



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



**MEYVE SİNEĞİ OPTİMİZASYON
ALGORİTMASININ PERFORMANSINI
İYİLEŞTİRMEK İÇİN YENİ YAKLAŞIMLAR**

HAZİM İŞCAN

DOKTORA TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Aralık-2019
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Hazim İŞCAN tarafından hazırlanan “Meyve Sineği Optimizasyon Algoritmasının Performansını İyileştirmek İçin Yeni Yaklaşımlar” adlı tez çalışması 24/12/2019 tarihinde aşağıdaki jüri üyeleri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda DOKTORA TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Prof.Dr. Mehmet ÇUNKAŞ

Danışman

Doç.Dr. Mesut GÜNDÜZ

Üye

Doç.Dr. Halife KODAZ

Üye

Dr.Öğr.Üyesi Ömer Kaan BAYKAN

Üye

Dr.Öğr.Üyesi Mehmet Akif ŞAHMAN

Yukarıdaki sonucu onaylarım.

Prof. Dr. Saadettin Erhan KESEN
Enstitü Müdürü

Bu tez çalışması Selçuk Üniversitesi Bilimsel Araştırma Projeleri Koordinatörlüğü tarafından 18101009 numaralı 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.



Hazim İŞCAN

Tarih: 24.12.2019

ÖZET

DOKTORA TEZİ

MEYVE SİNEĞİ OPTİMİZASYON ALGORİTMASININ PERFORMANSINI İYİLEŞTİRMEK İÇİN YENİ YAKLAŞIMLAR

Hazim İŞCAN

Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç.Dr. Mesut GÜNDÜZ

2019, 79 Sayfa

Jüri

Doç.Dr. Mesut GÜNDÜZ
Prof.Dr. Mehmet ÇUNKAŞ
Doç.Dr. Halife KODAZ
Dr.Öğr.Üyesi Ömer Kaan BAYKAN
Dr.Öğr.Üyesi Mehmet Akif ŞAHMAN

Optimizasyon problemlerinin çözümü son yıllarda dikkat çeken bir konu haline gelmiştir. Bu problemlerin çözümü için birçok meta-sezgisel yöntem geliştirilmiştir. Meta-sezgisel yöntemler optimum çözümü garanti etmezler. Meta-sezgisel yöntemler ile makul zamanda kabul edilebilir çözümler bulmak amaçlanır. Meta-sezgisel yöntemler çoğunlukla probleme özel olmazlar. Meta-sezgisel yaklaşımlar genel amaçlıdır, esnekler ve problemlere uyarlanabilirler. Meta-sezgisel yöntemler bu özelliklerinden dolayı optimizasyon problemlerinin çözümünde son yıllarda yoğun olarak kullanılmaktadır. Meyve Sineği Optimizasyon Algoritması (FOA) 2011 yılında sunulmuş bir meta-sezgisel algoritmadır. Meyve sineğinin yiyecek arama davranışından esinlenerek önerilmiştir. FOA basit yapılı, dizayn parametresi az, optimizasyon problemlerine kolay uyarlanabilir, anlaşılması ve programlanması kolay bir meta-sezgisel yaklaşımdır. Bu tür avantajları olmasına rağmen dezavantajları da mevcuttur. Lokal optimuma çabuk takılır. Karar fonksiyonu her zaman pozitifdir. Güncelleme stratejisi $[-1, 1]$ aralığında olduğu için küçüktür. Bu çalışmada, FOA'nın dezavantajlarını gidermek, algoritmanın performansını iyileştirmek ve daha kaliteli sonuçlar üretmesini sağlamak hedeflenmiştir. Bu amaçla FOA'da üç farklı geliştirme yapılmıştır. İlk geliştirmede FOA'ya işaret parametreleri ilave edilmiş ve SFOA olarak adlandırılmıştır. İkincide, FOA'nın karar verme stratejisi iki aşamalı hale getirilmiş ve saFOA olarak adlandırılmıştır. Üçüncüde, FOA'nın arama esnasında en kötü çözümlerinde dikkate alındığı iki farklı versiyon geliştirilmiş ve pFOA_v1 ve pFOA_v2 olarak adlandırılmıştır. Yeni önerilen FOA sürümlerinin performansı iyi bilinen 21 sayısal kıyaslama fonksiyonunda test edilerek araştırılmıştır. Elde edilen deneysel sonuçlar, literatürde iyi bilinen meta-sezgisel algoritmalar ile kıyaslanmıştır. Deneysel sonuçlar, önerilen FOA sürümlerinin sürekli optimizasyon problemleri için karşılaştırılabilir, başarılı ve rekabetçi sonuçlar ürettiğini göstermektedir.

Anahtar Kelimeler: Meta-sezgisel, meyve sineği optimizasyon algoritması, sezgisel algoritmalar, sürekli optimizasyon, sürü zekâsı.

ABSTRACT

PhD THESIS

NOVEL APPROACHES FOR PERFORMANCE IMPROVEMENT OF FRUIT FLY OPTIMIZATION ALGORITHM

Hazim İŞCAN

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

Advisor: Assoc.Prof.Dr. Mesut GÜNDÜZ

2019, 79 Pages

Jury

**Assoc.Prof.Dr. Mesut GÜNDÜZ
Prof.Dr. Mehmet ÇUNKAŞ
Assoc.Prof.Dr. Halife KODAZ
Assist.Prof.Dr. Ömer Kaan BAYKAN
Assist.Prof.Dr. Mehmet Akif ŞAHMAN**

The solution of optimization problems has become a subject of interest in recent years. Many meta-heuristic methods have been developed to solve these problems. Meta-heuristic methods do not guarantee an optimum solution. Meta-heuristic methods aim to find acceptable solutions in a reasonable time. Meta-heuristic methods are often not problem specific. Meta-heuristic approaches are general purpose, flexible and adaptable to problems. Meta-heuristic methods have been used extensively in the solution of optimization problems in recent years. Fruit Fly Optimization Algorithm (FOA) is a meta-heuristic algorithm introduced in 2011. Inspired by the fruit fly's foraging behavior. FOA is a simple structure, intuitive approach that is easy to understand and program, easy to adapt to optimization problems, with few design parameters. Although it has such advantages, it also has disadvantages. It has fast to the local optimum. The decision function is always positive. The update strategy is small because it is in the $[-1, 1]$ range. In this study, it is aimed to eliminate the disadvantages of the FOA, to improve the performance of the algorithm and to provide better quality results. For this purpose, three different improves have been made in FOA. In the first development, sign parameters were added to the FOA and called SFOA. In the second, the decision-making strategy of the FOA was made in two stages and called saFOA. In the third, two different versions have been developed in which FOA is considered in the worst-case solutions during the search and is named pFOA_v1 and pFOA_v2. The performance of the newly proposed FOA versions was tested in 21 well-known numerical benchmark functions. The experimental results are compared with the meta-heuristic algorithms which are well known in the literature. Experimental results show that the proposed FOA versions produce comparable, successful and competitive results for continuous optimization problems.

Keywords: Continuous optimization, fruit fly optimization algorithm, heuristic algorithms, Meta-heuristic, swarm intelligence.

ÖNSÖZ

Doktora çalışmamda bana yardımlarını ve desteğini eksik etmeyen danışman hocam Doç.Dr. Mesut GÜNDÜZ'e, çalışmama önerileriyle katkıda bulunan tez izleme komitesi hocalarım Prof.Dr. Mehmet ÇUNKAŞ ve Dr.Öğr.Üyesi Ömer Kaan BAYKAN'a, birlikte yaptığımız çalışmalarla bilgi ve tecrübelerini benden esirgemeyen Doç.Dr. Mustafa Servet KIRAN, Doç.Dr. İsmail BABAOĞLU ve Dr.Öğr.Üyesi Ahmet BABALIK'a teşekkür eder saygılarımı sunarım.

Maddi ve manevi desteklerini esirgemeyen babama, anneme, kardeşime, eşime, ve çocuklarıma teşekkür ederim.

Selçuk Üniversitesi Bilimsel Araştırma Projeleri Koordinatörlüğü'ne 18101009 numaralı proje desteğinden dolayı teşekkür ederim.

Hazim İŞCAN
KONYA-2019

İÇİNDEKİLER

ÖZET	iv
ABSTRACT	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
SİMGELER VE KISALTMALAR	viii
1. GİRİŞ	1
1.1. Tezin Amacı	1
1.2. Tezin Literatüre Katkısı	1
1.3. Tezin Organizasyonu	2
2. KAYNAK ARAŞTIRMASI	4
3. MATERYAL VE YÖNTEM	13
3.1. Meyve Sineği Optimizasyon Algoritması (FOA)	13
3.2. Benzitilmiş Tavlama Algoritması	19
3.3. Jaya Algoritması	19
3.4. İşaret Parametreleri Kullanılarak Geliştirilen Meyve Sineği Optimizasyon Algoritması (SFOA) (Önerilen Algoritma 1)	20
3.5. Benzitilmiş Tavlama Kullanılarak Geliştirilen Meyve Sineği Optimizasyon Algoritması (saFOA) (Önerilen Algoritma 2)	25
3.5. Yeni Bir Aday Çözüm Stratejisi Kullanılarak Geliştirilen Meyve Sineği Optimizasyon Algoritması (pFOA) (Önerilen Algoritma 3)	27
3.6. Deneyleerde Kullanılan Kıyaslama Fonksiyonları	31
4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA	34
4.1. SFOA'nın Deneysel Sonuçları.....	34
4.2. saFOA Algoritmasının Deneysel Sonuçları.....	43
4.3. pFOA Algoritmasının Deneysel Sonuçları	48
5. SONUÇLAR VE ÖNERİLER	70
5.1 Sonuçlar	70
5.2 Öneriler	72
KAYNAKLAR	73
ÖZGEÇMİŞ	77

SİMGELER VE KISALTMALAR

Kısaltmalar

- ABC : Yapay Arı Kolonisi Algoritması
ACO : Karınca Kolonisi Optimizasyonu
AFSA : Yapay Balık Sürüsü Algoritması
BFOA : Bakteriyel Besin Arama Optimizasyon Algoritması
CS : Guguk Kuşu Algoritması
CSO : Kedi Sürüsü Algoritması
FA : Ateşböceği Algoritması
FOA : Meyve Sineği Optimizasyon Algoritması
GA : Genetik Algoritma
GSO : Ateşböceği Sürü Optimizasyonu
PID : Oransal İntegral Türevsel Denetleyici
pFOA : Yeni bir aday çözüm stratejisi kullanılarak geliştirilen Meyve Sineği Optimizasyon Algoritması
PSO : Parçacık Sürü Optimizasyonu
SA : Benzetilmiş Tavlama
saFOA : Benzetilmiş Tavlama Kullanılarak Geliştirilen Meyve Sineği Optimizasyon Algoritması
SFOA : İşaret Parametreleri Kullanılarak Geliştirilen Meyve Sineği Optimizasyon Algoritması
TSA : Ağaç Tohum Algoritması
WCA : Kurt Koloni Algoritması

1. GİRİŞ

1.1. Tezin Amacı

Doğada bulunan canlıların zeki davranışları bilim insanlarının dikkatini çekmiştir. Karıncaların, arıların ve sineklerin yiyecek araması, kuşların havadaki uçuş düzeni, balıkların sürüler halinde birlikte yüzmesi bu davranışlara örnek olarak verilebilir. Bilim insanları canlıların bu davranışlarını model olarak alıp sürü zekâsına dayalı algoritmalar geliştirmişler ve bu algoritmaları optimizasyon problemlerinin çözümünde kullanmışlardır. Karınca Kolonisi Optimizasyonu (Ant Colony Optimization-ACO), Parçacık Sürü Optimizasyonu (Particle Swarm Optimization-PSO), Yapay Arı Kolonisi Optimizasyonu (Artificial Bee Colony-ABC), Ateşböceği Algoritması (Firefly Algorithm-FA), Ateşböceği Sürü Optimizasyonu (Glowworm Swarm Optimization – GSO), Yapay Balık Sürüsü Algoritması (Artificial Fish Swarm Algorithm-AFSA), Bakteriyel Besin Arama Optimizasyon Algoritması (Bacterial Foraging Optimization Algorithm-BFOA), Kurt Koloni Algoritması (Wolf Colony Algorithm-WCA) ve Kedi Sürüsü Algoritması (Cat Swarm Optimization-CSO) sürü zekâsı olarak geliştirilen algoritmalarındandır. Sürü zekâsı yöntemlerinin ortak özelliği, çözüm uzayını araştırırken birden fazla yapay ajanın çalışması ve optimum sonuca varabilmek için aralarında bir veya daha fazla ilişki bulunmasıdır. Bu ilişki sayesinde bir amaca yönelik davranışı gerçekleştirir ve hedefe ulaşmak için birlikte çalışır. Kolaylıkla gözlenebilen bu “kolektif zekâ” ajanlar arasında sık tekrarlanan davranışlardan doğmaktadır. Ajanlar faaliyetlerini idare etmek için basit bireysel kurallar kullanmakta ve grubun kalan kısmıyla etkileşerek sürü amaçlarına ulaşmaktadır.

Bu tez çalışmasında sürü zekâsı algoritmaları kapsamında yeni literatüre girmiş olan Meyve Sineği Optimizasyon Algoritmasının (Fruit Fly Optimization Algorithm-FOA) yeteneklerinin iyileştirilmesi, dezavantajlarının giderilmesi hedef alınarak optimizasyon problemlerinin daha başarılı bir şekilde çözülmesi amaçlanmaktadır.

1.2. Tezin Literatüre Katkısı

FOA'nın anlaşılması ve program koduna dönüştürülmesi kolaydır. Aynı zamanda sonuca hızlı bir şekilde yakınsamaktadır. Bu tür avantajları olmasına rağmen algoritmanın dezavantajları da bulunmaktadır. Sürünün ilk yerleşimi iyi bir dağılım

göstermemektedir. Güncelleme stratejisi iterasyon boyunca her zaman $[-1,1]$ aralığında olduğu için sabit ve küçüktür. Uygunluk (fitness) fonksiyonu her zaman pozitifdir. Her iterasyonda sadece en iyi çözüm dikkate alındığı için açgözlü bir yaklaşım içerisindedir. Bu nedenlerden dolayı yerel optimuma takılabilmektedir. Yapısı sürekli optimizasyon problemleri için uygundur.

Bu çalışmada yukarıda bahsedilen dezavantajların giderilmesi için FOA'da iki farklı geliştirme yapılmıştır.

İlk geliştirmede FOA'ya, sadece arama alanının pozitif tarafını değil, aynı zamanda tümünü değerlendirebilmesi için iki işaret parametresi eklenmiştir.

İkinci geliştirmede ise arama esnasında sadece en iyi çözümleri değil aynı zamanda en kötü çözümleri de dikkate alan yeni bir aday çözüm mekanizması kullanılmıştır.

Bu geliştirmeler sonucunda bu çalışmanın literatüre sağladığı katkılar aşağıda verilmiştir.

- Uygunluk fonksiyonu negatif değerler alabilmektedir.
- Arama uzayının sadece pozitif kısmını değil tamamını araştırabilmektedir.
- Çözüm alanı etkili bir şekilde aranabilmektedir.
- Genel arama yeteneği gelişmiştir.
- Sürekli optimizasyon problemlerini çözme konusundaki performansı artmıştır.
- Rekabetçi ve karşılaştırılabilir çözümler üretebilmektedir.
- Çok modlu fonksiyonların çözümünde temel versiyona göre daha etkilidir.

Bahsedilen bu katkılar ile bu çalışma, temel FOA'nın gelişmesine, daha verimli kullanılmasına, optimizasyon problemlerinde daha iyi sonuçlar üretmesine ve literatürdeki öneminin artmasına neden olacaktır.

1.3. Tezin Organizasyonu

Bu tez çalışması 5 bölümden oluşmaktadır. Giriş bölümünde, tezin amacı, literatüre katkısı ve organizasyonu hakkında bilgi verilmiştir.

İkinci bölümde, literatürde yapılan çalışmalar ile ilgili bilgi verilmiştir.

