TÜBİTAK 2209-A ÜNİVERSİTE ÖĞRENCİLERİ
YURT İÇİ ARAŞTIRMA PROJELERİ
DESTEK PROGRAMI

META-SEZGİSEL OPTİMİZASYON ALGORİTMALARININ DSK YÖNTEMİYLE GELİŞTİRİLMESİ VE KISITLI MÜHENDİSLİK TASARIM PROBLEMLERİNİN OPTİMİSAZYONU

KARADENİZ TEKNİK ÜNİVERSİTESİ
YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ

TEMATİK ALANI
YAZILIM MÜHENDİSLİĞİ
BİLİŞİM TEKNOLOJİLERİ MÜHENDİSLİĞİ

PROJE YÜRÜTÜCÜSÜ
MEHMET KATI

PROJE DANIŞMANI
DOÇ. DR. HAMDİ TOLGA KAHRAMAN

İÇİNDEKİLER

1. Özet

Klasik optimizasyon teknikleri problemin boyutunun büyük olması, lineer olmaması, çözüm uzayının geniş olması gibi nedenlerden dolayı yetersiz kalmaktadır. Bundan dolayı kesin çözüm yerine bulunabilecek en uygun çözümlere yönelmek için doğa-esinli meta sezgisel optimizasyon algoritmaları geliştirilmiştir [1].

Meta-sezgisel arama sürecinde doğadaki başarı öykülerini taklit eden bu algoritmaların uygulama alanları sadece optimizasyon ile sınırlı değildir. Tahmin, sınıflandırma ve kümeleme problemlerinin modellenmesinin yanı sıra melez algoritmaların tasarımında ve geliştirilmesinde de yaygın ve başarılı bir şekilde uygulanmaktadırlar. Özellikle yapay sinir ağlarının optimizasyonu, k-en yakın komşu sınıflandırıcının, karar ağaçlarının ve bulanık mantık-temelli algoritmaların melezleştirilmesi ve son dönemlerin popüler araştırma konularından derin öğrenme, büyük veri uygulamalarında, Endüstri 4.0 gibi modern otomasyon sistemlerinin ve uygulamalarının geliştirilmesinde meta-sezgisel optimizasyon tekniklerinden faydalanılmaktadır.

Geniş kullanım alanına sahip Meta-sezgisel algoritmalarının arama sürecinde iki temel problemle karşılaşmaktadırlar. Bu problemler, yerel minimum tuzaklarına yakalanmak (çeşitliliği sağlayamamak) ve komşuluk aramasını hassas bir şekilde gerçekleştirememektir. Algoritmaların mevcut zayıf yönlerini güçlendirip arama performanslarını artırmak için, algoritmalar DSK (Dağılım, Seçim, Kontrol) olarak adlandırılan 3 bölüme ayrılmıştır. Bu bölümlerde yapılacak iyileştirmeler ile kısıtlı mühendislik tasarım problemlerindeki arama performansı DSK yöntemleri ile optimize edilmiş meta sezgisel arama (MSA) algoritmalarının geliştirilmesi hedeflenmiştir.

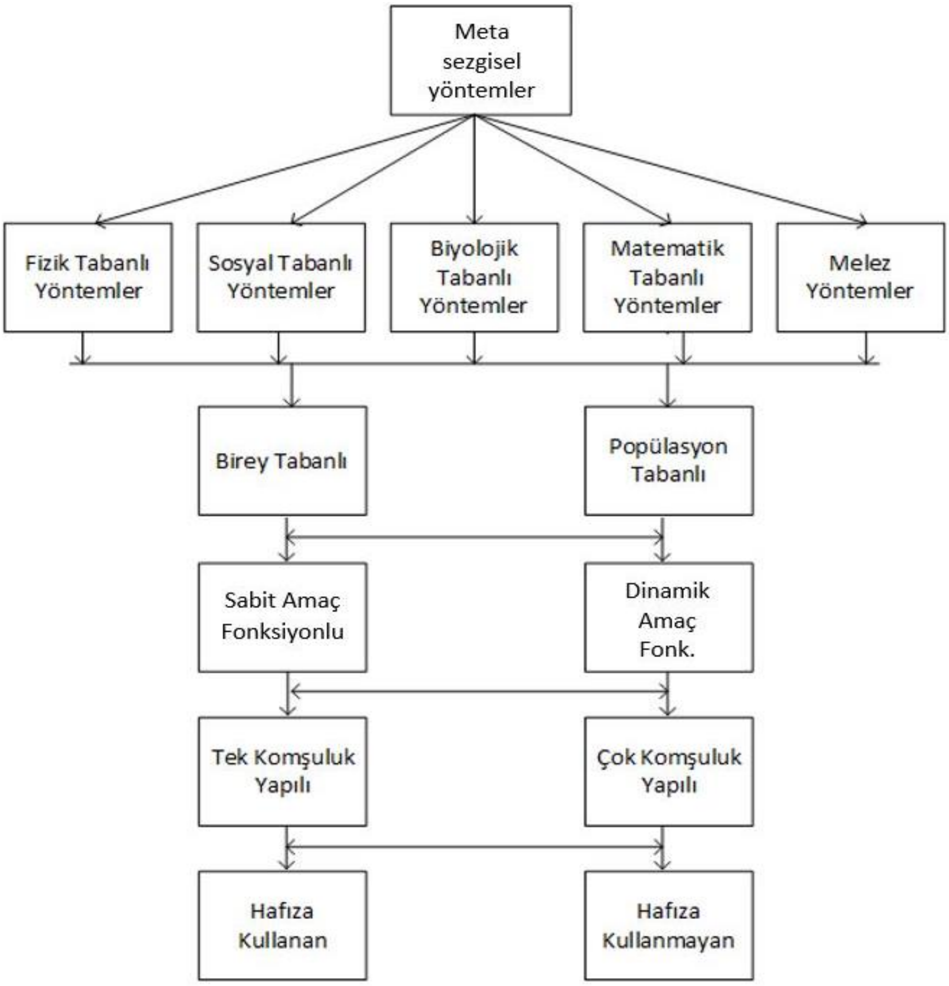
2. Giriş

Optimizasyon çeşitli problemleri çözmek amacıyla karar değişkenleri ve kısıtlar altında, en uygun çözümü arama yöntemlerine denir [2]. Herhangi bir alana optimizasyon işlemi uygulanarak performans ve minimum maliyeti sağlayan çözüm elde edilmeye çalışılır [1].

Değişen dünya ve gelişen teknoloji ortamında ortaya çıkan problemler gün geçtikçe zorlaşmaktadır [1]. Çoğunda problemin çözüm uzayı tüm çözümleri değerlendirilmeyeceği kadar büyük olur [2]. Klasik optimizasyon metotları ile gerçek dünya problemlerini çözmek çok zaman almakta ve etkili bir şekilde çözülememektedir [1].

Optimizasyon üzerine çalışan bazı bilim insanları, yüksek performanslı, dinamik ve esnek yöntemler geliştirmek için dikkatlerini doğaya vermişlerdir. Bunun sebebi, doğada var olan ve kendiliğinden muazzam bir işleyişe sahip olan sistemlerin, optimizasyon problemlerini çok daha efektif bir şekilde çözebilecekleri düşüncesidir. Çünkü doğa, zaten var olan kompleks ve zor olan optimizasyon problemini, yine doğada var olan çoğu zaman gizemini sürdüren yöntemlerle çözmektedir. Doğada var olan sistemleri ve olayları temel alarak oluşturulan optimizasyon teknikleri sezgisel yöntemler olarak adlandırılır. Sezgisel yaklaşım en iyiyi bulma garantisine sahip değildir ve bu nedenle genel olarak optimumdan daha kötü çözümler getirir. Bununla birlikte, sezgisel algoritmalar genellikle 'makul' bir sürede iyi çözümler bulurlar. Problemlerin bu denli efektif çözülmesindeki ana sebeplerden birisi rasgele davranma hareketidir. Rasgele hareketlilik, problemin çözümünde tek bir yoldan gidilmemesini sağlar ve arama uzayında taranmayan alan bırakmama başarısını getirir [3].

