



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



**BÜYÜK ÖLÇEKLİ OPTİMİZASYON
PROBLEMLERİNİN ÇÖZÜMÜNDE
YENİ YAKLAŞIMLAR**

Havva Gül KOÇER

DOKTORA TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Haziran-2023
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Havva Gül KOÇER tarafından hazırlanan “BÜYÜK ÖLÇEKLİ OPTİMİZASYON PROBLEMLERİNİN ÇÖZÜMÜNDE YENİ YAKLAŞIMLAR” adlı tez çalışması 09/06/2023 tarihinde aşağıdaki jüri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü BİLGİSAYAR MÜHENDİSLİĞİ Anabilim Dalı’nda DOKTORA TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Prof.Dr. Gülay TEZEL

.....

Danışman

Doç.Dr. Sait Ali UYMAZ

.....

Üye

Prof. Dr. İsmail BABAOĞLU

.....

Üye

Dr. Öğr. Üyesi Ayşe Merve ACILAR

.....

Üye

Dr. Öğr. Üyesi Onur İNAN

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Saadettin Erhan KESEN
Enstitü Müdürü

Bu tez çalışması Selçuk Üniversitesi Bilimsel Araştırma Projeleri tarafından 18101012 nolu proje ile desteklenmiştir.

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.

Havva Gül KOÇER
09/06/2023

ÖZET

DOKTORA TEZİ

BÜYÜK ÖLÇEKLİ OPTİMİZASYON PROBLEMLERİNİN ÇÖZÜMÜNDE YENİ YAKLAŞIMLAR

Havva Gül KOÇER

Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç.Dr. Sait Ali UYMAZ
2023, 94 Sayfa

Jüri

Prof.Dr. Gülay TEZEL
Prof. Dr. İsmail BABAÖĞLU
Doç.Dr. Sait Ali UYMAZ
Dr. Öğr. Üyesi Ayşe Merve ACILAR
Dr. Öğr. Üyesi Onur İNAN

Optimizasyon, gerçek dünya problemlerinde karar verme süreçlerini hızlandırmak veya karar verme kalitesini artırmak için kullanılmaktadır. Optimize edilmesi gereken çok fazla sayıda parametreye sahip problemler literatürde büyük ölçekli global optimizasyon (Large Scale Global Optimization, LSGO) problemleri olarak adlandırılmaktadır. Karar değişkenleri sayısının artmasıyla birlikte problemin arama uzayı ve dolayısıyla karmaşıklığı katlanarak artar. Bu nedenle ortaya çıkacak sorunların üstesinden gelmek için güçlü algoritma ve yöntemler tasarlanması veya var olan algoritmaların farklı mekanizmalar ile desteklenerek geliştirilmesi gerekmektedir. Bu tez kapsamında LSGO problemlerinin çözümü için problemi bir bütün olarak ele alan ayrıştırma esaslı olmayan çözüm yaklaşımları üzerinde çalışılmıştır. Standart evrimsel algoritmaların ek düzenleme ve tekniklerle geliştirilmesi kapsamında Yapay Alg Algoritmasının (Artificial Algae Algorithm, AAA) hem keşif hem de sömürü mekanizmasına katkıda bulunan helisel hareket aşamasında değişiklik yapılmış ve algoritmanın geliştirilen versiyonu Değiştirilmiş Yapay Alg Algoritması (Modified Artificial Algae Algorithm, MAAA) olarak adlandırılmıştır. CEC2010 test fonksiyon kümesi ile gerçekleştirilen testler yapılan değişikliğin algoritmanın LSGO çözüm performansını arttırdığını göstermiştir. LSGO zorluklarıyla baş etmede tercih edilen etkili çözüm yaklaşımlarından bir diğeri ise memetik algoritmalar. Bu tez kapsamında memetik algoritmaların performansında çok önemli bir etkiye sahip olan yerel arama aşamasında kullanılmak üzere dinamik adım boyutuna sahip altın oran rehberli yerel arama (Golden Ratio Guided Local Search with dynamic step size, GRGLS) adında yeni bir yerel arama yöntemi geliştirilmiştir. CEC2013 test fonksiyon kümesi ile gerçekleştirilen testler, önerilen algoritmanın karşılaştırma yapılan algoritmalar arasında en iyi çözümlere ulaştığını ve örtüşen fonksiyonlar ile ayrılamaz fonksiyonlarda diğerlerine göre üstün performans gösterdiğini kanıtlamıştır. Önerilen GRGLS yöntemini daha da geliştirmek amacıyla kaotik haritaların optimizasyon alanındaki kullanımı araştırılmıştır. Singer kaotik haritasından üretilen tekrarsız rastgele sayı, arama uzayında hareketi sağlayan adım büyüklüğü belirleme denklemine katsayı olarak kullanılmıştır. Geliştirilen yeni versiyona Kaotik Altın Oran Rehberli Yerel Arama (Caotic Golden Ratio Guided Local Search, CGRGLS) adı verilmiştir. CEC2015 Big-Opt test fonksiyon kümesi ile gerçekleştirilen üç ayrı performans değerlendirmesi yerel arama yönteminin yeni versiyonun önceki versiyona göre daha iyi sonuçlar elde ettiğini doğrulamıştır. Yapılan çalışmalardan elde edilen tüm sonuçlar, önerilen GRGLS ve CGRGLS yerel arama yöntemlerinin LSGO alanında kullanılabilir ve etkili ve verimli birer yerel arama yöntemi olduğunu göstermiştir.

Anahtar Kelimeler: AAA, altın oran, büyük ölçek, CEC, kaotik harita, LSGO, memetik algoritma, optimizasyon, yerel arama

ABSTRACT

PhD THESIS

NEW APPROACHES TO SOLVING LARGE-SCALE OPTIMIZATION PROBLEMS

Havva Gül KOÇER

**Konya Technical University
Institute of Graduate Studies
Department of Computer Engineering**

**Advisor: Assoc. Prof. Dr. Sait Ali UYMAZ
2023, 94 Pages**

Jury

**Prof.Dr. Gülay TEZEL
Prof. Dr. İsmail BABAOĞLU
Assoc. Prof. Dr. Sait Ali UYMAZ
Assist. Prof. Dr. Ayşe Merve ACILAR
Assist. Prof. Dr. Onur İNAN**

Optimization is used in real-world problems to speed up decision-making processes or to improve decision-making quality. Problems with many parameters that need to be optimized are called large-scale global optimization (LSGO) problems in the literature. As the number of decision variables increases, the search space and thus complexity of the problem increases exponentially. For this reason, it is necessary to design strong algorithms and methods to overcome the problems that may arise, or to develop existing algorithms by supporting them with different mechanisms. Within the scope of this thesis, non-decomposition based solution approaches that deal with the problem as a whole have been studied for the solution of LSGO problems. As part of the development of standard evolutionary algorithms with additional editing and techniques, changes were made in the helical motion stage of the Artificial Algae Algorithm (AAA), which contributes to both the exploration and exploitation mechanisms, and the developed version of the algorithm was named Modified Artificial Algae Algorithm (MAAA). The tests performed with the CEC2010 test function set showed that the change made increased the LSGO solution performance of the algorithm. Another effective solution approach preferred to cope with LSGO difficulties is memetic algorithms. Within the scope of this thesis, a new local search method called Golden Ratio Guided Local Search with dynamic step size (GRGLS) has been developed to be used in the local search phase, which has a very important effect on the performance of memetic algorithms. The tests performed with the CEC2013 test function set have proven that the proposed algorithm achieves the best solutions among the compared algorithms and outperforms the others in overlapping functions and non-separable functions. To further develop the proposed GRGLS method, the use of chaotic maps in the field of optimization has been investigated. The non-repeating random number generated from the Singer chaotic map is used as a coefficient in the step size determination equation that provides movement in the search space. The new version developed is called Chaotic Golden Ratio Guided Local Search (CGRGLS). Three separate performance evaluations with the CEC2015 Big-Opt test function set confirmed that the new version of the local search method achieved better results than the previous version. All the results obtained from the studies have shown that the proposed GRGLS and CGRGLS local search methods are effective and efficient local search methods that can be used in the LSGO area.

Keywords: AAA, CEC, chaotic map, golden ratio, optimization, large-scale, local search, LSGO, memetic algorithm

ÖNSÖZ

Öncelikle bu tez çalışmamda bana rehber olan ve her aşamada destek, emek ve katkılarıyla devam etmemi sağlayan danışman hocam Doç.Dr. Sait Ali UYMAZ 'a sonsuz sabrı ve anlayışından dolayı çok teşekkür ederim.

Fikirleri, güler yüzleri ve yapıcı eleştirileriyle bana yol göstererek tez çalışmamı iyileştirmemi sağlayan sayın tez izleme komitesi üyeleri Prof.Dr. Gülay TEZEL ve Dr.Öğr.Üyesi Ayşe Merve ACILAR 'a teşekkür ederim.

Öğrenim hayatımda bu noktaya gelmeme vesile olan, tüm hayatım boyunca her zaman ve her koşulda bana destek olan babam Mehmet Ali CEYLAN ve annem Ayşe CEYLAN sonsuz teşekkür, sevgi ve saygılarımı sunarım, emeğiniz için minnettarım.

Doktora öğrenimi için beni teşvik eden, bilgi, destek ve vizyonuyla hayatımın her alanında beni geliştiren değerli eşim merhum Doç.Dr. Barış KOÇER, çalışmalarım süresince gösterdikleri sabır, hoşgörü ve destekleriyle ayakta kalma sebebim, evlatlarım, Kenan Mert KOÇER ve Serra Duru KOÇER'e sonsuz sevgi ve teşekkürlerimi sunarım.

Bu araştırmaya 18101012 numaralı proje ile maddi destek sağlayan Bilimsel Araştırma Projeleri Koordinatörlüğü'ne ve bu vesileyle sürecin herhangi bir anına katkıda bulunan herkese camı gönülden teşekkürü bir borç bilirim.

Havva Gül KOÇER
KONYA-2023

İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
1. GİRİŞ	1
1.1. Tezin Önemi ve Amacı	2
1.2. Tezin Organizasyonu	5
2. KAYNAK ARAŞTIRMASI	6
2.1. Ayırıştırma Esaslı Yaklaşımlar	8
2.2. Ayırıştırma Esaslı Olmayan Yaklaşımlar	13
2.2.1. Standart optimizasyon algoritmaları	13
2.2.2. Memetik algoritmalar	16
3. MATERYAL VE YÖNTEM.....	22
3.1. Yapay alg algoritması	22
3.2. Doğrusal popülasyon boyutu azaltma ve yarı parametre uyarlamalı başarı geçmişine dayalı memetik diferansiyel gelişim çerçevesi (MLSHADE-SPA)	24
3.3. Altın Oran	26
3.4. Kaotik Haritalar	29
3.5. Test Fonksiyon Kümeleri.....	32
3.5.1. CEC2010.....	32
3.5.2. CEC2013.....	34
3.5.3. Big-Opt	35
4. ÖNERİLEN YENİ YEREL ARAMA YÖNTEMLERİ VE ALGORİTMALAR	38
4.1. Değiştirilmiş Yapay Alg Algoritması (MAAA)	38
4.2. Dinamik adım boyutuna sahip altın oran rehberli yerel arama (GRGLS) ve IMLSHADE-SPA	41
4.3. Kaotik altın oran rehberli yerel arama (CGRGLS) ve CIMLSHADE-SPA.....	47
5. DENEYSEL ÇALIŞMALAR VE TARTIŞMA	48
5.1. Yapay alg algoritmasının LSGO performansının değerlendirilmesi	48
5.2. Yapay alg algoritmasının LSGO performansının geliştirilmesi	50
5.3. GRGLS ve IMLSHADE-SPA	54
5.4. CGRGLS ve CIMLSHADE-SPA.....	68
5.4.1. Kaotik Harita Seçimi	69
5.4.2. Yerel Arama Metotlarının Karşılaştırılması	71
5.4.3. Diğer Algoritmalarla Karşılaştırma	74
5.4.4. LSGO yarışmalarını kazanan algoritmalarla karşılaştırma.....	77
6. SONUÇLAR VE ÖNERİLER	79

6.1 Sonular	80
6.2 neriler	83
KAYNAKLAR	86



KISALTMALAR

AAA	: Yapay Alg Algoritması (Artificial Algae Algorithm)
ABC	: Yapay arı kolonisi (Artificial Bee Colony algorithm)
ACO	: Karınca kolonisi optimizasyon (Ant Colony Optimization)
ANDE	: Yeni üçgensel mutasyon stratejili uyarlanabilir diferansiyel gelişim (Adaptive DE with Novel triangular mutation strategy)
CC	: İşbirliğine dayalı birlikte evrim (Cooperative Coevolution)
CCFR	: İşbirliğine dayalı çerçeve (Cooperative Coevolution Framework)
CCGA	: İşbirliğine dayalı genetik algoritma (Cooperative Coevolution Genetic Algorithm)
CCPSO	: İşbirliğine dayalı parçacık sürü optimizasyon (Cooperative Coevolution Particle Swarm Optimization)
CEC	: Evrimsel hesaplama kongresi (IEEE Congress of Evolutionary Computation)
CEC2008	: Evrimsel hesaplama kongresi 2008 (Congress of Evolutionary Computation 2008)
CECyıl	: Evrimsel hesaplama kongresi yıl (Congress of Evolutionary Computation year)
CGRGLS	: Kaotik altın oran rehberli yerel arama (Chaotic Golden Ratio Guided Local Search)
CSO	: Rekabetçi sürü optimize edicisi (Competitive Swarm Optimizer)
DE	: Diferansiyel gelişim (Differential Evolution)
DECC	: İşbirliğine dayalı Diferansiyel gelişim algoritması (Differential Evolution Cooperative Coevolution)
DECC-G	: Grup tabanlı problem ayrıştırma stratejisine sahip CC tabanlı diferansiyel gelişim çerçeve algoritması (Differential Evolution Cooperative Coevolution with Group)
EA	: Evrimsel algoritma (Evolution Algorithm)
EADE	: Geliştirilmiş uyarlanabilir diferansiyel gelişim algoritması (Enhanced Adaptive Differential Evolution)
EEG	: ElektroEnsefaloGrafı
EEGWO	: Keşif özelliği geliştirilmiş gri kurt optimize edicisi (Exploration-Enhanced Gray Wolf Optimizer)
FES	: Uygunluk değerlendirme sayısı (Fitness Evaluation)
GA	: Genetik algoritma (Genetic Algorithm)
GR	: Altın Oran (Golden Ratio)
GRGLS	: Altın oran rehberli yerel arama (Golden Ratio Guided Local Search)
GSS	: Altın bölüm araması (Golden Section Search)
GWO	: Gri kurt optimize edicisi (Gray Wolf Optimizer)
IGR	: Ters altın oran (Inverse Golden Ratio)
IHDELS	: Diferansiyel gelişim algoritmasının yerel arama ile yinelemeli hibridizasyonu (Iterative hybridization of DE with local search)
LSGO	: Büyük ölçekli global optimizasyon (Large Scale Global Optimization)
MA	: Memetik algoritma (Memetic Algorithm)
MAAA	: Değiştirilmiş yapay alg algoritması (Modified Artificial Algae Algorithm)
MA-SW-Chains	: Yerel arama zincirlerine dayalı memetik algoritma (Memetic Algorithm-Solis and Wets - Chains)
MCB	: Piyasa rekabeti davranış (Market Competition Behavior)

MCSO	: Değiştirilmiş rekabetçi sürü optimize edicisi (Modified Competitive Swarm Optimizer)
MLCC	: Çok düzeyli iş birliğine dayalı birlikte evrim (Multilevel Cooperative Coevolution)
MLSHADE-SPA	: Doğrusal popülasyon boyutu azaltma ve yarı parametre uyarlamalı başarı geçmişine dayalı memetik diferansiyel gelişim (Memetic Success History-based Differential Evolution with linear population size reduction and Semi-Parameter Adaptation)
MOS	: Çoklu yavru örnekleme algoritması (Multiple Offspring Sampling)
MMTS	: Değiştirilmiş çoklu yörünge araması (Modified Multiple Trajectory Search)
MTS	: Çoklu yörünge araması (Multiple Trajectory Search)
MTSLS1	: Çoklu yörünge araması yerel arama 1 (Multiple Trajectory Search Local Search 1)
PSO	: Parçacık sürü optimizasyon (Particle Swarm Optimization)
RDG	: Özyinelemeli diferansiyel gruplama (Recursive Differential Grouping)
RLLPSO	: Pekiştirmeli öğrenme düzeyi tabanlı parçacık sürü optimizasyonu (Reinforcement Learning Level-based Particle Swarm Optimization)
SabDE	: Kendi kendini uyarlayabilen ikili DE algoritması (Self-Adaptive Binary Differential Evolution)
SDENS	: Komşu arama ile geliştirilmiş ardışık DE (Sequential DE enhanced by neighborhood search)
SHADE-ILS	: Yinelemeli yerel aramaya sahip başarı geçmişine dayalı uyarlanabilir diferansiyel gelişim algoritması (Success History based Adaptive DE with Iterative Local Search)
SOCO2011	: Soft Computing Journal'ın 2011 özel sayısı (2011 Special Issue of the Soft Computing Journal)
SSW	: Sıralı Hareketli Pencere (Sequential Sliding Window)

1. GİRİŞ

Optimizasyon, belirli koşullar altında olası alternatifler arasından en iyi çözümü seçme yöntemidir. Genel olarak bir sistemin planlanmasındaki ana hedef ya istenen kârı maksimize etmek ya da gerekli çabayı(maliyeti) minimize etmektir. Bu durum da başlı başına bir tasarım ve optimizasyon problemidir. Bu problemin bilgisayar tarafından çözülebilmesi için önce matematiksel olarak modellenmesi gerekmektedir. Bu amaçla atılan ilk adım, karar değişkenleri olarak isimlendirilen parametre setini tanımlamak olmalıdır. Tanımlanan karar değişkenlerinin sayısı problemin boyutunu verir. İstenen kar veya gerekli çabanın karar değişkenlerinin bir fonksiyonu olarak ifade edilmesiyle problemin matematiksel modeli oluşturulur. Dolayısıyla optimizasyon sürecinin amacı, ulaşılmak istenen hedefe göre bu fonksiyonun minimum veya maksimum değerini oluşturan şartların bulunmasıdır. Başka bir ifadeyle optimizasyon, kısıtların her birine uyarak amaç fonksiyonunun hedeflenen en uygun değerini elde eden karar değişkenlerinin değerlerinin tespit edilmesidir.

Gelişen teknoloji ile bilgisayarların bilgi işleme gücünde kaydedilen gözle görülür gelişme, problemlerin boyutunda ve karmaşıklığında büyük bir artış meydana gelmesine neden olmuştur. Öyle ki, iş dünyası, bilim ve mühendislik alanında büyük boyutlu ve çeşitli gerçek dünya problemleri ortaya çıkmıştır (Singh ve Dulal, 2017). Büyük ölçekli elektronik sistem tasarımı, çok sayıda kaynağın bulunduğu planlama problemleri, büyük ölçekli trafik ağlarında araç yönlendirme, biyoinformatikte gen tanıma gibi birçok problem büyük ölçekli gerçek dünya problemi olarak tanımlanmaktadır (Mahdavi ve ark., 2015). Bu problemler, karar değişkenlerinin sayısına bağlı olarak farklı şekillerde ölçeklendirilmiştir. Büyük ölçek kavramının ilk çıktığı zamanlar da 100 ve üzeri karar değişkenine sahip problemler bu kategoride sayılırken günümüzde düzenlenen yarışma ve oturumlarda kullanılan güncel test fonksiyon kümesi (Li ve ark., 2013) baz alındığında 1000 ve üzeri karar değişkenine sahip problemler büyük ölçekli optimizasyon problemi olarak değerlendirilebilir. Literatürde bu alanı temsilen, “Büyük Ölçekli Global Optimizasyon” (Large Scale Global Optimization, LSGO) terimi kullanılmaktadır.

Büyük ölçekli optimizasyon problemlerinin değişken sayısı ve veri türlerine bağlı olarak hem modelleme hem de çözüm süreci oldukça karmaşık hale gelebilmektedir. Bu tür problemlerde en uygun çözümün elde edilmesinde kullanılan en yaygın yöntem metasezgisel algoritmalarıdır. Ancak büyük ölçekli optimizasyon problemlerinde karar değişkenlerinin sayısının diğer bir ifadeyle problem boyutunun artması, metasezgisel

algoritmaların tasarımıdaki etkinliklerini kaybetmelerine ve performanslarının düşmesine neden olmaktadır. Çünkü bu algoritmalar, yerel çözümlere takılma, kararsız performans sergileme, bellek verimliliğinden yoksun olma gibi kısıtlamalara sahip olabilmektedirler (Qiu ve Liu, 2016). Bu nedenle bu tip problemlerin çözümü için daha etkili, kararlı, ölçeklenebilir algoritmalara ve büyük boyutun neden olduğu sorunlarla başa çıkabilecek farklı yöntemlere ihtiyaç duyulmaktadır.

1.1. Tezin Önemi ve Amacı

Değişken sayısına ve veri tiplerine bağlı olarak optimizasyon problemlerinin zorluk dereceleri artabilmektedir. Problemin yapısına bağlı olarak hem modelleme hem de çözüm sürecinde ortaya çıkabilecek zorlukların üstesinden gelmek amacıyla sezgisel yöntemler geliştirilmiştir. Sezgisel yöntemler, küçük veya orta ölçekli problemlere uygulandığında etkili arama yetenekleri gösteriyor olsa da büyük ölçekli problemlere uygulandığında performansı düşmekte ve etkinliğini kaybetmektedir. Bu performans bozulmasının temel nedenleri (Li ve ark., 2013; Mahdavi ve ark., 2015) :

- Boyut artışına bağlı olarak arama uzayının katlanarak büyümesi sonucu arama uzayını etkin ve verimli bir şekilde keşfetme sorunu,
- Fonksiyonların karmaşıklığının ve karakteristiklerinin boyut artışı ile değişebilmesi,
- Birbirine bağlı değişkenlerin varlığı ve sayısındaki artış,
- Çok büyük hesaplama maliyetine ihtiyaç duyulmasıdır.

Büyük ölçekli problemlerin optimizasyonu hesaplama gücü, zaman ve kaynak gereksinimleri açısından zorlu bir görev olarak kabul edilir. Çünkü boyut artışına bağlı olarak artan arama alanı ve karmaşıklık, problemin çözümü aranırken olası her çözümün veya kombinasyonun aranmasını imkânsız kılmaktadır. Bu nedenle makul sürede kabul edilebilir kalitede çözümler bulabilen sezgisel yöntemlerin kullanılması tercih edilmektedir. Klasik sezgisel algoritmalar, boyut büyümesi nedeniyle ortaya çıkan problemlerle uğraşmakta zorluk çekerler. Bu durum için boyutsallığın laneti (curse of dimensionality) terimi kullanılmaktadır (Bellman, 1956; Chen ve ark., 2015). Boyutsallığın laneti, girdi değişkenlerinin sayısındaki artışa göre arama uzayının boyutunun katlanarak artmasından kaynaklanan optimum çözüm bulmanın zorluklarını

ifade etmektedir. Bu zorlukların üstesinden gelmek için büyük arama uzayında düşük hesaplama maliyetleri ile yüksek kaliteli çözümler elde etmek amacıyla daha etkili ve verimli farklı çözüm ve tekniklere ihtiyaç vardır. Daha iyi arama yapabilen algoritmalar geliştirilmesi, daha büyük ve karmaşık problemleri çözebilme veya aynı boyuttaki probleme daha iyi çözümler bulabilme gibi iki önemli sonucu beraberinde getirecektir (Omidvar ve ark., 2021b).

Evrimsel algoritmaların yerel arama yöntemleri ile birleştirilmesiyle tasarlanan algoritmalar MA olarak adlandırılmaktadır (Moscato, 1989). MA'lar, iyi çözümlerin keşfini hızlandırmak ve evrimle erişilemeyecek çözümlere ulaşmak amacıyla bir popülasyondaki bireylerin doğal adaptasyon ve yaşam boyu öğrenme kombinasyonundan ilham almaktadır. MA, büyük ölçekli optimizasyon problemlerinin çözümünde kullanılan yöntem ve algoritmalar arasında oldukça etkili ve çokça tercih edilen bir yöntemdir. Evrimsel Hesaplama Kongresi (IEEE Congress of Evolutionary Computation, CEC) bünyesinde düzenlenen büyük ölçekli optimizasyon yarışmalarına bakıldığında dereceye giren başarılı algoritmaların büyük oranda MA olması, bu yöntemin LSGO problemlerinin çözümünde oldukça etkili ve popüler bir yaklaşım olduğu görüşünü desteklemektedir (Cabrera, 2016; Molina ve ark., 2018a; Omidvar ve ark., 2021b; 2021a).

MA performansı büyük ölçüde bünyesinde kullandığı yerel aramaların başarısına bağlıdır (Smith, 2007; Molina ve ark., 2010a). Büyük ölçekli problemlerde artan sayıda karar değişkeni nedeniyle çözüm uzayı üstel olarak katlanarak büyüdüğünden bu geniş çözüm uzayını daha iyi keşfetmeyi sağlayacak etkili ve verimli arama stratejilerine ihtiyaç duyulmaktadır. Bu tez çalışmasında büyük ölçekli optimizasyon problemlerinin çözümüne katkıda bulunmak amacıyla MA'ları güçlendirecek, daha etkili ve kararlı hale getirecek iki yeni yerel arama yöntemi önerilmiştir.

Büyük ölçekli problemlerin çözümünde ortaya çıkabilecek sorunların üstesinden gelmek amacıyla yapılan çalışmalar incelendiğinde birbirinden farklı sınıflandırmalar yapıldığı görülmüştür. Genel kanı yapılan çalışmaları, problemi parçalar halinde ele alan İşbirliğine Dayalı Birlikte Evrim (Cooperative Coevolution, CC) algoritmaları ve problemi bir bütün olarak ele alan ayrıştırma esaslı olmayan algoritmalar olmak üzere iki temel yaklaşım kategorisine ayırmaktır (Mahdavi ve ark., 2015; Molina ve ark., 2018a; Jian ve ark., 2020; Mohapatra ve ark., 2022).

Bu tez kapsamında ayrıştırma esaslı olmayan algoritmalar bünyesinde aşağıdaki çalışmalar yapılmıştır :

- 1) Standart evrimsel algoritmaların ek düzenleme ve tekniklerle geliştirilmesi kapsamında Yapay Alg Algoritmasının (Artificial Algae Algorithm, AAA) LSGO performansı test edilmiştir. Boyut büyüdükçe düşen performansı artırmak amacıyla hem keşif hem de sömürü mekanizmasına katkıda bulunan helisel hareket aşamasında değişiklik yapılmış ve algoritmanın geliştirilen versiyonu Değiştirilmiş Yapay Alg Algoritması (Modified Artificial Algae Algorithm, MAAA) olarak adlandırılmıştır.
- 2) LSGO zorluklarıyla baş etmede tercih edilen etkili çözüm yaklaşımlarından memetik algoritmalar kapsamında algoritma performansında çok önemli bir etkiye sahip yerel arama aşamasında kullanılmak üzere dinamik adım boyutuna sahip altın oran rehberli yerel arama (Golden Ratio Guided Local Search with dynamic step size, GRGLS) adında yeni bir yerel arama yöntemi geliştirilmiştir.
- 3) Önerilen GRGLS yöntemi performansını daha da artırmak amacıyla kaotik haritaların tekrarsız rastgele sayı üretme özelliğinin kullanımı araştırılmıştır. Singer kaotik haritasından üretilen sayının arama uzayında hareketi sağlayan adım büyüklüğü belirleme denkleminde katsayı olarak kullanılmasıyla yeni bir yerel arama yöntemi önerilmiştir. Bu yeni versiyona Kaotik Altın Oran Rehberli Yerel Arama (Caotic Golden Ratio Guided Local Search, CGRGLS) adı verilmiştir.

Bu tez kapsamında geliştirilen GRGLS ve CGRGLS yerel arama yöntemleri ile LSGO alanında kullanılabilecek etkili ve verimli birer yerel arama yöntemi sunulmuş ve büyük ölçekli optimizasyon problemlerinin çözümüne katkıda bulunulmuştur.

1.2. Tezin Organizasyonu

Bu tez çalışması altı bölümden oluşmaktadır.

İlk bölümde tezin önemi ve amacı açıklanmış, organizasyonu hakkında bilgi verilmiştir.

İkinci bölümde, literatürdeki ilgili çalışmalar hakkında bilgi verilmiştir.

Üçüncü bölümde, tez kapsamında üzerinde çalışılan algoritmalar, yöntemi tasarlarken kullanılan yöntemler ve deneysel çalışmalar için kullanılan test fonksiyon kümeleri hakkında bilgilere yer verilmiştir.

Dördüncü bölümde, bu tez kapsamında önerilen yerel arama yöntemleri ve algoritmalar tanıtılmıştır.

Beşinci bölümde, tez kapsamında gerçekleştirilen dört çalışmaya dair bilgiler açıklanmış, karşılaştırma sonuçları, istatistik testler vb. ile yapılanlar anlatılarak her çalışmanın sonuçları yorumlanmıştır.

Altıncı bölümde ise genel sonuçlar değerlendirilip, gelecek çalışmalara dair önerilerde bulunulmuştur.

2. KAYNAK ARAŞTIRMASI

LSGO optimizasyon türü, boyut büyüklüğü sonucu oluşan oldukça büyük bir arama alanında en iyi çözümü bulmayı amaçlayan bir optimizasyon sürecidir. Literatürde bu alandaki çalışmalar oldukça çeşitlilik göstermekte ve geniş bir alana yayılmaktadır. Büyük ölçekli optimizasyon problemleri karmaşık matematiksel modellerle tanımlandığından ve genellikle çok sayıda kısıta sahip olduğundan zorlu bir görev olarak kabul edilmekte ve bu durum araştırmacıların LSGO alanına ilgisini arttırmaktadır.

Büyük ölçekli optimizasyon problemlerine artan ilginin bir sonucu olarak son 15 yılda optimizasyonun farklı dallarına ait özel sayılar yayınlanmış ve CEC gibi alanında öncü bir kongrede konuyla ilgili özel oturumlar ve bu oturumlar bünyesinde de yarışmalar (Tang ve ark., 2009; Li ve ark., 2013; Omidvar ve ark., 2021a) düzenlenmiştir. Bu yarışmalar sayesinde bu alanda farklı yıllardaki geliştirme eğilimleri ve hangi algoritmaların en iyi sonuçlar aldığı görülebilmektedir. Dolayısıyla bu özel oturum ve yarışmalar araştırma alanının kapsam ve gelişimine dair temel bilgiler vermekte ve gelecekte bu alanda neler yapılabileceği noktasında çok faydalı bir geri bildirim sağlamaktadır.

Mahdavi ve ark., büyük ölçekli optimizasyon alanında önerilen metasezgisel algoritmaları ve uygulamalarını incelemiş ve LSGO problemlerini çözmeye yönelik araştırma çalışmalarına genel bir bakış sağlamıştır. Literatürde bu alanda gerçekleştirilen çalışmaları, ayrıştırma esaslı yöntemler ve ayrıştırma esaslı olmayan yöntemler olarak iki grup altında değerlendirmişler ve bu yöntemlerdeki gelişmelere, test fonksiyon kümeleri ve performans ölçütlerine değinerek LSGO alanında tespit edilen bazı boşlukları tanımlamışlardır. Ayrıca yaptıkları araştırmalar sonucu elde edilen veriler ışığında gelecekte bu alanda yapılabilecekler dair fikirlerini listelemişlerdir (Mahdavi ve ark., 2015).

Büyük ölçekli optimizasyon problemleri çözüm yaklaşımları incelendiğinde, birçok alternatif algoritma, hibrit metot, özel teknik geliştirildiği görülmekle birlikte aslında iki yöntemin belirgin bir şekilde öne çıktığı fark edilmektedir. Cabrera, öne çıkan bu iki yöntemi(CC ve MA) kısaca tanıtmış, alana özel düzenlenen farklı yarışmaların kazananlarını inceleyerek en iyi sonuçları elde eden algoritmaların yapısına değinmiştir. Ayrıca, araştırma alanındaki bazı ilginç eğilimlere ve zorluklara işaret ederek bu kapsamda gelecekte yapılabilecekler dair önerilerde bulunmuştur (Cabrera, 2016).

Singh ve Dulal, LSGO alanında kullanılan güncel bazı metasezgisel algoritmalara özet bir giriş yapmış, algoritmaların CEC yarışmalarında gösterdikleri performansları değerlendirerek bu alanda yapılabilecek gelecek çalışmalara dair önerilerde bulunmuşlardır. Bu araştırma bünyesinde LSGO problemleriyle başa çıkmak için kullanılan çözüm önerileri standart evrimsel algoritmalar, CC tabanlı algoritmalar ve MA olmak üzere üç kategoride değerlendirilmiştir (Singh ve Dulal, 2017).

Molina ve ark., optimizasyonun farklı dallarında düzenlenen yarışmaların katılımcılarını, kıyaslamalarını, kazananlarını inceleyerek farklı bakış açıları ve yaklaşımları tanıtmışlardır (Molina ve ark., 2018a).

Jian ve ark., LSGO alanında geliştirilen evrimsel optimizasyon algoritmalarını iki kategoride incelemişler, farklılıklarını ve özelliklerini analiz etmek amacıyla yedi iyi bilinen algoritmanın işleyişine dair kısa özetler vererek CEC2010 ve CEC2013 büyük ölçekli optimizasyon test fonksiyon kümeleri üzerindeki performanslarını karşılaştırmışlardır. Ayrıca hangi algoritmaların farklı tipteki test fonksiyonları üzerinde daha iyi performans gösterdiğini inceleyerek gelecekteki bazı potansiyel araştırma alanlarına değinmişlerdir (Jian ve ark., 2020).

Omidvar ve ark., araştırmacılara ve uygulayıcılara LSGO alanını kuşbakışı gösterip ana eğilimler ve mevcut durum hakkında bilgi edinmelerine yardımcı olmak amacıyla son çalışmaları oldukça kapsamlı bir araştırma ve analizi yaparak incelemişlerdir. Araştırma, her biri iki yüzden fazla kaynağı referans veren iki ayrı makale şeklinde yayınlanmıştır. Birinci makalede, büyük ölçekli küresel optimizasyona yönelik iki ana algoritmik yaklaşım olan problem ayrıştırma yöntemlerini ve MA vb. hibritleştirme metotlarını incelemişlerdir (Omidvar ve ark., 2021b). İkinci makalede ise büyük ölçekli küresel optimizasyona yönelik bir dizi başka algoritmik yaklaşım incelenerek çok çeşitli problem alanlarını tanımlamışlar, alandaki tuzaklar ve zorlukları kapsayacak şekilde gelecekteki araştırmalar için çeşitli potansiyel alanlara değinmişlerdir (Omidvar ve ark., 2021a).

LSGO problemleri olarak adlandırılan yüksek karmaşıklık ve boyuta sahip optimizasyon problemlerinin üstesinden gelmek için modifikasyonlarıyla birlikte çok sayıda metasezgisel algoritma önerilmiştir. Ancak, gerçek hayattaki büyük ölçekli optimizasyon problemlerini çözmek için belirli bir algoritma seçmek pratik olarak zordur. Bu nedenle, Mohapatra ve ark. yaptıkları çalışmada probleme uygun büyük ölçekli optimizasyon algoritmasını bulma sürecini aydınlatmak amacıyla popüler büyük ölçekli problemleri ve bu problemleri çözmek için önerilen algoritmaları ayrıntılı olarak analiz

etmişlerdir. Önerilen algoritmaların CEC2013 LSGO test fonksiyon kümesi üzerinde yeterliliğini doğrulamak için deneysel çalışmalar ve istatistiksel incelemeler yapmışlardır. Nihayetinde çoklu yavru örnekleme (Multiple Offspring Sampling, MOS) algoritmasının 2013 versiyonunun üstün sonuçlar ürettiği ve çözüm mükemmelliği açısından başarılı birçok algoritmayı geride bıraktığı sonucuna varmışlardır (Mohapatra ve ark., 2022).

Diğer taraftan LSGO alanında önerilen algoritmaların çoğu genellikle yalnızca belirli bir fonksiyon setine göre değerlendirilerek performansı gösterilmeye çalışılmaktadır. Dolayısıyla algoritmaların farklı özellikler sunan diğer kıyaslamalardaki performansı hakkında bilgi edinmek oldukça zahmetli bir süreç haline gelebilmektedir. LaTorre ve ark., büyük ölçekli yarışmalarda en iyi performansa sahip algoritmaları da içeren 10 adet algoritmanın iyi bilinen üç test fonksiyon kümesine (CEC2010, CEC2013, SOCO2011) ait tüm fonksiyonlardaki performanslarını incelemişler, sonuçların karşılaştırmalı bir istatistiksel çalışması ve eşdeğer performansa sahip olanlar için algoritma profillerinin tanımlanması da dahil olmak üzere geniş bir değerlendirme yapmışlardır (LaTorre ve ark., 2015).

Büyük ölçekli optimizasyon problemleri için önerilen çözüm yaklaşımları incelendiğinde geliştirilen yöntem ve teknikler için farklı sınıflandırmalar yapıldığı görülse de çalışmaları ayrıştırma esaslı (decomposition) ve ayrıştırma esaslı olmayan (non-decomposition) şeklinde iki temel yaklaşım kategorisine ayırmak en çok rastlanılan sınıflandırmadır (Mahdavi ve ark., 2015; Jian ve ark., 2020; Mohapatra ve ark., 2022).

Bu çerçevede anlaşılabilirliği artırmak amacıyla kaynak araştırması, ayrıştırma esaslı yaklaşımlar, ayrıştırma esaslı olmayan yaklaşımlar şeklinde iki ana başlıkta sunulmuştur. Tezin amacına hizmet eden MA yaklaşımıyla ilgili yayınlar, ayrıştırma esaslı olmayan yaklaşımlar başlığı altında incelenmiştir.