Üçüncü bölümde, Temel FOA, Benzetilmiş Tavlama Algoritması, JAYA ve bu çalışmada geliştirilen üç algoritma ayrıntılı bir şekilde anlatılmıştır. Ayrıca bu çalışma için kullanılan kıyaslama fonksiyonları hakkında bilgi verilmiştir.

Dördüncü bölümde, bu çalışmadaki iyileştirmelerin, diğer algoritmalarla birlikte kıyaslama fonksiyonlarındaki karşılaştırmaları ve sonuçları hakkında bilgi verilmiştir. Beşinci bölümde, sonuçlar değerlendirilmiş ve önerilerde bulunulmuştur.



2. KAYNAK ARAŞTIRMASI

Bu bölümde FOA'nın geliştirilmiş versiyonları ile ilgili literatürde yapılan çalışmalar hakkında bilgi verilmiştir.

Fuqiang Xu ve Youtian Tao tarafından temel FOA ya çift değişkenli fonksiyonlarda kullanım için iyileştirme yapılan bir algoritma önerilmiştir. Temel FOA da S değeri, uzaklıkların karelerinin toplamının karekökünün tersi olduğu için hiçbir zaman negatif olmaz ve bu durum negatif değer alması gereken bazı problemlerde kullanılmasını engeller. Bu durumundan kurtulmak için algoritmada epsilon parametresi getirilmiştir. Epsilon parametresi koku konsantrasyon yargı değeri hesabında kullanılmıştır ve $[-1 \ 1]$ aralığında rastgele bir değerdir. Çift değişkenli kullanım içinde x_{axis} ve y_{axis} değerlerini iki boyutlu almış ve her iki boyut için koku konsantrasyon yargı değerini ayrı hesaplayarak bunu fonksiyonda yerine koymuştur. Algoritma 3 adet doğrusal olmayan fonksiyon kullanılarak test edilmiştir. Karşılaştırma sonuçlarına göre algoritmanın hesaplama süresi ve yakınsaması PSO, GA ve SA'dan daha iyidir. Yerel optimumlara takılmadan optimum sonucu hızlıca bulmaktadır. Temel FOA'nın uygulanabilirliğini genişletmiş ve daha kararlı hale getirmiştir (Xu ve Tao, 2013).

Liu ve arkadaşları modifiye edilmiş bir FOA'yı önermişlerdir. Bu çalışmanın yazarlarına göre, FOA'nın güncelleme stratejisi güçlü bir rastgelelik içerir ve bu güçlü rastgelelik nedeniyle çözüm güncellemede körlükler oluşur, yakınsama oranı yavaşlar ve istenmeyen sonuçlar verir. Bu duruma çözüm bulmak için bir modifiye gerçekleştirmişlerdir. Modifiye için Kaos arama, PSO ve SA'dan faydalanılmıştır. Kaos arama, sonucu en iyiye götürme ihtimali için başlangıç çözümün özelliklerini değiştirmede kullanılmıştır. Böylece başlangıçtaki arama çeşitliliği artırılmak istenilmiştir. PSO arama güncellemedeki kör aramayı azaltmak ve çözüm güncellemesini düzeltebilmek için kullanılmıştır. SA ise yerel aramada yakınsama oranını artırmak için kullanılmıştır. Modifiye edilen bu metodun verimliliğini test etmek için bir iki boyutlu rastrigin fonksiyonu ve PID (Proportional Integral Devices) kontroller kullanmışlardır. Simülasyon sonuçlarına göre modifiye edilmiş FOA, temel FOA'dan daha kaliteli uygunluk değerleri üretmiş, daha hızlı, daha verimli ve daha güçlüdür. (Liu ve ark., 2012).

Li ve arkadaşları tarafından modifiye edilmiş FOA geliştirilmiştir. Geliştirilen algoritma, PID kontrol parametrelerinin bulunmasında kullanılmıştır. Temel FOA'nın daha iyi performanslı olması, daha iyi bir çözüm sunması, daha hızlı ve daha kararlı

olabilmesi için algoritmayı geliştirmişlerdir. Bu çalışmanın yazarlarına göre temel FOA'da mesafe (distance) değerlerinin geniş bir alana yayılması koku konsantrasyon yargı değerini oldukça küçültmektedir. Bu değer küçük olması ile elde edilen uygunluk değerinin erken yakınsama olasılığına ve yerel optimuma takılmasına neden olmaktadır. Bu durumu aşabilmek için koku konsantrasyon yargı değerine yerelden kaçınma parametresi ilave edilmiştir. Algoritma, temel FOA ve PSO ile test edilmiş ve karşılaştırılmıştır. Karşılaştırmalar sonucu algoritmanın FOA'dan daha hızlı ve daha efektif, PSO'dan ise daha hızlı PID kontrol parametreleri ürettiği görülmüştür. Dolayısıyla algoritma, FOA'dan daha hızlı ve daha kararlı, PSO'dan daha performanslıdır (Li ve ark., 2012a).

Wang ve arkadaşları tarafından çok boyutlu sırt çantası probleminin çözümü için yeni bir FOA önerilmiştir. Temel FOA sürekli optimizasyon problemlerinin çözümünde kullanılmaktadır. Bu çalışmanın yazarları temel FOA'nın ayrık optimizasyon problemlerinin çözümünde de kullanılabilmesini hedeflemiş ve FOA'yı geliştirmiştir. Temel FOA'da yapılan geliştirmeler koku tabanlı arama, yerel görme tabanlı arama ve genel görme tabanlı aramadır. Başlangıçta bireyler ikili dizi (binary string) ile ifade edilir. Aramayı zenginleştirmek için yeni bireyler üretirken grup üretme olasılıksal vektörü kullanılır. Koku tabanlı aramada yerel komşuluklardan yararlanılmıştır. Yerel görme tabanlı aramada meyve sinekleri kendilerinin en iyi komşularıyla yer değiştirir. Genel görme tabanlı aramada, aramayı zenginleştirmek amacıyla yeni bireyler üretirken grup üretme olasılıksal vektörü güncellenir. İlave olarak bireylerin ve çözümün fizibilitesini garanti etmek için iki tane onarım operatörü kullanılır. Test için iki grup test örnekleri kullanılmıştır. Birincisinde 8 adet küçük büyüklükte problem, ikincisinde 11 adet orta ve geniş büyüklükte problem bulunmaktadır. Karşılaştırma sonuçlarına göre geliştirilen algoritma, küçük ve orta büyüklükteki problemlerin çözümünde iyidir. Özellikle geniş büyüklükteki problemlerin çözümünde ise daha iyidir. Bu da algoritmanın başarılı bir algoritma olduğunu göstermektedir (Wang ve ark., 2013).

Shan ve arkadaşları tarafından temel FOA'nın performansını iyileştirmek için bir geliştirme yapılmıştır. Temel FOA karmaşık optimizasyon problemlerini efektif olarak çözememektedir. Temel FOA doğrusal olmayan aday üretim mekanizmasına sahiptir. Bu mekanizma, FOA'nın performansını kısıtlayacak bazı dezavantajlara sahiptir. Koku konsantrasyon yargı değeri her zaman 0'dan büyüktür. Negatif değerler için bu bir sorundur. X_{axis} ye Y_{axis} değerleri sabit olduğunda koku konsantrasyon yargı değeri düzenli dağılım göstermez ve çözümler düzenli üretilmez. Bu da genel çözüm arama

kabiliyetini kaybettirir. X_{axis} ve Y_{axis} değerleri büyük, güncelleme değeri küçük olduğunda koku konsantrasyon yargı değerinin değişimi azalır ve yerel optimuma takılabilir. Güncelleme için eklenen rastgele değer artması X_{axis} ve Y_{axis} değerini büyük şekilde artırabilir ve koku konsantrasyon yargı değeri, 0 noktasına düşebilir. FOA'da S değeri her zaman 0'dan büyüktür. Bu dezavantajları kaldırmak için yeni bir doğrusal aday üretme mekanizması geliştirilmiştir. Bu mekanizma ile koku konsantrasyon yargı değeri tüm alanı kapsar. Lokasyon yeri sabit olsa da koku konsantrasyon yargı değeri düzenli dağılım gösterir. Önerilen yöntemde atalet ağırlığı adı verilen bir parametre kullanılır. Bu parametre yerel ve genel arama esnasında büyükten küçüğe doğru azalarak arama kabiliyetini artırır. 6 tane karmaşık fonksiyon ile karşılaştırmalar yapılmıştır. Karşılaştırmalar sonucunda yöntem, FOA'ya göre daha efektif ve daha güvenilirdir. PSO ve GA'dan da arama kalitesi, performans ve güvenilirlik olarak daha iyi olduğu söylenebilir (Shan ve ark., 2013).

Ramachandran ve Bellarmine tarafından temel FOA çok boyutlu hale getirilmiş ve bazı iyileştirmeler yapılmıştır. Temel FOA sadece iki boyutlu parametre optimizasyon problemleri için önerilmiştir. Bu çalışmada FOA modifiye edilerek çok boyutlu hale getirilmiştir. Bunu yaparken kâşif sineklerin belli bir kısmından elit sinekler ve en iyi sinekler oluşturulmuştur. Bu sineklerin hızları komşuluk arama tekniğine göre güncellenmiştir. Bir diğer geliştirme de adım sayısında olmuştur. Temel FOA'da adım uzunluğu sabittir. Adım uzunluğunun sabit olması optimuma ulaşmasını zorlaştırmakta ve yakınsama süresini artırmaktadır. Bu çalışmada adım uzunluğu iterasyona bağımlıdır ve iterasyon arttıkça azalır. Buda çözümün kesinliğini sağlar ve yerel optimuma takılmasını engeller. Bu çalışmada enerji nakil şebekelerindeki fazör ölçüm ünitelerinin optimum yerleşimi, temel FOA çok boyutlu hale getirilerek yapılmıştır. Simülasyon sonuçlarına göre gerçekleştirilen metot, çözümleri daha küçük arama uzayında bulmuş, yakınsamayı hızlandırmış ve fazör ölçüm ünitelerinin optimum yerleşimini yapabilmıştır (Ramachandran ve Bellarmine, 2014).

Pan tarafından modifiye edilmiş bir FOA sunulmuştur. Temel FOA'nın yerel optimumlara takılma ihtimali bulunmaktadır. Bu nedenle FOA'nın genel optimumları bulması zor olmaktadır. Bu FOA'nın uygunluk fonksiyonundan kaynaklanır. FOA'da arama adımları küçük ve sınırlı olduğundan uygunluk fonksiyonununun değişmesi gerekmektedir. FOA'da uygunluk fonksiyonu her zaman pozitifdir. Geliştirilen algoritmada bu sorunlar için bir modifiye yapılmıştır. Genel çözüm bulurken yerel çözümlere takılmaktan kurtulmak için bir kaçış parametresi uygunluk fonksiyonuna ilave

edilmiştir. Bu kaçış parametresi aynı zamanda uygunluk değerinin negatif olabilmesini de sağlamaktadır. FOA aramayı iki boyutta yaptığı için üç boyutlu optimum değerleri bulmada zorlanır. Bu nedenle bu algorithmada genel optimumu bulurken Z eksenini de kullanılmıştır. Önerilen algoritma 14 tane kıyaslama fonksiyonları kullanılarak test edilmiştir. Önerilen algoritma, PSO, AFSA ve FOA ile karşılaştırılmıştır. Önerilen algoritma diğer üç algoritmaya göre yüksek hesaplama doğruluğuna sahiptir ve çalışma süresi oldukça kısadır. FOA ve AFSA'dan daha kararlıdır. İlave olarak bu dört metot genel regresyon sinir ağı ile birlikte kullanılarak test edilmiştir. Bu testler sonucunda önerilen algoritmanın genel regresyon sinir ağı modeli diğer modellere göre daha yüksek tahmin kabiliyetine sahiptir ve daha hızlıdır (Pan, 2013a).

Dai ve arkadaşları tarafından temel FOA'ya bir geliştirme yapılmıştır. Temel FOA da koku konsantrasyon yargı değeri her zaman pozitifdir. Koku konsantrasyon yargı değerinin pozitif olması FOA'nın bazı problemlerin değişkenlerine uygulanmasını mümkün kılmamaktadır. Bu çalışmada bu problemi ortadan kaldırmak için bir geliştirme yapılmıştır. Temel FOA'da koku konsantrasyon yargı değeri hesaplandıktan sonra uygunluk fonksiyonuna girmektedir. Bu çalışmada ise bu iki adım arasına yeni bir adım eklenir. Bu adımda meyve sineğinin konumu koordinat sisteminin dörtte biri düşünülerek değerlendirilir. Eğer konumu 2 ve 4 numaralı çeyrekte ise koku konsantrasyon yargı -1 ile çarpılır, değilse aynen alınır. Bu da geliştirilmiş FOA'nın efektif bir geliştirme olduğunu göstermektedir (Dai ve ark., 2014).

Pan ve arkadaşları tarafından geliştirilmiş bir FOA sunulmuştur. Temel FOA'nın arama uzayını daha iyi arayabilmesi için bir kontrol parametresi eklenmiştir. Aynı zamanda algoritmanın doğruluğunu ve yakınsama oranını geliştirmek için yeni bir çözüm üretme metodu sunulmuştur. Temel FOA'nın koku arama fazında yeni çözümler mevcut çözüme $[-1 \ 1]$ aralığında rastgele değerler eklemektedir. Bu çap, yineleme boyunca değişmemektedir. Bu çap optimumu bulabilmek için ilk yinelemelerde çok küçük kalmakta, son yinelemelerde ise büyük kalmaktadır. FOA'nın bu problemini ortadan kaldırabilmek için bu çap yineleme sayısına göre dinamik bir şekilde değiştirilmiştir. İlave olarak iyi bir başlangıç bölgesi için rastgele popülasyon üretilmiş ve onların arasındaki en iyi meyve sineği başlangıç yeri olarak seçilmiştir. FOA'da her bir çözümde karar değişkenleri yenilenmektedir. Önerilen algorithmada ise yoğun aramayı geliştirmek için rastgele bir tane karar değeri seçilerek arama yapılmaktadır. Algoritmanın performansı için literatürde en çok kullanılan 29 adet kıyaslama fonksiyonu test edilmiştir. Bu fonksiyonların 15 adedi tek modlu 14 adedi çok modlu fonksiyondur.

Algoritma yukarıda belirtilen fonksiyonlar kullanılarak FA ve Harmoni aramanın 5 türevi ile karşılaştırılmıştır. Karşılaştırmalar sonucunda önerilen algoritmanın, FOA ve 5 adet harmoni arama türevinden daha iyi sonuçlar elde ettiği görülmüştür. Ayrıca popülasyon boyutu içinde testler yapılmış ve nesil sayısının (5-10 arası) düşük olması tavsiye edilmiştir (Pan ve ark., 2014).

Li ve Lu, yapısal hasar tespiti için birkaç doğal frekans ve mod şeklini kullanan çok sürümlü bir FOA geliştirmiştir. Objektif fonksiyon için doğal frekanslar ve bir yapının hasar görmüş ve bozulmamış durumundaki mod şekilleri arasındaki farkları kullanmışlardır. Geliştirdikleri algoritmanın etkinliğini ve doğruluğunu üç farklı yapı ile test etmişler ve temel FOA'dan daha iyi bir kapasiteye sahip olduğunu göstermişlerdir (Li ve Lu, 2015).

Wu ve arkadaşları temel FOA'nın yakınsama performansını artırmak için, bu FOA'ya dayalı normal bir bulut modeli önermiştir. Koku fazında meyve sineği sürüsünün toplayıcı davranışının rastlantısallığı ve bulanıklığı normal bulut modeli ile tanımlanmıştır. Ayrıca, normal bulut modelinde rastsallık için uyarlanabilir bir parametre stratejisi, erken aşamada, küresel arama yeteneğini geliştirmek ve son aşamada ise çözümün doğruluğunu artırmak için kabul edilmiştir. Önerilen yöntemin etkinliğini test etmek için 33 kıyaslama fonksiyonu kullanılmıştır. Sayısal sonuçlar, önerilen algoritmanın, literatürlerdeki üç gelişmiş FOA ve yedi adet akıllı optimizasyon algoritması teknolojisiyle karşılaştırıldığında, çoğu test fonksiyonu için daha iyi veya rekabetçi performans elde edebileceğini göstermiştir (Wu ve ark., 2015).