Meta sezgisel yöntemlerin çalışmaları 1950’li yıllara dayanmaktadır. Michigan üniversitesi Prof. John Holland ve öğrencilerinin geliştirdikleri Genetik Algoritma bu çalışmalara hız kazandırmıştır [9]. Çözüm uzayında etkili bir şekilde arama yapmak için, farklı yapılardaki alt kademe sezgisel algoritmaların zekice birleştirilmesi ile oluşturulmuş iteratif problem çözme yöntemlerdir. Bu yöntemler her iterasyonda bir çözümden veya çözüm koleksiyonundan yola çıkarak yeni çözümler üretirler. Çoğu meta sezgisel yaklaşım, çözüm uzayında stokastik fakat bilinçli bir şekilde arama yapar [4]. Şekil 1.1’de sezgisel yöntemlerin sınıflandırılması gösterilmiştir [6].



Şekil 1.1. Meta sezgisel yöntemler

Sezgisel algoritmalar işleyişlerinde rasgele hareketliliklerinde oluşabilecek daralma nedeniyle, çözüm uzayını aramak için oluşturulan çözüm adayları prematüre yakınsama problemine takılabilmektedir. Bu problem tüm çözüm adaylarının en iyi çözüm adayına benzemesine ve bir süre sonra arama uzayında birbirine çok yakın hatta aynı konumu temsil eden çözüm adaylarının oluşmasına neden olur. Böylece bulunan doğa-esinli sezgisel algoritmalar kesin çözüm yerine bulunabilecek en iyi çözüme yöneldiklerinden dolayı yerel minimum veya maksimum noktalarına takılır. Gerçek dünya problemlerindeki çok farklı özelliklere sahip olması ve doğa-esinli bu algoritmaların işleyişlerindeki oluşabilecek zayıf yönler yüzünden istenilen çözümlere ulaşılamamaktadır.

Sezgisel algoritmaların işleyişlerinde oluşan zayıf yönleri iyileştirmek için çok sayıda çalışma yapıldı. Bu bağlamda yapılan çalışmalara ilk örneklerden biri olarak Song ve arkadaşlarının 1999 yılındaki çalışması örnek verilebilir. Bu çalışmada, arı kolonisi algoritması, çok hedefli bir problem yaklaşımındaki kısıtlamaları çözmek için dağıtık hesaplama ve sezgisel bir aç gözlü yaklaşımla iyileştirilmiştir. İlerleyen yıllarda gelindiğinde, 2005 yılında Yan ve arkadaşları, sistematik iş planlama problemi için karınca kolonisini, geçilen yollardaki feromon dengesiyle oynayarak geliştirmişlerdir. 2005 yılında ise Liu ve arkadaşları, parçacık sürü optimizasyonunu, kaos optimizasyon algoritması ile hibritleyerek iyileştirmişlerdir [3]. Mirjalili ve Gandomi [5] yerçekimsel arama algoritmasında yer çekim sabiti olarak kullanılan G parametresinin doğrusal azalış yerine rasgele haraketlilik kazandırmak isen kaos haritalarını kullanarak iyileştirme yapmışlardır. Cigal [6] kaos haritaları ile balina optimizasyon algoritmasını birleştirerek algoritmada iyileşme hedeflemiştir. Haklı ve Uğuz’un 2014’te önerdikleri metotta Parçacık Sürü Optimisazyonu’nu Levy Uçuş Mekanizması ile gerçeklemişlerdir. Bu çalışmada, optimizasyon sırasında ajanların lokal minimaya takılması ve erken yakınsama problemi sebebiyle Levy Uçuş Mekanizması ile bir modifikasyon gerçeklenmiştir ve başarılı sonuçlar alınmıştır [3].

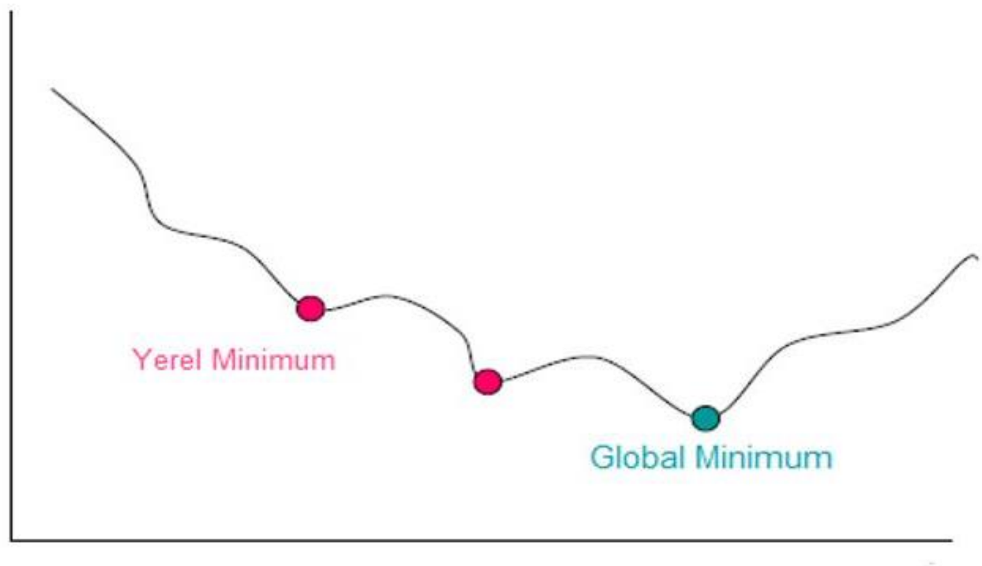
Yapılan çalışmalarında kullanılan yöntemleri DSK olarak kısaltmak ve üç başlık altında toplamak mümkündür. Bunlar sırasıyla popülasyon yaratma (D: dağılım), çözüm adayı seçimi (S: seçim) ve algoritma parametrelerinin kontrolüdür (K: kontrol). Bu çalışmalardan elde edilen tecrübeler göstermektedir ki DSK yöntemlerinden biri ya da birkaçı kullanılarak algoritmaların baz modelleri üzerinde yapılan çalışmalar algoritmaların performansları üzerinde oldukça başarılı sonuçlara yol açmaktadırlar. Bu proje çalışmasının amacı, MSA algoritmasını Kısıtlı Mühendislik Tasarım Problemleri için optimize ederek en etkili MSA yöntemlerinden biri haline getirmektir. Bu amaçla ilk olarak PSO, GSA, EFO ve ASO üzerinde başarılı olmuş yöntemler başta olmak üzere literatürdeki DSK yöntemleri araştırılacaktır. Elde edilecek bilgiler dikkate alınarak kısıtlı mühendislik tasarım problemlerindeki arama performansı DSK yöntemleri ile optimize edilmiş bir MSA algoritması geliştirilecektir.

3. Problem

3.1. Problemin tanımı

Karşılaştığımız problemler için sürekli olarak en iyi çözümleri aramaktayız. Şirketler, kurumlar ve tüm organizasyonlar karını en üst düzeye çıkarmak, maliyetleri düşürmek ve performansı en üst düzeye çıkartmak isterler. Bir şeyi maksimize etmeyi veya en aza indirmeyi planlıyor ve bunun için optimizasyon yapıyoruz. Optimizasyon belirlenen amaç fonksiyonunu, verilen kısıtlar dahilinde en uygun değeri bulmaktır. Aranan değer amaç fonksiyonunu en yüksek yapan ise maksimize problemi, en minimum yapan ise minimize problemidir. Mühendislik tasarımından finansal piyasalara, bilgisayar bilimlerinden endüstriyel uygulamalara kadar çok geniş alanda uygulanmaktadır.