2.1. Ayrıştırma Esaslı Yaklaşımlar

CC yöntemi, büyük ölçekli optimizasyon probleminin tamamının çözülmesi daha kolay olan küçük alt problemlere ayrıştırarak çözen bir parçala fethet (divide and conquer) tekniğidir. Bu yöntemin amacı, optimizasyon problemlerini daha küçük ve daha yönetilebilir alt problemlere ayrıştırarak optimizasyon sürecini hızlandırmak ve farklı popülasyonların veya alt problemlerin birbirleriyle iş birliği yaparak global optimumu bulmasını sağlamaktır. Burada her alt problem orijinal problemin karar değişkenlerinin

bir alt bileşenini içerir. Her bir alt problem veya popülasyon, kendi yerel arama stratejilerini kullanarak kendi bireylerini optimize etmektedir. Bireylerin performansının değerlendirilmesi sonucunda seçilen en iyi birey diğer alt problemlerle bir diğer deyişle popülasyonlarla paylaşılarak global optimum elde edilmeye çalışılmaktadır. Alt bileşenlerin birbirinden bağımsız olduğu ideal durumda, her bir alt problemin bulunduğu en iyi çözümünün kombinasyonu, orijinal problemin en iyi genel çözümü olmaktadır.

Bu yöntem, LSGO alanında da etkili çözüm yöntemlerinden biri olarak yaygın kullanıma sahiptir (Yang ve ark., 2008b; 2008c; Yang ve ark., 2009). Yüksek boyutlu problemleri ayrıştırarak çözüme fikri optimizasyon alanında ilk defa 1994 yılında Potter ve De Jong tarafından önerilmiş ve ilgili yayında standart Genetik Algoritmanın (Genetic Algorithm, GA) performansını iyileştirmek amacıyla CC metodu kullanılarak n boyutlu problem, n adet 1 boyutlu probleme ayrıştırılmıştır (Potter ve De Jong, 1994). Van den Bergh ve Engelbrecht ise, CC yöntemini Parçacık Sürü Optimizasyonuna (Particle Swarm Optimization Algorithm, PSO) uygulayan ve CCPSO'yu geliştiren ilk kişilerdir. Tüm boyutları ayrı ayrı ayıran CCGA'nın aksine, CCPSO algoritması n boyutlu bir problemi, k adet s boyutlu problemlere ayrıştırarak çözmektedir (Van den Bergh ve Engelbrecht, 2004).

Önerilen ilk CC yaklaşımı tamamen ayrılabilir problemlere uygun bir yapı önermekte ve değişkenler arasındaki karşılıklı ilişkiyi, bağlantıyı göz ardı etmektedir. Bu durum da tekniğin ayrılabilir problemlerde aynı performansı gösterememesine neden olmaktadır. Bu nedenle CC performansını etkileyen iki kritik etken, ayrıştırma yöntemi ve alt bileşenlerin optimizasyonunda kullanılacak algoritmayı seçmektir. Bu etkenlerden ayrıştırma yöntemi, değişkenleri alt bileşenlerde gruplama işleminde uygulanan yöntemdir. Burada amaç birbiriyle bağımlı diğer bir deyişle etkileşimli değişkenlerin aynı alt bileşende bulunmasını sağlamaktır. Pratikte problem hakkında önceden bilgi sahibi olmadan problemin nasıl ayrıştırılması gerektiği net olmadığından değişkenleri ayrıştırmak için en uygun yöntemi belirlemek kritik bir seçim işlemidir. Shi ve ark., Diferansiyel Gelişim Algoritmasında (Differential Evolution Algorithm, DE) CC yaklaşımını uygulayarak işbirliğine dayalı diferansiyel gelişim algoritmasını (Differential Evolution Cooperative Coevolution, DECC) tasarlarlarken, karar değişkenlerini eşit büyüklükte iki alt bileşene ayıran yarıya bölme stratejisi adı verilen bir ayrıştırma stratejisi de önermişlerdir (Shi ve ark., 2005). Değişkenleri ayrıştırmanın en basit yöntemi ise değişkenleri rastgele birkaç örtüşmeyen gruba bölen rastgele gruplama tekniğidir (Yang ve ark., 2008b). Bu teknik değişken etkileşimi ile baş etmenin en temel yoludur.

Çünkü bu teknik, etkileşimli değişkenleri aynı bileşene yerleştirme olasılığını artırmak için her birlikte evrim döngüsünden sonra karar değişkenlerini rastgele düzenlemektedir. Bundan dolayı etkileşim halindeki değişkenlerin bir alt bileşende gruplanma olasılığı, etkileşim halindeki değişkenlerin sayısı arttıkça hızla azalmaktadır. Omidvar ve ark., delta gruplama adını verdikleri tekniği uygulamak için tüm popülasyondaki belirli bir değişkendeki ortalama farkı ölçen delta değerlerini hesaplamışlar ve ayrılamayan bir problemin etkileşimli değişkenlerini sıralanmış delta değerlerine dayalı olarak birkaç alt bileşene gruplandırmışlardır (Omidvar ve ark., 2010). Bir değişkenin değişiminin fonksiyon değeri üzerindeki etkisi başka bir değişkenin değişiminin fonksiyon değeri üzerindeki etkisinden bağımsız ise, bu iki değişken ayrılabilir (Tezuka ve ark., 2004). Ayırıştırma yöntemlerinden bir diğeri de bu mantık temelinde küçük değişikliklerle çözümleri karşılaştırarak değişkenler arasındaki bağımlılıkları inceleyen ve elde ettiği bu bilgiyi değişkenleri aralarındaki bağımlılıklara göre gruplandırmak için kullanan diferansiyel gruplama tekniğidir (Omidvar ve ark., 2013). Bir diğer teknik ise seçilen bir karar değişkeni ile geri kalan değişkenler arasındaki etkileşimin özyinelemeli olarak incelendiği ve etkileşen tüm karar değişkenlerinin aynı alt probleme yerleştirildiği özyinelemeli diferansiyel gruplama tekniğidir (Sun ve ark., 2017a).

Yang ve ark., CC yapısındaki alt bileşen boyutunun uyarlanabilir olarak ayarlandığı yeni bir teknik içeren algoritma önermişler ve bu algoritmaya çoklu seviye iş birliğine dayalı birlikte evrim algoritması (Multilevel Cooperative Coevolution, MLCC) adını vermişlerdir. Bu algoritmada farklı grup boyutlarına sahip rastgele gruplandırma stratejisine dayalı olarak bir dizi problem ayırıştırıcı oluşturulmuştur. Evrim süreci birkaç döngüye bölünmüş olup her döngünün başında MLCC, tarihsel performansına göre bir ayırıştırıcının seçilmesi için kendi kendini uyarlayan bir mekanizma kullanmıştır. Farklı grup boyutları, orijinal amaç değişkenleri arasındaki farklı etkileşim düzeylerini yakaladığından dolayı algoritma farklı düzeyler arasında kendi kendine uyum sağlayabilmektedir. İlgili yayında bu algoritmanın performansı CEC2008 özel oturumu tarafından sağlanan bir dizi test fonksiyonu üzerinde çalıştırılarak test edilmiş ve elde edilen sonuçlar değerlendirilmiştir (Yang ve ark., 2008c).

CC'nin performansını artırmak için önce bağımlı değişkenler tanımlanarak ortak alt problemlerde gruplandırılmalı başka bir deyişle ayırıştırılmış alt problemler arasındaki birbirine bağımlı değişken sayısı azaltılmalıdır. Sayed ve ark. tarafından yapılan çalışmada, büyük ölçekli kısıtlı problemlerde bağımlı değişkenleri tanımlamak için yeni bir değişken etkileşimi tanımlama tekniği önerilmiş ve hem gerçek dünya probleminde

hem de farklı ayrılabilirlik ve karmaşıklık seviyelerine sahip örtüşen (overlapping) alt problemleri içeren yeni bir kısıtlı problem setinde test edilmiştir. Deneysel sonuçlar önerilen bu tekniğin diğer güncel gruplama teknikleriyle karşılaştırıldığında yüksek boyutta ayrıştırma yaklaşımına daha yüksek uygulanabilirlik ve daha az hesaplama maliyetiyle katkıda bulunduğunu göstermiştir (Sayed ve ark., 2015).

CC yaklaşımında değişkenler, büyük ölçekli problemin alt bileşenlerini oluşturmak amacıyla tümüyle bölünemediğinde CC tabanlı algoritmaların alt gruplara ayırma tekniği işlevselliğini kaybetmektedir. Bu nedenle klasik CC temelli algoritmalar, çok sayıda birbirine bağımlı etkileşen değişken içeren ayrılabilir (non-separable) problemlerin işlenmesinde verimsiz hale gelebilmektedir. Cao ve ark., büyük ölçekli optimizasyon problemlerinin çözümü için küresel ve yerel arama algoritmaları birlikte kullanan yeni bir CC çerçeve (framework) algoritması önermişlerdir. Sıralı Hareketli Pencere (Sequential Sliding Window, SSW) adını verdikleri yeni etkileşimli değişken gruplama yöntemi ile küresel ve yerel algoritmaları CC çerçevesine entegre ederek büyük ölçekli ayrılabilir problemlerin çözüm performansını etkin bir biçimde arttırmışlardır. CEC2010 LSGO test fonksiyon kümesiyle gerçekleştirilen test sonuçları, önerilen çerçevenin diğer mevcut CC çerçevelerinden daha etkili olduğunu göstermiştir (Cao ve ark., 2015).

CC, büyük ölçekli optimizasyon için güçlü bir parçala fethet mimarisi sağlamakla birlikte ayrıştırma başarısı da algoritmanın performansını büyük ölçüde etkilemektedir. Optimuma yakın ayrıştırmayı yapabilmek için geliştirilmiş ayrıştırma stratejilerinin çoğu ya doğru bağımlılık bilgisini elde edememekte ya da tanımlama için çok fazla uygunluk değerlendirme sayısı (Fitness Evaluation, FEs) harcamayı gerektirmektedir. Bu sınırlamaların etkisini azaltmak amacıyla Hu ve ark. tarafından büyük ölçekli optimizasyon için CC'de hızlı bağımlılık tanımlama algoritması önerilmiştir. Önerilen algoritma öncelikle ayrılabilir ve ayrılabilir değişkenleri etkili bir şekilde tanımlamakta, daha sonra ayrılabilir değişkenler için değişkenler arası bağımlılık bilgisini araştırmaktadır. CC için optimuma yakın ayrıştırma yapabilmek için algoritma ayrılabilir değişkenlerin değişkenler arası tam bağımlılık bilgisinin elde edilmesi gerekliliğinden kaçınmaktadır. Bu nedenle önemli miktarda FEs kazancı sağlanmaktadır. 2000 e varan değişkene sahip LSGO test fonksiyonları üzerinde geniş kapsamlı deneyler sonucunda, önerilen algoritmanın çok daha az FEs harcayarak birçok test fonksiyonunda doğru bir şekilde değişkenler arası bağımlılık bilgisini tanımlayabildiğini göstermiştir (Hu ve ark., 2017).

Sun ve ark., CEC2019 yarışmasında birinci olan CC-RDG3 adını verdikleri algoritma bünyesinde örtüşen büyük ölçekli problemlerin çözümü için bir ayrıştırma yöntemi önermişlerdir. RDG3 adını verdikleri yöntem, özyinelemeli diferansiyel gruplama (Recursive Differential Grouping, RDG) yönteminin birden fazla bileşen tarafından paylaşılan değişkenler arasındaki bağlantıyı keserek değiştirilmiş bir halidir. CEC2013 test fonksiyon kümesiyle gerçekleştirilen deneysel sonuçlarla, ayrıştırma yönteminin problem çözmeyi kolaylaştırdığını ve örtüşen problemlerde rastgele ayrıştırma ve diğer yöntemlerden daha iyi performans sergilediğini belirtmişlerdir (Sun ve ark., 2019).

Verimli kaynak tahsisi geleneksel CC'nin büyük ölçekli optimizasyon problemlerini çözüme performansını büyük ölçüde geliştiren bir yaklaşımdır. Çünkü CC yapısında alt popülasyonların en iyi çözüme ulaşma sürecine katkıları birbirinden farklı olabilmektedir. Bu kapsamda alt popülasyonlar arasında hesaplama kaynaklarını paylaştırırken genel amaç değerinin iyileştirilmesine olan katkılarına göre tahsis eden katkı tabanlı CC algoritmalar önerilmiştir (Omidvar ve ark., 2016; Yang ve ark., 2016). Bu algoritmalar alt problemlerin boyutluluğu gibi karakteristik özellikleri göz ardı ederek popülasyonu eşit alt popülasyonlara bölmektedir. Yang ve ark. ise yine katkı tabanlı çalışan bir algoritma bünyesinde iş birliğine dayalı birlikte evrim çerçevesi 2 (Cooperative Coevolution Framework2, CCFR2) adını verdikleri yöntemle tamamen ayrılabilir olan bir fonksiyonu çözmek için popülasyon yakınsamasını hızlandırabilecek küçük boyutlu bir alt popülasyon, tamamen ayrılamaz bir fonksiyonu çözmek için ise küresel arama yeteneğini geliştirebilecek büyük boyutlu bir alt popülasyon kullanmayı önermişlerdir (Yang ve ark., 2020).

Özetle, CC yaklaşımı LSGO için en çok tercih edilen tekniklerden biri olup geliştirmeye açık bir araştırma alanıdır. Bu yaklaşımın başarısı ayrıştırma stratejilerine karşı oldukça duyarlı olduğundan ayrılamaz fonksiyonlar söz konusu olduğunda performansı düşmektedir. Bu nedenle değişkenlerin otomatik olarak gruplandırılması için yeni yöntemler veya değişkenler arasındaki bağımlılıkları tespit eden farklı yaklaşımlar geliştirilebilir. Ayrıca ilgili maliyeti azaltarak CC tekniğini daha kullanışlı hale getiren kaynak tahsisi alanında da farklı geliştirmeler yapılabilir.

2.2. Ayrıştırma Esaslı Olmayan Yaklaşımlar

Bu bölümde, problemi bir bütün olarak ele alan yaklaşımlar değerlendirilmektedir. Keşif performansını artırmak için yeni operatörler tanımlama veya popülasyon büyüklüğünde değişiklikler yapma, yerel arama prosedürü tasarlama, farklı algoritma ve teknikleri hibrit kullanma vb. yöntemler ayrıştırma esaslı olmayan yaklaşımlar kategorisinde değerlendirilebilir. Birçok farklı sınıflandırma yapmak mümkün olsa da literatürde yapılan çalışmalar tez kapsamını destekleyecek şekilde standart optimizasyon algoritmaları ve memetik algoritmalar başlıkları adı altında sunulmuştur.

2.2.1. Standart optimizasyon algoritmaları

Standart optimizasyon algoritmaları büyük ölçekli optimizasyon problemlerinin çözümü için etkili yaklaşımlardan biri olarak kabul edilmektedir. GA, DE, PSO, karınca kolonisi algoritması (Ant Colony Optimization, ACO) ve yapay arı kolonisi (Artificial Bee Colony algorithm, ABC) gibi bilinen algoritmalar LSGO problemlerinin çözümü için de kullanılmıştır (Mahdavi ve ark., 2015).

Banitalebi ve ark. tarafından yapılan bir çalışmada, kendi kendini uyarlayabilen ikili DE algoritması (Self-Adaptive Binary Differential Evolution, SabDE) önerilmiştir. Farklılık ölçümüne dayanan algoritma bünyesinde, yeni olası çözümlerin nasıl üretileceğini seçmek için uyarlanabilir bir mekanizma ve parametre değerlerini uyarlamak içinse kaotik bir süreç kullanılmaktadır. Çalışma kapsamında SabDE, 10.000 boyuta kadar olan yüksek boyutlu sırt çantası (knapsack) problemlerinin yanı sıra CEC2015'in 15 öğrenmeye dayalı problemini içeren bir test fonksiyon kümesi üzerinde bir dizi algoritmayla karşılaştırılmıştır. Deneysel sonuçlar, önerilen algoritmanın rekabetçi bir performans gösterdiğini ve bazı durumlarda mevcut algoritmalarından üstün olduğunu ortaya koymuştur (Banitalebi ve ark., 2016).

Finans alanında da çözüm bulması zor büyük ölçekli karmaşık optimizasyon problemleri ile karşılaşmak mümkündür. Qiu ve Liu, büyük girdi boyutuna sahip büyük ölçekli karmaşık optimizasyon modelini çözme yeteneğine sahip yeni bir algoritma geliştirmek amacıyla piyasa rekabeti davranış algoritmasını (Market Competition Behavior Algorithm, MCB) önermişlerdir. Farklı mekanizmaların bir kombinasyonu şeklinde tasarlanan MCB algoritması, arama uzayındaki çözüm noktalarını atalet ve

yerçekimi kuralına göre güncellemekte, her iterasyonda eski işletmeleri hariç tutarken yeni işletmeleri tanıtarak yerel optimum çözüme düşmekten kaçınarak çözüm arama verimliliğini hızlandırmak için hız vektörünü yeniden birleştirmektedir. Bu çalışmada önerilen algoritma risksiz ve riskli varlıklar üzerine hibrit bir yatırım optimizasyon modeline uygulanarak sonuçları değerlendirilmiştir (Qiu ve Liu, 2016).

Mohapatra ve ark., rekabetçi sürü optimizeri (Competitive Swarm Optimizer, CSO) algoritmasının üçlü rekabet kriteri aracılığıyla popülasyonun üçte ikisinin güncellendiği bir versiyonunu önermişlerdir. Çözüm kalitesinde büyük bir fark yaratan bu küçük değişikliğin arkasındaki temel fikir, daha hızlı bir yakınsama oranı ile arama uzayında daha yüksek bir keşif imkânı sağlamaktır. Önerilen değiştirilmiş rekabetçi sürü optimizeri (Modified Competitive Swarm Optimizer, MCSO) standart CEC2008 ve CEC2013 LSGO test fonksiyon kümeleri üzerinde test edilmiş ve elde edilen sonuçlar önerilen yöntemin standart CSO'nun da aralarında bulunduğu birçok metasezgisel algoritmaya göre daha iyi performans gösterdiğini doğrulamıştır (Mohapatra ve ark., 2017).

Ismkhan, bir diğer bilinen optimizasyon algoritması olan ACO üzerine çalışmıştır. ACO, çok çeşitli optimizasyon problemlerine başarıyla uygulanmış olsa da yüksek zaman ve alan karmaşıklığı, büyük ölçekli örneklere uygulanmasını engellemektedir. Ayrıca, ACO'da performansını artırmak için kullanılan yerel arama, feromon değerlerinde saklanan buluşsal bilgiler kullanılmadan uygulanmaktadır. Bu sorunların üstesinden gelmek için Ismkhan, ACO'yu hızlandıran ve büyük ölçekli örneklere uygulanmasını sağlayan yeni stratejiler önermiş ve performans testlerinden elde ettiği sonuçlarla önerilen ACO versiyonunun doğruluk ve hız açısından diğer sürümlerden daha iyi performans gösterdiğini belirtmiştir (Ismkhan, 2017).

Long ve ark. tarafından yapılan bir diğer çalışmada ise gri kurt optimizasyon (Gray Wolf Optimizer, GWO) algoritması üzerine çalışılmıştır. GWO, az kontrol parametresi içermesi, güçlü global optimizasyon kabiliyeti ve uygulama kolaylığı olan popülasyon temelli bir optimizasyon tekniği olmakla birlikte sömürde iyidir. Ancak bu algoritma keşifte, zayıf olan pozisyon güncelleme denklemi nedeniyle hala yetersizliklere sahiptir. Long ve ark. çalışmalarında bu yetersizliği gidermek için çalışmalar yaparak keşif özelliği geliştirilmiş gri kurt optimizasyon algoritması (Exploration-Enhanced Gray Wolf Optimizer, EEGWO) adı verilen yöntem önermişlerdir. Keşif özelliğini güçlendirmek için popülasyondaki yeni aday çözümleri arayan ve rastgele bir bireye uygulanan yeni bir pozisyon güncelleme denklemi sunulmuştur. Buna ek olarak GWO algoritmasının keşif ve sömürü dengesinden tam olarak faydalanabilmek için doğrusal olmayan bir kontrol

parametresi stratejisi tanıtılmıştır. Bu çalışmada, 23 test fonksiyonu ve 4 mühendislik uygulamasından elde edilen sonuçlara göre yapılan geliştirme, GWO nun performansını önemli ölçüde artırmıştır (Long ve ark., 2018).

Çok sayıda optimizasyon algoritması arasında, DE (Storn ve Price, 1997) ve PSO (Eberhart ve Kennedy, 1995), basitlikleri ve verimlilikleri nedeniyle büyük ölçekli optimizasyon alanında diğer algoritmalarından daha fazla avantaja sahiptir. Bu nedenle bu iki algoritmaya dayalı birçok strateji önerilmiştir. Özellikle DE, LSGO alanında en etkili algoritmalarından biri olup geliştirilen bir çok etkili teknik, algoritma veya çerçevede de temel metasezgisel algoritma olarak çokça tercih edilmektedir (Molina ve ark., 2018a; Del Ser ve ark., 2019). Maučec ve Brest yayınladıkları derleme makalesinde teorik bir bakış açısıyla LSGO sorunları ile başa çıkmak için önerilen DE mekanizmalarını incelemiş ve yapılan çalışmaları 9 ayrı kategoride sınıflandırarak açıklamışlardır. Ayrıca LSGO alanında önerilmiş on altı algoritmanın CEC2013 test fonksiyon kümesi performanslarını değerlendirmişlerdir (Maučec ve Brest, 2019).

PSO, yüksek yakınsama hızına sahip bir algoritmadır. Ancak büyük ölçekli optimizasyon problemlerinin çözümünde en büyük dezavantajı erken yakınsamadır. PSO'nun bu dezavantajının üstesinden gelmek için Huang ve ark., bir yakınsama hız denetleyicisi önermişlerdir. Bu çalışmada PSO'nun ek bir operatörü olarak bir denetleyici periyodik ve bağımsız olarak uygulanmaktadır. Yakınsama hızı denetlenen PSO'nun etkinliği, CEC2010 test fonksiyon kümesi ile test edilmiş ve sayısal sonuçlar, önerilen denetleyicinin PSO'nun optimizasyon süreci boyunca yakınsama hızı ve sürü çeşitliliği arasında bir denge sağlamasına yardımcı olduğunu göstermiştir (Huang ve ark., 2019).

Hızlı yakınsama özelliği nedeniyle tercih edilen PSO, bazı büyük ölçekli optimizasyon problemlerinde rekabetçi performans göstermekle birlikte kolayca yerel optimuma takılıp kalabilmektedir. Takviyeli öğrenme yönteminin büyük ölçekli çözüm uzayında daha etkili bir şekilde daha iyi sonuçlar aramaya yardımcı olabilecek bir teknik olması nedeniyle Wang ve ark., pekiştirmeli öğrenme düzeyi tabanlı parçacık sürü optimizasyonu algoritması (Reinforcement Learning Level-based Particle Swarm Optimization, RLLPSO) adı verilen büyük ölçekli bir optimizasyon algoritması önermişlerdir. Arama verimliliğini artırmak için bir pekiştirmeli öğrenme stratejisi, popülasyon çeşitliliğini iyileştirmek için seviye tabanlı bir popülasyon yapısı, yakınsama yeteneğini daha da geliştirmek için bir seviye rekabet mekanizması içeren bu algoritma beş büyük ölçekli optimizasyon algoritmasıyla karşılaştırılmış ve önerilen algoritmanın çoğu durumda üstünlük gösterdiğine değinilmiştir (Wang ve ark., 2022).

Özetle, standart evrimsel algoritmalar büyük ölçekli problemlerin çözümünde boyut artışı ve beraberinde getirdiği arama uzayının katlanarak büyümesi nedeniyle orijinal tasarımındaki etkisini kaybetmektedir. Yaşanan bu performans düşüşünün önüne geçmek amacıyla halihazırda kullanılan standart optimizasyon algoritmaların ölçeklenebilirliğini artıracak yüksek boyuta adaptasyonunu sağlayacak geliştirme ve düzenlemeler yapılması gelişime açık bir araştırma alanıdır.

2.2.2. Memetik algoritmalar

Wolpert ve Macready tarafından 1997 yılında ortaya atılan Bedava Öğle Yemeği Yok Teoremi (No free lunch theorem), tüm olası problemlerin ve tüm optimizasyon algoritmalarının bir dağılımını düşünerek optimizasyon alanında evrensel bir en iyi algoritma veya yöntem bulunamayacağını tezini savunmaktadır. Bu teorem, optimizasyon problemlerinin ve algoritmalarının karmaşıklığına ve çeşitliliğine dikkat çekerek her problemin kendine özgü gereksinimleri olduğunu dolayısıyla kendi doğası gereği belirli bir optimizasyon yöntemini veya algoritmasını gerektireceğini savunmaktadır. Başka bir deyişle tek bir algoritmanın tüm problemlerde aynı derecede başarılı olamayacağını belirtmektedir. Optimizasyon araştırmacıları yıllarca literatürde bulunan diğer tüm algoritmalara göre üstün performansa sahip algoritmalar tasarlamaya çalışmışlardır. Fakat bu teoremin yayılımından sonra bu konu hakkındaki görüşlerini önemli ölçüde değiştirerek önerilen algoritmaların bileşenleri ve çözülmesi hedeflenen problemin yapısı arasındaki ilişkiyi anlamaya ve bu doğrultuda geliştirmeler yapmaya odaklanmışlardır.

Bedava Öğle Yemeği Yok Teoremi dikkate alındığında belirli algoritmaların diğerlerinden daha iyi performans gösterdiği problem alanlarının olduğu sonucuna da varılmaktadır. Her tekniğin zayıf yanları olduğu gibi güçlü yanları da bulunmaktadır. Bu durum bireysel algoritmaların tek başlarına elde ettiği çözümlerden daha yüksek kaliteye sahip çözümler elde etmek amacıyla bir iş birliği oluşturma fikrini ortaya çıkarmıştır. Hibritleştirme denilen bu yöntem, birden fazla farklı optimizasyon tekniğinin veya algoritmanın bir arada kullanıldığı bir optimizasyon yaklaşımıdır. Bu teknikle farklı optimizasyon yöntemlerinin güçlü yönlerini ve birbirini tamamlayıcı özelliklerini birleştirerek daha etkili ve verimli sonuçlar elde etmek, bir yöntemin dezavantajlarını diğer yöntemlerle telafi edebilmek amaçlanmaktadır.

Bu bağlamda LSGO alanının temel sorunu, büyüyen boyutla birlikte arama uzayının katlanarak artması nedeniyle arama uzayının etkin ve verimli bir şekilde taranması zorluğudur. Bir algoritmanın temel performansı bu taramayı ne kadar başarabildiğine bağlıdır. Arama sürecinde bunu gerçekleştirecek iki ana unsur keşif (exploration) ve sömürü (exploitation) mekanizmalarıdır. Keşif, küresel en uygun çözüme ulaşmak için arama uzayındaki tamamen yeni noktaları ziyaret ederek genel arama yaparken, sömürü varolan çözümleri iyileştirerek daha kaliteli çözümler üretmek için daha önce ziyaret edilen iyi çözümlerin çevresinde yerel arama yapmaktadır. Çözüm yaklaşımlarının başarısı, keşif ve sömürü arasında bazen birbiriyle çelişen iki hedef olsa dahi taktiksel bir denge kurulabilmesine bağlıdır. Dolayısıyla büyük çözüm uzayını daha iyi keşfedebilen daha iyi ölçeklenebilir algoritmalar her zaman ihtiyaç vardır. Bu amaçla evrimsel algoritmaların yerel arama teknikleriyle hibritleştirilmesi MA olarak tanımlanır (Moscato, 1989; 1999). Başka bir deyişle MA, arama alanından yararlanmak için uygunluk değerlerini iyileştirmek amacıyla ekstra yerel arama yöntemleri kullanmak gibi ayrı süreçler uygulayan bir tür evrimsel algoritmadır (Krasnogor ve Smith, 2005). MA terimi optimizasyon için yerel iyileştirme prosedürleri veya ayrı bireysel öğrenmeli popülasyona dayalı yaklaşımlar için yaygın bir şekilde kullanılmaktadır (Wang ve Feng, 2021).

MA'lar karmaşık gerçek dünya problemlerini çözmek için başarıyla uygulanmaktadır (Neri ve ark., 2007; Tirronen ve ark., 2008; Hasan ve ark., 2009; Li ve Pan, 2015; Lu ve ark., 2020; Lan ve ark., 2021). Ayrıca çok sayıda vakada gösterdikleri yüksek performans sayesinde geleneksel muadillerine göre daha verimli bir şekilde daha yüksek kaliteli çözümlere yakınsama sağlayabildikleri kanıtlanmıştır (Neri ve Cotta, 2012; Wang ve Feng, 2021).

İlgili makaleler ve özel oturumlarda düzenlenen yarışmalar dikkate alındığında, MA'ların LSGO araştırma alanında önemli ve etkili bir yere sahip olduğu açıkça görülmektedir. Bu yarışmalardaki katılımcı algoritmaların incelenmesi, farklı yıllardaki geliştirme eğilimlerini tespit etme, araştırma alanının güncel gelişimi ve gelecekte yapılabilecek geliştirmelere dair temel bilgiler edinme imkânı sağlamaktadır. Çizelge 2.1'de CEC yarışmalarında başarılı olmuş memetik yaklaşımı benimseyen etkili algoritmalar listelenmiştir.

Çizelge 2.1. CEC LSGO yarışmalarında dereceye giren memetik algoritmalar

Referans	Algoritma Adı	İçerdiği Algoritmalar	İçerdiği Yerel Arama Metotları	Yarışma Derecesi
(Tseng ve Chen, 2008)	Çoklu yörünge araması (MTS)	Üç farklı yerel aramanın kombinasyonu	MTSLS1, MTSLS2, MTSLS3	CEC2008 Birincisi
(Molina ve ark., 2010b)	Yerel arama zincirlerine dayalı memetik algoritma (MA-SW-Chains)	GA	Solis & Wets	CEC2010 Birincisi
(LaTorre ve ark., 2012)	Çoklu yavru örnekleme algoritması (MOS)	GA, DE, SADE, GODE, SAGODE	Solis & Wets ve MTSLS1	CEC2012 Birincisi
(LaTorre ve ark., 2013)	Geliştirilmiş çoklu yavru örnekleme algoritması (MOS2013)	GA, DE, SADE, GODE, SAGODE	Solis & Wets ve MTSLS1 Reduced	CEC2013 ve CEC2015 Birincisi
(Molina ve Herrera, 2015)	Diferansiyel gelişim algoritmasının yerel arama ile yinelemeli hibridizasyonu (IHDELS)	DE	MTSLS1 ve L-BFGS-B	CEC2015 İkincisi
(Molina ve ark., 2018b)	Yinelemeli yerel aramaya sahip başarı geçmişine dayalı uyarlanabilir diferansiyel gelişim algoritması (SHADE-ILS)	SHADE	MTSLS1 ve L-BFGS-B	CEC2018 Birincisi
(Hadi ve ark., 2019)	Doğrusal popülasyon boyutu azaltma ve yarı parametre uyarlamalı başarı geçmişine dayalı memetik diferansiyel gelişim çerçevesi (MLSHADE-SPA)	LSHADE-SPA	Modified MTSLS1	CEC2018 İkincisi

Literatürdeki MA'lar kullandıkları yerel arama yöntemleri açısından incelendiğinde MTSLS1 ve Solis and Wets yerel arama yöntemlerinin bariz bir şekilde öne çıktığı görülmektedir (Omidvar ve ark., 2021b).

LSGO yarışmalarının ilk galibi, CEC2008'de sunulan üç yerel aramanın bir kombinasyonu olan çoklu yörünge araması (Multiple Trajectory Search, MTS) olmuştur. Bu teknikte, çözüm uzayını eş zamanlı arayan birden çok ajan kullanılmakta ve her ajan aday, 3 yerel arama tekniğinden birini kullanan iteratif yerel arama yapmaktadır (Tseng ve Chen, 2008). Bu algoritmadaki yerel arama yöntemleri arasında LSGO için en etkili olanı çoklu yörünge araması yerel arama 1 (Multiple Trajectory Search Local Search 1, MTSLS1)'dir. MTSLS1, ilk boyuttan son boyuta kadar her boyutu tek tek değerlendiren ve ayrılabilir problemler için uygun ancak döndürmelere karşı duyarlı olan bir yerel

arama yöntemidir (Molina ve ark., 2018b). MTSLs1 yerel araması ve varyantları büyük ölçekli optimizasyon alanındaki sonraki çalışmalara da ilham olmuştur ve bu yöntem yayın olarak kullanılmıştır (Zhao ve ark., 2011; LaTorre ve ark., 2012; 2013; Molina ve Herrera, 2015; Salcedo-Sanz ve ark., 2016; Molina ve ark., 2018b; Hadi ve ark., 2019).

Solis ve Wets algoritması (Solis ve Wets, 1981), adım boyutunu hareketin ardışık başarı ve başarısızlıklarına göre uyarlayan yerel bir arama yöntemidir. LSGO alanında öncü olan algoritmalar tarafından memetik algoritmalar bünyesinde kullanılmıştır (Molina ve ark., 2010b; Molina ve ark., 2011; LaTorre ve ark., 2012; 2013; Liu ve Li, 2014; Cao ve ark., 2015).

MA tasarımında karar verilmesi gereken önemli konular: yerel arama uygulama sıklığı, yerel arama uygulanacak çözüm seçimi, yerel arama uygulama süresi, yerel arama algoritmasının seçimi olarak sıralanabilir (Tang ve ark., 2014).

Bu konulardan yerel arama uygulama sıklığını kontrol etmenin en yaygın yolu kullanıcı tanımlı bir parametre tarafından kontrol edilen düzenli aralıklarla yerel aramayı çalıştırmaktır (Salcedo-Sanz ve ark., 2016). Bir diğer yaklaşım, yerel aramayı sabit düzenli aralıklarla çalıştırmak yerine, yerel ve genel arama arasındaki oran sabit olacak şekilde yerel arama operatörleri uygulama yöntemini kullanmaktır (Fister ve Žumer, 2012). Bu kapsamda parametreleri başarı/başarısızlık oranına göre ayarlanan ve/veya uyarlanan önceden belirlenmiş bir dağılım kullanarak yerel aramanın çağrılmasına olasılıksal olarak karar veren yaklaşımlarda mevcuttur (Molina ve Herrera, 2015).

Yerel arama uygulanacak çözüm seçimi rastgele, performansa dayalı, çözümlerin tümü olacak şekilde yapılabilir. Performansa dayalı yöntemler yerel uygulamayı en iyi çözüme uygulamaktadır (Molina ve Herrera, 2015; Salcedo-Sanz ve ark., 2016). En iyi n adet çözüme uygulamak (Tseng ve Chen, 2008), popülasyonun en iyi yüzde x ine uygulamak (Zhao ve ark., 2008) ve önceki iterasyonlarda önceden tanımlanmış bir eşikten daha fazla geliştirilmiş olanlara uygulamak (Molina ve ark., 2010b) gibi birçok farklı yöntemden biri tercih edilebilir. Diğer taraftan Zhao ve ark.nın yaptığı gibi beş rasgele çözüm, en iyi çözüm ve popülasyonun ilk %25'ini seçmek gibi birkaç yöntemi birlikte kullanmak tercih edilebilir (Zhao ve ark., 2011).

Yerel arama yoğunluğu olarak ta geçen yerel arama uygulama süresini belirlemenin en temel yolu yerel arama operatörünü sabit ve önceden belirlenmiş sayıda FEs için çalıştırmaktır (Fister ve Žumer, 2012; Molina ve Herrera, 2015). Bir diğer uygulama açgözlü bir yaklaşım kullanarak hiçbir gelişme gözlemlenmeyene kadar yerel arama algoritmasını çalıştırmaktır (Tseng ve Chen, 2008; Salcedo-Sanz ve ark., 2016).

Bazı algoritmalar ise sabit bir yerel arama/global arama oranını zorlayarak arama sıklığı ve yoğunluğunu birbirine bağlar (Molina ve ark., 2011). Arama yoğunluğunu belirlemenin daha karmaşık bir yolu ise bunu yerel arama prosedürünün başarı/başarısızlık oranına dayalı olarak belirlemek gibi optimizasyon sırasında uyarlamalı olarak yapmaktır (Liu ve Li, 2014).