Niu ve arkadaşları temel FOA'ya, özellikle sıfır olmayan optimal çözümlerde daha evrensel çözümler üretebilmesi için rastlantısal aramanın yerine bir diferansiyel vektör ekleyerek, diferansiyel evrime dayanan geliştirilmiş bir meyve sineği optimizasyon algoritması önermiştir. Önerilen algoritma 12 kıyaslama fonksiyonunda test edilmiştir. Deneysel sonuçlar, geliştirilmiş FOA'nın temel FOA ve literatürdeki evrimsel algoritmalara göre daha güçlü bir genel arama yeteneğine ve daha hızlı yakınlaşmaya sahip olduğunu göstermiştir. Ayrıca geliştirilmiş FOA, iki karar değişkeni kullanarak gaz verimini en üst düzeye çıkarmak suretiyle Texaco gazlaştırma işleminin çalışmasını optimize etmek için de uygulanmış ve en iyi verimi hızla alabildiğini göstermiştir (Niu ve ark., 2015).

Sun ve Ye, kısa vadeli elektrik güç yükü tahmini için dalgacık dönüşümü temelli yeni bir model ve FOA ile optimize edilmiş en küçük kareler destek vektör makinesi önermiştir. Dalgacık dönüşümü, hata noktalarından uzaklaşmak ve verilerin kararlılığını

arttırmak için kullanılır. FOA, en küçük kareler destek vektör makinesinin parametrelerini optimize etmek, rastgele ayar yapmaktan ve parametre ayarlarındaki yanlışlıktan kaçınmak için uygulanmıştır (Sun ve Ye, 2015).

Zhang ve arkadaşları, FOA'nın, sınırlı arama alanı ve sürü çeşitliliğinin, küresel arama yeteneğini zayıflattığından bahsetmiş ve bu sınırlamanın üstesinden gelmek için, yeni birçok ölçekli kooperatif mutasyon FOA önermiştir. Algoritmanın etkinliği 29 kıyaslama fonksiyonu ile değerlendirilmiştir. Algoritma, literatürde son sunulan FOA'nın geliştirilmiş versiyonlarından çoğu kıyaslama fonksiyonunda daha iyi performans göstermiştir (Zhang ve ark., 2016a).

Zhang ve arkadaşları FOA'nın avantajlarını korurken dezavantajları ile başa çıkmak için uyarlamalı bir FOA önermiştir. Algoritma, sürünün her yinelemesinin tarihsel hafızasına göre sürü mesafesi parametresini dinamik olarak ayarlar ve daha doğru seçkin stratejiyi benimser, bu da sürünün genel optimal cepheye yaklaşmasını hızlandırmada ve çözümlerin çeşitliliğini korumada çok etkilidir. Önerilen algoritma, farklı özelliklere sahip 14 kıyaslama fonksiyonu ile test edilmiş ve sonuçlar, algoritmanın genel optimizasyon yeteneği ve yüksek evrenselliği olan yeni bir algoritma olduğunu göstermiştir (Zhang ve ark., 2016b).

Lin, yeni bir değiştirilmiş FOA önermiştir. Önerilen algoritmanın arama verimliliğinin ve kalitesinin temel FOA, GA ve PSO'dan üstün olduğunu göstermek için otuzdan fazla farklı doğrusal olmayan fonksiyonda test edilmiştir. Aynı zamanda bazı ekonomi konularına, iki teorik örneğe ve bir vaka çalışmasına uygulanmıştır (Lin, 2016).

Si ve arkadaşları, en küçük kareler destek vektör makinesinin parametrelerini optimize etmek için geliştirilmiş bir FOA sunmuş ve kesici kesme düzenini tanımlamak için kullanmıştır. Beş kesme düzeninin titreşim hızlanma sinyalleri toplanmış ve özel durum özellikleri çıkarılmıştır. Önerilen algoritma, PSO, GA ve FOA modelleri ile detaylı olarak karşılaştırılmıştır. Karşılaştırma sonuçları, önerilen yaklaşımın, uygulanabilir, verimli ve diğerlerinden daha iyi performansı olduğunu göstermiştir. Önerilen sistemin etkisi kömür madenciliğindeki endüstriyel bir uygulama örneğiyle gösterilmiştir (Si ve ark., 2016).

Meng ve Pan, çok boyutlu sırt çantası problemini çözmek için geliştirilmiş bir FOA önermiştir. Algoritmada, paralel arama, sömürü ve araştırmayı dengelemek için kullanılmıştır. Sürü zekâsından tam olarak yararlanmak için sürüler arasındaki işbirliğini artırmak amacıyla değiştirilmiş bir uyum arama algoritması önerilmiş ve uygulanmıştır. Uyum arama algoritmasında, sırt çantası problemi ve FOA'nın özel nitelikleri göz önüne

alınarak rastgele seçim kuralı geliştirilmiştir. Ayrıca, yerel optimuma takılmamak ve performansı daha da geliştirmek için dikey bir geçit tasarlanmıştır. Kapsamlı sayısal simülasyonlar gerçekleştirilmiş ve diğer modern algoritmalarla yapılan karşılaştırmalar, önerilen algoritmanın sırt çantası problemini çözmek için etkili bir alternatif olduğunu doğrulamıştır (Meng ve Pan, 2017).

Wang ve arkadaşları, doğrusal olmayan iki seviyeli programlama problemini ele almak için bir iki seviyeli geliştirilmiş FOA önermiştir. Bu algoritma iki tane geliştirilmiş FOA birleştirilerek oluşturulmuştur. Önerilen algoritmada, alt seviye problemi, üst seviye problemin sınırlarına dönüştürülmek yerine, doğrusal olmayan bir programlama problemi olarak ele alınmıştır. Algoritmanın performansını, çözümlerin doğruluğu ve kararlılığı yönünden değerlendirmek için düşük boyutlu ve yüksek boyutlu sorunları içeren 10 test problemi seçilmiştir. Sonuçlar algoritmanın, karşılaştırılan algoritmaları geride bıraktığını ve literatürde sunulan yöntemlerden önemli ölçüde daha iyi olduğunu ortaya koymuştur. Önerilen algoritma doğrusal olmayan iki seviyeli programlama problemi için etkili ve karşılaştırılabilir bir algoritmadır (Wang ve ark., 2017).

Yang ve arkadaşları, FOA'ya ve dalgacık sinir ağına dayanan yeni bir hibrit model önermiştir. Önerilen hibrit tahmin modeli, dalgacık sinir ağının ağırlıkları ve eşikleri optimize edilerek oluşturulmuş ve model, sualtı akustik sinyal tahminine uygulanmıştır. Deneysel sonuçlar, önerilen hibrit tahmin modelinin dalgacık sinir ağı tahmin modeli ve geri yayılım sinir ağı tahmin modeline kıyasla daha yüksek tahmin doğruluğu ve daha küçük tahmin hatası olduğunu göstermiştir.(Yang ve ark., 2018)

Li ve arkadaşları, temel FOA algoritmasına kedi haritalaması eklemiş ve arama hızı ve doğruluğunu artırmak için meyve sineği popülasyonunun bireysel dağıtım ve evrim mekanizmasını geliştirmiştir. Algoritma kıyaslama fonksiyonlarında test edilmiştir. Elde edilen sonuçlar, geliştirilmiş algoritmanın daha hızlı ve daha güvenilir optimizasyon yeteneğine sahip olduğunu göstermiştir. Algoritma daha sonra seyrek ayrıştırma teorisi (sparse decomposition theory) ile birleştirilmiştir. Geliştirilmiş algoritma, geçerliliğinin ve uygulanabilirliğinin doğrulanması için ultrasonik sinyallerdeki kirlenmeyi tespit işleminde kullanılmıştır (Li ve ark., 2018).

Song ve arkadaşları, geleneksel düğüm dağıtım stratejisinin kusurunu çözmek için, değişen adımlı geliştirilmiş FOA'yı geliştirmiş ve kablosuz sensör ağı ile birleştirerek şebeke kapsama optimizasyonu gerçekleştirmiştir. Izgara kapsama modeli kullanılmıştır. Şebeke kapsama alanı ve artıklık, grid bölümlenmesi ile karşılık gelen matematiksel değişkenlere dönüştürülmüştür. Sensör düğümlerinin maksimum etkili

yarıçapı mobil düğüm kablosuz sensör ağına sabitlenmiştir. Düğümlerin yeri rastgele fırlatılmıştır. Sensör düğümlerinin yeri, sabit konum düğümlerine yerleştirilmiştir. Karşılık gelen ağ modeli ile birleştirildiğinde, geliştirilmiş algoritma kablosuz sensör ağına uygulanmıştır. Sonuçlar, geliştirilmiş algoritmanın, kablosuz sensör ağlarının kapsama optimizasyonundan etkili ve üstün olduğunu göstermiştir (Song ve ark., 2018).

Cheng ve arkadaşları, FOA'nın yerel optimal aralığın dışına çıkma olasılığını analiz ederek, FOA'nın, optimal çözümü sıfır olmayan karmaşık optimizasyon problemlerinin çözümünde etkisiz olduğundan bahsetmiş ve FOA'nın performansını artırmak için, bir modifiye edilmiş bir FOA'yı tanıtmıştır. Önerilen algoritmada, küresel arama yeteneğini geliştirmek için aday çözümünü üretmede tek biçimli bir mekanizma kullanılır, optimum doğruluğu artırmak için uçuş aralığını kontrol eden kendinden uyarlamalı bir yol ve meyve sineğinin arama davranışını taklit etmek amacıyla popülasyon için basamaklı büyüme mekanizması tanıtılır. Önerilen algoritma, 12 kıyaslama fonksiyonunda, temel FOA'dan, Küresel Parçacık Sürü Optimizasyonu Algoritmasından ve diğer bir geliştirilmiş FOA'dan daha etkili ve güçlü olduğunu göstermiştir (Cheng ve ark., 2018).

Wang ve arkadaşları, sürekli fonksiyon problemlerini optimize etmek için ortak arama stratejileriyle geliştirilmiş bir FOA önermiştir. İlk olarak, yakınsama oranını iyileştirmek için yeni bir parametre içeren ortak bir grup araştırması gerçekleştirilmiştir. İkinci olarak, çözümün doğruluğunu artırmak için biyolojik belleğe, yani belleğin hareket yönüne benzer yeni bir arama stratejisi önerilmiştir. Üçüncüsü, işbirlikçi grup aramasında, yerel bir optimum noktasına takılmaması için bir irtifa iniş araması kullanılır. Son olarak, bir sinir ağındaki aktivasyon fonksiyonuna benzeyen yeni bir işlev, üç arama stratejisini birleştirmek için önerilmiştir. Önerilen algoritmanın sağlamlığını ve yakınsamasını test etmek için 29 karmaşık sürekli kıyaslama fonksiyonu kullanılmıştır. Sonuçlar, önerilen algoritmanın çoğu fonksiyon için diğer sezgisel algoritmaları geride bıraktığını göstermiştir (Wang ve ark., 2019).

Liu ve arkadaşları, temel FOA'ya iki yeni stratejinin dâhil edildiği geliştirilmiş bir FOA önermiştir. Spesifik olarak, yerel sömürüyü arttırmak için ters güç üretim mekanizması ile birlikte bir koku duyarlılığı parametresi tanıtılmıştır. Ayrıca, arama alanındaki küresel araştırmayı geliştirmek için sorunlu alanda arama alanını uyarlamalı olarak ayarlayan dinamik bir daraltıcı arama yarıçapı stratejisi dâhil edilmiştir. FOA, önerilen algoritma, üç adet son geliştirilen FOA varyantı ve altı meta-sezgisel, iyi bilinen tek modlu ve çok modlu 26 sayısal fonksiyonda test edilmiştir. Deneysel sonuçlar ve

karşılaştırmalar, önerilen algoritmanın üç FOA varyantından daha iyi performans göstermiştir. Çözüm doğruluğu ve yakınsama oranı açısından da diğer altı meta-sezgisele karşı rekabetçi performans göstermiştir (Liu ve ark., 2019).

Chikh ve Chikhi birleşik kümelenmiş negatif seçim algoritması ve FOA'yı birlikte kullanan yeni bir istenmeyen elektronik posta (spam) algılama yaklaşımı önermiştir. Spam, dünyadaki e-posta iletişimini tehlikeye atan bir sorundur. Bu tehdidin miktarını durdurmak veya azaltmak için negatif seçim algoritmasına dayalı önerilen yöntemler sürekli uyarlanabilirlikten yoksundur ve düşük tespit performansından mustarıptır. Üstelik bu yöntemler kendi kendine ait olmayan tüm alanları kapsayacak şekilde çok sayıda algılayıcı gerektirir. Bu nedenle, bu çalışma, klasik negatif seçim algoritmasının verimliliğini artırmak için gerçek negatif seçim algoritmasını k-aracı kümeleme (k-means clustering) ve FOA ile birleştirmiştir. Spam karşılaştırma testinde yapılan deney sonuçları, önerilen yöntemin performansının, özellikle tespit doğruluğu, pozitif tahmin ve hesaplama karmaşıklığı açısından, klasik negatif seçim algoritması ve negatif seçim algoritmasının PSO'lu versiyonundan daha iyi olduğunu göstermiştir (Chikh ve Chikhi, 2019).

Chen ve Pi, temel FOA'nın yüksek boyutlu karmaşık sürekli fonksiyonların çözümündeki hızlı yakınsama probleminin üstesinden gelmek ve çözümün hassasiyetini arttırmak için, uzamsal genişlemeye ve dinamik mutasyona dayalı yeni bir FOA önermiştir. Önerilen yöntem, orijinal sabit adım boyutunu odaklanmış bir arama yöntemine dönüştürmek ve algoritmanın gelişimine dinamik mutasyon stratejisini yerleştirmek ile öne çıkar. Yeni bireyler üretmek için en iyi birey üzerinde gauss haritalama işlemini kullanmaktadır. Arama yeteneğini geliştirmek amacıyla arama alanını genişleten ters çözüm tasarlanmıştır. 18 tanınmış kıyaslama fonksiyonlarına ait deneysel sonuçlara göre, algoritma verimli ve etkilidir. Algoritma kısa vadeli rüzgâr tahminleri için uygulanmıştır. Simülasyon sonuçları, algoritmanın geleneksel algoritmalarından daha iyi tahmin etkisine sahip olduğunu göstermiştir (Chen ve Pi, 2019).

