T.C.

KONYA TEKNİK ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YAPAY ALG ALGORİTMASININ KISITLI OPTİMİZASYON PROBLEMLERİNİN ÇÖZÜMÜ İÇİN GELİŞTİRİLMESİ

SEDA YILDIZ

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Ocak-2019

KONYA

TEZ KABUL VE ONAYI

Seda YILDIZ tarafından hazırlanan "Yapay Alg Algoritmasının Kısıtlı Optimizasyon Problemlerinin Çözümü için Geliştirilmesi " adlı tez çalışması 08/01/2019 tarihinde aşağıdaki jüri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri	**İmza**
Başkan Doç.Dr. Mustafa Servet KIRAN	....
Danışman Dr.Öğr.Üyesi Sait Ali UYMAZ	....
Üye Dr.Öğr.Üyesi Mehmet HACIBEYOĞLU	....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Yakup KARA
Enstitü Müdürü

TEZ BİLDİRİMİ

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

DECLARATION PAGE

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

Seda YILDIZ

Tarih: 08.01.2019

ÖZET

YÜKSEK LİSANS TEZİ

YAPAY ALG ALGORİTMASININ KISITLI OPTİMİZASYON PROBLEMLERİNİN ÇÖZÜMÜ İÇİN GELİŞTİRİLMESİ

Seda YILDIZ

Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr.Öğr.Üyesi Sait Ali UYMAZ

2019, 51 Sayfa

Jüri
Dr. Öğr.Üyesi Sait Ali UYMAZ
Doç.Dr. Mustafa Servet KIRAN
Dr. Öğr.Üyesi Mehmet HACIBEYOĞLU

Optimizasyon belirli koşullar altında en iyi çözümü bulma işidir. Gerçek dünyada var olan optimizasyon problemlerinin bir çoğu, kısıtlara sahiptir. Kısıtlar, arama uzayını uygulanabilir ve uygulanabilir olmayan alanlar olarak ayırmaktadır. Bu tip problemlerin en zorlu kısmı, kısıtları işleme süreçleridir. Var olan bir çok metasezgisel optimizasyon algoritmalarının orijinali, kısıtsız problemler için tasarlanmıştır. Kısıtları işleme yöntemleri bu algoritmalara uygun çözümlerin bulunduğu bölgelerde aramaya kılavuzluk etmesi amacı ile eklenen metotlardır. Yapay Alg Algoritması(AAA) mikro alglerin yaşam davranışlarından esinlenilerek ortaya konmuş metasezgisel bir optimizasyon algoritmasıdır. AAA kısıtsız problemlerde başarısını ortaya koymuştur ancak kısıtlı problemleri çözmesi için bir versiyonu bulunmamaktadır. AAA üzerinde A_p parametresi ve popülasyon sayısı değişiminin etkisini gözlemlemek için, mühendislik tasarım optimizasyon problemleri üzerinde testler yapılmıştır. AAA üzerine kısıt işleme yöntemlerinden Deb’s Rule, dinamik penaltı ve ϵ-kısıt işleme tekniği uygulanarak, kısıtlı optimizasyon problemlerini çözebilen AAA_{dr}, AAA_{dp} ve AAA_{ε} algoritmaları önerilmiştir. Önerilen algoritmaların performansı, kısıtlı fonksiyon setinde test edilmiştir. AAA_{dr}, AAA_{dp} ve AAA_{ε} arasında kıyaslama yapılmış ve öne çıkan AAA_{dr} literatürdeki iyi bilinen diğer kısıtlı problemler için uyarlanmış algoritmalar ile kıyaslanmıştır. Yapılan çalışmalar sonunda AAA_{dr}’nin rakebetçi sonuçlar ortaya koyduğu görülmüştür.

Anahtar Kelimeler: Kısıt işleme yöntemleri, Metasezgisel Algoritmalar, Optimizasyon, Yapay Alg Algoritması

ABSTRACT

MS THESIS

IMPROVING ARTIFICIAL ALGAE ALGORITHM FOR SOLUTION OF CONSTRAINED OPTIMIZATION PROBLEMS

Seda YILDIZ

Konya Technical University
Institute of Graduate Studies
Department of Computer Engineering

Advisor: Asst.Prof.Dr. Sait Ali UYMAZ

2019, 51 Pages

Jury
Asst.Prof.Dr. Sait Ali UYMAZ
Assoc.Prof.Dr. Mustafa Servet KIRAN
Asst.Prof.Dr. Mehmet HACIBEYOĞLU

Optimization is the job of finding the best solution under certain conditions. Many of the optimization problems that exist in the real world have constraints. Constraints allocate the search space as applicable and non-applicable areas. The most challenging part of such problems is the processing of constraints. The original of many existing metaheuristic optimization algorithms is designed for unconstrained problems. The methods of handling constraints are methods that are added to guide the search in areas where these algorithms are appropriate. Artificial Algae Algorithm (AAA) is a metaheuristic optimization algorithm which is inspired by the life behaviors of micro algae. The AAA has demonstrated its success in unconstrained problems, but there are no versions of it to solve constraints problems. Tests on engineering design optimization problems were performed to observe the effect of Ap parameter and population number change on AAA. AAA_{dr}, AAA_{dp} and AAA_{ϵ} algorithms, which can solve the constrained optimization problems by applying Deb's Rule, dynamic penalty and ϵ-constraint handling technique on AAA, have been proposed. The performance of the proposed algorithms has been tested in a restricted set of functions. AAA_{dr}, AAA_{dp} and AAA_{ϵ} were compared and the prominent AAA_{dr} was compared with the algorithms adapted for other well-known limited problems in the literature. The studies revealed, AAA_{dr} seem to produce competitive results.

Keywords: Constrained Handling Techniques, Metaheuristic Algorithms, Optimization, Artificial Algae Algorithm

ÖNSÖZ

Bu çalışmayı hazırlarken geçirdiğim süreçte benden yardımlarını esirgemeyen danışmanım Dr. Öğr. Üyesi Sait Ali UYMAZ'a ve çalışmalarım sırasında yardım ve desteklerini esirgemeyen Selçuk Üniversitesi Bilgi İşlem Daire Başkanlığı'nda çalışan iş arkadaşlarıma saygı ve teşekkürlerimi sunarım.

Seda YILDIZ
KONYA-2019

İÇİNDEKİLER

ÖZET .... **iv**

ABSTRACT .... **v**

ÖNSÖZ .... **vi**

İÇİNDEKİLER .... **vii**

SİMGELER VE KISALTMALAR .... **ix**

1. GİRİŞ **1**

2. KAYNAK ARAŞTIRMASI **3**

3. MATERYAL VE YÖNTEM **8**

3.1. Optimizasyon 8
3.2. Kısıtlı Optimizasyon 8
3.3. Kısıt İşleme Yöntemleri 9
3.3.1.Penaltı Fonksiyonları 11
3.3.2.Decoder Yöntemi 12
3.3.3.Özel Operatörler 13
3.3.4. Amaç Fonksiyonlarının ve Kısıtların Ayrılması 13
3.3.5. Uygulanabilirlik Kuralları 14
3.3.6. Stokastik Sıralama 14
3.3.7. ϵ-kısıt İşleme 15
3.3.8. Yeni Penaltı Fonksiyonları 15
3.3.9. Yeni Özel Operatörler 15
3.3.10. Çok Amaçlı Kavramlar 16
3.3.11. Kısıt İşleme Tekniklerinin Birleştirilmesi 16
3.3.12. Kısıtlama Yaklaşımı 16
3.3.13. Dinamik Kısıtlar 17
3.3.14. Teori 17
3.4. Yapay Alg Algoritması(AAA) 17
3.4.1. Helisel Hareket 17
3.4.2.Adaptasyon 18
3.4.3.Evrimsel Süreç 19
3.4.4.AAA Sözde Kodu 19
3.5.Kısıtlı Problemler için AAA 19
3.5.1.Deb’s Rule Yöntemi Kullanılarak Kısıtlı Problemler için AAA (AAA_{dr}) 20
3.5.2.Dinamik Penaltı Yöntemi Kullanılarak Kısıtlı Problemler için AAA (AAA_{dp}) 21
3.5.3. ε-Kısıt İşleme Yöntemi Kullanılarak Kısıtlı Problemler için AAA (AAA_{ε}) 21
3.6. Algoritmanın Test Edilmesi 23
3.6.1.Mühendislik Tasarım Optimizasyon Problemleri 23
3.6.2.Kısıtlı Optimizasyon Fonksiyon Seti 28

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA **30**

4.1. Mühendislik Tasarım Optimizasyon Problemlerinde Performans Değerlendirmesi 30
4.2. Kısıtlı Optimizasyon Fonksiyon Setinde Performans Değerlendirmesi 34

5. SONUÇLAR VE ÖNERİLER 42

5.1 Sonuçlar 42
5.2 Öneriler 43

KAYNAKLAR 44

EKLER 48

ÖZGEÇMİŞ 51

SİMGELER VE KISALTMALAR

Simgeler

A_p	: AAA'da adaptasyon parametresi
E	:AAA'da enerji kaybı parametresi
F	:Amaç Fonksiyonu
G	:Eşitsizlik kısıt fonksiyonu
H	:Eşitlik kısıt fonksiyonu

Kısaltmalar

AAA	:Yapay alg algoritması (Artifical Algae Algorithm)
ABC	:Yapay arı kolonisi(Artifical Bee Colony Algorithm)
PSO	:Parçacık sürü optimizasyonu algoritması(Particle Swarm Optimization Algorithm)
GA	:Genetik algoritma(Genetic algorithm)
DE	:Diferansiyel gelişim algoritması(Differential evolution algorithm)
CTSA	:Kısıtlı problemler için ağaç-tohum algoritması(Constraint Tree-Seed algorithm)
AAA_{dr}	:Kısıtlı problemler için Deb's Rule yöntemi kullanılarak AAA
AAA_{dp}	:Kısıtlı problemler için Dinamik Penaltı yöntemi kullanılarak AAA
AAA_{ε}	:Kısıtlı problemler için ε-kısıt işleme yöntemi kullanılarak AAA

1. GİRİŞ

Optimizasyon beklenen şartlar altında, kısıtlı kaynaklar ile en verimli sonucu üretme işidir. Üretilen bu sonuç için problemlerin karmaşıklığından dolayı tüm olası kombinasyonların tahmini zordur. Matematiksel model yaklaşımları bu problemleri çözmek için problemleri basitleştirir. Aynı zamanda sonuçlar için varsayımlar yaparak tüm olası kombinasyonlar yerine arama uzayını küçültürler. Basitleştirilen ve sonuçları sınırlandırılan bu problemin çözümü ile gerçek problemin çözümü arasında önemli ölçüde fark ortaya çıkabilir. Matematiksel modellerin bu dezavantajını ortadan kaldırmak için sezgisel yöntemler kullanılmaktadır. Sezgisel yöntemler çözüm uzayını sınırlandırmak yerine detaylı bir arama yaparak daha kaliteli çözümler sunmaktadır (Yang, 2010).Sezgisel yöntemler her zaman en iyi durumu garanti etmez. Ancak yaklaşık hesap yaklaşımı olarak adlandırılırlar ve buradaki amaç kabul edilebilir uygun çözümleri bulmaktır.

Sezgisel yöntemlerin farklı problemlere uyarlanması zor olduğundan araştırmacılar tarafından metasezgisel ifadesi ortaya atılmıştır. Metasezgisel kavramı kaliteli çözümler bulan, sezgisel kavramını da kapsayan ve problemlere uygulanması kolay olan bir yöntemdir (Uymaz ve ark., 2015a).

Sezgisel ve metasezgisel algoritmalar ile yapılan çalışmalarda, optimizasyon problemleri için en uygun ve en kaliteli çözümün bulunması hedeflenmiştir. Ancak metasezgisel algoritmaların orijinal versiyonları, genellikle kısıtsız arama uzayında arama yapmak için tasarlanmıştır. Gerçek dünya optimizasyon problemlerinin birçoğu, karar değişkenleri üzerine tanımlanan kısıtlamalar içermektedir. Kısıtlı problemlerin temelini oluşturan, amaç fonksiyonunun istenen değerini bulmayı sağlayan asıl etmenler, probleme ait kısıtlardır. Karar değişkenlerinde tanımlanan kısıtlamalar, problem için en uygun noktayı değiştirebilecek durumdadır. Bu kısıtların gerektiği şekilde sağlanması gerekirken bu tip problemler için en zor ve önemli nokta kısıtların işlenmesidir. Kısıtların işlenmesi aşamasında sayısal yöntemlerin yetersiz kaldığı durumlarda kısıt işleme yöntemlerine başvurulmuş ve bu konu geniş bir araştırma konusu olmuştur. Bu yöntemlerin amacı, arama uzayında kısıtlara takılan çözümler yerine kısıtlara takılmayan bölgelerde arama sağlamak ve algoritmanın uygun bölgelerde çözümler bularak performansını arttırmaktır.

Kısıt işleme yöntemleri, literatürde sıklıkla kullanılan ve popüler hale gelmiş bir araştırma konusudur. Bulunan yöntemler çalışmalarda farklı şekillerde sınıflandırılmış ve her bir yöntem için avantaj ve dezavantajlar sunulmuştur.

Bu çalışmada Mikro alglerin yaşamlarından esinlenilerek ortaya konmuş metasezgisel bir algoritma olan AAA, kısıtlı optimizasyon problemleri için uygun hale getirilmiştir. Diğer metasezgisel algoritmalarda olduğu gibi, kısıtsız arama uzayı için tasarlanan bu algoritmanın arama uzayında kendisine rehberlik etmesi için kısıt işleme yöntemleri kullanılmıştır.

Tezin içeriği aşağıdaki gibidir:

2.bölümde kısıtlı optimizasyon problemlerine dair yapılan çalışmalar kronolojik olarak verilmiştir. Metasezgisel algoritmalara uyarlanması, kullanılan kısıt işleme yöntemleri ve bu yöntemlerin avantaj, dezavantajları üzerine çalışılmıştır.

3.bölümde optimizasyon kavramından ve kısıtlı optimizasyon kavramından bahsedilmiştir. Kısıtlı Optimizasyon problemleri için kullanılan kısıt işleme yöntemleri sunulmuştur. Devamında, AAA tanıtılmıştır. Kısıtlı optimizasyon problemleri için uyarlanmış Yapay Alg algoritmaları tanıtılmıştır. Önerilen algoritmaların performansını ortaya koymak için kullanılan test fonksiyonları belirtilmiştir.

Yapılan çalışmaya ait performans testleri, sonuçlar ve analizler bölüm 4’te bulunmaktadır.

5.Bölümde, yapılan çalışmaya ait sonuçlar değerlendirilmiş ve gelecek çalışmalar için öneriler sunulmuştur.

2. KAYNAK ARAŞTIRMASI

Kısıt işleme teknikleri üzerine yapılan çalışmalar yeni bir araştırma alanı olarak popüler hale gelmiştir ve günümüzde popülaritesini korumaktadır. Kısıt işleme yöntemleri ile ilgili çalışmalar ve araştırmalar bu bölümde sunulmuştur.

Schoenauer and Xanthakis (1993) ‘ın yaptığı çalışmada, kısıtları tek tek ele alan yeni bir yöntem önermişlerdir. Başlangıçta popülasyon rastgele oluşturulur ve algoritma ilk kısıtı işlemeye çalışır. İlk kısıtlama yerine getirildiğinde, algoritma sıradaki kısıtlar ile ilgilenir. Bu işlem tüm kısıtlar sağlanıncaya kadar devam eder. Problemden bağımsız bir yöntem olduğu ve hesaplanabilir kısıtların kaldırılabileceğini göstermektedir (Schoenauer ve Xanthakis, 1993). Ancak yapılan çalışmalarda (Michalewicz, 1995) bu yöntemin başarılı bulunmadığı ve probleme bağlı olduğu ortaya konmuştur.

Kısıt işleme yöntemleri ile ilgili yapılan ilk çalışmalardan olan ve detaylı bir içerik sunulan 1995 yılındaki çalışmada, uygun olmayan çözümler ile başa çıkmak için kullanılan yöntemler gözden geçirilerek, bu yöntemlerin avantaj ve dezavantajları sunulmuştur (Michalewicz, 1995).

1996 yılında yapılan çalışmada, evrimsel hesaplama yöntemleri ile kısıtlı optimizasyon hakkında genel bilgiler sunulmuştur (Michalewicz ve Schoenauer, 1996).

Kısıtlı optimizasyon problemleri için kendinden uyarlamalı ceza yöntemi sunulmuştur. Bu yaklaşım yalnızca toplam kısıt ihlallerini değil, aynı zamanda karşılanamayan kısıtların sayısını da kullanmaktadır. Arama işlemi sırasında farklı bir popülasyon ile optimize edilen ağırlıklandırma faktörleri, kontrol parametresi olarak kullanılır. Bu çalışmada sadece eşitsizlik kısıtları ele alınmıştır. Önerilen yöntemin performansı mühendislik optimizasyon problemleri üzerinde ve kısıtlı fonksiyon seti üzerinde test edilmiştir (Coello Coello, 2000).