MA tasarımında kullanılacak yerel arama yönteminin seçimi, algoritmanın performansını büyük ölçüde etkileyen bir öneme sahiptir. Çoğu durumda, küresel keşif sürecinde tek bir yerel arama prosedürü kullanılır. Bununla birlikte uygulamasının hibrit bir çerçeve tarafından koordine edilmesi gereken yerel veya küresel ikiden fazla aramanın aynı anda kullanıldığı algoritmalar da mevcuttur. Örneğin, Fister ve Žumer tarafından önerilen bir algoritma, parametresi popülasyon çeşitliliğinin bir ölçüsüne göre uyarlanan üstel bir dağılıma dayalı olarak rastgele yürüyüş ile Nelder-Mead (Nelder ve Mead, 1965) arasında olasılıksal olarak seçim yapar (Fister ve Žumer, 2012). MTS, üç yerel arama operatörünü her döngünün başında n adet çözüm üzerinde çalıştırarak koordine eder ve döngünün geri kalanında durgunlaşana kadar en iyi performans gösteren operatörü kullanır (Tseng ve Chen, 2008). Bir MA çerçevesinde CC kullanan bazı algoritmalar ise problemi ayrılabilir ve ayrılmaz bileşenlere ayırır ve her bileşen için uygun farklı yerel arama operatörleri kullanır (Liu ve ark., 2017; Sun ve ark., 2017b). MOS, iki arama operatörünün koordinasyonunu sistemleştiren, beş adet evrimsel algoritmanın yeniden üretim operatörlerinin üstünde bir soyutlama katmanı olan çerçeve bir algoritmadır (LaTorre ve ark., 2011; LaTorre ve ark., 2012; 2013). MOS çerçevesi yüksek boyutlu problemlerdeki ölçeklenebilirliği sayesinde büyük ölçekli küresel optimizasyonda CEC2013 ve CEC2015 yarışma birincisi olmuştur. Çerçeve algoritma mantığını uygulayan bir diğer algoritma CEC2018 yarışma ikincisi olan ve bu tez kapsamında kullanılan ana algoritma olan doğrusal popülasyon boyutu azaltma ve yarı parametre uyarlamalı başarı geçmişine dayalı memetik diferansiyel gelişim (Memetic Success History-based Differential Evolution with linear population size reduction and Semi-Parameter Adaptation, MLSHADE-SPA) algoritmasıdır (Hadi ve ark., 2019). Memetik çerçeve bünyesinde 3 popülasyon tabanlı algoritma ve 1 yerel arama yöntemi kullanılmaktadır. Popülasyon tabanlı algoritmalar, doğrusal popülasyon boyutu azaltma ve yarı parametre uyarlamalı başarı geçmişine dayalı diferansiyel gelişim (Success History-based Differential Evolution with linear population size reduction and Semi-Parameter Adaptation, LSHADE-SPA), geliştirilmiş uyarlanabilir diferansiyel gelişim algoritması (Enhanced Adaptive Differential Evolution, EADE) ve yeni üçgenel

mutasyon stratejili uyarlanabilir diferansiyel gelişim algoritması (Adaptive DE with Novel triangular mutation strategy, ANDE) dır. Çerçeve bünyesinde kullanılan yerel arama ise MTSLS1'in deęiştirilmiş bir versiyonu olan deęiştirilmiş çoklu yörünge araması (Modified Multiple Trajectory Search, MMTS) metodudur.

Popülasyona dayalı bir yöntem ile yerel aramanın kombinasyonu düzgün bir şekilde yönetilirse, keşif ve sömürü arasında iyi bir denge kurularak etkili ve verimli başarılar elde edilebilir. MA bünyesinde kullanılan yerel arama algoritmasının arama performansı üzerinde büyük bir etkiye sahip olduęu giderek daha fazla kabul görmektedir (Ong ve Keane, 2004; Smith, 2007). Bu tez kapsamında büyük ölçekli optimizasyon problemlerinin çözümünde kullanılan memetik algoritmaları güçlendirecek, daha etkili ve kararlı hale getirecek yeni bir yerel arama yöntemi geliştirilmesi ve bu yolla büyük ölçekli optimizasyon problemlerinin çözümüne katkıda bulunulması hedeflenmiştir.

Özetle Memetik Algoritmalar, LSGO alanında oldukça etkili ve başarılı yöntemlerden biri olduęu için bileşenlerini ayrı ayrı veya kombinasyonlarını kullanarak farklı teknik ve hibritleştirme yöntemleriyle geliştirilmeye oldukça açık bir alandır. MA'lar karmaşık sürekli optimizasyon problemleri için yüksek hassasiyette güvenilir çözümler elde etme zorluęunu gidermek için etkili araçlar olarak kullanılmaya ve geliştirilmeye devam etmektedir.

3. MATERYAL VE YÖNTEM

Bu bölümde, geliştirme amacıyla üzerinde çalışılan iki ana algoritma olan Yapay Alg Algoritması (Artificial Algae Algorithm, AAA) ve MLSHADE-SPA algoritmalarının işleyişine, yerel arama yöntemi geliştirilmesinde yararlanılan kavramlar olan altın oran ve kaotik haritalar kavramlarına ve son olarak deneysel çalışmalarda kullanılan test fonksiyon kümelerine dair bilgilere yer verilmiştir.

3.1. Yapay alg algoritması

AAA, hareketli mikro alglerin karakteristik özellikleri ve yaşam davranışlarından esinlenilerek tasarlanan biyolojik ilhamlı bir metasezgisel optimizasyon algoritmasıdır (Uymaz, 2015).

Popülasyon tabanlı olan algoritmada popülasyonu oluşturan her bir birey yapay alg kolonisi olarak adlandırılır yani her bir yapay alg kolonisi problem uzayında bir çözüme karşılık gelmektedir. Bir alg kolonisi, birlikte yaşayan alg hücrelerinden oluşur dolayısıyla bir alg kolonisindeki her bir alg hücresi, problemin bir parametresini temsil etmektedir.

Yapay alg, gerçek hayattaki algin fotosentez yapabilmek için ışık kaynağına doğru hareket etmesi gibi çözüm uzayında daha iyi çözümlere doğru helisel yüzerek hareket eder. Yaşam döngüleri çevreye uyum sağlamak, baskın türleri değiştirmek ve mitozla çoğalmaktan ibarettir. AAA, bu yaşam döngüsü davranışlarını helisel hareket, evrimsel süreç ve adaptasyon olmak üzere üç aşamada taklit ederek optimum çözümü bulmaya çalışmaktadır.

Yapay alg hücresi yeterli ışık aldığı anda gelişip büyümekte, yeterli ışık alamayan yapay alg hücresi ise bir süreliğine yaşamasına rağmen sonuçta ölmektedir. Tüm alg kolonileri için bir çevrim tamamlandıktan sonra işletilen evrimsel süreçte, alg kolonileri büyüklüklerine göre sıralanır. En küçük yani en başarısız alg kolonisinin rastgele seçilen bir boyuttaki bir hücresi ölür ve bu ölen hücrenin yerine en büyük yani en başarılı alg kolonisinin rastgele belirlenen bir hücresi kopyalanarak evrimsel süreç işletilmiş olur.

Alg kolonilerinin açlık seviyeleri kılavuzluğunda işletilen adaptasyon sürecinde ise, hayatta kalan fakat bulunduğu ortamda yeterince büyümeyen alg kolonisi bulunduğu ortama uyum sağlamak amacıyla en büyük alg kolonisine benzemeye çalışmaktadır. Başlangıçta tüm alg kolonilerinin açlık seviyeleri sıfır olarak kabul edilir. Her bir alg

kolonisi helisel hareketi sonucu çözüm uzayında ya daha iyi bir çözümü temsil eden daha iyi bir yerine ya da daha kötü bir çözümü temsil eden daha kötü bir yerine gitmektedir. Daha iyi bir çözüm bulan yani başarılı bir geliştirme yapan alg kolonisinin açlık seviyesi aynı kalır fakat hareketi sonucu daha iyi bir çözüm bulamayan yani kendini geliştiremeyen alg kolonisinin açlık seviyesi bir artar. Her bir helisel hareket çevrimi sonucu, yani tüm alg kolonileri hareketlerini tamamladıktan sonra, adaptasyon sürecinin gerçekleşip gerçekleşmeyeceğini belirlemek için 0 ile 1 arasında rastgele gerçek değerli bir sayı üretilerek başlangıçta verilen parametre ile karşılaştırılır. Eğer üretilen sayı parametreden küçük ise açlık seviyesi en yüksek olan alg kolonisi adaptasyon işlemine tabi tutulur.

Algoritmanın bir diğer aşaması olan helisel hareket ile AAA'daki aday çözümleri temsil eden alg kolonilerinin hücreleri güncellenmekte ve çözüm uzayında yeni olası çözüm aramak için yer değiştirmesi sağlanmaktadır. Genellikle yüzer vaziyette olan alg hücreleri ve kolonileri, yeterli ışık kaynağı su yüzeyinde bulunduğundan hayatta kalabilmek için su yüzeyine yakın olmak amacıyla kamçıları sayesinde sıvı içerisinde helisel yani sarmal şekilde yüzerek hareket etmektedirler. Her alg kolonisi belirli bir enerjiye sahiptir ve alg kolonisinin her hareketi belirli bir enerji kaybına neden olmaktadır. Çünkü yüzen bir alg hücresinin ileri hareketi, yerçekimi ve hareketli organizma hücresinin etrafındaki su ile arayüzünde gerçekleşen viskoz sürtünme kuvveti tarafından engellenmeye çalışılır. Alg kolonisinin enerjisi çözüm kalitesi ile doğru orantılıdır. Yani alg kolonisi ışığa ne kadar yakın olursa büyüklüğü ve dolayısıyla enerjisi arttığından sıvı içerisinde daha fazla hareket imkânı bulurken, ışığa uzak olan alg kolonileri enerjileri oranında daha az hareket etmektedirler. Her alg hücresinin bu etkenlere bağlı gerçekleşen farklı hareketi algoritmanın hem keşif hem de sömürü yeteneklerini etkilemektedir. Şöyle ki, büyüyen alg hücrelerinin sürtünme yüzeylerinin büyük olması nedeniyle sıvı içerisinde daha yavaş hareket etmesi, büyük alglerin yani algoritmadaki iyi değerlerin yerinin daha az değişmesi dolayısıyla o alanda yoğunlaşmaya olanak sağlayarak yerel arama(sömürü) yeteneğini artırmaktadır. Küçük alg hücrelerinin ise sürtünme yüzeylerinin daha az olması nedeniyle sıvı içerisinde daha hızlı ve daha uzağa hareket edebilmesi algoritmanın kötü çözüm alanlarından uzaklaşmasını sağlayarak küresel arama(keşif) yeteneğini artırmaktadır.

AAA'yı oluşturan süreçlerden, evrimsel süreç ve adaptasyon süreci sömürü mekanizmasını desteklerken helisel hareket süreci hem keşif hem de sömürü mekanizmasına katkıda bulunmaktadır. Algoritma tasarımındaki bu aşamalar döngüsel

olarak tekrarlanmakta ve sonlandırma kriterine ulaşıldığı andaki en büyük alg kolonisi algoritmanın bulduğu çözümü ifade etmektedir. Algoritmanın sözde kodu Çizelge 3.1’de verilmekle birlikte algoritma hakkında ayrıntılı bilgi için (Uymaz, 2015) incelenmelidir.

Çizelge 3.1. Yapay Alg Algoritmasının sözde kodu (Kocer ve Uymaz, 2018)

```

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

```

3.2. Doğrusal popülasyon boyutu azaltma ve yarı parametre uyarlamalı başarı geçmişine dayalı memetik diferansiyel gelişim çerçevesi (MLSHADE-SPA)

MLSHADE-SPA, büyük ölçekli optimizasyon problemlerini çözmeye amacıyla üç popülasyon tabanlı algoritma ve bir yerel arama yönteminin hibritleştirilmesiyle tasarlanan memetik bir çerçeve algoritmadır (Hadi ve ark., 2019). Bu algoritma bünyesinde popülasyona dayalı algoritmalar küresel keşif için kullanılırken yerel arama metodu sömürü yeteneğini artırmak için kullanılmaktadır. Çerçeve algoritma bünyesinde kullanılan algoritmalar, LSHADE-SPA (Mohamed ve ark., 2017), EADE (Mohamed, 2017) ve ANDE (Mohamed ve Almazayad, 2017) dir. Yerel arama olarak ise MTSLS1'in değiştirilmiş bir versiyonu olan MMTS (Hadi ve ark., 2019) metodu kullanılmaktadır. Bu algoritma işleyişinin bir parçası olarak nüfus azaltma, parçala ve fethet yaklaşımı, hesaplama kaynaklarını paylaşırma gibi çeşitli teknikleri de bünyesinde uygulamaktadır (Hadi ve ark., 2019).

Algoritmanın temel işleyişi, anlaşılabilirliği artırmak amacıyla başlatma, popülasyon tabanlı algoritmalar, yerel arama ve sonlandırma olmak üzere dört aşamada açıklanabilir.

Başlatma aşamasındaki ilk işlem arama uzayına dağılmış olası çözümleri temsilen rastgele üretilmiş n bireye sahip bir başlangıç popülasyonu oluşturmaktır ($n=250$). Ardından çerçeve bünyesinde kullanılan algoritma ve metotlara (LSHADE_SPA, EADE, ANDE, MMTS) ait tüm parametreler belirtilerek başlangıç değerleri atanır.

Optimizasyon sürecinde performansı etkileyen önemli faktörlerden biri de optimum çözüme ulaşabilmek için tüketilen uygunluk değerlendirme sayısıdır. Bu kapsamda başlatma aşamasında çerçeve algoritma bünyesinde kullanılacak FEs tamamı (maxFEs) iterasyon sayısına ($\text{it}=50$) bölünerek iterasyon başına harcanabilecek uygunluk değerlendirme sayısı ($\text{iFEs}=\text{maxFEs}/\text{it}$) tespit edilir. Ardından her bir iterasyon kendi bünyesinde iki ayrı süreç içerdiği için eşit kaynak paylaşımı yapmak amacıyla iFEs değeri ikiye bölünür ve yarısı popülasyon tabanlı algoritmalar tarafından tüketilirken ($\text{iFEsPop}=\text{iFEs}/2$), diğer yarısı da yerel arama yöntemi tarafından tüketilir ($\text{iFEsLS}=\text{iFEs}/2$).

Çerçeve algoritmanın popülasyon tabanlı algoritmalar aşamasında, LSHADE_SPA, ANDE, EADE olmak üzere üç algoritma görev yapmaktadır. İterasyonun popülasyon tabanlı algoritmalar kısmı da iFEsPop , iki eşit parçaya bölünür. Yarısını bütün boyutlar üzerinde çalışan LSHADE_SPA algoritması tüketirken diğer yarısı da üç algoritma (LSHADE_SPA, ANDE, EADE) tarafından tüketilir. Bu üç algortmadan her biri parçala fethet anlayışıyla kendisine düşen boyutlarda kendine düşen değerlendirme sayısını tüketerek çalışırlar. Üç algoritma arasındaki paylaşım ilk iterasyonda eşit yapılırken daha sonraki iterasyonlarda her algoritmanın kendi performansını gösteren iyileştirme oranına göre $\text{iFEsPop}/2$ kısmı üç algoritma arasında paylaşılır. Bu aşamada bahsi geçen her bir algoritma yani LSHADE_SPA, ANDE, EADE algoritmalarından her biri, üç rastgele bölünmüş gruptan biri üzerinde çalışmaktadır. Başka bir deyişle, popülasyon tabanlı algoritmaların her biri, alt bileşen optimize ediciler olarak, kendisine atanan boyutlara odaklanır, performansına göre kendisine tahsis edilen FEs tüketerek sorunu optimize etmeye çalışır.

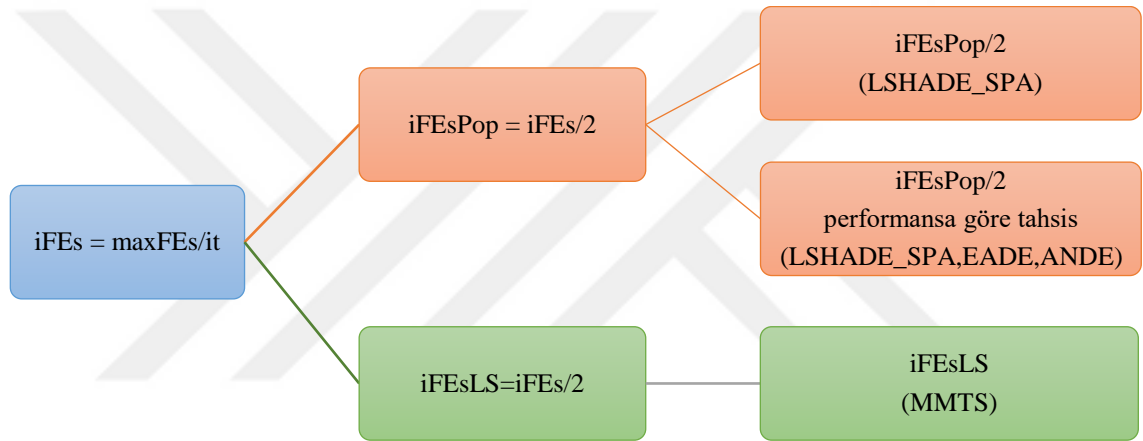
Algoritma bünyesinde herbir iterasyonda gerçekleştirilen FEs paylaşımı Şekil 3.1’de gösterilmektedir.

Evrimsel algoritmalarda arama süreci genellikle keşifle başlar ve ardından yavaş yavaş sömürüye dönüşür (Črepinšek ve ark., 2013). Popülasyon tabanlı algoritmalar aşaması ile keşif sürecini işleten çerçeve algoritma, yerel arama aşamasıyla sömürü sürecine geçer. İterasyonun bu aşamasında, iFEsLS kadar uygunluk değerlendirme sayısı tüketen değiştirilmiş çoklu yörunge araması metodu (MMTS), şimdiye kadar bulunan en iyi çözümü daha da geliştirmek amacıyla en iyi çözümün her bir boyutu üzerinde tek tek işletilir.

Çerçeve algoritmanın son aşaması olan sonlandırma kısmında ise, algoritmanın performansını artırmak amacıyla her iterasyon sonunda popülasyon büyüklüğünü lineer

bir fonksiyona göre azaltan doğrusal popülasyon büyüklüğünü azaltma yöntemi uygulanır. Bu yöntemle göre hesaplanan küçültme sonrası oluşacak popülasyon büyüklüğü hesaplandıktan sonra bireyler uygunluk değerine göre sıralanır. En kötü uygunluk değerine sahip bireyler, hesaplanan küçültme sonrası oluşacak popülasyon büyüklüğüne ulaşana kadar mevcut popülasyondan çıkartılır. Başlangıçta n olan popülasyon büyüklüğü, kullanılabilir tüm hesaplama kaynağının (maxFEs) ilk yarısında minimum birey sayısı olan 20'ye ulaşır.

50 iterasyon süresince tüm aşamalar döngüsel tekrarlanıp tamamlandığında veya kullanılabilir tüm hesaplama kaynağı tüketilmiş olduğunda algoritmanın elde edebileceği en iyi çözüme ulaşılmış olur.



Şekil 3.1. MLSHADE-SPA iterasyon başına düşen FEs paylaşımı

MLSHADE-SPA çalışma mantığı ve işleyişi aşamalar halinde özet olarak bu bölümde açıklanmakla birlikte algoritma hakkında ayrıntılı bilgi için (Hadi ve ark., 2019) incelenmelidir.

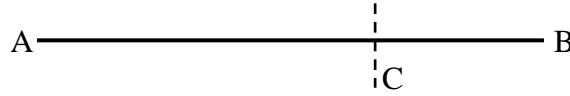
3.3. Altın Oran

Leonardo Fibonacci, orta çağın en ünlü matematikçisi olup birbiri arasında ardışık ilişki ve olağanüstü bir oran bulunduğunu iddia ettiği sayıların kaşifidir. Kainattaki muhteşem düzenle birebir örtüşen bu keşfi sebebiyle, altın orana da adının ilk iki harfi olan F_i (Φ) sayısı da denmektedir. Çünkü altın oran kaynağını fibonacci dizisinden almaktadır: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584,

...şeklinde ilerleyen dizide on üçüncü sayıdan itibaren bu sayıların bir öncekine bölünmesiyle ortaya çıkan sonuç sabitlenmekte olup bu sonuç 1.618' dir.

Altın oran (Golden Ratio, GR), fraktal özelliklere sahip irrasyonel bir sayıdır ve 1.61803398874989 değerine eşittir ve ters altın oran (Inverse Golden Ratio, IGR) ise 0.6180339887'dir. GR ve IGR virgülden sonra aynı rakamlara sahiptir.

Bir AB bölümü Şekil 3.2'de gösterildiği gibi herhangi bir C noktasından bölündüğünde mevcut parçalar arasında $AB / AC = AC / CB$ şeklinde sıra dışı bir oran varsa buna altın oran adı verilmektedir.



Şekil 3.2. Bir segmentin bölünmesi

Eğer AB bölümü x boyutunda ve uzun kısım AC , 1 e normalize edilirse CB bölümü de $x-1$ olacaktır. Bu durumda altın oran kuralı $AB / AC = AC / CB$ Denklem 3.1'deki gibi olacaktır. Bu denklem çözüldüğünde altın oran (Golden Ratio) ve ters altın oran (Inverse Golden Ratio) olarak anılan katsayılar bulunmuş olur.

$$x : 1 = 1 : (x - 1) \quad (3.1)$$

$$x^2 - x - 1 = 0 \quad (3.2)$$

$$x = \frac{1+\sqrt{5}}{2} = \varphi \approx 1.61803398874989 \quad (3.3)$$

$$x - 1 \approx 0.6180339887 \quad (3.4)$$

Altın oran, kristallerin geometrisi, bitkilerde yaprak yerleşimi, insan vücudundaki bölümlerinin oranları, gezegen yörüngeleri gibi atomikten astronomik boyuta kadar değişen uzunluk ölçeklerinde meydana gelen birçok doğa olayında rastlanılan özel bir oranı tanımlamaktadır. Gezegenlerin bir yıldızın etrafındaki yörüngelerini incelendiğinde, fibonacci sayılarının ve altın oranın ilginç bir tezahürü güneş sisteminde fark edilmektedir. Gezegenlerin Güneş'e olan ortalama uzaklıklarının altın orana göre yaklaşık olarak birbiriyle ilişkili olduğu kaydedilmiştir (Marples ve Williams, 2022). Her bitkinin yaprak dizilimi, yaprakların birbirinin üstüne gelmeyecek dolayısıyla güneşi en iyi alabilecek matematiksel bir düzendedir. Bitkinin en fazla sayıda tomurcuk yetiştirme yeteneğinin, dönüş açısı altın oran olduğunda optimize edildiği gösterilmiştir (Bergeron

ve Reutenauer, 2019). Leonardo da Vinci, insan vücudunun matematiksel oranlarını resimledi ve insanın vücudunda 1.618 oranlarına dayalı olarak ortalama altın oranlar olduğunu gösterdi (Dabbour, 2012). Bunun yanısıra günümüzde insan vücudu dış görünüşünden ziyade kalp ve beyin yaratılışında da altın oranın bulunduğu alanlar olduğu bilinmektedir (Tamargo ve Pindrik, 2019; Marples ve Williams, 2022). Örneğin Henein ve ark. kalp tasarımında üç ayrı altın oran kullanımı tespit etmişler ve sağlıklı kalpte aynı ölçümler yapıldığında bu değerlerde sapma olduğuna değinmişlerdir (Henein ve ark., 2011). Sonuç olarak altın oran, kâinatın var oluşundan beri var olan matematiksel harika bir simetri ya da tasarım özelliğidir. Doğa, biyoloji, hesaplamalı bilimler, borsa, matematik, mühendislik, endüstriyel tasarım, mimarlık, sanat vb. birçok alanda altın oran kullanımı mevcuttur (Livio, 2008; Dabbour, 2012; Shekhawat, 2015; Ciucurel ve ark., 2018; Iosa ve ark., 2018). Altın oran hakkında daha detaylı bilgi için (Iosa ve ark., 2018; Lüttge ve Souza, 2019; Marples ve Williams, 2022) incelenebilir.

Altın oranının birçok alandaki mükemmel tasarım ve işleyişteki tezahürü, optimizasyon alanında algoritmalar bünyesinde altın oranın katsayı, arama tekniği veya seçim yöntemi olarak kullanılmasına yol açmıştır. Örneğin parçacık sürü optimizasyonunun tasarımında etkili bir yere sahip hız denkleminde kullanılan c_1 ve c_2 parametreleri sırasıyla ters altın oran (IGR) ve altın oran (GR) ile değiştirilerek Altın oran parçacık sürü optimizasyonu önerilmiştir (Manikantan ve ark., 2012). Altın bölüm araması (Golden Section Search, GSS), “Kötüyü azalt, iyiyi koru” ilkesine dayanan, arama kapsamını kademeli olarak daraltmak için simetrik ve eşit daraltma yöntemlerini benimseyen yerel bir arama yöntemidir. Çözüm uzayı her seferinde düşürülerek iki nokta arasındaki en uygun konum hızla bulunabilmektedir. Bu yöntemde altın oran, arama uzayını bölmek amacıyla bir sonraki adımın yeni alt ve üst sınırlarını belirlemek için kullanılmaktadır (Neri ve Tirronen, 2009; Koupaei ve ark., 2016; Aurasopon ve Khamsen, 2019). Pu ve ark. genetik algoritmanın önemli yapı taşları olan çaprazlama ve mutasyon operatörünün en iyi uyarlanabilir noktasını tespit etmek ve yerel optimizasyonu gerçekleştirebilmek amacıyla altın oran kullanmışlardır (Pu ve ark., 2018). Altın oranın seçim yöntemi olarak kullanılmasında, popülasyon her biri benzer niteliklere sahip belirli sayıda birey içerecek şekilde altın oranla alt gruplara ayrılır. Her bireyin seçilme olasılığı, ait olduğu gruba bağlıdır. Bu yaklaşım, elitizm ve çeşitlilik arasında daha iyi bir denge sağlamaktadır (Cuevas ve ark., 2018).

Altın oran değişken eşitsizliklerinin çözümü içinde kullanılmaktadır (Hieu ve ark., 2019; Malitsky, 2020). Örneğin Hieu ve ark. değişken eşitsizliklerin çözümü için biri

önceden seçilmiş azalan bir dizi adım boyutu kullanan diğeri ise adım boyutu her yinelemede basit bir hesaplamayla güncellenen iki altın oran algoritması önermişlerdir (Hieu ve ark., 2019).

3.4. Kaotik Haritalar

Kaos, matematiksel olarak basit deterministik sistemler tarafından üretilen "rastgelelik" olarak tanımlanır. Bu sistemler başlangıç parametrelerine karşı yüksek hassasiyet gösterir. Ön parametrelerdeki küçük kaotik iyileştirmeler, sistemin çıktısında ve performansında önemli değişikliklere neden olur. Kaotik sistemlerin başlangıç koşullarına olan hassasiyeti nedeniyle rastgele ve öngörülemez gibi görünse de, düzenlilik özelliği de vardır, bu nedenle kaos, düzensizlikten düzenin doğmasını sağlayabilir (Sheikholeslami ve Kaveh, 2013). Bu sistemlerin bir diğer özelliği de ergodik (ergodicity) olmalarıdır. Kaosun ergodik olma özelliği, kaotik değişkenlerin kendi yasalarına göre belirli bir aralık içinde tüm durumları tekrar etmeden geçmesini sağlayabilir. Başka bir deyişle belirli bir aralıkta tekrara düşmemeyi yani aynı sayıdan tekrar üretilmemesini garanti eder. Bu nedenle, birey üretiminde çeşitliliğin azalmasından veya yerel minimumlara takılıp kalmaktan kurtarmak için kaotik haritalar kullanılabilir (Mirjalili ve Gandomi, 2017).

Metasezgisel algoritmaların performansını etkileyen faktörlerden biri de arama uzayındaki olası çözümleri temsil eden bireylerin oluşturduğu başlangıç popülasyonudur. Çünkü algoritmalar, arama uzayında en iyi çözüme ulaşmak için yapılan işlemler aracılığıyla başlangıç popülasyonunda bulunan bu bireyleri geliştirmektedirler. Kaotik haritalar uzun periyotlu rastgele sayılar üretmede sıklıkla kullanılan stokastik, deterministik ve doğrusal olmayan bir stratejilerdir. Kaotik haritalar, oluşturduğu sayı dizilerinde sayıların tekrarlanmaması, belirli bir aralıkta sıkışmayıp geniş bir spektruma yayılmış olması ve bunun yanında dizi üretim ve saklama maliyetinin düşük olması gibi önemli avantajlara sahiptir. Bu nedenle kaos, metasezgisel algoritmalar bünyesinde rastgele sayı üretici olarak kullanılır. Bu üreteçten elde edilen sayı, başlangıç popülasyon birey üretimi için kullanılabilir gibi olası çözüm iyileştirme sürecinde gerçekleştirilen işlemlerde bir katsayı olarak da kullanılabilir (Sheikholeslami ve Kaveh, 2013; Wang ve Zhong, 2015; Kuyu ve Vatansever, 2018; Koçer ve ark., 2023). Kaos kullanımı global optimuma yakınsama hızı ve çözüm uzayını arama performansı açısından optimizasyon sürecine katkı sağlayabilmektedir.

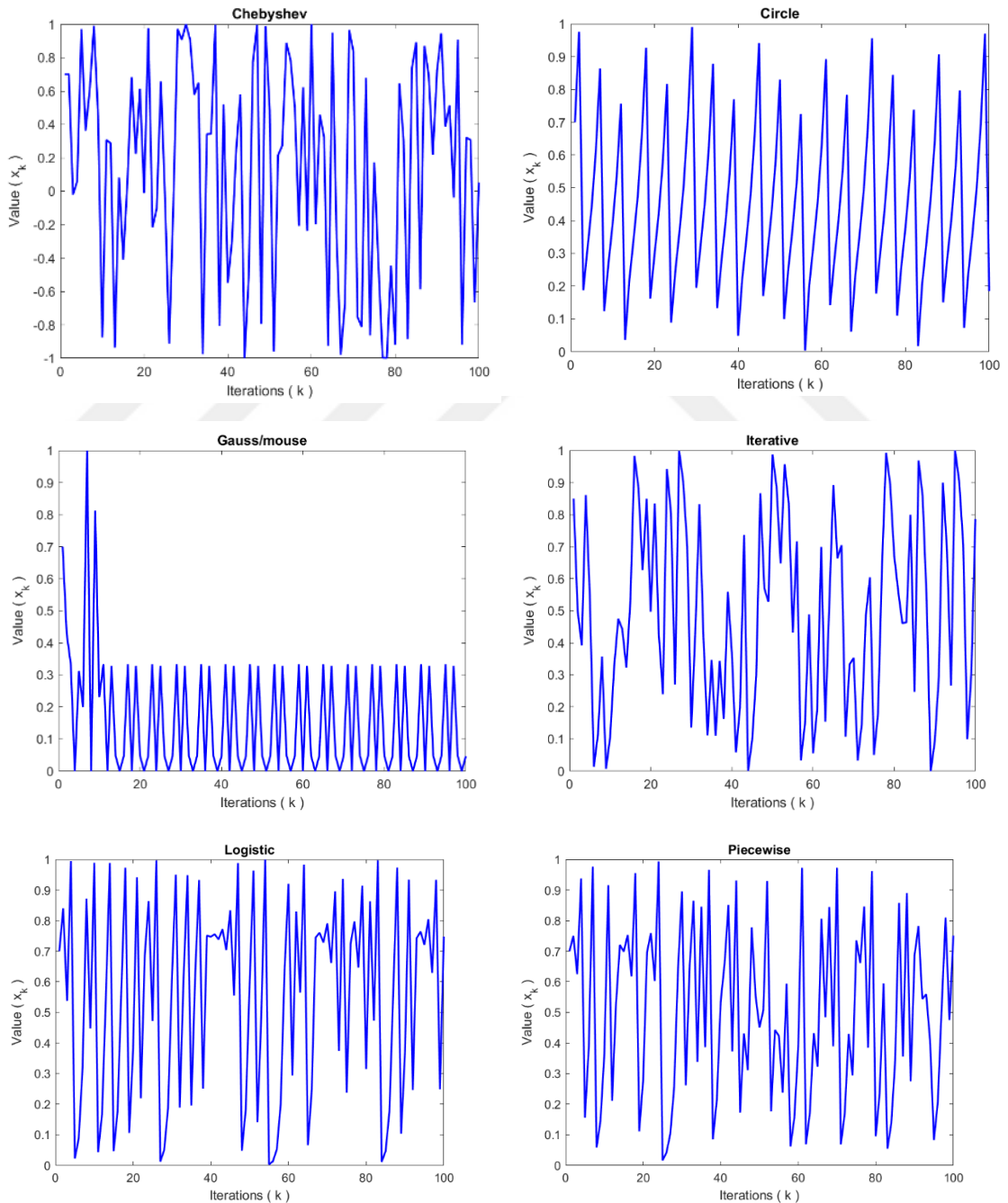
Belirli bir sistematığı olan, tüm spektrumunu kapsayabilen ve tekrara düşmeyen kaos yaklaşımı, algoritmaların performans kararlılığını sağlayacak keşif ve sömürü dengesini sağlamak için kullanışlı bir yöntem olarak tercih edilmektedir (Kuyu ve Vatansever, 2018). Dolayısıyla genetik algoritma, parçacık sürü optimizasyonu, diferansiyel gelişim algoritması, ateşböceği algoritması, gri kurt algoritması, kelebek algoritması, çekirge algoritması, meyve sineği algoritması vb. gibi literatürde popüler olan birçok metasezgisel algoritmanın kaotik versiyonları da geliştirilmiştir(Koçer ve ark., 2023).

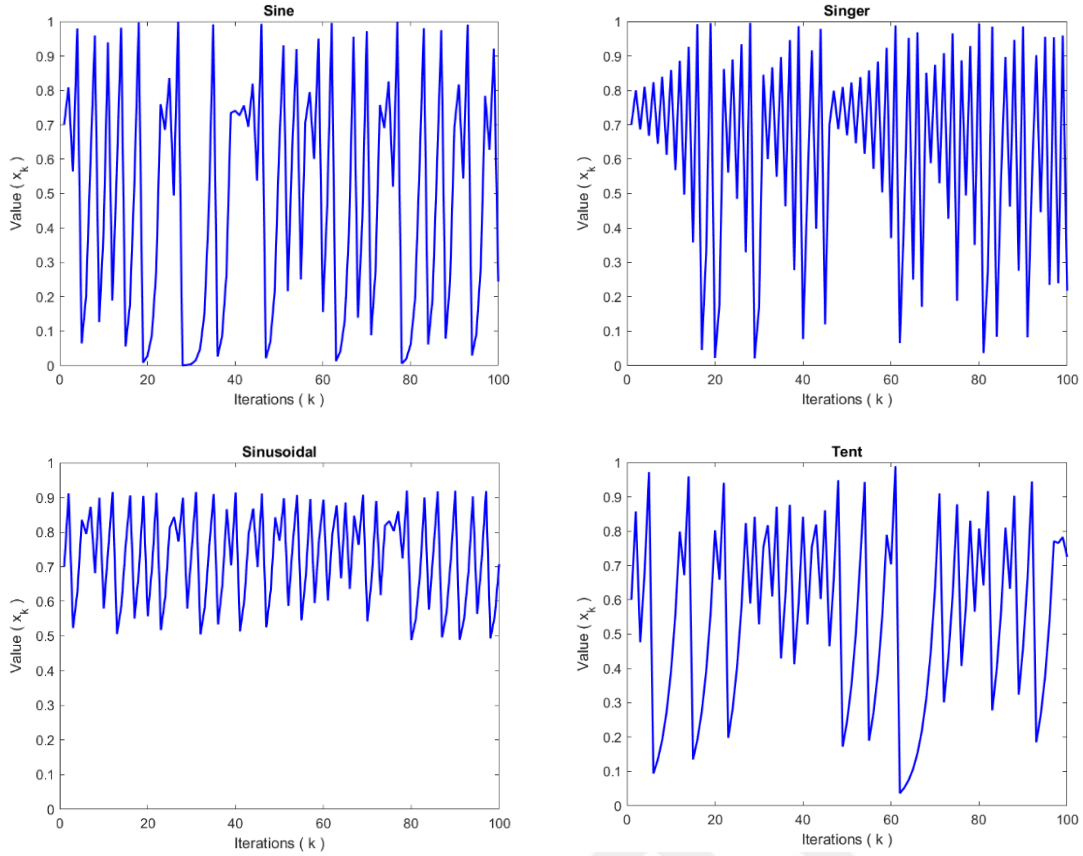
Çizelge 3.2. Kaotik Haritalar(Türkoğlu, 2022)

No.	Kaotik Harita Adı	Tanımı
H1	Chebysev	$x_{k+1} = \cos (k \cos^{-1} (x_k))$
H2	Çember(Circle)	$x_{k+1} = x_k + b - \left(\frac{a}{2\pi}\right)\sin (2\pi k)\text{mod}(1)$
H3	Gauss(Gaussian)	$x_{k+1} \begin{cases} 0 & x_k = 0, \\ \frac{1}{x_k \text{mod}(1)} & \text{otherwise, } x \geq 0 \end{cases}$
H4	İteratif(Iterative)	$x_{k+1} = \sin \left(\frac{a\pi}{x_k}\right), a \in (0, 1)$
H5	Logistik(Logistic)	$x_{k+1} = ax_k(1 - x_k)$
H6	Parçalı(Piecewise)	$x_{k+1} = \begin{cases} \frac{x_k}{p}, & 0 \leq x_k < P \\ \frac{x_k - P}{0.5 - P}, & P \leq x_k < \frac{1}{2} \\ \frac{1 - P - x_k}{0.5 - P}, & \frac{1}{2} \leq x_k < 1 - P \\ \frac{1 - x_k}{P}, & 1 - P \leq x_k < 1 \end{cases}$
H7	Sinüs(Sine)	$x_{k+1} = \frac{a}{4} \sin (\pi x_k), 0 \leq a < 4$
H8	Şarkıcı(Singer)	$x_{k+1} = \mu(7.86x_k - 23.31x_k^2 + 28.75x_k^3 - 13.302875x_k^4)$
H9	Sinüzoidal(Sinusoidal)	$x_{k+1} = ax_k^2 \sin (\pi x_k)$
H10	Çadır(Tent)	$x_{k+1} = \begin{cases} \frac{x_k}{0.7}, & x_k < 0.7 \\ \frac{10}{3}(1 - x_k), & x_k \geq 0.7 \end{cases}$

Literatürde birçok farklı kaotik harita türü olmasına rağmen çalışmalarda sıklıkla tercih edilen on kaotik harita bulunmaktadır. Bunlar formülleri Çizelge 3.2’de listelenen Chebyshev (Tang ve ark., 2018), Circle (Jordehi, 2015), Gaussian (Kaur ve Arora, 2018),

Iterative (Rather ve Bala, 2020), Logistic (Demir ve ark., 2020), Piecewise (Wang ve ark., 2014), Sine (Saremi ve ark., 2014), Singer (Gandomi ve ark., 2013b), Sinusoidal (Talatahari ve ark., 2012) ve Tent (Gandomi ve ark., 2013a) kaotik haritalarıdır. Bu kaotik harita denklemlerinden bir başlangıç parametresi ile istenen büyüklükte kaotik diziler oluşturabilir ve gerektiği yerde rastgele sayılar kullanmak yerine, oluşturulan kaotik diziden bir sonraki sayı çekilir ve kullanılır. Bahsi geçen 10 harita kullanılarak oluşturulan 100 adet sayı spektrumları Şekil 3.3'te gösterilmektedir.