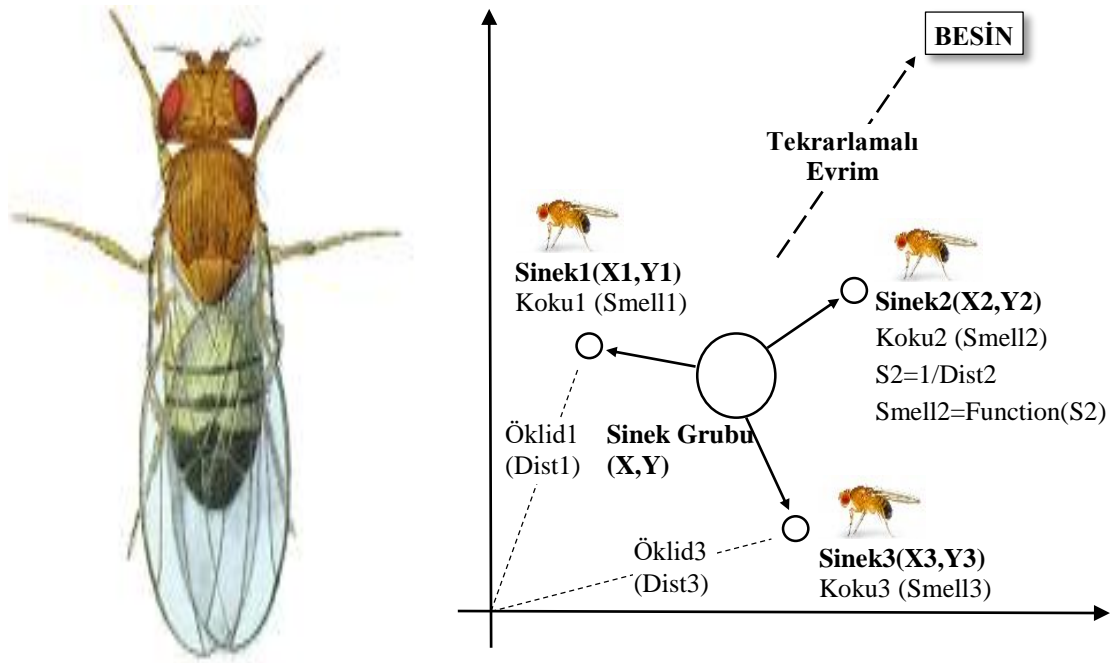
3. MATERYAL VE YÖNTEM

3.1. Meyve Sineği Optimizasyon Algoritması (FOA)

FOA, When-Tsao Pan tarafından 2011’de literatüre girmiştir (Wen-Tsao, 2011; Pan, 2012). Doğadaki meyve sineğinin besin bulma davranışına bağlı yeni bir meta-sezgisel algoritmadır. Meyve sineğinin yiyecek arama davranışını temel alır. Global optimizasyon problemlerini çözmek için kullanılır.

Meyve sineği, ılıman ve tropik iklim bölgelerinde yaşar. His ve algılama noktasında özellikle görüş ve koku algılama konularında diğer sinek türlerine göre üstündür. Meyve sineklerinin koku algılama organları havadaki her çeşit kokuyu 40 km uzaklıktan bile alabilmektedir. Besine yaklaşıncı da hassas görme algılarını kullanarak besine doğru uçarlar.

Meyve sineği, önce koku alma organı ile yiyeceği koklar ve o yöne doğru uçmaya başlar. Yiyeceğe yaklaştığında ise görme algısını kullanır ve yiyeceği bulur (Pan, 2012; Li ve ark., 2013; Niu ve ark., 2015). Meyve sineğinin vücut yapısı ve yiyecek arama davranışı Şekil 3.1’de gösterilmektedir.



Şekil 3.1.a. Meyve sineği vücut yapısı b. Meyve sineği grubunun yiyecek arama tekrarlaması

Meyve sineğinin besin bulma özelliğine dayanılarak oluşturulan algoritma için bir model geliştirilmiştir. Bu model;

- 1- Şekil 3.1.b'de gösterildiği üzere, meyve sineğinin rastgele ilk başlangıç konumu belirlenir.

$$Init X; Init Y \quad (3.1)$$

- 2- Besin arayışında koku algısı kullanılarak rastgele yön ve uzaklık verilir.

$$\begin{aligned} X_i &= X + Rastgele Değer [-1, 1] \\ Y_i &= Y + Rastgele Değer [-1, 1] \end{aligned} \quad (3.2)$$

- 3- Besinin yeri bilinemediğinden, önce mesafenin karşılığı anlamındaki orijine olan uzaklık tahmin edilir (Dist), sonra koku konsantrasyonu karar verme değeri hesaplanır (S).

$$Dist_i = \sqrt{X_i^2 + Y_i^2} \quad (3.3)$$

$$S_i = 1/Dist_i$$

- 4- Koku konsantrasyonu karar değeri (S), koku konsantrasyon karar fonksiyonunda (fitness fonksiyonu) yerine konur ve sineğin yerinin koku konsantrasyonu (smell) bulunur.

$$Smell_i = Function(S_i) \quad (3.4)$$

- 5- Azami koku konsantrasyonuna sahip sinek, sinek kümesinde bulunur.

$$[bestSmell \ bestIndex] = min/max(Smell) \quad (3.5)$$

- 6- En yüksek koku konsantrasyon değeri ve x,y koordinatları bulunur ve sinek kümesinin yeni lokasyonu olarak kullanılır.

$$Smellbest = bestSmell$$

$$X = X_{bestIndex}$$

$$Y = Y_{bestIndex} \quad (3.6)$$

- 7- 2-5 adımları tekrar edilir. Mevcut iterasyondaki koku konsantrasyonu bir önceki koku konsantrasyonundan daha iyiye 6. adım uygulanır.

kurallarından oluşmaktadır.

Bu algoritmanın kolay anlaşılır olması yanında program koduna dönüştürülmesi de kolaydır. Bununla beraber program kodu, diğer algoritmalara kıyasla çok uzun değildir ve bu nedenle optimizasyon problemlerine uygulama anlamında kolaylıklar içerir. FOA'nın sözde kodu Şekil 3.2'de verilmiştir.

Algoritma. FOA**Parametreler.** Popülasyon sayısı (N), maksimum iterasyon (maxgen), rand 0 ile 1 arasında rastgele bir değer

```

// Sürünün başlangıç pozisyonlarını oluştur
   $x_0 = 10 * rand()$ 
   $y_0 = 10 * rand()$ 

// Sürüyü oluştur ve en iyi çözümü bul, koku fazı
for  $i=1,2,\dots,N$ 
   $x_i = x_0 + (2 * rand - 1)$ 
   $y_i = y_0 + (2 * rand - 1)$ 
   $d_i = \sqrt{x_i^2 + y_i^2}$ 
   $S_i = \frac{1}{d_i}$ 
endfor
 $f_i = fit(\vec{S}_i)$ 

[bestS bi] = min/max( $f_i$ )
 $f_b = bestS$  &  $S_b = S_{(bi)}$  &  $x_0 = x_{(bi)}$  &  $y_0 = y_{(bi)}$ 

// Maksimum iterasyona kadar evrimi tekrarla, görme fazı
for  $ITR=1,2,\dots,maxITR$ 
  for  $i=1,2,\dots,N$ 
     $x_i = x_0 + (2 * rand - 1)$ 
     $y_i = y_0 + (2 * rand - 1)$ 
     $d_i = \sqrt{x_i^2 + y_i^2}$ 
     $S_i = \frac{1}{d_i}$ 
  endfor
   $f_i = fit(\vec{S}_i)$ 

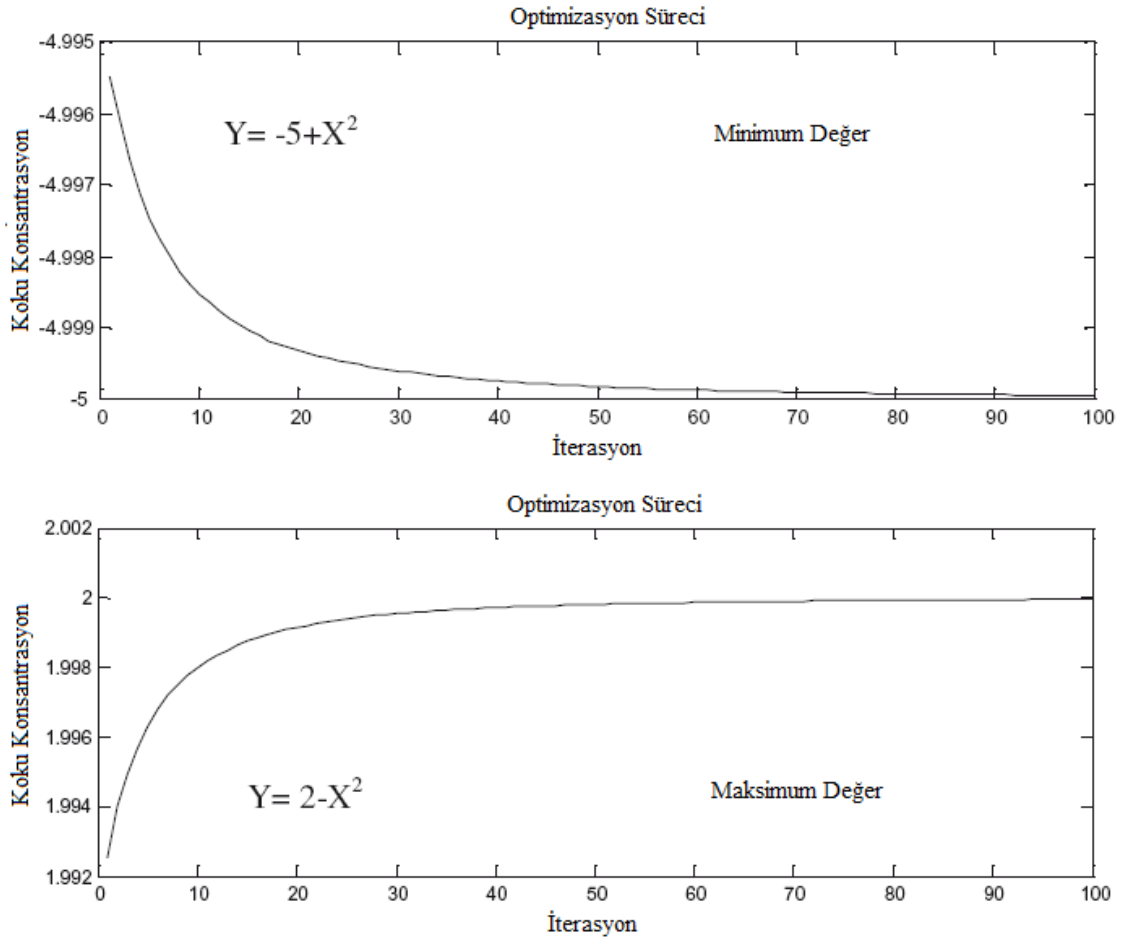
  [bestS bi] = min/max( $f_i$ )

  if  $bestS < f_b$ 
     $f_b = bestS$  &  $S_b = S_{(bi)}$  &  $x_0 = x_{(bi)}$  &  $y_0 = y_{(bi)}$ 
  endif
endfor

```

Şekil 3.2. FOA'nın sözde kodu.

FOA'nın minimum ve maksimum noktaları bulması iki örnek fonksiyon ile Şekil 3.3'de gösterilmiştir.



Şekil 3.3. Minimum ve maksimum çözüm eğrileri (Pan, 2012).

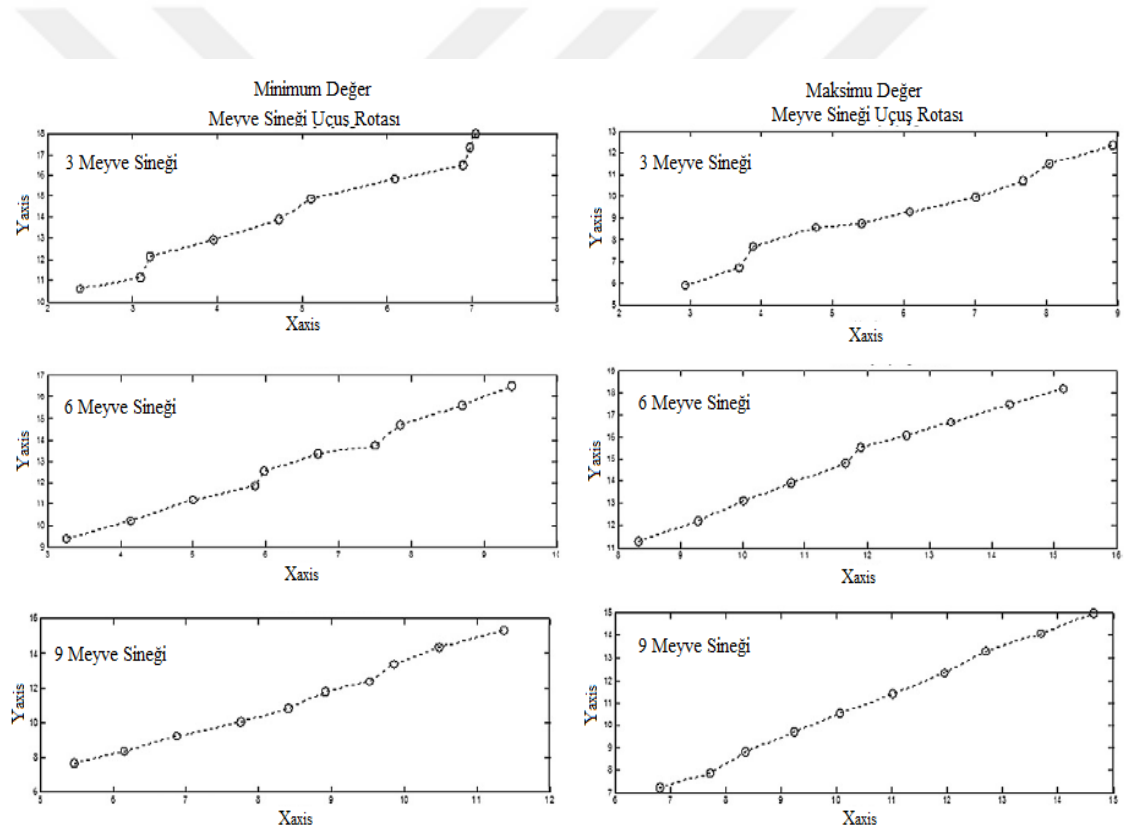
Şekil 3.3'deki yakınsama eğrilerinden de görüldüğü gibi fonksiyonların FOA ile minimum ve maksimum değerleri bulunmuştur.

$$\begin{aligned} Y &= -5 + X^2 \quad (\text{minimizasyon}) \\ Y &= 2 - X^2 \quad (\text{maksimizasyon}) \end{aligned} \quad (3.7)$$

Karar değişkeni X sıfır değerine yakınsanarak optimum sonuç elde edilir. Fonksiyonlarda görüleceği üzere minimum çözüm değeri ilk fonksiyon için -5 ve maksimum çözüm değeri ikinci fonksiyon için 2 olmalıdır. Fonksiyonların çözümünde meyve sineği sürüsünün rastgele başlama konumu için [0,10] aralığı, meyve sineğinin gıda arama tekrarı için konum güncellemesi rastgele [-1 1] aralığında seçilmiştir. 100 kez tekrarlanan minimum ve maksimum değer bulma işleminden sonra programın uygulanması sonucu yavaş yavaş fonksiyonel uç değer çözümüne yaklaşım olmaktadır.

Şekil 3.3 bu süreci anlatmaktadır. Şekil 3.3’de görüleceği gibi minimum değer -5’e ve maksimum değer 2’ye ulaşmaktadır.

Şekil 3.4’de meyve sineği sayısının, çözümü nasıl etkilediği görülmektedir. 10 arama tekrarından sonra, minimum veya maksimum değer için fark etmemekte, az sayıda meyve sineği kümesinin kararsız bir arama yolunun ve daha yavaş bir yakınsama hızının dezavantajına sahip olacağı gözükse de işlem hızının yüksek olması avantaj olarak ortaya çıkmaktadır. Çok sayıda meyve sineği kümesinin ise istikrarlı bir arama yolu ve daha hızlı bir yakınsama avantajlarına sahip olduğu görülmekle birlikte program yürütme hızının yavaşlaması dezavantajına sahiptir (Pan, 2012). Bu nedenle optimizasyon probleminin karmaşıklığı göz önüne alınarak meyve sineği sayısının ve rastsallık parametrelerinin uygun seçilmesi gerekmektedir.



Şekil 3.4. Farklı sayıda meyve sineği için arama yolu grafiği (Pan, 2012).

Algoritmanın avantajları aşağıda maddeler halinde verilmiştir.

- Basit yapılıdır.
- Çözümü yakınsamada hızlıdır.
- Uygulamalara uyarlanabilmesi kolaydır.

- Program kodu kısa ve anlaşılması kolaydır.
- Kararlıdır.
- Az parametreye sahiptir.

Algoritmanın bahsedilen avantajlarına rağmen eksik yönleri de mevcuttur.