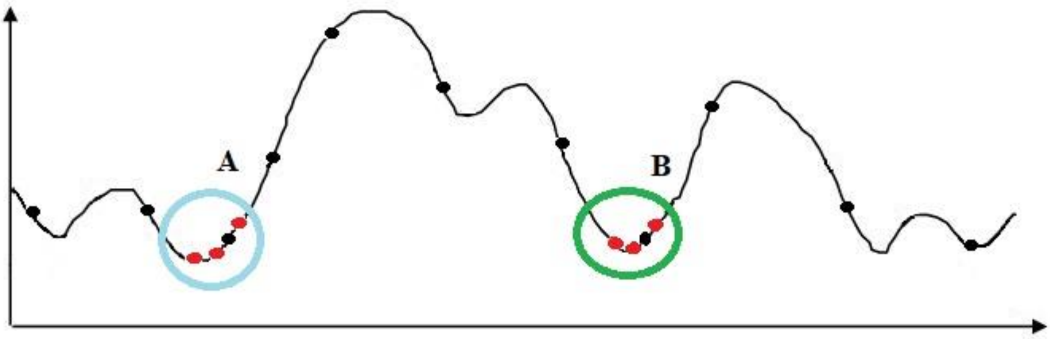
En uygun çözümü bulmayı hazine avcılığına benzetebiliriz. Bulmak istediğimiz şekil 1.2’de verilen minimize problemi için global minimum noktasıdır. Gün geçtikçe optimizasyon yapacağımız problemler zorlaşmakta ve çözüm uzayları büyümekte. Çok geniş bir alanda klasik optimizasyon yöntemleri ile hazine avcılığı yapmak zorlaşmakta ve uzun zaman almaktadır. Bu durum maliyeti artırmaktadır. Optimum çözüme veya optimum çözüme yakın sonuçları kısa zamanda bulmak için doğa-esinli sezgisel algoritmalar geliştirilmiştir.



Şekil 1.2

Sezgisel algoritmalarda temel iki adım bulunmaktadır. Komşuluk araması ve çeşitlilik. Çeşitlilik ile çözüm uzayında büyük atlamalar yapılarak farklı noktalar bulunur. Komşuluk araması adımında ise, çeşitlilik adımında bulunan noktaların etrafında küçük adımlar ile arama yapılır.

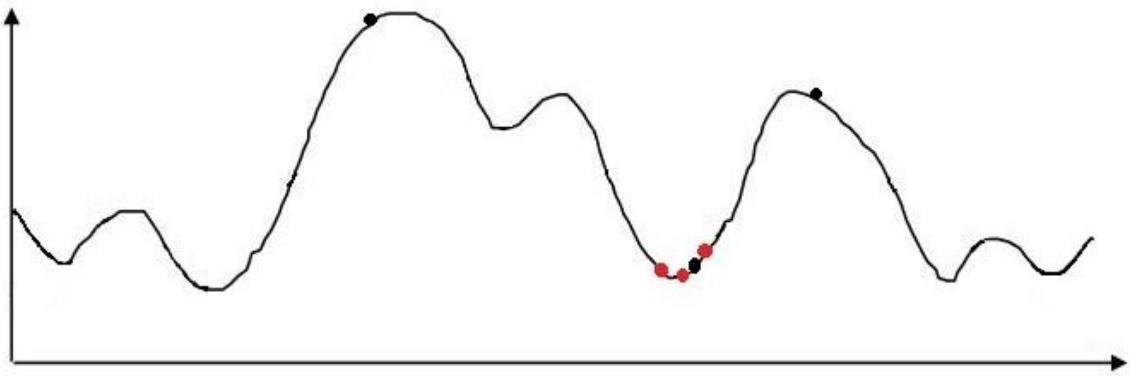
Doğada var olan mükemmel işleyişlerin matematiksel dönüşümleri olsalar da algoritmaların işleyişlerinde bazı problemlerle karşılaşılmaktadır. Geliştirilen sezgisel algoritmalar çok zor ve karmaşık problemlerde beklenilen çözümlerde global çözümler yerine yerel çözüm tuzaklarına takılabilmektedir. Bu problemin temel nedenlerinden birisi çeşitlilik adımının algoritmanın işleyişinde zamanla azalmasıdır. Arama sürecinde uygunluk değeri en büyük olan çözüm adaylarının kullanılması halinde hızlı bir yakınsama oluşmakta ancak bu yakınsama genellikle yerel çözüm tuzağına yakalanmaya neden olmaktadır. Bu durum literatürde prematüre (engelli) yakınsama problemi olarak ifade edilmektedir. Bu durumu minimize problemi olan Şekil 1.3 ve 1.4 üzerinde inceleyelim.



Şekil 1.3

Popülasyon tabanlı sezgisel algoritmalarda, oluşturulan popülasyon ile çözüm uzayı gezinir. Keşif (çeşitlilik) ve sömürü (komşuluk araması) yapılarak en uygun değer bulunmaya çalışılır. Şekil 1.3'de grafik üzerinde verilen siyah daireler çeşitliliği ve kırmızı daireler ise komşuluk aramasında bulunan adımı ifade etmektedir. Siyah noktaların neredeyse tüm çözüm uzayına dağıldığı ve mavi daire - A ile işaretlenmiş alanda görüldüğü gibi komşuluk araması yapılarak global minimum noktasına yaklaşılmaktadır. Çeşitlilik sayesinde yeşil daire - B ile işaretlenmiş alana takılması yani yerel minimum tuzaklarından kaçınılması sağlanır.

Şekil 1.4'de sezgisel algoritmada çeşitlilik azalmıştır. Bu durumda çözüm uzayı tamamen gezinemeyip, algoritma yerel çözüm tuzaklara takılır. Sezgisel algoritmaların performansları keşif ve sömürü arasındaki dengeye bağlıdır ve bu dengelerin ince ayarlar ile iyileştirilmesi gerekmektedir.



Şekil 1.4

3.2. Çalışmanın amacı

Optimizasyon mühendislik tasarımından finansal piyasalara, bilgisayar bilimlerinden endüstriyel uygulamalara kadar çok geniş alanda uygulanmaktadır. Geliştirilen sezgisel optimizasyon algoritmaları ile kısa zamanda optimuma yakın sonuçlar elde edilmektedir. Algoritmalarda elde edilen sonuçları iyileştirmek için literatürde birçok çalışma yapılmıştır. Bu çalışmanın amacı , literatürde MSA algoritmaları üzerinde yapılan çalışmaları DSK altında toplayıp, daha sistematik bir çalışma ortamı hazırlayarak daha etkili sonuçların elde edilmesi hedeflenmiştir.

3.3. Araştırma sorusu ve/veya hipotez

Literatürdeki MSA algoritmaları üzerindeki iyileştirmeler incelendiğinde, çalışmaların birçoğu mevcut MSA tekniklerinin çeşitli yöntemlerle iyileştirilmesini ve varyasyonlarının geliştirilmesini konu almaktadır. Algoritmalarda arama performansları üzerinde etkili olan iki temel öğe 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.

4. Projede Kullanılan Yöntem ve Metotlar

4.1. 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 çözümlenmesi 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 iv’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 [10], seçim yöntemleri [11], arama operatörleri [12-13] ve arama stratejisidir [14-15]. Arama operatörleri ve arama stratejisi kontrol başlığı altında gruplandırılarak algoritmalar için etkili olan faktörleri üç ana başlık altında toplayıp DSK olarak adlandırılmıştır. Algoritmalar kendi içlerinde DSK yöntemine göre bölünecektir. Bu bölümler MSA algoritmaları için temel adımlar olan dağılım, seçim ve kontrol bölümlerinden oluşmaktadır.

4.1.1. 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 ve Kaos Haritalarıdır.

4.1.2. Kontrol Prosedürü

MSA algoritmalarında sezgisel arama sürecine 4 numaralı adıma eşlik eden algoritmalara 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.

