



**TÜBİTAK-2209-A ÜNİVERSİTE ÖĞRENCİLERİ ARAŞTIRMA  
PROJELERİ DESTEĞİ PROGRAMI**

**ARAŞTIRMA ÖNERİSİ FORMU**

2019

Ekim (Güz) Dönem Başvurusu

**2209/A ÜNİVERSİTE ÖĞRENCİLERİ ARAŞTIRMA PROJELERİ DESTEĞİ PROGRAMI**  
**ARAŞTIRMA ÖNERİSİ FORMU**

**A. GENEL BİLGİLER**

<b>Başvuru Sahibinin Adı Soyadı:</b> Mehmet KATI
<b>Araştırma Önerisinin Başlığı:</b> Meta-Sezgisel Optimizasyon Algoritmalarının DSK Yöntemiyle Geliştirilmesi ve Kısıtlı Mühendislik Tasarım Problemlerinin Optimizasyonu
<b>Danışmanın Adı Soyadı:</b> Hamdi Tolga KAHRAMAN
<b>Araştırmanın Yürütüleceği Kurum/Kuruluş:</b> Karadeniz Teknik Üniversitesi

**ÖZET**

Klasik optimizasyon teknikleri, problem boyutunun, yani tasarım değişkenlerinin sayısının çok olması, problemin lineer olmaması, arama uzayının büyük olması durumlarında kabul edilebilir bir çözüm bulmakta yetersiz ve etkisiz kalmaktadırlar. Karmaşıklık düzeyi yüksek optimizasyon problemleri için en uygun çözümü bulmak zor bir görevdir. Günümüzde karmaşıklık düzeyi yüksek arama uzaylarına sahip optimizasyon problemlerinin çözülmesinde sıklıkla meta-sezgisel arama algoritmaları kullanılmaktadır. Meta sezgisel arama (MSA) algoritmaları doğadan esinlenilerek geliştirilmiş yöntemlerdir [1].

MSA algoritmalarının tatbik edildiği alanlar optimizasyon problemleri ile de sınırlı değildir. MSA algoritmaları başta tahmin, sınıflandırma ve kümeleme problemlerinin modellenmesinin yanı sıra melez yapay zekâ algoritmalarının tasarımında ve geliştirilmesinde de yaygın ve başarılı bir şekilde uygulanmaktadır. Özellikle yapay sinir ağlarının optimizasyonu [2], k-en yakın komşu sınıflandırıcısının [3], karar ağaçlarının [4-5] ve bulanık mantık-temelli algoritmaların [6] melezleştirilmesi ve son dönemlerin popüler araştırma konularından derin öğrenme [7], büyük veri uygulamalarında [8], Endüstri 4.0 [9-10] gibi modern otomasyon sistemlerinin ve uygulamalarının geliştirilmesinde meta-sezgisel optimizasyon tekniklerinden faydalanılmaktadır.

MSA algoritmalarının performansları, komşuluk araması ve çeşitlilik görevlerini yerine getirmelerindeki başarılarına bağlıdır. Özellikle, karmaşıklık düzeyi yüksek problemlerin çözülmesinde MSA algoritmalarının üstesinden gelmeleri gereken iki zorluk bulunmaktadır. Bunların ilki, çok modlu problemlerin (multi-modal) arama uzaylarında çok sayıda yer alan yerel minimum tuzaklarını aşamamak iken, ikincisi ise arama sürecinin sonunda küresel çözüme yeterince yakınsayamamaktır. Yerel çözüm tuzaklarına yakalanmanın başlıca nedeni, algoritmaların çeşitlilik görevlerini etkili bir şekilde yerine getirememeleridir. Yakınsama konusundaki problemler ise algoritmaların komşuluk aramasını hassas bir şekilde gerçekleştirememelerinden kaynaklanmaktadır. Mevcut yöntemlerden daha güçlü arama performansı sergileyen MSA algoritmaları geliştirmek için 1980'li yıllardan bu yana yüzlerce çalışma yapılmıştır. Bu çalışmaların bir kısmı doğadan esinlenilerek geliştirilen yeni MSA algoritmaları iken, büyük bir bölümü ise mevcut MSA algoritmalarının çeşitli yöntemlerle yeniden tasarlanarak (modifiye edilerek) performanslarının iyileştirilmesi esasına dayanmaktadır [11-25]. MSA algoritmalarının yeniden tasarlanmalarında ve melezleştirilmelerinde ise çoğunlukla doğadan esinlenilerek geliştirilmiş çeşitli yöntemler kullanılmaktadır. Bu yöntemleri üç başlık altında toplamak mümkündür. Bunlar, "dağılım", "seçim" ve "kontrol" (DSK) olarak adlandırılan yöntemlerdir.

Günümüzde, üzerinde yoğun bir şekilde çalışılan optimizasyon uygulamalarının başında kısıtlı mühendislik tasarım problemleri gelmektedir. Mevcut MSA algoritmalarının baz modelleri ile bu problemlerin birçoğu çözümlenememekte ya da kabul edilebilir bir çözüme ulaşılamamaktadır. Araştırmacılar bu durumda MSA algoritmalarını kendi problemlerine yönelik olarak özelleştirmek suretiyle modifiye etmekte ve daha iyi çözümleri keşfetmeye çalışmaktadırlar. Algoritmaların probleme özgü olarak tasarlanması ise uzmanlık isteyen ve zorluklarla dolu bir süreçtir. Bu süreçte MSA algoritmaları çok çeşitli tekniklerin tatbik edilebildiği DSK yöntemleri ile esnek ve etkili bir şekilde tasarlanabilmeli, test edilebilmeli ve doğrulanabilmelidirler. Böylesi bir çalışma ise ancak, yazılım tasarım prensiplerine bağlı olarak modüler yapıda geliştirilmiş bir yazılım mimarisini, DSK (dağılım, seçim ve kontrol) yöntemlerini, çeşitli karşılaştırma ve mühendislik test problemlerini, güçlü ve çok sayıda alternatif MSA algoritmalarını içeren ve algoritmaların performanslarını karşılaştırmak için istatistiksel analiz yöntemlerinin tatbik edilebildiği bir platformda gerçekleştirilebilir.

Bu proje çalışmasının amacı, DSK yöntemlerini kullanarak melez ve güçlü MSA algoritmaları geliştirmektir. Böylelikle, günümüzde üzerinde yoğun olarak çalışılan mühendislik tasarım problemlerini mevcut tekniklerden daha başarılı bir şekilde çözümleyen MSA algoritmalarının geliştirilmesi hedeflenmektedir. Proje kapsamında, MSA algoritmalarının performanslarını iyileştirme amacıyla araştırma ve geliştirme faaliyetleri yürütülecektir. Araştırma sürecinde literatürdeki en popüler teknikler (güçlü ve güncel 15 MSA algoritması), çeşitli DSK yöntemleri, 90 adet klasik karşılaştırma problemi, 5 adet mühendislik tasarım problemi (MSA makalelerinde 2-4 arası probleme yer verilmektedir) [26-29], Wilcoxon [30] ve Friedman [31] test ve analiz yöntemleri kullanılacaktır. Bu süreçte üzerinde çalışmalar yürütülecek olan MSA algoritmalarının kaynak kodları MATLAB File Exchange platformundan elde edilmiştir. Çalışma sürecinin ilk adımında, literatürdeki en güncel ve en yaygın kullanılan MSA algoritmaları arasında 15'i (on beşi) seçilecektir. Hâlihazırda bu algoritmaların belirlenmesi için ön çalışma yapılarak 26 MSA algoritmasının makalelerine ve MATLAB kodlarına erişilmiştir [28, 29, 32-55]). Bu 26 algoritma arasında 15'i seçilecektir. Bu algoritmalar arasından ise mühendislik tasarım problemlerinde en iyi performansa sahip olan ilk 3'ü belirlenecektir (algoritmaların mühendislik tasarım problemlerindeki performansları Friedman yöntemiyle analiz edilecek ve sıralanacaktır). İkinci adımda, bu üç (3) algoritmaya çeşitli DSK yöntemleri tatbik edilerek algoritmaların arama performansları iyileştirilmeye çalışılacaktır. DSK yöntemleri ile güçlendirilen algoritmaların

## 2209/A ÜNİVERSİTE ÖĞRENCİLERİ ARAŞTIRMA PROJELERİ DESTEĞİ PROGRAMI ARAŞTIRMA ÖNERİSİ FORMU

mühendislik tasarım problemlerindeki performansları araştırılacaktır. Bu süreçte algoritmaların komşuluk araması ve çeşitlilik görevlerini dengeli ve daha etkili bir şekilde yerine getirebilmeleri için DSK yöntemlerinden faydalanılacaktır. Üçüncü adımda, modifiye edilmiş MSA algoritmalarının klasik test problemleri, CEC 2014 problem havuzu [56], CEC 2017 problem havuzu [57] ve mühendislik tasarım problemleri [58-61] üzerindeki performansları araştırılacaktır. Bu süreçte algoritmaların baz modelleri ile yeniden tasarlanmış modelleri arasında performans karşılaştırmaları yapılacaktır. Dolayısıyla toplamda altı (6) rakip yöntem arasından en başarılı olanı belirlenmiş olacaktır. Son olarak algoritmaların performansları istatistiksel test ve analiz yöntemleri (wilcoxon ve friedman testleri) ile analiz edilerek kısıtlı mühendislik problemleri için en güçlü MSA algoritması literatüre kazandırılacaktır. Geliştirilecek algoritmaların ve yapılacak çalışmaların uluslararası konferanslarda ve saygın akademik dergilerde yayınlanması için gerekli hazırlıklar ve başvurular yapılacaktır.

### 1. ÖZGÜN DEĞER

#### - *Konunun Önemi*

Yapay zekânın bir türü olan MSA algoritması çalışmaları 1970'li yıllara dayanmaktadır. Michigan üniversitesinde Prof. John Holland ve öğrencilerinin geliştirdikleri Genetik Algoritma bu çalışmalara hız kazandırmıştır. Son yıllarda gerek algoritma geliştirme çalışmalarında gerekse de problemlere tatbik edilmeleri hususunda elde edilen başarılar meta-sezgisel algoritmaların önemini giderek artırmaktadır. MSA algoritmaları yapay zekanın uygulandığı her alanda, tahmin, kümeleme, sınıflandırma gibi problemleri çözümlmek için melez algoritmaların geliştirilmesinde ve asıl olarak optimizasyon problemlerinin çözümünde yaygın bir şekilde kullanılmaktadırlar. Maliyetleri azaltmanın ve verimliliği artırmanın kritik önem kazandığı çağımızda süreçleri ve sistemleri optimum şekilde modellemenin etkili yollarından biri olarak meta-sezgisel algoritmalara başvurulmaktadır. Enerji, inşaat, pazarlama, üretim, bilgi teknolojileri, havacılık ve uzay sanayii gibi birçok alanda binlerce sistem ve sürecin optimizasyonunda ve Endüstri 4.0 gibi modern otomasyon sistemlerinin ve uygulamalarının geliştirilmesinde meta-sezgisel optimizasyon tekniklerinden faydalanılmaktadır. Bunun yanında günümüzde mühendislik tasarım problemlerinin optimizasyonu önemli bir problem haline gelmektedir. Birbirinden önemli ölçüde farklı karakterlere sahip bu problemler için zorluk düzeyleri arttıkça küresel en iyi çözümü (global optimum) bulmak imkânsız hale gelmektedir. bu durumda uzun ve çetin bir araştırma süreci başlamaktadır. Bu konuda en bilinen problemlerden biri güç sistemlerinin optimizasyonudur. Her yıl binlerce akademik çalışmanın yapıldığı ve en optimum çözümün arandığı dinamik özellikte çok sayıda mühendislik tasarım problemlerini içeren güç sistemlerinin optimizasyonunda sağlanan küçük bir iyileşme dahi milyarlarca dolarlık tasarrufu sağlayabilmektedir. Tüm bu anlatılanlar optimizasyon algoritmalarında yapılabilecek iyileştirme ve geliştirmelerin ne denli önemli olduğunu açıkça göstermektedir.