Şekil 3.3. Kaotik harita spektrumları (Türkoğlu, 2022)

3.5. Test Fonksiyon Kümeleri

2008 yılından bu yana, geleneksel olarak CEC özel oturumlarında LSGO alanında yarışmalar düzenlenmektedir. Bu yarışmalarda araştırmacıları adil bir şekilde değerlendirmek için 2008, 2010 ve 2013 yıllarında ortak kurallar ve test fonksiyon kümeleri sunulmuştur.

Tez kapsamında gerçekleştirilen deneysel çalışmalarda CEC2010, CEC2013 test fonksiyon kümeleri ile gerçek dünya problemi olarak 2015 Büyük Veri Optimizasyonu yarışması için modellenen Big-Opt test fonksiyon kümesi kullanılmıştır.

3.5.1. CEC2010

CEC2010 test fonksiyon kümesi Tang ve ark. tarafından LSGO alanında kullanılmak üzere önerilmiştir. Gerçek dünyadaki optimizasyon problemleri, kendi içlerinde güçlü bağımlılıklar olan fakat büyük olasılıkla diğer gruplarla arasında çok az

etkileşim olan farklı parametre gruplarından oluşacaktır. Bu durumun test fonksiyonlarına yansıtılması gerekir (Tang ve ark., 2009).

Fonksiyonlardaki ayrılabilirlik seviyesi bize problemlerin zorluğunun bir ölçüsünü sağlar. Çeşitli türlerde kısmen ayrılabilir problemler mevcut olmakla birlikte bir diğer grup olan ayrılabilir problemler genellikle en kolay problemler, tamamen ayrılamayan problemler ise genellikle en zor problemler olarak değerlendirilir (Griewank ve Toint, 1982; Colson ve Toint, 2005). CEC2010 test fonksiyon kümesi farklı ayrılabilirlik seviyelerine sahip minimizasyon problemleri içermektedir. Bu test fonksiyon kümesinde F1, F2, F3 fonksiyonları ayrılabilir, F19 ve F20 ise ayrılamaz fonksiyonlardır. Bünyesinde farklı ayrılabilirlik seviyelerine sahip fonksiyonlar içeren test fonksiyon kümesi hakkında daha detaylı bilgiye (Tang ve ark., 2009)'dan ulaşılabilir.

Çizelge 3.3. CEC2010 LSGO test fonksiyon kümesi fonksiyonları

No	İsim	Aralık
F1	Kaydırılmış Elliptic Fonksiyonu	[-100, 100]
F2	Kaydırılmış Rastrigin Fonksiyonu	[-5, 5]
F3	Kaydırılmış Ackley Fonksiyonu	[-32, 32]
F4	Tek grup kaydırılmış ve m-döndürülmüş Elliptic Fonksiyonu	[-100, 100]
F5	Tek grup kaydırılmış ve m-döndürülmüş Rastrigin Fonksiyonu	[-5, 5]
F6	Tek grup kaydırılmış ve m-döndürülmüş Ackley Fonksiyonu	[-32, 32]
F7	Tek grup kaydırılmış m-boyutlu Schwefel 1.2 Fonksiyonu	[-100, 100]
F8	Tek grup kaydırılmış m-boyutlu Rosenbrock Fonksiyonu	[-100, 100]
F9	D/2m grup kaydırılmış ve m-döndürülmüş Elliptic Fonksiyonu	[-100, 100]
F10	D/2m grup kaydırılmış ve m-döndürülmüş Rastrigin Fonksiyonu	[-5, 5]
F11	D/2m grup kaydırılmış ve m-döndürülmüş Ackley Fonksiyonu	[-32, 32]
F12	D/2m grup kaydırılmış m-boyutlu Schwefel 1.2 Fonksiyonu	[-100, 100]
F13	D/2m grup kaydırılmış m-boyutlu Rosenbrock Fonksiyonu	[-100, 100]
F14	D/m grup kaydırılmış ve m-döndürülmüş Elliptic Fonksiyonu	[-100, 100]
F15	D/m grup kaydırılmış ve m-döndürülmüş Rastrigin Fonksiyonu	[-5, 5]
F16	D/m grup kaydırılmış ve m-döndürülmüş Ackley Fonksiyonu	[-32, 32]
F17	D/m grup kaydırılmış m-boyutlu Schwefel 1.2 Fonksiyonu	[-100, 100]
F18	D/m grup kaydırılmış m-boyutlu Rosenbrock Fonksiyonu	[-100, 100]
F19	Kaydırılmış Schwefel 1.2 Fonksiyonu	[-100, 100]
F20	Kaydırılmış Rosenbrock Fonksiyonu	[-100, 100]

Bu test fonksiyon kümesi 1000 boyutlu 20 fonksiyondan oluşmaktadır. Çizelge 3.3'te listelenen tüm fonksiyonlar minimizasyon problemleridir ve tüm problemler için optimum fonksiyon değerleri 0'dır. Çizelge 3.3'te fonksiyon isminde geçen D problem

boyutunu temsil etmekte olup 1000 değerine eşittir. Fonksiyon isminde geçen m değeri ise, her gruptaki değişken sayısını kontrol etmek ve dolayısıyla ayrılabilirlik derecesini tanımlamak için kullanılan bir parametredir. Varsayılan olarak $m = 50$ olmakla birlikte kullanıcılar bu parametreyi kendi amaçları için rahatlıkla kontrol edebilirler.

3.5.2. CEC2013

LSGO alanındaki ilerlemeler neticesinde daha geniş bir gerçek dünya problemi yelpazesinin özelliklerini temsil etmek ve ayrıştırma için bazı yeni zorluklar ortaya koymak için CEC2013 test fonksiyon kümesi geliştirilmiştir.

Çizelge 3.4. CEC2013 LSGO test fonksiyon kümesi fonksiyonları

No	Fonksiyon Adı	Aralık	Özellikler	Grup
F1	Elliptic Fonksiyonu	$[-100, 100]$	Tek Modlu, Kaydırılmış	Tamamen ayrılabilir fonksiyonlar
F2	Rastrigin Fonksiyonu	$[-5, 5]$	Çok Modlu, Kaydırılmış	
F3	Ackley Fonksiyonu	$[-32, 32]$	Çok Modlu, Kaydırılmış	
F4	Elliptic Fonksiyonu	$[-100, 100]$	Tek Modlu, Kaydırılmış	Ayrılabilir bir alt bileşeni bulunan fonksiyonlar
F5	Rastrigin Fonksiyonu	$[-5, 5]$	Çok Modlu, Kaydırılmış	
F6	Ackley Fonksiyonu	$[-32, 32]$	Çok Modlu, Kaydırılmış	
F7	Schwefel 1.2 Fonksiyonu	$[-100, 100]$	Çok Modlu, Kaydırılmış	Ayrılabilir bir alt bileşeni olmayan fonksiyonlar
F8	Elliptic Fonksiyonu	$[-100, 100]$	Tek Modlu, Kaydırılmış	
F9	Rastrigin Fonksiyonu	$[-5, 5]$	Çok Modlu, Kaydırılmış	
F10	Ackley Fonksiyonu	$[-32, 32]$	Çok Modlu, Kaydırılmış	Ayrılabilir bir alt bileşeni olmayan fonksiyonlar
F11	Schwefel 1.2 Fonksiyonu	$[-100, 100]$	Tek Modlu, Kaydırılmış	
F12	Rosenbrock Fonksiyonu	$[-100, 100]$	Çok Modlu, Kaydırılmış	
F13	Uyumlu Örtüşen Alt Bileşenlere sahip Schwefel Fonksiyonu	$[-100, 100]$	Tek Modlu, Kaydırılmış	Örtüşen Fonksiyonlar
F14	Uyumsuz Örtüşen Alt Bileşenlere sahip Schwefel Fonksiyonu	$[-100, 100]$	Tek Modlu, Kaydırılmış	
F15	Schwefel 1.2 Fonksiyonu	$[-100, 100]$	Tek Modlu, Kaydırılmış	Ayrılamaz fonksiyonlar

Büyük ölçekli küresel optimizasyon algoritmalarını test etmek ve karşılaştırmak için uygun bir değerlendirme platformu sağlayan CEC2013 test fonksiyon kümesi, alana özgü tasarlanmış en güncel test fonksiyon kümesi olması nedeniyle araştırmacılar tarafından çokça tercih edilmektedir. Mümkün olduğunca gerçek dünya problemlerine

benzer olması için CEC2010 test fonksiyon kümesine tekdüze olmayan alt bileşen boyutları, alt bileşenlerin katkısında dengesizlik, örtüşen alt bileşenlere sahip fonksiyonlar, temel fonksiyonlara yeni dönüşümler gibi bazı karakteristik özellikler eklenerek CEC2013 test fonksiyon kümesi oluşturulmuştur. Bu test fonksiyon kümesi, farklı ayrılabilirlik derecelerine sahip 15 sürekli optimizasyon fonksiyonu içermektedir. Test fonksiyon kümesindeki tüm fonksiyonlar minimizasyon problemleridir ve optimum fonksiyon değerleri sıfırdır. Boyutun 905 olduğu F13 ve F14 fonksiyonları hariç diğer tüm fonksiyonların boyutları 1000'dir. Test fonksiyon kümesi ve bünyesindeki fonksiyonlar hakkında daha detaylı bilgiye (Li ve ark., 2013)dan ulaşılabilir.

3.5.3. Big-Opt

Elektroensefalografi (EEG), beyindeki elektriksel aktiviteyi ölçen ve kafa derisine yerleştirilen elektrotlar aracılığıyla beyin dalga modellerini kaydeden bir testtir. EEG sinyalleri, nörolojik bozuklukların ve sinir sistemindeki belirli durumların saptanmasında önemli bir rol oynar. EEG gerçek sinyallerin yanısıra artefakt olarak adlandırılan yutma, öksürme, göz ve kalp hareketleri gibi beyin dışı elektriksel aktivitelerde elektrotlar aracılığıyla kaydedilebilmektedir. EEG sinyallerindeki gürültü, inceleme ve yorumlama süresini arttırır ve nörolojik bozuklukların doğru teşhis edilmesini engelleyebilir (Yıldırım ve ark., 2021). Sinyaller, artefaktların ortadan kaldırılıp olması gereken gerçek beyin sinyallerinin yeniden oluşturulması için iyi bilinen bir kör kaynak ayırma (Blind Source Separation, BSS) tekniği olan Bağımsız Bileşen Analizinden geçirilir (Goh ve ark., 2016). Goh ve ark., 2015 yılında Büyük Verinin Optimizasyonu Yarışması (Optimization of Big Data 2015 Competition) için NP-Hard Büyük Veri optimizasyon problemini (Big-Opt) modellemişlerdir.

Bu problem, çok sayıda karar değişkeni içermesi, doğrusal olmaması ve EEG sinyallerinin gerçek zamanlı olarak işlenmesini gerektirmesi nedeniyle karmaşık ve zorlu bir büyük veri problemi olarak kategorize edilmektedir (Goh ve ark., 2015; Goh ve ark., 2016). Big-Opt probleminin temel amacı cihazdan ölçülen her bir saniye için artefaktları en aza indirerek doğru EEG sinyallerine ulaşmaktır. Bu nedenle problem, elde edilen EEG sinyalini iki bileşene ayırmaya çalışmaktadır, bu bileşenlerden biri orijinal sinyale en çok benzeyen sinyali elde etmeye odaklanırken, diğer bileşen mümkün olduğu kadar artefaktları tespit edip silmeye çalışır.

Goh ve ark., 4, 12 ve 19 tane birbirine bağılı zaman serisi üzerinden 256 adet bir saniyelik ölçüm örneği olarak sonuçlarını kaydetmişlerdir. Böylece EEG sinyal ölçümlerinden sentetik veriler oluşturulmuş ve bu veriler altı farklı problem örneği olarak ayrılmışlardır. Yerel sinyal sayısı, artefakt kaynakları ve gürültü seviyesi açısından farklı zorluk ve boyutlara sahip altı veri seti Çizelge 3.5'te gösterildiği gibidir. Birbirine bağılı zaman serilerinin sayısına atıfta bulunan D4, D12 ve D19 isimleri, kaydedilen bir saniyelik ölçümlerinin varsayılan durumu için kullanılırken D4N, D12N ve D19N adları, varsayılan duruma belirli bir miktarda artefakt eklenmiş hali için kullanılır.

Problemdeki değişken sayısı, zaman serisi sayısı ile EEG sinyallerinin uzunluğu çarpılarak tanımlanır. Her bir zaman serisi için uzunluk 256 olarak belirlenmiştir yani veri setinde 4 zaman serisi varsa bu $4 \times 256 = 1024$ değişkenli bir optimizasyon problemi olduğu anlamına gelmektedir. Bu nedenle D4, D12 ve D19 problemlerinde sırasıyla 1024, 3072 ve 4864 değişken vardır. Bu test fonksiyon kümesi, gerçek dünya problemlerinden oluşmakta ve LSGO alanında nadir rastlanan gürültülü ve gürültüsüz problemler içermektedir (Molina ve ark., 2019).

N , birbirine bağılı zaman serilerinin sayısını, M , elde edilen örnek sayısını, $N \times M$ ise ölçüm sonuçlarını saklayan X matrisinin boyutunu temsil etmektedir. A , doğrusal bir dönüşüm matrisi ve S , N satır ve N sütun içeren bir sinyal matrisi olmak üzere X matrisi aşağıdaki gibi tanımlanabilir:

$$X = A * S \quad (3.5)$$

Problem, EEG sinyalini temsil eden S matrisini, her biri S ile aynı sayıda boyuta sahip olan S_1 ve S_2 olmak üzere iki matrise bölmektir (Goh ve ark., 2015). S_1 matrisi, S matrisinin gürültüden ayrılmış kısmını temsil ederken S_2 matrisi, S matrisinin gürültü ile ilişkili kısmını temsil eder (Arslan ve Aslan, 2022).

$$S = S_1 + S_2 \quad (3.6)$$

$$X = A * S_1 + A * S_2 \quad (3.7)$$

Goh ve ark. X , A ve S_1 matrislerini kullanarak Denklem 3.8 ile hesaplanan Pearson korelasyon katsayılarının S matrisini en uygun şekilde S_1 ve S_2 matrislerine bölmek için

kullanılabileceğini belirlemişlerdir. Denklem 3.8.'de C , X ve AxS_1 arasındaki Pearson korelasyon katsayısını temsil ederken *covar* kovaryans matrisini ve σ ise varyansı temsil eder. Amaç, köşegen dışı öğeleri sıfıra indirirken C 'nin köşegen öğelerini maksimize etmektir. S ve S_1 arasındaki fark mümkün olduğu kadar küçük olmalıdır; başka bir deyişle, S_1 , S ile mümkün olduğu kadar aynı olmalıdır (Goh ve ark., 2015).

$$C = \frac{\text{covar}(X, AxS_1)}{\sigma(X) \times \sigma(AxS_1)} \quad (3.8)$$

Sinyal segmentinin kalitesini belirlemeyi amaçlayan amaç fonksiyonları aşağıdaki gibi tanımlanır (Goh ve ark., 2015; Goh ve ark., 2016):

$$\min f_1 = \frac{1}{N^2 - N} \sum_i \sum_{j \neq i} C_{ij}^2 + \frac{1}{N} \sum_i (1 - C_{ii})^2 \quad (3.9)$$

$$\min f_2 = \frac{1}{N \times M} \sum_i \sum_j (S_{ij} - S_{1ij})^2 \quad (3.10)$$

$$\min F = f_1 + f_2 \quad (3.11)$$

Çizelge 3.5. Big-Opt test fonksiyon kümesi fonksiyonları

Örnek Veri Seti	Artefakt Sayısı	Kaynak Sayısı	Veri Uzunluğu	Boyut	Gürültü
D4	2	4	256	1024	Hayır
D4N	2	4	256	1024	Evet
D12	6	12	256	3072	Hayır
D12N	6	12	256	3072	Evet
D19	6	19	256	4864	Hayır
D19N	6	19	256	4864	Evet

Bu kıyaslamamanın çok amaçlı versiyonu ve tek amaçlı versiyonu vardır. Çok amaçlı versiyon, C matrisinin köşegen dışı ve köşegen elemanlarının minimizasyon ve maksimizasyon özellikleri ile ilgiliyken, tek amaçlı versiyon, S_1 matrisinin S matrisine ne kadar benzediğini değerlendirmekle ilgilidir (Goh ve ark., 2014; Goh ve ark., 2015).

4. ÖNERİLEN YENİ YEREL ARAMA YÖNTEMLERİ VE ALGORİTMALAR

Bu tez kapsamında LSGO problemlerinin çözümü için problemi bir bütün olarak ele alan ayrıştırma esaslı olmayan çözüm yaklaşımları üzerinde çalışılmıştır. Standart evrimsel algoritmaların ek düzenleme ve tekniklerle geliştirilmesi amacıyla gerçekleştirilen çalışmalar sonucu Değiştirilmiş Yapay Alg Algoritması (MAAA) önerilmiştir. En etkili çözüm yaklaşımlarından biri olan memetik algoritmalar bünyesinde kullanılmak üzere dinamik adım boyutuna sahip altın oran rehberli yerel arama (GRGLS) ve kaotik altın oran rehberli yerel arama (CGRGLS) olmak üzere iki yeni yerel arama yöntemi geliştirilmiştir. Bu yeni yerel arama yöntemlerinin MSHADE-SPA adı verilen çerçeve algoritmanın yerel arama kısmında kullanılmasıyla oluşan IMLSHADE-SPA ve CIMHADE_SPA adında iki memetik çerçeve algoritma önerilmiştir. Bu bölümde tez kapsamında önerilen iki yerel arama yöntemi ve üç algoritma hakkında ayrıntılı bilgi verilmektedir.

4.1. Değiştirilmiş Yapay Alg Algoritması (MAAA)

Optimizasyon teknolojisi, yönetim ve mühendislik problemlerinde karar verme süreçlerini hızlandırmak ve karar verme kalitesini artırmak için kullanılmaktadır. Hızla gelişen teknoloji gerçek dünya problemlerini daha büyük ve karmaşık hale getirmiştir. LSGO problemlerini çözmek için önerilen birçok optimizasyon yöntemi, arama uzayının boyutluluğu arttıkça performanslarının hızla bozulması anlamına gelen “boyutsallığın laneti”nden muzdariptir. Bu nedenle bu durumla başa çıkabilecek alana özel geliştirilmiş daha verimli ve sağlam algoritmalar geliştirilmesine veya var olan algoritmaların farklı teknik ve güncelleştirmeler yoluyla LSGO alanına uyumlu hale getirilmesine ihtiyaç vardır.

DE Algoritmasının LSGO alanında en etkili algoritmalarından biri olup geliştirilen bir çok etkili teknik, algoritma veya çerçevede de temel metasezgisel algoritma olarak kullanılması (Maučec ve Brest, 2019) ve çok başarılı sonuçlar elde etmesi nedeniyle algoritma işleyişi analiz edilmiştir. DE mutasyon stratejilerinden ilham alınarak AAA orijinal hali üzerinde değişiklik yapılmış ve büyük boyutla daha iyi baş edebilecek bir versiyonu geliştirilmeye çalışılmıştır. Çalışmalar neticesinde önerilen yeni versiyona Değiştirilmiş Yapay Alg Algoritması (Modified Artificial Algae Algorithm, MAAA) adı verilmiştir.

Mutasyon, bazı parametrelerin rastgele deęişimi olarak ele alınır. Mutasyon fazındaki deęişiklikler sayesinde kromozomu temsil eden çözüm noktası çözüm uzayında hareket ettirilir. Literatürde mutasyon aşamasında, birçok DE deneme vektörü oluşturma stratejisi önerilmiş ve iyi bilinen mutasyon stratejileri (Gong ve ark., 2011; Banitalebi ve ark., 2016)'de listelenmiştir. Bunlar arasından bu çalışma kapsamında ilham alınan mutasyon stratejisi aşağıdaki gibi ifade edilmektedir:

$$V_{g+1} = x_g + F(x_{bg} - x_g) + F(x_{r1g} - x_{r2g}) + F(x_{r3g} - x_{r4g}) \quad (4.1)$$

Bu eşitlikte V, hesaplanan donör vektörü, g şimdiki nesli, g + 1 bir sonraki nesli, b en iyi bireyi ifade ederken formülasyonda bulunan dört farklı parametre vektörü r1, r2, r3, r4 mevcut popülasyondan seçilen rastgele örnekleri temsil etmektedir. Bu vektörlerin birbirlerinden farkı bir F skaler sayısı ile ölçeklenir ve ölçeklenen farklar genel toplama ilave edilerek sonuca ulaşılır.

LSGO alanında başarılı sonuçlar elde edebilmek için keşif ve sömürü gücü yüksek bir algoritma kullanmak gerekmektedir. AAA'nın evrim ve adaptasyon süreçleri sömürü mekanizmasını desteklerken helisel hareket süreci de ilave olarak hem keşif hem de sömürü mekanizmalarına katkıda bulunmaktadır. Diğer bir deyişle, AAA'nın keşif ve sömürü süreci, alg kolonilerinin modifikasyonu yoluyla oluşan helisel bir hareketle sağlanmaktadır. Bu nedenle AAA arama kabiliyetini artırmak için helisel hareket sürecinde DE mutasyon stratejisinden ilham alınarak deęişiklik yapılmasına karar verilmiştir.

Helisel hareket, her alg kolonisini güncelleyen bir süreçtir. Bu süreçte turnuva seçimi ile koloni seçilir ve yine rastgele seçilen ve k, l ve m olarak temsil edilen her alg kolonisinden sadece üç alg hücresi aşağıdaki formüllerle deęiştirilerek helisel hareket gerçekleştirilmiş olur. Bu formüllerde kullanılan Δ parametresi alg kolonilerinin kesme kuvvetini ve τ parametresi ise alg kolonilerinin sürtünme yüzeyini temsil etmektedir.

Açısal hareket hesaplama denklemleri

$$x_{ik}^{t+1} = x_{ik}^t + (x_{jk}^t - x_{ik}^t)(\Delta - \tau^t(x_i)) \cos \alpha \quad (4.2)$$

$$x_{il}^{t+1} = x_{il}^t + (x_{jl}^t - x_{il}^t)(\Delta - \tau^t(x_i)) \sin \beta \quad (4.3)$$

Doğrusal hareket hesaplama denklemi

$$x_{im}^{t+1} = x_{im}^t + (x_{jm}^t - x_{im}^t)(\Delta - \tau^t(x_i))p \quad (4.4)$$

AAA'daki helisel hareketin hemen ardından aynı alg kolonisinin dördüncü bir alg hücresinde güncelleme yapılmasının algoritmanın keşif kabiliyetini artırmada faydalı olabileceği düşünülmüştür. Bu nedenle Denklem 4.1'deki diferansiyel gelişim mutasyon stratejisinden esinlenerek AAA helisel hareket sürecinde bir değişiklik yapılarak LSGO için geliştirilmiş bir versiyonu önerilmiştir. DE mutasyon sürecinde mevcut popülasyondaki rastgele vektörlerin farklılıklarını kullanırken, MAAA'da Denklem 4.2 ile hesaplanan x_{ik}^{t+1} , Denklem 4.3 ile hesaplanan x_{il}^{t+1} ve Denklem 4.4 ile hesaplanan x_{im}^{t+1} helisel hareket hesaplama sonuçları kullanılmıştır. Bu sonuçların birbirinden farkları popülasyondaki en iyi bireye eklenerek AAA işleyişinde değişen üç hücreye ek olarak aynı alg kolonisinin rastgele seçilen dördüncü bir alg hücresi aşağıdaki formülle değiştirilmiştir.

$$x_{in}^{t+1} = x_{best}^t + (x_{im}^{t+1} - x_{ik}^{t+1}) + (x_{ik}^{t+1} - x_{il}^{t+1}) + (x_{il}^{t+1} - x_{im}^{t+1}) \quad (4.5)$$

Çizelge 3.1'de sözde kodu verilen orijinal AAA tasarımı üzerinde yapılan değişikliğe ait sözde kod Çizelge 4.1'de gösterilmektedir.

Çizelge 4.1. Helisel harekette yapılan değişikliğe ait sözde kod (Kocer ve Uymaz, 2018) (adım: 2 ve 4)

-
- 1 : Turnuva seçimi ile bir koloni seç
 - 2 : Seçilen kolonide rastgele dört alg hücresi (k, l, m ve n) seç
 - 3 : Koloninin seçilen alg hücrelerinden üçünün (k, l ve m) (4.2), (4.3) ve (4.4) denklemleri aracılığıyla güncelle
 - 4 : Bir önceki adımın sonuçlarını kullanan Denklem (4.5) aracılığıyla aynı koloninin dördüncü hücresini (n) güncelle
 - 5 : Hareketin neden olduğu enerjiyi azalt
 - 6 : Oluşan yeni aday çözüm daha iyiyse, pozisyonu güncelle, aksi takdirde enerjiyi azalt
 - 7 : Koloninin enerjisi bitmediyse 1. adıma geç
 - 8 : Koloni daha iyi bir çözüm bulamadıysa, koloninin açlığını artır
-

4.2. Dinamik adım boyutuna sahip altın oran rehberli yerel arama (GRGLS) ve IMLSHADE-SPA

Evrimsel algoritmalar (EA), optimizasyon problemlerinin optimal çözümünü elde etmek için genellikle doğadan ilham alan kurallar ve rastgelelik kombinasyonunu kullanır. EA'lar, keşif ve sömürü olmak üzere iki temel arama aşaması gerçekleştirir. Bir arama alanındaki tamamen yeni noktaları ziyaret etmeye keşif denirken daha önce ziyaret edilen konumların çevresinde iyileştirme yapmak amacıyla bu noktalarda ince ayar yapmaya sömürü denilmektedir. EA'nın başarısı, keşif ve sömürü arasında iyi bir denge sağlama kapasitesine bağlıdır. Bu dengenin amacı, optimum olmayan yerel çözümlere takılıp kalmamak veya mevcut çözüme yakın ve hedef çözüme daha yakın olabilecek hassas çözümleri tespit etmektir. EA'larda arama süreci genellikle keşifle başlar ve daha sonra kademeli olarak sömürüye dönüşür (Črepinšek ve ark., 2013).

MA'lar, nüfusa dayalı küresel arama ile yerel aramanın birleşimidir. Keşfetme aşamasında popülasyon tabanlı algoritma, çözüm uzayını keşfederek aday çözümlerin aranmasına ve bulunmasına rehberlik eder. Daha sonra, yerel arama yöntemi çağrılarak sömürü aşaması tetiklenir. Yerel arama genellikle bir aday çözümden başlar ve yinelemeli olarak bir komşu çözüme doğru hareket eder, böylece arama uzayındaki komşuluktan yararlanarak bir aday çözümü daha da geliştirebilir. Global arama ve yerel arama arasındaki bu iş birliği, çözüm uzayında optimal çözümün çok daha fazla aranmasını dolayısıyla MA'ların arama verimliliğinin evrimsel algoritmalarından daha iyi performans göstermesini sağlar. Sonuç olarak memetik algoritmalar, tek başına popülasyon tabanlı küresel arama veya tek başına yerel arama kullanmaktan daha yüksek performans sağlamaktadır (Lin ve Chen, 2011).

MA'lar bünyesinde kullanılan yerel arama yöntemleri göz önüne alındığında, baştan sona kadar her bir boyutta arama yapmanın LSGO alanında kullanılan yaygın ve etkili bir yöntem olduğu görülmektedir. Bu tekniği kullanan MTSLS1 yöntemi ve varyantları, başarılı algoritmalar tarafından çokça tercih edilmiştir (Zhao ve ark., 2011; LaTorre ve ark., 2012; 2013; Molina ve Herrera, 2015; Salcedo-Sanz ve ark., 2016; Molina ve ark., 2018b; Hadi ve ark., 2019). Hem Solis and Wets hem de MTSLS1 yerel arama yöntemlerinde kullanılan bir diğer etkili işlem, yeni olası çözümü ararken mevcut boyutun zıt yönlerindeki noktalara bakmaktır. Bu tez çalışmasında önerilen yöntem dinamik adım boyutuna sahip altın oran rehberli yerel arama (Golden ratio guided local search with dynamic step size , GRGLS), altın oran rehberliğinde adım boyutunu dinamik

olarak değiştirip mevcut çözümün zıt yönlerinde çözümü araştırarak olası çözümün her bir boyutunda tek tek çalışarak daha kaliteli bir çözüm elde etmeyi hedeflemektedir. Dolayısıyla önerilen yerel arama yöntemi bu alanda kabul görmüş iki etkili tekniği de bünyesinde barındıracak şekilde tasarlanmıştır.

Önerilen GRGLS yerel arama yönteminde problemin her bir boyutunun kendi adım büyüklüğü vardır. Adım büyüklüğüne farklı başlangıç değerleri verilerek yapılan deneylerin sonuçları değerlendirildikten sonra her bir boyut için adım büyüklüğü optimum başlangıç değeri 0.1 olarak belirlenmiştir. Bu adım boyutları, problem boyutu ile aynı büyüklükte öge içeren dinamik adım boyutu (*dab*) dizisinde saklanır. Her boyutun adım büyüklüğü, en iyi bireyin uygunluk değerinin nasıl geliştiğine bağlı olarak arama süreci boyunca değişir. Mevcut çözümü iyileştirerek daha iyi yeni bir çözüm elde etmek için her boyutta o boyutun adım büyüklüğü kadar zıt noktalar değerlendirilir. Bu yöntemin performansını etkileyen en önemli faktör uyarlanabilir adım boyutudur.

GRGLS yerel arama yöntemi popülasyondaki en yüksek uygunluk değerine sahip olan en iyi birey üzerinde baştan sona kadar her bir boyut üzerinde çalışmaktadır. Yöntem ile ilgili tüm denklemler ve şekillerde, x_i^t ilgili boyutun mevcut değerini, x_i^{t+1} ilgili boyutun bir sonraki değerini, x_{eniyi}^t popülasyonda bulunan en iyi çözümü, dab_i^t ilgili boyuta karşılık gelen adım büyüklüğünü, dab_i^{t+1} ilgili boyuta karşılık gelen adım büyüklüğünün yeni değeri, *GR* altın oran (1.618) değerini, *IGR* ters altın oran (0.618) değerini temsil etmektedir.

İlk olarak, en iyi bireyden daha iyi bir çözüm araştırmak amacıyla, en iyi bireyin ilgili boyutun kendi adım büyüklüğü kadar ilerisinde olan yeni aday çözüm Denklem 4.6 kullanılarak hesaplanır.

$$x_i^{t+1} = x_i^t + dab_i^t \quad (4.6)$$

En iyi bireyin ilgili boyut değerinin adım büyüklüğü kadar artırılmasıyla oluşan yeni aday çözüm, amaç fonksiyonu tarafından değerlendirilir. Aday çözüm popülasyondaki en iyi çözümden daha iyi bir sonuç veriyor ise aday çözüm popülasyonun en iyi bireyi olarak belirlenir ve ilgili boyutun bu çözüme ulaşmayı sağlayan adım büyüklüğü altın oranla çarpılarak güncellenir. Aday çözüm popülasyondaki en iyi çözümden daha kötüyse, bu sefer daha iyi bir olası çözümü araştırmak amacıyla popülasyondaki en iyi bireyin kendi adım büyüklüğü kadar gerisindeki nokta Denklem 4.7 kullanılarak hesaplanır.

$$x_i^{t+1} = x_i^t - dab_i^t \quad (4.7)$$

En iyi bireyin ilgili boyut değerinin adım büyüklüğü kadar eksiltilmesiyle oluşan yeni aday çözüm, amaç fonksiyonu tarafından değerlendirilir. Eğer aday çözüm en iyi çözümden daha iyiyse, aday çözüm popülasyonun en iyi bireyi olarak belirlenir ve ilgili boyuta ait adım büyüklüğü altın oranla çarpılarak güncellenir, aksi takdirde ise en iyi birey değişmez ve değiştirilen boyuta ait adım büyüklüğü ters altın oranla çarpılarak güncellenir.

$$dab_i^{t+1} = \begin{cases} dab_i^t * GR, & f(x_i^{t+1}) \leq f(x_{eni}^t) \\ dab_i^t * IGR, & f(x_i^{t+1}) > f(x_{eni}^t) \end{cases} \quad (4.8)$$

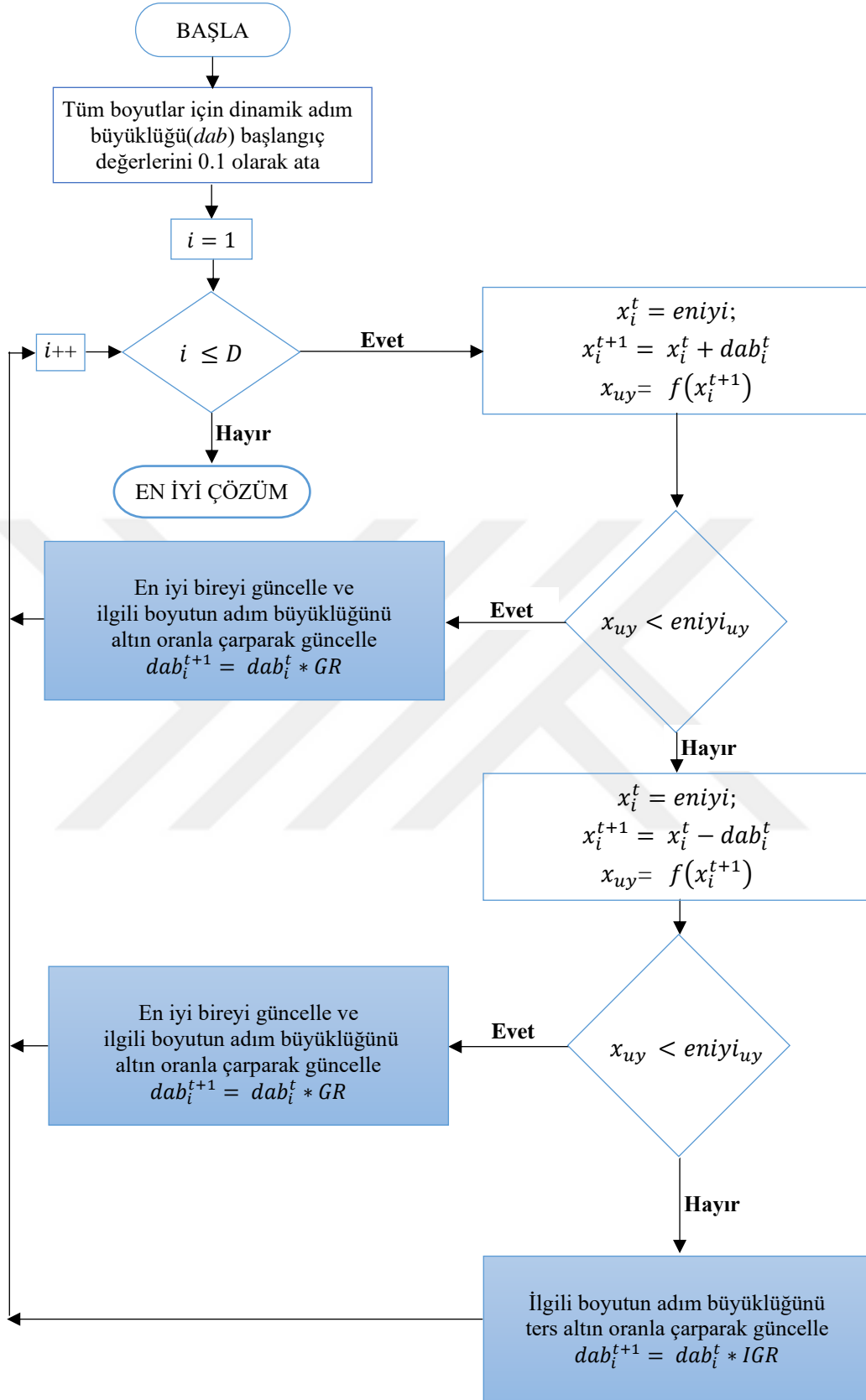
Adım büyüklüklerinin eklenip çıkarılmasıyla gerçekleşen her hareket sonrası konumlanılan yeni noktaları için fonksiyon aralık sınırları dışında mı kontrolü yapıp, sınır dışındaysa sınır değerleri olarak güncellenerek devam edilir. Anlatılan yerel arama işleyişindeki tüm bu adımlar, problemin ilk boyutundan son boyutuna ulaşılan kadar bir sonraki boyuta geçilerek tekrarlanır. Tüm boyutlar bittiğinde veya maxFEs tükendiğinde en iyi bireyin gelişimi tamamlanmış olur ve elde edilen çözüm algoritma tarafından elde edilen en iyi çözümdür denir.

GRGLS yerel arama yönteminin sözde kodu Çizelge 4.2’de gösterilmektedir. Bu sözde kod bünyesinde kullanılan D ifadesi problemin boyutunu, i ifadesi tüm boyutları sırayla gezmeyi sağlayan sayacı temsil etmektedir.