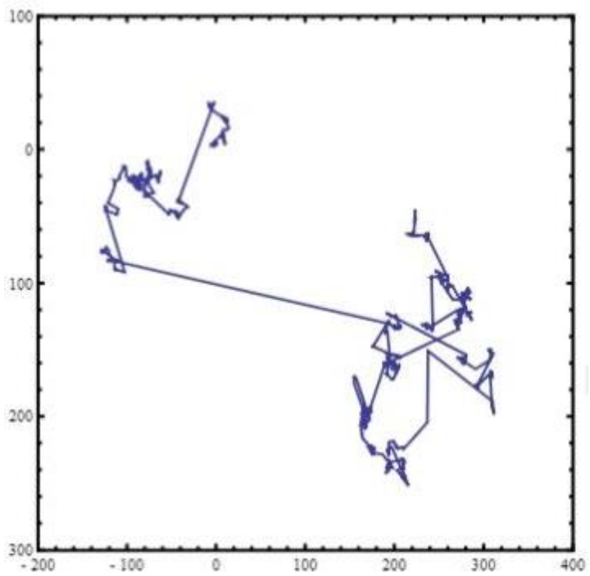
4.1.3. Seçim Prosedürü

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.

4.2. Tatbik Edilecek Teknikler

4.2.1. Levy Uçuşu

Randomizasyon, sürü zekasına dayalı optimizasyon tekniklerinde en önemli bileşendir. Sezgisel algoritmalarda randomizasyonu sağlayabilmek için düzgün dağılmış sayılar üreten rand fonksiyonu kullanılır. Bu fonksiyonun yerine ve MSA algoritmalarının performanslarını artıran Levy Uçuş Mekanizması kullanılmaktadır. Bu mekanizma α-kararlı dağılımına dayanan ve farklı ölçülerde adım büyüklüğü kullanarak büyük boyuttaki mesafelerde hareket etme kabiliyeti olan rastgele işlemlerdir. Levy α-kararlı dağılımı; ağır kuyruklu olasılık yoğunluk fonksiyonu, fraktal istatistik ve anormal dağınım ile son derece bağlantılıdır [7]. Şekil 1.5’te Levy Uçuşu’nun ilk 1000 adımdaki örneği [3]



Şekil 1.5

Randomizasyon işleminin Levy Uçuşu ile belirlenmesi şu şekildedir:

- Levy formülde 0.01 sayısı adım boyutunun kontrolü için kullanılmaktadır. $Levy(\beta) \sim 0.01 \frac{u}{|v|^{1/\beta}}$

- U ve N sayıları iki normal olasılıksal dağılımı ifade etmektedir $u \sim N\left(0, \sigma_u^2\right) \; v \sim N\left(0, \sigma_v^2\right)$

- σ_u formülünde kullanılan Γ ifadesi standart gamma fonksiyonudur $\sigma_u = \left\{ \frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma\left[(1+\beta)/2\right]\beta 2^{(\beta-1)/2}} \right\}^{1/\beta}, \quad \sigma_v = 1$

4.2.2. Kaos Haritaları

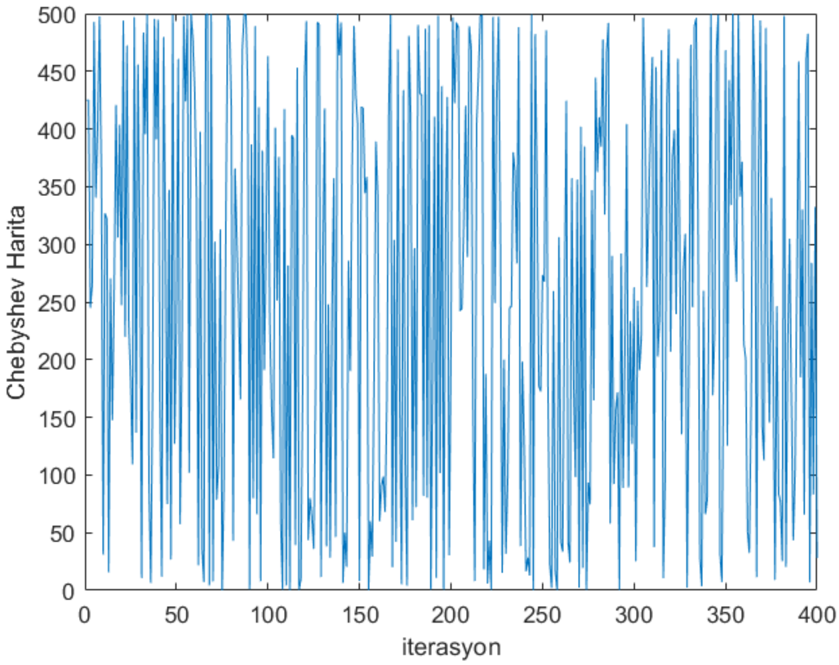
MSA algoritmalarında rastgele hareketi arttırmak için kullanılan Kaos Haritaları periyodik olmayan, yakınsamayan ve sınırlı olan, doğrusal olmayan dinamik sistemler bulunan deterministtik, rasgele benzeri bir süreçtir. Ayrıca başlangıç şartları ve parametrelerine oldukça bağlıdır. Kaosun doğası görünürde rasgele ve tahmin edilemezdir. Ayrıca kendi içerisinde bir düzene sahiptir [8].

MSA algoritmalarında iyileştirme yapmak için kullanılacak bazı Kaos Haritalarının grafikleri ve sözde kodları aşağıda verilmiştir.

4.2.2.1. Chebyshev Haritası

Chebyshev Haritası sözde kodu

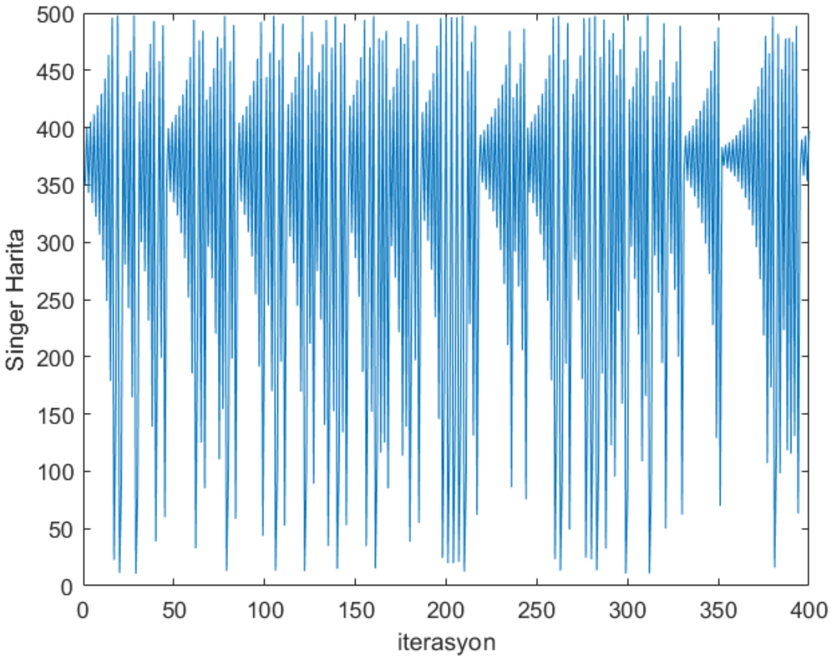
```
Value = 500; // Üretilecek en büyük sayı
X(1) = 0.7;
MaxGen = Maksimum iterasyon sayısı
for i:1= MaxGen
   x(i+1)=cos(i*acos(x(i)));
   KaosDeger(i)=((x(i)+1)*Value)/2;
end
```