Kısıt işleme yöntemlerinde sıklıkla kullanılan penaltı fonksiyonlarının dezavantajlarından kurtulmak amacıyla 2000 yılında penaltı fonksiyonlarına stokastik bir yaklaşım getirilmiştir. Bu yöntem penaltı faktörünün tanımı yerine, uygun olmayan çözümlerin kıyaslanması için kullanıcı tanımlı kontrol parametresini kullanmaktadır. Yeni bir kısıt işleme yöntemi olarak sunulan stokastik sıralama yöntemi iyi bilinen 13 test fonksiyonu üzerinde değerlendirilmiştir (Runarsson ve Yao, 2000).

Kısıtlı optimizasyon problemleri üzerinde kullanılan penaltı fonksiyonlarının tatmin etmeyen sonuçlarına karşılık, Deb (2000) tarafından turnuva seçim yöntemine üç uygunluk kuralı ekleyerek Deb’s Rule yöntemi geliştirilmiştir. Çalışmada yapılan testler sonucunda, yöntemin başarısı ortaya konmuştur (Deb, 2000).

Coello (2002) tarafından yapılan çalışmada evrimsel algoritmalarla kullanılan en popüler kısıt işleme teknikleri hakkında detaylı bir araştırma sunulmuştur. Bu çalışmada kısıt işleme yöntemleri 5 sınıfa ayrılmıştır. Bunlar; Penaltı Fonksiyonları, özel temsiller ve operatörler, onarım algoritmaları, amaç fonksiyonu ve kısıtların ayrılması, hibrit yöntemler olarak tanımlanmaktadır. Aynı zamanda yöntemler için yapılan çalışmalar göz önüne alınarak avantaj ve dezavantajları da sunulmuştur (Coello, 2002).

ASCHEA yöntemi, popülasyon temelli bir uyarlanabilir ceza işlevi olan, uygulabilir çözümlerin uygun olmayan çözümler ile eşleştiği ve uygunluk kurallarına benzer şekilde seçim yapılan adaptif bir optimizasyon yöntemidir. Yöntem 11 kısıtlı optimizasyon test fonksiyon seti üzerinde test edilmiş ve performansı ortaya konmuştur (Hamida ve Schoenauer, 2002).

Genetik algoritma genellikle kısıtsız optimizasyon problemlerini çözmek için kullanılan bir optimizasyon yöntemidir. Genetik algoritmaların kısıtlı optimizasyon problemlerine uygulanması zordur. Yeniay (2005) tarafından yapılan çalışmada, kısıtların ele alınmasına kullanılan yöntemlerden penaltı yöntemi incelenmiş, avantaj ve dezavantajları sunulmuştur (Yeniay, 2005).

2009 yılında yapılan bu çalışmada kısıtlı optimizasyon problemleri için yapay arı algoritması temel alınarak yeni bir sürü istihbarat yaklaşımı sunulmuştur. Scout-behavior modified Artifical Bee Colony (SB-ABC) olarak adlandırılan bu algoritmada, izci arıların davranışı mevcut en iyi çözümün çevresini kullanma yeteneğini elde edecek şekilde değiştirilmiştir. Rastgele besin kaynağı üretmek yerine, keşif arısı şimdiye kadar en iyi besin kaynağından ve rastgele bir besin kaynağından etkilenmiş şekilde üretilmiştir. Bu sayede izci arı, uygun çözümlerin bulunmasına yardımcı olacaktır. Bu algoritma kısıtlı optimizasyon problemlerinde ABC algoritmasının performansını geliştirmeye çalışmaktadır. İyi bilinen 13 test fonksiyonu üzerinde ABC ve SB-ABC algoritmalarının karşılaştırılması yapılmıştır. Aynı zamanda SB-ABC algoritmasının performansı diğer biyolojik ilhamlı algoritmalarla da karşılaştırılmıştır (Mezura-Montes ve Cetina-Domínguez, 2009).

2009 yılında kısıtlı optimizasyon problemleri için Evrimsel Algoritma kullanılarak yapılan çalışmada Deb’s Rule yöntemi ve Adaptif Penaltı yöntemlerinin en iyi özellikleri kullanılarak yenilikçi bir yaklaşım sunulmuştur. Önerilen hibrit yöntemde iki ayrı popülasyon kullanılır ve bu iki popülasyon birbiriyle bazı çözümleri paylaşır. İki yöntem birbirinin eksikliklerini karşılamaktadır. Adaptif Penaltı yöntemi ile daha iyi bir keşif avantajı sağlanırken, optimum çözümün uygunluğu Deb's Rule yöntemi ile sağlanmaktadır (Mani ve Patvardhan, 2009).

Kısıt işleme yöntemlerinde, amaç kısıtsız problemler için tasarlanmış algoritmaların kısıtlı problemler için de uygun hale gelmesini sağlayan yöntemlerdir. Arama noktalarını amaç fonksiyonlarına ve kısıt ihlallerine göre ele alır. 2010 yılında yapılan bu çalışmada, kısıtlı yöntem ve diferansiyel gelişim algoritmasının birleşimi olan ε sınırlı diferansiyel gelişim algoritması (εDE) önerilmiştir. Önerilen algoritmanın hızı ve başarısı ortaya konmuştur. Aynı zamanda gradyan ve arşiv tabanlı diferansiyel gelişim algoritması ile birleştirilerek önerilen algoritma ile çeşitliliği arttırmak hedeflenmiştir. Algoritmanın başarısı CEC2010 Kısıtlı optimizasyon problemleri ile test edilerek ortaya konmuştur (Takahama ve Sakai, 2010).

Mezura-Montes ve Coello (2011) tarafından yapılan çalışmada, doğa esinli algoritmaların orijinalinde sayısal bir problemin kısıtlarıyla başa çıkmak için kullanılan kısıt işleme yöntemlerini ilk yıl çalışmaları, günümüz çalışmaları ve gelecek eğilimler olarak üç temel sınıfa ayrılmıştır. Temel yöntemlerin ilk yıl çalışmalarında bulunduğu sınıflandırma sisteminde, günümüz çalışmaları genel olarak yöntemler üzerine yapılan iyileştirmeleri ve yenilikleri kapsamaktadır. Gelecek eğilimlerde ise, üzerinde çalışmaların yeterli olmadığı düşünülen ve önerilen yöntemler bulunmaktadır (Mezura-Montes ve Coello, 2011).

Karaboga ve Akay (2011) 'ın çalışmasında kısıtlanmamış optimizasyon problemleri için önerilmiş olan Yapay Arı Kolonisi (ABC) algoritması üzerinde modifikasyon yapılarak kısıtlı optimizasyon problemleri için uygun hale getirilmiştir. Kısıt işleme yöntemlerinden Deb's Rule yöntemi seçilerek algoritmaya eklenmiş ve iyi bilenen 13 test fonksiyonu üzerinde testler yapılmıştır. Yapılan deneysel çalışmalar sonucunda, değişiklik yapılarak sunulan ABC algoritmasının kısıtlı optimizasyon problemleri üzerinde performansının iyi olduğu gözlemlenmiştir (Karaboga ve Akay, 2011).

2014 yılında yapılan çalışmada; çok amaçlı genetik algoritma için yeni bir kısıt işleme tekniği önerilmiştir. Önerilen yeni yöntemde her bir aşamadan sonra gelecek nesile miras kalan çözümlerin uygun çözümler olması sağlanır. Çalışmada yapılan deneyler sonucunda önerilen yöntemin başarısı daha önce yapılan çok amaçlı algoritmalar üzerindeki kısıt işleme yöntemleri ile karşılaştırılarak sunulmuştur (Long, 2014).

2014 yılında sosyal örümceklerin işbirlikçi davranışlarından esinlenilerek kısıtlı optimizasyon problemleri için yeni bir algoritma olan Social spider optimization(SSO-C) algoritması önerilmiştir. Algoritmaya eklenen iki yeni özellik ile uygun olmayan çözümleri etkisiz hale getirmek hedeflenmiştir. İlk olarak kısıt ihlallerini belirlemek amaçlı bir ceza yöntemi ve ikinci olarak yeni bireylerin uygun bölgeye yönelmesini sağlayarak daha iyi çözümler üretmektir. Yapılan deneysel çalışmalar sonucunda algoritmanın etkinliği, verimliliği ve kararlılığı ortaya konmuştur (Cuevas ve Cienfuegos, 2014).

İhlal kısıt işleme yöntemi(VCH) ,GA üzerinde önerilen ihlal faktörünü kullanan bir kısıt işleme tekniğidir. Çalışmada literatürdeki penaltı kısıt işleme teknikleri ile karşılaştırılmıştır. VCH parametre içermeyen ve uygunluk kuralları yöntemine benzeyen bir yöntemdir (Chehouri ve ark., 2016).

Kısıtlı optimizasyon problemlerine uyarlanan bir diğer yöntem ise Box Complex metot olmuştur. Yapılan çalışmada en uygun popülasyon üyelerinden oluşturulan bir alanda gelen yeni üyenin durumuna göre kötü üyelerin Box alanının merkezine doğru hareketini temel alır. Yöntem iyi bir yerel yakınsama özelliğine sahipken, aramanın kötü sömürüsüne maruz kalmaktadır (Zade ve ark., 2017).

UDE algoritmasında Evrimsel Algoritma üzerinde yapılan çalışmalardan esinlenilerek literatürdeki çoğu Evrimsel Algoritma çeşitlerini aksine nesil değiştirme stratejisi kullanan kısıtlı optimizasyon problemlerinde başarı göstermiştir. Kısıt işleme yöntemi olarak statik penaltı yöntemi kullanılmaktadır (Trivedi ve ark., 2017).

Kısıtlı bir Evrimsel Algoritma iki bileşene sahiptir. Bunlardan ilki Arama algoritması ve bir diğeri kısıt işleme teknikleridir. Diferansiyel Gelişim algoritması yapısı basit olduğundan, az kontrol parametresi içerdiğinden ve CEC fonksiyonları üzerindeki başarısı gözlemlendiğinden bu çalışmada arama algoritması olarak seçilmiştir. Diferansiyel Gelişim Algoritma üzerinde yapılan bir çalışmayı (Wang ve ark., 2011) temel almaktadır. Kısıt işleme yöntemlerinden ϵ-kısıt işleme tekniği ve uygulanabilirlik kuralları yöntemini kullanmaktadır. Kısıt işleme teknikleri, Diferansiyel Gelişim Algoritmasında deneme vektörü olarak üç yavruda hangisinin en iyi olduğunun seçiminde ve hedef vektör ile deneme vektörünün karşılaştırılması aşamasında kullanılmıştır (Wang ve ark., 2018).

Rüzgar türbini düzen optimizasyon problem çalışmasında, farklı türbinlerle ve farklı arazi müsaitlik seviyeleri ile rüzgar türbini optimizasyonuna odaklanılmıştır. Çok amaçlı bir optimizasyon problemi kullanılan bu çalışmada, önerilen hibrit yöntem, dinamik penaltı yaklaşımını ve kısıtlı programlama yöntemlerini temel alır. Böylece hem yerel hem de küresel keşif davranışlarını dengelemektedir (Sorkhabi ve ark., 2018).

Geliştirilmiş dengeli sıralama metotu (E-BRM), evrimsel algoritmalar ile çözülen kısıtlı optimizasyon problemleri için geliştirilmiş bir yöntemdir. Bu yöntem kendinden uyarlamalı bir yöntemdir. Uygun çözümler ve uygun olmayan çözümler için iki ayrı liste oluşturulur ve bu listeler keşif sürecinde birleştirilir. Literatürde sıklıkla kullanılan kısıtlı fonksiyon setleri ve mühendislik tasarım optimizasyon problemleri üzerinde algoritmanın performans testleri yapılmıştır. Yapılan testler sonucunda, önerilen yöntemin çalışmada kullanılan algoritmalardan daha iyi performans gösterdiği gözlemlenmiştir (Rodrigues ve ark., 2018).

Ahmet Babalık ve ark.(2018),yaptıkları çalışmada, TSA(Tree-Seed Algoritması) üzerinde modifikasyon yaparak algoritmayı kısıtlı optimizasyon problemlerini çözmeye odaklanmışlardır. Kısıt işleme yöntemi olarak Deb's Rule yöntemi kullanılmıştır. İyi bilinen 13 test fonksiyonu ve mühendislik tasarım optimizasyon problemleri üzerine testler yapılmıştır ve sonuçlar gözlemlenmiştir. Kısıtlı optimizasyon problemlerini çözmek için önerilen bu algoritmanın, çözüm kalitesi ve dayanıklılığı bakımından en gelişmiş yöntemler ile karşılaştırıldığında rekabetçi bir algoritma olduğu görülmüştür (Babalik ve ark., 2018).

3. MATERYAL VE YÖNTEM

3.1. Optimizasyon

Optimizasyon, belirli sınırlamaları sağlamak için, en uygun bilinmeyen parametre değerlerini bulma işidir. Bir diğer tanımla, verilen koşullar yada kısıtlar altında en iyi sonucu üretme işine optimizasyon adı verilir. Gerçek dünyada da örneklerine rastlanabilecek problemlerde, amaç kaynakların minimum düzeyde tutularak istenilen verimin sağlanmasıdır.

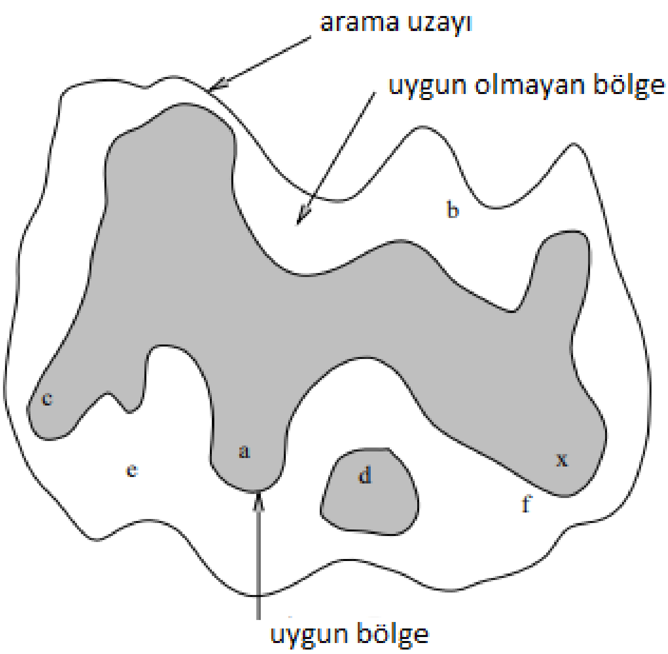
3.2. Kısıtlı Optimizasyon

Kısıtlı optimizasyon problemleri, eşitlik, eşitsizlik, alt sınır ve üst sınır içeren, optimizasyon sürecinde bu kısıtların dikkate alındığı problemlerdir. Genel olarak kısıtlı optimizasyon problemleri, Denklem 3.1 ile Denklem 3.4 arasındaki denklemler gibi formüle edilir;

$$Minimum\ yada\ Maksimum\ f(X) \tag{3.1}$$
$$h_i(X) = 0\ i = 1,2, \dots, m \tag{3.2}$$
$$g_i(X) \geq 0\ i = m + 1, \dots, p \tag{3.3}$$
$$l_i \leq X \leq u_i\ \ i = 1,2 \dots n \tag{3.4}$$

Denklem 3.1 ve Denklem 3.4 arasındaki denklemlerde, f(x) amaç fonksiyonu, h(x) eşitlik kısıtı, g(x) eşitsizlik kısıtı, l ve u değişkenleri alt ve üst sınırları ifade etmektedir. Teorik olarak belirtildiğinde h(x) ve g(x) kısıt fonksiyonları dikkate alınarak, değişkenler için alt ve üst sınırlar arasında uygun f(x) fonksiyonunu bulma işlemi kısıtlı optimizasyon olarak adlandırılmaktadır.

Çoğu gerçek dünya problemi kısıtlı optimizasyon problemine örnek olarak verilebilir. Bu tip problemleri çözmenin en zorlu kısmı, sahip oldukları kısıtları işleme sürecidir. Problemlerin temelini bu kısıtlar oluşturduğundan, problem için önem arz eder ve problemin çözümünü zorlaştırmaktadır.



Şekil 3.1.Kısıtlı problemler için arama uzayı(Michalewicz, 1995)

Şekil 3.1'de gösterilen arama uzayında, uygun olmayan bölgeler ve uygun olan bölgeler belirtilmiştir. c, a ve x çözümleri uygun bölgede yer almaktadır. b, e, d ve f çözümleri uygun olmayan bölgede yani kısıtlara takılmış bölgede yer alır. Optimum çözüm ise x çözümüdür.

3.3. Kısıt İşleme Yöntemleri

Metasezgisel optimizasyon algoritmalarının orijinal versiyonları genellikle kısıtsız arama uzayında arama yapmak için tasarlanmıştır. Kısıtları işleme yöntemleri metasezgisel algoritmalara uygun çözümlerin bulunduğu bölgelerde aramaya rehberlik etmesi amacı ile eklenen yöntemlerdir. Bu yöntemler sayesinde arama uzayında yapılan keşif sırasında, arama algoritması uygun olmayan çözümlerden uzaklaşacak ve uygun olan çözümlere yönelecektir.