Çizelge 4.2. GRLGS'nin sözde kodu (Koçer ve Uymaz, 2021)

1 :	Tüm boyutların adım büyüklüklerinin başlangıç değerini 0.1 olarak ata
2 :	for i=1 to D
3 :	En iyi bireyi aday çözüm olarak ata
4 :	Denklem 4.6 aracılığıyla daha iyi bir çözüm araştır
5 :	Adım hareketi sonrası fonksiyon sınır ihlalinin kontrol et ve gerekiyorsa düzelt
6 :	Aday çözümün amaç fonksiyon değerini hesapla
7 :	if aday çözüm uygunluk değeri en iyi bireyinkinden daha iyi ise
8 :	Bireyin pozisyonunu aday çözümün pozisyonunu olarak ve bireyi en iyi birey olarak ata
9 :	İlgili boyutun adım büyüklüğünü Denklem 4.8 altın oran aracılığıyla güncelle
10 :	else if
11 :	En iyi bireyi aday çözüm olarak ata
12 :	Denklem 4.7 aracılığıyla daha iyi bir çözüm araştır
13 :	Adım hareketi sonrası fonksiyon sınır ihlalinin kontrol et ve gerekiyorsa düzelt
14 :	Aday çözümün amaç fonksiyon değerini hesapla
15 :	if aday çözüm uygunluk değeri en iyi bireyinkinden daha iyi ise
16 :	Bireyin pozisyonunu aday çözümün pozisyonunu olarak ata
17 :	Bireyi en iyi birey olarak ata
18 :	İlgili boyutun adım büyüklüğünü Denklem 4.8 altın oran aracılığıyla güncelle
19 :	else if
20 :	İlgili boyutun adım büyüklüğünü Denklem 4.8 ters altın oran aracılığıyla güncelle
21 :	End
22 :	End

GRGLS yerel arama yönteminin işleyişi Şekil 4.1'de akış şeması olarak gösterilmektedir.



Şekil 4.1. GRGLS Akış Seması (Koçer ve Uymaz, 2021)

Önerilen yerel arama yönteminin performansını değerlendirmek için memetik bir algoritmaya ihtiyaç vardır. Memetik algoritma veya çerçeveler, tasarımları nedeniyle bir veya daha fazla yerel arama yöntemi kullanabilir. Önerilen yöntemin etkinliğini ve başarısını değerlendirmek için diğer tekniklerin katkısından bağımsız değerlendirme yapabilmek amacıyla bünyesinde tek bir yerel arama yöntemi kullanan MA'lar ele alınmıştır. Önerilen tekniğin etkisi, katkısı ve farkını gösterebilmek amacıyla özellikle önerilen yöntemle ortak özelliklere sahip yerel arama yöntemi kullanan algoritmalar üzerinde durulmuştur. Bu nedenlerden dolayı LSGO yarışmalarında da başarısını kanıtlamış olan MLSHADE-SPA (Hadi ve ark., 2019) çerçeve algoritması testlerde kullanılacak ana algoritma olarak seçilmiştir. Çerçeve algoritmasının orijinal halinde yerel arama aşamasında kullanılan MMTS yöntemi çerçeve algoritmadan tamamen çıkartılarak yerine önerilen yeni yerel arama yöntemi GRGLS kullanılmıştır. Oluşan yeni çerçeve algoritmasına Geliştirilmiş MLSHADE_SPA (Improved MLSHADE_SPA, IMLSHADE_SPA) adı verilmiştir. Şekil 4.2'de IMLSHADE_SPA çerçeve algoritmasının yapısı ve işleyişi anlatılarak bu yapı bünyesinde GRGLS kullanımı mavi alanla vurgulanarak gösterilmiştir.

MEMETİK ÇERÇEVE ALGORİTMASI (Improved MLSHADE-SPA)	Başlatma	Rastgele üretilmiş 250 bireye sahip başlangıç popülasyonu oluşturulur. Çerçeve bünyesinde kullanılan algoritma ve yöntemlerin (LSHADE_SPA, EADE, ANDE, MMTS) parametrelerine başlangıç değerleri atanır. Çerçeve algoritmanın tamamında kullanılacak FEs (maxFEs), iterasyon sayısına (it= 50) bölünerek iterasyon başına tüketilebilecek FEs belirlenir. Her bir iterasyon bünyesinde popülasyon tabanlı algoritmalar ve yerel arama olmak üzere iki aşama olduğu için iterasyon başına tüketilebilecek FEs iki eşit parçaya bölünüp yarısını popülasyon tabanlı algoritmaların (iFEsPop) diğer yarısını da yerel arama yönteminin (iFEsLS) tüketmesi sağlanır. (Şekil 3.1'de FEs paylaşımı gösterilmiştir)
	Popülasyon Tabanlı Evrimsel Algoritmalar	iFEsPop ikiye bölünür ve yarısı tüm boyutlar üzerinde çalışan LSHADE_SPA algoritması tarafından tüketilirken diğer yarısı ise LSHADE_SPA, ANDE, EADE algoritmaları arasında paylaştırılarak tüketilir. Bu üç algoritmadan her biri parçala fethet anlayışıyla kendisine düşen boyutlarda kendine düşen değerlendirme sayısını tüketerek çalışırlar. Üç algoritma arasındaki paylaşım ilk iterasyonda eşit yapılırken daha sonraki iterasyonlarda her algoritmanın kendi performansını gösteren iyileştirme oranına göre iFEsPop/2 kısmı üç algoritma arasında paylaştırılır.
	Yerel Arama	İterasyon başına tüketilecek uygunluk sayısının yarısı yerel arama metodu tarafından tüketilmektedir. Bu aşamada önerilen yerel arama yöntemi GRGLS elde edilen en iyi çözümün her bir boyutu üzerinde tek tek çalışarak çözümü daha da geliştirmeye çalışır.
	Sonlandırma	Algoritmanın performansını artırma amacıyla her iterasyon sonunda doğrusal popülasyon büyüklüğü azaltma yöntemi uygulanır. maxFEs'in ilk yarısında popülasyon büyüklüğü minimum değer olan 20 ye ulaşır. 50 iterasyon tüm aşamalar döngüsel tekrarlanıp tamamlanınca ya da maxFEs tamamen tüketildiğinde elde edilebilecek en iyi sonuca ulaşılmış olur.

Şekil 4.2. IMLSHADE-SPA yapısı

4.3. Kaotik altın oran rehberli yerel arama (CGRGLS) ve CIMLSHADE-SPA

Düzenin düzensizlikten kaynaklandığı biyolojik sistemler, birçok metasezgisel optimizasyon tekniğine ilham vermiştir. Kaos, başlangıç koşullarına aşırı duyarlı, düzenli bir düzensizlik hali olarak tanımlanması dolayısıyla düzensizlikten düzen yaratmaya yardımcı olabilmektedir. Kaotik haritalar, oluşturduğu sayı dizilerinde sayıların tekrarlanmaması, belirli bir aralıkta sıkışmayıp geniş bir spektruma yayılmış olması ve dizi üretim ve saklama maliyetinin düşük olması gibi nedenlerle metasezgisel algoritmalar bünyesinde rastgele sayı üretici olarak sıklıkla kullanılmaktadır. Kaostan ilham alınarak önerilen yerel arama yönteminin performansı geliştirilmiş ve geliştirilen bu yeni versiyon Kaotik Altın Oran Rehberli Yerel Arama (Chaotic Golden ratio guided local search, CGRGLS) olarak adlandırılmıştır. Bu yöntemde kaos, rastgele sayı üretici olarak kullanılmaktadır. Önceki versiyon olan GRGLS'nin, uyarlanabilir adım büyüklüğü, altın oran ve ters altın oranın katsayılarıyla değiştirilmekteyken yeni versiyon CGRGLS'de Singer kaotik haritasının ürettiği rastgele değer, adım büyüklüğü değişimini hesaplayan mevcut denklemde altın oran veya ters altın oran çarpanının yanısıra bir çarpan olarak kullanılmıştır. Bu sayede altın oran rehberliğinde yerel aramayı yönlendiren hareketleri sağlayan adım büyüklüğü değişiminin kaotik olarak etkilenmesi sağlanmıştır.

CGRGLS yerel arama yöntemi adım büyüklüğü değiştirme adımları hariç GRGLS yerel arama yöntemiyle birebir aynı işleyişe sahiptir. Şekil 4.1'de mavi renkte gösterilen adımlarda adım büyüklüğünü güncelleyen Denklem 4.8 yerine aşağıda gösterilen Denklem 4.9 kullanılarak mevcut yerel arama yönteminin performansı artırılmış ve yeni versiyon CGRGLS olarak adlandırılmıştır.

Denklemde kaotik sayıyı temsilen kullanılan KS çarpanı Singer kaotik haritası tarafından üretilen bir sayıdır. Böylelikle yeni olası çözüm arayışını sağlayan hareketlerin kaotik olarak etkilenmesi sağlanmıştır.

$$dab_i^{t+1} = \begin{cases} dab_i^t * GR * KS, & f(x_i^{t+1}) \leq f(x_{eni}^t) \\ dab_i^t * IGR * KS, & f(x_i^{t+1}) > f(x_{eni}^t) \end{cases} \quad (4.9)$$

CGRLGS yerel arama yönteminin MLSHADE-SPA algoritmasının yerel arama kısmında kullanılmasıyla büyük ölçekli optimizasyon problemleri çözümü için kullanılabilir CIMLSHADE-SPA algoritması önerilmiştir.

5. DENEYSSEL ÇALIŞMALAR VE TARTIŞMA

Bu bölümde, büyük ölçekli optimizasyon problemlerinin çözümüne katkı sağlamak amacıyla gerçekleştirilen deneysel çalışmalar açıklanmış ve elde edilen sonuçlar listelenip yorumlanmıştır.

5.1. Yapay alg algoritmasının LSGO performansının değerlendirilmesi

Optimizasyon problemlerinin çözümü için etkili yaklaşımlar olarak bilinen metasezgisel algoritmalar, normal büyüklükteki problemlerde yüksek arama kabiliyetleri sergilemekle birlikte, büyük ve karmaşık problemlere uygulandığında çoğu zaman tasarımındaki performanslarını ve etkisini kaybetmektedir (Tang ve ark., 2009). Bu deneysel çalışmada, Yapay Alg Algoritmasının LSGO problemleri çözüm performansı test edilmiş ve elde edilen sonuçlar LSGO problemlerini çözme amacıyla tasarlanmış bir algoritma olan grup tabanlı problem ayrıştırma stratejisine sahip CC tabanlı diferansiyel gelişim çerçeve algoritması (Differential Evolution Cooperative Coevolution with Group, DECC-G) (Yang ve ark., 2008b) ile karşılaştırılarak büyük ölçekli optimizasyon alanındaki potansiyeli değerlendirilmiştir.

İlk olarak problem boyutu büyüdükçe AAA'nın performansının nasıl etkilendiğini görmek için algoritma, boyutları 100, 500, 1000 boyut olacak şekilde test fonksiyon kümesinin ilk 3 fonksiyonunun çözümü için çalıştırılmıştır. MaxFEs değerleri ise boyutla orantılı olacak şekilde 100, 500 ve 1000 boyut için sırasıyla 3.0e5, 1.5e6 ve 3.0e6 olarak ayarlanmıştır.

Tüm deneysel çalışmalar Matlab kullanılarak yapılmıştır. AAA parametreleri (Uymaz ve ark., 2015) de önerildiği gibi enerji kaybı olan $e = 0.3$, kesme kuvveti olan $\Delta = 2$ ve adaptasyon parametresi olan $A_p = 0.5$, popülasyon boyutu olan $pop = 40$ olarak kullanılmıştır. 25 ayrı çalıştırma sonucunun ortalamaları olarak elde edilen tüm sonuçlar Çizelge 5.1'de gösterilmiştir.

Çizelge 5.1. Boyut artışının AAA performansı üzerindeki etkisi

	AAA(D=100, FEs=3.0e5)			AAA(D=500, FEs=1.5e6)			AAA(D=1000, FEs=3.0e6)		
	Best	Mean	Std.	Best	Mean	Std.	Best	Mean	Std.
F1	0.00E+00	0.00E+00	0.00E+00	1.25E-17	1.02E-16	9.30E-17	4.39E-17	3.02E-16	2.42E-16
F2	0.00E+00	1.28E+00	1.63E+00	1.37E-12	4.38E+00	3.41E+00	2.98E+00	8.14E+00	5.29E+00
F3	1.42E-14	1.73E-14	1.18E-15	3.19E-11	6.55E-04	3.28E-03	2.89E-11	5.48E-03	2.51E-02

Çizelge 5.1'de görüldüğü gibi AAA, F1 ve F2 fonksiyonlarında 100 boyut için optimal sonuca ulaşmış fakat boyut büyüdükçe bu başarısını koruyamamıştır. Sonuçların standart sapma değerleri dikkate alındığında boyut arttıkça standart sapmanın da büyüdüğü görülmektedir. Bu durum katlanarak artan arama uzayında yerel çözümlere takılıp kalmaktan kaynaklanabilir. Çizelge 5.1 problem boyutu arttıkça AAA performansının beklendiği gibi kötüleştiğini göstermektedir.

AAA ile elde edilen sonuçlar CEC2010 LSGO özel oturum düzenleyicileri tarafından önerilen DECC-G (Yang ve ark., 2008b) algoritması ile karşılaştırılmıştır. DECC-G algoritması, büyük ölçekli optimizasyon problemlerini çözme amacıyla tasarlanmış bir CC yaklaşımıdır. Algoritma, amaç vektörünü alt bileşenlere ayırarak bileşenlerden her birini bir evrimsel algoritma ile geliştirmektedir. Temel alt bileşen optimize edicisi olarak güçlü bir DE algoritması olan SaNSDE'yi (Yang ve ark., 2008a) kullanmaktadır. Bu CC çerçevesi, etkileşim halindeki değişkenlerin birlikte optimize edilmesi şansını artıracak şekilde gruplama yapılarını dinamik olarak değiştiren grup tabanlı bir problem ayrıştırma stratejisi uygulamaktadır. Tüm alt bileşenler arasında uyumu sağlamak için ise her döngüden sonra alt bileşenlerin her birine bir ağırlık uygulayan ve ağırlık vektörünü belirli bir optimize edici ile geliştiren uyarlanabilir bir ağırlıklandırma stratejisi uygulamaktadır.

DECC-G ve AAA algoritmalarının Çizelge 3.3'te listelenen CEC2010 test fonksiyon kümesi üzerindeki performansı maximum uygunluk değerlendirme sayısı (MaxFES) 3.0e6 olacak şekilde test edilmiş ve tüm sonuçlar Çizelge 5.2'de listelenmiştir.

Çizelge 5.2 incelendiğinde, AAA'nın orijinal halinin büyük ölçekli optimizasyon problemlerinin çözümüne yönelik herhangi bir düzenleme veya müdahale yapılmamasına rağmen 20 fonksiyonun 11'inde büyük ölçekli problem çözümü için tasarlanmış olan DECC-G algoritmasından daha iyi sonuçlar elde ettiği görülmüş olup sonuçlar aşağıdaki şekilde yorumlanmıştır:

- ✓ Test fonksiyon kümesinin ilk üç fonksiyonu tamamen ayrılabilir olan F1, F2, F3 fonksiyonlarıdır ve AAA, bu fonksiyon grubunda DECC-G'den daha iyi sonuçlar elde etmiştir.
- ✓ Test fonksiyon kümesinin tek grup m-ayrılmaz fonksiyonlar kategorisinde, DECC-G algoritması, F8 dışında AAA'dan daha iyi sonuçlar elde etmiştir. DECC-G algoritmasının uyguladığı uyarlamalı ağırlıklandırma stratejisi ve CC yaklaşımı ayrılamayan problemlerle başa çıkmada algoritmaya avantaj sağlamaktadır.

- ✓ Kategori bazlı başarıyı değerlendirmek için diğer kategorilere bakıldığında algoritmalarından birinin diğerinden net bir şekilde daha başarılı olduğunu söylemek mümkün değildir.

Çizelge 5.2. AAA ve DECC-G karşılaştırma sonuçları

	DECC-G			AAA		
	En İyi	Ortalama	Std.	En İyi	Mean	Std.
F1	1.63E-07	2.93E-07	8.62E-08	4.39E-17	3.02E-16	2.42E-16
F2	1.25E+03	1.31E+03	3.26E+01	2.98E+00	8.14E+00	5.29E+00
F3	1.20E+00	1.39E+00	9.73E-02	2.89E-11	5.48E-03	2.51E-02
F4	7.78E+12	1.70E+13	5.37E+12	1.23E+13	1.86E+13	4.35E+12
F5	1.50E+08	2.63E+08	8.44E+07	3.34E+08	4.32E+08	5.73E+07
F6	3.89E+06	4.96E+06	8.02E+05	1.89E+07	1.95E+07	1.90E+05
F7	4.26E+07	1.63E+08	1.37E+08	7.55E+09	1.28E+10	3.61E+09
F8	6.37E+06	6.44E+07	2.89E+07	4.15E+03	2.99E+06	3.88E+06
F9	2.66E+08	3.21E+08	3.38E+07	1.65E+08	2.13E+08	2.51E+07
F10	1.03E+04	1.06E+04	2.95E+02	5.65E+03	6.12E+03	2.64E+02
F11	2.06E+01	2.34E+01	1.78E+00	1.97E+02	2.12E+02	3.49E+00
F12	7.78E+04	8.93E+04	6.87E+03	1.45E+05	1.92E+05	2.53E+04
F13	1.78E+03	5.12E+03	3.95E+03	4.10E+02	7.55E+02	1.84E+02
F14	6.96E+08	8.08E+08	6.07E+07	4.92E+08	5.85E+08	5.07E+07
F15	1.09E+04	1.22E+04	8.97E+02	1.01E+04	1.10E+04	5.47E+02
F16	5.97E+01	7.66E+01	8.14E+00	3.93E+02	3.94E+02	8.54E-01
F17	2.50E+05	2.87E+05	1.98E+04	4.22E+05	4.99E+05	4.37E+04
F18	5.61E+03	2.46E+04	1.05E+04	2.26E+03	6.02E+03	3.67E+03
F19	1.02E+06	1.11E+06	5.15E+04	2.51E+06	3.23E+06	2.91E+05
F20	3.59E+03	4.06E+03	3.66E+02	1.62E+03	1.97E+03	1.77E+02

DeneySEL çalışmalar sonucunda AAA algoritmasının LSGO problemleri çözümünde kullanılabilecek potansiyele sahip olduğu görülmüş ve algoritmanın büyük ölçekli problem çözüm performansını geliştirme amaçlı çalışmalar yapılmasına karar verilmiştir.

5.2. Yapay alg algoritmasının LSGO performansının geliştirilmesi

Yapay Alg Algoritmasını LSGO alanında daha etkili hale getirebilmek amacıyla işleyişi incelenmiş ve algoritmanın hem keşif hem de sömürü mekanizmasına katkıda bulunan helisel hareket aşamasında diferansiyel gelişim algoritmasının mutasyon stratejilerinden ilham alınarak ayrıntılarının Bölüm 4.1'de anlatıldığı gibi değişiklik

yapılmıştır. Oluşan bu yeni versiyon, Değiştirilmiş Yapay Alg Algoritması (Modified Artificial Algae Algorithm, MAAA) olarak adlandırılmıştır.

AAA ve MAAA, Çizelge 3.3'te listelenen test fonksiyon kümesinin ilk 10 fonksiyonunun çözümü için eşit şartlarda ve aynı parametrelerle çalıştırılarak performansları birbirleriyle karşılaştırılmıştır. Her biri 1000 boyutlu olan 10 fonksiyonun tamamı minimizasyon problemi olup optimum çözüm değerleri 0'dır. Yapılan testlerde MaxFEs, 3.0e6 olarak kullanılmıştır.

AAA parametreleri (Uymaz ve ark., 2015) de önerildiği gibi enerji kaybı olan $e = 0.3$, kesme kuvveti olan $\Delta = 2$ ve adaptasyon parametresi olan $A_p = 0.5$, popülasyon boyutu olan $pop = 40$ olarak kullanılmıştır. Tüm deneysel çalışmalar Matlab kullanılarak yapılmıştır ve elde edilen tüm sonuçlar, 25 ayrı çalıştırma sonucunun ortalamalarıdır.

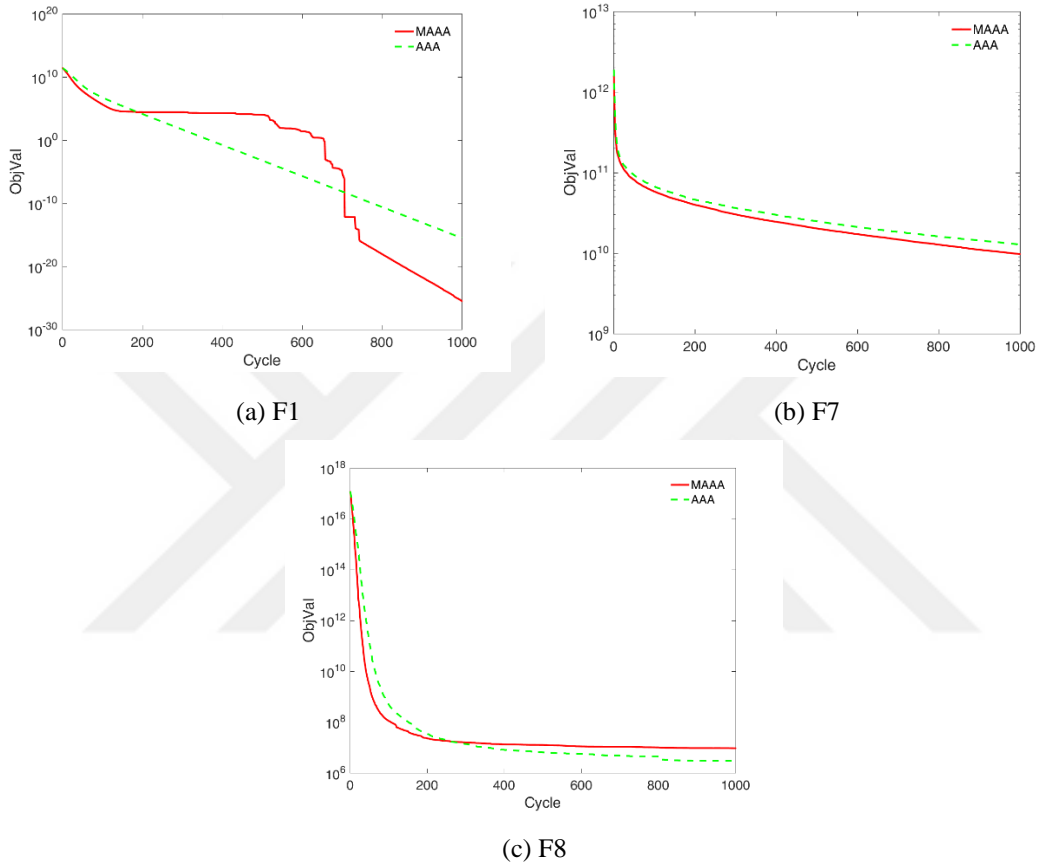
Herbir çalıştırma sonunda elde edilen sonuçlar sıralanarak 1. (En İyi), 13. (Medyan) ve 25. (En Kötü) olmak üzere çalıştırma ortalaması (Ortalama) ve standart sapma (Std.) hesaplanarak ilgili fonksiyona ait en iyi değerler koyu renkle işaretlenerek tüm sonuçlar Çizelge 5.3'te sunulmuştur.

Çizelge 5.3. AAA ve MAAA karşılaştırma sonuçları

	AAA				MAAA			
	En İyi	Medyan	Ortalama	Std.	En İyi	Medyan	Ortalama	Std.
F1	4.39E-17	2.60E-16	3.02E-16	2.42E-16	0.00E+00	2.71E-28	3.06E-26	1.24E-25
F2	2.98E+00	6.26E+00	8.14E+00	5.29E+00	6.08E+01	8.08E+01	8.06E+01	1.45E+01
F3	2.89E-11	8.99E-11	5.48E-03	2.51E-02	2.52E-13	4.12E-13	4.71E-13	2.06E-13
F4	1.23E+13	1.89E+13	1.86E+13	4.35E+12	8.22E+12	1.55E+13	1.62E+13	4.62E+12
F5	3.34E+08	4.21E+08	4.32E+08	5.73E+07	2.97E+08	4.19E+08	4.24E+08	6.88E+07
F6	1.89E+07	1.96E+07	1.95E+07	1.90E+05	1.78E+07	1.95E+07	1.94E+07	3.76E+05
F7	7.55E+09	1.24E+10	1.28E+10	3.61E+09	4.26E+09	9.48E+09	9.72E+09	2.87E+09
F8	4.15E+03	1.61E+06	2.99E+06	3.88E+06	1.46E+04	1.90E+06	9.38E+06	1.31E+07
F9	1.65E+08	2.21E+08	2.13E+08	2.51E+07	1.49E+08	1.86E+08	1.91E+08	2.16E+07
F10	5.65E+03	6.16E+03	6.12E+03	2.64E+02	6.41E+03	6.75E+03	6.78E+03	3.02E+02

Çizelge 5.3'te listelenen sonuçlara göre, önerilen MAAA test edilen on fonksiyondan yedisinde hem ortalama hem de en iyi değerlerde AAA algoritmasının orijinal haline göre daha iyi sonuçlar elde etmiştir. Bu da yapılan değişikliğin orijinal versiyonu geliştirerek LSGO problemlerin çözümünde performans artışı sağladığını göstermektedir. Hatta F1 fonksiyonunda MAAA optimum çözüme ulaşabilmiştir.

Örnek olarak seçilen üç problem (F1, F7, F8) için algoritmanın orijinal hali AAA ile değiştirilmiş hali MAAA'nın yakınsama eğrileri Şekil 5.1'de sunulmaktadır. Y eksenini, ilgili fonksiyonun 25 bağımsız çalışmasının ortalama değerlerini temsil eder. X eksenini, her 3000 uygunluk değerlendirmesinde kaydedilen bir noktayı temsil eder. Örneğin x eksenindeki 200 değeri $6.0e5$ FE'lere karşılık gelmektedir.



Şekil 5.1. Örnek olarak seçilen fonksiyonlarda AAA ve MAAA yakınsama eğrileri

Şekil 5.1'de yakınsama eğrileri verilen fonksiyonlardan F1 algoritmanın bu fonksiyonda optimum çözüme ulaşması nedeniyle F7 ve F8 ise farklı algoritmaların üstün olduğu fonksiyonlar olması nedeniyle örnek olarak seçilmiş ve yakınsama eğrileri aşağıdaki gibi yorumlanmıştır:

- Şekil 5.1(a), MAAA'nın $6.0e5$ uygunluk değerlendirmeye kadar AAA'dan çok daha hızlı olduğunu göstermektedir. Bu noktadan sonra AAA daha iyi sonuçlar bulmaya devam ederken MAAA $1.5e6$ uygunluk değerlendirmelere kadar sabit bir grafik çiziyor. Bu noktadan sonra ise, MAAA'nın çok hızlı bir şekilde optimuma yakınsadığı görülmektedir.

- Şekil 5.1(b), MAAA'nın AAA ile aynı yakınsama özelliğine sahip olduğunu ve bu nedenle paralel yakınsama grafiğini çizdiklerini göstermektedir. Ancak MAAA nispeten daha hızlı yakınsaması nedeniyle AAA'dan daha iyi sonuçlar bulmaktadır.
- Şekil 5.1(c), MAAA'nın yaklaşık $8e5$ uygunluk değerlendirmeye kadar AAA'dan çok daha hızlı olduğunu göstermektedir. Ancak bu noktadan sonra AAA, MAAA'yı çok yakın bir performansla geçerek daha iyi sonuçlar elde etmiştir.

MAAA'nın rakiplerine göre performansının değerlendirilebilmesi amacıyla önerilen algorithmadan elde edilen sonuçlar Çizelge 5.4'te listelenen beş algoritmanın sonuçlarıyla karşılaştırılarak Çizelge 5.5'te listelenmiştir.

Çizelge 5.4. MAAA ile karşılaştırılan algoritmalar

Algoritma	Ad	Referans
SDENS	Komşu arama ile geliştirilmiş ardışık DE	(Wang ve ark., 2010)
DNPSO	Çeşitliliği geliştirilmiş komşuluk aramalı PSO	(Wang ve ark., 2013)
DASA	Diferansiyel Karınca Damgalama Algoritması	(Korošec ve ark., 2010)
DECC-G	Rastgele gruplamalı CC yaklaşımı DE	(Yang ve ark., 2008b)
LOCUST SWARMS (LS)	Çekirge Sürüleri çoklu optimum arama tekniği	(Chen)

MAAA sonuçları, Matlab kodlarını çalıştırarak elde edilmiş olup diğer algoritmalarının sonuçları doğrudan Çizelge 5.4'te belirtilen referanslardan alınmıştır. Karşılaştırmalar, $3.0e6$ FEs tüketerek gerçekleştirilen 25 bağımsız çalıştırmanın ortalaması baz alınarak yapılmıştır.

Çizelge 5.5. MAAA diğer algoritmalarla karşılaştırma sonuç ve sıralamaları

	Ortalama Değerler						Sıralamalar					
	MAAA	SDENS	DNPSPO	DASA	DECC-G	LS	MAAA	SDENS	DNPSPO	DASA	DECC-G	LS
F1	3.06E-26	5.73E-06	1.87E+07	1.52E-21	2.93E-07	3.48E+05	1	4	6	2	3	5
F2	8.06E+01	2.21E+03	5.85E+03	8.48E+00	1.31E+03	1.24E+03	2	5	6	1	4	3
F3	4.71E-13	2.70E-05	1.93E+01	7.20E-11	1.39E+00	2.67E+00	1	3	6	2	4	5
F4	1.62E+13	5.11E+12	2.25E+12	5.05E+11	1.70E+13	6.65E+12	5	3	2	1	6	4
F5	4.24E+08	1.18E+08	1.57E+08	6.20E+08	2.63E+08	3.60E+08	5	1	2	6	3	4
F6	1.94E+07	2.02E-04	1.75E+06	1.97E+07	4.96E+06	1.91E+07	5	1	2	6	3	4
F7	9.72E+09	1.20E+08	8.60E+06	7.78E+00	1.63E+08	6.32E+08	6	3	2	1	4	5
F8	9.38E+06	5.12E+07	1.31E+07	4.98E+07	6.44E+07	3.10E+07	1	5	2	4	6	3
F9	1.91E+08	5.63E+08	3.16E+08	3.60E+07	3.21E+08	2.22E+08	2	6	4	1	5	3
F10	6.78E+03	6.87E+03	6.90E+03	7.29E+03	1.06E+04	6.04E+03	2	3	4	5	6	1
Sıralama Ortalamaları =							3.0	3.4	3.6	2.9	4.4	3.7

Çizelge 5.5'te listelenen sonuçlara göre, MAAA en iyi sonuçları F1, F3 ve F8'de alırken, DASA en iyi sonuçları F2, F4, F7 ve F9'da alır. Bu nedenle MAAA, DASA'dan sonraki ikinci algoritmadır. İlgili çizelgenin sıralamalar kısmı, her bir algoritmanın tüm algoritmalar arasındaki sıralamasını göstermektedir. Her algoritmanın ortalama sıraları, MAAA'nın altı algoritma arasında en iyi ikinci algoritma olduğunu kanıtlamaktadır.

LSGO sorunlarıyla başa çıkmak için tercih edilen çözüm yaklaşımlarından biri standart evrimsel algoritmaların ek düzenleme ve tekniklerle geliştirilmesidir. Deneysel çalışmalar göstermiştir ki DE'nin mutasyon stratejilerinden esinlenerek AAA helisel hareketinden sonra aynı koloninin rastgele seçilen dördüncü bir boyutunun (alg hücresinin) güncellenmesi algoritmanın keşif aşamasına katkıda bulunmuş ve LSGO problemlerinin çözümünde AAA algoritmasının performansını artırmıştır.

5.3. GRGLS ve IMLSHADE-SPA

Evrimsel algoritmaların yerel arama yöntemleri ile birleştirilmesiyle tasarlanan algoritmalar MA olarak adlandırılmaktadır (Moscato, 1989). Keşif ve sömürü arasında iyi bir denge sağlayan bu teknik, LSGO'nun zorluklarını aşmak için etkili ve verimli algoritmaların tasarlanmasını sağlamıştır. MA'lar, büyük ölçekli optimizasyon problemlerinin çözümünde kullanılan yöntemler arasında çokça tercih edilen bir

yöntemdir (Omidvar ve ark., 2021b; 2021a). MA performansı büyük ölçüde bünyesinde kullandığı yerel aramaların başarısına bağlıdır (Smith, 2007; Molina ve ark., 2010a).

MA bünyesinde kullanılacak, boyuta bağlı katlanarak büyüyen çözüm uzayında daha kaliteli çözümler elde ederek algoritmanın performansını artıracak yeni bir yerel arama yöntemi geliştirilmiştir. Önerilen yönteme, dinamik adım boyutuna sahip altın oran rehberli yerel arama (GRGLS) adı verilmiştir.

Altın oran, kâinatın var oluşundan beri var olan doğa, biyoloji, hesaplamalı bilimler, borsa, matematik, mühendislik, endüstriyel tasarım, mimarlık, sanat vb. birçok alanda kullanılan matematiksel harika bir simetri ya da tasarım özelliğidir (Iosa ve ark., 2018). Altın oranın birçok alandaki başarılı tezahürü, optimizasyon algoritmaları bünyesinde katsayı (Manikantan ve ark., 2012), arama tekniği (Aurasopon ve Khamsen, 2019), seçim yöntemi (Cuevas ve ark., 2018) vb. olarak algoritma geliştirme amaçlı kullanılmasına yol açmıştır. Optimizasyon bünyesinde altın oran kullanımlarından ilham alınarak arama uzayını ararken ki hareketi yöneten adım büyüklüklerinin altın oran kılavuzluğunda belirlenmesinin arama performansını artıracığı öngörüsüyle GRGLS yöntemi tasarlanmıştır.

Altın oran (Golden Ratio,GR), fraktal özelliklere sahip irrasyonel bir sayıdır ve 1.61803398874989 değerine eşittir ve ters altın oran (Inverse Golden Ratio, IGR) ise 0.6180339887'dir. GR ve IGR virgülden sonra aynı rakamlara sahiptir. Arama işlemi sırasında keşif ve sömürü arasında bir denge sağlamak amacıyla bireyin uygunluk değerindeki iyileşmeye göre adım boyutu artırılır veya azaltılır. Bu iki altın oran, önerilen arama yöntemi bünyesinde hareketi sağlayan adım büyüklüğünü belirleme ve dinamik olarak değiştirmek amacıyla katsayı olarak kullanılmıştır. Böylece GRGLS yönteminin problem özelliklerine (unimodal, multimodal vb.) göre herhangi bir sınırlama olmaksızın uygulanabilmesi sağlanır.

Bu bölüm kapsamında ilk deneysel çalışma yerel arama yönteminin etkisini ve farkını göstermek amacıyla yapılmıştır. Ana çerçeve olarak seçilen MLSHADE-SPA (Hadi ve ark., 2019) algoritmasının yerel arama kullandığı kısmına her seferinde MTSL1, MMTS ve GRGLS arama yöntemlerinden sadece birini yerleştirerek çerçeve algoritma çalıştırılmış böylece yerel arama yöntemlerin algoritma performansına etkileri incelenerek bu 3 yerel arama yöntemi birbirleriyle kıyaslanmıştır. Bu üç arama yönteminden biri olan GRGLS Bölüm 4.2 de ayrıntılarıyla anlatılmış ve diğer iki yerel arama yöntemine de aşağıda kısaca değinilmiştir.

MTSLS1'de her bir boyut için başlangıç adım değerleri üst sınır ile alt sınır arasındaki fark olarak belirlenir. Yerel arama sadece en iyi bireyin tüm boyutlarına uygulanır. Her boyutun adım boyutu farklı olabilir. Yerel arama, üzerinde çalışılan boyuta adım boyutu eklenerek veya çıkarılarak uygulanır. İlk olarak, adım boyutu çözümden çıkarılır. Eğer o noktada daha iyi bir sonuca ulaşılmazsa adım boyutunun yarısı mevcut boyutun orijinal değerine eklenir. Bu iki hareket sonucunda hâlâ iyi bir sonuç alınamazsa adım boyutunu yarıya indirir ve yeni adım boyutunun $1.00E-15$ 'in altında olup olmadığı kontrol edilir. Adım boyutu bu değer altına düşerse, boyutun üst ve alt sınırları arasındaki fark 0,4 ile çarpılarak adım boyutu güncellenir. Bu yöntem hakkında daha detaylı bilgi (Tseng ve Chen, 2008) den alınabilir.

MLSHADE-SPA orijinal halinde kullanılan MMTS yerel arama yöntemi aslında MTSLS1 den ilham alan onun değiştirilmiş bir versiyonudur. Bu yöntemde her bir boyut için başlangıç adım değerleri, $[0, 1]$ aralığındaki rasgele bir sayı ile o boyutun mevcut minimum ve maksimum değerleri arasındaki farkın çarpılmasıyla belirlenir. Buna ek olarak, her boyut boyunca küçük hareketleri desteklemek amacıyla adım boyutu üst sınırını problemin üst sınırı ile alt sınırı farkının 0,2 değeri ile çarpılması sonucu elde edilen değer olarak belirlenir. Daha iyi bir çözüm araştırmak için en iyi bireyin ilgili boyutunda kendi adım büyüklüğü kadar ilerisindeki nokta değerlendirilir. Daha iyi bir sonuca ulaşırsa, daha iyi bir sonuç üretmeyene veya üst sınıra ulaşana kadar başka bir adım boyutu daha ekleyerek arama bu yönde devam eder. Başka bir deyişle, sonuç başarılı olduğu sürece adım boyutu doğrusal olarak artar. Daha sonra aynı süreç son başarılı sonuca ulaştığı noktadan geriye doğru işletilir. Bu yöntem hakkında daha detaylı bilgi (Hadi ve ark., 2019)'den alınabilir.

Karşılaştırma amacıyla seçilen üç yerel arama yönteminin benzerlik ve farkları aşağıda özetlenmiştir.