4.2.2.2. Singer Haritası

Singer Haritası sözde kodu

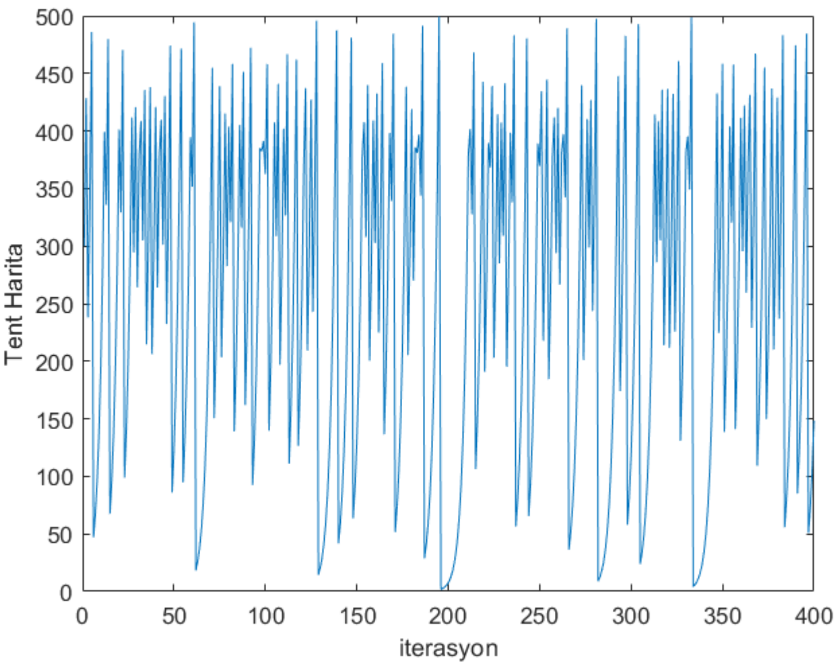
```
Value = 500; // Üretilecek en büyük sayı
X(1) = 0.7;
MaxGen = Maksimum iterasyon sayısı
u=1.07;
for i:1= MaxGen
   x(i+1) = u*(7.86*x(i)-23.31*(x(i)^2)+28.75*(x(i)^3)-13.302875*(x(i)^4));
   KaosDeger(i) = (x(i))*Value;
end
```



4.2.2.3. Tent Haritası

Tent Haritası sözde kodu

```
Value = 500; // Üretilecek en büyük sayı
X(1) = 0.6;
MaxGen = Maksimum iterasyon sayısı
for i:1= MaxGen
    if x(i)<0.7
       x(i+1)=x(i)/0.7;
    end
    if x(i)>=0.7
       x(i+1)=(10/3)*(1-x(i));
    end
   KaosDeger(i) = (x(i))*Value;
end
```



4.2.3. 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

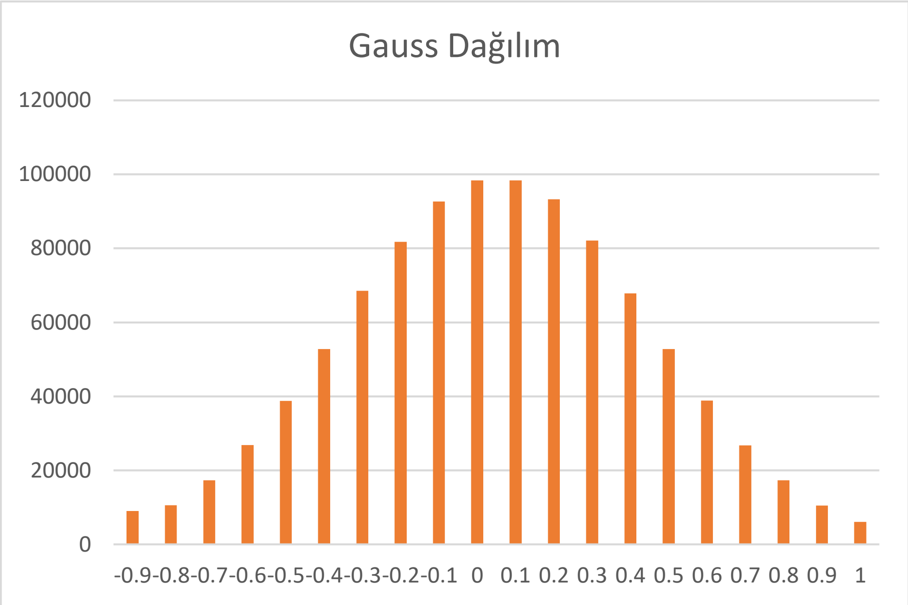
n = MSA algoritmasında çözüm adayı sayısı
m = Optimizasyon probleminin boyutu
$X_{[n,m]}$ = Çözüm adayları topluluğu,
$U_{[n]}$ = Çözüm adaylarının uygunluk değerleri
$R_{[n]}$ = Çözüm adaylarının rulet tekerleği yüzdeleri
$K_{[n]}$ = Çözüm adaylarının rulet tekerleği konumları,
t = 0, $K_{[0]}$ = 0;
    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

4.2.4. Gauss Dağılımı

Gauss, iki parametreye sahip ve sürekli dağılım sağlayan bir yöntemdir. Bu parametreler konum (değer) bilgisini temsil eden μ (aritmetik ortalama) ve varyans bilgisini temsil eden σ^2 yayılımdır [16-17]. Aritmetik ortalama parametresi çan eğrisinin tepe noktasını belirler. Normal dağılımın matematiksel ifadesi Eşitlik-1'de verilmektedir [10].

$$X_{[i]} = \mu + \sigma^2 * \left[\sqrt{-2 * \ln(\mathrm{rand})} * \sin(2 * \pi * \mathrm{rand})\right] \quad (1)$$

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 6'da gösterilmektedir [10].



Şekil 1.6

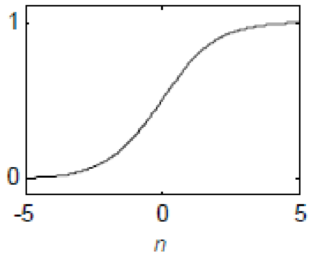
4.3. Tekniklerin Tatbik Edilme Yöntemi

Levy Uçuşu, Kaos Haritaları ve Rulet Tekerleğinin MSA algoritmalarında tatbik ederken hangi oranda ve nasıl uygulanacağının bulunması 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 1.7 de verilen Sigmoid 'ir.

