1928>.
- [52] **Oğuzlar A.**, 2011. *Temel Metin Madenciliği*, Dora Yayınları, Kitap ISBN: 978-605-4118-90-8.
- [53] **Przemysław K.**, 2015. *Applications of Social Media and Social Network Analysis*, ISBN 978-3-319-19003-7 (eBook).

- [54] **Cortes C., Vapnik, V.**, 1995. Support vector networks, Machine Learning, 20,1-25.
- [55] **Lewis D. D.**, 1998. Naive Bayes at forty: The independence assumption in information re-trieval, In Proceedings of the Tenth European Conference on Machine Learning, 4-15.
- [56] **Altman N. S.**, 1992. An introduction to kernel and nearest-neighbor nonparametric regression, The American Statistician. 46 (3): 175–185.
- [57] **Quinlan J. R.**, 1987. Simplifying decision trees. International Journal of Man-Machine Studies. 27 (3): 221.
- [58] **Dipanjan S.**, 2016. Text Analytics with Python A Practical Real-World Approach to Gaining Actionable Insights from your Data, ISBN (electronic): 978-1-4842-2388-8.



ÖZGEÇMİŞ

1992 Elazığ doğumlu V. Cem BAYDOĞAN 2014 yılında Fırat Üniversitesi Yazılım Mühendisliği lisans programından mezun oldu. 2015 yılında Fırat Üniversitesi Fen Bilimleri Enstitüsü Yazılım Mühendisliği Ana Bilim Dalı'nda yüksek lisans eğitimine başladı. 2016 yılında Mersin Üniversitesi Tarsus Teknoloji Fakültesi Yazılım Mühendisliği Bölümü'ne araştırma görevlisi olarak atandı. Yapay zeka ve metasezgisel algoritmalar üzerine akademik çalışmalarını yürütmektedir.