Benzerlikler:

- ✓ Yerel arama yöntemini popülasyondaki en yüksek uygunluk değerine sahip olan en iyi birey üzerine uygulamak
- ✓ Birinci boyuttan sonuncu boyuta kadar her bir boyut üzerinde tek tek süreci işletmek
- ✓ Yeni çözüm ararken mevcut boyutun zıt yönlerini değerlendirmek

Farklar:

- ✓ Her boyut için adım büyüklüklerini tutan diziye başlangıç değerleri atarken her yöntemin farklı bir yol izlemesi
- ✓ Adım büyüklüğünü değiştirirken her yöntemin farklı katsayılar kullanması
- ✓ GRGLS iyi bir çözüm bulduğunda adım büyüklüğünü katlanarak (doğrusal olmayan) değiştirirken diğer yöntemlerin adım büyüklüğünü ya aynı bırakması ya da doğrusal olarak arttırması

Deneysel çalışmalar Matlab kullanılarak yapılmıştır. Çerçeve algoritma olarak seçilen MSHADE-SPA algoritmasının orijinal halinin Matlab uygulama kodlarına yazarlarının internet sitesinden (Mohamed, 2023), bu tez kapsamında önerilen GRGLS yerel arama metodu, IMLSHADE-SPA çerçeve algoritması ve MTSLS1 kaynak kodlarına ise (UYMAZ, 2023) den ulaşılabilir.

Karşılaştırmalarda temel alınan ana algoritma MLSHADE-SPA, CEC2018 yılı yarışma ikincisi olduğundan adil bir karşılaştırma yapabilmek amacıyla tüm deneylerde, CEC özel oturumları ve yarışmaları için önerilen kural ve uygulama talimatlarına uyulmuştur. Algoritma bünyesinde popülasyon boyutu 250, minimum popülasyon boyutu 20, iterasyon sayısı ise 50 olarak alınmıştır. Karşılaştırılan sonuçlar, her test fonksiyonu için 25 bağımsız çalıştırmanın ortalamalarıdır. CEC yarışma yönergeleri uyarınca maksimum uygunluk değerlendirmesi (maxFEs) $3.0e6$ olarak alınırken gerçekleştirilen tüm deneysel çalışmalarda FEs = $1.2e5$, $6.0e5$ ve $3.0e6$ olduğunda sonuçlar ayrıca kaydedilmiştir. Bu sonuçların kaydedilmesi önerilen yöntemin sadece sonuçta değil süreçte de performansını değerlendirme fırsatı sunmaktadır. Elde edilen FEs = $1.2e5$ sonuçları Çizelge 5.6'de, FEs = $6.0e5$ sonuçları Çizelge 5.8'de ve FEs = $3.0e6$ sonuçları ise Çizelge 5.10'de listelenmiştir.

Her çalışmanın sonunda kaydedilen en iyi değerler kullanılarak istatistiksel analiz için Wilcoxon işaretli sıralama testi yapılmıştır. GRGLS ve diğer iki yerel arama yöntemi arasındaki bu istatistiksel analizlerin sonuçları, karşılaştırma tablolarının hemen ardında bulunan Çizelge 5.7'de, Çizelge 5.9 ve Çizelge 5.11'de gösterilmiştir.

Çizelge 5.6. 1.2e5 FEs için yerel arama yöntemleri karşılaştırma sonuçları

	Yerel Arama Metotları			Sıralamalar		
	MTSLS1	MMTS	GRGLS	MTSLS1	MMTS	GRGLS
F1	2.50E+03	5.72E+07	7.05E+07	1	2	3
F2	1.34E+03	1.81E+03	2.14E+04	1	2	3
F3	8.85E-03	3.59E+00	1.97E+01	1	2	3
F4	3.33E+11	6.77E+11	2.27E+11	2	3	1
F5	1.04E+07	1.06E+07	1.12E+07	1	2	3
F6	9.69E+05	9.90E+05	9.69E+05	1.5	3	1.5
F7	5.23E+09	7.51E+09	4.99E+09	2	3	1
F8	1.12E+16	3.14E+16	8.48E+15	2	3	1
F9	7.82E+08	8.51E+08	9.27E+08	1	2	3
F10	2.00E+07	3.47E+07	5.21E+07	1	2	3
F11	1.02E+12	9.91E+11	4.72E+11	3	2	1
F12	4.27E+03	1.47E+07	4.85E+07	1	2	3
F13	4.33E+10	4.16E+10	1.87E+10	3	2	1
F14	8.45E+11	1.01E+12	5.66E+11	2	3	1
F15	2.57E+09	6.38E+09	4.66E+08	2	3	1
			Ortalama Sıralamalar	1.63	2.40	1.96

Çizelge 5.6 göstermektedir ki 15 fonksiyondan 8 tanesinde en iyi sonucu alan yöntem GRGLS'dir. Sonuçlar bu aşamada, önerilen yöntemin genel sıralamada üç yöntem arasında ikinci olarak tam ayrılabilir fonksiyonlarda diğer yöntemlerden daha düşük performansa sahip olduğunu göstermektedir.

Çizelge 5.7. 1.2e5 FEs için Wilcoxon işaretli sıralama testi sonuçları ($\alpha=0.05$)

Fonk.	GRGLS ile MTSLS1			GRGLS ile MMTS		
	p-Value	T	W	p-Value	T	W
F1	1.23E-05	325	-	6.96E-01	177	=
F2	1.23E-05	325	-	1.23E-05	325	-
F3	1.23E-05	325	-	1.23E-05	325	-
F4	3.96E-02	86	+	2.26E-05	5	+
F5	1.35E-01	218	=	1.22E-01	220	=
F6	6.57E-01	146	=	3.51E-03	54	+
F7	1.92E-01	114	=	1.38E-02	71	+
F8	3.47E-02	84	+	1.57E-05	2	+
F9	3.51E-03	271	-	6.53E-02	231	=
F10	1.23E-05	325	-	4.07E-05	315	-
F11	3.82E-03	55	+	1.89E-03	47	+
F12	1.23E-05	325	-	1.23E-05	325	-
F13	1.23E-05	0	+	1.23E-05	0	+
F14	4.93E-04	33	+	1.13E-04	19	+
F15	1.23E-05	0	+	1.39E-05	1	+
		+/=-	6/3/6		+/=-	8/3/4

Çizelge 5.8. 6.0e5 FEs için yerel arama yöntemleri karşılaştırma sonuçları

	Yerel Arama Metotları			Sıralamalar		
	MTSLS1	MMTS	GRGLS	MTSLS1	MMTS	GRGLS
F1	5.11E+02	3.36E+00	1.09E+01	3	1	2
F2	1.32E+03	2.26E+02	7.68E+03	2	1	3
F3	3.45E-03	1.52E-03	6.56E+00	2	1	3
F4	2.80E+10	2.54E+10	2.41E+10	3	2	1
F5	2.73E+06	4.50E+06	3.46E+06	1	3	2
F6	2.47E+03	4.48E+03	5.56E+03	1	2	3
F7	6.21E+08	8.08E+08	3.63E+08	2	3	1
F8	2.78E+14	2.98E+14	1.75E+14	2	3	1
F9	2.36E+08	3.81E+08	3.23E+08	1	3	2
F10	5.66E+03	7.04E+03	4.20E+03	2	3	1
F11	8.70E+09	1.17E+10	4.49E+09	2	3	1
F12	2.93E+03	3.16E+03	2.41E+03	2	3	1
F13	1.57E+10	1.76E+10	4.79E+09	2	3	1
F14	1.09E+11	1.69E+11	9.78E+10	2	3	1
F15	2.53E+07	7.20E+07	1.56E+07	2	3	1
		Ortalama Sıralamalar		1.93	2.47	1.60

Çizelge 5.8, GRGLS'nin 15 fonksiyonun 9'unda en iyi sonucu alan yöntem olarak tüm fonksiyonlarda kendini geliştirdiği göstermektedir. Çok modlu olan ve iki farklı fonksiyon grubunun üyesi olan F10 ve F12 fonksiyonlarının gelişimi dikkat çekicidir. GRGLS bu aşamada genel sıralamada üç yöntem arasında birinci sıradadır.

Çizelge 5.9. 6.0e5 FEs için Wilcoxon işaretli sıralama testi sonuçları ($\alpha=0.05$)

Fonk.	GRGLS ile MTSLS1			GRGLS ile MMTS		
	p-Value	T	W	p-Value	T	W
F1	1.23E-05	0	+	6.85E-03	62	-
F2	1.23E-05	325	-	1.23E-05	325	-
F3	1.23E-05	325	-	1.23E-05	325	-
F4	1.04E-01	102	=	4.59E-01	135	=
F5	8.71E-03	260	-	2.26E-03	49	+
F6	1.57E-03	280	-	2.21E-01	208	=
F7	2.70E-03	51	+	4.57E-05	11	+
F8	6.02E-04	35	+	2.66E-04	27	+
F9	1.77E-05	322	-	1.73E-02	74	+
F10	1.58E-01	110	=	2.14E-02	77	+
F11	3.22E-03	53	+	4.07E-05	10	+
F12	9.68E-01	161	=	2.70E-03	51	+
F13	1.23E-05	0	+	1.23E-05	0	+
F14	3.39E-01	127	=	3.28E-04	29	+
F15	4.03E-04	31	+	1.23E-05	0	+
		+ / = / -	6/4/5		+ / = / -	10/2/3

Çizelge 5.10. 3.0e6 FEs için yerel arama yöntemleri karşılaştırma sonuçları

	Yerel Arama Metotları			Sıralamalar		
	MTSLS1	MMTS	GRGLS	MTSLS1	MMTS	GRGLS
F1	3.24E-07	2.60E-22	4.97E-11	3	1	2
F2	1.18E+03	8.39E+01	4.65E+03	2	1	3
F3	2.30E-08	9.96E-14	2.28E+00	2	1	3
F4	8.88E+08	7.74E+08	3.78E+08	3	2	1
F5	1.93E+06	1.91E+06	1.75E+06	3	2	1
F6	1.53E+03	2.59E+03	3.67E+03	1	2	3
F7	4.05E+05	4.76E+04	4.90E+03	3	2	1
F8	1.21E+13	1.23E+13	5.40E+12	2	3	1
F9	1.48E+08	1.51E+08	1.63E+08	1	2	3
F10	5.23E+02	6.16E+02	4.95E+02	2	3	1
F11	4.36E+07	3.51E+07	2.00E+06	3	2	1
F12	3.18E+02	1.41E+02	1.08E+02	3	2	1
F13	3.76E+08	8.29E+07	3.05E+06	3	2	1
F14	1.42E+07	1.44E+07	5.44E+06	2	3	1
F15	5.31E+06	2.62E+07	1.23E+06	2	3	1
			Ortalama Sıralamalar	2.33	2.07	1.60

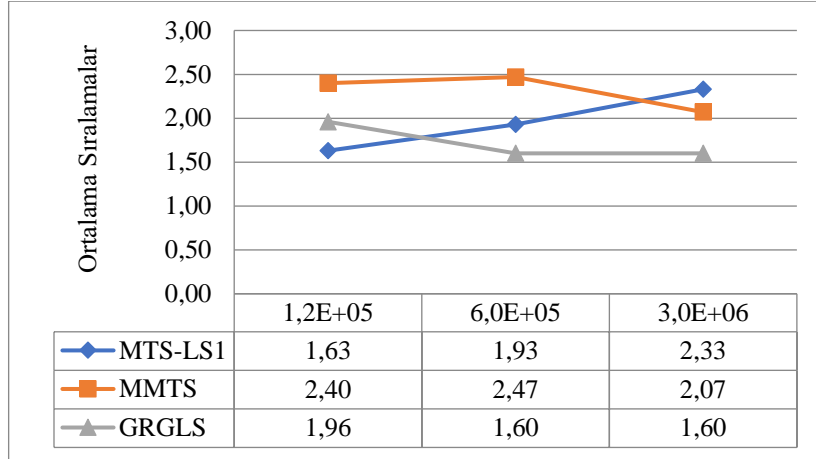
Çizelge 5.10, GRGLS'nin 15 fonksiyonun 10'unda en iyi sonucu alan yöntem olarak tüm sürecin sonunda üç yöntem arasında birinci olduğunu göstermektedir.

Çizelge 5.11. 3.0e6 FEs için Wilcoxon işaretli sıralama testi sonuçları ($\alpha=0.05$)

Fonk.	GRGLS ile MTSLS1			GRGLS ile MMTS		
	p-Value	T	W	p-Value	T	W
F1	1.23E-05	0	+	1.23E-05	325	-
F2	1.23E-05	325	-	1.23E-05	325	-
F3	1.23E-05	325	-	1.23E-05	325	-
F4	4.46E-04	32	+	2.66E-04	27	+
F5	7.36E-02	96	=	8.27E-02	98	=
F6	1.19E-02	256	-	7.80E-02	228	=
F7	1.23E-05	0	+	1.23E-05	0	+
F8	2.96E-04	28	+	7.33E-04	37	+
F9	1.60E-02	252	-	3.47E-02	241	-
F10	7.16E-01	176	=	2.11E-01	116	=
F11	1.39E-05	1	+	1.39E-05	1	+
F12	7.33E-04	37	+	3.67E-01	129	=
F13	1.23E-05	0	+	1.23E-05	0	+
F14	1.23E-05	0	+	1.23E-05	0	+
F15	1.23E-05	0	+	1.23E-05	0	+
		+ / = / -	9/2/4		+ / = / -	7/4/4

Metasezgisel algoritmalar için parametrik olmayan istatistiksel testlerin kullanılması önerilmektedir. Bunun nedeni küçük boyutlu sonuç örnekleri üzerinde kullanılabilirliği ve parametrik testlerden daha az kısıtlayıcı olmalarıdır (García ve ark., 2009). Üç yerel arama yöntemiyle gerçekleştirilen test sonuçlarının karşılaştırılması ve yorumlanmasında ikili karşılaştırmalar için kullanılan Wilcoxon işaretli sıra testi uygulanmıştır. Büyük ölçekli optimizasyon problemlerinin çözümünde kilometre taşı (milestone) olarak da anılan 1.2e5, 6.0e5, 3.0e6 dönüm noktalarında elde edilen sonuçlarda önerilen yöntem ve diğer yerel arama yöntemlerinden her biri ayrı olarak ikili ve problem temelli kıyaslanmaktadır. Algoritmaların problem temelli ikili karşılaştırması için 25 çalışma sonucunda elde edilen global minimum değerler kullanılarak yapılan istatistik test ile iki yöntemin elde ettiği sonuçlar arasında anlamlı bir fark bulunup bulunmadığı tespit edilmeye çalışılmıştır. Tablolardaki W sütununda yer alan +, =, - sembolleri algoritmalar arasındaki bu farkları temsil etmektedir. = sembolü ise algoritmalar arasında anlamlı bir fark olmadığını, + sembolü, yöntemin karşılaştırma yapılan diğer yöntemle göre anlamlı bir gelişme gösterdiğini, - sembolü ise yöntemin karşılaştırma yapılan diğer yöntemden anlamlı bir şekilde geride kaldığını göstermektedir. Çizelge 5.7, Çizelge 5.9 ve Çizelge 5.11'nin son satırları, ikili karşılaştırmadaki üç istatistiksel anlamlılık durumu için (+/=/-) formatında toplam sayılarını göstermektedir. Wilcoxon işaretli sıra testi $\alpha = 0.05$ anlamlılık düzeyinde kullanılmıştır. Sonuçlar incelendiğinde GRGLS'nin karşılaştırılan diğer yöntemlere göre istatistiksel olarak daha iyi sonuçlar elde edebildiği anlaşılmaktadır. Özellikle GRGLS'nin avantajı FEs arttıkça daha iyi gözlemlenmektedir.

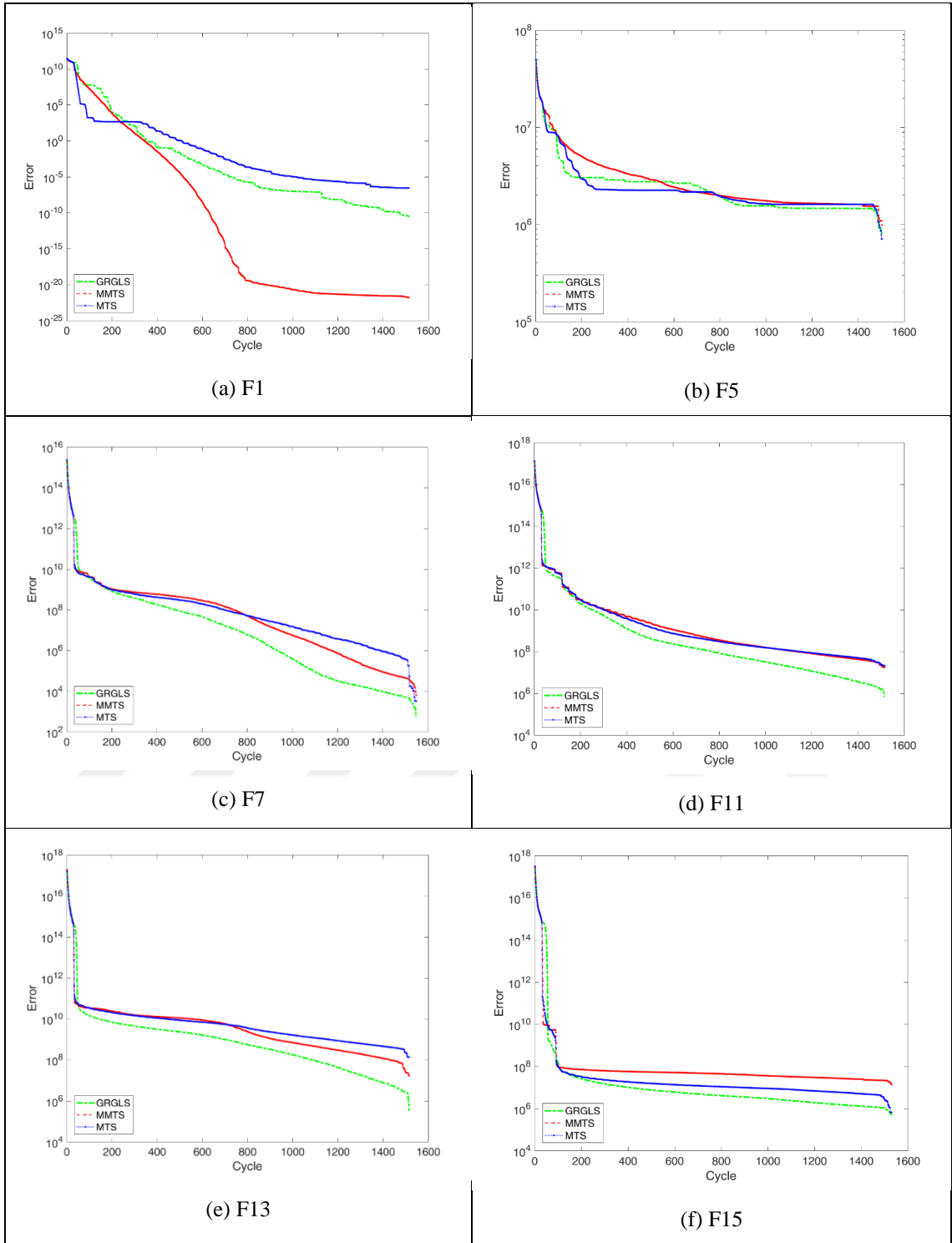
Çizelge 5.6'de, Çizelge 5.8 ve Çizelge 5.10'de belirtilen sıralama ortalamalarına göre yerel arama metodlarının genel performansını temsil eden grafik Şekil 5.2'de gösterilmiştir. Sıralama ortalamalarındaki en düşük değer, en başarılı algoritmayı temsil eder. Şekil 5.2, önerilen yöntem GRGLS'nin diğerlerinden daha başarılı olduğunu göstermektedir. MTSLS1, 1.2e5 uygunluk değerlendirmesinde en iyi yöntem olmasına rağmen, 6.0e5 uygunluk değerlendirmesinden sonra GRGLS ilk sırayı almış ve 3.0e6 uygunluk değerlendirmesinde 15 fonksiyondan 10'unda en iyi sonucu elde ederek birincilik başarısını korumuştur.



Şekil 5.2. Yerel arama metotları ortalama sıralamalar performans grafiği

GRGLS, MTSLS1, MMTS'nin farklı fonksiyon gruplarından seçilen altı problem (F1, F5, F7, F11, F13, F15) için yakınsama eğrileri Şekil 5.3'te sunulmuştur. Y eksenleri ilgili fonksiyonun 25 bağımsız çalışmasının ortalama değerlerini temsil eder. X eksenleri ise her 2000 uygunluk değerlendirmesinde kaydedilen bir noktayı temsil eder. Örneğin x eksenindeki 300 değeri 6.0e5 uygunluk değerlendirmesine karşılık gelmektedir.

Şekil 5.3'te gösterilen yakınsama eğrileri GRGLS'nin diğer yerel arama yöntemlerine kıyasla daha başarılı olduğunu doğrulamaktadır. Üç yöntem, yaklaşık 1.5e6 uygunluk değerlendirmelerine kadar tam ayrılabilir fonksiyonlar dışında benzer sonuçlar vermektedir. Bu noktadan itibaren yöntemlerin performansları birbirinden farklılık göstermektedir. Önerilen yöntem, baştan sona güçlü bir yakınsama karakteristiği çizmektedir. Dolayısıyla GRGLS'nin diğer yerel arama yöntemlerine göre daha başarılı olduğu söylenebilir.



Şekil 5.3. Örnek olarak seçilen CEC2013 fonksiyonları için yerel arama yöntemlerinin yakınsama eğrileri

İkinci deneysel çalışmada ise MLSHADE-SPA çerçevesi ile önerilen yerel arama yönteminin (GRGLS) birleşiminden oluşan Geliştirilmiş MLSHADE-SPA (IMLSHADE-SPA) çerçeve algoritması, özet bilgi ve referansları Çizelge 5.12’te listelenen dokuz algoritma ile karşılaştırılmıştır. Deneysel çalışmada, temel özellikleri Çizelge 3.4’te listelenen CEC2013 test fonksiyon kümesi (Li ve ark., 2013) kullanılmıştır.

Matlab’ta kodlanan IMLSHADE_SPA algoritması her fonksiyon için CEC özel oturumları ve yarışmaları için önerilen talimatlar gereğince MaxFEs 3.0e6 alınarak 25 kere çalıştırılma ortalamasından elde edilmiştir. Diğer algoritmaların sonuçları Çizelge 5.12’te belirtilen referanslardan direk olarak alınmıştır. Elde edilen tüm sonuçlar Çizelge 5.13’te listelenmiştir.

Çizelge 5.12. IMLSHADE-SPA ile karşılaştırılan algoritmalar

Algoritma	Tanım	Referans
LSGOjDE	LSGO problemlerin çözümü için tasarlanmış popülasyon boyutunu azaltmalı kendi kendini uyarlayan jDE algoritması	(Maučec ve ark., 2018)
CBCC3-DG2	Diferansiyel gruplamanın geliştirilmiş versiyonunu kullanan katkıya dayalı işbirlikçi birlikte evrim algoritması	(Omidvar ve ark., 2017)
CRO-LS	Substrat katmanları ve yerel arama içeren bir mercan resifleri optimizasyon algoritması	(Salcedo-Sanz ve ark., 2016)
CCFR-IDG2	Verimli kaynak tahsisi için geliştirilmiş diferansiyel gruplama kullanan iş birliğine dayalı birlikte evrim çerçevesi	(Yang ve ark., 2016)
IHDELS	Diferansiyel Gelişim Algoritmasının yerel aramalı yinelemeli hibridizasyonu	(Molina ve Herrera, 2015)
VMO-DE	Değişken Ağ Optimizasyonu (VMO) ve Diferansiyel Gelişim (DE) içeren hibrit bir algoritma	(López ve ark., 2015)
VGDE	Yardımcı bir fonksiyon kullanan Değişken Gruplamaya Dayalı Diferansiyel Gelişim Algoritması	(Wei ve ark., 2014)
MOS	Çoklu Yavru Örnekleme, farklı metasezgisel yöntemlerin ve yerel aramaların bir HRH (Yüksek Düzeyli Aktarma Hibrit) yaklaşımıyla birleştirilmesine izin veren bir çerçevedir.	(LaTorre ve ark., 2013)
MA-SW-Chains	Yerel arama algoritması Solis & Wets ile genetik algoritmayı birleştiren memetik bir algoritma	(Molina ve ark., 2010b)

Çizelge 5.13. IMLSHADE-SPA algoritmasının diğer LSGO algoritmalarıyla karşılaştırma sonuçları

	MA-SW-Chains	MOS	VGDE	VMO-DE	IHDELS	CRO-LS	CBCC3-DG2	CCFR-IDG2	LSGOjDE	IMLSHADE-SPA
F1	1.14E-12	0.00E+00	0.00E+00	1.29E-03	4.34E-28	1.84E+06	8.65E+05	2.00E-05	5.64E-03	4.97E-11
F2	1.18E+03	8.32E+02	4.56E+01	5.53E+03	1.32E+03	9.84E+02	1.41E+04	3.60E+02	5.77E+00	4.65E+03
F3	6.78E-13	9.17E-13	3.98E-13	3.70E-04	2.01E+01	2.01E+01	2.06E+01	2.10E+01	2.00E+01	2.28E+00
F4	3.80E+09	1.74E+08	5.96E+08	9.13E+09	3.04E+08	1.55E+10	3.39E+07	9.60E+07	6.71E+08	3.78E+08
F5	2.26E+06	6.94E+06	3.00E+06	7.28E+14	9.59E+06	2.38E+07	2.14E+06	2.80E+06	1.72E+06	1.75E+06
F6	1.07E+04	1.48E+05	1.31E+05	2.15E+05	1.03E+06	1.06E+06	1.05E+06	1.10E+06	1.05E+06	3.67E+03
F7	3.78E+06	1.62E+04	1.85E+03	3.43E+06	3.46E+04	2.78E+08	2.95E+07	2.00E+07	6.18E+05	4.90E+03
F8	4.63E+13	8.00E+12	7.00E+14	6.94E+13	1.36E+12	4.56E+14	6.74E+10	7.00E+10	3.00E+12	5.40E+12
F9	1.14E+08	3.83E+08	2.31E+08	7.68E+08	6.74E+08	5.27E+08	1.70E+08	1.90E+08	3.14E+08	1.63E+08
F10	3.66E+04	9.02E+05	1.57E+02	9.10E+06	9.16E+07	9.44E+07	9.28E+07	9.50E+07	9.24E+07	4.95E+02
F11	2.10E+08	5.22E+07	7.52E+07	1.66E+08	1.07E+07	2.91E+10	7.70E+08	4.00E+08	9.27E+11	2.00E+06
F12	1.23E+03	2.47E+02	2.52E+03	4.45E+03	3.77E+02	3.69E+03	5.81E+07	1.60E+09	1.88E+03	1.08E+02
F13	1.98E+07	3.40E+06	1.36E+09	2.46E+07	3.80E+06	5.33E+09	6.03E+08	1.20E+09	1.67E+07	3.05E+06
F14	1.45E+08	2.56E+07	2.29E+10	9.54E+07	1.58E+07	6.08E+10	1.11E+09	3.40E+09	4.37E+07	5.44E+06
F15	5.90E+06	2.35E+06	3.44E+06	1.10E+07	2.81E+06	1.88E+07	7.11E+06	9.80E+06	2.87E+06	1.23E+06

Her bir algoritmanın ilgili fonksiyon için tüm algoritmalar arasındaki başarı sırası Çizelge 5.14'te listelenmiştir.

Çizelge 5.14. Algoritmaların CEC 2013 fonksiyonlarındaki başarı sıralamaları

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
MA-SW-Chains	4	6	2	8	4	2	7	7	1	3	6	4	5	6	6
MOS	1.5	4	3	3	7	4	3	6	7	4	3	2	2	3	2
VGDE	1.5	2	1	6	6	3	1	10	5	1	4	6	9	9	5
VMO-DE	7	9	4	9	10	5	6	8	10	5	5	8	6	5	9
IHDELS	3	7	7.5	4	8	6	4	3	9	6	2	3	3	2	3
CRO-LS	10	5	7.5	10	9	9	10	9	8	9	9	7	10	10	10
CBCC3-DG2	9	10	9	1	3	7.5	9	1	3	8	8	9	7	7	7
CCFR-IDG2	6	3	10	2	5	10	8	2	4	10	7	10	8	8	8
LSGOjDE	8	1	6	7	1	7.5	5	4	6	7	10	5	4	4	4
IMLSHADE-SPA	5	8	5	5	2	1	2	5	2	2	1	1	1	1	1

Çizelge 5.13'teki fonksiyonların en iyi sonuçları ve Çizelge 5.14'teki sıralama ortalaması verileri dikkate alınarak aşağıdaki sonuçlara ulaşılmıştır:

- ✓ Tamamen ayrılabilir fonksiyonlarda (F1, F2, F3) en iyi sonuçlar sırasıyla MOS, VGDE, LSGOjDE, VGDE algoritmaları tarafından elde edilmiştir. Çizelge 5.14'teki ortalama sıralamalar fonksiyon grubu olarak değerlendirildiğinde tam ayrılabilir fonksiyonlarda en başarılı algoritma VGDE'dir.
- ✓ Ayrılabilir bir alt bileşeni bulunan fonksiyonlar (F4, F5, F6, F7) ve ayrılabilir bir alt bileşeni olmayan fonksiyonlar (F8, F9, F10, F11) söz konusu olduğunda her fonksiyon için farklı algoritmalar en iyi sonuçları elde etmiştir ancak ortalama sıralamalar değerlendirildiğinde, sonuçlar gösteriyor ki IMLSHADE-SPA, her iki fonksiyon grubunda da en başarılı algoritmadır.
- ✓ Örtüşen fonksiyonlar (F12, F13, F14) ve ayrılabilir fonksiyon olan F15'te en iyi sonuçların tamamı IMLSHADE-SPA tarafından elde edilmiştir ve bu nedenle IMLSHADE-SPA bu fonksiyon grupları için en başarılı algoritmadır.
- ✓ 15 fonksiyondaki tüm algoritmalar arasında yapılan bu karşılaştırma sonucunda önerilen yöntem IMLSHADE-SPA birinci sırayı alırken, ikinci algoritma MOS ve üçüncü algoritma VGDE'dir.

Formula 1 Puanı (Formula One Score), formula 1 yarış puanlama sisteminden ilham alan bir metriktir. Yöntem uyarınca, algoritmalar performansa göre en iyiden en kötüye doğru sıralanır. Daha sonra en iyi 10 algoritma sırasıyla 25, 18, 15, 12, 10, 8, 6, 4, 2 ve 1 puan alır. Algoritmaların geri kalanına sıfır verilir. Algoritmanın Formula 1 puanı, her işlev için kazanılan puanların toplanmasıyla hesaplanır. Tüm puanlar hesaplandıktan sonra en çok puan kazandıran algoritma en iyi algoritma olarak seçilir. Bu metrik LSGO algoritmalarının performansını karşılaştırmak için de kullanılmakta olup bu çalışmada da aynı amaçla kullanılmıştır (Hadi ve ark., 2019; Maučec ve Brest, 2019).

Her bir algoritmanın CEC 2013 test fonksiyon kümesi fonksiyonlarındaki başarılarını gösteren sıralama ve Formula 1 Puanı olmak üzere iki farklı metrikteki algoritma puanları Çizelge 5.15'te listelenmiştir. Her bir metrik için algoritmalar en iyiden en kötüye doğru sıralanmıştır. Sonuçlar IMLSHADE-SPA algoritmasının her iki metriğe göre de on algoritma arasında en iyi algoritma olduğunu kanıtlamaktadır.

Çizelge 5.15. Algoritmaların sıralama ve formula 1 skorları

Algoritma	DereceOrt	Yer	Formula1 Skoru	Yer
MA-SW-Chains	4.73	5	173	4
MOS	3.63	2	210	2
VGDE	4.63	3	194	3
VMO-DE	7.07	9	92	9
IHDELS	4.70	4	172	5
CRO-LS	8.83	10	45	10
CBCC3-DG2	6.57	7	125	7
CCFR-IDG2	6.73	8	111	8
LSGOjDE	5.30	6	159	6
IMLSHADE-SPA	2.80	1	268	1

Deneysel sonuçlara göre, önerilen yeni yerel arama yöntemi GRGLS, genel olarak tüm fonksiyonlarda iyi sonuçlar vermektedir ve GRGLS'yi yerel arama yöntemi olarak kullanan bir memetik çerçeve, karşılaştırılan algoritmalarından önemli ölçüde daha iyi performans göstermektedir. Önerilen yöntem tüm fonksiyonlarda iyi sonuçlar elde ederken, örtüşen ve ayrılamaz fonksiyonlarda diğerlerine göre üstün performans göstermektedir.

Sonuç olarak, deneysel çalışmalar, testler ve analizler, önerilen GRGLS'nin LSGO araştırma alanı için etkili ve verimli bir yerel arama yöntemi olarak kullanılabileceğini göstermiştir.

5.4. CGRGLS ve CIMLSHADE-SPA

Bu bölümde yapılan deneysel çalışmalar, önerilen Altın Oran Kılavuzlu Yerel Arama yöntemini kaos teorisinden ilham alarak daha da geliştirmek ve geliştirilen yeni versiyonu LSGO performansını ölçen klasik test fonksiyon kümeleri yerine gerçek dünya problemini temsilen Büyük Veri Optimizasyon Problemleri çözümünde test etmek amacıyla gerçekleştirilmiştir. Deneysel çalışmalar sonucunda geliştirilen bu yeni versiyon Kaotik Altın Oran Rehberli Yerel Arama (CGRGLS) olarak adlandırılmıştır

Çok boyutlu optimizasyon problemlerinde geniş arama uzayı, verimli arama stratejileri gerektirmektedir. CGRGLS arama yönteminde arama verimliliği artırmak için kaotik haritaların tekrarsız rastgele sayı üretme özelliği kullanılmıştır. Yapılan çalışmalar ve karşılaştırmalar, kaotik haritalama ile elde edilen değer, algoritmanın en önemli faktörlerinden biri olan algoritmanın adım büyüklüğünü değiştirme kısmında katsayı olarak kullanılmasının, algoritmanın etkinliğini artıran verimli bir güncelleme olduğunu göstermiştir. Özellikle problem boyutu arttıkça yeni versiyon CGRGLS yerel arama yönteminin başarısı da artmaktadır.

Bu kapsamda gerçekleştirilen deneysel çalışmaların tamamı Bölüm 3.5.3'te açıklanarak Çizelge 3.5'te fonksiyonları listelenen Big-Opt test fonksiyon kümesi üzerinde gerçekleştirilmiştir.

Yapılan deneysel çalışmalar şu şekilde sıralanabilir:

1) Literatürde iyi bilinen 10 farklı kaotik haritayı karşılaştırarak yöntem bünyesinde kullanılacak en başarılı kaotik harita seçildi.

2) MMTS, bir önceki versiyon GRGLS ve geliştirilen yeni versiyon CGRGLS yerel arama yöntemlerinin aynı memetik çerçeve bünyesinde aynı koşullar ve kurallar altında ayrı ayrı çalıştırılmasıyla elde edilen sonuçlar karşılaştırıldı.

3) Ana algoritma olarak kullanılan MLSHADE-SPA algoritmasının yerel arama kısmında geliştirilen yeni versiyon CGRGLS yerel aramasının kullanılmasıyla önerilen CIMLSHADE-SPA memetik çerçevesinin performansı test edildi.

4) Önerilen yöntemin başarısı, CEC LSGO yarışmalarının kazanan algoritmaları ile karşılaştırılarak doğrulandı.

5.4.1. Kaotik Harita Seçimi

Kaotik haritalarla ilgili yayınlar incelendiğinde hangi kaotik haritanın en iyi performansı göstereceğini belirlemek kolay değildir. Bu nedenle literatürde iyi bilinen ve Çizelge 3.2’de listelenen 10 farklı kaotik harita, algoritma bünyesinde ayrı ayrı test edilmiş ve yöntemin performansını artıracak en uygun kaotik harita belirlenmeye çalışılmıştır.

Deneysel sonuçlar, 25 bağımsız çalışmanın ortalamasından elde edilmiştir. Durdurma kriteri olarak 1024 değişkenli D4 ve D4N için 50.000 FEs (Fitness Evolution), 3072 değişkenli D12 ve D12N için 150.000 FEs, 4864 değişkenli D19 ve D19N için 250.000 FEs kullanılmıştır. Her bir harita kullanılarak gerçekleştirilen testlerin sonuçlarını Çizelge 5.16’de ayrı ayrı listelenmektedir. Sonuçlar std, ortalama ve en iyi değerler olarak gösterilmekle birlikte her fonksiyondan elde edilen en iyi ortalama değer kalın yazı tipiyle işaretlenmiştir. Tablonun alt kısmında genel toplam sıralama değerleri yer almaktadır.

Çizelge 5.16’deki sonuçlar, Harita8 kullanan algoritmanın on harita arasında en iyi sonuçları elde ettiğini göstermektedir. 6 fonksiyondan 4’ünün en iyi değeri aldığı yani fonksiyonların %67’sinde Harita8’in başarılı olduğu görülmektedir. Harita8’den sonra en başarılı haritalar, tüm veri kümesinin yalnızca birinde en iyi değere sahip olan Harita2 ve Harita10’dur. Elde edilen sonuçlar nedeniyle önerilen yerel arama yöntemi adım büyüklüğü değişiminde katsayı olarak Singer kaotik haritası (Harita8) tarafından üretilen sayının çarpan olarak kullanılmasına karar verilmiştir.