#### - *Araştırma Önerisinin Özgün Değeri*

Bu proje çalışmasının özgünlüğü aşağıdaki faaliyetlerle sağlanacaktır:

- i) Meta-sezgisel arama algoritmalarının tasarımında bir ilk gerçekleştirilecektir. Literatürde ilk defa bu proje önerisinde bir araya getirilerek sunulan DSK (dağılım, seçim ve kontrol) yöntemi tatbik edilecektir. Literatürde DSK isimli bir yöntem bulunmamaktadır. Literatürde çeşitli dağılım, seçim ve kontrol yöntemlerinden genellikle birini uygulandığı ve melez MSA algoritmalarının geliştirildiği görülmektedir. Ancak bu üç yöntemin içerdikleri alt tekniklerle birlikte sistematik bir yaklaşımla sınıflandırıldığı ve tatbik edildiği bir çalışma yoktur. Proje çalışmasıyla bu sağlanacaktır. Bu proje çalışmasında DSK yöntemi isimli bir terminolojik ifade de literatüre kazandırılmış olunacaktır.
- ii) DSK yöntemiyle geliştirilecek olan melez algoritmalar arasından mühendislik tasarım problemleri üzerinde en etkili olanı belirlenecek ve literatüre kazandırılacaktır.
- iii) Literatürde melez MSA algoritmalarının geliştirildiği çalışmalarda test ve doğrulamada kullanılan problemlerin ve rakip algoritmaların gerek sayılarında gerekse de niteliklerinde problemlerle karşılaşılmaktadır. Algoritmaların test edilmelerinde ve doğrulanmalarında kısaca deneysel çalışmalarında yetersizlikler ve standartsızlıklar vardır. Algoritmalar sadece daha başarılı oldukları rakip algoritmalarla ve sadece üstünlük kurdıkları problemler üzerinden kıyaslanmaktadırlar. Bu durum MSA algoritmalarının gerçek performanslarının belirlenememesine ve araştırmacıların yanılgılarına yol açmaktadır. Oysa algoritmalar farklı karakterlerde optimizasyon problemlerini içeren problem havuzlarında test edilmelidirler. Bunun yanında doğrulama çalışmalarında kullanılan rakip algoritmaların en güncel ve güçlü algoritmalar arasından seçilmeleri gerekir. MSA algoritmalarının deneysel çalışmalarında ortalama problem sayısı yaklaşık 15-20 arası iken rakip algoritma sayısı ise ortalama 8-12 arasında değişmektedir [2-3]. Bu proje çalışmasında ise literatürdeki en güçlü ve modern 26 MSA algoritması kullanılacaktır. Üstelik 90 adet sürekli değerli test problemi ve 5 adet de ayrık değerli kısıtlı mühendislik tasarım problemi kullanılacaktır. Bu yönüyle de proje çalışması literatürde bir ilk olacaktır.

#### - *Araştırma Sorusu/Hipotezi*

Literatürdeki MSA algoritmaları üzerindeki iyileştirmeler incelendiğinde, çalışmaların birçoğu mevcut MSA tekniklerinin çeşitli yöntemlerle iyileştirilmesini ve varyasyonlarının geliştirilmesini konu almaktadır. Algoritmalarda

## 2209/A ÜNİVERSİTE ÖĞRENCİLERİ ARAŞTIRMA PROJELERİ DESTEĞİ PROGRAMI ARAŞTIRMA ÖNERİSİ FORMU

arama performansları üzerinde etkili olan iki temel öge seçim yöntemleri ve arama operatörleridir ve iyileştirme çalışmaları bu noktalar üzerinde yoğunlaşmaktadır. MSA algoritmalarının performanslarını etkileyebilecek noktalar olan popülasyon yaratma (D: dağılım), çözüm adayı seçimi (S: seçim) ve algoritma parametrelerinin kontrol (K: kontrol) bölümlerini DSK başlığı altında toplayarak iyileştirme yaklaşımlarını daha sistematik hale getirmek mümkündür. MSA algoritmalarına DSK yöntemlerinden biri ya da birkaçı kullanılarak algoritmaların baz modelleri üzerinde iyileşme elde etmek mümkündür.

### 1.2. Amaç ve Hedefler

Optimizasyon, mühendislik tasarımından finansal piyasalara, bilgisayar bilimlerinden endüstriyel uygulamalara kadar çok geniş alanda uygulanmaktadır. Geliştirilen MSA algoritmaları karmaşıklık düzeyi yüksek olan optimizasyon problemlerini yerel çözüm tuzaklarına takılarak kabul edilebilir çözümlere ulaşamamaktadır. Literatürde MSA algoritmalarının performanslarını artırarak yerel çözüm tuzaklarına takılma problemleri, çoğunluğunu doğadan esinlenerek geliştirilen tekniklerin tatbik edilmesiyle çözülmeye çalışılmıştır.

MSA algoritmalarını geliştirmek için kullanılan teknikleri ve MSA algoritmalarının yaşam döngülerinin adımlarını üç başlık altında toplanacaktır. Bu çalışmayla birlikte daha sistematik bir çalışma ortamı hazırlanarak optimizasyon ve kısıtlı mühendislik tasarım problemlerini başarılı bir şekilde çözümleyebilen melez MSA algoritmaları geliştirilecektir.

## 2. YÖNTEM

### 2.1. DSK Yöntemi

MSA algoritmaları doğadaki işleyişlerin taklit edilmesiyle oluşturulmuştur. Bu algoritmaların yetenekleri ve özellikleri farklı olsa da temel olarak algoritma yaşam döngüleri aynı adımlardan oluşmaktadır. Bir problemin MSA algoritma ile çözülmesi algoritma 1'de verilmiştir.

#### Algoritma 1. MSA algoritmalarının arama süreci

- 1) Problemin yaratılması (uygunluk fonksiyonunun, ceza fonksiyonunun tanımlanması)
- 2) Çözüm adayının tasarımı ve çözüm adayları topluluğunun yaratılması
- 3) Adayların uygunluk değerlerinin hesaplanması
- 4) İteratif süreç
  - a. Komşuluk Araması
  - b. Çeşitliliğin Sağlanması
  - c. Çözüm adayı topluluğunun güncellenmesi
- 5) Sonlandırma kriteri sağlandı mı?
  - a. Hayır (Adım 4'e dön)
  - b. Evet (Arama sürecini sonlandır ve en iyi çözüm adayını kaydet)

Arama sürecinde verilen 1, 2, 3 ve 5 numaralı adımlar MSA algoritmaları için ortak adımlardır. 4 numaralı adım ise MSA algoritmalarına özgü operatörlerin ve işlemlerin uygulandığı adımdır. Arama sürecinin başarısı bu adıma bağlıdır.

MSA algoritmalarının yerine getirmesi gereken iki gereksinimi olan komşuluk araması ve çeşitliliği başarılı bir şekilde yerine getirmesi için birçok faktör bulunmaktadır. Bu faktörler dört ana başlık altında toplanabilir. Bunlar sırayla dağılım yöntemleri [62], seçim yöntemleri [63], arama operatörleri [64] ve arama stratejisidir [65-66]. Algoritmalar kendi içlerinde DSK yöntemine göre bölünecektir. Bu bölümler MSA algoritmaları için temel adımlar olan dağılım, seçim ve kontrol bölümlerinden oluşmaktadır.

#### 2.1.1. Dağılım Prosedürü

MSA algoritmaların yaşam döngüsü, algoritmalarının arama sürecinde 2 numaralı adım olan çözüm adayının tasarımı ve çözüm adayları topluluğunun yaratılması ile başlar. Çözüm adayları alternatif dağılım yöntemleri kullanılarak arama uzayında konumlandırılırlar. Bu adım **dağılım** prosedürlerine dahil olmaktadır. Algoritmalarda iyileştirme elde edebilmek için çözüm adaylarının oluşturulmasında tatbik edilecek teknikler Levy Uçuşları, Gauss Dağılımı, Düzgün Dağılım ve Kaos Haritalarıdır. Dağılım prosedüründe yapılacak iyileştirme ile çözüm adaylarının arama uzayında iyice yayılması hedeflenmektedir.

#### 2.1.2. Kontrol Prosedürü

MSA algoritmalarında sezgisel arama sürecine 4 numaralı adıma eşlik eden algoritmaların ait tasarım parametreleri bulunmaktadır. Bu tasarım parametrelerine yerçekimsel arama algoritmasında kullanılan yerçekimi sabiti örnek verilebilir. Bu adım **kontrol** prosedürlerine dahil olmaktadır. Kontrol işleminde amaç MSA algoritmalarına ait tasarım parametrelerinin optimize edilmesidir. Kontrol prosedürlerinde iyileşme elde edebilmek için tatbik edilecek teknikler Levy Uçuşları ve Kaos Haritalarıdır. Kontrol prosedüründe tatbik edilen teknikler ile MSA algoritmalarının keşif ve sömürü arasındaki dengenin, algoritmaların arama yaşam döngülerinde bozulmadan

kabul edilebilir sonuçların bulunması hedeflenmektedir.

### 2.1.3. Seçim Prosedürü

Sezgisel arama sürecinde 4 numaralı adımda bulunur. Arama sürecini yönlendirecek – rehberlik edecek konumlar belirlenir. Bu adım **seçim** prosedürlerine dahil olmaktadır. Komşuluk araması ve çeşitlilik formüllerinde kullanılan çözüm adaylarını kapsamaktadır. Seçim prosedürlerinde iyileşme elde edebilmek için tatbik edilecek teknik Rulet Tekerleğidir. Seçim prosedürlerinde belirlenen hedef kontrol prosedürleri ile aynıdır. MSA algoritmalarının arama sürecine etki edecek konumların belirlenmesinde sürekli aç gözlü yaklaşımın uygulandığı MSA algoritmalarında çözüm adayları, çözüm adayları topluluğu içerisindeki en uygun çözüm bireyine yönelmektedir. Bu yönelim ile algoritmaların rasgele hareketlerinde daralma olmaktadır. İteratif süreç ile birlikte en uygun çözüm bireyine yaklaşan çözüm adayları topluluğu, çeşitlilik ve sömürü arasındaki dengenin, sömürüye doğru kaymasından dolayı çeşitlilik özelliğini kaybederek prematüre yakınsama problemine takılmaktadır. Seçim prosedürleri ile hedeflenen MSA algoritmalarının arama sürecini yönlendirecek konumların belirlenmesinde rasgele hareketliliklerinde daralma oluşumunu önleyerek keşif ve sömürü arasındaki dengenin korunmasını sağlamaktadır.

## 2.2. Tatbik Edilecek Teknikler

### 2.2.1. Levy Uçuşu

Geliştirilen optimizasyon tekniklerinde genel olarak 2 tip randomizasyon kullanılmıştır. Bunlardan birisi klasik rasgele hareket, yani işlemcinin üreteceği rasgele sayıya dayalı randomizasyon, diğeri ise Levy Uçuş mekanizmasıdır. Bu mekanizmada da yine işlemcinin üreteceği rasgele sayı vardır fakat mekanizma bir istatistiksel matematik formülüne dayanmaktadır. Kullanılan her iki yöntem de problemlerin çözümünde önemli iyileştirmeler sağlamıştır [67].

Levy uçuşları, Levy hareketi olarak da bilinir. Gauss olmayan rasgele işlemlerin sabit artışlarla Levy sabit dağılımına göre dağıtıldığı Fransız matematikçi Pierre Lévy tarafından çalışılmış bir sınıfı temsil etmektedir [68]. Akışkanlar dinamiği, deprem analizi, ışınır moleküllerin difüzyonu gibi birçok doğal ve yapay olay Levy uçuşları ile tanımlanabilmektedir [69].

Levy Uçuşları'nda, varsayılan rastgele yürüyüş yaklaşımından farklı olarak, hareketin yapılması aşamasında olasılık dağılımlarından faydalanılması söz konusu olmaktadır. Levy Uçuşları'nda hareket halindeki unsurun atacağı adımın (konum değiştirmenin) boyutu değişkenlik göstermektedir. Bu değişkenlik, konum değişikliği süreci boyunca fraktal ve fraktal olmayan bir akış seyri ortaya koymaktadır [70]. Araştırmalar genel olarak bu davranışın doğadaki birçok canlı tarafından ortaya konulduğunu göstermiştir [71 – 72]. Doğadaki rastgele canlı hareketlerini daha hassas açıklayan bu yaklaşımı, varsayılan rastgele yürüyüş yerine tercih etmişlerdir. Yang ve Deb [73] Guguk Kuşu Arama'da oluşturmak için Levy uçuşu dağılımını kullandı, ayrıca Xin-She Yang [74] Ateş Böceği Algoritmasının yeni bir versiyonu olan Levy uçuşu Ateş Böceği Algoritmasını, bu algoritmanın rasgeleleştirmesini düzeltmek için Levy uçuşu arama stratejisi ile kombine etti. Bu alandaki diğer bir çalışma ise, Heidari ve Pahlavani'nin 2016 yılındaki çalışmalarıdır [75]. Çalışmalarında, sezgisel bir optimizasyon algoritması olan Gri Kurt Optimizasyonu'na Levy Uçuş Mekanizması'nı adapte etmişlerdir ve PSO'daki soruna benzer şekilde, kurtların konum çeşitliliğinin fazla olmamasının lokal minimaya sebep olduğunu öngörmüşlerdir ve bu sorunu Levy Uçuş Mekanizması ile çözmüşlerdir. Seyedali Mirjalili tarafından 2016 yılında geliştirilen Yusufçuk Algoritması'nda Levy Uçuş Mekanizması kullanılmıştır [76].