Kısıt işleme yöntemlerinin sahip olması gereken özellikler şunlardır;

1. Hesaplamada verimli olmaları gerekir.
2. Uygulanabilir çözümler için uygun değerleri tercih etmelidir.
3. Tüm çözüm uzayını araştırmalı ve erken yakınsamalardan kaçınmalıdır.

4. Yöntemde kullanılacak parametrelerin hassas ayarını en aza indirmeli ve parametrik değerler evrensel olmalıdır.
5. Kavramada ve uygulamada basit olmalıdır.

Kısıt işleme tekniklerinin en iyi bilinenlerinin karşılaştırılması, Çizelge 3.1'de verilmiştir. Her bir kısıtlama taşıma tekniği bu kriterleri, belirli bir dereceye kadar yerine getirse de, kriterleri karşılama konusundaki genel yetenekleri Çizelge 3.1 'de sunulmuştur (Mani ve Patvardhan, 2009).

Çizelge 3.1. Kısıt işleme tekniklerinin karşılaştırılması(Mani ve Patvardhan, 2009)

Kriter	Penaltı Metodu	Death Penaltı/ Feasibility Rules	Stochastic Ranking	Yöntemden beklenen
Verimlilik	İyi	İyi	İyi	İyi
Uygun çözüm her zaman kazanır	Hayır	Evet	Hayır	Evet
Keşif	İyi	Zayıf	İyi	İyi
Hassas Ayar	Evet	Hayır	Evet	Hayır
Kavrama ve Uygulama Basitliği	Evet	Evet	Hayır	Evet

Kısıt işleme teknikleri üzerine yapılan ilk çalışmalar için ilk sınıflandırma Michalewicz ile Schoenauer ve Coello Coello tarafından sunulmuştur. Fonksiyonlar Penaltı fonksiyonları ve hibrit metotlar olarak iki temel sınıfa ayrılmıştır. Penaltı fonksiyonları günümüzde popülaritesini korumaktadır (Coello, 2002)(Michalewicz, 1995).

2011 yılında Mezura ve Coello'nun kısıt işleme teknikleri üzerine yapılan çalışmalarında, var olan fonksiyonlar 3 sınıfa ayrılmıştır. Bunlar; ilk yıl çalışmaları, günümüz teknikleri ve gelecek eğilimlerdir.(Mezura-Montes ve Coello, 2011)

İlk yıl çalışmaları;

1. Penaltı Fonksiyonları
2. Çözücüler(Decoders)
3. Özel operatörler
4. Amaç Fonksiyonlarının ve kısıtların ayrılması

Günümüz çalışmaları;

1. Uygulanabilirlik Kuralları
2. Stokastik sıralama
3. ϵ-kısıt işleme
4. Yeni penaltı fonksiyonları

5. Yeni özel operatörler
6. Çok amaçlı kavramlar
7. Kısıt işleme tekniklerinin birleştirilmesi/toplanması

Gelecek eğilimler;

1. Kısıtlama yaklaşımı
2. Dinamik kısıtlar
3. Teori

3.3.1.Penaltı Fonksiyonları

Penaltı fonksiyonlarında, amaç fonksiyonlarına penaltı fonksiyonlarını adapte ederek metot dönüştürülmektedir. Yöntemin amacı; uygun çözümlerin seçimini kolaylaştırmak için uygun olmayan çözümlerin uygunluk değerini arttırarak onların seçimini azaltmaktır.

$$Q(x) = f(x) + p(x) \tag{3.5}$$

$$p(x) = \sum_{i=1}^{m} r_i \,.\max(0, g_i(x))^2 + \sum_{j=1}^{p} c_j .\,|h_j(x)| \tag{3.6}$$

$Q(x)$ optimizasyon için genişletilmiş amaç fonksiyonudur. $p(x)$ ise penaltı fonksiyonudur. Denklem 3.5 ile genişletilmiş amaç fonksiyonu hesaplaması ve Denklem 3.6 ile penaltı fonksiyonu hesaplanması sunulmuştur. Denklemde bulunan r_i ve c_j değerleri penaltı faktörü olarak adlandırılan pozitif sabit değerlerdir.

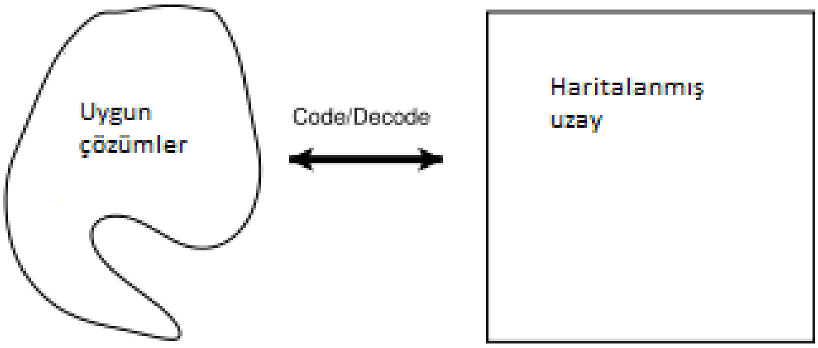
Penaltı yöntemlerinden bazıları;

1. Death Penaltı
2. Dinamik Penaltı
3. Adaptif Penaltı
4. Fuzzy Adaptif Penaltı
5. Static Penaltı
6. Annealing Penaltı

En yaygın bilinen penaltı yöntemi death penaltı yöntemidir. Uygun olmayan çözümlere çok yüksek ceza değerleri verilir ve basitçe optimizasyon sürecinden bu çözümler elimine edilir. Ayrıca dinamik penaltı, adaptif penaltı, fuzzy adaptif penaltı, annealing penaltı gibi yöntemler de literatürde sıkça kullanılmıştır. Genel olarak tüm penaltı fonksiyonlarının dezavantajı; penaltı faktörünün ayarlanmasındaki hassasiyet ve ilave parametre ihtiyaçlarıdır (Joines ve R. Houck, 1996; Mezura-Montes ve Coello, 2011).

3.3.2.Decoder Yöntemi

İlk yılların en rekabetçi kısıt işleme tekniklerinden biridir. Bu teknikler, algoritmanın daha iyi performans sağlayabildiği örnek alanından arama uzayının uygun bölgelerini haritalama fikrine dayanır.(Şekil 3.2)



Şekil 3.2. Genel bir decoder'in grafiği(Mezura-Montes ve Coello, 2011)

Bir çözücünün haritalama sürecinde dikkate alınması gereken 3 özellik bulunmaktadır. Bunlar;

1. Haritalanmış uzayın, arama uzayındaki tüm uygun bölgeleri içerdiğini garanti etmelidir.
2. Tüm uygun çözümler, haritalanmış uzaydaki çözüm sayısı ile aynı olmalıdır.
3. Haritalama işlemi hızlı olmalıdır.

Koziel ve Michalewicz (1999) tarafından, n boyutlu bir küp ile uygulanabilir bir arama uzayı arasında eşleme önerilmiştir. Bu yaklaşımın ana fikri, orijinal problemi GA tarafından optimize edilmesi daha kolay olan başka bir fonksiyona dönüştürmektir. Problem boyutundan bağımsız bir yöntemdir. Uygulanabilir bölgeyi GA için arama yapmayı kolaylaştıracak şekle dönüştürmek için Riemann eşlemeleri kullanan benzer yaklaşımlar da önerilmiştir. Ancak önerilen bu versiyon, düşük boyutlu problemlerde kullanılabilmektedir (Gyu Kim ve Husbands, 1998; Koziel ve Michalewicz, 1999).

Bu yöntemlerin dezavantajı yüksek hesaplama maliyeti gerektirmesidir. Aynı zamanda, bir dizi işlem gerektirerek elde edilmesi gereken parametre kullanması dezavantaj olarak gösterilmektedir. Performansları değerlendirilirken sadece uygunluk fonksiyonun maliyetinin değerlendirilmesi de eksiklik olarak belirtilmektedir. Bu yüzden günümüzde nadir görülen bir yaklaşımdır (Mezura-Montes ve Coello, 2011).

3.3.3.Özel Operatörler

Kısıtlı optimizasyon problemi çözülürken özel operatörler bir çözümün uygunluğunu korumak veya arama uzayının belirli bir bölgesine (uygun bölge sınırlarında) taşımak için kullanılır. Bu yöntemler bazı ön işlemler gerektirmektedir. Bazı araştırmacılar bu yöntem ile oldukça rekabetçi sonuçlar elde etmiştir. En büyük dezavantajı uygulanabilirliğinin sınırlı olmasıdır (Mezura-Montes ve Coello, 2011).

Başlangıç çözümlerinde en az bir uygun çözüm gerektiğinden, oldukça fazla kısıta sahip problemlerde zor ve hesaplama maliyeti yüksek olur.

3.3.4. Amaç Fonksiyonlarının ve Kısıtların Ayrılması

Bu teknik, penaltı yöntemleri gibi amaç fonksiyonundan dönen değer ve kısıtlardan dönen değeri tek bir değerde birleştirmez. Seçim süreçlerinde her iki değeri de korur. Ana fikir arama işlemini iki bölüme ayırmasıdır. Birinci aşamada amaç fonksiyonundan dönen değer ne olursa olsun uygun çözümleri bulmaktır. Belirli bir sayıda uygun çözüm bulduktan sonra ikinci aşamada amaç fonksiyonun hedefini bulmayı hedefler.

Kısıt ihlallerini ve amaç işlevlerini ayırma yaklaşımını benimseyen birkaç yaklaşım vardır; Örneğin Powel ve Skolnick (1993) amaç fonksiyonunu 1 ve -∞ arasında ölçeklendirirken, kısıt fonksiyonlarını 1 ve +∞ aralığına ölçeklendirmektedir. Çözüm uygun olmadığı durumda, kısıt fonksiyonları amaç fonksiyonlarına eklenmez. Arama sırasında her bir birey Denklem 3.7'ye göre değerlendirilir.

$$\Psi(x) = \begin{Bmatrix} \text{eğer uygun çözüm ise; } \mathrm{f(x)} \\ \text{diğer durumlar; } 1 + A\left(\sum_{i=1}^{n} g_i(x) + \sum_{j=1}^{m} h_j(x)\right) \end{Bmatrix} \tag{3.7}$$

Denklem 3.7'deki A parametresi kullanıcı tarafından belirlenen bir sabittir. Bu yöntemdeki zorluk A parametresinin tanımı değil, uygulanabilir çözümler arasındaki üstünlük durumudur. Deb (2000) benzer bir yaklaşımı kullanarak Denklem 3.8'i önerir.

$$\Psi(x) = \begin{Bmatrix} eğer\ uygun\ çözüm\ ise;\ \mathrm{f(x)} \\ diğer\ durumlar; f_{worst} + \sum_{i=1}^{m+n} k_i(x) \end{Bmatrix} \quad (3.8)$$

Denklem 3.8'de gösterilen f_{worst} değişkeni popülasyon içindeki en kötü uygulanabilir çözümü sunar ve $k_i(x)$ eşitsizlik fonksiyonları ile eşitsizlik fonksiyonlarına dönüştürülmüş eşitlik fonksiyonlarını ifade eder (Deb, 2000; Chehouri ve ark., 2016).

3.3.5. Uygulanabilirlik Kuralları

Deb tarafından önerilen uygunluk kuralları günümüzde hala etkili bir yöntemdir. Günümüzde uygunluk kuralları üzerine geliştirmeler, bu kuralları algoritma elemanları üzerine yapılan yenilikler ile birleştirilerek devam ettirilmektedir. Bu yöntemlerin hala popüler olmasının sebebi basit ve esnek olmasıdır. Seçim mekanizmalarına kolaylıkla adapte edilebilir. Fakat sadece uygun seçimlerin korunması, erken yakınsama problemini beraberinde getirmektedir. Bunu aşmak için çeşitliliği koruyan mekanizmalar eklenmektedir (Deb, 2000).

3.3.6. Stokastik Sıralama

Runarsson ve Yao tarafından önerilmiştir. Penaltı metotlarının doğal eksiklikleri ile başa çıkmak için tasarlanmıştır. Stokastik sıralama (SR)' de penaltı faktörünün tanımı yerine uygun olmayan çözümlerin kıyaslanması için kullanıcı tanımlı kontrol parametresini kullanır. Arama uzayının uygun olmayan bölgelerinin sıralama içinde amaç fonksiyonuna göre kıyaslanabilmesinin olasılığı P_f parametresi ile sunulmuştur. Bu sayede iki birey uygun ise amaç fonksiyonuna göre, değilse P_f olasılığında amaç fonksiyonuna göre, aksi takdirde kısıt cezalarına göre kıyaslanır (Runarsson ve Yao, 2000).

3.3.7. ϵ-kısıt İşleme

Takahama ve Sakai (2010) tarafından önerilmiştir. Bu yöntem, kısıtlı bir problemi kısıtsıza dönüştürür ve iki ana bileşeni vardır: Birincisi, bir çözümün uygun olabilmesi için, kısıtların toplamına dayalı, karşılaştırma kriteri olarak kendi amaç fonksiyon değerini kullanmak ve kısıt değerlerini kullanmaktır. İkincisi ise sıralama mekanizmasıdır. Kısıtların toplamının minimizasyonu, amaç fonksiyonun minimizasyonundan önce gelir. Yani öncelikle kısıtların toplamının minimum noktaya getirilmesi amaçlanır ve böylece problemin kısıtlardan kurtarılması hedeflenir. Temelde uygunluk kurallarına dayanır fakat ε değeri dikkate alınarak işlemler kontrollü şekilde yapılır (Mallipeddi ve Suganthan, 2010; Takahama ve Sakai, 2010).

3.3.8. Yeni Penaltı Fonksiyonları

Tartışılan eski kısıt işleme tekniklerinin penaltı fonksiyonundan kaçınmalarına karşı, oldukça rekabetçi sonuçlar sağlayan penaltı metotları sunulmuştur. Penaltı metotları hala günümüzde bazı değişiklikler ile kullanılmaya devam etmektedir.

Uygulama ve kavramadaki basitliğinden dolayı, hassas parametre ayarı dezavantajına rağmen, penaltı yöntemleri üzerine yapılan çalışmalar popülerdir. Yöntem üzerinde, parametre ayarları ile ilgili değişiklikler yapılmaktadır (Mezura-Montes ve Coello, 2011)

3.3.9. Yeni Özel Operatörler

Günümüzde kullanılan yöntemleri içeren yeni özel operatörler yaklaşımı, uygun ve uygun olmayan çözümler arasında ikili arama yapan bir operatör kullanmaktadır. Bu yöntem işlenen kısıtın sınırında bulunan ancak diğer kısıtları ihlal eden çözümlerle başa çıkmak için ek bir kısıtlama tekniğine ihtiyaç duymaktadır. Bu durum dezavantaj olarak görülmektedir.

Genel olarak penaltı metodu ile birlikte kullanılır. Bu operatöre ek olarak çift popülasyon yaklaşımı da önerilmiştir (Mezura-Montes ve Coello, 2011).

3.3.10. Çok Amaçlı Kavramlar

Deneysel çalışmalar bu tekniklerin kısıtlı optimizasyon yöntemlerini çözmek için çok uygun olmadığını gösterse de bu kavramlara dayalı kısıt işleme teknikleri vardır. Bu yöntemin kullanılması sırasında genellikle Deb's Rule, Penaltı fonksiyonları gibi diğer kısıt işleme teknikleri ile birlikte kullanılmıştır. Aynı zamanda amaç fonksiyonuna bir amaç fonksiyonu daha ilave edilmiştir. İlave fonksiyon kısıtların şiddetlerinin ölçülmesine dayanır. İlki amaç fonksiyonudur, ikinci amaç eşitsizlik kısıtlarının toplamı, üçüncüsü ise eşitlik kısıtlarının toplamı olmuştur (Mezura-Montes ve Cetina-Domínguez, 2009; Long, 2014; Zain ve ark., 2018).

3.3.11. Kısıt İşleme Tekniklerinin Birleştirilmesi

Kısıt işleme tekniklerinin birlikte kullanılması ile yapılan çalışmalardır. Mallipeddi ve Suganthan (2010), 4 kısıt işleme tekniğinin (uygunluk kuralları, stokastik sıralama, self-adaptif penaltı yöntemi ve ε-kısıt işleme yöntemi) birleşimini önermişlerdir ve her bir tekniği alt popülasyonlar üzerinde uygulamışlardır. Bu yönteme benzer tekniklerin birlikte kullanımları güncel uygulamalarda sunulmuştur (Mezura-Montes ve Coello, 2011).

Çok amaçlı optimizasyon problemleri için kısıt işleme tekniklerinin geliştirilmesi literatürde nispeten az ilgi gören bir alandır. Bunun nedeni, çoğu araştırmacı tarafından tek amaçlı optimizasyon problemleri için kullanılan yöntemlerin çok amaçlı optimizasyon algoritmaları için de kullanılabilir olduğunu düşünmesi olarak belirtilmektedir. Çok amaçlı optimizasyon problemlerinde kısıt işleme yöntemleri ile ilgili yapılacak bir çok geliştirme vardır ve gelecek eğilimlerin bu yönde olacağı düşünülmektedir (Mallipeddi ve Suganthan, 2010).