Bunlar;

- Sürünün ilk konumunun çeşitliliği zayıftır (Liu ve ark., 2012; Wang ve ark., 2013).
- Konum güncellemesi iterasyon boyunca her zaman $[-1, 1]$ aralığında değer aldığı için sabit ve küçüktür (Shan ve ark., 2013; Ramachandran ve Bellarmine, 2014).
- Lokal optimuma takılabilmektedir (Abidin ve ark., 2011; Li ve ark., 2012a; Shan ve ark., 2013; Pan ve ark., 2014; Pan, 2014; Ramachandran ve Bellarmine, 2014).
- Uygunluk (Fitness) fonksiyonu her zaman pozitifdir (Pan, 2013a; Xu ve Tao, 2013; Dai ve ark., 2014; Pan, 2014).
- Yapısı, sürekli optimizasyon problemleri için uygundur (Wang ve ark., 2013; Pan ve ark., 2014).

FOA önerildiği günden bu yana birçok araştırmacının ilgisini çekmiş ve birçok optimizasyon probleminin çözümü için başarılı şekilde kullanılmıştır. Bu problemlerden bazıları aşağıda listelenmiştir.

- Finansal stres (Pan, 2012; Tu ve ark., 2012; Yang ve ark., 2012b; Chen ve ark., 2013; Dong-qin ve Bu-hai, 2013).
- Güç yükü tahmini (Li ve ark., 2012b; Li ve ark., 2013; Niu ve ark., 2014).
- İş akış zamanlaması (Zhang ve Wang, 2014).
- PID kontrol parametre ayarı (Han ve ark., 2012; Li ve ark., 2012a; Liu ve ark., 2012; Zhang ve ark., 2012).
- Yarı iletken final test zamanlaması (Zheng ve ark., 2014).
- Web müzayede lojistik servisi (Lin, 2013).
- Yüzey araç uygulaması (Abidin ve ark., 2011).
- GRNN optimizasyonu (Yang ve ark., 2012a; Pan, 2013b; Pan, 2014).
- Mağaza planlama problemi (Liu ve ark., 2014; Rana ve Sharma, 2014).
- Fazör ölçüm ünitelerinin (PMU) optimum yerleşimi (Ramachandran ve Bellarmine, 2014).
- Çok boyutlu sırt çantası problemi (Wang ve ark., 2013).
- Gemi manevra tahmini (Wang ve Zou, 2012; Wang ve Zou, 2013).

Bu yöntemin optimizasyon problemlerinin çözümünde daha iyi sonuçlar üretebilmesi için yukarıda bahsedilen dezavantajlarının giderilmesi gerekmektedir.

Bu tez çalışmasında bu dezavantajları giderecek iyileştirmeler yapılmıştır. Bu iyileştirmeler için arama ve sezgisel algoritmalarından faydalanılmış ve hibrit yöntemler geliştirilmiştir.

3.2. Benzetilmiş Tavlama Algoritması

Benzetilmiş Tavlama algoritması genel optimizasyon problemlerini çözmek için kullanılan bir algoritmadır. Kirkpatrick ve arkadaşları tarafından 1983 yılında önerilmiştir. Yapısı, metallerin tavlama işlemine benzer. Tavlama benzetiminde kötü çözümler bazen kabul edilir. Sıcaklık değeri kötü bir çözümün kabul edilme olasılığını belirler. Kötü çözümlerin kabul edilmesi belli olasılıkla belirlenir. Sıcaklığın yüksek olması kötü çözümlerin kabul oranını artırır. Sıcaklık düştükçe kabul oranı azalır. Sıcaklık ilk başta yüksektir, sonra kademeli şekilde azaltılır (Kirkpatrick ve ark., 1983).

Algoritmanın karar mekanizması Denklem 3.8’de verilmiştir.

$$\begin{aligned} d &= \text{Mevcut çözüm} - \text{Önceki çözüm} \\ pa &= \exp\left(-\frac{d}{k*T}\right) \\ r &= \text{Rastgele değer} \end{aligned} \quad (3.8)$$

Burada d, mevcut çözümün önceki çözümden farkı, k, soğutma katsayısı, T sıcaklık değeri, r, [0, 1] aralığında rastgele bir değer, pa ise mevcut çözümün kabul olma olasılığıdır.

Mevcut çözüm önceki çözümden daha iyi değilse, Denklem 3.8’deki eşitlikler sırasıyla gerçekleştirilir. Eğer r, pa’dan küçük ise ve mevcut çözüm kabul edilir. Aksi durumda önceki çözüm korunur.

3.3. Jaya Algoritması

Jaya algoritması 2016 yılında Rao tarafından önerilmiştir. (Rao, 2016; Rao ve ark., 2016). Algoritma her zaman en iyi çözüme ulaşmaya ve en kötü çözümden uzaklaşmaya çalışır. Algoritma, bu amaçla, yeni çözümü elde etmek için Denklem 3.9’u kullanmaktadır.

$$X_{k,j}' = X_{k,j} + r_{1,j} * (Best_j - |X_{k,j}|) - r_{2,j} * (Worst_j - |X_{k,j}|) \quad (3.9)$$

Bu denklemde k , işlenen aday çözüm indeksidir, Best ve Worst sırasıyla şu ana kadar elde edilen en iyi ve en kötü çözümdür. j , boyut indeksidir, r_1 ve r_2 , $[0, 1]$ aralığında üretilen rasgele değerleri temsil eder.

“ $r_{1,j}*(Best - |X_{k,j}|)$ ” terimi, çözümün en iyi çözüme yaklaşma eğilimini gösterir.

“ $-r_{2,j}*(Worst - |X_{k,j}|)$ ” terimi, çözümün en kötü çözümü engelleme eğilimini gösterir.

Aday çözüm üretildikten sonra, adayın uygunluk değerleri ve mevcut çözümler hesaplanır. Mevcut ve aday çözümler arasında açgözlü bir yaklaşım uygulanmaktadır. Aday çözümün uygunluğu mevcut çözümün uygunluğundan daha iyiye, aday çözüm popülasyondaki mevcut çözüm ile değiştirilir. Jaya algoritması yinelemeli bir optimizasyon algoritmasıdır ve yukarıda verilen işlem bir sonlandırma koşulu sağlanana kadar tekrar edilir.

3.4. İşaret Parametreleri Kullanılarak Geliştirilen Meyve Sineği Optimizasyon Algoritması (SFOA) (Önerilen Algoritma 1)

FOA'nın, negatif karar değişkenleri kullanarak elde edilen optimum çözümleri olan problemleri çözemediği literatürde ve Denklem 3.10'da görülmektedir. (Dai ve ark., 2014; Niu ve ark., 2015)

$$\begin{aligned} Dist_i &= \sqrt{X_i^2 + Y_i^2} \\ S_i &= 1/Dist_i \\ Smell_i &= Function(S_i) \end{aligned} \quad (3.10)$$

Bir başka ifadeyle FOA'da koku konsantrasyon değerleri olarak belirtilen aday çözümler her zaman çözüm uzayının pozitif kısmında aranır. Bu durum, uygunluk değerini hesaplamak için kullanılan karar değişkenleri olan koku konsantrasyon değerlerinin hesaplanmasından ve aynı zamanda koku konsantrasyon değerlerini elde etmek amacıyla kullanılan öklid uzaklık fonksiyonundan kaynaklanmaktadır. Öklid uzaklık fonksiyonunun denkleminde de görüleceği gibi elde edilen koku konsantrasyon değerleri her zaman pozitif değerlerdir. Bu nedenle FOA'nın orijinal versiyonu karar değişkenleri için negatif değerler üretememektedir. Bu durum FOA'nın literatürdeki diğer algoritmalarla adil olarak karşılaştırılmasını engellemektedir. Bu sorunun üstesinden gelmek için, bu çalışmada FOA'nın geliştirilmiş bir versiyonu sunulmuştur.(Babalık ve ark., 2018)

Bu çalışmada r ve q olarak ifade edilen iki yeni parametre her bir sineğin koku değerine işaret vermek amacıyla FOA'ya eklenmiştir. Bir başka deyişle, sineklerin dört boyutlu arama uzayında uçtuğu varsayılmıştır.

Bu varsayımda, algoritmanın meta-sezgisel yaklaşımla güncellenen 4 tane parametreye sahip olduğu ifade edilmeye ve sineklerin uzayda 4 yönde hareket ettikleri kabullenmeye çalışılmıştır.

İlk iki yön sineğin konumlarının tanımı amacıyla, diğer iki yön ise konuma pozitif ya da negatif yönde olduğunu gösteren bir işaret vermek amacıyla kullanılmıştır. Bu kabullenmeyle işaret değişkenlerinden ilki diğerinden büyükse koku değeri pozitif olarak aksi halde negatif olarak kabul edilmiştir.

Geliştirilen FOA'nın adımları aşağıda verilmiştir.

- 1- Başlatma aşaması: Popülasyon sayısı N, başlangıç sürü pozisyonları \vec{x}_0 ve \vec{y}_0 , başlangıç işaret değişkenleri \vec{r}_0 ve \vec{q}_0 olan SFOA'nın ana parametreleri oluşturulur. Meyve sineğinin ilk başlangıç konumu ve işaret değişkenleri Denklem 3.11'deki gibi belirlenir. Rand [0, 1] arasında rastgele bir değerdir.

$$\begin{aligned} x_{0,j} &= \max X * (2 * rand - 1) \\ y_{0,j} &= \max Y * (2 * rand - 1) \\ r_{0,j} &= rand \\ q_{0,j} &= rand \end{aligned} \quad (3.11)$$

İşaretleme \vec{r}_0 ve \vec{q}_0 kullanılarak gerçekleştirilir. \vec{x}_0 ve \vec{y}_0 sırasıyla [-maxX, maxX] ve [-maxY, maxY] aralığında başlatılır. Ayrıca, probleme özgü sabitler maxX ve maxY, birbirine eşit olarak Denklem 3.12 deki gibi hesaplanır;

$$\max X = \max Y = ub \quad (3.12)$$

Burada ub, kıyaslama fonksiyon değişkenlerinin üst sınırlarını temsil etmektedir. Koku konsantrasyon yargı değerleri başlangıç noktasına olan Öklid uzaklığına bölünerek elde edildiği için maxX ve maxY değerleri, denklemde olduğu gibi birbirine eşit olarak alınır. Deklem 3.12 ile kıyaslama fonksiyonlarının üst sınırları kullanılmıştır. Bu sabitler eşit ve adil bir çalışma ortamı oluşturmak için hem FOA hem de SFOA'da eşit şekilde kullanılır.

- 2- Popülasyon başlatma aşaması: Sineklerin pozisyonları, bu aşamada Denklem 3.13 kullanarak ayarlanır. Besin arayışında koku algısı kullanılarak rastgele yön ve uzaklık verilir.

$$\begin{aligned}
x_{i,j} &= x_{0,j} + (2 * rand - 1) \\
y_{i,j} &= y_{0,j} + (2 * rand - 1) \\
r_{i,j} &= r_{0,j} + (2 * rand - 1) \\
q_{i,j} &= q_{0,j} + (2 * rand - 1)
\end{aligned} \tag{3.13}$$

Burada $r_{i,j}$ ve $q_{i,j}$, i'ninci sineğin j'ninci boyuttaki işaret değişkenlerini belirtir.

- 3- Sineklerin değerlendirilmesi: Her sineğin uzaklığı, işareti ve koku konsantrasyon değerleri Denklem 3.14, 3.15 ve 3.16 kullanılarak hesaplanır. Ayrıca, her sineğin uygunluk değeri denklem 3.17 ile hesaplanır. Besinin yeri bilinmediğinden, orijine uzaklık önce tahmin edilir (Dist), sonra koku konsantrasyonu karar verme değeri hesaplanır (S).

$$d_{i,j} = \sqrt{x_{i,j}^2 + y_{i,j}^2} \tag{3.14}$$

$$Sign_{i,j} = \begin{cases} 1 \text{ (pozitif)} & r_{i,j} > q_{i,j} \\ -1 \text{ (negatif)} & \text{aksi halde} \end{cases} \tag{3.15}$$

$$S_{i,j} = Sign_{i,j} \frac{1}{d_{i,j}} \tag{3.16}$$

$$f_i = fit(\vec{S}_i) \tag{3.17}$$

Burada \vec{Sign}_i i sineğinin işaret vektörünü temsil eder.

- 4- Yargılama ve güncelleme aşaması: En iyi sinek sürü arasından seçilir ve sürü içinde daha iyi bir koku değeri elde edildiğinde, başlangıç sürü pozisyonları ve işaret değerleri, en iyi sineklerin pozisyonlarına ve işaret değerlerine Denklem 3.18 ve 3.19 ile eşit olarak güncellenir. Koku konsantrasyonu karar değeri (S), koku konsantrasyon karar fonksiyonunda (fitness fonksiyonu) yerine konur ve sineğin yerinin koku konsantrasyonu (smell) bulunur.

$$[bestS \ bi] = min/max(\vec{f}_i) \tag{3.18}$$

$$\begin{aligned}
f_b &= bestS \\
\vec{x}_b &= \vec{x}_{(bi)} \quad , \quad \vec{x}_0 = \vec{x}_b \\
\vec{y}_b &= \vec{y}_{(bi)} \quad , \quad \vec{y}_0 = \vec{y}_b \\
\vec{r}_b &= \vec{r}_{(bi)} \quad , \quad \vec{r}_0 = \vec{r}_b \\
\vec{q}_b &= \vec{q}_{(bi)} \quad , \quad \vec{q}_0 = \vec{q}_b
\end{aligned} \tag{3.19}$$

Burada \vec{r}_b ve \vec{q}_b , mevcut sürü içerisinde yineleme işlemi sırasında şimdiye kadar elde edilen optimal uygunluk değerine sahip sineğin işaret değişkenlerini belirtir.

- 5- Durdurma kriteri kontrol aşaması: İterasyon süreci boyunca durdurma kriterine yani maksimum iterasyona ulaşıp ulaşılmadığı kontrol edilir. Kriter

sağlandığında yineleme işlemi durdurulur ve $\vec{f}_b, \vec{x}_b, \vec{y}_b, \vec{r}_b$ ve \vec{q}_b elde edilir, aksi takdirde adım 2'ye geri dönlür.

Önerilen yaklaşımda işaret vektörünü hesaplamak amacıyla kullanılan r ve q vektörleri x ve y vektörlerinin hesaplanması amacıyla kullanılan aynı yöntemle güncellenmiştir. Burada r ve q vektörleri iterasyon boyunca elde edilen en optimal uygunluk değerine sahip sinek ele alınarak güncellenmiştir. Bu önermeyle geliştirilen algorithmada meta sezgisel yaklaşım korunmuştur. SFOA'nın sözde kodu Şekil 3.5'de verilmiştir.

SFOA'nın işaret değerleri sineklerin negatif koku konsantrasyon değerlerini almasına izin verir ve algoritma diğer algoritmalarla daha eşit ve daha adil rekabet edebilir.