Levy uçuşu kullanılarak dağılımın nasıl yapıldığını biraz ayrıntılı şekilde inceleyelim. Levy uçuşu ile çözüm adayının yeni konumu;

$$X^{T+1} = X^T + (\alpha \oplus Levy(d)) \oplus X^T \quad 1$$

Denklem 1 kullanılarak hesaplanır.  $\alpha$  parametresi, çözüm adayına Levy uçuşunun uygulandıktan sonra ne kadar sapacağını başka bir deyişle Levy uçuşundan dönecek sayının adım boyutunu kontrol etmektedir.  $\alpha$  parametresi için genellikle kullanılan 0.01 değeri için Levy uçuşunun formülü denklem 6'da verilmiştir. Bu  $\oplus$  sembol çoklu çarpım anlamına gelmektedir.

$$Levy(x) = 0.01 * \frac{R_1 * \delta}{|R_2|^{1/\beta}} \quad 2$$

Denklem 2'de verilen  $R_1$  ve  $R_2$  [0 – 1] aralığında rasgele sayılardır.  $\beta$  parametresi ise Levy uçuşu için önemli noktalardan bir tanesi olup sabit bir değerdir.  $\beta$  parametresinin farklı değerlerde farklı sonuçlar vermektedir. Farklı karakteristikteki test fonksiyonları için ayrı bir  $\beta$  parametresinin kullanılması daha etkili sonuçların verebileceği söylenebilir. Lee and Yao [77] Evrim Algoritmasında Levy uçuşun  $\beta$  parametresinin 4 farklı durumu ile 4 farklı çözüm adayı oluşturmuştur. Oluşturulan 4 çözüm adayı kendi içlerinde kıyaslanarak en iyisi seçilerek mutasyon işlemi gerçekleştirilmiştir.  $\delta$  parametresi denklem 3'de verilmiştir.

$$\delta = \left( \frac{\Gamma(1 + \beta) * \sin(\frac{\pi * \beta}{2})}{\Gamma(\frac{1 + \beta}{2}) * \beta * 2^{(\frac{\beta-1}{2})}} \right)^{1/\beta}$$

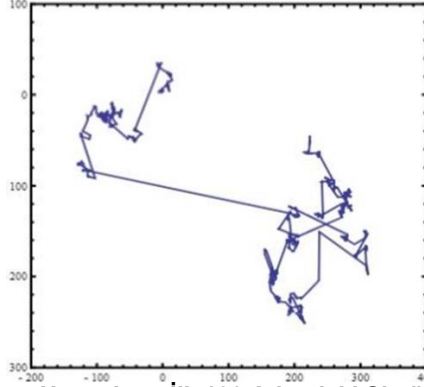
3

Denklem 3'de verilen  $\Gamma$  standart gamma fonksiyonudur. Matematiksel ifadesi denklem 4'de verilmiştir.

$$\Gamma = (X - 1)!$$

4

Şekil 1'de Levy uçuşunun ilk 1000 adımıdaki örneği gösterilmektedir:



**Şekil 1. Levy Uçuşu'nun İlk 100 Adımıdaki Simülasyonu [67]**

### 2.2.2. Kaotik Haritalar

Meta sezgisel optimizasyon algoritmalarının stokastik yapılarından dolayı rastgele sayı dizileri bu alanda sıkça kullanılmaktadır. Rastgele üretilen sayı dizilerinin dağılımları algoritmaların global optimuma yakınsamalarında oldukça etkili olmaktadır. Bu nedenle rastgele sayı üretici olarak kaotik haritalar meta sezgisel optimizasyon algoritmaları ile kullanılabilir [78].

Kaotik haritalar kaotik davranış sergileyen ayrık zamanlı sistemlerdir ve kaotik haritalarla üretilen sayıların tahmin edilemez, yayılmış spektrumlu karakteristiğe sahip olduğu ve periyodik olmadığı teorik olarak kanıtlanmıştır [79]. Temelinde kaotik haritalar olarak adlandırılan fonksiyonlar bulunmaktadır [80].

Kaotik sistemler başlangıç değerlerine aşırı bağımlıdır. Kaotik bir sistem birbirine çok yakın iki farklı başlangıç noktasından başlatılırsa bu küçük farklılık zamanla üstel olarak artar [81]. Ancak kaotik olmayan bir sistemde ise fark zamanla doğrusal olarak artan bir hataya dönüşebilmektedir. Kaotik sistemlerin hesaplama maliyetleri düşüktür. Üretilen sayılar için fazla depolama alanı kullanılmamalı ve istenen bir doğruluğa ulaşmak için fazla zamana gereksinim duyulmamalıdır. Yapılan çalışmalar kaotik sayı dizilerinin üretilmelerinin ve depolanmalarının kolay ve hızlı olduğunu göstermektedir [82].

Kaotik haritalar, algoritmaların performansını hem yerel optimumdan kaçınma hem de yakınsama hızı açısından geliştiren en iyi yöntemlerden biridir [83]. Sezgisel algoritmaların performansını arttırmak için kaotik haritaları kullanan çalışmalar literatürde mevcuttur. Global optimizasyon için Kaotik Arı Kolonisi Algoritmaları [79], Kaotik Haritalı Parçacık Sürü Optimizasyon Algoritmaları [84], Kaotik Haritalı Hibrit Parçacık Sürü Optimizasyon Algoritması [85], Kaotik Ateş Böceği Algoritması [86], Kaotik Gri Kurt Optimizasyon Algoritması [87], Kaotik Big-Bang Optimizasyonu [88], Kaotik Armoni Arama Algoritmaları [89], Kaotik Haritalı Balina Optimizasyon Algoritması [90] bu çalışmalara örnek olarak verilebilir.

Genellikle arama operatörleri üzerinde tatbik edilen Kaos Haritalarının literatürde on (10) farklı versiyonu bulunmaktadır. Bunlar: Chebyshev, Daire, Gauss, İteratif, Lojistik, Parçalı, Sinüs, Singer, Sinüsoidal ve Çadır [41]. Kaos Haritalarının grafikleri ve matematiksel formülleri verilmiştir.

#### 2.2.2.1. Lojistik Harita

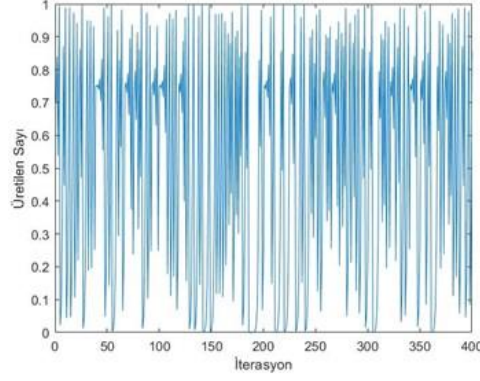
En basit ve en çok kullanılan haritalardan birisidir. Genellikle Lojistik harita ayrık zamanlı dinamik bir sistemdir. Tek boyutludur ve doğrusal değildir. Lojistik haritanın matematiksel ifadesi Denklem 5'de verilmiştir.

$$X_{n+1} = a X_n (1 - X_n)$$

5

Denklemden verilen n iterasyon sayısını,  $X_n$  n. Kaotik sayıyı, a parametresi ise bifürkasyon parametresi olup  $3.57 \leq a \leq 4$  için kaotik davranış göstermektedir. a = 4 ve 0.7 başlangıç noktası ile 400 iterasyon sonunda lojik haritanın ürettiği sayılar Şekil 2'de verilmiştir.





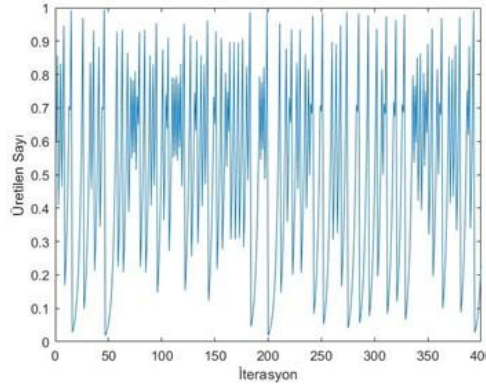
**Şekil 2. Lojik Harita Grafiği**

#### 2.2.2.2. Çadır Harita

Çadır harita Logistic haritaya benzerliği ile bilinmektedir. Çadır haritanın matematiksel ifadesi Denklem 6'da verilmiştir. Başlangıç noktası 0.6 ve 400 iterasyon sonunda lojik haritanın ürettiği sayılar Şekil 3'de verilmiştir.

$$X_{n+1} = \begin{cases} \frac{X_n}{0.7} & X_n < 0.7 \\ \frac{X_n}{p} (1 - X_n) & 0.7 \leq X_n \end{cases}$$

6



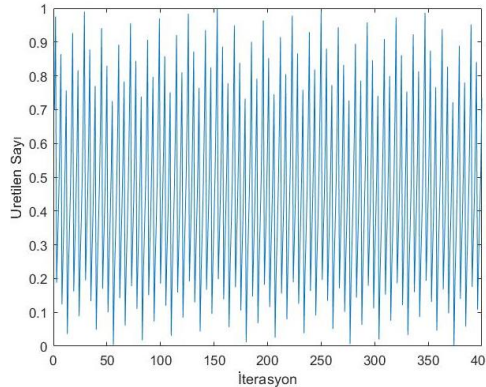
**Şekil 3. Çadır Harita Grafiği**

#### 2.2.2.3. Çember Harita

İlk olarak Andrey Kolmogorov tarafından geliştirilen Çember harita denklemi aynı zamanda elektronikteki faz kilitlemeli döngü denklemini de ifade etmektedir [91]. Çember haritanın matematiksel ifadesi Denklem 7'de verilmiştir. Kontrol parametreleri  $a = 0.5$ ,  $b = 0.2$ ,  $X_0 = 0.7$  ve 400 iterasyon sonunda lojik haritanın ürettiği sayılar Şekil 4'de verilmiştir.

$$X_{n+1} = X_n + b - \left(\frac{a}{2\pi}\right) \sin(2\pi X_n) \bmod(1)$$

7



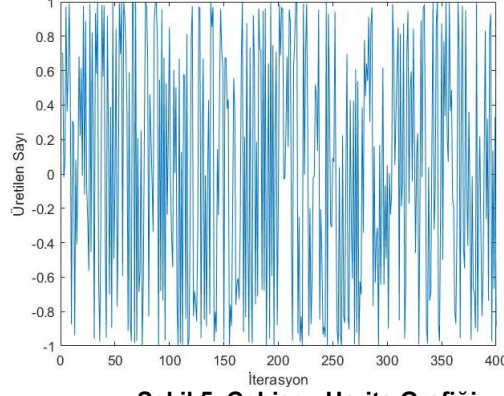
**Şekil 4. Çember Harita Grafiği**

#### 2.2.2.4. Çebişev Haritası

Matematikte Pafnuty Chebysev'in adını taşıyan çebişev polinomları moivre formülü ile ilişkili ve iteratif şekilde tanımlanabilen ortogonal polinomlar dizisidir [92]. Çerbişev haritanın matematiksel ifadesi Denklem 8'de verilmiştir. Başlangıç noktası 0.7 ve 400 iterasyon sonunda lojik haritanın ürettiği sayılar Şekil 5'de verilmiştir.

$$X_{n+1} = \cos(n * \cos^{-1}(X_n))$$

8



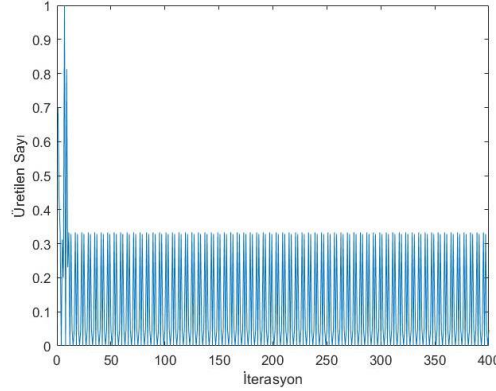
**Şekil 5. Çebişev Harita Grafiği**

#### 2.2.2.5. Gaus Harita

Gaus haritanın matematiksel ifadesi Denklem 9'da verilmiştir. Başlangıç noktası 0.7 ve 400 iterasyon sonunda lojik haritanın ürettiği sayılar Şekil 6'da verilmiştir.

$$X_{n+1} = \begin{cases} 0 & X_n = 0 \\ \frac{1}{X_n} \bmod(1) & X_n \in (0,1) \end{cases}$$

9



**Şekil 6. Gaus Harita Grafiği**

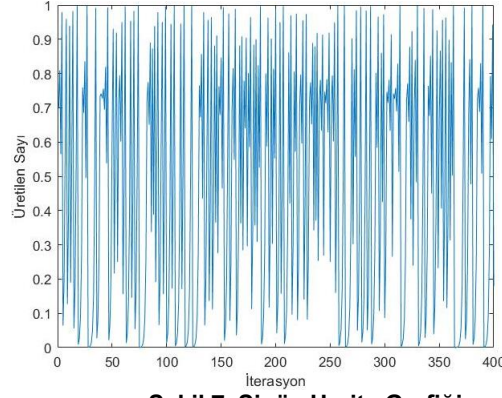
#### 2.2.2.6. Sinüs Harita

Sinüs haritanın matematiksel ifadesi Denklem 10'da verilmiştir. Başlangıç noktası 0.7 ve 400 iterasyon sonunda lojik haritanın ürettiği sayılar Şekil 7'de verilmiştir.

$$X_{n+1} = \left\{ \frac{a}{4} \sin(\pi X_n) \right\} \quad 0 < a \leq 4$$

10





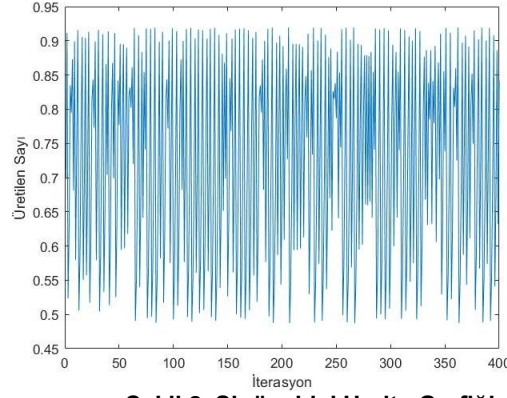
Şekil 7. Sinüs Harita Grafiği

#### 2.2.2.7. Sinüsoidal Harita

Sinüsoidal haritanın matematiksel ifadesi Denklem 11'de verilmiştir. Başlangıç noktası 0.7,  $a=2.3$  ve 400 iterasyon sonunda lojik haritanın ürettiği sayılar Şekil 8'de verilmiştir.