3.3.12. Kısıtlama Yaklaşımı

Kısıtlı optimizasyon problemlerine sık sık uygulanan uygunluk yaklaşımlarına rağmen, bu metotların kısıtlı sayısal optimizasyon problemleri üzerine kullanımı azdır. Kısıtlı sayısal optimizasyon problemleri ile uğraşmak için kısıtlara yaklaşım tekniklerinin kullanımı daha azdır ve gelişmeye açık bir alandır (Jin, 2005).

3.3.13. Dinamik Kısıtlar

Günümüzde dinamik optimizasyon problemleri ile uğraşan bir çok çalışma bulunmaktadır. Ancak bu çalışmaların büyük bir kısmı kısıtsız arama uzayına odaklanmıştır. Bu alanda daha fazla çalışma gerektiği ve buna uygun test fonksiyonlarına ihtiyaç duyulduğu belirtilmektedir.(Mezura-Montes ve Coello, 2011)

3.3.14. Teori

Kısıtlı optimizasyon problemleriyle başa çıkmak için önemli miktarda araştırma bulunmaktadır ancak bunların çoğu kısıtsız arama uzaylarına odaklanmıştır ve kısıtlı problemler için yapılan çalışmalar oldukça azdır. Gelecek eğilimleri içinde ilgi görecek bir alan olduğu düşünülmektedir (Nguyen ve Yao, 2009).

3.4. Yapay Alg Algoritması(AAA)

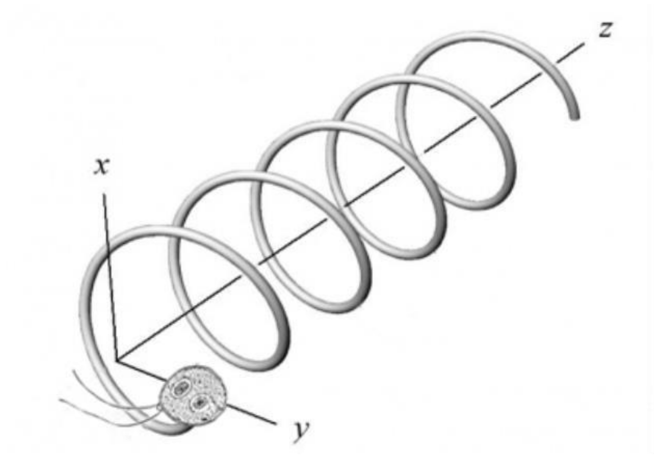
AAA mikro alglerin yaşam davranışlarından esinlenilerek ortaya konmuş bir optimizasyon algoritmasıdır.

AAA, yapay alglerin davranışlarından esinlenilerek 2015 yılında ortaya konmuş doğa esinli bir optimizasyon algoritmasıdır. Yapay alg, gerçek yaşamdaki bir alg gibi, fotosentez yapabilmek için ışık kaynağına doğru hareket eder ve hareketi helisel yüzme şeklindedir. Ortama adapte olup, baskın türü değiştirebilir ve mitoz bölünme ile çoğalabilir. Problem uzayındaki her bir çözüme bir yapay alg kolonisi karşılık gelmektedir. Her bir alg kolonisindeki alg hücresi sayısı, problem boyutuna eşittir. Bir alg kolonisi ideal çözüme ulaştığında optimum elde edilmiş olur. AAA 3 temel adımdan oluşur. Bunlar; helisel hareket, adaptasyon ve evrimsel süreç' tir (Uymaz ve ark., 2015a).

3.4.1. Helisel Hareket

Alglerin hayatta kalabilmeleri için gereken ışık, su yüzeyinden bulunduğundan, algler su yüzeyine yakın şekilde yaşamalıdırlar. Yapay alg hücreleri, ışığa doğru helisel olarak hareket etmektedir (Şekil 3.3).

Su yüzeyine doğru yapılan hareketi engelleyen yerçekimi kuvveti ve hareketli hücre ile etrafındaki su arasındaki viskoz sürtünme kuvveti mevcuttur.



Şekil 3.3. Alglerin hareket deseni(Uymaz ve ark., 2015a)

Her alg hücresinin hareketi farklıdır. Büyük alg hücrelerinin sürtünme yüzeyleri arttığından daha yavaş hareket eder. Büyük alg hücrelerinin aksine küçük alg hücrelerinin sürtünme yüzeyleri daha fazla olduğundan hareketleri de daha hızlıdır. Alglerin hareketlerindeki bu 2 farklılık, büyük alglerin yani algoritmadaki iyi değerlerin yerinin daha az değişmesine o alanda yoğunlaşmaya olanak sağlarken, küçük alglerle ifade edilen kötü çözüm alanlarından algoritmanın uzaklaşmasını sağlamaktadır. Bu sayede algoritmanın keşif ve sömürü yetenekleri artar.

Her bir helisel hareket sonucunda koloninin uzayda yer değiştirip değiştirmeyeceğini enerjileri belirler. Her döngünün başlangıcında, enerji koloni büyüklüğüne doğru orantılı olarak hesaplanır ve bu enerji çözümün kalitesini göstermektedir (Uymaz ve ark., 2015b).

3.4.2.Adaptasyon

Adaptasyon, yeterince büyüyemeyen bir alg kolonisi tarafından çevrede bulunan en büyük alg kolonisine benzemeye çalıştığı süreçtir. Bu benzeme aşamasında baskın türü değiştirmektedir. Helisel hareket sonucunda belirlenen açlık seviyesi kullanılır. Bu açlık seviyesi, algoritma başlangıcında sıfır olarak belirlenir. Daha iyi çözüme giden kolonide açlık seviyesi değişmezken, kötüleşen koloninin açlık seviyesi artar. Her bir helisel hareketten sonra, en yüksek açlık değerine sahip olan koloni adaptasyona uğrar. Her bir çevrimin sonunda A_p parametresi kontrol edilerek adaptasyon işlemi buna göre adaptasyon uygulanır.

3.4.3.Evrimsel Süreç

Yapay alg hücresi yeterli ışık aldığında büyür, gelişir ve bölünme ile iki yapay alg hücresi oluşur. Yeterli ışık alamayan alg hücresi ise bir süre sonra ölür. Evrimsel süreç; en küçük alg kolonisinin ölen her bir hücresi yerine en büyük alg kolonisinin bir alg hücresinin kopyalandığı aşamadır. Algoritma başlangıcında koloni büyüklükleri bir olarak kabul edilirken helisel hareket sonucunda kolonilerin büyüklükleri değişir. Büyüklüğe göre sıralanan alg kolonileri rastgele seçilen boyutta evrimsel sürece tabi tutulur.

3.4.4.AAA Sözde Kodu

AAA'ya ait sözde kod Şekil 3.4 ile verilmiştir.

```
AAA
Rastgele çözümlere sahip n adet alg kolonileri ile başlangıç popülasyonu oluştur.
Her bir koloni için Amaç Fonksiyonu olan f(x) değerlerini ve büyüklükleri hesapla.
While (durdurma kriteri sağlanıncaya kadar devam et)
For i=1:n
While i. Koloninin enerjisi bitine kadar devam et
Koloniyi helisel hareket ile değiştir.
End while
End for
Evrimsel Süreci uygula
Adaptasyon işlemini uygula
End While
```

Şekil 3.4.AAA'nın sözde kodu

3.5.Kısıtlı Problemler için AAA

AAA kısıtsız problemlerdeki başarısını ortaya koymasına rağmen, kısıtsız problemler için ortaya konmuş bir versiyonu bulunmamaktadır. Bu çalışmada kısıt işleme fonksiyonları Yapay Alg Alg algoritmasına adapte edilerek, algoritmanın kısıtsız problemleri çözmesi hedeflenmiştir. Önceden yapılmış çalışmalar ışığında 3 kısıt işleme yöntemi seçilmiştir. Bunlar; Deb's Rule kısıt işleme yöntemi, ε-kısıt işleme yöntemi ve dinamik penaltı yöntemidir. AAA'ya uygulanan ilk yöntem Deb's Rules ile kısıtlı optimizasyon problemleri için AAA_{dr} algoritması ortaya konmuştur. Dinamik penaltı yönteminin uyarlanması ile ortaya çıkan algoritma AAA_{dp} olarak adlandırılmıştır. ε-kısıt metodunun AAA'ya uyarlanması ile AAA_{ε} algoritması ortaya çıkmıştır.

3.5.1.Deb's Rule Yöntemi Kullanılarak Kısıtlı Problemler için AAA (AAA_{dr})

Kısıtlı optimizasyon problemlerinde , aramayı uygun bölgeye yönlendirmek için 3 uygunluk kriteri Deb (2000) tarafından eklenmiştir. Deb's Rules kısıt işleme yönteminin temel avantajı, erken yakınsamalar yol açabilmesine rağmen, kullanıcı tanımlı parametrelerin bulunmamasıdır. Bu yöntemdeki kurallar şu şekildedir;

1. İki uygun çözüm arasında karşılaştırma yapılırken amaç fonksiyonu en büyük olan seçilir.
2. Uygun çözüm ve uygun olmayan çözüm karşılaştırılırken uygun çözüm seçilir.
3. İki uygun olmayan çözüm arasından seçim yapılırken, kısıt ihlali en düşük olan çözüm seçilir (Brajevic ve ark., 2012).

AAA'da, başlangıçta bireyler sadece alt ve üst sınırlara göre belirlendiğinden uygun yada uygun olmayan amaç fonksiyonları üretebilir. Algoritma sürecinde Deb's Rule yöntemi ile çözümler uygulanabilir bölgeye doğru yönlendirilir.

Evrimsel Süreçte büyüme kinetiğinin hesaplanmasında uygun çözüm bulan bireyler dışındakiler hariç tutuldu. Hesaplanan kısıt değerine göre, kısıta takılmış herhangi bir çözümün değeri optimum değere ne kadar yakın olursa olsun, kısıtları sağlayamadığından bu çözümler göz ardı edildi. Yani sadece uygun çözüm bulan bireyler gelişme gösterdi ve büyüdü.

Helisel Hareket sürecinde algin hareketini sağlayan enerji hesaplaması önce yapılan büyüklüklere göre sıralamada, çözümlerin kısıta takılma durumları da değerlendirildi. AAA'da, büyüklük sıralaması sadece amaç fonksiyonuna göre yapılırken, modifiye edilmiş AAA'da kısıt değerine sahip bireyden başlanarak amaç fonksiyonları sıralandı.

Işık kaynağı seçiminde kısıt işleme yöntemlerinden Deb's Rule yöntemi kullanıldı.

En iyi çözümü karşılaştırma ve belirleme gibi iki birey arasında yapılacak seçimlerde, sadece amaç fonksiyonuna bakmak yerine kısıtların da dikkate alındığı Deb's

Rule yöntemi kullanıldı. Böylece seçim yapılacak iki birey arasında, çözümü optimuma yakın ancak kısıtlara takılmış bir birey ile kısıtlara takılmayan ancak çözümün biraz daha geride kaldığı durumlar arasında algoritmanın doğru seçim yapması hedeflenmiştir.

Adaptasyon sürecinde, baskın tür olarak eğer uygun çözüm yok ise en düşük kısıt ihlaline sahip olan uygun olmayan çözüm seçildi.

3.5.2.Dinamik Penaltı Yöntemi Kullanılarak Kısıtlı Problemler için AAA (AAA_{dp})

Dinamik Penaltı yönteminde amaç fonksiyonuna dinamik penaltı yöntemi ile elde edilen değer eklenmiştir. Statik penaltı yöntemindeki temel eksiklik, parametrelerin kullanıcı tarafından belirlenmesidir. Bu yöntemde, arama ilerledikçe cezanın şiddeti artmaktadır. Aramanın başlangıcında uygun olmayan bölgede amaç fonksiyonları üretebilmesine rağmen, arama ilerledikçe çözümleri uygun alana taşımak için verilen ceza artmaktadır (Kuri ve Gutiérrez-García, 2002). Genel formülü denklem 3.9'da verilmiştir.

$$p(x) = (\sum_{i=1}^{N} c_i) * (\frac{t}{2})^2 \tag{3.9}$$

Denklem 3.9'daki t ifadesi her bir değeri ifade ederken c değeri kısıt değerlerini ifade etmektedir.. Eğer p(x) değeri çok düşükse uygun olmayan çözümler ortaya çıkabilir yada $p(x)$ değeri çok şiddetli ise arama optimum olmayan uygun çözümlere yakınlaşabilir (Smith ve ark., 1998).

Önerilen algoritmada en iyi birey seçiminde ve popülasyon büyüklük hesaplanmasında kısıt değerleri dikkate alınarak kısıta takılmayan değerlerin seçilmesi sağlanmıştır.

3.5.3. ε-Kısıt İşleme Yöntemi Kullanılarak Kısıtlı Problemler için AAA (AAA_{ε})

ϵ-kısıt yönteminde arama algoritmalarının sıralamalarını epsilon ile karşılaştırarak kısıtsız problemleri kısıtlı problemlere dönüştürmektedir. Bu yöntem ile çözülen bir optimizasyon problemi, sıradan bir sıralama yerine ϵ düzeyinde bir karşılaştırma ile tanımlanır. Algoritmanın optimumu arama sürecinde fonksiyon hesaplama sayısının toplam fonksiyon hesaplama sayısına oranı iterasyon sayacı olarak adlandırılmıştır. Bu çalışmada kullanılan ϵ değeri, iterasyon sayacı T_c değerine ulaşıncaya kadar güncellenir. Jenerasyon sayacı T_c değerini aştığında kısıt ihlali olmayan çözümler elde etmek için 0 olarak ayarlanır.

$$\epsilon(0) = \upsilon(X_i) \tag{3.10}$$

$$\epsilon(k)=\left\{\frac{\epsilon(0)\ \left(1-\frac{k}{T_c}\right)^{cp}}{0,\quad k\geq T_c}\right\}\ , 0 < k < T_c \tag{3.11}$$

$$cp = -\frac{log_{\epsilon_0} + \lambda}{\log(1-p)} \tag{3.12}$$

$$0.1T_{maks} \leq T_C \leq 0.8T_{maks} \tag{3.13}$$

$$2 \leq cp \leq 10 \tag{3.14}$$

Denklem 3.10 ve 3.14 arasında verilen X_i i. Bireyi temsil etmektedir ve i=(0.005* Popülasyon Sayısı) olarak hesaplanır. Sabit verilen T_c ve cp değerleri için önerilen aralık denklem 3.13 ve 3.14 ‘de belirtilmiştir. T ifadesi maksimum iterasyon sayısını göstermektedir. İki birey arasında Denklem 3.15 ve Denklem 3.17 arasındaki şartlar sağlandığı takdirde ; X_i bireyinin X_j bireyinden daha iyi olduğu söylenir (Takahama ve Sakai, 2012; Wang ve ark., 2018).

Eğer $G(X_i) \leq \epsilon\ ve\ G(X_j) \leq \epsilon)$; $\quad f(X_i) < f(X_j)$ (3.15)

Eğer $G(X_i) = G(X_j)$; $\quad f(X_i) < f(X_j)$ (3.16)

Diğer Durumlar ;$G(X_i) < G(X_j)$ (3.17)

Bu yöntemin uygulanması aşamasında öncelikle Alglerin sıralama durumu değişmiştir. Belirlenen Epsilon değeri dikkate alınarak Alg popülasyonu sıralama işlemine gönderilmiştir. Burada öncelikle kısıtlara takılmayan değerlerin yani kısıt fonksiyonundan sıfır veya daha düşük değerler gelen bireylerin kısıt değerleri sıfır değerinde eşitlenmiştir. Öncelikle kısıt değerlerine daha sonra amaç fonksiyonlarına göre popülasyon sırası güncellenerek algoritmaya başlanır. Her bir çevrim sonucunda popülasyonun sıralaması güncellenerek en az kısıta sahip olan ve en iyi amaç fonksiyonuna sahip olan birey popülasyonun ilk sırasında bulunur. Bu aşamada güncellenen epsilon değerine göre iki birey arasında karşılaştırma denklemlere göre karşılaştırma yapılır.

Helisel Hareket sürecinde alg hücresinin hareketini sağlayan enerji hesaplaması önce yapılan büyüklüklere göre sıralamada, çözümlerin kısıta takılma durumları da değerlendirilmiştir. AAA'da, büyüklük sıralaması sadece amaç fonksiyonuna göre yapılırken, ϵ-kısıt yöntemi ile modifiye edilmiş AAA'da kısıt değerine sahip bireyden başlanarak amaç fonksiyonları sıralanmıştır.

Işık kaynağı seçiminde kısıt işleme yöntemlerinden ϵ-kısıt tekniğine göre karşılaştırma yapılmıştır.

Adaptasyon sürecinde, baskın tür olarak eğer uygun çözüm yok ise en düşük kısıt ihlaline sahip olan uygun olmayan çözüm seçilmiştir.

3.6. Algoritmanın Test Edilmesi

Algoritmaların performanslarının test edilmesi için literatürde kullanılan birçok test fonksiyonu bulunmaktadır. Algoritmalar üzerinde yapılan çalışmalar sonucunda hem algoritmanın başarısının gözlemlenmesi hem de daha önceden yapılan algoritmalarla karşılaştırılmasının yapılabilmesi için bu test fonksiyonları kullanılmaktadır. Standart bir fonksiyon seti bulunmasa da yapılan çalışmaya uygun ve karşılaştırmalara uygun fonksiyon seti seçimi yapılmalıdır.