Çizelge 5.16. Kaotik haritaların performans karşılaştırma sonuçları

		Harita1	Harita2	Harita3	Harita4	Harita5	Harita6	Harita7	Harita8	Harita9	Harita10
D4	En İyi	6.895E-02	6.973E-02	6.824E-02	6.814E-02	6.953E-02	6.960E-02	6.795E-02	6.831E-02	6.741E-02	6.901E-02
	Ortalama	7.315E-02	7.492E-02	7.208E-02	7.310E-02	7.209E-02	7.333E-02	7.299E-02	7.168E-02	7.261E-02	7.358E-02
	Std.	2.980E-03	7.918E-03	3.018E-03	3.975E-03	1.500E-03	3.939E-03	3.530E-03	2.413E-03	3.553E-03	5.764E-03
	Sıralama	7	10	2	6	3	8	5	1	4	9
D4N	En İyi	6.682E-02	6.647E-02	6.648E-02	6.697E-02	6.578E-02	6.682E-02	6.642E-02	6.675E-02	6.592E-02	6.727E-02
	Ortalama	7.066E-02	7.099E-02	7.088E-02	7.105E-02	7.135E-02	7.233E-02	7.130E-02	6.948E-02	6.961E-02	7.058E-02
	Std.	3.488E-03	3.645E-03	3.952E-03	4.380E-03	3.761E-03	4.065E-03	3.413E-03	2.373E-03	2.267E-03	3.021E-03
	Sıralama	4	6	5	7	9	10	8	1	2	3
D12	En İyi	3.601E-02	3.776E-02	4.020E-02	4.094E-02	3.569E-02	4.632E-02	3.589E-02	3.408E-02	3.601E-02	3.924E-02
	Ortalama	5.392E-02	5.544E-02	5.908E-02	5.736E-02	5.959E-02	5.713E-02	5.341E-02	5.287E-02	5.749E-02	5.836E-02
	Std.	1.025E-02	1.010E-02	1.373E-02	1.005E-02	1.061E-02	7.781E-03	8.253E-03	1.009E-02	1.344E-02	9.868E-03
	Sıralama	3	4	9	6	10	5	2	1	7	8
D12N	En İyi	4.278E-02	4.344E-02	3.867E-02	4.285E-02	4.395E-02	4.151E-02	4.170E-02	3.417E-02	4.111E-02	3.878E-02
	Ortalama	5.658E-02	5.696E-02	5.451E-02	5.694E-02	5.855E-02	5.943E-02	5.507E-02	5.606E-02	5.653E-02	5.280E-02
	Std.	8.928E-03	9.255E-03	8.064E-03	1.066E-02	1.130E-02	9.942E-03	8.839E-03	1.161E-02	1.021E-02	1.046E-02
	Sıralama	6	8	2	7	9	10	3	4	5	1
D19	En İyi	4.844E-02	5.093E-02	4.790E-02	4.830E-02	4.800E-02	4.451E-02	4.846E-02	4.743E-02	4.793E-02	4.987E-02
	Ortalama	6.314E-02	6.223E-02	6.276E-02	6.293E-02	6.384E-02	6.223E-02	6.238E-02	6.298E-02	6.256E-02	6.277E-02
	Std.	8.622E-03	9.358E-03	8.925E-03	8.969E-03	9.451E-03	9.304E-03	7.469E-03	1.086E-02	9.130E-03	9.808E-03
	Sıralama	9	1	5	7	10	2	3	8	4	6
D19N	En İyi	4.642E-02	4.412E-02	5.230E-02	4.612E-02	4.992E-02	4.738E-02	4.700E-02	4.587E-02	4.678E-02	4.755E-02
	Ortalama	6.340E-02	6.210E-02	6.333E-02	6.228E-02	6.219E-02	6.215E-02	6.221E-02	6.046E-02	6.178E-02	6.169E-02
	Std.	7.960E-03	9.513E-03	8.579E-03	9.655E-03	8.541E-03	9.583E-03	9.081E-03	8.496E-03	8.875E-03	9.267E-03
	Sıralama	10	4	9	8	6	5	7	1	3	2
En İyi Sayısı		0	1	0	0	0	0	0	4	0	1
Sıralama Toplamı		39	33	32	41	47	40	28	16	25	29

5.4.2. Yerel Arama Metotlarının Karşılaştırılması

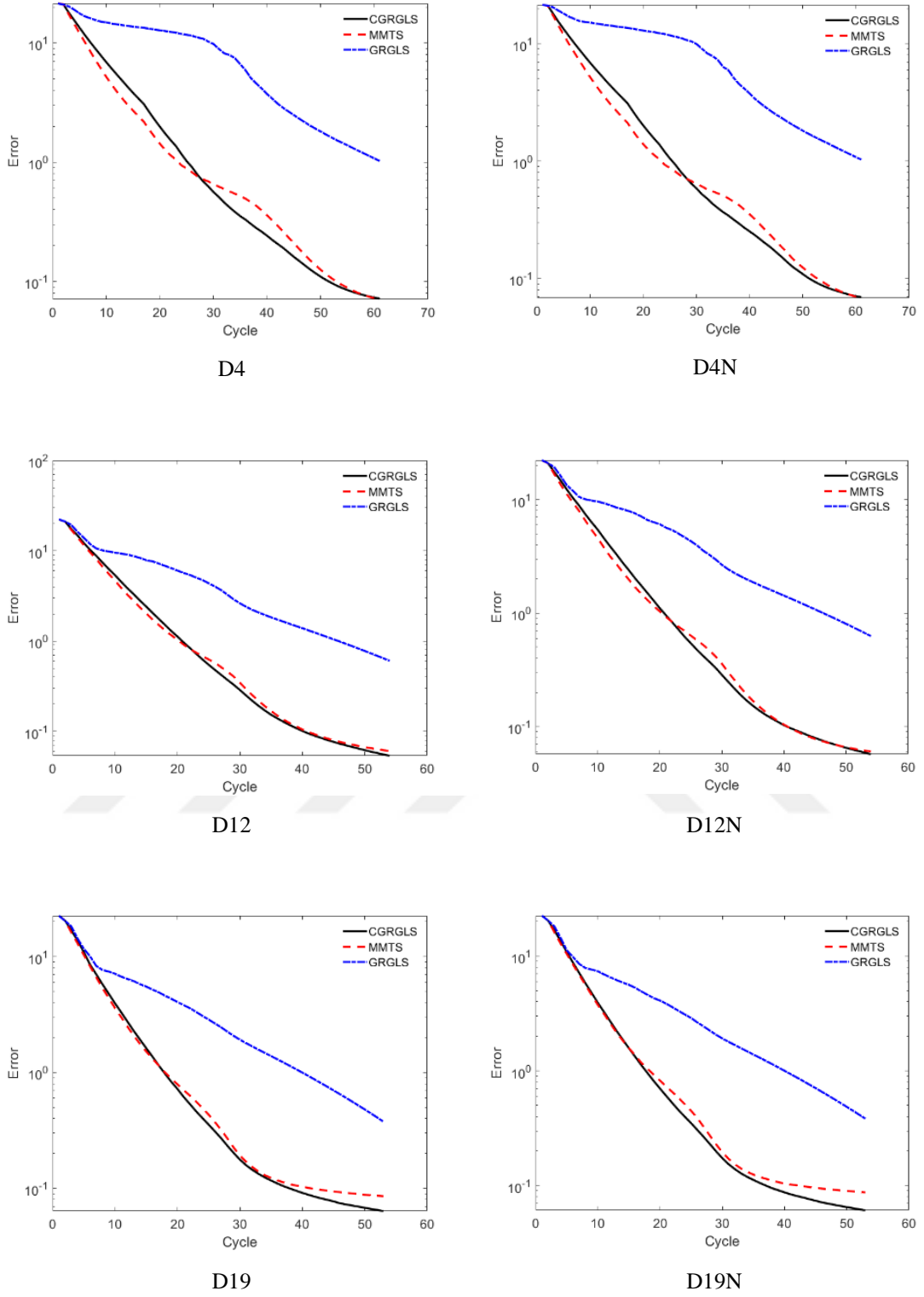
Yerel arama yönteminin yeni versiyonu CGRGLS'nin performansını test etmek için karşılaştırma amacıyla MLSHADE SPA memetik çerçevesinin orijinalinde kullanılan yerel arama yöntemi MMTS ve önerilen yerel aramanın ilk versiyonu GRGLS yöntemleri seçilmiştir. Bu üç yöntemin ortak noktası, MTSLs1 yönteminden esinlenerek önerilen yöntemler olmalarıdır. Bölüm 4.2 de GRGLS, Bölüm 5.3.1 de ise MMTS yerel arama yöntemi işleyişi anlatılmıştır.

Çizelge 5.17, test sonuçlarının standart sapmalarını, ortalama ve en iyi sonuç değerlerini göstermektedir. CGRGLS, altı veri setinden dördünde (D12, D12N, D19, D19N) en iyi sonuçları elde ederek birinciliği elde ederken, MMTS yöntemi D4 ve D4N verilerinde en iyi sonuçları elde ederek ikinci olmuştur. Ancak önerilen yöntemin bir önceki versiyonu olan GRLGS algoritması hiçbir veri setinde en iyi değeri elde edememiştir. Bu durum, CGRGLS yönteminin kendisinin önceki sürümünü etkili bir şekilde geliştirdiğini kanıtlamaktadır. CGRGLS algoritmasının güçlü performansı, kaotik haritaların ergodikliğinin ve çözüm uzayının etkin bir şekilde keşfedilmesinin sonucudur.

Çizelge 5.17. CGRGLS'nin diğer yerel arama metotları ile karşılaştırma sonuçları

	MMTS			GRGLS			CGRGLS		
	En İyi	Ortalama	Std	En İyi	Ortalama	Std	En İyi	Ortalama	Std
D4	6.683E-02	7.052E-02	2.119E-03	9.376E-01	1.023E+00	5.247E-02	6.831E-02	7.168E-02	2.413E-03
D4N	6.559E-02	6.817E-02	1.347E-03	8.651E-01	1.031E+00	8.335E-02	6.675E-02	6.948E-02	2.373E-03
D12	5.097E-02	5.975E-02	3.771E-03	5.299E-01	5.798E-01	3.430E-02	3.408E-02	5.287E-02	1.009E-02
D12N	5.328E-02	5.953E-02	4.309E-03	5.228E-01	6.020E-01	3.990E-02	3.417E-02	5.606E-02	1.161E-02
D19	7.653E-02	8.520E-02	5.668E-03	3.082E-01	3.611E-01	2.347E-02	4.743E-02	6.298E-02	1.086E-02
D19N	7.743E-02	8.711E-02	5.848E-03	3.118E-01	3.687E-01	2.469E-02	4.587E-02	6.046E-02	8.496E-03

Optimizasyon yöntemleri, optimum çözümü ararken benzersiz arama davranışları gösterir. Yakınsama eğrileri, bir optimizasyon yönteminin çözüm arama davranışını ortaya koymada en önemli araçtır. Yakınsama eğrileri, iterasyonlar boyunca yöntem tarafından elde edilen en iyi çözüme dayalı olarak üretilir. Üç farklı yerel arama tekniğinin Big-Opt test fonksiyon kümesi fonksiyonlarında elde ettiği sonuçlardan üretilen yakınsama eğrileri Şekil 5.4'te gösterilmektedir.



Şekil 5.4. Big-Opt fonksiyonları için CGRGLS yakınsama eğrileri

Şekil 5.4'te görüldüğü gibi CGRGLS ve MMTS yöntemleri 1024 boyutlu D4 ve D4N üzerinde GRGLS yöntemine göre birbirine yakın sonuçlar vermektedir. Problem boyutu arttıkça D12 ve D19'un yakınsama eğrileri kaotik yaklaşımın yerel arama yönteminin başarısını artırdığını göstermekte olup bu durum Çizelge 5.17'deki sonuçları desteklemektedir. Kaotik yöntemin başarısı problemin boyutu arttıkça artmaktadır. Ayrıca iterasyonlar ilerledikçe düzenli bir iyileştirmeye sahiptir. Diğer bir deyişle, D19 ve D19N üzerindeki eğriler incelendiğinde, daha fazla iterasyon çalıştırılırsa CGRGLS'nin diğer yöntemlere göre çok daha iyi çözümlere ulaşacağı sonucuna varılabilir.

MMTS yöntemi, ilk versiyon olan GRGLS algoritmasından daha iyi performans göstermektedir. GRGLS erken yakınsamakla birlikte yerel minimumlara takılarak herhangi bir veri setinde önemli bir eğri oluşturamamıştır. GRGLS ve CGRGLS'nin performansı, kaotik katkının yöntemi önceki versiyona göre etkili bir şekilde geliştirdiğini kanıtlamaktadır.

Yerel arama yöntemlerinin problem çözme performansını istatistiksel olarak analiz etmek için Wilcoxon işaretli sıra testi kullanılmıştır. Algoritmaların problem tabanlı ikili karşılaştırması için 25 çalıştırma sonucunda elde edilen global minimum değerler kullanılarak test gerçekleştirilmiştir. Çizelge 5.18, CGRGLS ve diğer yerel arama yöntemleri arasında Wilcoxon işaretli sıra testi ile yapılan istatistiksel karşılaştırmaları göstermektedir.

Çizelge 5.18. Yerel arama yöntemleri için Wilcoxon işaretli sıralama testi sonuçları

Fonk.	CGRGLS ile GRGLS			CGRGLS ile MMTS		
	<i>p Value</i>	<i>T</i>	<i>W</i>	<i>p Value</i>	<i>T</i>	<i>W</i>
D4	1,23E-05	0	+	1,50E-01	216	=
D4N	1,23E-05	0	+	8,71E-03	260	-
D12	1,23E-05	0	+	6,31E-03	61	+
D12N	1,23E-05	0	+	1,15E-01	104	=
D19	1,23E-05	0	+	1,23E-05	0	+
D19N	1,23E-05	0	+	1,23E-05	0	+
	+ / = / -		6/0/0	+ / = / -		3/2/1

Çizelge 5.18’da listelenen p-Değeri, T ve W değerlerini ışığında W değeri "=" ise istatistiksel fark yoktur (eşittir), "+" ise sonuçlar arasında anlamlı fark vardır ve CGRGLS yöntemi diğer yonteme göre daha başarılıdır (kazanır), "-" ise, sonuçlar arasında anlamlı bir fark vardır ve diğer yöntem daha başarılıdır (kaybeder). Çizelge 5.18’un son satırından Wilcoxon işaretli sıra testi sonucunda karşılaştırılan iki yöntem için toplam kazanma, eşitleme ve kaybetme değerleri verilmiştir. Sonuçlar göstermektedir ki CGRGLS anlamlılık düzeyi $\alpha = 0.05$ olmak üzere karşılaştırma algoritmalarından istatistiksel olarak daha iyi sonuçlar elde etmiştir. FEs arttıkça CGRGLS'nin avantajı daha iyi olduğu görülmektedir.

5.4.3. Diğer Algoritmalarla Karşılaştırma

Bu deneysel çalışmada MLSHADE-SPA memetik çerçeve algoritmasından yerel arama yöntemi olan MMTS çıkarılarak yerine kaotik haritalar aracılığı ile geliştirilen ve önerilen yerel arama yönteminin son versiyonu olan CGRGLS yerleştirilmiştir. Oluşan yeni algoritma Kaotik İyileştirilmiş MLSHADE-SPA (CIMLSHADE-SPA) olarak adlandırılmıştır. Önerilen algoritmanın performansının diğer evrimsel veya sürü zekası tabanlı optimizasyon algoritmalarına göre yeterli olup olmadığını analiz etmek için elde edilen sonuçlar doğrudan (Aslan, 2020)'dan alınan 5 algoritmanın sonuçları ile karşılaştırılmıştır. Her problem için 100.000 uygunluk değerlendirmesi tüketerek gerçekleştirilen 30 çalışma sonucunda CIMLSHADE-SPA algoritmasından elde edilen sonuçlar ile diğer 5 algoritmanın sonuçları da Çizelge 5.19’de listelenmiştir.

Karşılaştırma amaçlı seçilen algoritmalar aşağıdaki gibidir:

1. COABC: Yakınsayan-Gözcü ABC, gözcü aşamasında yeni bir çözüm arama denklemi içeren ABC algoritmasının değiştirilmiş bir versiyonudur (Luo ve ark., 2013).
2. ACDE: DE'lerin otomatik konfigürasyonları, popülasyon tabanlı küresel arama olarak Diferansiyel Gelişim Algoritması varyantlarını ve yerel arama algoritması olarak iç nokta yöntemini (interior point method) birleştiren memetik bir çerçevedir (Elsayed ve Sarker, 2015).

3. SaNSDE-CC: CC yöntemini kullanan komşuluk aramalı kendi kendini uyarlayabilen Diferansiyel Gelişim Algoritması, geliştirilmiş bir Diferansiyel Gelişim algoritması varyantıdır (Omidvar ve ark., 2013).

4. JADE: İsteğe bağlı arşivli “DE/current-to-pbest” mutasyon stratejisini uygulayan, parametrelerini uyarlamalı bir şekilde kontrol eden bir Diferansiyel gelişim algoritma çeşididir (Zhang ve Sanderson, 2009).

5. SHADE: Başarı geçmişine dayalı uyarlanabilir DE, geçmişe dayalı bir parametre uyarlama şeması kullanan uyarlanabilir bir diferansiyel gelişim algoritmasıdır (Tanabe ve Fukunaga, 2013).

Çizelge 5.19 incelendiğinde CIMLSHADE-SPA, COABC ve ACDE yöntemlerinin birbirine yakın olduğu ve diğerlerine göre daha başarılı sonuçlar verdiği görülmektedir. CIMLSHADE-SPA, D4 ve D4N, COABC D19 ve ACDE D12, D12N ve D19N verilerinde en iyi sonuçları vermiştir. Toplam sıra değerleri, bu 3 yöntemin diğerlerinden çok daha başarılı olduğunu kanıtlıyor. CIMLSHADE-SPA en iyi veya en yakın değerleri vererek tutarlı sonuçlar veren bir yöntem olduğunu göstermektedir.

Friedman testi, aynı örnekten ölçülen birden fazla değişken arasındaki farklılıkları test etmek için kullanılan parametrik olmayan istatistiksel bir testtir. Küçük örneklerde normal dağılım varsayımının sağlanmadığı durumlarda tercih edilir ve verilerin sıralamasını kullanarak sonuç üretir. Çizelge 5.19 ve Çizelge 5.20’deki diğer algoritmalar tarafından elde edilen sonuçlar, referans verilen ilgili yayınlardan elde edilmiş olup sonuçların alındığı yayınlarda her çalıştırma sonucu yerine sadece sonuçların ortalaması verilmektedir. Bu nedenle listelenen sonuçlar arasında anlamlı bir fark olup olmadığını göstermek için Friedman testi uygulanmıştır. Bu istatistiksel test için anlamlılık düzeyi 0,05 olarak ayarlanmıştır. p değerinin 0,05’ten küçük olması, elde edilen tüm sonuçlar arasında istatistiksel olarak anlamlı bir fark olduğu anlamına gelir. Aksi takdirde, önemli bir fark yoktur denir. Friedman testinden elde edilen p değerinin anlamlılık düzeyinden küçük olduğu Çizelge 5.19’nin son satırında görülmektedir bu da elde edilen tüm sonuçlar arasında istatistiksel olarak anlamlı bir fark olduğunu doğrulamaktadır.

Çizelge 5.19. CIMLSHADE-SPA'nın diğer algoritmalarla karşılaştırma sonuçları

		CIMLSHADE-SPA	COABC	ACDE	SaNSDE-CC	JADE	SHADE
D4	Ortalama	6.14E-02	6.35e-02	6.55e-02	5.11E+00	2.70E+00	3.22E+00
	En İyi	6.11E-02	6.29e-02	6.13e-02	3.28E+00	1.81E+00	1.42E+00
	Std.	1.51E-04	2.80e-04	3.01e-06	1.57e-01	4.76e-01	7.23e-01
	Sıralama	1	2	3	6	4	5
D4N	Ortalama	5.91E-02	6.14e-02	5.93e-02	1.11E+01	9.45E+00	4.19E+00
	En İyi	5.90E-02	6.10e-02	5.93e-02	1.79E+00	1.02E+00	2.11E+00
	Std.	8.09E-05	2.62e-04	3.51e-06	2.11E+00	9.62e-01	1.41e-01
	Sıralama	1	3	2	6	5	4
D12	Ortalama	2.72E-01	6.26e-02	5.12e-02	6.22E+00	8.05E+00	5.16E+00
	En İyi	1.49E-01	5.26e-02	3.61e-02	4.22E+00	4.12E+00	2.64E+00
	Std.	5.09E-02	4.62e-03	9.60e-03	1.02e-01	2.73E+00	1.84e-01
	Sıralama	3	2	1	5	6	4
D12N	Ortalama	2.56E-01	6.21e-02	5.09e-02	1.27E+01	1.38E+01	7.73E+00
	En İyi	1.68E-01	5.48e-02	3.63e-02	5.11E+00	7.28E+00	3.17E+00
	Std.	5.13E-02	4.54e-03	1.11e-02	1.13E+00	1.09E+00	1.27e-01
	Sıralama	3	2	1	5	6	4
D19	Ortalama	1.07E+00	1.58e-01	2.15e-01	1.97E+02	1.08E+01	9.10E+00
	En İyi	8.41E-01	1.40e-01	6.28e-02	1.23E+02	5.17E+00	6.17E+00
	Std.	1.24E-01	1.08e-02	7.49e-02	2.32E+01	2.30E+00	1.76e-01
	Sıralama	3	1	2	6	5	4
D19N	Ortalama	1.15E+00	1.59e-01	1.36e-01	1.86E+02	1.67E+01	8.26E+00
	En İyi	7.41E-01	1.40e-01	7.02e-02	1.47E+02	9.72E+00	4.42E+00
	Std.	1.90E-01	8.88e-03	5.16e-02	1.26E+01	1.28E+00	1.282e-01
	Sıralama	3	2	1	6	5	4
Sıralama Toplamı		14	12	10	34	31	25
Friedman Test	Ortalama Sıra	2.333333	2	1.666667	5.666667	5.166667	4.166667
	p-Value	1.10E-04					

5.4.4. LSGO yarışmalarını kazanan algoritmalarla karşılaştırma

LSGO, çok sayıda karar değişkenini içeren özel bir küresel optimizasyon problemleri kategorisi için kullanılan bir terimdir. LSGO'da değişken sayısındaki artıştan kaynaklanan bazı sorunlar nedeniyle hem modelleme hem de çözüm süreci oldukça karmaşık ve zor hale gelebilmektedir. Bu faktörler, boyutun artmasıyla birlikte arama alanının üstel olarak artması, bazı fonksiyonların karmaşıklığının ve özelliklerinin değişebilmesi, birbirine bağımlı değişken sayısının artması vb. olarak sıralanabilir. Bu sebeplerden dolayı normal boyutta yüksek performanslı algoritmalar boyut büyüdükçe etkinliğini kaybedebilir. Bu sorunların üstesinden gelmek için çeşitli algoritmalar ve teknikler geliştirmeyi teşvik etmek ve motive etmek amacıyla CEC, 2008 yılından bu yana bu alanda yarışmalar ve özel oturumlar düzenliyor. CEC özel oturumları ve LSGO yarışmaları hakkında ayrıntılı bilgi (Molina ve ark., 2018a)'da bulunabilir.

Bu tez bünyesinde bir test fonksiyon kümesi olarak kullanılan Big-Opt problemleri karmaşık ve zorlu bir LSGO problemi olarak değerlendirilmektedir. Önerilen yöntemin yeterliliğini ve performansını göstermek için Big-Opt problemlerinin çözümünde CEC LSGO yarışmalarının kazananları ile CGRGLS içeren CIMLSHADE-SPA algoritmasını karşılaştırmak etkili bir gösterge olacaktır.

Kazanan algoritmaların Big-Opt test fonksiyon kümesinde elde ettikleri sonuçlar (Molina ve ark., 2019) yayından alınmıştır. Adil bir karşılaştırma sağlamak için ilgili yayındaki kurallar ve sonlandırma koşulu dikkate alınmıştır. CIMLSHADE-SPA memetik çerçeve algoritmasının 1.000.000 uygunluk değerlendirmesi tüketerek 10 kez çalıştırılmasıyla elde edilen ortalama sonuçlar, yarışma kazananlarının sonuçlarıyla karşılaştırılarak Çizelge 5.20'de gösterilmiştir

MOS-2011, bu yarışma için özel olarak oluşturulmuş bir test fonksiyon kümesi üzerindeki performansı ile SOCO 2011 organizasyonunun kazananıdır (LaTorre ve ark., 2011). MOS-2013 ve varyantları, büyük ölçekli küresel optimizasyon için düzenlenen CEC2012, CEC2013, CEC2015 yarışmalarının kazananıdır (LaTorre ve ark., 2013). 2015 yılında bir Büyük Veri Yarışması gerçekleştirilmiş ve büyük ölçekli küresel optimizasyon için yeni bir test fonksiyon kümesi önerilmiştir. Bu özel oturum için birkaç yöntem önerilmiş olsa da gerçek bir yarışma yapmak için yeterli gelmemiştir ancak MAGA (A Multi-Agent Genetic Algorithm) (Zhang ve ark., 2015), aralarında en iyi sonuçları veren algoritmadır. 2018'de SHADEILS, CEC2013 LSGO test fonksiyon kümesi üzerindeki performansı ile CEC2018 yarışmasının galibi oldu (Molina ve ark., 2018b) MLSHADE-

SPA, CEC2018 yarışmasında MOS-2013'ten daha iyi sonuçlar alarak ikinci oldu (Hadi ve ark., 2019).

Daha az sayıda değişken içeren gerçek dünya probleminin çözülmesinin daha kolay olduğu düşünülebilir, ancak bazen algoritmanın, sinyaller arasındaki korelasyonu belirlemeye ve artefaktları bunlardan ayırmaya yönelik yeterli bilgiyi elde etmesi için belirli sayıda değişkene ihtiyacı vardır. Bu, D4 (1024)'ün problemler arasında gerçek bir ortamı en az temsil ettiği anlamına gelir. D12 probleminde optimize edilecek daha fazla değişken olmasına rağmen (3072 değişken), daha fazla bilgiye erişilebildiği için algoritmalar boyuta rağmen en iyi sonuçları almıştır. En büyük arama uzayına sahip 4864 değişkenli D19 probleminin, D12'den daha düşük sonuçlar üreteceği tahmin edilmekle birlikte önemli sayıda değişkenden elde edilen bilgi nedeniyle yine de D4'ten daha etkili bir şekilde optimize edildiği görülmektedir (Zhang ve ark., 2015).

Çizelge 5.20'de listelenen sonuçlar, farklı algoritmalar tarafından üretilen sonuçların oldukça benzer ve hatta aynı olduğunu göstermektedir. Bu da önerilen CIMLSHADE-SPA algoritmasının diğer LSGO kazanan algoritmaları gibi hedeflenen optimum değerlere başarıyla ulaştığını göstermektedir. Çizelge 5.20'nin en alt satırında gösterilen Friedman testi sonucu elde edilen p değerinin anlamlılık düzeyinden (0,05) büyük olması bu durumu doğrulamaktadır.

Çizelge 5.20. CIMLSHADE-SPA'nın CEC Yarışması Kazananları ile Karşılaştırma Sonuçları

	MOS-2011	MOS-2013	MAGA	SHADE-ILS	MLSHADE-SPA	CIMLSHADE-SPA
D4	0.06103	0.06103	0.0610	0.06103	0.06103	0.06103
D4N	0.05897	0.05897	0.0590	0.05897	0.05897	0.05897
D12	0.00198	0.00194	0.0019	0.00194	0.00194	0.00194
D12N	0.00188	0.00183	0.0018	0.00183	0.00183	0.00183
D19	0.08940	0.00251	0.0025	0.00252	0.00252	0.00252
D19N	0.09180	0.00256	0.0026	0.00256	0.00256	0.00256
Friedman Test	Ortalama Sıra 5.166667	3.083333	2.5	3.416667	3.416667	3.416667
	p-Value 6.19E-02					

6. SONUÇLAR VE ÖNERİLER

Optimizasyon, matematiksel olarak çeşitli değişkenlere bağlı olarak değişen bir amaç fonksiyonunu mümkün olan alternatiflerden belirli koşullar dikkate alınarak en uygun (en küçük veya en büyük) yapan değerlerin bulunması işlemidir. Günlük hayatımızın hemen hemen her alanında karşılaştığımız problemler bir mühendislik tasarımı ve optimizasyon süreci gerektirmektedir. Bir işletmenin karının maksimize edilmesi, büyük ölçekli trafik ağlarında araç yönlendirme, bir alana sığacak maksimum kutu yerleşimi, bir tekstil üretiminde minimum kumaş firesi vererek maksimum parça üretiminin sağlanması, maksimum çekim alanı oluşturacak şekilde cihaz yerleşimi ve benzeri gibi çoğaltabileceğimiz birçok problemin çözümü optimizasyon tekniği kullanılarak yapılabilir. Günümüzde teknolojinin hızla ilerlemesi nedeniyle ihtiyaçlar değişmekte ve gittikçe çözülmesi gereken problemler daha karmaşık ve daha büyük boyutlu hale gelmektedir.

Problem boyutu, olası çözümlerin sayısı ve probleme özgü kısıtlamalar, bir optimizasyon probleminin karmaşıklığını etkileyen önemli faktörlerdir. Karmaşık ve büyük boyutlu optimizasyon problemleri için optimum çözümü bulmak zorlaşır ve yüksek maliyet (zaman, bellek vb.) gerektirir. Gerçek dünya problemlerinin çoğu hedeflenen optimalliği garanti edebilecek matematiksel optimizasyon araçlarının dayattığı matematiksel gereklilikleri karşılayamaz bir diğer deyişle klasik yöntemler bu tür problemlerin çözümü için yetersiz kalır. Bu nedenle çözüme ulaşmak için rastgelelik içeren stokastik yöntemler ile ideal çözüme ulaşılmaya çalışılır. Stokastik olan metasezgisel yöntemler, optimum çözümü garanti etmese de probleme bağlı olmadığından daha geniş bir uygulama alanına sahiptir ve kabul edilebilir bir zaman içinde ulaşabildiği en uygun çözümü sunabilmeleri nedeniyle yaygın kullanıma sahiptirler. Metasezgisel optimizasyon algoritmaları, genellikle küçük optimizasyon problemlerinde yeterli arama yetenekleri gösterir, ancak karar değişkeni sayısı yani problem boyutu arttıkça süreç karmaşıklaşır ve bu algoritmalar genellikle tasarımındaki etkinliğini ve performansını boyut büyüdükçe kaybederler. Büyük ölçekli optimizasyon problemlerindeki boyutla birlikte üstel olarak genişleyen arama alanı, çözüm için farklı teknik ve mekanizmalarla algoritmaların desteklenmesini veya farklı çözüm yaklaşımlarının sunulmasını gerektirmektedir. Bu kapsamda gerçekleştirilen çalışmalar oldukça geniş bir alana yayılmış olmak ile birlikte genel olarak ayrıştırma esaslı yaklaşımlar ve ayrıştırma esaslı olmayan yaklaşımlar olmak üzere iki kategoride

değerlendirilebilir. Bunların birbirinden farkı problemi parçalara ayırarak veya bir bütün olarak çözmeye çalışmalarıdır. Bu tez kapsamında ayrıştırma esaslı olmayan, problemi bir bütün olarak değerlendirerek çözmeyi hedefleyen teknikler üzerinde çalışılmıştır.

6.1 Sonuçlar

Ayrıştırma esaslı olmayan yaklaşımlar kendi arasında standart optimizasyon algoritmaları ve memetik algoritmalar olarak iki ayrı sınıfta değerlendirilebilir. Bu tez kapsamında standart optimizasyon algoritmaları alanında iki ve memetik algoritmalar alanında iki olmak üzere dört başlıkta deneysel çalışmalar, testler ve istatistiksel analizler gerçekleştirilmiştir.

LSGO sorunlarıyla başa çıkmak için tercih edilen çözüm yaklaşımlarından biri standart evrimsel algoritmaların ek düzenleme ve tekniklerle geliştirilmesidir. Bu kapsamda gerçekleştirilen çalışmalarda Yapay Alg Algoritması (Artificial Algae Algorithm, AAA) ve CEC2010 test fonksiyon kümesi kullanılmıştır. Öncelikle algoritmanın boyut büyüdükçe performansının nasıl etkilendiği test edilmiştir. AAA setin ilk 3 fonksiyonu için 100, 500 ve 1000 boyut olacak şekilde çalıştırılmıştır. Elde edilen sonuçlar beklendiği gibi boyut büyüdükçe yaşanan performans düşüklüğünü doğrular nitelikte olmuştur. AAA'nın LSGO problemlerini çözme performansını değerlendirmek için setin tüm fonksiyonlarında çalıştırılmasıyla elde edilen sonuçlar, LSGO problemlerini çözme amacıyla tasarlanmış DECC-G algoritmasıyla karşılaştırılmıştır. AAA'nın orijinal halinin, 20 fonksiyonun 11'inde DECC-CG algoritmasından daha iyi sonuçlar elde etmesi bu alanda umut vadeden bir algoritma olabileceğini göstermiştir.

Bir sonraki adımda algoritmanın LSGO performansını artırmak amacıyla geliştirme denemeleri yapılmış ve arama uzayında hem keşif hem de sömürüye katkı sağlamak amacıyla AAA helisel hareket aşamasına odaklanılmıştır. Diferansiyel gelişim algoritması LSGO alanında en çok tercih edilen algoritmalarından biridir. Bu algoritmanın mutasyon adı verilen aşamasında bazı parametrelerin rastgele değişimi sayesinde olası çözüm noktası çözüm uzayında hareket ettirilmektedir. Aynı amaçla çözüm uzayındaki yer değişimi AAA bünyesinde rastgele seçilen üç alg hücrelerinin değiştirilmesi sonucu gerçekleşen helisel hareketle sağlanmaktadır. Literatürde DE bünyesinde kullanılmak üzere önerilen birçok farklı mutasyon stratejisi mevcuttur bunlardan parametre farklarını kullanan bir mutasyon stratejisinden ilham alınarak AAA bünyesinde helisel hareketin hemen ardından aynı alg kolonisinin dördüncü bir hücresinde daha değişiklik yapılmasına

karar verilmiştir. Dördüncü boyutun değiştirilmesinde helisel hareketi sağlayan üç hesaplamaların sonuçlarını kullanan denklem kullanılmıştır. Bu yeni versiyon, Değiştirilmiş Yapay Alg Algoritması (Modified Artificial Algae Algorithm, MAAA) olarak adlandırılmıştır. Literatürdeki 5 algoritma ile yapılan karşılaştırma sonucu MAAA'nın ikinci olması, yapılan değişikliğin algoritmanın orijinal versiyonu geliştirerek LSGO problemlerin çözümünde performans artışı sağladığını doğrulamıştır.

LSGO sorunlarıyla başa çıkmak için çokça tercih edilen başarılı çözüm yaklaşımlarından bir diğeri de memetik algoritmalar. Evrimsel algoritmaların yerel arama yöntemleri ile hibritleştirilmesi sonucu oluşan bu algoritmalar keşif ve sömürü dengesini sağlayarak daha iyi çözümlere ulaşmayı amaçlamaktadır. Memetik algoritmalar kategorisinde, boyuta bağlı katlanarak büyüyen çözüm uzayında daha kaliteli çözümler bularak algoritmaların LSGO performansını artırmak amacıyla yeni bir yerel arama yöntemi geliştirilmiştir. Geliştirilen bu yönteme dinamik adım boyutuna sahip altın oran rehberli yerel arama (Golden Ratio Guided Local Search with dynamic step size, GRGLS) adı verilmiştir. Yöntem bünyesinde kâinatın var oluşundan beri hayatın birçok alanında kullanılan matematiksel bir simetri ve harika bir tasarım özelliği olarak nitelenen altın oran kullanılmıştır. Arama uzayında yeni olası çözümleri aramak için yapılan hareket adım büyüklüğü aracılığıyla gerçekleşme olup adım büyüklüğü değişimi ise altın oran ve ters altın oran katsayıları kılavuzluğunda değiştirilmektedir. Yeni yerel arama yönteminin temel özellikleri: popülasyondaki en yüksek uygunluk değerine sahip olan en iyi birey üzerinde çalışması, birinci boyuttan sonuncu boyuta kadar her bir boyut üzerinde tek tek çalışması, yeni olası çözüm ararken mevcut boyutun adım büyüklüğü kadar zıt yönlerde arama yapması, adım büyüklüğü değişiminde altın oran ve ters altın oran katsayısını kullanması olarak sıralanabilir. Performans değerlendirmesi için öncelikle ana algoritma olarak MLSHADE-SPA çerçeve algoritması seçilmiş ve bu algoritmanın yerel arama aşamasına her seferinde sadece biri olmak üzere MTSL1, MMTS ve GRGLS yerel arama yöntemleri yerleştirilerek sonuçlar birbiriyle kıyaslanmıştır. GRGLS yerel arama yöntemini kullanan versiyonun orijinal algoritmanın elde ettiği sonuçlardan daha iyi sonuçlar bulması GRGLS yerel arama yönteminin algoritmanın LSGO performansını artırdığını göstermiştir. MLSHADE_SPA çerçeve algoritmasının yerel arama kısmına GRGLS yerleştirilerek oluşan algoritma Geliştirilmiş MLSHADE-SPA algoritması (IMLSHADE-SPA) olarak adlandırılarak diğer algoritmalarla karşılaştırılmıştır. CEC2013 test fonksiyon kümesi üzerinde gerçekleştirilen testlerde önerilen yöntemin kullanıldığı yeni memetik çerçeve

algoritması karşılaştırılan 10 algoritma arasında en iyi sonuçları elde etmiştir. Algoritma tüm fonksiyonlarda iyi sonuçlar elde ederken, örtüşen ve ayrılmaz fonksiyonlarda diğerlerine göre üstün performans göstermektedir.