$$X_{n+1} = aX_n^2 \sin(\pi X_n)$$

11



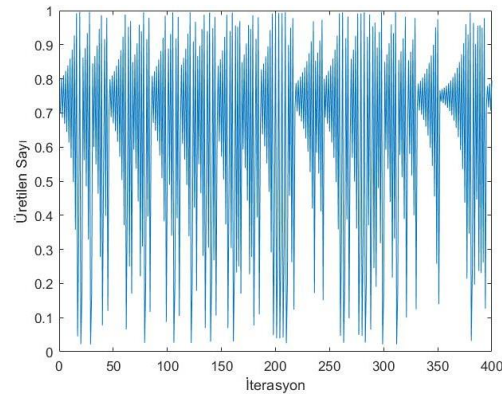
Şekil 8. Sinüsoidal Harita Grafiği

#### 2.2.2.8. Singer Harita

Singer haritanın matematiksel ifadesi Denklem 12'de verilmiştir. Başlangıç noktası 0.7,  $\mu=1.07$  ve 400 iterasyon sonunda lojik haritanın ürettiği sayılar Şekil 9'de verilmiştir.

$$X_{n+1} = \mu(7.86X_n - 23.31X_n^2 + 28.75X_n^3 - 13.302875X_n^4)$$

12



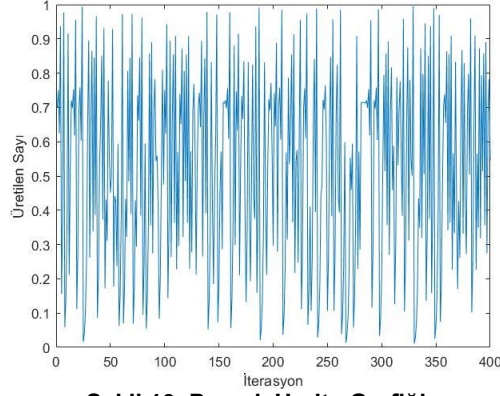
Şekil 9. Singer Harita Grafiği

#### 2.2.2.9. Parçalı Harita

Parçalı haritanın matematiksel ifadesi Denklem 13'de verilmiştir. Başlangıç noktası 0.7,  $P=0.4$  ve 400

iterasyon sonunda lojik haritanın ürettiği sayılar Şekil 10'da verilmiştir.

$$X_{n+1} = \begin{cases} \frac{X_n}{P} & 0 \leq X_n < P \\ \frac{X_n - P}{0.5 - P} & P \leq X_n < 0.5 \\ \frac{1 - P - X_n}{0.5 - P} & 0.5 \leq X_n < 1 - P \\ \frac{1 - X_n}{P} & 1 - P \leq X_n < 1 \end{cases} \quad 13$$

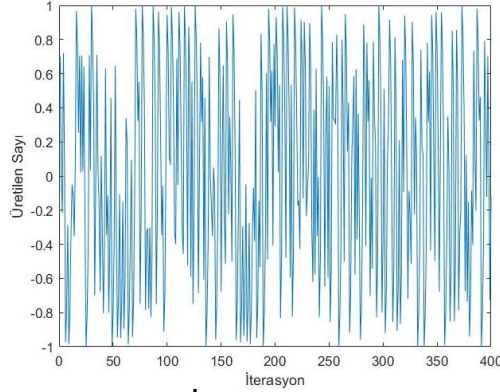


**Şekil 10. Parçalı Harita Grafiği**

#### 2.2.2.10. İteratif Harita

İteratif haritanın matematiksel ifadesi Denklem 14'de verilmiştir. Başlangıç noktası 0.7, a=0.7 ve 400 iterasyon sonunda lojik haritanın ürettiği sayılar Şekil 11'de verilmiştir.

$$X_{n+1} = \sin\left(\frac{a * \pi}{X_n}\right) \quad 14$$



**Şekil 11. İteratif Harita Grafiği**

#### 2.2.3. Rulet Tekerleği

Rulet Tekerleği olasılıksal seçim yöntemidir. Topluluk içerisindeki çözüm adaylarının uygunluk değerlerine bağlı olarak seçilme olasılıkları hesaplanır. Seçim işlemi bu olasılıklara bağlı olarak tek adımda gerçekleşir. Rulet Tekerleğinin sözde kodu algoritma 2'de verilmiştir.

**Algoritma 2. Rulet seçim yöntemi sözde kodu**

```

n = MSA algoritmasında çözüm adayı sayısı
m = Optimizasyon probleminin boyutu
X[n,m] = Çözüm adayları topluluğu,
U[n] = Çözüm adaylarının uygunluk değerleri
R[n] = Çözüm adaylarının rulet tekerleği yüzdeleri
K[n] = Çözüm adaylarının rulet tekerleği konumları,
t = 0, K[0] = 0;
  for i=1:n
    t=t+ U[n]
  end
  for i=1:n
    R[i]=U[i]/ U[n]
    K[i]=R[i] + K[i-1]
  end
  konum=rand (0,1) // rulet tekerleğini döndür ve tekerleğin durduğu konumu belirle
  for i=1:n
    if (K[i-1]<konum<= K[i])
      Seçilen çözüm adayı=X[i]
    end
  end

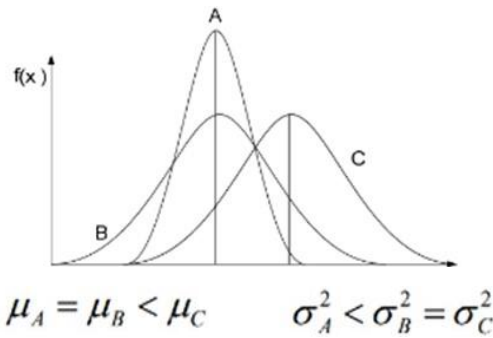
```

#### 2.2.4. Gauss Dağılımı

Rulet Tekerleği olasılıksal seçim yöntemidir. Topluluk içerisindeki çözüm adaylarının uygunluk değerlerine bağlı olarak seçilme olasılıkları hesaplanır. Seçim işlemi bu olasılıklara bağlı olarak tek adımda gerçekleşir. Rulet Tekerleğinin sözde kodu algoritma 2’de verilmiştir. Gauss istatistikteki en önemli ve en çok kullanılan dağılımlardan birisidir. De Moivre tarafından 1733’de bulunan bu dağılım 1800’lü yılların başlarında Fransız Pierre Simon LAPLACE ve Alman Carl Friedrich GAUSS tarafından geliştirilmiştir. Bu nedenle bu dağılıma literatürde normal dağılımın yanı sıra “Laplace-Gauss Dağılımı” ya da “Gauss Dağılımı” da denmektedir [93].

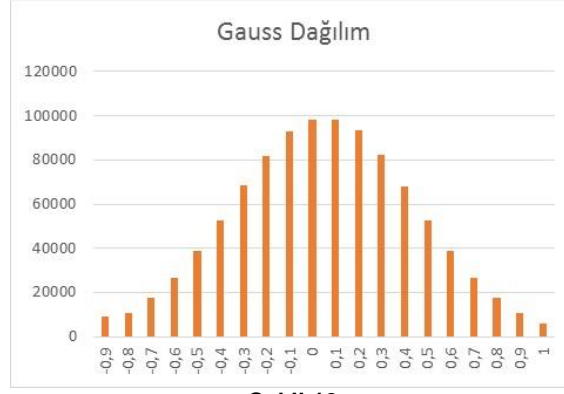
Sürekli dağılım türlerinden biridir ve pratikte birçok durumda verilerin normal dağılım gösteren bir ana kütleden geldiği varsayılır. Günlük yaşamda karşılaşılan pek çok sürekli rassal değişken normal dağılır. Gauss iki parametreye sahiptir. Bu parametreler konum (değer) bilgisini temsil eden  $\mu$  (aritmetik ortalama) ve varyans bilgisini temsil eden  $\sigma^2$  yayılımıdır [94-95]. Aritmetik ortalama parametresi çan eğrisinin tepe noktasını belirler. Normal dağılımın matematiksel ifadesi Denklem 15’de ve  $\mu$  ve  $\sigma^2$  parametrelerinin dağılım üzerindeki etkisi Denklem 16’da verilmiştir.

$$X_{[i]} = \mu + \sigma^2 * \left[ \sqrt{-2 * \ln(\text{rand})} * \sin(2 * \pi * \text{rand}) \right] \quad 15$$



16

Gauss dağılımında değerler [-1, 1] aralığında üretilmektedir. Değerleri bu aralıkta üretmek için mean=0, varyans=0,4 olarak belirlenmiştir. Elde edilen değerlerin yoğunluk grafiği Şekil 12’de gösterilmektedir [62].

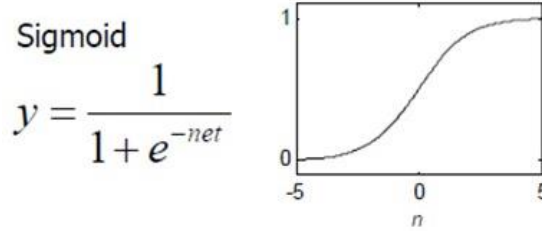


Şekil 12

### 2.3. Tekniklerin Tatbik Edilme Stratejisi

Levy Uçuşu, Kaos Haritaları ve Rulet Tekerleğinin MSA algoritmalarında tatbik ederken hangi oranda ve nasıl uygulanacağını bulması gerekmektedir. Bunun için deney çalışması yapılacaktır. Bu deney çalışmaları:

1. Tatbik edilecek yöntemi belli oranlarda uygulayarak algoritmanın değişime verdiği tepkiyi test etmektir. Bu amaçla tatbik edilen yöntemleri %0,1-100 arasında problemin boyutuna göre dinamik değişen veya sabit oran ile algoritma içerisinde uygulanması.
2. Tatbik edilecek yöntemi, algoritmanın yaşam döngüsü içerisinde azalan oranda veya artan oranda uygulamaktır. Bu oranı belirlemede kullanılacak yöntemler doğrusal veya doğrusal olmayan fonksiyonlardır. Bu amaçla kullanılacak ilk fonksiyon Şekil 13' de verilen Sigmoid 'ır.

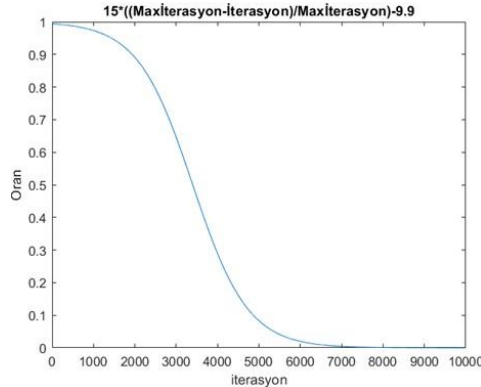


Şekil 13. Sigmoid Fonksiyonu

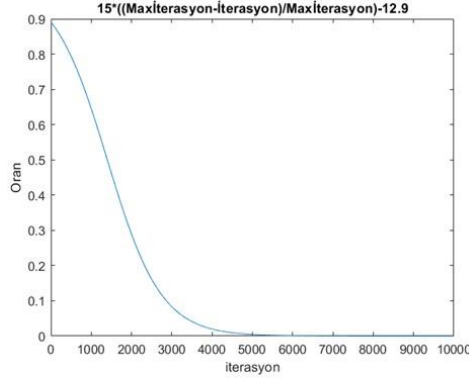
Sigmoid fonksiyonu yapay sinir ağlarında aktivasyon fonksiyonu olarak kullanılmaktadır. Verilen değişkeni farklı bir boyuta taşıyan doğrusal olmayan bir fonksiyondur. Sigmoid fonksiyonunda net değerin hesaplanması için önerilen formüller:

- Artan oranda Levy Uçuşu, Kaos Haritaları ve Rulet Tekerleğinin tatbik edilmesi için  $15 * (\text{İterasyon} / \text{Maxİterasyon}) - A$
- Azalan oranda Levy Uçuşu, Kaos Haritaları ve Rulet Tekerleğinin tatbik edilmesi için  $15 * ((\text{Maxİterasyon} - \text{İterasyon}) / \text{Maxİterasyon}) - A$

Önerilen formüllerde normalizasyon işlemi kullanılarak iterasyon değeri 0-1 arasında ölçeklenmiştir. A katsayısı ile oranlardaki değişim kontrol edilmektedir. Bu değişim miktarları azalan oran için Şekil 14 ve 15 de görülmektedir.