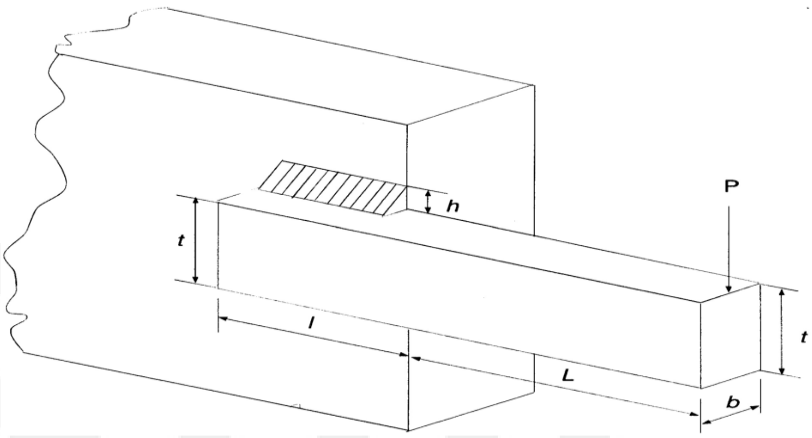
Bu çalışmada iyi bilinen ve literatürde sıklıkla kullanılan 13 Benchmark fonksiyon seti ve mühendislik tasarım optimizasyon problemleri kullanılmıştır. Test aşamasında kullanılan tüm fonksiyonlar kısıtlara sahip fonksiyonlardır.

3.6.1.Mühendislik Tasarım Optimizasyon Problemleri

Mühendislik Optimizasyon problemleri, optimizasyon problemlerinin performansını değerlendirmek için literatürde sıklıkla kullanılan problemlerdir. Çoğu mühendislik optimizasyon problemi, amaç fonksiyonu ve yerine getirilmesi gereken kısıtlamalardan oluştuğu için karmaşık problemlerdir. Bu kısıt önemlidir, çünkü sorunun temelini oluşturur ve araştırmayı zorlaştırır. Bu çalışmada 5 adet mühendislik tasarım optimizasyon problemi kullanılmıştır. Bunlar; Kaynaklı Kiriş Tasarım Optimizasyon Problemi, Hız Düşürücü Tasarım Optimizasyon Problemi, Basınçlı Kap Tasarım Optimizasyon Problemi, Germe/Sıkıştırma Yay Tasarım Optimizasyon Problemi ve Üç Çubuklu Makas Tasarım Optimizasyon Problemidir.

3.6.1.1. Kaynaklı Kiriş Tasarım Optimizasyon Problemi

Kaynaklı Kiriş Tasarım Optimizasyon probleminin çözümündeki temel amaç; minimum genel üretim maliyetleri ile belirli bir yük taşımak için uygun kaynaklı kiriş tasarımı yapmaktır. Problem 7 lineer olmayan eşitsizlik kısıtı ve 4 adet sürekli değişkenden oluşmaktadır (Sadollah ve ark., 2013).



Şekil 3.5. Kaynaklı Kiriş Tasarımı (Sadollah ve ark., 2013)

$$F(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14+x_2) \quad (3.18)$$

$$g_1(x)=\tau(x) - \tau_{max} \leq 0 \quad (3.19)$$

$$g_2(x)=\sigma(x) - \sigma_{max} \leq 0 \quad (3.20)$$

$$g_3(x)=x_1 - x_4 \leq 0 \quad (3.21)$$

$$g_4(x)=0.10471x_1^2 + 0.04811x_3x_4(14+x_2)-5 \leq 0 \quad (3.22)$$

$$g_5(x)=0.125 - x_1 \leq 0 \quad (3.23)$$

$$g_6(x)=\delta(x) - \delta_{max} \leq 0 \quad (3.24)$$

$$g_7(x)=P - P_c(x) \leq 0 \quad (3.25)$$

$$0.1 \leq x_i \leq 2 , i=1, 4 \quad (3.26)$$

$$0.1 \leq x_i \leq 10, i=2,3 \quad (3.27)$$

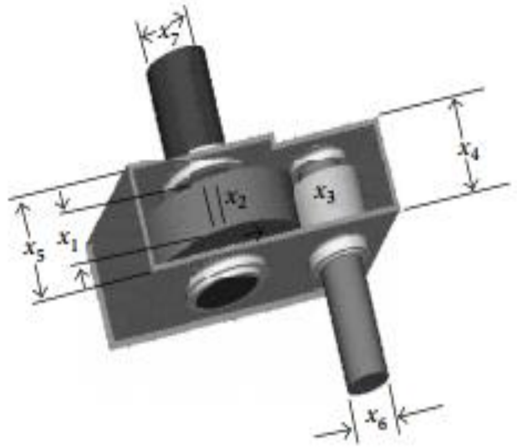
$$M = P\left(L + \frac{x^2}{2}\right) \quad (3.28)$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2} \quad (3.29)$$

$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1+x_2}{2}\right)^2\right]\right\} \quad (3.30)$$

3.6.1.2. Hız Düşürücü Tasarım Optimizasyon Problemi

Hız düşürücü tasarım probleminin çözümündeki temel amaç; verilen kısıtları yerine getirirken hız düşürücünün toplam ağırlığını en aza indirmektir. Problem 11 lineer olmayan eşitsizlik kısıtı ve 7 adet sürekli değişkenden oluşmaktadır.7 tasarım değişkeninden oluşması problem çözümünü zorlaştırmaktadır (Brajevic ve ark., 2010; Lin ve ark., 2013).



Şekil 3.6. Hız Düşürücü Tasarımı (Lin ve ark., 2013)

$$F(x) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) g_1(x) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0 \tag{3.31}$$

$$g_2(x) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0 \tag{3.32}$$

$$g_3(x) = \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \leq 0 \tag{3.33}$$

$$g_4(x) = \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \leq 0 \tag{3.34}$$

$$g_5(x) = \frac{\left[(745(x_4/x_2x_3))^2 + 16.9 \times 10^6\right]^{1/2}}{110x_6^3} - 1 \leq 0 \tag{3.35}$$

$$g_6(x) = \frac{\left[(745(x_5/x_2x_3))^2 + 157.5 \times 10^6\right]^{1/2}}{85x_7^3} - 1 \leq 0 \tag{3.36}$$

$$g_7(x) = \frac{x_2x_3}{40} - 1 \leq 0 \tag{3.37}$$

$$g_8(x) = \frac{5x_2}{x_1} - 1 \leq 0 \tag{3.38}$$

$$g_9(x) = \frac{x_1}{12x_2} - 1 \leq 0 \tag{3.39}$$

$$g_{10}(x) = \frac{1.5x_6+1.9}{x_4} - 1 \leq 0 \tag{3.40}$$

$$g_{11}(x) = \frac{1.1x_7+1.9}{x_5} - 1 \leq 0 \tag{3.41}$$

$$2.6 \leq x_1 \leq 3.6 \tag{3.42}$$

$$0.7 \leq x_2 \leq 0.8 \tag{3.43}$$

$$17 \leq x_3 \leq 28 \tag{3.44}$$

$$7.3 \leq x_4 \leq 8.3 \tag{3.45}$$

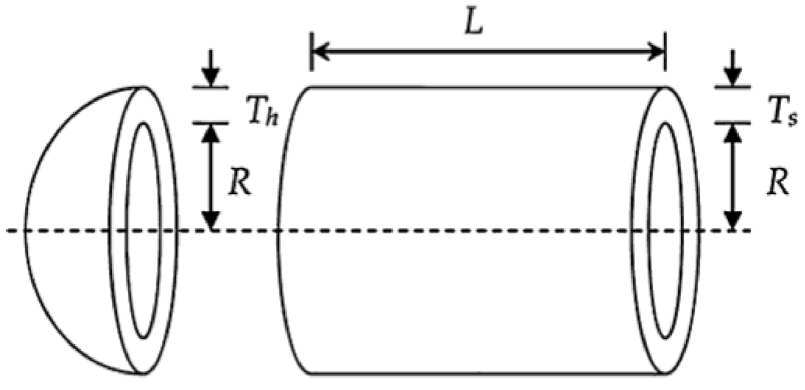
$$7.3 \leq x_5 \leq 8.3 \tag{3.46}$$

$$2.9 \leq x_6 \leq 3.9 \tag{3.47}$$

$$5.0 \leq x_7 \leq 5.5 \tag{3.48}$$

3.6.1.3.Basınçlı Kap Tasarım Optimizasyon Tasarım Problemi

Basınçlı kap tasarım problemini çözmedeki temel amaç verilen kısıtları yerine getirirken en düşük maliyetle üretime ulaşmaktır. Problem dört doğrusal olmayan eşitsizlik kısıtı ve dört sürekli değişkenden oluşmaktadır (Brajevic ve ark., 2010; Liu ve ark., 2010). Uymaz ve ark. (2015) tarafından yapılan çalışmada basınçlı kap tasarım problemi üzerine AAA testi yapılmıştır (Uymaz ve ark., 2015a).



Şekil 3.7. Basınçlı Kap Tasarımı(Liu ve ark., 2010)

$$f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \tag{3.49}$$

$$g_1(x) = -x_1 + 0.0193x \tag{3.50}$$

$$g_2(x) = -x_2 + 0.00954x_3 \leq 0 \tag{3.51}$$

$$g_3(x) = -\pi x_3^2 x_4 - \left(\frac{4}{3}\right)\pi x_3^3 + 1{,}296{,}000 \leq 0 \tag{3.52}$$

$g_4(x) = \ x_4 - 240 \le 0$ (3.53)

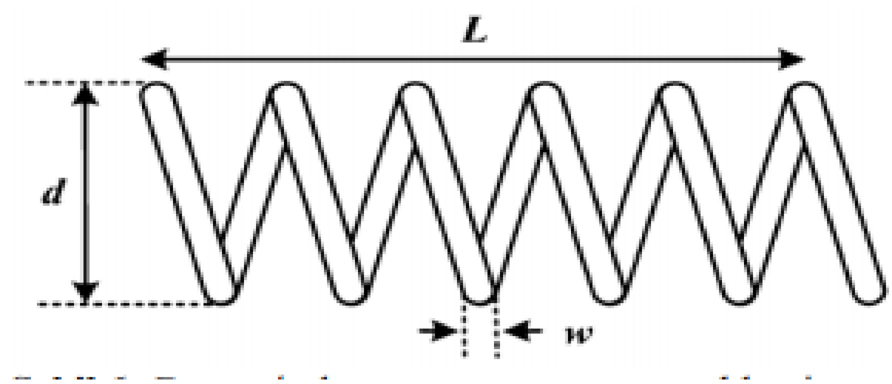
$1.125 \le x_1 \le 12.5$ (3.54)

$0.625 \le x_2 \le 12.5$ (3.55)

$0 \le x_i \le 240 \ , \qquad i = 3, 4$ (3.56)

3.6.1.4.Germe/Sıkıştırma Yay Tasarım Optimizasyon Problemi

Germe/sıkıştırma tasarım problem çözümünde temel amaç minimum değerleri elde ederek minimum ağırlıklı bir yay tasarımı yapmaktır. Problem ,4 lineer olmayan eşitsizlik kısıtı ve tel çapı w(x1), ortalama bobin çapı(x2) ,uzunluk veya bobin sayısı L(x3) olmak üzere 3 sürekli değişkenden oluşmaktadır (Shan ve ark., 2013).



Şekil 3.8. Germe/Sıkıştırma Yay Tasarımı (Shan ve ark., 2013)

$f(x) = \ (x_3 + 2)x_2x_1^2$ (3.57)

$g_1(x) = \ 1 - (x_2^3x_3/71{,}785x_1^4) \le 0$ (3.58)

$g_2(x) = \ (4x_2^2 - x_1x_2/12{,}566(x_2x_1^3 - x_1^4)) + (1/5108x_1^2) - 1 \le 0$ (3.59)

$g_3(x) = \ 1 - (140.45x_1/x_2^2x_3) \le 0$ (3.60)

$g_4(x) = (x_2 + x_1)/1.5 - 1 \le 0$ (3.61)

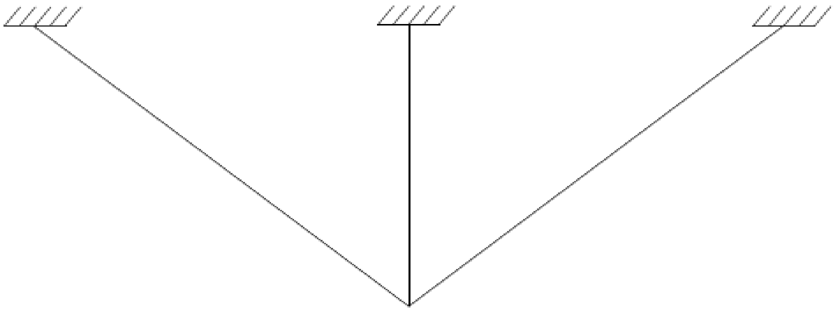
$0.05 \le x_1 \le 2.00$ (3.62)

$0.25 \le x_2 \le 1.30$ (3.63)

$2.00 \le x_3 \le 15.00$ (3.64)

3.6.1.5.Üç Çubuklu Makas Tasarım Optimizasyon Problemi

Üç çubuklu makas probleminde, makasın ağırlığını en aza indirmek amaçlanmıştır. Problem, doğrusal olmayan üç eşitsizlik kısıtı ve iki sürekli değişkenden oluşmaktadır (Liu ve ark., 2010).



Şekil 3.9. Üç Çubuklu Makas Tasarımı (Liu ve ark., 2010)

$$f(x) = \left(2\sqrt{2}x_1 + x_2\right) \times l \quad (3.65)$$

$$g_1(x) = \frac{\sqrt{2}x_1+x_2}{\sqrt{2}x_1^2+2x_1x_2}P - \sigma \leq 0 \quad (3.66)$$

$$g_2(x) = \frac{x_2}{\sqrt{2}x_1^2+2x_1x_2}P - \sigma \leq 0 \quad (3.67)$$

$$g_3(x) = \frac{1}{\sqrt{2}x_2+x_1}P - \sigma \leq 0 \quad (3.68)$$

$$0 \leq x_1 \leq 1 \quad (3.69)$$

$$0 \leq x_2 \leq 1 \quad (3.70)$$

$$l = 100cm \quad (3.71)$$

$$P = 2\frac{kN}{cm^3} \quad (3.72)$$

$$\sigma = 2kN/cm^2 \quad (3.73)$$

3.6.2.Kısıtlı Optimizasyon Fonksiyon Seti

Standart 13 Benchmark Test Fonksiyonu kullanılmıştır.2,3,8 ve 12. .fonksiyonlar maksimum noktayı bulmayı hedefleyen fonksiyonlar iken, geri kalan diğer 9 fonksiyon minimum noktayı bulmayı hedefleyen fonksiyonlardır. Maksimizasyon fonksiyonlarında optimum değer –f(x) şeklinde alınarak algoritmada minimizasyon fonksiyonu şeklinde değerlendirilmiştir (Runarsson ve Yao, 2000; Babalik ve ark., 2018). Çizelge 3.2’de gösterilen fonksiyon setine ait özelliklerde, LI özelliği doğrusal eşitsizlik, NI doğrusal olmayan eşitsizlik, LE doğrusal eşitlik ve NE doğrusal olmayan eşitlik anlamına gelmektedir. Fonksiyonların kısıtlarına dair bilgi vermektedir (Koziel ve Michalewicz, 1999; Xiao ve ark., 2014). Kullanılan kısıtlı fonksiyon setine dair formüller Ek-1 içerisinde sunulmuştur.

Çizelge 3.2. Kısıtlı optimizasyon problemleri için fonksiyon seti özellikleri(Koziel ve Michalewicz, 1999)

	Boyut	Problem Tip	LI	NI	LE	NE
G01(min)	13	Quadratic	9	0	0	0
G02(max)	20	Nonlinear	1	1	0	0
G03(max)	10	Nonlinear	0	0	0	1
G04(min)	5	Quadratic	0	6	0	0
G05(min)	4	Nonlinear	2	0	0	3
G06(min)	2	Nonlinear	0	2	0	0
G07(min)	10	Quadratic	3	5	0	0
G08(max)	2	Nonlinear	0	2	0	0
G09(min)	7	Nonlinear	0	4	0	0
G10(min)	8	Linear	3	3	0	0
G11(min)	2	Quadratic	0	0	0	1
G12(max)	3	Quadratic	0	9^3	0	0
G13(min)	5	Nonlinear	0	0	1	2

Çizelge 3.2’de verilen problemler üzerinde, algoritmada eşitsizlik kısıtları kullanıldığından eşitlik kısıtları eşitsizlik kısıtlarına dönüştürülmüştür.

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Bu bölümde kısıtlı optimizasyon problemleri için önerilen AAA_{dr}, AAA_{db} ve AAA_{e} algoritmalarına dair performans test sonuçları sunulmuştur. Statik penaltı yöntemi kullanılarak kısıtlı optimizasyon problemleri için uygun hale getirilmiş AAA ile,bir önceki bölümde açıklanan mühendislik tasarım optimizasyon problemleri üzerinde testler yapılmıştır. Bu testlerde, AAA'ya ait popülasyon sayısı ve A_p parametresinin değişimini etkisi üzerinde durulmuştur. Kısıtlı optimizasyon problemleri için önerilen algoritmaların testi için, literatürde sıkça kullanılan fonksiyon seti üzerindeki performans değerlendirmesi bulunmaktadır.