Memetik algoritmalar kapsamında ikinci çalışma GRGLS yönteminin LSGO performansını daha da geliştirmek amacıyla yapılmıştır. GRGLS yönteminin performansını belirleyen en önemli etken, arama uzayında hareketi sağlayan adım büyüklüklerini dinamik olarak uyarlama işlemidir. Bu noktada kaotik haritaların tekrarsız rastgele sayı üretme özelliği kullanılarak Singer Kaotik haritasından üretilen sayı adım büyüklüğü belirleme denkleminde ilave bir katsayı olarak kullanılmıştır. Böylelikle arama uzayında gerçekleştirilen hareketlerin altın oran ve ters altın oranla birlikte kaotik olarak da etkilenmesi sağlanmıştır. GRGLS yönteminin geliştirilmiş bu versiyonuna Kaotik Altın Oran Rehberli Yerel Arama yani CGRGLS adı verilmiştir. Yöntemin performansını değerlendirmek üzere klasik test fonksiyon kümesi yerine gerçek dünya problemlerini temsil eden CEC2015 Big-Opt test fonksiyon kümesi kullanılmıştır. Önerilen yöntemin performansını değerlendirmek amacıyla üç ayrı karşılaştırma yapılmıştır. Birinci karşılaştırmada önerilen yeni yerel arama yöntemi bir önceki versiyonuna göre gelişimini incelemek amacıyla MLSHADE-SPA algoritması yerel arama kısmına CGRGLS yerleştirilerek diğer arama yöntemleri ile kıyaslanmıştır. Sonuçlar incelendiğinde kaotik katsayı kullanımının varolan yerel arama yöntemini etkili bir şekilde geliştirdiği anlaşılmıştır. Yerel arama aşamasında CGRGLS kullanan MLSHADE-SPA memetik çerçeve algoritması, CIMLSHADE-SPA olarak adlandırılmış ve iki ayrı deneysel çalışmada performansı değerlendirilmiştir. Birinci çalışma bünyesinde literatürdeki 5 algoritmayla kıyaslanarak tutarlı sonuçlar veren bir yöntem olduğunu gösterilirken diğer çalışma bünyesinde LSGO yarışmalarında dereceye giren 5 algoritmayla kıyaslanarak bu algoritmalar gibi hedeflenen optimum değerlere başarıyla ulaşabildiği kanıtlanmıştır.

Sonuç olarak, bu tez kapsamında büyük ölçekli optimizasyon problemlerinin çözümünde ayrıştırma esaslı olmayan yaklaşımlar üzerinde çalışılmıştır. Standart optimizasyon algoritmaları ve memetik algoritmalar olmak üzere iki temel kategoride deneysel çalışmalar yapılmış, testler ve istatistiksel analizler yorumlanmıştır. Standart algoritmalar kategorisinde Yapay Alg Algoritmasının LSGO performansı değerlendirilerek işleyişi incelenmiş ve yapılan değişiklik sonucu önerilen MAAA algoritması ile algoritmanın orijinal halinin bu alandaki çözüm performansına olumlu katkı sağlanmıştır. Memetik Algoritmalar bünyesinde kullanılmak üzere yeni bir yerel

arama metodu (GRGLS) geliştirilmiş ve performans artışı amaçlı yapılan çalışmalar neticesinde kaotik haritalar yardımıyla bir üst versiyonu olan CGRGLS önerilmiştir. Yapılan tüm deneysel çalışmalar GRGLS ve CGRGLS yerel arama yöntemlerinin LSGO araştırma alanı için etkili ve verimli bir yerel arama yöntemi olarak kullanılabileceğini doğrulamıştır.

6.2 Öneriler

Metasezgisel optimizasyon algoritmalarının keşif ve sömürü olmak üzere iki önemli arama stratejisi vardır ve algoritma başarısı bu iki stratejinin uyumlu ve dengeli çalışmasına bağlıdır. Keşif özelliği yüksek algoritmalar optimum çözüme yaklaşmakta başarısız olurken, sömürü özelliği yüksek bir algoritma yerel bir optimum çözüme takılıp kalabilir. Bu nedenle güçlü bir algoritmanın arama alanını etkili bir şekilde keşfederken umut verici çözümlere de ince ayar yapması beklenmektedir. Memetik algoritmalar bünyesinde arama uzayını keşfetme görevini bir EA üstlenirken sömürü görevini ise EA tarafından oluşturulan ve gelecek vaat eden bireyleri daha da iyileştiren bir yerel arama yöntemi üstlenmektedir. Amaç bu iki mekanizmayı bir araya getirerek keşif ve sömürü arasında hedeflenen dengeyi kurmaktır. MA tasarımının arkasındaki mantık, EA'ların yerel sömürüdeki eksikliğini ve yerel aramanın küresel keşifteki yetersizliğini birbirleri sayesinde gidererek karlı bir sinerji elde etmek ve işbirlikçi bir şekilde çalışmalarını sağlayarak etkili ve verimli bir küresel optimizasyon metodu oluşturmaktır. MA'yı oluşturan iki ana bileşen (evrimsel algoritma ve yerel arama yöntemi) birbirinden bağımsız olarak tasarlanabilir ve uygulanabilir. Bu sayede, geliştirilmeleri kolaylaşmakta ve performansları belirli problemleri tek algortmada hem keşif hem de sömürü özelliğini birleştirerek çözmeye çalışan algoritmalara kıyasla daha yüksek olabilmektedir. Dolayısıyla bu alanda kullanılabilecek yeni yerel arama yöntemleri geliştirilebilir. Ya da bu tez kapsamında önerilen iki yerel arama yöntemi farklı yaklaşım ve tekniklerle desteklenerek performansları geliştirilebilir ve farklı evrimsel algoritmalarla birlikte kullanılarak yeni memetik algoritmalar tasarlanabilir.

Önerilen GRGLS yöntemi genel olarak tüm işlevlerde iyi sonuçlar vermekle birlikte karşılaştırma sonuçlarındaki örtüşen ve ayrılamayan fonksiyonlar üzerindeki üstün performansı dikkat çekicidir. Ancak yöntemin tamamen ayrılabilir fonksiyonlardaki başarısı düşüktür ve gelecekteki çalışmalar için yöntemin tam ayrılabilir

fonksiyonlardaki performansı, farklı teknikler uygulanarak veya başka bir güçlü yerel arama yöntemiyle hibrit kullanılarak geliştirme çalışmaları yapılabilir.

CEC2015 Big-Opt test fonksiyon kümesi problemlerinin çözümünde Majdouli ve diğerleri tarafından tanıtılan popülasyon başlatma tekniği dikkat çekmektedir(El Majdouli ve ark., 2016). Bu teknikte, belirtilen alt/üst sınırlar içinde potansiyel uygulanabilir çözümlerin parametrelerini rastgele başlatmak yerine doğrudan orijinal EEG sinyalinden gelen veriler kullanılmaktadır. IMLSHADE-SPA veya CIMLSHADE-SPA algoritmaları bünyesinde popülasyon başlatma tekniği olarak kullanılarak tekniğin LSGO performansına etkisi araştırılabilir.

Ayrıştırma esaslı yaklaşımların performansı büyük ölçüde kullanılan ayrıştırma tekniğine bağlı olduğundan bu kapsamda değişkenler arası karşılıklı bağımlılıkları tespit edip bu bilgiyi dikkate alarak değişkenleri gruplandırmak için verimli ayrıştırma tekniklerinin geliştirilmesi üzerine çalışılabilir.

LSGO problemlerinin en belirgin özelliği çok büyük arama uzayına sahip olması nedeniyle etkili arama stratejileri gerektirmesidir. Dolayısıyla her bir algoritmanın etkinliği büyük ölçüde optimize edilecek arama alanını ne kadar iyi tarayabildiğine bağlıdır. MOS gibi bazı algoritmalar, her bir fonksiyon için en iyi davranışa sahip algoritmanın aramayı yönlendirebileceğini öngören çok farklı algoritmalarından oluşmaktadır. Bu tür algoritmalar çerçeve algoritma olarak adlandırılmaktadır ve ilave kontrol mekanizmaları ile bünyesindeki algoritma ve teknikleri organize etmektedir. Bu tez kapsamında ana algoritma olarak kullanılan MLSHADE-SPA algoritması da memetik bir çerçeve algoritmasıdır. Bu bağlamda LSGO problemlerini çözmek amacıyla farklı evrimsel algoritmalar, yerel arama yöntemleri, ayrıştırma teknikleri, popülasyon başlatma ve azaltma yöntemleri gibi birçok yöntemin avantajı bir çerçeve altında farklı kombinasyonlarla birleştirilerek yeni çerçeve algoritmalar tasarlanabilir.

Yüksek boyutlu bir problemi optimize etmek genellikle çok zaman alır, bu nedenle gerçek uygulamalarda bu tür problemlerin çözümü için tasarlanan algoritmalar, işlem sürelerini azaltmak için yeni yapılacak geliştirmelerde paralel çalışacak şekilde ve sadece sonuçta değil süreçte de performansını koruyacak şekilde tasarlanabilir.

Algoritmaların performansı değerlendirilirken maliyet yani hedeflenen optimum çözüme ulaşmak için harcadığı uygunluk değerlendirme sayısı önemlidir. En iyi çözüme ulaşan genelde daha çok FEs harcayan algoritma olacaktır. O nedenle algoritma bünyesinde FEs tahsisini yönetmek için ekstra mekanizmalar dahil edilebilir veya genel

uygunluk deęeri üzerinde daha fazla katkıya sahip olan boyutlar veya bileşenler üzerinde daha fazla FEs harcayacak algoritmalar tasarlanabilir.

Büyük ölçekli global optimizasyon alanı geliştirmeye ve araştırmaya açık birçok alt kategori içermektedir. Bu alanda çözüm kalitesine katkı yapacak her bir çözüm yaklaşımı araştırmacılara farklı bir bakış açısı sunarak araştırma alanlarını daha da genişletebilir.



KAYNAKLAR

- Arslan, S. ve Aslan, S., 2022, A new lattice based artificial bee colony algorithm for EEG noise minimization, *JOURNAL OF THE FACULTY OF ENGINEERING AND ARCHITECTURE OF GAZI UNIVERSITY*, 38 (1).
- Aslan, S., 2020, A comparative study between artificial bee colony (ABC) algorithm and its variants on big data optimization, *Memetic Computing*, 12 (2), 129-150.
- Aurasopon, A. ve Khamsen, W., 2019, An improved local search involving bee colony optimization using lambda iteration combined with a golden section search method to solve an economic dispatch problem, *Przeglad Elektrotechniczny*.
- Banitalebi, A., Abd Aziz, M. I. ve Aziz, Z. A., 2016, A self-adaptive binary differential evolution algorithm for large scale binary optimization problems, *Information Sciences*, 367, 487-511.
- Bellman, R., 1956, Dynamic programming, *RAND CORP SANTA MONICA CA*.
- Bergeron, F. ve Reutenauer, C., 2019, Golden ratio and phyllotaxis, a clear mathematical link, *Journal of mathematical biology*, 78, 1-19.
- Cabrera, D. M., 2016, Evolutionary algorithms for large-scale global optimisation: a snapshot, trends and challenges, *Progress in Artificial Intelligence*, 5, 85-89.
- Cao, Z., Wang, L., Shi, Y., Hei, X., Rong, X., Jiang, Q. ve Li, H., 2015, An effective cooperative coevolution framework integrating global and local search for large scale optimization problems, *2015 IEEE Congress on Evolutionary Computation (CEC)*, 1986-1993.
- Chen, S., Locust Swarms for Large Scale Global Optimization of Nonseparable Problems, *Kukkonen, Benchmarking the Classic Differential Evolution Algorithm on Large-Scale Global Optimization Google Scholar*.
- Chen, S., Montgomery, J. ve Bolufé-Röhler, A., 2015, Measuring the curse of dimensionality and its effects on particle swarm optimization and differential evolution, *Applied Intelligence*, 42, 514-526.
- Ciucurel, C., Georgescu, L. ve Iconaru, E. I., 2018, ECG response to submaximal exercise from the perspective of Golden Ratio harmonic rhythm, *Biomedical Signal Processing and Control*, 40, 156-162.
- Colson, B. ve Toint, P. L., 2005, Optimizing partially separable functions without derivatives, *Optimization methods and software*, 20 (4-5), 493-508.
- Črepinšek, M., Liu, S.-H. ve Mernik, M., 2013, Exploration and exploitation in evolutionary algorithms: A survey, *ACM computing surveys (CSUR)*, 45 (3), 1-33.
- Cuevas, E., Enríquez, L., Zaldívar, D. ve Pérez-Cisneros, M., 2018, A selection method for evolutionary algorithms based on the golden section, *Expert Systems with Applications*, 106, 183-196.
- Dabbour, L. M., 2012, Geometric proportions: The underlying structure of design process for Islamic geometric patterns, *Frontiers of Architectural research*, 1 (4), 380-391.
- Del Ser, J., Osaba, E., Molina, D., Yang, X.-S., Salcedo-Sanz, S., Camacho, D., Das, S., Suganthan, P. N., Coello, C. A. C. ve Herrera, F., 2019, Bio-inspired computation: Where we stand and what's next, *Swarm and Evolutionary Computation*, 48, 220-250.
- Demir, F. B., Tuncer, T. ve Kocamaz, A. F., 2020, A chaotic optimization method based on logistic-sine map for numerical function optimization, *Neural Computing and Applications*, 32, 14227-14239.

- Eberhart, R. ve Kennedy, J., 1995, Particle swarm optimization, *Proceedings of the IEEE international conference on neural networks*, 1942-1948.
- El Majdouli, M. A., Bougrine, S., Rboubh, I. ve El Imrani, A. A., 2016, A fireworks algorithm for single objective big optimization of signals, *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, 1-7.
- Elsayed, S. ve Sarker, R., 2015, An adaptive configuration of differential evolution algorithms for big data, *IEEE congress on evolutionary computation (CEC)*, *IEEE*, 702.
- Fister, I. ve Žumer, J. B., 2012, Memetic artificial bee colony algorithm for large-scale global optimization, *2012 IEEE Congress on evolutionary computation*, 1-8.
- Gandomi, A. H., Yang, X.-S., Talatahari, S. ve Alavi, A. H., 2013a, Firefly algorithm with chaos, *Communications in Nonlinear Science and Numerical Simulation*, 18 (1), 89-98.
- Gandomi, A. H., Yun, G. J., Yang, X.-S. ve Talatahari, S., 2013b, Chaos-enhanced accelerated particle swarm optimization, *Communications in Nonlinear Science and Numerical Simulation*, 18 (2), 327-340.
- García, S., Molina, D., Lozano, M. ve Herrera, F., 2009, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization, *Journal of Heuristics*, 15, 617-644.
- Goh, S. K., Abbass, H. A., Tan, K. C. ve Mamun, A. A., 2014, Artifact removal from EEG using a multi-objective independent component analysis model, *Neural Information Processing: 21st International Conference, ICONIP 2014, Kuching, Malaysia, November 3-6, 2014. Proceedings, Part I 21*, 570-577.
- Goh, S. K., Tan, K. C., Al-Mamun, A. ve Abbass, H. A., 2015, Evolutionary big optimization (BigOpt) of signals, *2015 IEEE Congress on Evolutionary Computation (CEC)*, 3332-3339.
- Goh, S. K., Abbass, H. A., Tan, K. C. ve Al-Mamun, A., 2016, Decompositional independent component analysis using multi-objective optimization, *Soft Computing*, 20, 1289-1304.
- Gong, W., Fialho, Á., Cai, Z. ve Li, H., 2011, Adaptive strategy selection in differential evolution for numerical optimization: an empirical study, *Information Sciences*, 181 (24), 5364-5386.
- Griewank, A. ve Toint, P. L., 1982, Partitioned variable metric updates for large structured optimization problems, *Numerische Mathematik*, 39 (1), 119-137.
- Hadi, A. A., Mohamed, A. W. ve Jambi, K. M., 2019, LSHADE-SPA memetic framework for solving large-scale optimization problems, *Complex & Intelligent Systems*, 5, 25-40.
- Hasan, S. K., Sarker, R., Essam, D. ve Cornforth, D., 2009, Memetic algorithms for solving job-shop scheduling problems, *Memetic Computing*, 1, 69-83.
- Henein, M. Y., Zhao, Y., Nicoll, R., Sun, L., Khir, A. W., Franklin, K., Lindqvist, P. ve Collaborators, G. R., 2011, The human heart: application of the golden ratio and angle, Elsevier. 150: 239-242.
- Hieu, D. V., Cho, Y. J. ve Xiao, Y. B., 2019, Golden ratio algorithms with new stepsize rules for variational inequalities, *Mathematical Methods in the Applied Sciences*, 42 (18), 6067-6082.
- Hu, X.-M., He, F.-L., Chen, W.-N. ve Zhang, J., 2017, Cooperation coevolution with fast interdependency identification for large scale optimization, *Information Sciences*, 381, 142-160.

- Huang, H., Lv, L., Ye, S. ve Hao, Z., 2019, Particle swarm optimization with convergence speed controller for large-scale numerical optimization, *Soft Computing*, 23, 4421-4437.
- Iosa, M., Morone, G. ve Paolucci, S., 2018, Phi in physiology, psychology and biomechanics: The golden ratio between myth and science, *Biosystems*, 165, 31-39.
- Ismkhan, H., 2017, Effective heuristics for ant colony optimization to handle large-scale problems, *Swarm and Evolutionary Computation*, 32, 140-149.
- Jian, J.-R., Zhan, Z.-H. ve Zhang, J., 2020, Large-scale evolutionary optimization: a survey and experimental comparative study, *International Journal of Machine Learning and Cybernetics*, 11 (3), 729-745.
- Jordehi, A. R., 2015, Chaotic bat swarm optimisation (CBSO), *Applied Soft Computing*, 26, 523-530.
- Kaur, G. ve Arora, S., 2018, Chaotic whale optimization algorithm, *Journal of Computational Design and Engineering*, 5 (3), 275-284.
- Kocer, H. G. ve Uymaz, S. A., 2018, A modified artificial algae algorithm for large scale global optimization problems, *International Journal of Intelligent Systems and Applications in Engineering*, 6 (4), 306-310.
- Koçer, H. G. ve Uymaz, S. A., 2021, A novel local search method for LSGO with golden ratio and dynamic search step, *Soft Computing*, 25 (3), 2115-2130.
- Koçer, H. G., Türkoğlu, B. ve Uymaz, S. A., 2023, Chaotic golden ratio guided local search for big data optimization, *Engineering Science and Technology, an International Journal*, 41, 101388.
- Korošec, P., Tashkova, K. ve Šilc, J., 2010, The differential ant-stigmergy algorithm for large-scale global optimization, *IEEE Congress on Evolutionary Computation*, 1-8.
- Koupaei, J. A., Hosseini, S. M. M. ve Ghaini, F. M., 2016, A new optimization algorithm based on chaotic maps and golden section search method, *Engineering Applications of Artificial Intelligence*, 50, 201-214.
- Krasnogor, N. ve Smith, J., 2005, A tutorial for competent memetic algorithms: model, taxonomy, and design issues, *IEEE Transactions on evolutionary computation*, 9 (5), 474-488.
- Kuyu, Y. Ç. ve Vatansever, F., 2018, The chaos-based approaches for actual metaheuristic algorithms.
- Lan, W., Ye, Z., Ruan, P., Liu, J., Yang, P. ve Yao, X., 2021, Region-focused memetic algorithms with smart initialization for real-world large-scale waste collection problems, *IEEE Transactions on evolutionary computation*, 26 (4), 704-718.
- LaTorre, A., Muelas, S. ve Peña, J.-M., 2011, A MOS-based dynamic memetic differential evolution algorithm for continuous optimization: a scalability test, *Soft Computing*, 15, 2187-2199.
- LaTorre, A., Muelas, S. ve Pena, J.-M., 2012, Multiple offspring sampling in large scale global optimization, *2012 IEEE Congress on Evolutionary Computation*, 1-8.
- LaTorre, A., Muelas, S. ve Pena, J.-M., 2013, Large scale global optimization: Experimental results with MOS-based hybrid algorithms, *2013 IEEE congress on evolutionary computation*, 2742-2749.
- LaTorre, A., Muelas, S. ve Pena, J.-M., 2015, A comprehensive comparison of large scale global optimizers, *Information Sciences*, 316, 517-549.
- Li, J.-q. ve Pan, Q.-k., 2015, Solving the large-scale hybrid flow shop scheduling problem with limited buffers by a hybrid artificial bee colony algorithm, *Information Sciences*, 316, 487-502.

- Li, X., Tang, K., Omidvar, M. N., Yang, Z., Qin, K. ve China, H., 2013, Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization, *gene*, 7 (33), 8.
- Lin, J.-Y. ve Chen, Y.-P., 2011, Analysis on the collaboration between global search and local search in memetic computation, *IEEE Transactions on evolutionary computation*, 15 (5), 608-623.
- Liu, C. ve Li, B., 2014, Memetic algorithm with adaptive local search depth for large scale global optimization, *2014 IEEE Congress on Evolutionary Computation (CEC)*, 82-88.
- Liu, H., Wang, Y., Liu, L., Gao, X.-Z. ve Cheung, Y.-m., 2017, A new grouping strategy-based hybrid algorithm for large scale global optimization problems, *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 171-172.
- Livio, M., 2008, The golden ratio: The story of phi, the world's most astonishing number, Crown, p.
- Long, W., Jiao, J., Liang, X. ve Tang, M., 2018, An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization, *Engineering Applications of Artificial Intelligence*, 68, 63-80.
- López, E. D., Puris, A. ve Bello, R. R., 2015, VMODE: a hybrid metaheuristic for the solution of large scale optimization problems, *Investigación Operacional*, 36 (3), 232-240.
- Lu, Y., Benlic, U. ve Wu, Q., 2020, An effective memetic algorithm for the generalized bike-sharing rebalancing problem, *Engineering Applications of Artificial Intelligence*, 95, 103890.
- Luo, J., Wang, Q. ve Xiao, X., 2013, A modified artificial bee colony algorithm based on converge-onlookers approach for global optimization, *Applied Mathematics and Computation*, 219 (20), 10253-10262.
- Lüttge, U. ve Souza, G. M., 2019, The Golden Section and beauty in nature: The perfection of symmetry and the charm of asymmetry, *Progress in biophysics and molecular biology*, 146, 98-103.
- Mahdavi, S., Shiri, M. E. ve Rahnamayan, S., 2015, Metaheuristics in large-scale global continues optimization: A survey, *Information Sciences*, 295, 407-428.
- Malitsky, Y., 2020, Golden ratio algorithms for variational inequalities, *Mathematical Programming*, 184 (1-2), 383-410.
- Manikantan, K., Arun, B. ve Yaradoni, D. K. S., 2012, Optimal multilevel thresholds based on Tsallis entropy method using golden ratio particle swarm optimization for improved image segmentation, *Procedia Engineering*, 30, 364-371.
- Marples, C. R. ve Williams, P. M., 2022, The Golden Ratio in Nature: A Tour Across Length Scales, *Symmetry*, 14 (10), 2059.
- Maučec, M. S., Brest, J. ve Bošković, B., 2018, Improved differential evolution for large-scale black-box optimization, *IEEE Access*, 6, 29516-29531.
- Maučec, M. S. ve Brest, J., 2019, A review of the recent use of Differential Evolution for Large-Scale Global Optimization: An analysis of selected algorithms on the CEC 2013 LSGO benchmark suite, *Swarm and Evolutionary Computation*, 50, 100428.
- Mirjalili, S. ve Gandomi, A. H., 2017, Chaotic gravitational constants for the gravitational search algorithm, *Applied Soft Computing*, 53, 407-419.
- Mohamed, A. W., 2017, Solving large-scale global optimization problems using enhanced adaptive differential evolution algorithm, *Complex & Intelligent Systems*, 3, 205-231.

- Mohamed, A. W. ve Almazyad, A. S., 2017, Differential evolution with novel mutation and adaptive crossover strategies for solving large scale global optimization problems, *Applied Computational Intelligence and Soft Computing*, 2017.
- Mohamed, A. W., Hadi, A. A., Fattouh, A. M. ve Jambi, K. M., 2017, LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems, *2017 IEEE Congress on evolutionary computation (CEC)*, 145-152.
- Mohamed, A. W., 2023, Optimization Project, <https://sites.google.com/view/optimization-project/files?pli=1>: [01.05.2023].
- Mohapatra, P., Das, K. N. ve Roy, S., 2017, A modified competitive swarm optimizer for large scale optimization problems, *Applied Soft Computing*, 59, 340-362.
- Mohapatra, P., Roy, S., Das, K. N., Dutta, S. ve Raju, M. S. S., 2022, A review of evolutionary algorithms in solving large scale benchmark optimisation problems, *International Journal of Mathematics in Operational Research*, 21 (1), 104-126.
- Molina, D., Lozano, M., Garcia-Martinez, C. ve Herrera, F., 2010a, Memetic algorithms for continuous optimisation based on local search chains, *Evolutionary computation*, 18 (1), 27-63.
- Molina, D., Lozano, M. ve Herrera, F., 2010b, MA-SW-Chains: Memetic algorithm based on local search chains for large scale continuous global optimization, *IEEE congress on evolutionary computation*, 1-8.
- Molina, D., Lozano, M., Sánchez, A. M. ve Herrera, F., 2011, Memetic algorithms based on local search chains for large scale continuous optimisation problems: MA-SSW-Chains, *Soft Computing*, 15, 2201-2220.
- Molina, D. ve Herrera, F., 2015, Iterative hybridization of DE with local search for the CEC'2015 special session on large scale global optimization, *2015 IEEE congress on evolutionary computation (CEC)*, 1974-1978.
- Molina, D., LaTorre, A. ve Herrera, F., 2018a, An insight into bio-inspired and evolutionary algorithms for global optimization: review, analysis, and lessons learnt over a decade of competitions, *Cognitive Computation*, 10, 517-544.
- Molina, D., LaTorre, A. ve Herrera, F., 2018b, SHADE with iterative local search for large-scale global optimization, *2018 IEEE congress on evolutionary computation (CEC)*, 1-8.
- Molina, D., Nesterenko, A. R. ve LaTorre, A., 2019, Comparing Large-Scale Global Optimization Competition winners in a real-world problem, *2019 IEEE Congress on Evolutionary Computation (CEC)*, 359-365.
- Moscato, P., 1989, On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms, *Caltech concurrent computation program, C3P Report*, 826 (1989), 37.
- Moscato, P., 1999, Memetic algorithms: A short introduction, *New ideas in optimization*, 219-234.
- Nelder, J. A. ve Mead, R., 1965, A simplex method for function minimization, *The computer journal*, 7 (4), 308-313.
- Neri, F., Toivanen, J. ve Mäkinen, R. A., 2007, An adaptive evolutionary algorithm with intelligent mutation local searchers for designing multidrug therapies for HIV, *Applied Intelligence*, 27, 219-235.
- Neri, F. ve Tirronen, V., 2009, Scale factor local search in differential evolution, *Memetic Computing*, 1, 153-171.
- Neri, F. ve Cotta, C., 2012, Memetic algorithms and memetic computing optimization: A literature review, *Swarm and Evolutionary Computation*, 2, 1-14.

- Omidvar, M. N., Li, X. ve Yao, X., 2010, Cooperative co-evolution with delta grouping for large scale non-separable function optimization, *IEEE congress on evolutionary computation*, 1-8.
- Omidvar, M. N., Li, X., Mei, Y. ve Yao, X., 2013, Cooperative co-evolution with differential grouping for large scale optimization, *IEEE Transactions on evolutionary computation*, 18 (3), 378-393.
- Omidvar, M. N., Kazimipour, B., Li, X. ve Yao, X., 2016, CBCC3—A contribution-based cooperative co-evolutionary algorithm with improved exploration/exploitation balance, *2016 IEEE congress on evolutionary computation (CEC)*, 3541-3548.
- Omidvar, M. N., Yang, M., Mei, Y., Li, X. ve Yao, X., 2017, DG2: A faster and more accurate differential grouping for large-scale black-box optimization, *IEEE Transactions on evolutionary computation*, 21 (6), 929-942.
- Omidvar, M. N., Li, X. ve Yao, X., 2021a, A review of population-based metaheuristics for large-scale black-box global optimization—Part II, *IEEE Transactions on evolutionary computation*, 26 (5), 823-843.
- Omidvar, M. N., Li, X. ve Yao, X., 2021b, A review of population-based metaheuristics for large-scale black-box global optimization—Part I, *IEEE Transactions on evolutionary computation*, 26 (5), 802-822.
- Ong, Y. S. ve Keane, A. J., 2004, Meta-Lamarckian learning in memetic algorithms, *IEEE Transactions on evolutionary computation*, 8 (2), 99-110.
- Potter, M. A. ve De Jong, K. A., 1994, A cooperative coevolutionary approach to function optimization, *Parallel Problem Solving from Nature—PPSN III: International Conference on Evolutionary Computation The Third Conference on Parallel Problem Solving from Nature Jerusalem, Israel, October 9–14, 1994 Proceedings 3*, 249-257.
- Pu, W., Sheng, D., Gao, X. ve Gao, H., 2018, Optimization of train energy saving based on golden ratio genetic algorithm, *2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 869-874.
- Qiu, H. ve Liu, Y., 2016, Novel heuristic algorithm for large-scale complex optimization, *Procedia Computer Science*, 80, 744-751.
- Rather, S. A. ve Bala, P. S., 2020, Swarm-based chaotic gravitational search algorithm for solving mechanical engineering design problems, *World Journal of Engineering*.
- Salcedo-Sanz, S., Camacho-Gómez, C., Molina, D. ve Herrera, F., 2016, A coral reefs optimization algorithm with substrate layers and local search for large scale global optimization, *2016 IEEE Congress on Evolutionary Computation (CEC)*, 3574-3581.
- Saremi, S., Mirjalili, S. ve Lewis, A., 2014, Biogeography-based optimisation with chaos, *Neural Computing and Applications*, 25, 1077-1097.
- Sayed, E., Essam, D., Sarker, R. ve Elsayed, S., 2015, Decomposition-based evolutionary algorithm for large scale constrained problems, *Information Sciences*, 316, 457-486.
- Sheikholeslami, R. ve Kaveh, A., 2013, A survey of chaos embedded meta-heuristic algorithms, *Int J Optim Civil Eng*, 3 (4), 617-633.
- Shekhawat, K., 2015, Why golden rectangle is used so often by architects: A mathematical approach, *Alexandria Engineering Journal*, 54 (2), 213-222.
- Shi, Y.-j., Teng, H.-f. ve Li, Z.-q., 2005, Cooperative co-evolutionary differential evolution for function optimization, *Advances in Natural Computation: First*

- International Conference, ICNC 2005, Changsha, China, August 27-29, 2005, Proceedings, Part II 1*, 1080-1088.
- Singh, A. ve Dulal, N., 2017, A survey on metaheuristics for solving large scale optimization problems, *Int. J. Comput. Appl*, 170 (5), 1-7.
- Smith, J. E., 2007, Coevolving memetic algorithms: A review and progress report, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37 (1), 6-17.
- Solis, F. J. ve Wets, R. J.-B., 1981, Minimization by random search techniques, *Mathematics of operations research*, 6 (1), 19-30.
- Storn, R. ve Price, K., 1997, Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces, *Journal of global optimization*, 11 (4), 341.
- Sun, Y., Kirley, M. ve Halgamuge, S. K., 2017a, A recursive decomposition method for large scale continuous optimization, *IEEE Transactions on evolutionary computation*, 22 (5), 647-661.
- Sun, Y., Kirley, M. ve Halgamuge, S. K., 2017b, A memetic cooperative co-evolution model for large scale continuous optimization, *Artificial Life and Computational Intelligence: Third Australasian Conference, ACALCI 2017, Geelong, VIC, Australia, January 31–February 2, 2017, Proceedings 3*, 291-300.
- Sun, Y., Li, X., Ernst, A. ve Omidvar, M. N., 2019, Decomposition for large-scale optimization problems with overlapping components, *2019 IEEE congress on evolutionary computation (CEC)*, 326-333.
- Talatahari, S., Azar, B. F., Sheikholeslami, R. ve Gandomi, A., 2012, Imperialist competitive algorithm combined with chaos for global optimization, *Communications in Nonlinear Science and Numerical Simulation*, 17 (3), 1312-1319.
- Tamargo, R. J. ve Pindrik, J. A., 2019, Mammalian skull dimensions and the golden ratio (Φ), *The Journal of Craniofacial Surgery*, 30 (6), 1750.
- Tanabe, R. ve Fukunaga, A., 2013, Evaluating the performance of SHADE on CEC 2013 benchmark problems, *2013 IEEE Congress on evolutionary computation*, 1952-1959.
- Tang, D., Cai, Y., Zhao, J. ve Xue, Y., 2014, A quantum-behaved particle swarm optimization with memetic algorithm and memory for continuous non-linear large scale problems, *Information Sciences*, 289, 162-189.
- Tang, K., Li, X., Suganthan, P. N., Yang, Z. ve Weise, T., 2009, Benchmark Functions for the CEC'2010 Special Session and Competition on Large-Scale.
- Tang, R., Fong, S. ve Dey, N., 2018, Metaheuristics and chaos theory, *Chaos theory*, 182-196.
- Tezuka, M., Munetomo, M. ve Akama, K., 2004, Linkage identification by nonlinearity check for real-coded genetic algorithms, *Genetic and Evolutionary Computation–GECCO 2004: Genetic and Evolutionary Computation Conference, Seattle, WA, USA, June 26-30, 2004. Proceedings, Part II*, 222-233.
- Tirronen, V., Neri, F., Kärkkäinen, T., Majava, K. ve Rossi, T., 2008, An enhanced memetic differential evolution in filter design for defect detection in paper production, *Evolutionary computation*, 16 (4), 529-555.
- Tseng, L.-Y. ve Chen, C., 2008, Multiple trajectory search for large scale global optimization, *2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence)*, 3052-3059.
- Türkoğlu, B., 2022, Veri bilimi ve mühendislik optimizasyon problemlerinin çözümü için yeni bir yaklaşım: Kaotik yapay alg algoritması.

- Uymaz, S. A., 2015, Yeni bir biyolojik ilhamlı metasezgisel optimizasyon metodu: Yapay alg algoritması.
- Uymaz, S. A., Tezel, G. ve Yel, E., 2015, Artificial algae algorithm (AAA) for nonlinear global optimization, *Applied Soft Computing*, 31, 153-171.
- UYMAZ, S. A., 2023, Projects, <http://sauymaz.pythonanywhere.com/>: [01.05.2023].
- Van den Bergh, F. ve Engelbrecht, A. P., 2004, A cooperative approach to particle swarm optimization, *IEEE Transactions on evolutionary computation*, 8 (3), 225-239.
- Wang, F., Wang, X. ve Sun, S., 2022, A reinforcement learning level-based particle swarm optimization algorithm for large-scale optimization, *Information Sciences*, 602, 298-312.
- Wang, G.-G., Guo, L., Gandomi, A. H., Hao, G.-S. ve Wang, H., 2014, Chaotic krill herd algorithm, *Information Sciences*, 274, 17-34.
- Wang, H., Wu, Z., Rahnamayan, S. ve Jiang, D., 2010, Sequential DE enhanced by neighborhood search for large scale global optimization, *IEEE congress on evolutionary computation*, 1-7.
- Wang, H., Sun, H., Li, C., Rahnamayan, S. ve Pan, J.-s., 2013, Diversity enhanced particle swarm optimization with neighborhood search, *Information Sciences*, 223, 119-135.
- Wang, L. ve Zhong, Y., 2015, Cuckoo search algorithm with chaotic maps, *Mathematical problems in engineering*, 2015.
- Wang, L. ve Feng, L., 2021, Guest Editorial: Special issue on memetic algorithms with learning strategy, *Memetic Computing*, 13 (2), 147-148.
- Wei, F., Wang, Y. ve Zong, T., 2014, Variable grouping based differential evolution using an auxiliary function for large scale global optimization, *2014 IEEE Congress on Evolutionary Computation (CEC)*, 1293-1298.
- Yang, M., Omidvar, M. N., Li, C., Li, X., Cai, Z., Kazimipour, B. ve Yao, X., 2016, Efficient resource allocation in cooperative co-evolution for large-scale global optimization, *IEEE Transactions on evolutionary computation*, 21 (4), 493-505.
- Yang, M., Zhou, A., Li, C., Guan, J. ve Yan, X., 2020, CCFR2: A more efficient cooperative co-evolutionary framework for large-scale global optimization, *Information Sciences*, 512, 64-79.
- Yang, Z., Tang, K. ve Yao, X., 2008a, Self-adaptive differential evolution with neighborhood search, *2008 IEEE congress on evolutionary computation (IEEE World Congress on Computational Intelligence)*, 1110-1116.
- Yang, Z., Tang, K. ve Yao, X., 2008b, Large scale evolutionary optimization using cooperative coevolution, *Information Sciences*, 178 (15), 2985-2999.
- Yang, Z., Tang, K. ve Yao, X., 2008c, Multilevel cooperative coevolution for large scale optimization, *2008 IEEE congress on evolutionary computation (IEEE World Congress on Computational Intelligence)*, 1663-1670.
- Yang, Z., Zhang, J., Tang, K., Yao, X. ve Sanderson, A. C., 2009, An adaptive coevolutionary differential evolution algorithm for large-scale optimization, *2009 IEEE Congress on Evolutionary Computation*, 102-109.
- Yıldırım, S., Kocer, H. E. ve Ekmekci, A. H., 2021, Quantitative analysis of EEG slow wave activity based on minpeakprominence method, *Traitement du Signal*.
- Zhang, J. ve Sanderson, A. C., 2009, JADE: adaptive differential evolution with optional external archive, *IEEE Transactions on evolutionary computation*, 13 (5), 945-958.
- Zhang, Y., Zhou, M., Jiang, Z. ve Liu, J., 2015, A multi-agent genetic algorithm for big optimization problems, *2015 IEEE congress on evolutionary computation (CEC)*, 703-707.

- Zhao, S.-Z., Liang, J. J., Suganthan, P. N. ve Tasgetiren, M. F., 2008, Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization, *2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence)*, 3845-3852.
- Zhao, S.-Z., Suganthan, P. N. ve Das, S., 2011, Self-adaptive differential evolution with multi-trajectory search for large-scale optimization, *Soft Computing*, 15, 2175-2185.