Şekil 14. Sigmoid Fonksiyon Grafiği



**Şekil 15. Sigmoid Fonksiyon Grafiği**

#### **2.4. Deneysel Çalışmalar**

Literatürde sürekli değer optimizasyon problemleri tiplerine göre genellikle dört kategoriye ayrılmaktadır. Bunlar tek modlu, çok modlu, melez ve kompozisyon (compostion) problemlerdir. Bu problem tipleri, 1994 yılından buyana her yıl düzenli bir şekilde gerçekleştirilen IEEE evrimsel hesaplama (IEEE CEC) konferanslarındaki çalışmaların da katkılarıyla ortaya çıkmıştır [56]. MSA algoritmalarını test etmek ve rakip algoritmalarla karşılaştırarak arama performanslarını doğrulamak amacıyla bu dört problem tipinden test fonksiyonlarını içeren karşılaştırma problemleri havuzu oluşturulmaktadır. Buna göre tek modlu problemlerde yerel çözüm tuzakları bulunmamaktadır. Bu problemler algoritmaların yakınsama hızlarının test edilmesi amacıyla kullanılmaktadırlar [96]. Komşuluk arama yeteneği yüksek algoritmalar bu problem türünde başarılı olmaktadır. Çok modlu problemler ise yerel çözüm tuzakları barındıran problem türüdür. Örneğin Michalewicz test fonksiyonunda yerel minimum sayısı, problem boyutuna (n) bağlı olarak faktöriyel (n!) ifadesiyle değişmektedir. Yerel çözüm tuzakları çok modlu problemlerin optimizasyonu zorlaştırmaktadır [97]. Çok modlu problemlerin arama uzaylarındaki tuzaklardan kurtulmak algoritmaların çeşitlilik sağlama yeteneğine bağlıdır. MSA algoritmalarının çeşitlilik işlevlerini test etmek amacıyla çok modlu test fonksiyonları kullanılmaktadır. Melez ve derleme problem türleri ise algoritmaların hem komşuluk araması hem de çeşitlilik yeteneklerini dengeli bir şekilde yönetmelerini gerektirmektedir. Dolayısıyla bu iki problem türündeki test fonksiyonları da MSA algoritmalarının yakınsama hızı ve çeşitlilik dengesini ölçmek amacıyla kullanılmaktadırlar.

MSA algoritmalarında geliştirme çalışmaları Matlab R2018a programında yapılacaktır. Geliştirme çalışmalarında kullanılacak test ve doğrulama problemleri 30'u Klasik Benchmark, 30'u CEC 2014 [56] ve 30'u CEC 2017 [57] olmak üzere toplam 90 test problemi üzerinde çalışılacaktır. CEC2017 ve CEC2014 'de sürekli değerli ve dinamik yapıli test problemleri bulunmaktadır. Dolayısıyla problemlere ait tasarım parametrelerinin arama uzayındaki optimum konumlarının da kaydırma ve döndürme işlevleri yoluyla dinamik olarak değiştirilebildiği ve bu yolla MSA algoritmalarının çeşitli yollarla optimum noktaları yakalayacak avantajlar yaratmasının önüne geçilmeye çalışılacaktır. CEC konferanslarındaki test problemlerinin tamamı dinamik olarak boyutlandırılabilen problemlerle oluşturulmuştur. Böylelikle küçük boyutlu arama uzaylarında hızlı yakınsama özellikleri sayesinde başarılı görülen algoritmaların aynı problemlerin orta ve büyük boyutlu arama uzaylarındaki performanslarını da ortaya çıkarmak amaçlanmıştır.

Deneysel çalışmanın son aşamasında, modifiye edilmiş MSA algoritmalarının CEC havuzundaki ve mühendislik tasarım problemleri üzerindeki performansları araştırılacaktır.

##### **2.4.1. Ayarlar**

Deneysel çalışmada arama uzayı 30, 50 ve 100 boyutlu olarak tasarlanacaktır. Böylelikle algoritmaların küçük, orta ve büyük arama uzaylarındaki performansları test edilmiş olunacaktır. Arama süreci sonlandırma kriteri olarak uygunluk fonksiyonunu (fitness function) çağırma sayısı dikkate alınacaktır. Algoritmalar arasında böylelikle fırsat eşitliği ve adil bir karşılaştırma ortamı sağlanacaktır. Uygunluk fonksiyonunu azami çağırma sayısı 10.000\*d, yani problem boyutunun 10 bin katı olarak tatbik edilecektir. Tüm bu ayarların belirlenmesinde CEC konferanslarındaki standartlar referans alınmıştır [56-57].

##### **2.4.1.1. Test ve Karşılaştırma Problemleri**

Geliştirilecek olan MHS algoritmalarını literatürdeki güçlü ve güncel MHS teknikleri ile karşılaştırmak ve arama performansını doğrulamak için 90 adet kıyaslama problemi ve 5 adet mühendislik tasarım problemi kullanılacaktır. Bunlar; 30 adet klasik kıyaslama problemi, CEC 2014 (30) [56] ve CEC 2017 (30) [57] test problemi havuzlarıdır. Test havuzlarında 4 tipte problem bulunmaktadır. Bunlar tek modlu, çoklu modlu, melez ve kompozisyon tipleridir. Tek modlu problemler algoritmaların yakınsama performansını, çok modlu problemler algoritmaların çeşitlilik sağlama performansını, melez fonksiyonlar her iki yeteneğini (yakınsama ve tuzaklardan kurtulma) ve kompozisyon fonksiyonlar ise algoritmaların dengeli arama yeteneklerini ölçmek amacıyla geliştirilmişlerdir. Devam eden alt bölümlerde problemler hakkında bilgi verilmektedir.

#### 2.4.1.1.1. Klasik Test Problemleri

Algoritmaların arama performanslarını belirlemek üzere literatürde en sık kullanılan problemlerden oluşan bir test seti kullanılacaktır. Bu sette 30 tane sınırsız optimizasyon problemi vardır. Bu kıyaslama problemlerinin 17 tanesi çok modlu, diğerleri ise tek modludur. Problemler Şekil 16'da verilmiştir.

Name	Function	Type	Range	Min
Ackley	$f_1(x) = 20 - 20 \exp(-0.2 \sqrt{\frac{\sum_{i=1}^D x_i^2}{D}}) - \exp(\frac{\sum_{i=1}^D \cos(2\pi x_i)}{D}) + e$	M	[-100, 100]	0
Alpine	$f_2(x) = \sum_{i=1}^D  x_i \sin(x_i) + 0.1 x_i $	M	[-100, 100]	0
Cigar	$f_3(x) = x_1^2 + 10^6 \sum_{i=2}^D x_i^2$	U	[-100, 100]	0
DixonPrice	$f_4(x) = (x_1 - 1)^2 + \sum_{i=2}^D i(2x_i^2 - x_{i-1})^2$	U	[-10, 10]	0
Elliptic	$f_5(x) = \sum_{i=1}^D ((10^6)^{\frac{i-1}{D-1}} x_i^2)$	U	[-100, 100]	0
Exponential	$f_6(x) = \exp(0.5 \sum_{i=1}^D x_i^2)$	M	[-10, 10]	0
Griewank	$f_7(x) = 1 + \frac{\sum_{i=1}^D x_i^2}{4000} - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}})$	M	[-600, 600]	0
I.C.M. <sup>1</sup>	$f_8(x) = \frac{D}{10} - (\frac{\sum_{i=1}^D \cos(3\pi x_i)}{10} - \sum_{i=1}^D x_i^2)$	M	[-100, 100]	0
Levy	$f_9(x) = \sin^2(3\pi x_1) +  x_D - 1  * (1 + \sin^2(3\pi x_D)) + \sum_{i=1}^{D-1} [(x_i - 1)^2 (1 + \sin^2(3\pi x_{i+1}))]$	M	[-10, 10]	0
Michalewicz	$f_{10}(x) = D - \sum_{i=1}^D \sin(x_i) \sin^{20}(\frac{\pi x_i^2}{D})$	M	[0, $\pi$ ]	0
Penalized-1	$f_{11}(x) = \frac{\pi}{D} [10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} [(y_i - 1)^2 (1 + 10 \sin^2(\pi y_{i+1}))] + (y_D - 1)^2] + \sum_{i=1}^D u(x_i, 10, 100, 4)$ $y_i(x_i) = 1 + \frac{x_i + 1}{4}$ $u_i(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > 0 \\ 0 & a \leq x_i \leq -a \\ k(-x_i - a)^m & -a > x_i \end{cases}$	M	[-50, 50]	0
Penalized-2	$f_{12}(x) = 0.1   \sin^2(3\pi x_1) + \sum_{i=1}^D [(x_i - 1)^2 (1 + \sin^2(3\pi x_{i+1}))] + (x_D - 1)^2 (1 + \sin^2(2\pi x_D))   + \sum_{i=1}^D u(x_i, 5, 100, 4)$ $u_i(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > 0 \\ 0 & a \leq x_i \leq -a \\ k(-x_i - a)^m & -a > x_i \end{cases}$	M	[-50, 50]	0
Powell	$f_{13}(x) = \sum_{i=1}^{D/4} [(x_{4i-3} - 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^2]$	U	[-4, 5]	0
Rastrigin	$f_{14}(x) = 10D + \sum_{i=1}^D x_i^2 - 10 \cos(2\pi x_i)$	M	[-100, 100]	0
Rosenbrock	$f_{15}(x) = \sum_{i=1}^{D-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	M	[-10, 10]	0
R.H.E. <sup>2</sup>	$f_{16}(x) = \sum_{i=1}^D (\sum_{j=1}^i x_j)^2$	U	[-100, 100]	0
Salomon	$f_{17}(x) = 1 - \cos(2\pi \sqrt{\sum_{i=1}^D x_i^2}) + 0.1 \sqrt{\sum_{i=1}^D x_i^2}$	M	[-100, 100]	0
Schaffer	$f_{18}(x) = 0.5 + \frac{\sin^2(\frac{\sum_{i=1}^D x_i^2}{4000}) - 0.5}{(1 + 0.001 \sum_{i=1}^D x_i^2)^2}$	M	[-100, 100]	0
Schwefel	$f_{19}(x) = 418.982887272434 + D - \sum_{i=1}^D x_i \sin(\sqrt{ x_i })$	U	[-500, 500]	0
Schwefel 1.20	$f_{20}(x) = \sum_{i=1}^D  x_i $	U	[-100, 100]	0
Schwefel 2.21	$f_{21}(x) = \max_{i=1, \dots, D}  x_i $	U	[-100, 100]	0
Schwefel 2.22	$f_{22}(x) = \sum_{i=1}^D  x_i  + \prod_{i=1}^D  x_i $	U	[-10, 10]	0
Sphere	$f_{23}(x) = \sum_{i=1}^D x_i^2$	U	[-100, 100]	0
Step	$f_{24}(x) = \sum_{i=1}^D (x_i + 0.5)^2$	U	[-100, 100]	0
Styblinski-Tang	$f_{25}(x) = 39.1661657037714 + D + 0.5 \sum_{i=1}^D x_i^4 - 16x_i^2 + 5x_i$	M	[-5, 5]	0
SumPower	$f_{26}(x) = \sum_{i=1}^D  x_i ^{i+1}$	M	[-10, 10]	0
SumSquares	$f_{27}(x) = \sum_{i=1}^D i x_i^2$	U	[-10, 10]	0
Quartic	$f_{28}(x) = \sum_{i=1}^D i x_i^4$	U	[-10, 10]	0
Weierstrass	$f_{29}(x) = \sum_{i=1}^D [\sum_{j=0}^k 0.5^j \cos(2\pi 3^j(x_i + 0.5))] - D \sum_{j=0}^k 0.5^j \cos(\pi 3^j)$ $k = \begin{cases} 20 & D \geq 20 \\ D & D < 20 \end{cases}$	M	[-1, 1]	0
Zakharov	$f_{30}(x) = \sum_{i=1}^D x_i^2 + (\sum_{i=1}^D 0.5 i x_i)^2 + (\sum_{i=1}^D 0.5 i x_i)^4$	M	[-5, 10]	0

<sup>1</sup> Inverted Cosine Mixture

<sup>2</sup> Rotated Hyper Ellipsoid

**Şekil 16. Klasik Test Problemleri**

#### 2.4.1.1.2. CEC 2014 Test ve Karşılaştırma Problem Havuzu

CEC 2014 test paketinde 3 adet tek modlu (f1-f3), 13 adet basit çoklu mod (f4-f16), 6 adet hibrit (f17-f22) ve 8 adet kompozisyon (f23-f30) tipi fonksiyon bulunmaktadır [56].



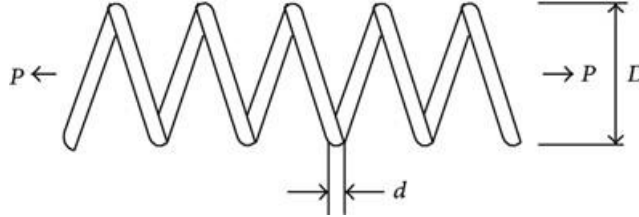
#### 2.4.1.1.3. CEC 2017 Test ve Karşılaştırma Problem Havuzu

CEC 2017 test takımında 3 tek modal (f1-f3), 7 basit çoklu modal (f4-f10), 10 hibrit (f11-f20) ve 10 kompozisyon (f20-f30) tipi fonksiyon bulunur. [57]:

#### 2.4.1.1.4. Mühendislik Tasarım Problemleri

##### 2.4.1.1.4.1. Gerginlik / Sıkıştırma Yayı Tasarımı

Bu sorunun asıl amacı yayın ağırlığını en aza indirmektir. Sorunun tanımında üç parametre vardır. Bunlar tel çapı (d), ortalama bobin çapı (D) ve aktif bobinlerin sayısı (P). Optimizasyon sürecinde, problem sınırlamaları dalgalanma frekansı, minimum sapma ve kayma gerilmesi dikkate alınmaktadır. Şekil 17 'de Gerginlik / Sıkıştırma Yayı Tasarım Sorunu verilmiştir.