İlk olarak mühendislik tasarım optimizasyon problemlerinde AAA algoritması için önerilen Ap parametresi, popülasyon sayısı için çalışma yapılmıştır.

Kısıtlı optimizasyon fonksiyonlarından oluşan set üzerinde, mühendislik tasarım optimizasyon problemleri üzerinde test edilerek elde edilen parametrelerin kullanımı ile kısıtlı optimizasyon problemleri için önerilen AAA_{dr}, AAA_{db} ve AAA_{e} algoritmalarına dair testler yapılmıştır. Önerilen algoritmalar kendi aralarında kıyaslanmış ve ardından öne çıkan algoritma farklı algoritmalar ile karşılaştırılmıştır.

4.1. Mühendislik Tasarım Optimizasyon Problemlerinde Performans Değerlendirmesi

Mühendislik tasarım optimizasyon problemleri üzerinde *Ap* parametresinin ve popülasyon değişikliğinin performans üzerine etkisi gözlemlenmiştir. Kısıt işleme yöntemleri olarak, literatürdeki ilk yöntemlerden olan statik penaltı fonksiyonu kullanılmıştır. Algoritma üzerinde modifikasyon yapılmadı. Kısıt fonksiyonlarından dönen ceza değeri, amaç fonksiyon değerlerine eklenerek uygulama tipik kısıtsız problemlerde olduğu gibi optimum değeri aramaktadır. Popülasyon boyutu 40,60 ve 100 olarak denenmiştir. Adaptasyon parametresi olan A_p ise 0.5,0.75 ve 1 değerleri için AAA'nın performansı gözlemlenmiştir. Popülasyon hesaplama sayısı 100.000 alınmış ve tüm çalışmalar 30 çalışmanın ortalaması alınarak hesaplanmıştır. AAA'ya ait kesme kuvveti 2 ve enerji kaybı 0.3 olarak belirlenmiştir. Tüm testler Matlab (R2015a sürümü) programı üzerinde yapılmıştır. Elde edilen sonuçlar Çizelge 4.1 ve Çizelge 4.5 arasında sunulmuştur.

Çizelge 4.1. Kaynaklı Kiriş Tasarım Optimizasyon Problemi İçin AAA

	40n			60n			100n		
	Ap =0.5	*Ap* =0.75	*Ap* =1	*Ap* =0.50	*Ap* =0.75	*Ap* =1	*Ap* =0.50	*Ap* =0.75	*Ap* =1
En iyi	1.724	1.724	1.724	1.724	1.724	1.724	1.724	1.724	1.724
Ort.	1.732	1.751	1.732	1.725	1.729	1.736	1.729	1.726	1.725
S.S.	2.319e-02	5.881E-02	1.875E-02	2.533E-03	1.471E-02	3.133E-02	1.699E-02	2.731E-03	2.082E-03
FFevs	**92410**	100000	98510	96703	99726	99965	99884	100000	100000

Çizelge 4.1'de gösterilen sonuçlar ışığında, kaynaklı kiriş tasarım optimizasyon probleminde en iyi değerin *Ap* parametresinin tüm durumlarında ve popülasyon sayısının tüm durumlarında bulunduğu görülmektedir. Yapılan teste göre, en iyi değeri en az fonksiyon hesaplama sayısı ile bulan parametreler; popülasyon sayısının 40 olduğu *Ap* =0.5 durumudur.

Ortalamaya bakıldığında, 60 popülasyon sayısı ve *Ap* =0.75 durumu ile 100 popülasyon sayısı ve *Ap* =1 değerinin en iyi değeri bulduğu görülmektedir. Bu iki değer arasında karşılaştırma yapıldığında, en iyi değeri en düşük fonksiyon hesaplama sayısı ile bulan 60 popülasyon, *Ap* =0.50 parametreleri olduğu gözlemlenmiştir.

Standart sapmaları göz önüne alındığında, 100 popülasyon sayısı ve *Ap*=1 parametresinde en düşük standart sapma değerine ulaşıldığı görülmektedir.

Çizelge 4.2. Hız Düşürücü Tasarım Optimizasyon Problemi İçin AAA

	40n			60n			100n		
	Ap =0.50	*Ap* =0.75	*Ap* =1	*Ap* =0.50	*Ap* =0.75	*Ap* =1	*Ap* =0.50	*Ap* =0.75	*Ap* =1
En iyi	2996.348	2996.348	2996.348	2996.348	2996.348	2996.348	2996.348	2996.348	2996.348
Ort.	2996.348	2996.348	2996.348	2996.348	2996.348	2996.348	2996.348	2996.348	2996.348
S.S.	1.581E-07	1.708E-07	2.028E-07	1.975E-07	2.170E-07	1.733E-07	1.494E-07	1.591E-07	1.338E-07
FFevs	10697	**10176**	10439	15540	16494	15677	26019	25550	25836

Hız düşürücü tasarım optimizasyon problemi üzerinde yapılan test sonuçları Çizelge 4.3'te verilmiştir. Verilen sonuçlara göre, en iyi değer tüm *Ap* parametreleri ve tüm popülasyon değerleri için bulunmuştur. En düşük fonksiyon hesaplama sayısı ile en iyi değeri bulan parametre ayarları, 40 popülasyon ve *Ap*=0.75'dir.

Ortalama değerler göz önüne alındığında, tüm *Ap* parametrelerinde ve tüm popülasyon değerlerinde en iyi ortalamanın elde edildiği gözlemlenmektedir. Bu değerler arasında karşılaştırma yapıldığında, en iyi ortalama değere sahip ve en az fonksiyon hesaplama sayısı ile optimumu bulan parametreler *Ap*=0.75 ve 40 popülasyon sayısı olduğu gözlemlenmiştir.

Standart sapmanın en düşük olduğu 60 popülasyon sayısı ve *Ap*=1 parametresidir.

Çizelge 4.3. Basınçlı Kap Tasarım Optimizasyon Problemi İçin AAA

	40n			60n			100n		
	Ap =0.50	*Ap* =0.75	*Ap* =1	*Ap* =0.50	*Ap* =0.75	*Ap* =1	*Ap* =0.50	*Ap* =0.75	*Ap* =1
En iyi	7197.728	7197.728	7197.728	7197.728	7197.728	7197.728	7197.728	7197.728	7197.728
Ort.	7197.728	7197.728	7201.209	7197.728	7197.728	7197.728	7197.728	7197.728	7197.728
S.S.	2.740E-06	3.541E-06	1.906E+01	2.815E-06	2.389E-06	2.935E-06	4.302E-05	3.052E-05	3.842E-05
FFevs	**53047**	61749	100000	71358	88956	68629	99014	98449	99997

Basınçlı kap tasarım optimizasyon problemi üzerindeki AAA’nın parametre değişimlerine dair test sonuçları Çizelge 4.3’te sunulmuştur. Tüm *Ap* parametrelerinde ve tüm popülasyon sayılarında en iyi değer elde edilmiştir. En iyi değeri, en düşük hesaplama sayısı ile bulan değerler; 40 popülasyon ve *Ap*=0.5 değerleri olmuştur.

Ortalama değerlere bakıldığında, 40 popülasyon ve *Ap*=1 değeri haricinde diğer parametre değerlerinde en iyi sonucun bulunduğu ve eşit olduğu görülmektedir. Eşit olan bu değerler arasında karşılaştırma yapılacak olursa, en az fonksiyon hesaplama sayısına sahip A=0.50 ve 40 popülasyon değerlerinin öne çıktığı görülmektedir.

Standart sapma değerleri incelendiğinde, en düşük standart sapmanın *Ap*=0.5 ve 60 popülasyonda olduğu, en büyük standart sapma değerinin ise *Ap*=1 ve 40 popülasyonda elde edildiği gözlemlenmektedir.

Çizelge 4.4. Germe/Sıkıştırma Yay Tasarım Optimizasyon Problemi İçin AAA

	40n			60n			100n		
	Ap =0.50	*Ap* =0.75	*Ap* =1	*Ap* =0.50	*Ap* =0.75	*Ap* =1	*Ap* =0.50	*Ap* =0.75	*Ap* =1
En iyi	0.012	0.012	0.012	0.012	0.012	0.012	0.012	0.012	0.012
Ort.	6.666E+04	0.012	6.666E+04	0.012	0.012	0.012	0.012	0.012	0.012
S.S.	2.537E+05	1.506E-04	2.537E+05	1.199E-04	1.315E-04	1.354E-04	1.274E-04	1.631E-04	9.141E-05
FFevs	100000	**24014**	100000	27974	100000	100000	100000	100000	100000

Çizelge 4.4'te verilen germe/sıkıştırma yay tasarım problemi üzerinde yapılan AAA test sonuçları incelendiğinde, en iyi değerin tüm *Ap* ve tüm popülasyon sayısında elde edildiği görülmektedir. En iyi değerin tüm parametre değişimlerinde bulunduğu bu durumda, en düşük fonksiyon hesaplama sayısına sahip parametreler *Ap* =0.75 ve 40 popülasyon olmuştur. Bunu izleyen 60 popülasyon ve *Ap* =0.50 değerinin ardından diğer parametreler geride kalmıştır.

Ortalama değerler dikkate alındığında,40 popülasyon ve *Ap* =0.50, *Ap* =1 değerleri dışında diğer parametrelerin en iyi sonuca ortalama değerde de ulaştığı görülmektedir. Ortalamada da en iyi sonucu bulan bu parametreler fonksiyon hesaplama sayısı dikkate alınarak karşılaştırıldığında, *Ap* =0.75 ve 40 popülasyon değerlerinin daha iyi olduğu gözlemlenmiştir.

Standart sapma değerlerine göre, *Ap* =1 ve 100 popülasyonda bulunan sonuçların standart sapmasının en düşük olduğu gözlemlenmiştir.40 popülasyon ve *Ap* =1, 0.50 değerlerinde standart sapması en yüksektir.

Çizelge 4.5. Üç Çubuklu Makas Tasarım Optimizasyon Problemi İçin AAA

	40 n			60n			100n		
	Ap =0.50	*Ap* =0.75	*Ap* =1	*Ap* =0.50	*Ap* =0.75	*Ap* =1	*Ap* =0.50	*Ap* =0.75	*Ap* =1
En iyi	263.895	263.895	263.895	263.895	263.895	263.895	263.895	263.895	263.895
Ort.	263.895	263.895	263.895	263.895	263.895	263.895	263.895	263.895	263.895
S.S.	8.737E-05	5.988E-05	1.462E-04	3.008E-05	3.660E-05	1.036E-04	2.644E-05	5.122E-05	3.789E-05
FFevs	58749	76630	**18160**	82605	76581	93710	100000	100000	96736

Çizelge 4.5’de gösterilen sonuçlar ışığında, üç çubuklu makas tasarım optimizasyon probleminde en iyi değeri tüm parametre değerlerinin elde ettiği gözlemlenmiştir. En iyi değerler arasında en az fonksiyon hesaplama sayısının 40 popülasyon ve *Ap* parametresinin 1 olduğu durumda elde edildiği gözlemlenmektedir.

Ortalama değerin tüm parametre değerlerinde elde edilmiştir ve bu değerler en az fonksiyon hesaplama sayısı ile kıyaslandığında, *Ap* parametresinin 1 olduğu 40 popülasyon değeri öne çıkmaktadır.

Standart sapma değerleri arasında, en düşük standart sapma değerine sahip *Ap*=0.50 ve 100 popülasyon değerinin olduğu görülmektedir.

4.2. Kısıtlı Optimizasyon Fonksiyon Setinde Performans Değerlendirmesi

Kullanılan Benchmark fonksiyon seti optimumu minimum ve maksimum olan 13 adet kısıtlı fonksiyondan oluşmaktadır. Popülasyon sayısı 40 olarak belirlenmiştir. Fonksiyon hesaplama sayısı durdurma kriteri olarak belirlenmiş ve diğer çalışmalar göz önüne alınarak 240.000 olarak alınmıştır. Başlangıçta oluşturulan popülasyon bireyleri, her bir fonksiyona ait alt ve üst sınırlar arasında rastgele değerler alınarak oluşturulmuştur. Her bir fonksiyon için 30 çalışmanın ortalaması alınarak sonuçlar elde edilmiştir. AAA için daha önce yapılan çalışmalarda sunulan parametre değerleri kullanılmıştır. Enerji kaybı 0.3, kesme kuvveti 2 ve Adaptasyon katsayısı 0.5 belirlenmiştir. AAAϵ algoritmasında ϵ değeri başlangıçta 0 olarak alınmıştır. T_c değeri 0.8 ve cp=2 olarak belirlenmiştir. AAA_{dr} ve AAA_{dp} algoritmalarında bu tarz özel ayarlamalar bulunmamaktadır. Tüm testler Matlab (R2015a sürümü) programı üzerinde yapılmıştır. Çizelge 4.6’da $AAA_{\boldsymbol{dr}}$’nin, Çizelge 4.7’de $AAA_{\boldsymbol{dp}}$’nin ve Çizelge 4.8’de $AAA_{\boldsymbol{\epsilon}}$ ‘ın kısıtlı fonksiyon seti üzerindeki sonuçları verilmiştir.

Çizelge 4.6. AAA_{dr} algoritmasının fonksiyon seti üzerindeki sonuçları

	Optimal	Ortalama	En iyi	En kötü	Standart S.	Medyan
G01	-15.000	-1.500e+01	-1.500e+01	-1.500e+01	0.000e+00	-1.500e+01
G02	0.803619	7.925e-01	8.032e-01	7.848e-01	6.176e-03	7.922e-01
G03	1.000	3.140e-04	3.140e-04	0.000e+00	9.932e-05	0.000e+00
G04	-30665.53	-3.066e+04	-3.066e+04	-3.066e+04	1.212e-12	-3.066e+04
G05	5126.498	5.373e+03	5.126e+03	6.112e+03	3.434e+03	5.252e+03
G06	-6961.814	-6.961e+03	-6.961e+03	-6.961e+03	1.917e-12	-6.961e+03
G07	24.306	2.454e+01	2.438e+01	2.481e+01	1.614e-01	2.451e+01
G08	0.095825	9.582e-02	9.582e-02	9.582e-02	1.387e-17	9.582e-02
G09	680.630	6.806e+02	6.806e+02	6.806e+02	1.029e-02	6.806e+02
G10	7049.250	7.234e+03	7.053e+03	7.414e+03	1.037e+02	7.245e+03
G11	0.750	9.288e-01	7.501e-01	9.999e-01	7.642e-02	9.347e-01
G12	1.000	1.000e+00	1.000e+00	1.000e+00	0.000e+00	1.000
G13	0.05395	9.617e-01	5.127e-01	1.103e+00	1.611e-01	9.997e-01

AAA_{dr}’nin kısıtlı fonksiyon seti üzerinde uygulanması sonucunda performansı Çizelge 4.6’da sunulmuştur. Çizelge 4.6’da optimal değer olarak gösterilenler, fonksiyonların ulaşılması gereken en iyi çözümleridir. Ortalama, en iyi ,en kötü, standart sapma ve medyan değerleri önerilen algoritmanın her bir problem için sonucunu göstermektedir.

13 test fonksiyonundan 6’sında ortalama değer olarak optimum noktayı bulmuştur. Algoritma 3.fonksiyonda başarılı bir performans gösteremezken, 10. Fonksiyonda ve 11. Fonksiyonda en iyi değerlere yaklaşmasına rağmen ortalama değer optimum noktayı elde edememiştir. 13.fonksiyonda algoritma optimum noktaya yakınlaşmış ancak en iyi değerinde de bulamamıştır. Önerilen algoritma geriye kalan fonksiyonlarda uygun çözüme çok yaklaşmıştır.

Çizelge 4.7.AAA_{dp} algoritmasının fonksiyon seti üzerindeki sonuçları

	Optimal	Ortalama	En iyi	En kötü	Standart S.	Medyan
G01	-15.000	-1.266e+01	-1.435e+01	-1.160e+01	7.035e+01	-1.257e+01
G02	0.803619	7.522e-01	9.924e-01	6.640e-01	3.317e-02	7.583e-01
G03	1.000	1.529e-02	6.064e-02	4.800e-22	2.099e-02	-3.993e-03
G04	-30665.539	-3.068e+04	-3.085e+04	-3.066e+04	4.487e+01	-3.066e+04
G05	5126.498	2.713e+07	1.790e+06	7.318e+07	1.955e+07	2.360e+07
G06	-6961.814	-7.954e+03	-7.970e+03	-7.970 e+03	1.590e+01	- 7.960e+03
G07	24.306	6.075e+02	2.535e+01	1.097e+03	2.912e+02	6.316e+02
G08	0.095825	9.095e+01	5.874e+02	9.582e-02	1.478e+02	3.300e+01
G09	680.630	6.806e+02	6.806e+02	6.807e+02	3.338e-02	6.806e+02
G10	7049.250	1.004e+04	7.274e+03	1.738e+04	2.613e+03	9.233e+03
G11	0.750	7.522e-01	7.498e-01	7.982e-01	8.982e-03	7.499e-01
G12	1.000	1.000e+00	1.000e+00	1.000e+00	0.000e+00	1.000e+00
G13	0.053	1.831e+00	5.976e-01	2.724e+00	4.215e-01	1.844e+00

Çizelge 4.7'de verilen dinamik penaltı yönteminin uyarlanması ile ortaya çıkan AAA_{dp} 'nin kısıtlı fonksiyonlar üzerindeki performansı ortaya konmuştur. Optimal değeri, çözüm aranan problemin ulaşılması gereken en iyi çözümü iken, çizelgedeki diğer sonuçlar önerilen algoritmanın bulduğu sonuçlardır.