Sigmoid

$$y = \frac{1}{1+e^{-net}}$$

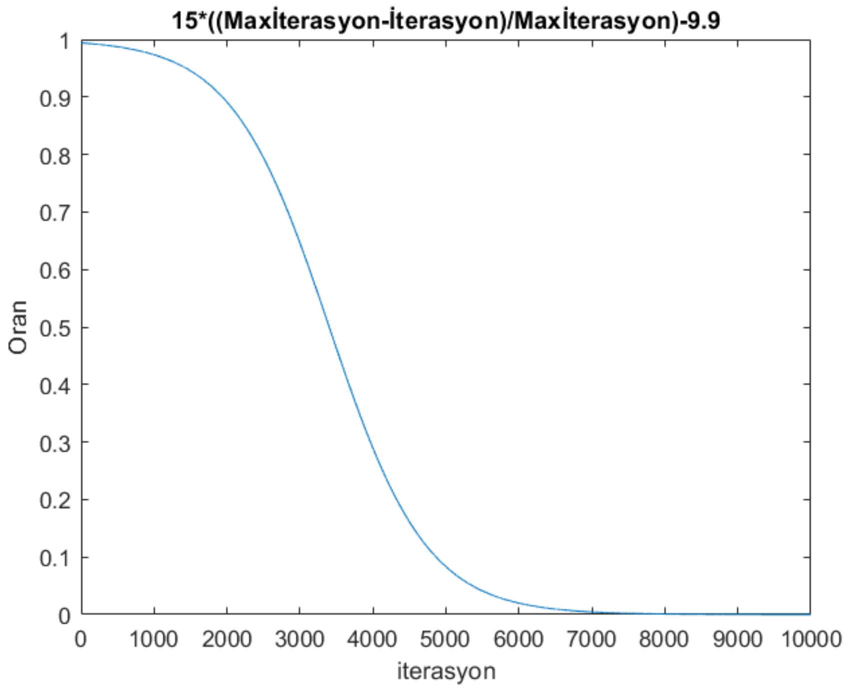


Şekil 1.7

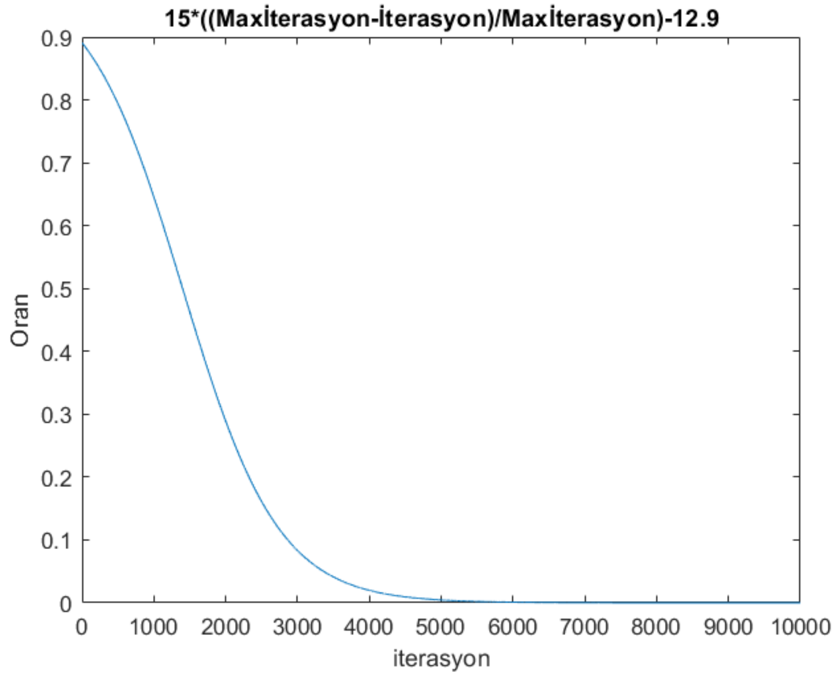
Sigmoid fonksiyonu yapay sinir ağlarında aktivasyon fonksiyonu olarak kullanılmaktadır. Verilen değişkeni farklı bir boyuta taşıyan doğrusal olmayan bir fonksiyondur. Sigmoid fonksiyonunda net değerin hesaplanması için önerilen formüller:

- Artan oranda Levy Uçuşu, Kaos Haritaları ve Rulet Tekerleğinin tatbik edilmesi için 15*(İterasyon/Maxİterasyon)-A
- Azalan oranda Levy Uçuşu, Kaos Haritaları ve Rulet Tekerleğinin tatbik edilmesi için 15*((Maxİterasyon-İterasyon)/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 1.8 ve Şekil 1.9 de görülmektedir.



Şekil 1.8



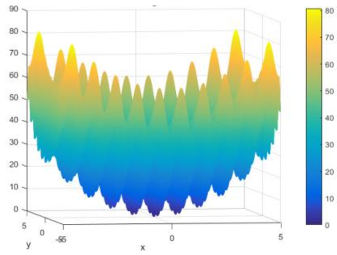
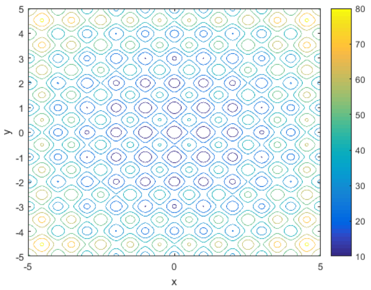
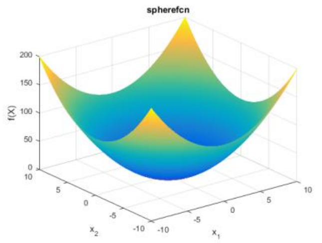
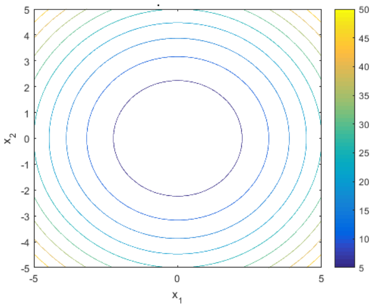
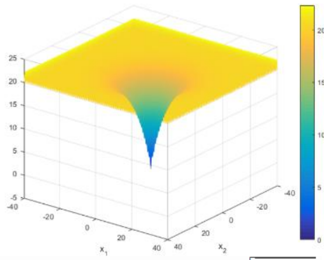
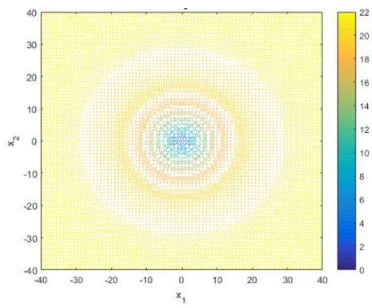
Şekil 1.9

4.4. Deneysel Çalışmalar

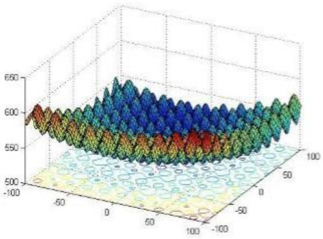
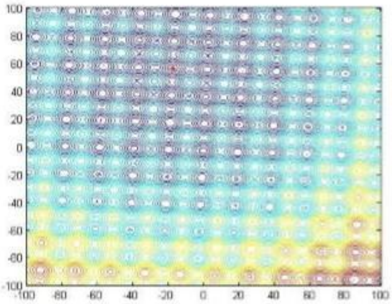
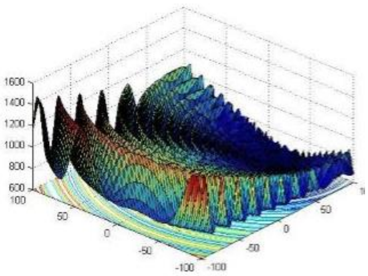
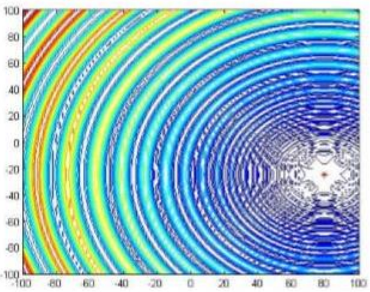
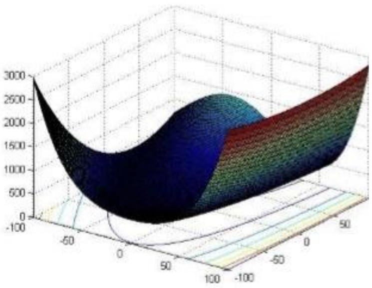
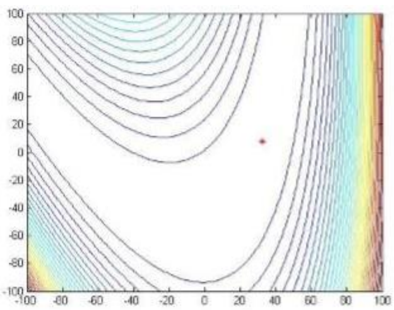
Literatürde sürekli değer optimizasyon problemleri tiplerine göre genellikle dört kategoriye ayrılmaktadır. Bunlar tek modlu, çok modlu, melez ve komposizyon (compostion) problemlerdir. Bu problem tipleri, 1994 yılından buyana her yıl düzenli bir şekilde gerçekleştirilen IEEE evrimsel hesaplama (IEEE CEC) konferanslarındaki çalışmaların da katkılarıyla ortaya çıkmıştır [22]. 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 [23-24]. Komşuluk arama yeteneği yüksek algoritmalar bu problem türünde başarılı olmaktadırlar. Çok modlu problemler ise yerel çözüm tuzakları barındıran problem türüdür. Örneğin Michalewicz test fonksiyonunda yerel minimum sayısı, problem boyutuna (n) bağlı olarak faktöriyeli (n!) ifadesiyle değişmektedir. Yerel çözüm tuzakları çok modlu problemlerin optimizasyonu zorlaştırmaktadır [25]. 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 [19, 20] ve 30'u CEC 2017 [21] olmak üzere toplam 90 test problemi üzerinde çalışılacaktır. CEC2017 ve CEC2014 'de sürekli değerli ve dinamik yapılı 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. Tablo 1'de MSA algoritmalarının geliştirildiği çalışmalarda en sık kullanılan test ve karşılaştırma problemleri[18], Tablo 2'de CEC2017 'den alınmış çok modlu, kaydırma ve döndürme işlevli test ve karşılaştırma problemlerinin birkaçı ve bu problemlere ait bilgiler verilmektedir.