**Şekil 17. Gerginlik / Sıkıştırma Yayı Tasarım Problemi**

#### I. Parametreler

$$\vec{x} = [x_1 x_2 x_3] = [dDP],$$

#### II. Formül

$$\text{Minimize } f(\vec{x}) = (x_3 + 2)x_2x_1^2,$$

#### III. Kısıtlar

$$g_1(\vec{x}) = 1 - \frac{x_2^2 x_3}{71785 x_1^4} \leq 0$$

$$g_2(\vec{x}) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} \leq 0$$

$$g_3(\vec{x}) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0$$

$$g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0$$

#### IV. Parametre Çözüm Aralığı

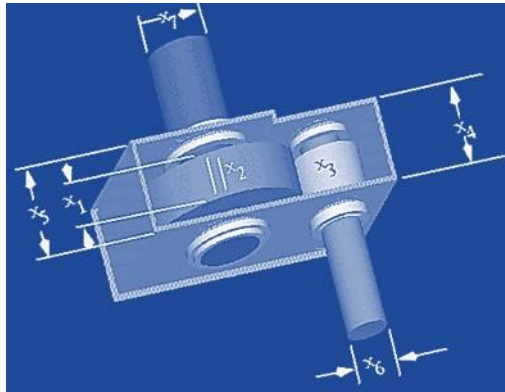
$$0.05 \leq x_1 \leq 2.0$$

$$0.25 \leq x_2 \leq 1.3$$

$$2.0 \leq x_3 \leq 15.0$$

##### 2.4.1.1.4.2. Hız Düşürücü Tasarımı

Hız düşürücü tasarımı, minimize etme sorunudur. Bu sorunun amacı, asgari hız düşürücü ağırlığını bulmaktır. Yeni tasarım parametresine sahiptir. Bunlar yüz genişliği (x1), diş modülü (x2), piyondaki diş sayısı (x3), yataklar arasındaki ilk milin uzunluğu (x4), yataklar arasındaki ikinci milin uzunluğu (x5) ve iki şaftın çapları (x6, x7). Şekil 18' de Hız Düşürücü Tasarım Problemi [76] verilmiştir.



**Şekil 18. Hız Düşürücü Tasarım Problemi**

Şekil 2.1, sorunun şematik gösterimini göstermektedir. Değişkenlerin aralıkları, amaç fonksiyonun matematiksel formülasyonu ve problemin kısıtlamaları aşağıdaki şekilde tanımlanır:

I. Formül

- Minimize  $f(\vec{x}) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2)$ ,

II. Kısıtlar

$$g_1(\vec{x}) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0$$

$$g_2(\vec{x}) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0$$

$$g_3(\vec{x}) = \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \leq 0$$

$$g_4(\vec{x}) = \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \leq 0$$

$$g_5(\vec{x}) = \frac{\sqrt{[(745(x_4/x_2x_3))^2 + 16.9 \cdot 10^6]}}{110x_6^3} - 1 \leq 0$$

$$g_6(\vec{x}) = \frac{\sqrt{[(745(x_5/x_2x_3))^2 + 157.5 \cdot 10^6]}}{85x_7^3} - 1 \leq 0$$

$$g_7(\vec{x}) = \frac{x_2x_3}{40} - 1 \leq 0$$

$$g_8(\vec{x}) = \frac{5x_2}{x_1} - 1 \leq 8$$

$$g_9(\vec{x}) = \frac{x_1}{12x_2} - 1 \leq 0$$

$$g_{10}(\vec{x}) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0$$

$$g_{11}(\vec{x}) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0$$

III. Parametre Çözüm Aralığı

$$2.6 \leq x_1 \leq 3.6$$

$$0.7 \leq x_2 \leq 0.8$$

$$17 \leq x_3 \leq 28$$

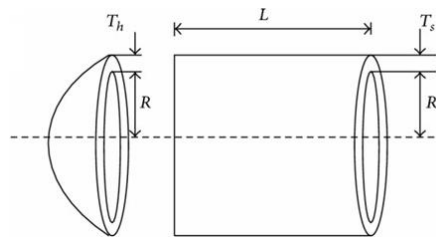
$$7.3 \leq x_4, x_5 \leq 8.3$$

$$2.9 \leq x_6 \leq 3.9$$

$$5.0 \leq x_7 \leq 5.5$$

**2.4.1.1.4.3. Basıncılı Kap Tasarımı**

Basıncılı kap problemi malzemelerin maliyetini, biçimlendirilmesini ve kaynağını içeren yapısal bir mühendislik optimizasyon problemidir. Dört adet tasarım değişkenine sahiptir. Bunlar basınç kabının kalınlığı ( $T_s$ ), kafanın kalınlığı ( $T_h$ ), kabın iç yarıçapı ( $R$ ) ve başsız kabın ( $L$ ) uzunluğu. Şekil 19'da Basıncılı Kap Tasarım Problemi verilmiştir.



**Şekil 19. Basıncılı Kap Tasarım Problemi**

Şekil 2.1, sorunun şematik gösterimini göstermektedir. Değişkenlerin aralıkları, amaç fonksiyonun matematiksel formülasyonu ve problemin kısıtlamaları aşağıdaki şekilde tanımlanır:

I. Parametreler

- $\vec{x} = [x_1x_2x_3x_4] = [T_sT_hRL]$ ,

II. Formül

- Minimize  $f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$

**III. Kısıtlar**

$$g_1(\vec{x}) = 0.0193x_3 - x_1 \leq 0$$

$$g_2(\vec{x}) = 0.00954x_3 - x_2 \leq 0$$

$$g_3(\vec{x}) = 1296000 - \pi x_3^2 x_4 - \frac{4}{3} \pi x_3^3 \leq 0$$

$$g_4(\vec{x}) = x_4 - 240 \leq 0$$

**IV. Parametre Çözüm Aralığı**

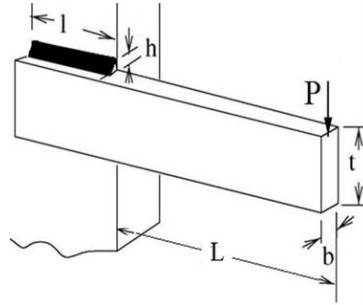
$$0.0625 \leq x_1$$

$$x_2 \leq 99 * 0.0625$$

$$10.0 \leq x_3 x_4 \leq 200.0$$

**2.4.1.1.4.4. Kaynaklı Kiriş Tasarımı**

Kaynaklı kiriş problemi yapısal mühendislik optimizasyon problemidir. Bu sorunun amacı, bir P yükünü taşımak ve minimum maliyetli üretim yapmak için kullanılan b, t, h ve l tasarım değişkenleri için en iyi boyutları bulmaktır. Verilen dört tasarım değişkeni: kaynak kalınlığı (h), çubuğun bağlı olduğu kısmın uzunluğu (l), çubuğun yüksekliği (t) ve çubuğun kalınlığıdır (b). Şekil 20'de Kaynaklı Kiriş Tasarım Problemi verilmiştir.



**Şekil 20. Kaynaklı Kiriş Tasarım Problemi**

Şekil 2.1, sorunun şematik gösterimini göstermektedir. Parametre aralıkları, amaç fonksiyonun matematiksel formülasyonu ve problemin kısıtlamaları aşağıdaki şekilde tanımlanır:

**I. Parametreler**

$$\vec{x} = [x_1 x_2 x_3 x_4] = [hltb]$$

**II. Formül**

$$\text{Minimize } f(\vec{x}) = 1.10471x_1^2 x_2 + 0.04811x_3 x_4 (14.0 + x_2)$$

**III. Kısıtlar**

$$g_1(\vec{x}) = \tau(\vec{x}) - 13600 \leq 0$$

$$g_2(\vec{x}) = \sigma(\vec{x}) - 30000 \leq 0$$

$$g_3(\vec{x}) = \delta(\vec{x}) - 0.25 \leq 0$$

$$g_4(\vec{x}) = x_1 - x_4 \leq 0$$

$$g_5(\vec{x}) = 6000 - P_c(\vec{x}) \leq 0$$

$$g_6(\vec{x}) = 0.125 - x_1 \leq 0$$

$$g_7(\vec{x}) = 1.10471x_1^2 + 0.04811x_3 x_4 (14.0 + x_2) - 5.0 \leq 0$$

$$\tau(\vec{x}) = \sqrt{(\tau')^2 + (\tau'')^2 + \frac{(x_2 \tau' \tau'')}{\sqrt{0.25(x_2^2 + (x_1 + x_3)^2)}}$$

$$\tau' = \frac{6000}{\sqrt{2}x_1 x_2}$$

$$\sigma(\vec{x}) = \frac{504000}{x_3^2 x_4}$$

$$\delta(\vec{x}) = \frac{65856000}{(30 * 10^6)x_4 x_3^3}$$

$$\tau'' = \frac{6000(14 + 0.5x_2)\sqrt{0.25(x_2^2 + (x_1 + x_3)^2)}}{2 \left[ 0.707x_1 x_2 \left( \frac{x_2^2}{12} + 0.25(x_1 + x_3)^2 \right) \right]}$$

$$P_c(\vec{x}) = 64746.022(1 - 0.0282346x_3)x_3 x_4^3$$

## 2209/A ÜNİVERSİTE ÖĞRENCİLERİ ARAŞTIRMA PROJELERİ DESTEĞİ PROGRAMI ARAŞTIRMA ÖNERİSİ FORMU

### IV. Parametre Çözüm Aralığı

$$0.1 \leq x_1, x_2$$

$$x_3 \leq 10.0$$

$$x_4 \leq 2.0$$

#### 2.4.1.1.4.5. Dişli Tren Tasarımı

Bu tasarım probleminin amacı, dişli oranını en aza indirmek için dört dişli ( $T_a$ ,  $T_b$ ,  $T_d$ ,  $T_f$ ) için en uygun diş sayısını araştırmaktır. Parametre aralıkları, amaç fonksiyonun matematiksel formülasyonu ve problemin kısıtlamaları aşağıdaki şekilde tanımlanır:

##### I. Parametreler

$$\vec{x} = [x_1 x_2 x_3 x_4] = [T_a T_b T_d T_f]$$

##### II. Formül

$$\text{Minimize } f(\vec{x}) = \left( \frac{1}{6.931} - \frac{x_2 x_3}{x_1 x_4} \right)^2$$

##### III. Parametre Çözüm Aralığı

$$12 \leq x_1, x_2, x_3, x_4 \leq 60$$

#### 2.4.2. Test Çalışmaları

Test çalışmalarının işlevi bir çeşit ön incelemedir. Test çalışmaları aşamasında 25 MSA algoritması arasından en iyi performansa sahip 15 algoritma belirlenecektir. Bunun için 90 adet problem kullanılacaktır. Bu problemler test problemleri başlığı altında açıklanmıştır. Algoritmaların sıralaması ise Friedman testi ile yapılacaktır. Test çalışmaları ayrıca algoritma geliştirme çalışmasında da işletecektir. Algoritma geliştirme aşamasında, DSK yöntemiyle melezlenecek algoritmaların performansları araştırılacaktır.

#### 2.4.3. Doğrulama Çalışmaları

Doğrulama çalışmalarının amacı test çalışmalarından elde edilen sonuçların doğruluğunu araştırmak ve onaylamaktır. Bu proje çalışmasında iki defa doğrulama çalışması yapılacaktır. İlk çalışma, test aşamasında belirlenen 15 Algoritma arasından mühendislik tasarım problemleri üzerinde en iyi performans sergileyen ilk üç algoritma belirlenmesidir. Bunun için 15 algoritmanın mühendislik problemlerindeki performansı test edilecek ve birbirleriyle karşılaştırılacaktır. Belirlenen 3 MSA tekniği daha sonra DSK yöntemleriyle geliştirilmeye çalışılacaktır. Geliştirme çalışmaları testlerle incelenecektir. Yani üzerinde geliştirme çalışması yapılan algoritmalar tekrar testlere tabi tutulacaktır. Test çalışmaları neticesinde belirlenen MSA algoritmalarında geliştirmeler yapıldıktan sonra algoritmaların performanslarının doğrulanmasına ihtiyaç vardır. Dolayısıyla ikinci doğrulama çalışmaları bu aşamada gerçekleştirilecektir.

#### 2.4.4. Analiz Yöntemleri

Deneyisel çalışma sonuçlarında algoritmaların performansları istatistiksel test ve analiz yöntemleri olan wilcoxon ve friedman testleri ile analiz edilecektir. Wilcoxon testi iki örneklem ortalamaları arasında anlamlı olan farklılıkları tespit etmeyi amaçlar. Eğer iki algoritmanın çıkışlarının kıyaslanması için kullanılacaksa, test pratik olarak iki algoritmanın karşılıklı davranışlarını değerlendirir. Friedman testi ise çoklu karşılaştırmalarda algoritmaları arama hatalarının performanslarına göre sıralamak için kullanılacaktır.