AAA_{dp}, algoritma fonksiyonların çoğunda optimum değere yaklaşmasına rağmen tam olarak istenen değerler bulunamamıştır. 1,2 ve 6. fonksiyonlarda en iyi değer optimum değere çok yakınken, ortalama değer başarısını gösterememiştir. G12 fonksiyonunda en iyi çözüm en iyi ve ortalamada elde edildiği gözlemlenmiştir.

Çizelge 4.8. AAA_{ε} algoritmasının fonksiyon seti üzerindeki sonuçları

	Optimal	Ortalama	En iyi	En kötü	Standart S.	Medyan
G01	-15.000	-1.493e+01	-1.498e+01	-1.480e+01	4.201e-02	-1.494e+01
G02	0.803619	7.428e-01	7.624e-01	7.239e-01	8.839e-03	7.422e-01
G03	1.000	1.365e-04	4.097e-03	0.000e+00	7.480e-04	0.000e+00
G04	-30665.539	-3.066e+04	-3.066e+04	-3.065e+04	2.124e+00	-3.066e+04
G05	5126.498	5.475e+03	5.116e+03	6.125e+03	3.987e+02	5.264e+03
G06	-6961.814	-6.407e+03	-6.843e+03	-5.277e+03	3.651e+02	-6.526e+03
G07	24.306	2.911e+01	2.721e+01	3.176e+01	1.160e+00	2.886e+01
G08	0.095825	9.576e-02	9.582e-02	9.558e-02	6.362e-05	9.578e-02
G09	680.630	6.844e+02	6.821e+02	6.867e+02	1.399e+00	6.842e+02
G10	7049.250	8.144e+03	7.642e+03	8.671e+03	2.634e+02	8.092e+03
G11	0.750	7.640e-01	7.499e-01	8.887e-01	2.776e-02	7.539e-01
G12	1.000	9.999e-01	9.999e-01	9.999e-01	9.721e-06	9.999e-01
G13	0.05395	8.472e-01	5.910e-02	1.276e+00	2.453e-01	9.235e-01

Çizelge 4.8 ‘de $AAA_{\boldsymbol{\varepsilon}}$ ile kısıtlı optimizasyon problem seti üzerinde yapılan testler ortaya konmuştur. Optimal değer Çizelge 4.6 ve Çizelge 4.7’de olduğu gibi problemin en uygun çözümünü ifade etmektedir. $AAA_{\boldsymbol{\varepsilon}}$’nın Çizelge 4.8’deki sonuçları göz önüne alınarak, ortalama ve en iyi değerlerinde, optimum noktaya yakın olanı bulabilen sonuçlar ortaya koyduğu söylenebilir. Bulduğu sonuçlar arasında en uygun çözümler olmasa da, optimum ile aralarında çok fark bulunmamaktadır. Önerilen algoritma, G04’te ortalama ve en iyi değerlerimde optimum noktayı bulmuştur. G01, G02, G05, G08, G09, G11 ve G12 problemlerinde ortalama ve en iyi değerlerde çözüme oldukça yakın sonuçlar ortaya konmuştur.

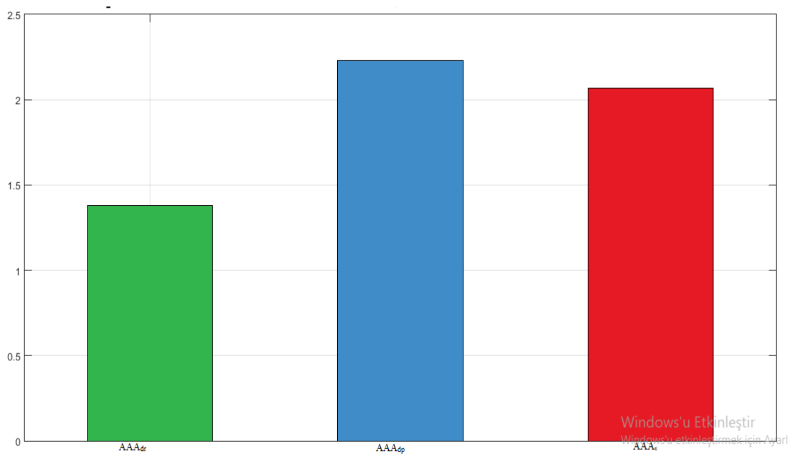
Çizelge 4.8 ile gösterilen sonuçlar doğrultusunda, bu metodun başarısının arttırılabileceği öngörülebilir.

Çizelge 4.9’da AAA_{dr}, AAA_{dp}ve AAA_{ϵ} algoritmaları karşılaştırılmıştır.

Çizelge 4.9. AAA_{dr}, AAA_{dp} ve $AAA_{\varepsilon c}$ algoritmalarının karşılaştırılması

	Optimal	AAA_{dr}	AAA_{dp}	AAA_{ε}
G01	-15.000	-15.000(1)	-12.66(3)	-14.9365(2)
G02	0.803619	0.792527(1)	0.7522(2)	0.74287(3)
G03	1.000	0.0000314(3)	0.001529(1)	0.000136(2)
G04	-30665.539	-30665.538(1)	-30683.670(3)	-30662.313(2)
G05	5126.498	5373.836(1)	0.000027(3)	5475.414(2)
G06	-6961.814	-6961.813(1)	-7954.599(3)	-6407.582(2)
G07	24.306	24.381(1)	607.5(3)	29.91154(2)
G08	0.095825	0.095825(1)	0.095825(1)	0.095765(2)
G09	680.630	680.6494(1)	680.666(2)	684.450(3)
G10	7049.250	7234.8498(1)	10043.51(3)	8144.132(2)
G11	0.750	0.9288(3)	0.7522(1)	0.764072(2)
G12	1.000	1.000(1)	1.000(1)	0.99998(2)
G13	0.05395	0.96174(2)	1.831(3)	0.847227(1)
Ort.Rank		1.38(18/13)	2.23(29/13)	2.07(27/13)

Kısıtlı optimizasyon problemleri için önerilen 3 algoritma Çizelge 4.9'da karşılaştırılmıştır. AAA_{dr} ile test fonksiyonlarının çözümünde AAA_{dp} ve AAA_{ε} algoritmalarını geride bırakmıştır. AAA_{dp} 3.fonksiyonda diğer iki algoritmadan daha iyi sonuç vermiştir.8. fonksiyonda AAA_{dr} ve AAA_{dp} algoritmalarının ikisi uygun çözümü bulmaktadır. Diğer test fonksiyonlarında AAA_{dr} uygun çözümleri ya da uygun çözümlere en yakın değerleri bulmuştur.



Şekil 3.10. AAA_{dr}, AAA_{dp} ve AAA_{ε} algoritmalarının ranking ortalamaları

Şekilde 3.10 'da verilen sıralama ortalamaları, çizelge 4.9'dan çıkarılan ortalama sıralama değerinin grafiğidir. Şekil 3.10 dikkate alınarak, fonksiyon bazlı bakmayıp genel olarak tüm fonksiyonlar ele alındığında, AAA_{dr} algoritmasının diğer iki algoritmadan daha başarılı olduğu görülmektedir. AAA_{ϵ} küçük bir fark ile, AAA_{dp} 'yi geçmektedir.

Çizelge 4.10'da 30 çalışma üzerinden ortalama değerler dikkate alınarak, AAA_{dr} algoritmasının ABC, PSO, GA, DE ve CTSA algoritmaları ile kıyaslanması sunulmuştur.

Bu kıyaslamalarda, kullanılan algoritmalar ve sonuçları (Babalik ve ark., 2018) tarafından yapılan çalışmadan alınmıştır. Kıyaslamaların adil olabilmesi için, AAA_{dr} yönteminde kullanılan maksimum fonksiyon hesaplama sayısı diğer yöntemlerle eşit olarak (240000FEvs) belirlenmiştir.

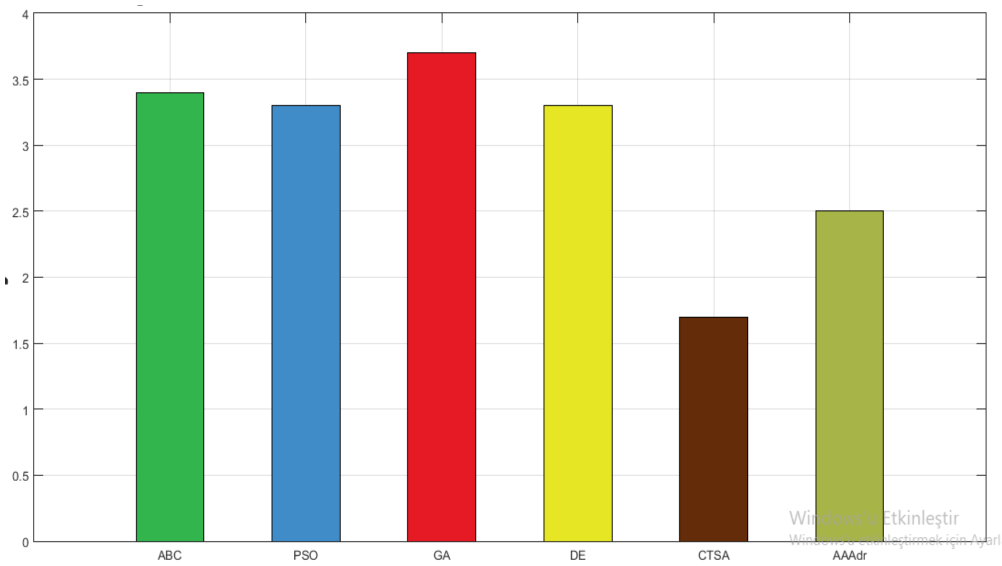
Çizelge 4.10. $\mathrm{AAA}_{dr}{}'nin$ diğer algoritmalarla karşılaştırılması

	Optimal	ABC	PSO	GA	DE	CTSA	AAA_{dr}
G01	-15.000	-15.0205	-10.5551	-14.236	-14.2406	-15.000	-15.000
Rank		2	5	4	3	1	1
G02	0.803619	0.4795	0.4043	0.788588	0.6660	0.8005	0.792527
Rank		5	6	3	4	1	2
G03	1.000	3.0191	1.1675	0.976	1.1694	1.0158	0.0000314
Rank		6	3	2	4	1	5
G04	-30665.53	-30610.9	-30661.7	-30590.4	-30665.54	-30665.5	-30665.5
Rank		3	2	4	1	1	1
G05	5126.498	5115.056	5298.284	N/A	5329.197	5172.377	5373.836
Rank		1	3	6	4	2	5
G06	-6961.81	-7579.63	-6961.81	-6872.20	-6765.48	-6961.81	-6961.81
Rank		5	2	3	4	1	1
G07	24.306	29.0956	28.7418	34.980	24.3160	24.5008	24.381
Rank		5	4	6	1	3	2
G08	0.095825	6.5347	0.0847	0.0958	0.0958	0.0958	0.095825
Rank		3	2	1	1	1	1
G09	680.630	683.8941	680.7815	692.064	680.6308	680.6433	680.6494
Rank		5	4	6	1	2	3
G10	7049.250	7259.028	8128.793	10003.225	7162.592	7116.206	7234.8498
Rank		4	5	6	2	1	3
G11	0.750	0.7171	0.7626	0.750	0.9545	0.8003	0.9288
Rank		3	2	1	6	4	5
G12	1.000	1.0001	1.000	1.000	1.000	1.000	1.000
Rank		2	1	1	1	1	1
G13	0.05395	0.0955	1.4228	N/A	0.9492	0.9671	0.96174
Rank		1	5	6	2	4	3
Ort. Rank		3.4(45/13)	3.3(44/13)	3.7(49/13)	3.3(43/13)	1.7(23/13)	2.5(33/13)

Çizelge 4.10’da 30 çalışma üzerinden ortalama değerler dikkate alınarak, AAA_{dr} algoritmasının ABC, PSO, GA, DE ve CTSA algoritmaları ile kıyaslanması sunulmuştur.

Bu kıyaslamalarda, kullanılan algoritmalar ve sonuçları (Babalik ve ark., 2018) tarafından yapılan çalışmadan alınmıştır. Kıyaslamaların adil olabilmesi için, AAA_{dr} yönteminde kullanılan maksimum fonksiyon hesaplama sayısı diğer yöntemlerle eşit olarak (240000FEvs) belirlenmiştir.

Çizelge 4.10’da yapılan karşılaştırma ile daha önceki çalışmalardan alınan sonuçlar ve bu çalışmada önerilen algoritmalardan başarısı yüksek olan AAA_{dr}’nin performansları gösterilmiştir. Birinci fonksiyonda CTSA ve AAA_{dr} en uygun çözümü bularak diğer algoritmalardan öne çıkmıştır. İkinci fonksiyonun en uygun değerine yaklaşan ilk algoritma CTSA iken onu takip eden ikinci algoritma AAA_{dr}’dir.3.fonksiyon bir maksimizasyon fonksiyonudur ve en uygun çözüme yaklaşan en iyi algoritma GA olmuştur. En uygun çözüme yakın değer bulan ikinci algoritma AAA_{dr}’dır. 4 ve 6 numaralı fonksiyonlarda AAA_{dr} en yakın çözümü bulan algoritma olmuştur. 5.fonksiyonda AAA_{dr} sonuca en çok yakınlaşan ikinci fonksiyonken, diğer algoritmalar arasında en yakın değeri bulan CTSA olmuştur.8.fonksiyonda CTSA, DE ve AAA_{dr} en uygun çözümü bulmuştur ve fonksiyon 12’de ABC dışında tüm fonksiyonlar en iyi çözüme ulaşmıştır. Fonksiyon 13 için en iyi çözüme yaklaşan algoritmanın ABC algoritması olduğu görülmüştür.



Şekil 3.11. Algoritmaların ranking işleminin ortalaması

Şekil 3.11’de gösterilen grafikte, algoritmaların ortalamalarına göre yapılan sıralama ifade edilmektedir. Bu grafiğe bakılarak, AAA_{dr}’nin CTSA’ dan sonra en iyi sonuçları bulan algoritma olduğu söylenebilir.

5. SONUÇLAR VE ÖNERİLER

5.1 Sonuçlar

Optimizasyon uygun koşullar altında en iyi durumu bulma işlemdir. Günümüzde karşımıza çıkan bir çok problemde, optimizasyon yöntemi uygulanabilir. Sınırlı kaynak ile verimli üretim sağlamak hedeflenirken, kısıtlara takılmamak gerekmektedir.

Problemlerin çözümünde matematiksel yöntemlerin eksik kaldığı yerde, sezgisel yöntemler uygulanmaya başlamıştır. Bunun en büyük nedeni problemlerin karmaşıklaşması ve matematiksel modellerin çözüm uzayını sınırlandırarak varsayımlar üzerine gitmesidir. Sezgisel algoritmaların problemlere uygulanmasının zorlaştığı noktada, metasezgisel kavramı ortaya çıkmış ve bu alanda çalışmalar yapılmıştır.

Ortaya çıkan bir çok metasezgisel optimizasyon yöntemi kısıtsız problemler için tasarlandığından kısıtlı problemlerin çözümünde eksik kalmaktadır. Bu sorunu aşmak için kısıt işleme teknikleri önerilmiştir.

Literatürde kullanılan birçok kısıt işleme tekniği bulunmaktadır. Bu teknikler metasezgisel algoritmalar ile birleşerek arama uzayında, algoritmaların kısıtlı alanlara takılmamasını hedeflemektedir. Tekniklerin her biri farklı avantaj ve dezavantajlara sahiptir. Kısıtlı optimizasyon problemleri için yapılan birçok çalışmada, bu tekniklerin performansları ortaya konmuştur.

AAA mikro alglerin davranışlarından yola çıkılarak optimum değeri bulmayı hedefleyen metasezgisel bir algoritmadır. Algoritmanın kısıtsız problemlerde başarısı ortaya koyulmuştur. Bu çalışmada, kısıt işleme yöntemleri ile AAA birleştirilerek kısıtlı optimizasyon problemlerini çözebilecek şekilde geliştirilmiş Yapay Alg Algoritmaları sunulmuştur.

Literatürdeki çalışmalar dikkate alındığında başarısı ortaya konan 3 kısıt işleme tekniği seçilmiş ve bu kısıt işleme teknikleri AAA’ya uyarlanmıştır. Deb’s Rules kısıt işleme tekniği ile birleştirilmiş AAA, AAA_{dr} olarak önerilmiştir. Dinamik Penaltı kısıt işleme tekniğinin algoritmaya uyarlanması sonucunda AAA_{dp} ve ε-kısıt işleme tekniği ile uyarlanmış algoritma sonucu AAA_{ε} algoritması sunulmuştur.