Tablo 1. En Fazla Kullanılan Benchmark Problemlerinin Birkaçı ve Bu Problemlere Ait Bilgiler

Problem Adı	Grafiksel Gösterim ve Matematiksel İfade	Kountur Grafiği	Özellikleri
Rastrigin	 $f(x,y) = 10n + \sum_{i=1}^{n}(x_i^2 - 10cos(2\pi x_i))$		- Sürekli - Konveks - n-boyutlu uzayda tanımlanabilir - Çok modlu (birden fazla yerel min.) - Türevlenebilir - Ayrılabilir
Sphere	 $f(\mathbf{x}) = f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} x_i^2.$		- Sürekli - Konveks - n-boyutlu uzayda tanımlanabilir - Tek modlu - Türevlenebilir - Ayrılabilir - xi∈[−5.12,5.12],i=1..n
Ackley	 $f(\mathbf{x}) = f(x_1, \ldots, x_n) = -a.\,exp(-b\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - exp(\frac{1}{n}\sum_{i=1}^{n} cos(cx_i)) + a + exp(1)$		- Sürekli - Konveks değil - n-boyutlu uzayda tanımlanabilir - Çok modlu - Ayrılabilir - xi∈[−32,32], i=1..n

Tablo 2. CEC2017 Çok Modlu Problemlerinin Birkaçı ve Bu Problemlere Ait Bilgiler

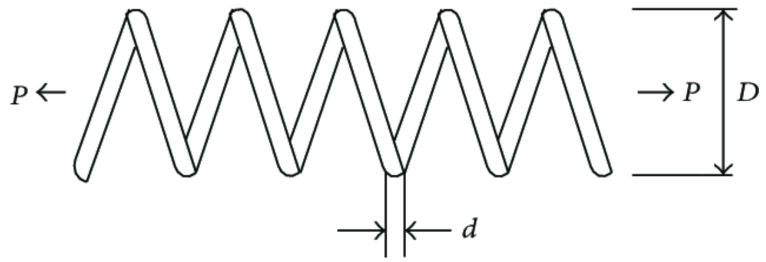
Problem Adı	Grafiksel Gösterim ve Matematiksel İfade	Kountur Grafiği	Özellikleri
Rastrigin	 $F_5(\boldsymbol{x}) = f_5(\mathbf{M}(\boldsymbol{x}-\boldsymbol{o}_5)) + F_5{}^*$ $f_5(\boldsymbol{x}) = \sum_{i=1}^{D}(x_i{}^2 - 10\cos(2\pi x_i) + 10)$		- Çok modlu (birden fazla yerel min.) - Ayrılmaz - İkinci daha iyi yerel optimum, küresel optimumdan uzaktır.
Schaffer	 $F_6(\boldsymbol{x}) = f_{20}(\mathbf{M}(\frac{0.5(\boldsymbol{x}-\boldsymbol{o}_6)}{100})) + F_6{}^*$ $f_{20}(\boldsymbol{x}) = \left[\frac{1}{D-1}\sum_{i=1}^{D-1}\left(\sqrt{s_i}\cdot\left(\sin\left(50.0s_i^{0.2}\right)+1\right)\right)\right]^2,\ s_i = \sqrt{x_i^2 + x_{i+1}^2}$		- Çok modlu (birden fazla yerel min.) - Ayrılmaz - Asimetrik
Rosenbrock	 $F_4(\boldsymbol{x}) = f_4(\mathbf{M}(\frac{2.048(\boldsymbol{x}-\boldsymbol{o}_4)}{100}) + 1) + F_4{}^*$ $f_4(\boldsymbol{x}) = \sum_{i=1}^{D-1}(100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$		- Çok modlu (birden fazla yerel min.) - Ayrılmaz

Çalışmalarda amaç fonksiyonu azami çağırma sayısı baz alınacaktır. Amaç fonksiyonu azami çağırma sayısı problemin boyutunun 10 bin katı olarak belirlenmiştir. Deneysel çalışma sonuçları istatistiksel olarak wilcoxon runk sum test ile analiz edilecektir.

Çalışmalarda iyileştirme yapılan algoritmalar, kısıtlı mühendislik tasarım problemlerinde tatbik edilecektir. Tatbik edilerek optimizasyonu yapılacak ve kısıtlı mühendislik tasarım problemlerinin tanımları aşağıda verilmiştir.

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 2.0 ‘da Gerginlik / Sıkıştırma Yayı Tasarım Sorunu [26] verilmiştir.



Şekil 2.0

Şekil 2.0, sorunun şematik gösterimini göstermektedir. Değişkenlerin aralıkları, amaç fonksiyonun matematiksel formülasyonu ve problemin kısıtlamaları aşağıdaki şekilde tanımlanır:

I. Parametreler

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

II. Formül

- Minimize $f(\vec{x}) = (x_3 + 2)x_2 x_1^2$,

III. Kısıtlar

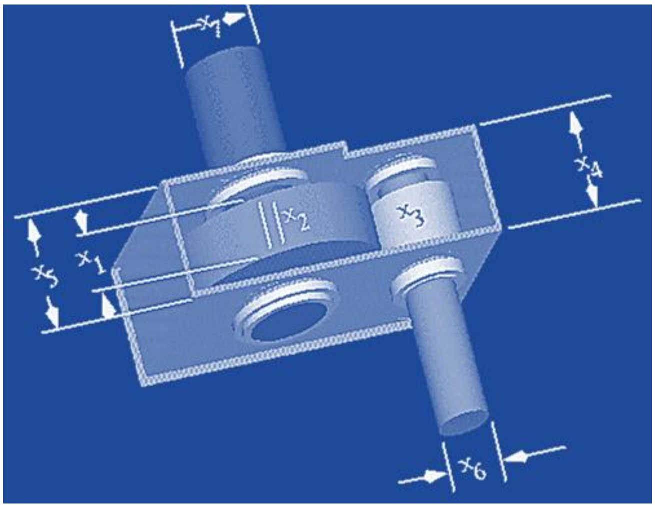
- $g_1(\vec{x}) = 1 - \frac{x_2^2 x_3}{71785 x_1^4} \leq 0$
- $g_2(\vec{x}) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} \leq 0$,
- $g_3(\vec{x}) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0$
- $g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0$

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. 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 2.1' de Hız Düşürücü Tasarım Problemi [27] verilmiştir.