#### 2.5. Sonuç

Bu proje çalışmasında, MSA algoritmalarının DSK yöntemiyle iyileştirilmeleri ve mühendislik tasarım problemlerinin optimizasyonu konularında yapılması planlanan araştırma ve geliştirme faaliyetleri sunulmuştur. MSA algoritmalarının optimizasyon çalışmaları açısından önemi açıkça ortaya koyulmuştur. MSA algoritmalarının geliştirilmeleri, test edilmeleri ve doğrulanmaları süreçleri genel hatlarıyla açıklanmıştır. MSA algoritmalarına ilişkin iyileştirme çalışmalarının yürütülmesinde DSK yönteminin önemi ve rolü açıklanmıştır. DSK yönteminin teknik detayları sunulmuştur. MSA algoritmalarında temel iki gereksinim olan yakınsama (komşuluk araması) ve çeşitlilik (yerel çözüm tuzaklarından kurtulma) işlevleri detaylı tartışılmıştır. Algoritmaların melezlenmelerinde ve DSK yönteminin tatbik edilmesindeki amacın bu iki gereksinimin en yüksek seviyede karşılanması olduğuna dikkat çekilmiştir. Projenin amacının MSA algoritmalarının arama performanslarını iyileştirmek olduğu belirtilmiştir. Bu amaca yönelik olarak takip edilecek yol planlanmış ve sunulmuştur. Algoritma geliştirme çalışmalarındaki başlıca hedefin mühendislik tasarım problemleri için daha optimum çözümlerin elde edilmesi olduğu belirtilmiştir. Bu hedefe yönelik olarak mühendislik tasarım problemleri örnekleri verilmiştir.

## 3 PROJE YÖNETİMİ

## 3.1 İş- Zaman Çizelgesi

## İŞ-ZAMAN ÇİZELGESİ (\*)

İP No	İş Paketlerinin Adı ve Hedefleri	Kim(ler) Tarafından Gerçekleştirileceği	Zaman Aralığı (.-. Ay)	Başarı Ölçütü ve Projenin Başarısına Katkısı
1	Problem Tanımı	Başvuru Sahibi	1.9.2019 – 1.10.2019	MSA algoritmalarının doğada var olan mükemmel işleyişlerin matematiksel dönüşümleri olsalar da işleyişlerindeki problemlerin tanımlanması/vurgulanması. Katkı %15
2	Analiz	Başvuru Sahibi	1.10.2019 – 2.12.2019	Matlab File Exchange kaynak havuzundan MSA algoritmalarının kodlarına erişilip 5 adet kısıtlı mühendislik problemi, 90 adet kıyaslama ve karşılaştırma test problemi üzerindeki performanslarının istatistiksel test ve analiz yöntemleriyle en iyi 3 algoritmanın tespiti. Katkı %15
3	Tasarım	Başvuru Sahibi	2.1.2020 – 20.3.2020	Literatürde bulunan mevcut DSK uygulamalarının makale ,yayın ve tezlerine erişilip DSK tekniklerinin analizi. Katkı %30
4	Gerçekleştirim Çalışması	Başvuru Sahibi	20.3.2020 – 22.5.2020	Literatürdeki DSK uygulamalarının analizinde elde edilen deneyimler ile belirlenen 3 MSA algoritmasının DSK teknikleri ile geliştirilme çalışmaları. Katkı %20
5	Test ve Doğrulama	Başvuru Sahibi	22.5.2020 – 30.6.2020	DSK teknikleri ile varyasyonları çıkartılan MSA algoritmaları ve baz modellerinin performanslarının istatistiksel test ve analiz yöntemleriyle karşılaştırılıp en iyi melez MSA algoritmasının tespiti. Katkı %20

**2209/A ÜNİVERSİTE ÖĞRENCİLERİ ARAŞTIRMA PROJELERİ DESTEĞİ PROGRAMI**  
**ARAŞTIRMA ÖNERİSİ FORMU**

**3.2 Risk Yönetimi**

**RİSK YÖNETİMİ TABLOSU\***

İP No	En Önemli Riskler	Risk Yönetimi (B Planı)
1	Zaman kısıtından dolayı projeyi yetiştirememesi	Bilgisayarları daha etkin kullanabilmek için problemler bölünerek paralel hesaplama yapılarak iş yükü azaltılacaktır.
2	Algoritmalarından beklenen başarının elde edilememesi	Hali hazırda danışmanım ile birlikte proje konusunda çalışmalar yürütmekteyiz. Bu çalışmalarda uyguladığımız levy ve kaos haritalarının algoritmaların performanslarını iyileştirdiklerini gördük. Dolayısıyla projede bu riskin oluşması beklenmemektedir.

**3.3. Araştırma Olanakları**

**ARAŞTIRMA OLANAKLARI TABLOSU (\*)**

Kuruluşta Bulunan Altyapı/Ekipman Türü, Modeli (Laboratuvar, Araç, Makine-Teçhizat, vb.)	Projede Kullanım Amacı
Bilgisayar labortavuarları	Geliştirilecek olan algoritmanın test ve doğrulaması için kullanılacaktır.

**4. YAYGIN ETKİ**

**ARAŞTIRMA ÖNERİSİNDEN BEKLENEN YAYGIN ETKİ TABLOSU**

Yaygın Etki Türleri	Önerilen Araştırmadan Beklenen Çıktı, Sonuç ve Etkiler
<b>Bilimsel/Akademik</b> (Makale, Bildiri, Kitap Bölümü, Kitap)	Proje çerçevesinde 1 (bir) adet bildiri ve 1 adet makale çalışması yapılacaktır.
<b>Ekonomik/Ticari/Sosyal</b> (Ürün, Prototip, Patent, Faydalı Model, Üretim İzni, Çeşit Tescili, Spin-off/Start-up Şirket, Görsel/İşitsel Arşiv, Envanter/Veri Tabanı/Belgeleme Üretimi, Telif Konu Olan Eser, Medyada Yer Alma, Fuar, Proje Pazarı, Çalıştay, Eğitim vb. Bilimsel Etkinlik, Proje Sonuçlarını Kullanacak Kurum/Kuruluş, vb. diğer yaygın etkiler)	Araştırmacılar, proje çerçevesinde geliştirilecek olan melez MSA algoritmalarına web ortamından erişim sağlayabilecekler ve optimizasyon problemlerini ücretsiz bir şekilde çözümlenebileceklerdir.
<b>Araştırmacı Yetiştirilmesi ve Yeni Proje(ler) Oluşturma</b> (Yüksek Lisans/Doktora Tezi, Ulusal/Uluslararası Yeni Proje)	Proje konusu araştırırken çok sayıda makale ve teze rastladım. Dolayısıyla proje konusu lisansüstü veya doktora çalışmaların yapılabileceği niteliktedir. Lisans sonrası lisansüstü ve doktora eğitimine devam etme planım bulunmaktadır. Meta Sezgisel Optimizasyon konusundaki bu projeyi başarıyla tamamlamam ve bu konuda danışmanım ile birlikte makale hazırlamam durumunda lisansüstü eğitim ve yeni projeler konusunda da ciddi bir yol almam söz konusu olabilecektir. Hali hazırda Havelsan Komuta Kontrol Savunma Teknolojileri bölümünde iş yeri eğitimim devam etmektedir. Mezuniyet sonrası için Havelsan'da Komuta Kontrol Savunma Teknolojileri bölümünde ARGE yazılım mühendisi olarak işe başlamam söz konusudur. Bu birimde yazılımların optimizasyonu konusunda proje tecrübesinin önemli katkı sağlayacağını düşünmekteyim.

**5. BELİRTMEK İSTEDİĞİNİZ DİĞER KONULAR**

Proje çerçevesinde geliştirilecek olan MSA algoritmalarının kodlarına uluslararası bir kaynak havuzu (özel uygulamalar [görüntü işleme, yapay zeka ], kod örnekleri, simülasyon modelleri, komut dosyaları vb.) paylaşım uygulaması olan "Matlab File Exchange" kullanılacaktır. Bu uygulamaya aşağıdaki bağlantıdan erişilebilmektedir:

<https://www.mathworks.com/matlabcentral/fileexchange/>

**6. EKLER**

**EK-1: KAYNAKLAR**



**2209/A ÜNİVERSİTE ÖĞRENCİLERİ ARAŞTIRMA PROJELERİ DESTEĞİ PROGRAMI**  
**ARAŞTIRMA ÖNERİSİ FORMU**

1. Haklı, H. (2013). Sürekli fonksiyonların optimizasyonu için doğa esinli algoritmaların geliştirilmesi (Yüksek Lisans Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü).
2. Kahraman H.T., Aras S., Gedikli E., "Meta-Sezgisel Optimizasyon Çalışmalarında Benchmark Problemlerinde Karşılaşılan Standartsızlıklar Ve Çözüm Önerileri", International Academic Research Congress, Antalya, Türkiye, 30 Ekim – 3 Aralık 2018, Pp. 1494-1501.
3. Kahraman Hamdi Tolga, Aras Sefa, Gedikli Eyüp, Meta-Sezgisel Algoritmaların Deneysel Çalışmalarındaki Standartsızlıklar Ve Çözüm Önerileri, International Academic Research Congress, 30 Ekim-3 Kasım 2018, 1466-1472.
4. Lai, W., Zhou, M., Hu, F., Bian, K., & Song, Q. (2019). A New DBSCAN Parameters Determination Method Based on Improved MVO. IEEE Access, 7, 104085-104095.
5. Pham, H. N. A., & Triantaphyllou, E. (2009). An application of a new meta-heuristic for optimizing the classification accuracy when analyzing some medical datasets. Expert Systems with Applications, 36(5), 9240-9249.
6. Kahraman, H. T. (2016). A novel and powerful hybrid classifier method: Development and testing of heuristic k-nn algorithm with fuzzy distance metric. Data & Knowledge Engineering, 103, 44-59.
7. Fong, S., Deb, S., & Yang, X. S. (2018). How meta-heuristic algorithms contribute to deep learning in the hype of big data analytics. In Progress in Intelligent Computing Techniques: Theory, Practice, and Applications (pp. 3-25). Springer, Singapore.
8. Tayal, A., & Singh, S. P. (2018). Integrating big data analytic and hybrid firefly-chaotic simulated annealing approach for facility layout problem. Annals of Operations Research, 270(1-2), 489-514.
9. Dosoglu, M. K., Guvenc, U., Duman, S., Sonmez, Y., & Kahraman, H. T. (2018). Symbiotic organisms search optimization algorithm for economic/emission dispatch problem in power systems. Neural Computing and Applications, 29(3), 721-737.
10. Zhang, J., Ding, G., Zou, Y., Qin, S., & Fu, J. (2019). Review of job shop scheduling research and its new perspectives under Industry 4.0. Journal of Intelligent Manufacturing, 30(4), 1809-1830.
11. Tian, D., Zhao, X., & Shi, Z. (2019). Chaotic particle swarm optimization with sigmoid-based acceleration coefficients for numerical function optimization. Swarm and Evolutionary Computation, 100573.
12. Gupta, S., & Deep, K. (2019). A hybrid self-adaptive sine cosine algorithm with opposition based learning. Expert Systems with Applications, 119, 210-230.
13. Jana, B., Mitra, S., & Acharyya, S. (2019). Repository and Mutation based Particle Swarm Optimization (RMPSO): A new PSO variant applied to reconstruction of Gene Regulatory Network. Applied Soft Computing, 74, 330-355.
14. Wu, L., Liu, Q., Tian, X., Zhang, J., & Xiao, W. (2018). A new improved fruit fly optimization algorithm IAFOA and its application to solve engineering optimization problems. Knowledge-Based Systems, 144, 153-173.
15. Sun, G., Ma, P., Ren, J., Zhang, A., & Jia, X. (2018). A stability constrained adaptive alpha for gravitational search algorithm. Knowledge-Based Systems, 139, 200-213.
16. Long, W., Jiao, J., Liang, X., & Tang, M. (2018). An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization. Engineering Applications of Artificial Intelligence, 68, 63-80.
17. Awad, N. H., Ali, M. Z., Mallipeddi, R., & Suganthan, P. N. (2018). An improved differential evolution algorithm using efficient adapted surrogate model for numerical optimization. Information Sciences, 451, 326-347.
18. Al-Bahrani, L. T., & Patra, J. C. (2018). A novel orthogonal PSO algorithm based on orthogonal diagonalization. Swarm and Evolutionary Computation, 40, 1-23.
19. Torabi, S., & Safi-Esfahani, F. (2018). Improved raven roosting optimization algorithm (IRRO). Swarm and Evolutionary Computation, 40, 144-154.
20. Tian, D., & Shi, Z. (2018). MPSO: Modified particle swarm optimization and its applications. Swarm and Evolutionary Computation.
21. Chegini, S. N., Bagheri, A., & Najafi, F. (2018). PSOSCALT: A new hybrid PSO based on Sine Cosine Algorithm and Levy flight for solving optimization problems. Applied Soft Computing, 73, 697-726.
22. Zhong, F., Li, H., Zhong, S. 2017. "An improved artificial bee colony algorithm with modified-neighborhood-based update operator and independent-inheriting-search strategy for global optimization", Engineering Applications of Artificial Intelligence, 58, 134-156.
23. Ouyang, H. B., Gao, L. Q., Li, S., Kong, X. Y., Wang, Q., Zou, D. X. 2017. "Improved harmony search algorithm: LHS", Applied Soft Computing, 53, 133-167.
24. Harfouchi, F., Habbi, H., Ozturk, C., & Karaboga, D. (2017). Modified multiple search cooperative foraging strategy for improved artificial bee colony optimization with robustness analysis. Soft Computing, 1-24.
25. Awad, N. H., Ali, M. Z., Suganthan, P. N., & Reynolds, R. G. (2017). CADE: a hybridization of cultural algorithm and differential evolution for numerical optimization. Information Sciences, 378, 215-241.
26. Mortazavi, A., Toğan, V., & Nuhuğlu, A. (2018). Interactive search algorithm: a new hybrid metaheuristic optimization algorithm. Engineering Applications of Artificial Intelligence, 71, 275-292.
27. Ewees, A. A., Elaziz, M. A., & Houssein, E. H. (2018). Improved grasshopper optimization algorithm using opposition-based learning. Expert Systems with Applications, 112, 156-172.