Algoritma. SFOA

Parametreler. Popülasyon sayısı (N), problemin boyutu (D), maksimum iterasyon (maxgen) ve problemin sınırları (maxX and maxY)

// Sürünün başlangıç pozisyonlarını oluştur

for $j=1,2,\dots,D$

$$x_{0,j} = \text{maxX} * (2 * \text{rand} - 1)$$

$$y_{0,j} = \text{maxY} * (2 * \text{rand} - 1)$$

$$r_{0,j} = \text{rand}$$

$$q_{0,j} = \text{rand}$$

endfor

// Sürüyü oluştur ve en iyi çözümü bul, koku fazı

for $i=1,2,\dots,N$

for $j=1,2,\dots,D$

$$x_{i,j} = x_{0,j} + (2 * \text{rand} - 1)$$

$$y_{i,j} = y_{0,j} + (2 * \text{rand} - 1)$$

$$r_{i,j} = r_{0,j} + (2 * \text{rand} - 1)$$

$$q_{i,j} = q_{0,j} + (2 * \text{rand} - 1)$$

$$d_{i,j} = \sqrt{x_{i,j}^2 + y_{i,j}^2}$$

$$\text{Sign}_{i,j} = \begin{cases} 1 (\text{pozitif}) & r_{i,j} > q_{i,j} \\ -1 (\text{negatif}) & \text{otherwise} \end{cases}$$

$$S_{i,j} = \text{Sign}_{i,j} \frac{1}{d_{i,j}}$$

endfor

$$f_i = \text{fit}(\vec{S}_i)$$

endfor

[bestS bi] = min/max(f_i)

$$f_b = \text{bestS} \quad \& \quad \vec{S}_b = \vec{S}_{(bi)} \quad \& \quad \vec{x}_b = \vec{x}_{(bi)} \quad \& \quad \vec{y}_b = \vec{y}_{(bi)} \quad \& \quad \vec{r}_b = \vec{r}_{(bi)} \quad \& \quad \vec{q}_b = \vec{q}_{(bi)}$$

$$\vec{x}_0 = \vec{x}_b \quad \& \quad \vec{y}_0 = \vec{y}_b \quad \& \quad \vec{r}_0 = \vec{r}_b \quad \& \quad \vec{q}_0 = \vec{q}_b$$

// Maksimum iterasyona kadar evrimi tekrarla, görme fazı

for $ITR=1,2,\dots,\text{maxITR}$

for $i=1,2,\dots,N$

for $j=1,2,\dots,D$

$$x_{i,j} = x_{0,j} + (2 * \text{rand} - 1)$$

$$y_{i,j} = y_{0,j} + (2 * \text{rand} - 1)$$

$$r_{i,j} = r_{0,j} + (2 * \text{rand} - 1)$$

$$q_{i,j} = q_{0,j} + (2 * \text{rand} - 1)$$

$$d_{i,j} = \sqrt{x_{i,j}^2 + y_{i,j}^2}$$

$$\text{Sign}_{i,j} = \begin{cases} 1 (\text{pozitif}) & r_{i,j} > q_{i,j} \\ -1 (\text{negatif}) & \text{otherwise} \end{cases}$$

$$S_{i,j} = \text{Sign}_{i,j} \frac{1}{d_{i,j}}$$

endfor

$$f_i = \text{fit}(\vec{S}_i)$$

endfor

[bestS bi] = min/max(f_i)

if $\text{bestS} < f_b$

$$f_b = \text{bestS} \quad \& \quad \vec{S}_b = \vec{S}_{(bi)} \quad \& \quad \vec{x}_b = \vec{x}_{(bi)} \quad \& \quad \vec{y}_b = \vec{y}_{(bi)} \quad \& \quad \vec{r}_b = \vec{r}_{(bi)} \quad \& \quad \vec{q}_b = \vec{q}_{(bi)}$$

$$\vec{x}_0 = \vec{x}_b \quad \& \quad \vec{y}_0 = \vec{y}_b \quad \& \quad \vec{r}_0 = \vec{r}_b \quad \& \quad \vec{q}_0 = \vec{q}_b$$

endif

endfor

Şekil 3.5. SFOA'nın sözde kodu.

3.5. Benzetilmiş Tavlama Kullanılarak Geliştirilen Meyve Sineği Optimizasyon Algoritması (saFOA) (Önerilen Algoritma 2)

FOA her bir yinelemede sürü içerisindeki en iyi sineği hedef alır. En iyi sinek konumu başlangıç konumu olarak belirlenir ve yineleme bu şekilde devam eder. Besin kaynağına bu yöntemle ulaşmaya çalışır. Bu yöntem FOA'nın yerel minimum noktasına sık takılmasına neden olur. Bu çalışmada, bahsedilen dezavantajı ortadan kaldırabilmek için Benzetilmiş Tavlama algoritmasından faydalanılmıştır. Benzetilmiş Tavlama algoritmasının çalışma yapısı temel FOA'nın karar verme stratejisine eklenmiştir. Bu değişiklik ile temel FOA'nın karar verme stratejisi iki aşamalı hale getirilmiştir. İlk aşamada temel FOA'nın orijinal yapısı aynen alınmış, ikinci aşamada ise Benzetilmiş Tavlama algoritmasının çalışma yapısı kullanılmıştır. Bu iyileştirme ile sadece en iyi çözümler dikkate alınmak yerine belli oranda kötü çözümler de çözüm olarak kabul edilmiştir. Bu iyileştirme ile önerilen yöntemin, temel FOA'ya göre yerel minimuma takılma ihtimalinin azalması ve daha iyi sonuçlar elde ederek performansının iyileştirilmesi hedeflenmiştir (Gunduz ve ark., 2018).

Bu çalışmada, FOA'da bahsedilen 7. adımdaki karar mekanizması iki aşamalı şekilde değiştirilmiştir. saFOA'nın sözde kodu Şekil 3.6'da verilmiştir.

1. Aşama: Temel FOA'da olduğu şekli ile aynen alınmıştır. Yeni bulunan koku konsantrasyon değeri önceki koku konsantrasyon değerinden daha iyi ise, Denklem 3.6'daki eşitlik gerçekleştirilir.

2. Aşama: Diğer durumlarda Benzetilmiş Tavlama algoritmasının karar verme stratejisi uygulanmıştır. Denklem 3.20 sırasıyla uygulanır.

$$\begin{aligned} d &= bestSmell - Smellbest \\ pa &= \exp\left(-\frac{d}{k*T}\right) \\ r &= rastgele\ de\ ger \end{aligned} \quad (3.20)$$

Eğer r , pa 'dan küçük ise Denklem 3.6'daki eşitlik gerçekleştirilir ve mevcut iterasyonda elde edilen en iyi çözüm global en iyi olarak seçilir. Böylece global en iyiden daha kötü olan çözümlerin seçilme olasılığı sağlanmıştır. Eğer r , pa 'dan büyük ise bu ana kadar elde edilen global en iyi korunur.

Burada, pa kabul olasılığı, T sıcaklık, k soğutma katsayısı, r ise $[0, 1]$ arasında rastgele bir sayıdır.

Algoritma. saFOA

Parametreler. Popülasyon sayısı (N), problemin boyutu (D), maksimum iterasyon (maxgen) ve problemin sınırları (maxX and maxY)

// Sürünün başlangıç pozisyonlarını oluştur

for $j=1,2,\dots,D$

$$x_{0j} = \text{maxX} * (2 * \text{rand} - 1)$$

$$y_{0j} = \text{maxY} * (2 * \text{rand} - 1)$$

endfor

// Sürüyü oluştur ve en iyi çözümü bul, koku fazı

for $i=1,2,\dots,N$

for $j=1,2,\dots,D$

$$x_{i,j} = x_{0,j} + (2 * \text{rand} - 1)$$

$$y_{i,j} = y_{0,j} + (2 * \text{rand} - 1)$$

$$d_{i,j} = \sqrt{x_{i,j}^2 + y_{i,j}^2}$$

$$S_{i,j} = \frac{1}{d_{i,j}}$$

endfor

$$f_i = \text{fit}(\vec{S}_i)$$

endfor

$$[\text{bestS bi}] = \min/\max(f_i)$$

$$f_b = \text{bestS} \quad \& \quad \vec{S}_b = \vec{S}_{(bi)} \quad \& \quad \vec{x}_b = \vec{x}_{(bi)} \quad \& \quad \vec{y}_b = \vec{y}_{(bi)} \quad \& \quad \vec{x}_0 = \vec{x}_b \quad \& \quad \vec{y}_0 = \vec{y}_b$$

// Maksimum iterasyona kadar evrimi tekrarla, görme fazı

for $ITR=1,2,\dots,\text{maxITR}$

for $i=1,2,\dots,N$

for $j=1,2,\dots,D$

$$x_{i,j} = x_{0,j} + (2 * \text{rand} - 1)$$

$$y_{i,j} = y_{0,j} + (2 * \text{rand} - 1)$$

$$d_{i,j} = \sqrt{x_{i,j}^2 + y_{i,j}^2}$$

$$S_{i,j} = \frac{1}{d_{i,j}}$$

endfor

$$f_i = \text{fit}(\vec{S}_i)$$

endfor

$$[\text{bestS bi}] = \min/\max(f_i)$$

if $\text{bestS} < f_b$

$$f_b = \text{bestS} \quad \& \quad \vec{S}_b = \vec{S}_{(bi)} \quad \& \quad \vec{x}_b = \vec{x}_{(bi)} \quad \& \quad \vec{y}_b = \vec{y}_{(bi)} \quad \& \quad \vec{x}_0 = \vec{x}_b \quad \& \quad \vec{y}_0 = \vec{y}_b$$

else

$$d = \text{bestS} - f_b$$

$$pa = \exp\left(-\frac{d}{k * T}\right)$$

$r = \text{rastgele deęer}$

if $r < pa$

$$f_b = \text{bestS} \quad \& \quad \vec{S}_b = \vec{S}_{(bi)} \quad \& \quad \vec{x}_b = \vec{x}_{(bi)} \quad \& \quad \vec{y}_b = \vec{y}_{(bi)} \quad \& \quad \vec{x}_0 = \vec{x}_b \quad \& \quad \vec{y}_0 = \vec{y}_b$$

endif

endif

endfor

Şekil 3.6. saFOA'nın sözde kodu.

3.5. Yeni Bir Aday Çözüm Stratejisi Kullanılarak Geliştirilen Meyve Sineği Optimizasyon Algoritması (pFOA) (Önerilen Algoritma 3)

Temel FOA analiz edildiğinde, algoritmadaki tüm yapay ajanların en iyi çözümü izlediği görülmektedir. Bu nedenle, algoritma hızlı bir yakınsama özelliği gösterir, ancak bu, popülasyonda araştırma ve sömürü başarımını düşürmektedir. Popülasyondaki en iyi çözüm, arama alanı üzerinde yerel bir minimum nokta olduğunda, hızlı bir yakınsama çözüm uzayının iyi bir şekilde taranmasını engelleyebilmekte ve lokal optimumlara takılarak daha kaliteli çözümlerin elde edilmesine ket vurabilmektedir. Bu durum FOA'nın arama sırasında optimum çözüm elde etmesini engeller. Temel algoritmadaki diğer bir problem, çözüm uzayının sadece pozitif tarafında arama yapılmasıdır, çünkü mesafe (distance) daima pozitif bir değerle sonuçlanır.

Temel algoritma, yukarıda bahsedilen problemleri çözmek ve performansını geliştirmek için değiştirilmiştir. Bu çalışmadaki koku fazı, temel FOA ile aynı olmasına rağmen, görsel fazı yeni bir güncelleme kuralı dikkate alınarak değiştirilmiştir. Görsel fazda, temel FOA ve JAYA algoritmasının güncelleme stratejisi birlikte kullanılmıştır. Aynı zamanda, yeni aday çözümler elde etmek için iki farklı formül (sürüm 1 ve sürüm 2) kullanılmıştır. Önerilen yaklaşım, pFOA_v1 ve pFOA_v2 olarak adlandırılmıştır. (Iscan ve ark., 2019)

pFOA_v1'de, Jaya algoritmasında önerilen ve kullanılan Denklem 3.21 tarafından yeni çözümler elde edilmiştir.

$$S_{i,j} = S_{axis,j} + rand_1 * (Best_j - |S_{axis,j}|) - rand_2 * (Worst_j - |S_{axis,j}|) \quad (3.21)$$

pFOA_v2'de, Denklem 3.21'in değiştirilmiş bir versiyonu kullanılmıştır. Yeni aday çözümler Denklem 3.22 kullanılarak elde edilmiştir.

$$S_{i,j} = S_{axis,j} + rand_1 * Best_j - rand_2 * Worst_j \quad (3.22)$$

Bu formül, genel arama esnasında yinelemenin ilk zamanlarında arama adımının büyük olmasını, yineleme sayısı arttıkça da arama adımının küçülmesini sağlamaktadır. Bu durum çözümün kesinliğini ve yerel optimumdan kaçınmasını sağlar. Bu sayede pFOA_v2'nin yoğunlaşma kabiliyeti güçlenmiş ve yakınsaması hızlanmıştır.

Algoritmanın görsel fazındaki değişiklikler aşağıdaki gibidir;

Koku konsantrasyon yargı değerleri (S_i) Denklem 3.23 ve 3.24'deki gibi güncellenmiştir:

pFOA_v1 için;

$$S_{new_{i,j}} = S_{i,j} + rand_1 * (Best_j - |S_{i,j}|) - rand_2 * (Worst_j - |S_{i,j}|) \quad (3.23)$$

pFOA_v2 için;

$$S_{new_{i,j}} = S_{i,j} + rand_1 * Best_j - rand_2 * Worst_j \quad (3.24)$$

Optimizasyon problemine özgü amaç fonksiyon değeri (f_{new_i}) yeni koku konsantrasyon yargı değeri (S_{new_i}) Denklem 3.25 ile elde edilmiştir:

$$f_{new_i} = fit(\overrightarrow{S_{new_i}}) \quad (3.25)$$

Yeni ve mevcut aday çözümlerine, açgözlü bir seçim uygulanmıştır. Yeni koku değeri, mevcut koku değerinden daha iyiyse, her meyve sineğinin yeni koku ve koku konsantrasyon yargı değerleri (S_i) hafızaya alınmıştır. Bu prosedür Denklem 3.26 ile verilmiştir:

$$\begin{aligned} f_{new_i} < f_i \\ f_i = f_{new_i} \quad \& \quad \overrightarrow{S}_i = \overrightarrow{S_{new_i}} \end{aligned} \quad (3.26)$$

Maksimum yinelemenin her yüzde yirmisinde, FOA'nın temel yaklaşımını korumak amacıyla, en iyi koku değerine sahip meyve sineği Denklem 3.27 kullanılarak sabitlenmiştir. En iyi koku konsantrasyonu yargı değerinin, başlangıç değeri olduğu varsayılmış ve meyve sinekleri, sürümüne bağlı olarak Denklem 3.21 veya Denklem 3.22 kullanılarak yeniden üretilmiştir.

$$\overrightarrow{S}_{axis} = \overrightarrow{S}_{best} \quad (3.27)$$

Bu değişiklik temel FOA'ya iki büyük katkı sağlamıştır. Birincisi, en iyi ve en kötü çözümleri değerlendirmek ve gerçek çözümü elde etmektir. Bu, temel algoritmanın genel arama yeteneğini geliştirmiş ve çözüm alanı, FOA ile etkin bir şekilde aranmıştır. İkincisi, uygunluk fonksiyonunun negatif değerler alabilmesidir. Böylece, geliştirilen algoritma negatif çözümleri içeren problemleri de çözebilir. Bu durum, geliştirilmiş pFOA'ların temel algoritmaya göre farklı problemleri çözmek için kullanılmasına izin vermiştir.

Önerilen pFOA versiyonlarının sözde kodu Şekil 3.7'de akış şeması Şekil 3.8'de verilmiştir.

Şekil 3.7'den görüldüğü gibi, modifikasyon, görsel fazda verilmiştir ve algoritmanın geri kalanı temel FOA ile aynıdır. Bu nedenle, optimizasyon problemlerini çözmek için temel versiyon olarak kolayca uygulanabilir.