Yapılan çalışmalarda öncelikle AAA üzerinde parametrelerden A_p kesme kuvveti değişimi 0.5, 0.75 ve 1 alınarak en iyi parametre değeri gözlemlenmiştir. Mühendislik tasarım optimizasyon problemleri üzerinde yapılan bu testte, aynı zamanda popülasyon büyüklüğünün performans üzerindeki etkisi gözlemlenmiştir. A_p kesme kuvveti parametresinin problem bağımlı olarak değiştiği gözlemlenmiştir ve popülasyonun 40 olarak alındığında daha başarılı olduğu kanıtlanmıştır.

Mühendislik tasarım optimizasyon problemleri üzerinde yapılan çalışma sonucundaki veriler ve daha önceden AAA üzerine yapılan çalışmalar dikkate alınarak parametre ayarları yapılmış ve kısıtlı optimizasyon problemleri için tasarlanan $AAA_{dr}, AAA_{dp}, AAA_{\varepsilon}$ algoritmalarının kısıtlı fonksiyon seti üzerinde performans değerlendirmesi yapılmıştır. AAA_{dr}'nin diğer iki algoritmadan daha iyi sonuçlar ortaya koyduğu gözlemlenmiştir. ABC, PSO, DE, GA, DE ve CTSA algoritmaları ile kıyaslanarak AAA_{dr} algoritmasının başarısı sunulmuştur. AAA_{dr} yapılan testler sonucunda rekabetçi sonuçlar ortaya koymuş ve alternatif olarak kullanılabilecek bir algoritma olmuştur.

5.2 Öneriler

AAA'nın kısıtlı optimizasyon problemleri üzerindeki başarısı bu çalışmada ortaya konmuştur. Farklı metotların uygulanması ile optimizasyon problemlerini çözmedeki başarısı arttırılabilir. Aynı zamanda yaygın olarak kullanılan hibrit yöntemler uygulanarak algoritmanın başarısı arttırılabilir. Hibrit yöntemler, kullandıkları yöntemlerin avantaj ve dezavantajlarını kullanarak daha verimli sonuçlar ortaya çıkarabilir.

KAYNAKLAR

Babalik, A., Cinar, A. C. ve Kiran, M. S., 2018, A modification of tree-seed algorithm using Deb's rules for constrained optimization, *Applied Soft Computing*, 63, 289-305.

Brajevic, I., Tuba, M. ve Subotic, M., 2010, Improved Artificial Bee Colony Algorithm for Constrained Problems, *Recent Advances in Neural Networks, Fuzzy Systems & Evolutionary Computing*, 185-+.

Brajevic, I., Tuba, M. ve Bacanin, N., 2012, Firefly Algorithm with a Feasibility-Based Rules for Constrained Optimization, p.

Chehouri, A., Younes, R., Perron, J. ve Ilinca, A., 2016, A Constraint-Handling Technique for Genetic Algorithms using a Violation Factor, *JCS*, 12, 350-362.

Coello, C. A. C., 2002, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, *Computer Methods in Applied Mechanics and Engineering*, 191 (11-12), 1245-1287.

Coello Coello, C. A., 2000, Use of a self-adaptive penalty approach for engineering optimization problems, *Computers in Industry*, 41 (2), 113-127.

Cuevas, E. ve Cienfuegos, M., 2014, A new algorithm inspired in the behavior of the social-spider for constrained optimization, *Expert Systems with Applications*, 41 (2), 412-425.

Deb, K., 2000, An efficient constraint handling method for genetic algorithms, *Computer Methods in Applied Mechanics and Engineering*, 186 (2-4), 311-338.

Gyu Kim, D. ve Husbands, P., 1998, Mapping Based Constraint Handling for Evolutionary Search; Thurston's Circle Packing and Grid Generation, p.

Hamida, S. B. ve Schoenauer, M., 2002, ASCHEA: New results using adaptive segregational constraint handling, *Cec'02: Proceedings of the 2002 Congress on Evolutionary Computation, Vols 1 and 2*, 884-889.

Jin, Y., 2005, A comprehensive survey of fitness approximation in evolutionary computation, *Soft Computing*, 9 (1), 3-12.

Joines, J. ve R. Houck, C., 1996, On the Use of Non-Stationary Penalty Functions to Solve Nonlinear Constrained Optimization Problems with GA's, p.

Karaboga, D. ve Akay, B., 2011, A modified Artificial Bee Colony (ABC) algorithm for constrained optimization problems, *Applied Soft Computing*, 11 (3), 3021-3031.

Koziel, S. ve Michalewicz, Z., 1999, Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization, *Evolutionary Computation*, 7 (1), 19-44.

Kuri, A. ve Gutiérrez-García, J., 2002, Penalty Function Methods for Constrained Optimization with Genetic Algorithms: A Statistical Analysis, p.

Lin, M. H., Tsai, J. F., Hu, N. Z. ve Chang, S. C., 2013, Design Optimization of a Speed Reducer Using Deterministic Techniques, *Mathematical Problems in Engineering*.

Liu, H., Cai, Z. X. ve Wang, Y., 2010, Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization, *Applied Soft Computing*, 10 (2), 629-640.

Long, Q., 2014, A constraint handling technique for constrained multi-objective genetic algorithm, *Swarm and Evolutionary Computation*, 15, 66-79.

Mallipeddi, R. ve Suganthan, P. N., 2010, Ensemble of Constraint Handling Techniques, *Ieee Transactions on Evolutionary Computation*, 14 (4), 561-579.

Mani, A. ve Patvardhan, C., 2009, A Novel Hybrid Constraint Handling Technique for Evolutionary Optimization, *2009 Ieee Congress on Evolutionary Computation, Vols 1-5*, 2577-+.

Mezura-Montes, E. ve Cetina-Domínguez, O., 2009, Exploring promising regions of the search space with the scout bee in the artificial bee colony for constrained optimization, p.

Mezura-Montes, E. ve Coello, C. A. C., 2011, Constraint-handling in nature-inspired numerical optimization: Past, present and future, *Swarm and Evolutionary Computation*, 1 (4), 173-194.

Michalewicz, Z., 1995, A survey of constraint handling techniques in evolutionary computation methods, *Evolutionary Programming Iv*, 135-155.

Michalewicz, Z. ve Schoenauer, M., 1996, Evolutionary Algorithms for Constrained Parameter Optimization Problems, *Evolutionary Computation*, 4 (1), 1-32.

Nguyen, T. T. ve Yao, X., 2009, Benchmarking and Solving Dynamic Constrained Problems, *2009 Ieee Congress on Evolutionary Computation, Vols 1-5*, 690-697.

Rodrigues, M. d. C., Guimarães, S. ve de Lima, B. S. L. P., 2018, E-BRM: A constraint handling technique to solve optimization problems with evolutionary algorithms, *Applied Soft Computing*, 72, 14-29.

Runarsson, T. P. ve Yao, X., 2000, Stochastic ranking for constrained evolutionary optimization, *Ieee Transactions on Evolutionary Computation*, 4 (3), 284-294.

Sadollah, A., Bahreininejad, A., Eskandar, H. ve Hamdi, M., 2013, Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems, *Applied Soft Computing*, 13 (5), 2592-2612.

Schoenauer, M. ve Xanthakis, S., 1993, Constrained Ga Optimization, *Proceedings of the Fifth International Conference on Genetic Algorithms*, 573-580.

Shan, D., Cao, G. ve Dong, H., 2013, LGMS-FOA: An Improved Fruit Fly Optimization Algorithm for Solving Optimization Problems, *Mathematical Problems in Engineering*, 2013, 9.

Smith, A., Coit, D., Bäck, T., Fogel, D. ve Michalewicz, Z., 1998, Penalty Functions, p.

Sorkhabi, S. Y. D., Romero, D. A., Beck, J. C. ve Amon, C. H., 2018, Constrained multi-objective wind farm layout optimization: Novel constraint handling approach based on constraint programming, *Renewable Energy*, 126, 341-353.

Takahama, T. ve Sakai, S., 2010, Constrained Optimization by the e Constrained Differential Evolution with an Archive and Gradient-Based Mutation, *2010 Ieee Congress on Evolutionary Computation (Cec)*.

Takahama, T. ve Sakai, S., 2012, Efficient Constrained Optimization by the epsilon Constrained Rank-Based Differential Evolution, *2012 Ieee Congress on Evolutionary Computation (Cec)*.

Trivedi, A., Sanyal, K., Verma, P. ve Srinivasan, D., 2017, A Unified Differential Evolution Algorithm for Constrained Optimization Problems, *2017 Ieee Congress on Evolutionary Computation (Cec)*, 1231-1238.

Uymaz, S. A., Tezel, G. ve Yel, E., 2015a, Artificial algae algorithm (AAA) for nonlinear global optimization, *Applied Soft Computing*, 31, 153-171.

Uymaz, S. A., Tezel, G. ve Yel, E., 2015b, Artificial algae algorithm with multi-light source for numerical optimization and applications, *Biosystems*, 138, 25-38.

Wang, B., Li, H., Li, J. ve Wang, Y., 2018, Composite Differential Evolution for Constrained Evolutionary Optimization, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 1-14.

Wang, Y., Cai, Z. X. ve Zhang, Q. F., 2011, Differential Evolution with Composite Trial Vector Generation Strategies and Control Parameters, *Ieee Transactions on Evolutionary Computation*, 15 (1), 55-66.

Xiao, J., Huang, Y., Cheng, Z., He, J. ve Niu, Y., 2014, A hybrid membrane evolutionary algorithm for solving constrained optimization problems, *Optik*, 125 (2), 897-902.

Yang, X. S., 2010, A New Metaheuristic Bat-Inspired Algorithm, *Nicso 2010: Nature Inspired Cooperative Strategies for Optimization*, 284, 65-74.

Yeniay, Ö., 2005, Penalty Function Methods for Constrained Optimization with Genetic Algorithms, *Mathematical and Computational Applications*, 10 (1).

Zade, A., Patel, N. ve Padhiyar, N., 2017, Effective Constrained Handling by Hybridized Cuckoo Search Algorithm with Box Complex Method, *Ifac Papersonline*, 50 (2), 209-214.

Zain, M. Z. B., Kanesan, J., Chuah, J. H., Dhanapal, S. ve Kendall, G., 2018, A multi-objective particle swarm optimization algorithm based on dynamic boundary search for constrained optimization, *Applied Soft Computing*, 70, 680-700.

EKLER

EK-1

G01

$$f(x) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i$$
$$g_1(x) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leq 0$$
$$g_2(x) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leq 0$$
$$g_3(x) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leq 0$$
$$g_4(x) = -8x_1 + x_{10} \leq 0$$
$$g_5(x) = -8x_2 + x_{11} \leq 0$$
$$g_6(x) = -8x_3 + x_{12} \leq 0$$
$$g_7(x) = -2x_4 - x_5 + x_{10} \leq 0$$
$$g_8(x) = -2x_6 - x_7 + x_{11} \leq 0$$
$$g_9(x) = -2x_8 - x_9 + x_{12} \leq 0$$
$$0 \leq x_i \leq 1\ (i = 1, \ldots, 9)$$
$$0 \leq x_i \leq 100\ (i = 10, 11, 12)$$
$$0 \leq x_{13} \leq 1$$

G02

$$f(x) = \left| \frac{\sum_{i=1}^{n} cos(x_i)^4 - 2\prod_{i=1}^{n}(x_i)^2}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \right|$$
$$g_1(x) = 0.75 - \prod_{i=1}^{n} x_i \leq 0$$
$$g_2(x) = \sum_{i=1}^{n} x_i - 7.5n \leq 0$$
$$n = 20\ ve\ 0 \leq x_i \leq 10\ (i = 0, \ldots, n)$$

G03

$$f(x) = (\sqrt{n})^n \prod_{i=1}^{n} x_i$$
$$h_1(x) = \sum_{i=1}^{n} x_i^2 - 1 = 0$$
$$n = 10\ ve\ 0 \leq x_i \leq 1\ (1, \ldots, n)$$

G04

$$f(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$$
$$g_1(x) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \leq 0$$
$$g_2(x) = -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \leq 0$$
$$g_3(x) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \leq 0$$
$$g_4(x) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \leq 0$$
$$g_5(x) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \leq 0$$

$$g_6(x) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \le 0$$

$78 \le x_1 \le 102$
$33 \le x_2 \le 45$
$27 \le x_i \le 45 \ (i = 3,4,5)$

G05

$f(x) = 3x_1 + 0.000001{x_1}^3 + 2x_2 + ({}^{0.000002}/_3){x_2}^3$
$g_1(x) = -x_4 + x_3 - 0.55 \le 0$
$g_2(x) = -x_3 + x_4 - 0.55 \le 0$
$h_3(x) = 1000\sin(-x_3 - 0.25) + 1000\sin(-x_4 - 0.25) + 894.8 - x_1 = 0$
$h_4(x) = 1000\sin(x_3 - 0.25) + 1000\sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0$
$h_5(x) = 1000\sin(x_4 - 0.25) + 1000\sin(x_4 - x_3 - 0.25) + 1294.8 = 0$
$0 \le x_1 \le 1200$
$0 \le x_2 \le 1200$
$-0.55 \le x_3 \le 0.55$
$-0.55 \le x_4 \le 0.55$

G06

$f(x) = (x_1 - 10)^3 - (x_2 - 20)^3$
$g_1(x) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \le 0$
$g_2(x) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \le 0$
$13 \le x_1 \le 100$
$0 \le x_2 \le 100$

G07

$$f(x) = {x_1}^2 + {x_2}^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5{x_7}^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$$

$g_1(x) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \le 0$
$g_2(x) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \le 0$
$g_3(x) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \le 0$
$g_4(x) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \le 0$
$g_5(x) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \le 0$
$g_6(x) = x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_3 - 6x_6 \le 0$
$g_7(x) = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \le 0$
$g_8(x) = -3x_1 + 6x_2 + 12(x_9 - 8) - 7x_{10} \le 0$
$-10 \le x_i \le 10 \quad i=1,...10$

G08

$$F(X) = \frac{\sin^2(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1 + x_2)}$$

$g_1(x) = x_1^2 - x_2 + 1 \le 0$
$g_2(x) = 1 - x_1 + (x_2 - 4)^2 \le 0$
$0 \le x_1 \le 10$
$0 \le x_2 \le 10$

G09

$$F(x) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$$

$g_1(x) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_3 \leq 0$

$g_2(x) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_3 \leq 0$

$g_3(x) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0$

$g_4(x) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0$

$-10 \leq x_i \leq 10$ (i=1,..7)

G10

$f(x) = x_1 + x_2 + x_3$

$g_1(x) = -1 + 0.0025(x_4 + x_6) \leq 0$

$g_2(x) = -1 + 0.0025(x_5 + x_7 - x_4) \leq 0$

$g_3(x) = -1 + 0.01(x_8 - x_5) \leq 0$

$g_4(x) = -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \leq 0$

$g_5(x) == -x_2x_7 + 1250x_3 + x_2x_4 - 1250x_4 \leq 0$

$g_6(x) = -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \leq 0$

$100 \leq x_1 \leq 10000$

$1000 \leq x_i \leq 10000$ (i=2,3)

$10 \leq x_i \leq 1000$ (i=4,..8)

G11

$f(x) = x_1^2 + (x_2 - 1)^2$

$h(x) = x_2 - x_1^2 = 0$

$-1 \leq x_1 \leq 1$

$-1 \leq x_2 \leq 1$

G12

$F(x) = (100 - (x_1 - 5)^2 - (x_2 - 5)^2 - (x_3 - 5)^2)/100$

$g(x) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0.0625 \leq 0$

$0 \leq x_i \leq 10$ (i=1,2,3) p,q,r=1,2,....9

G13

$f(x) = \epsilon^{x_1x_2x_3x_4x_5}$

$h_1(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0$

$h_2(x) = x_2x_3 - 5x_4x_5 = 0$

$h_3(x) = x_1^3 + x_2^3 + 1 = 0$

$-2.3 \leq x_i \leq 2.3$ (i=1,2)

$-3.2 \leq x_i \leq 3.2$ (i=3,4,5)

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	**:**	Seda YILDIZ
Uyruğu	**:**	T.C.
Doğum Yeri ve Tarihi	**:**	Çan/1994
Telefon	**:**	542 206 9765
E-Posta	**:**	sedayildiz@selcuk.edu.tr

EĞİTİM

Derece		**Adı İlçe İl**	**Bitirme Yılı**
Lise	:	**Süleyman Demirel Fen Lisesi, Edirne**	**2012**
Üniversite	:	**Selçuk Üniversitesi , Konya**	**2016**
Yüksek Lisans	:	**Konya Teknik Üniversitesi, Konya**	**-**

İŞ DENEYİMLERİ

Yıl	**Kurum**	**Görevi**
2016	Selçuk Üniversitesi Bilgi İşlem Daire Başkanlığı	Bilgisayar Mühendisi

YABANCI DİLLER

İngilizce Yökdil(60)

YAYINLAR

YILDIZ SEDA, UYMAZ SAİT ALİ (2017).Performance of Artifical Algae Algorithm on Constrained Engineering Design Optimization Problems, II.INTERNATIONAL ACADEMICRESEARCHCONGRESS, 1024-1030.(Tam Metin Bildiri/Sözlü Sunum)(Yayın No:3982805)