Şekil 2.1

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

- $g_1(\vec{x}) = \frac{27}{x_1 x_2^2 x_3} - 1 \leq 0$

- $g_2(\vec{x}) = \frac{397.5}{x_1 x_2^2 x_3^2} - 1 \leq 0$

- $g_3(\vec{x}) = \frac{1.93 x_4^3}{x_2 x_6^4 x_3} - 1 \leq 0$

- $g_4(\vec{x}) = \frac{1.93 x_5^3}{x_2 x_7^4 x_3} - 1 \leq 0$

- $g_5(\vec{x}) = \frac{\sqrt{\left[\left(745(x_4/x_2 x_3)\right)^2 + 16.9*10^6\right]}}{110 x_6^3} - 1 \leq 0$

- $g_6(\vec{x}) = \frac{\sqrt{\left[\left(745(x_5/x_2 x_3)\right)^2 + 157.5*10^6\right]}}{85 x_7^3} - 1 \leq 0$

- $g_7(\vec{x}) = \frac{x_2 x_3}{40} - 1 \leq 0,$

- $g_8(\vec{x}) = \frac{5 x_2}{x_1} - 1 \leq 0$

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

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

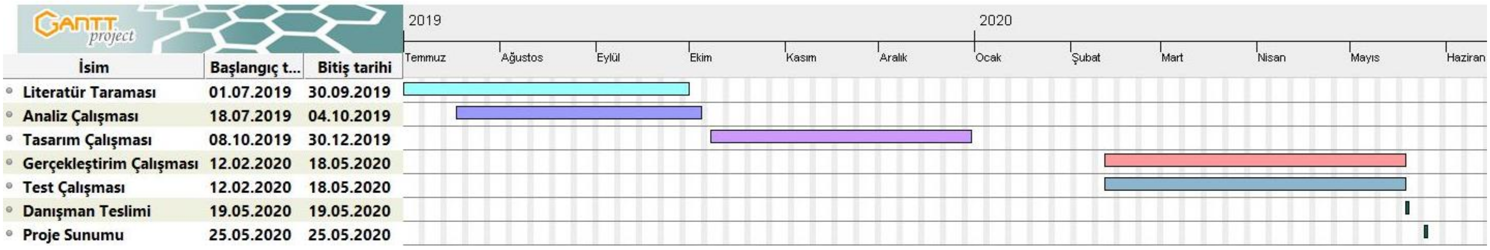
- $g_{11}(\vec{x}) = \frac{1.1 x_7 + 1.9}{x_5} - 1 \leq 0$

III. Parametre Çözüm Aralığı

- $2.6 \leq x_1 \leq 3.6$
- $0.7 \leq x_2 \leq 0.8$
- $17 \leq x_3 \leq 28$
- $7.3 \leq x_4 \leq 8.3$
- $7.3 \leq x_5 \leq 8.3$
- $2.9 \leq x_6 \leq 3.9$
- $5.0 \leq x_7 \leq 5.5$

5. Proje İş-Zaman Planı

Proje iş zaman çizelgesi, yöntem bölümünün takvime bağlanmasıyla hazırlanmıştır. Bu proje çalışmasının hazırlıkları danışmanımızın yönlendirmesiyle 2020 yılı mayıs ayına kadar uzanmaktadır. Dolayısıyla ilişkili çalışmalara yönelik literatür taraması temmuz ayından itibaren başlanmıştır. Temmuz – Eylül döneminde literatür taraması neticesinde elde edilen dokümanların analizi yapılarak, MSA algoritmaları üzerinde yapılmış iyileştirme çalışmaları ve bu çalışmalarda kullanılan yöntemlerin uygulanış teknikleri incelenmiştir. Ekim ayı itibariyle, belirlenen MSA algoritmaları DSK yöntemlerine uyun biçimde kendi içlerinde bölümlendirilerek uygulanabilecek yöntemler belirlenecektir. Hazırlanan MSA algoritmalarına belirlenen yöntemleri, metotlar kısmında belirlenmiş uygulanış teknikleri ile gerçekleştirim çalışmalarına başlanırken aynı zamanda test çalışmaları sürdürülecektir. Proje iş zaman planı Şekil 2.3 ‘ de verilmiştir.



Şekil 2.3

6. Sonuç

Klasik optimizasyon tekniklerinin yetersiz kalmasından dolayı geliştirilen doğa esinli MSA algoritmaları birçok alanda kullanılmaktadır. Yerel çözüm tuzaklarına takılabilen MSA algoritmalarında iyileştirme yapılarak performanslarını artırmak için literatürde yüzlerce çalışma yapılmıştır. Yapılan çalışmaların büyük çoğunluğu MSA algoritmalarının arama stratejisi ve arama operatörleri üzerinde yapılmaktadır.

Bu proje önerisinde MSA algoritmalarında yapılacak iyileştirme çalışmaları DSK yöntemine göre algoritmalar kendi içlerinde üç bölüme ayrılmıştır. Bu bölümler ve bu bölümlerde kullanılabilecek teknikler tanımlanmış ve tanıtılmıştır. DSK yöntemine göre bölünmüş güncel MSA algoritmaları belirlenen teknikler ile performansları iyileştirilerek optimizasyon problemlerinde maliyetler azaltılacaktır.

7. Kaynaklar

1. Haklı, H. (2013). Sürekli fonksiyonların optimizasyonu için doğa esinli algoritmaların geliştirilmesi (Yüksek Lisans Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü).
2. Katırcıoğlu, F. (2016). Yerçekimi arama algoritması için yeni operatörlerin geliştirilmesi (Doktora Tezi, Düzce Üniversitesi Fen Bilimleri Enstitüsü).
3. Gülcan, H. (2018). Yusufçuk algoritmasının brownian hareketi ile iyileştirilmesi (Yüksek Lisans Tezi, Mersin Üniversitesi Fen Bilimleri Enstitüsü).
4. Kazak, N. (2011). Geliştirilmiş yerçekimsel arama algoritması(Yüksek Lisans Tezi, Bilecik Üniversitesi, Fen Bilimleri Enstitüsü).
5. Mirjalili, S., & Gandomi, A. H. (2017). Chaotic gravitational constants for the gravitational search algorithm. Applied soft computing, 53, 407-419.

6. Cigal, T. (2018). Sürekli zamanlı kaotik sistem tabanlı balina optimizasyon algoritmasının geliştirilmesi (Yüksek Lisans Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü).
7. Hınıslıoğlu, Y. (2018). Kaotik güve sürü algoritması kullanarak rüzgâr gücü entegreli optimal güç akışı (Yüksek Lisans Tezi, Düzce Üniversitesi, Fen Bilimleri Enstitüsü).
8. Alataş, B., Akın, E., & Özer, A. B. (2007). Kaotik Haritalı Parçacık Sürü Optimizasyon Algoritmaları. Eskişehir Osmangazi Üniversitesi Elektrik Elektronik Bilgisayar Biyomedikal Mühendisliği Ulusal Kongresi.
9. Holland, J.H., 1975. "Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence". Q. Rev. Biol. 1, 211. http://dx.doi.org/10.1086/418447.
10. 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.
11. 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.
12. W. Zhao, L. Wang and Z. Zhang, Atom search optimization and its application to solve a hydrogeologic parameter estimation problem, Knowledge-Based Systems (2018), 163, 283-304.
13. 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.
14. Tian, M., & Gao, X. (2019). Differential evolution with neighborhood-based adaptive evolution mechanism for numerical optimization. Information Sciences, 478, 422-448.
15. Draa, A., Chettah, K., & Talbi, H. (2018). A Compound Sinusoidal Differential Evolution algorithm for continuous optimization. Swarm and Evolutionary Computation.
16. N. Higashi, H. Iba, Particle swarm optimization with gaussian mutation, in: Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE, 72–79.
17. 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).
18. Kahraman H.T., Aras S., Gedikli E., "META-SEZGİSEL OPTİMİZASYON ÇALIŞMALARINDA BENCHMARK PROBLEMLERİNDE KARŞILAŞILAN STANDARTSIZLIKLAR VE ÇÖZÜM ÖNERİLERİ", INTERNATIONAL ACADEMIC RESEARCH CONGRESS, ANTALYA, TÜRKİYE, 30 Ekim – 3Aralık 2018, pp. 1494-1501.
19. Xiang, W. L., Meng, X. L., Li, Y. Z., He, R. C., An, M. Q. 2018. "An improved artificial bee colony algorithm based on the gravity model", Information Sciences, 429, 49-71.
20. J. J. Liang, B-Y. Qu, P. N. Suganthan, Alfredo G. Hernández-Díaz, "Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session and Competition on Real-Parameter Optimization", Technical Report 201212, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, January 2013.
21. 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.
22. 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 Nanyang Technological University, Singapore.
23. 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.

24. Tang, D., Liu, Z., Yang, J., & Zhao, J. (2018). Memetic frog leaping algorithm for global optimization. Soft Computing, 1-29.
25. 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.
26. 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).
27. 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).