Algoritma. pFOA

Parametreler. Popülasyon sayısı (N), problemin boyutu (D), maksimum iterasyon (maxgen) ve problemin sınırları (maxX and maxY)

```

// Sürünün başlangıç pozisyonlarını oluştur
for j=1,2,..., D
    x0j = maxX * (2 * rand - 1)
    y0j = maxY * (2 * rand - 1)
endfor

// Sürüyü oluştur ve en iyi çözümü bul, koku fazı
for i=1,2,..., N
    for j=1,2,..., D
        xi,j = x0j + (2 * rand - 1)
        yi,j = y0j + (2 * rand - 1)
        di,j =  $\sqrt{x_{i,j}^2 + y_{i,j}^2}$ 
        Si,j =  $\frac{1}{d_{i,j}}$ 
    endfor
    fi = fitness( $\vec{S}_i$ )
endfor

// Maksimum iterasyona kadar evrimi tekrarla, görme fazı
for gen=1,2,..., maxgen

    [bestS bi] = min/max(fi)
    Best =  $\vec{S}_{(bi)}$ 
    [worstS wi] = max/min(fi)
    Worst =  $\vec{S}_{(wi)}$ 

    if gen, maxgen'in her %20'sinde ise
         $\vec{S}_{axis} = \vec{Best}$ 
        for i=1,2,..., N
            Si,j = Saxis,j + rand1 * (Bestj - |Saxis,j|) - rand2 * (Worstj - |Saxis,j|) // versiyon 1
            Si,j = Saxis,j + rand1 * Bestj - rand2 * Worstj // versiyon 2
        endfor
    endif

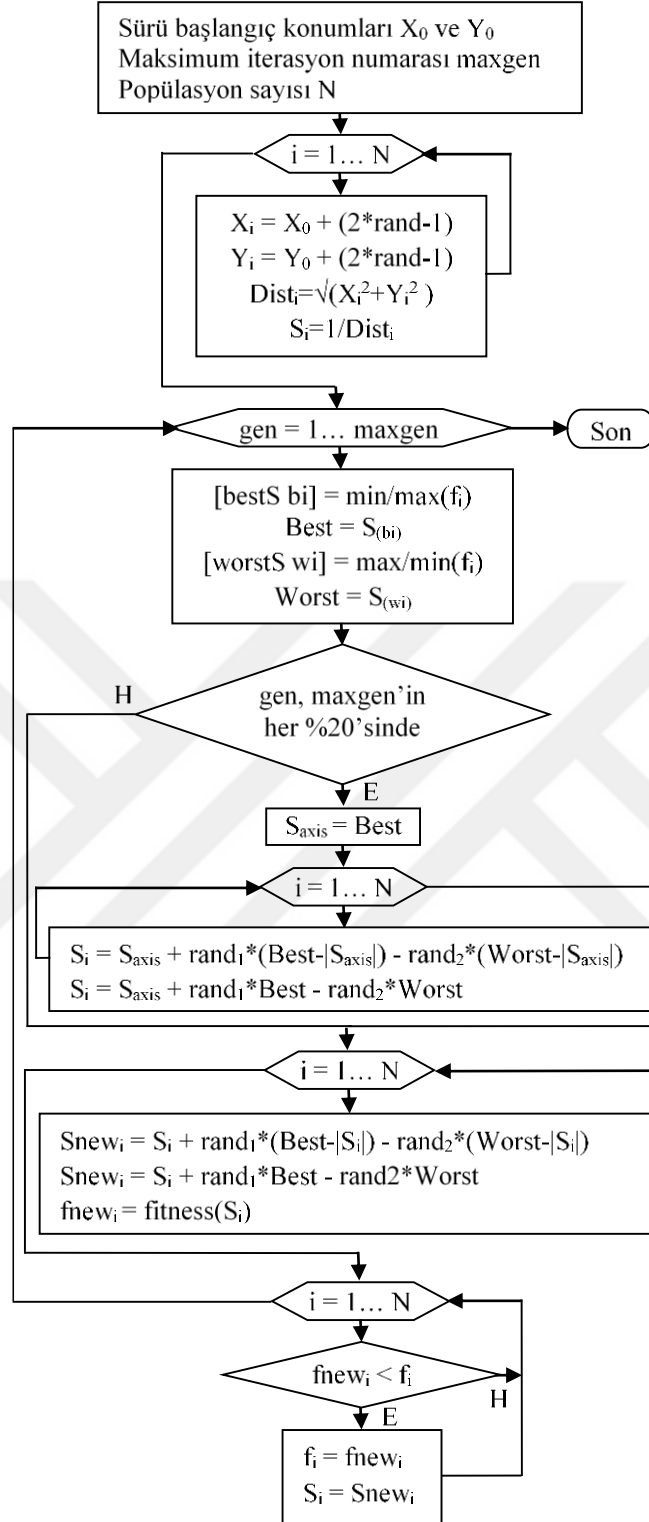
    for i=1,2,..., N
        for j=1,2,..., D
            Snewi,j = Si,j + rand1 * (Bestj - |Si,j|) - rand2 * (Worstj - |Si,j|) // versiyon 1
            Snewi,j = Si,j + rand1 * Bestj - rand2 * Worstj // versiyon 2
        endfor
        fnewi = fitness( $\vec{Snew}_i$ )
    endfor

    for i=1,2,..., N
        if fnewi < fi
            fi = fnewi &  $\vec{S}_i = \vec{Snew}_i$ 
        endif
    endfor

endfor

```

Şekil 3.7. pFOA'nın Sözde Kodu



Şekil 3.8. pFOA'nın Akış Şeması

3.6. Deneylerde Kullanılan Kıyaslama Fonksiyonları

Önerilen üç algoritma SFOA, saFOA ve pFOA, literatürde yaygın olarak kullanılan, iyi bilinen 21 tanınmış sayısal kıyaslama fonksiyonları üzerinde test edilmiştir. Bu fonksiyonlar ve özellikleri Çizelge 3.1'de verilmiştir.

Aynı zamanda, önerilen üçüncü algoritma olan pFOA, literatürdeki üç farklı çalışmanın (Chen ve Pi, 2019; Liu ve ark., 2019; Wang ve ark., 2019) sonuçları ile karşılaştırılması için çalışmalarda ortak kullanılan 16 sayısal kıyaslama fonksiyonları üzerinde test edilmiştir. Bu kıyaslama fonksiyonları ve özellikleri bölüm 4.3'deki Çizelge 4.11'de verilmiştir.

Kıyaslama fonksiyonları, özelliklerine göre birbirinden farklıdır. Bu fonksiyonlar da iki önemli farklılaşma grubu vardır. Fonksiyonlar tek modlu (unimodal) veya çok modlu (multimodal), ayrılabilir veya ayrılamaz olabilir. Tek modlu fonksiyonlar tek yerel (local) optimuma sahip olarak tanımlanır ve bu aynı zamanda genel (global) optimumdur. Çok modlu fonksiyonların birden fazla yerel optimumu ve bir veya daha fazla genel optimumu vardır. Ayrıca, ayrılabilir fonksiyonlar n adet fonksiyonun toplamı olarak yazılabilirken, ayrılamayan fonksiyonlar değişkenler arasındaki ilişkilerden dolayı ayrılabilenler gibi parçalanamazlar (Karaboga ve Akay, 2009; Kiran ve ark., 2015). Bir metodun yerel arama yeteneği tek modlu fonksiyonlarla test edilirken, genel ve yerel arama yetenekleri ise çok modlu fonksiyonlarla test edilir. Ayrılabilir fonksiyonlardaki karar değişkenleri arasında bir ilişki olmadığı için bu tip problemlerin çözümü ayrılamayan fonksiyonların çözümünden daha kolaydır. Kıyaslama fonksiyonları değerlendirildiğinde boyutluluk, arama uzayının üstel olarak genişlemesi sebebiyle hesaplama karmaşıklığını ve dolayısıyla zaman ve çözümün karmaşıklığını etkileyen diğer bir önemli faktördür.

Çizelge 3.1'de, n fonksiyonların boyutunu göstermektedir. Düşük boyutlu fonksiyonlar kolayca çözülebilir çünkü fonksiyonların boyutlarını arttırdığımızda, çözüm alanı katlanarak büyür. Bu nedenle, yüksek boyutlu fonksiyonların çözümü, düşük boyutlu fonksiyonların çözümlerinden daha zordur. SR sütunu arama aralığı göstermektedir ve bu, karar değişkenlerinin bu aralıktaki değerleri alabileceği anlamına gelir. Bir karar değeri üst sınırdan yüksek veya alt sınırdan düşükse, bu değerlere atanır. Fonksiyonların bir diğer önemli özelliği ise fonksiyonların karakteristikleridir ve C sütununda gösterilmiştir. U (unimodal), M (multimodal), S (separable), N (non-separable), sırasıyla tek modlu, çok modlu, ayrılabilir ve ayrılmaz anlamına gelir. Kısaca,

önerilen algoritmaların performansı bu fonksiyonlar kullanılarak bu fonksiyonlar üzerinde incelenmiştir. Çalışmada ele alınan kıyaslama problemleri iki farklı perspektif açısından zor problemlerdir. Bunlardan biri problemlerin özellikleri yani tek ve çok modlu olmasıdır. Karşılaştırılan algoritmaların yakınsama özellikleri, tek modlu problemler üzerinde test edilir ve çok modlu problemler, algoritmaların genel arama özelliğini inceler. Bazı problemler tek modlu olsalar da diğerleri çok modludur. Algoritmalar için diğer zorluk, optimizasyon problemlerinin boyutudur. Önerilen algoritmaların arama kapasitesini doğrulamak için ilk çalışmada 30,60 ve 90, ikinci ve üçüncü çalışmada 10, 30 ve 50 boyutlu problemler ele alınmıştır. Bu nedenle, algoritmaların etkinliğini test etmek için farklı özelliklere ve boyutlara sahip 21 kıyaslama fonksiyonunun yeterli olduğu öngörülmüştür.

Çizelge 3.1.a. Kıyaslama Fonksiyonları

Num Adı	SR	C	Fonksiyon
1 Sphere	[-100,100]	US	$f_1(\vec{X}) = \sum_{i=1}^n x_i^2$
2 Elliptic	[-100,100]	UN	$f_2(\vec{X}) = \sum_{i=1}^n (10^6)^{(i-1)/(n-1)} x_i^2$
3 SumSquares	[-10,10]	US	$f_3(\vec{X}) = \sum_{i=1}^n ix_i^2$
4 SumPower	[-10,10]	MS	$f_4(\vec{X}) = \sum_{i=1}^n x_i ^{(i+1)}$
5 Schwefel2.22	[-10,10]	UN	$f_5(\vec{X}) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $
6 Schwefel2.21	[-100,100]	UN	$f_6(\vec{X}) = \max_i \{ x_i , 1 \leq i \leq n \}$
7 Step	[-100,100]	US	$f_7(\vec{X}) = \sum_{i=1}^n (\lfloor x_i + 0.5 \rfloor)^2$
8 Quartic	[-1.28,1.28]	US	$f_8(\vec{X}) = \sum_{i=1}^n ix_i^4$
9 QuarticWN	[-1.28,1.28]	US	$f_9(\vec{X}) = \sum_{i=1}^n ix_i^4 + \text{random}[0,1)$
10 Rosenbrock	[-10,10]	UN	$f_{10}(\vec{X}) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
11 Rastrigin	[-5.12,5.12]	MS	$f_{11}(\vec{X}) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$
			$f_{12}(\vec{X}) = \sum_{i=1}^n [y_i^2 - 10 \cos(2\pi y_i) + 10]$
12 Non-Continuous Rastrigin	[-5.12,5.12]	MS	$y_i = \begin{cases} x_i & x_i < \frac{1}{2} \\ \frac{\text{round}(2x_i)}{2} & x_i \geq \frac{1}{2} \end{cases}$

Çizelge 3.1.b. Kıyaslama Fonksiyonları

Num Adı	SR	C	Fonksiyon
13 Griewank	[-600,600]	MN	$f_{13}(\vec{X}) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
14 Schwefel2.26	[-500,500]	UN	$f_{14}(\vec{X}) = 418.98 * n - \sum_{i=1}^n x_i \sin\left(\sqrt{ x_i }\right)$
15 Ackley	[-32,32]	MN	$f_{15}(\vec{X}) = -20 \exp\left\{-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right\} - \exp\left\{\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right\} + 20 + e$
16 Penalized1	[-50,50]	MN	$f_{16}(\vec{X}) = \frac{\pi}{n} \{10 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{1}{4}(x_i + 1)$ $u_{x_i, a, k, m} = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leq x_i \leq a \\ k(x_i + a)^m & x_i < -a \end{cases}$
17 Penalized2	[-50,50]	MN	$f_{17}(\vec{X}) = \frac{1}{10} \left\{ \sin^2(\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2 [1 + \sin^2(2\pi x_{i+1})] \right\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$
18 Alpine	[-10,10]	MS	$f_{18}(\vec{X}) = \sum_{i=1}^n x_i \cdot \sin(x_i) + 0.1 \cdot x_i $
19 Levy	[-10,10]	MN	$f_{19}(\vec{X}) = \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + \sin^2(3\pi x_1) + x_n - 1 [1 + \sin^2(3\pi x_n)]$
20 Weierstrass	[-0.5,0.5]	MN	$f_{20}(\vec{X}) = \sum_{i=1}^D \left(\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_i + 0.5))] \right) - D \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k 0.5)]$ $a = 0.5, b = 3, k_{\max} = 20$
21 Schaffer	[-100,100]	MN	$f_{21}(\vec{X}) = 0.5 + \frac{\sin^2\left(\sqrt{\sum_{i=1}^n x_i^2}\right) - 0.5}{\left(1 + 0.001 * \left[\sum_{i=1}^n x_i^2\right]\right)^2}$

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

4.1. SFOA'nın Deneysel Sonuçları

Önerilen SFOA'nın performansı, Çizelge 3.1'deki 21 tanınmış nümerik kıyaslama fonksiyonu üzerinde incelenmiştir. Önerilen yaklaşımın etkinliğini göstermek için SFOA'nın sonuçları, temel FOA, PSO'nun gelişmiş versiyonlarından olan ve Omran tarafından önerilen SPSO2007 (Omran, 2007), PSO'nun bir diğer versiyonu SPSO2011(Bonyadi ve Michalewicz, 2014), Guguk Kuşu algoritması (Cuckoo Search, CS)(Yang ve Deb, 2009) ve Ateşböceği algoritması (Firefly Algorithm, FA)(Yang, 2009) sonuçları ile karşılaştırılmıştır. Maksimum yineleme sayısı, tüm algoritmalar için 300 ve 500 yinelemeye eşit olacak şekilde sonlandırma koşulu olarak kullanılmıştır. Ayrıca, algoritmanın etkinliği, 30, 60 ve 90 gibi değişen boyutlara sahip kıyaslama fonksiyonlarında değerlendirilmiştir. Popülasyon, karşılaştırmalarda kullanılan tüm algoritmalar için, kıyaslama fonksiyonunun boyutlarına bağlı olarak kullanılmış ve Denklem 4.1 ile hesaplanmıştır.

$$N = \text{floor} (10 + 2 * \sqrt{D}) \quad (4.1)$$

D, değerlendirmede kıyaslama fonksiyonunun boyutunu belirtir. Bu denklem, Omran tarafından önerilen, SPSO2007 algoritması içinde kullanılmaktadır (Omran, 2007). Bu varsayıma göre, tüm kıyaslama fonksiyonlarında, sorunun boyutuna bağlı olarak değişen popülasyon büyüklükleri 20, 25 ve 28 olarak değerlendirilmiştir.

Sonuçların güvenilirliğini, istikrarlılığını ve sağlamlığını göstermek için, tüm algoritmalar rastgele tohumlarla 30 kez gerçekleştirilmiştir, ortalama ve standart sapma değerleri sırasıyla 300 ve 500 yineleme için Çizelge 4.1 ve 4.2'de rapor edilmiştir.

Tüm algoritmalarda eşit olarak kullanılan genel rekabet parametrelerine ek olarak, SPSO2007 ve SPSO2011'in sosyal ve bilişsel parametreleri ile atalet ağırlık parametreleri, sırasıyla 1.1931, 1.1931 ve 0.7213'e eşit olarak, orijinal yaklaşımda olduğu gibi kullanılmıştır (Omran, 2007). CS algoritmasında yabancı bir yumurtanın keşif hızı 0.25'e eşit olarak kullanılmıştır. Işık soğurma katsayısı, çekim katsayısı baz değeri, mutasyon katsayısı ve mutasyon katsayısı sönmüleme oranı FA'da sırasıyla 1, 2, 0.2 ve 0.95'e eşit olarak kullanılmıştır.

Önerilen yaklaşım SFOA'nın etkinliğini ve başarısını göstermek için, SFOA ve FOA'nın yakınsaklıklarının karşılaştırılması sırasıyla 30, 60 ve 90 boyutsal fonksiyonlar için Şekil 4.1, 4.2 ve 4.3'te verilmiştir.