28. Arora, S., & Singh, S. (2019). Butterfly optimization algorithm: a novel approach for global optimization. *Soft Computing*, 23(3), 715-734.
29. Civicioglu, P., Besdok, E., Gunen, M. A., & Atasever, U. H. (2018). Weighted differential evolution algorithm for numerical function optimization: a comparative study with cuckoo search, artificial bee colony, adaptive differential evolution, and backtracking search optimization algorithms. *Neural Computing and Applications*, 1-15.
30. Derrac, J., García, S., Molina, D., & Herrera, F. 2011. "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms". *Swarm and Evolutionary Computation*, 1, (1), 3-18.
31. Martin, L., Leblanc, R., & Toan, N. K. 1993. "Tables for the Friedman rank test". *Canadian journal of statistics*, 21, 1, 39-43.
32. Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, 97, 849-872.
33. W. Zhao, L. Wang and Z. Zhang, Atom search optimization and its application to solve a hydrogeologic parameter estimation problem, *Knowledge-Based Systems* (2019), 163, 283-304.
34. Mohamed, A. W., & Mohamed, A. K. (2019). Adaptive guided differential evolution algorithm with novel mutation for numerical optimization. *International Journal of Machine Learning and Cybernetics*, 10(2), 253-277.
35. Yadav, A. (2019). AEFA: Artificial electric field algorithm for global optimization. *Swarm and Evolutionary Computation*.
36. Tang, D., Liu, Z., Yang, J., & Zhao, J. (2018). Memetic frog leaping algorithm for global optimization. *Soft Computing*, 1-29.
37. Chen, X., & Xu, B. (2018, June). Teaching-learning-based artificial bee colony. In *International Conference on Swarm Intelligence* (pp. 166-178). Springer, Cham.
38. Wang, G. G. (2018). Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems. *Memetic Computing*, 10, 151-164.
39. Pierezan, J., & Coelho, L. D. S. (2018, July). Coyote optimization algorithm: a new metaheuristic for global optimization problems. In *2018 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1-8). IEEE.
40. Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, 114, 163-191.
41. Mirjalili, S., & Gandomi, A. H. (2017). Chaotic gravitational constants for the gravitational search algorithm. *Applied soft computing*, 53, 407-419.
42. Abedinpourshotorban, H., Shamsuddin, S. M., Beheshti, Z., & Jawawi, D. N. (2016). Electromagnetic field optimization: A physics-inspired metaheuristic optimization algorithm. *Swarm and Evolutionary Computation*, 26, 8-22.
43. Punnnathanam, V., & Kotecha, P. (2016). Yin-Yang-pair Optimization: A novel lightweight optimization algorithm. *Engineering Applications of Artificial Intelligence*, 54, 62-79.
44. Mittal, H., Pal, R., Kulhari, A., & Saraswat, M. (2016, August). Chaotic kbest gravitational search algorithm (ckgsa). In *2016 Ninth International Conference on Contemporary Computing (IC3)* (pp. 1-6). IEEE.
45. Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Computers & Structures*, 169, 1-12.
46. Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, 95, 51-67.
47. Mirjalili, S. (2016). SCA: a sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems*, 96, 120-133.
48. Salimi, H. (2015). Stochastic fractal search: a powerful metaheuristic algorithm. *Knowledge-Based Systems*, 75, 1-18.
49. Shareef, H., Ibrahim, A. A., & Mutlag, A. H. (2015). Lightning search algorithm. *Applied Soft Computing*, 36, 315-333.
50. Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, 89, 228-249.
51. Cheng, Min-Yuan, and Doddy Prayogo. "Symbiotic organisms search: a new metaheuristic optimization algorithm." *Computers & Structures* 139 (2014): 98-112.
52. Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. *Information sciences*, 179(13), 2232-2248.
53. Karaboga, D., & Akay, B. (2009). A comparative study of artificial bee colony algorithm. *Applied mathematics and computation*, 214(1), 108-132.

54. Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization. *Swarm intelligence*, 1(1), 33-57.
55. Storn, R., & Price, K. (1997). Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4), 341-359.
56. Liang, J. J., Qu, B. Y., & Suganthan, P. N. (2013). Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore.
57. N. H. Awad, M. Z. Ali, J. J. Liang, B. Y. Qu and P. N. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Bound Constrained Real-Parameter Numerical Optimization," Technical Report, Nanyang Technological University, Singapore, November 2016.
58. Long, W., Wu, T., Liang, X., Xu, S.: Solving high-dimensional global optimization problems using an improved sine cosine algorithm. *Expert systems with applications* 123, 108-126 (2019).
59. Lin, X., Zhang, F., Xu, L.: Design of Gear Reducer Based on FOA Optimization Algorithm. In *International Conference on Smart Vehicular Technology, Transportation, Communication and Applications*, pp. 240-247. Springer, Cham (2017).
60. Dong, M., Wang, N., Cheng, X., Jiang, C.: Composite differential evolution with modified oracle penalty method for constrained optimization problems. *Mathematical problems in engineering*, 1-15 (2014), <http://dx.doi.org/10.1155/2014/617905>.
61. Amir M.: Towards An Approach For Effectively Using Intuition In Large-Scale Decision-Making Problems, PhD Thesis, University of Debrecen (2013).
62. Kahraman, H. T., Aras, S., Guvenc, U., & Sonmez, Y. (2017, October). Exploring the effect of distribution methods on meta-heuristic searching process. In *2017 International Conference on Computer Science and Engineering (UBMK)* (pp. 371-376). IEEE.
63. Sun, W., Lin, A., Yu, H., Liang, Q., & Wu, G. (2017). All-dimension neighborhood based particle swarm optimization with randomly selected neighbors. *Information Sciences*, 405, 141 – 156.
64. Tu, Q., Chen, X., & Liu, X. (2019). Multi-strategy ensemble grey wolf optimizer and its application to feature selection. *Applied Soft Computing*, 76, 16-30.
65. Tian, M., & Gao, X. (2019). Differential evolution with neighborhood-based adaptive evolution mechanism for numerical optimization. *Information Sciences*, 478, 422-448.
66. Draa, A., Chettah, K., & Talbi, H. (2018). A Compound Sinusoidal Differential Evolution algorithm for continuous optimization. *Swarm and Evolutionary Computation*.
67. Gülcan, H. (2018). Yusufçuk algoritmasının brownian hareketi ile iyileştirilmesi (Yüksek Lisans Tezi, Mersin Üniversitesi Fen Bilimleri Enstitüsü).
68. Chechkin, A.V., Metzler, R., Klafter, J. and Gonchar, V.Y., 2008, *Anomalous Transport: Foundations and Applications*, Klages, R. , Radons, G. , and Sokolov, I. M., John Wiley & Sons, Weinheim, 129-162.
69. Chen, Y. , 2010, Research and simulation on Levy Flight model for DTN, 2010 3rd International Congress on Image and Signal Processing, Yantai, China, 4421- 4423
70. Cheng, Z. ve Savit, R., 1987, Fractal and nonfractal behavior in Levy flights, *Journal of mathematical physics*, 28 (3), 592-597.
71. Brown, C. T., Liebovitch, L. S. ve Glendon, R., 2007, Lévy flights in Dobe Ju'hoansi foraging patterns, *Human Ecology*, 35 (1), 129-138
72. Pavlyukevich, I., 2007, Lévy flights, non-local search and simulated annealing, *Journal of Computational Physics*, 226 (2), 1830-1844.
73. Yang, X.-S. and Deb, S., 2013, Multiobjective cuckoo search for design optimization, *Computers & Operations Research*, 40, 1616-1624.
74. Yang, X.-S., 2010a, Firefly Algorithm, Levy Flights and Global Optimization, Bramer, M., Ellis, R. and Petridis, M. (Eds.), *Research and Development in Intelligent Systems XXVI*, Springer London, 209-218.
75. Heidari, A. A., Pahlavani, P. (2017). An efficient modified grey wolf optimizer with Lévy flight for optimization tasks. *Applied Soft Computing Journal*, 60, 115–134. doi:10.1016/j.asoc.2017.06.044
76. Mirjalili, S. (2016). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*, 27(4), 1053–1073. doi:10.1007/s00521- 015-1920-1
77. Lee, C.-Y. and Yao, X., 2001, Evolutionary Algorithms with Adaptive Levy Mutations,. *Proceedings of the 2001 Congress on Evolutionary Computation*, Seoul, South Korea, 568-575.
78. Cigal, T. (2018). Sürekli zamanlı kaotik sistem tabanlı balina optimizasyon algoritmasının geliştirilmesi (Yüksek Lisans Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü).
79. Alatas B., 2010. Chaotic bee colony algorithms for global numerical optimization. *Expert Systems with Applications*, 37(8), 5682-5687.
80. YILDIRIM, G., AYDIN, G., ALLİ, H., & TATAR, Y. Hadoop ile Kaos Temelli FCW Optimizasyon Algoritmasının Analizi An Analysis of Chaos-Based the FCW Optimization Algorithm by Hadoop.

81. Kaya M. Tarım arazisi verimliliği algoritmasının başlangıç popülasyonunun kaotik haritalarla oluşturulması (Yüksek Lisans Tezi, Düzce Üniversitesi Fen Bilimleri Enstitüsü).
82. Alataş, B., 2007. Kaotik haritalı parçacık sürü optimizasyonu algoritmaları geliştirme (Doktora Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elâzığ).
83. Demir, F. B., Tuncer, T., & Kocamaz, A. F. Lojistik-Gauss Harita Tabanlı Yeni Bir Kaotik Sürü Optimizasyon Yöntemi. *Anatolian Science-Bilgisayar Bilimleri Dergisi*, 47-53.
84. Alatas B., Akin E., & Ozer A. B. 2009. Chaos embedded particle swarm optimization algorithms. *Chaos, Solitons & Fractals*, 40(4), 1715-1734.
85. Tan Y. Tan G. Deng S., 2014. Hybrid particle swarm optimization with chaotic search for solving integer and mixed integer programming problems, *Journal of Central University*, Volume 21, Issue 7, 2731-2742
86. Gandomi A. H., Yang X. S., Talatahari S., & Alavi A. H. 2013. Firefly algorithm with chaos. *Communications in Nonlinear Science and Numerical Simulation*, 18(1), 89- 98
87. Kohli M. & Arora S. 2017. Chaotic grey wolf optimization algorithm for constrained optimization problems. *Journal of Computational Design and Engineering*.
88. Alatas B. 2011. Uniform big bang–chaotic big crunch optimization. *Communications in Nonlinear Science and Numerical Simulation*, 16(9), 3696-3703
89. Alatas B. 2010. Chaotic harmony search algorithms. *Applied Mathematics and Computation*, 216(9), 2687-2699.
90. Tanyıldızı E. & Cigal T. 2017. Kaotik Haritalı Balina Optimizasyon Algoritması. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, 29(1).
91. Caponetto R., Fortuna L., Fazzino S. & Xibilia, M. G., 2003. Chaotic sequences to improve the performance of evolutionary algorithms. *IEEE transactions on evolutionary computation*, 7(3), 289-304.
92. Stoyanov B., 2014. Pseudo-random Bit Generation Algorithm Based on Chebyshev Polynomial and Tinkerbell Map, *Applied Mathematical Sciences*, Vol. 8, 2014, no. 125, 6205 – 6210.
93. Büyükuysal, M. (2014). Farklı örneklem genişliklerinde normal dağılım testlerinin karşılaştırılması (Doktora Tezi, Bülent Ecevit Üniversitesi Sağlık Bilimleri Enstitüsü).
94. N. Higashi, H. Iba, Particle swarm optimization with gaussian mutation, in: *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*, 72–79.
95. Mahi, M., Baykan, Ö. K., Kodaz, H., "A new hybrid method based on Particle Swarm Optimization, Ant Colony Optimization and 3-Opt algorithms for Traveling Salesman Problem", *Applied Soft Computing*, 30, 484–490, (2015).
96. Han, X., Liu, Q., Wang, H., & Wang, L. (2018). Novel fruit fly optimization algorithm with trend search and co-evolution. *Knowledge-Based Systems*, 141, 1-17.
97. W. Gao, S. Liu, L. Huang, A novel artificial bee colony algorithm based modified search equation and orthogonal learning, *IEEE Trans. Cybern.* 43 (3) (2013) 1011–1024.