SFOA, FOA, PSO2007, PSO2011, CS ve FA'nın 300 ve 500 tekrarlama ile elde edilen karşılaştırmalı tablolarda verilen sonuçları, rank analizi ve parametrik olmayan Wilcoxon işaretli sıralama testi ile karşılaştırılmıştır. Parametrik olmayan Wilcoxon işaretli sıralama testi ile iki yöntemin sonucu birbiriyle karşılaştırılır ama iki yöntem arasında hangisinin iyi olduğunu göstermez. Wilcoxon işaretli sıralama testi, iki yöntem arasında 0.05 anlamlılık düzeyinde fark olup olmadığını gösterir. (+), test edilen iki sonuç arasında anlamlı bir fark olduğunu, (-) ise testin sonuçlarında anlamlı bir fark olmadığını belirtir. Algoritmaların sıralama değerleri, algoritmalarla elde edilen sonuçların artan sırasına göre verilmiştir. Sıralama analiz işlemi en iyi (en az) değere sahip algoritma için sıralama değeri olarak 1, en kötü (en yüksek) değere sahip algoritma için 6 değeri verilmiştir.

Çizelge 4.1 ve 4.2'deki 300 ve 500 yineleme için yapılan değerlendirmelerin sonuçlarına göre, SFOA ve FA sonuçlarının neredeyse tüm kıyaslama fonksiyonları için diğer algoritmalarından daha başarılı sonuçlar elde ettiği söylenebilir. FOA, 300 iterasyonda 30, 60 ve 90 boyutlarındaki step fonksiyonu için SFOA'nın sonucuna eşit olan en iyi sonucu bulmuştur. Ayrıca, Levy işlevi için SPSO 2007 algoritması, 300 yineleme için 30 boyut için en iyisini sağlamıştır.

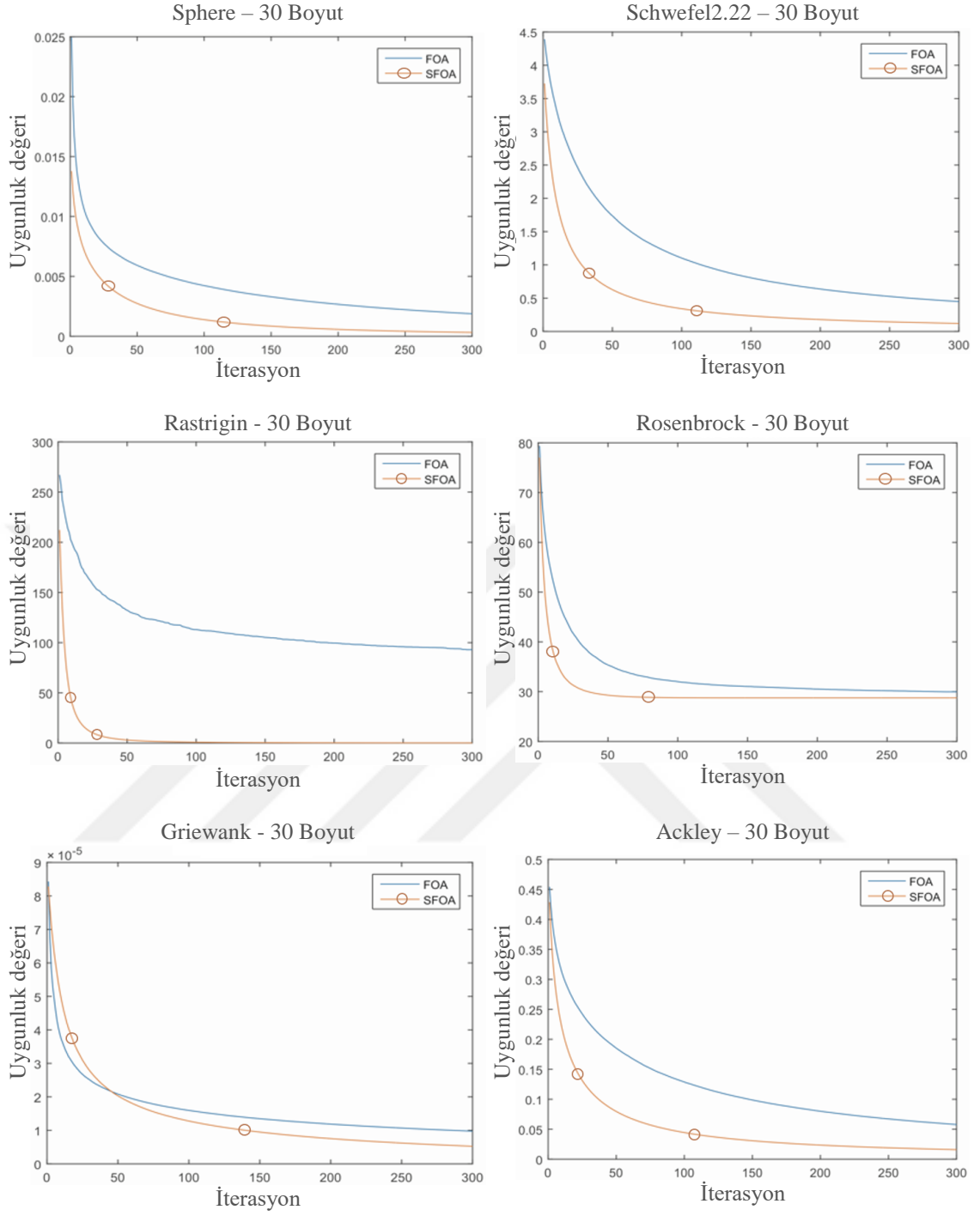
Çizelge 4.1'deki 300 yineleme için yapılan değerlendirmelerin sonuçlarına göre FA, kıyaslama fonksiyonlarından SumSquares, Schwefel2.22, Quartic, Schwefel2.26, Levy ve Weierstrass için neredeyse her üç boyutta en iyi sonuçları elde etmiştir. SFOA, kıyaslama fonksiyonlarının geri kalanı için, neredeyse her üç boyutta da en iyi sonuçları elde etmiştir.

Çizelge 4.2'deki 500 yinelemeye ilişkin değerlendirmelerin sonuçlarına göre FA, kıyaslama fonksiyonlarından Sphere, Elliptic, SumSquares, Schwefel2.22, Quartic, Schwefel2.26, Penalized1, Penalized2, Alpine, Levy and Weierstrass için neredeyse üç boyutta en iyi sonuçları elde etmiştir. SFOA, kıyaslama fonksiyonlarının geri kalanı için, neredeyse her üç boyutta da en iyi sonuçları elde etmiştir.

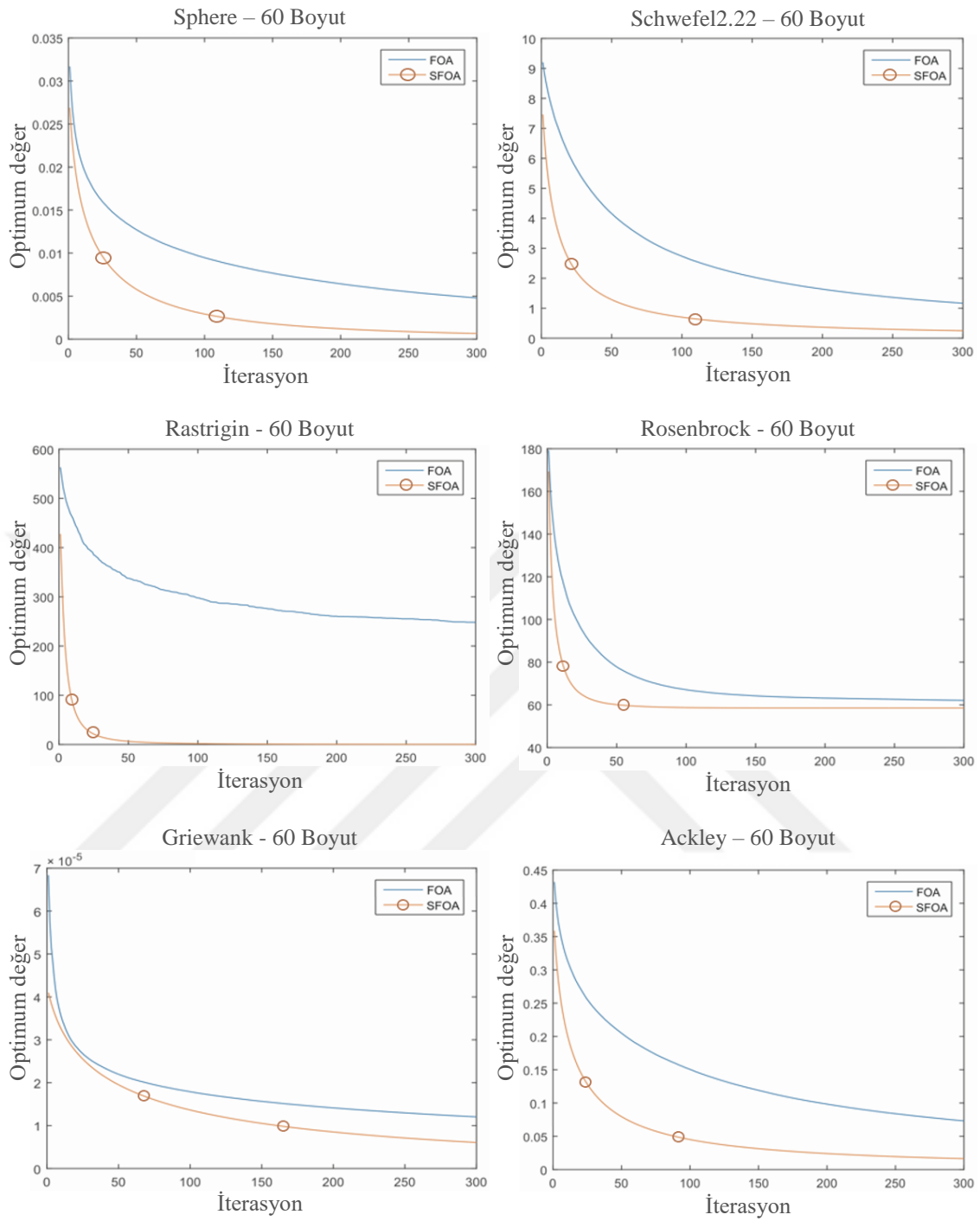
Çizelge 4.1 ve 4.2'deki sıra analizi sonuçları, SFOA'nın FOA, SPSO2007, SPSO2011, CS ve FA ile karşılaştırıldığında, çözüm kalitesi ve sağlamlığı bakımından oldukça iyi performans gösterdiğini ortaya koymaktadır. Parametrik olmayan Wilcoxon işaretli sıralama test sonuçlarına göre, SFOA'nın performansı, neredeyse tüm değerlendirmelerde FOA, SPSO2007, SPSO2011, CS ve FA açısından istatistiksel olarak anlamlıdır. SFOA'nın yakınsama performansı 30, 60 ve 90 boyutlu fonksiyonlarda FOA'dan daha iyidir.

Çizelge 4.2.b. Maksimum 500 yineleme için algoritmaların karşılaştırmalı sonuçları

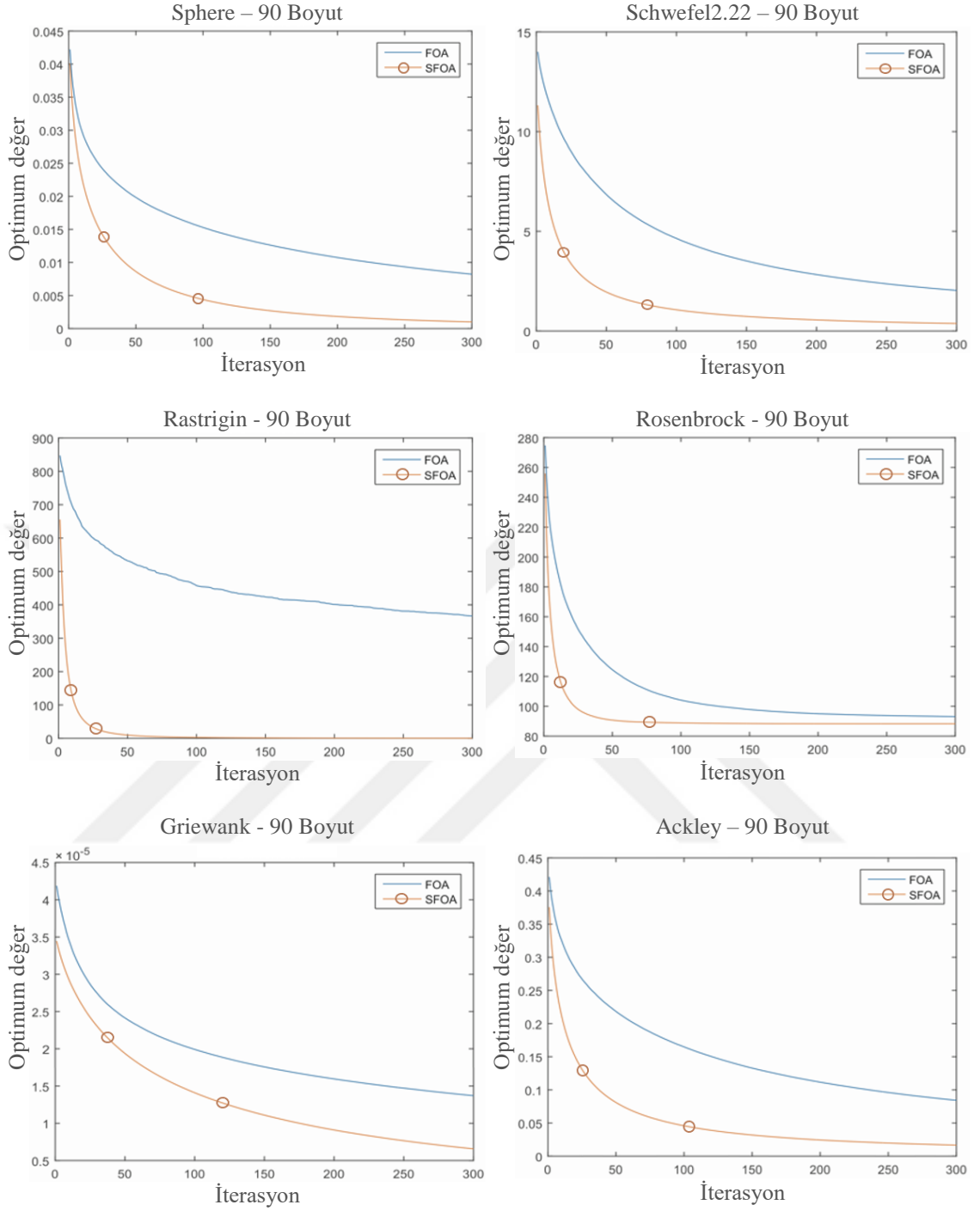
Fonk.	SFOA (Önerilen)			FOA			SPSO2007			SPSO2011			CS			FA			
	D	Mean	Std. Dev. R	Mean	Std. Dev. R	S1	Mean	Std. Dev. R	S2	Mean	Std. Dev. R	S3	Mean	Std. Dev. R	S4	Mean	Std. Dev. R	S5	
Griewank	30	3.08E-06	3.93E-07	1	7.25E-06	1.28E-06	2 +	2.31E-02	3.07E-02	5 +	1.30E-02	1.10E-02	4 +	1.09E+00	5.29E-02	6 +	3.45E-03	5.44E-03	3 -
	60	3.46E-06	3.36E-07	1	9.64E-06	1.71E-06	2 +	1.11E+00	4.23E-01	5 +	2.30E-01	6.73E-02	4 +	5.87E+00	1.14E+00	6 +	4.94E-04	1.88E-03	3 +
	90	3.92E-06	3.14E-07	1	1.08E-05	1.70E-06	2 +	4.79E+00	3.13E+00	5 +	9.91E-01	9.76E-02	4 +	2.20E+01	4.33E+00	6 +	2.06E-03	4.67E-03	3 +
Scwefel2.26	30	1.26E+04	2.62E-03	6	1.25E+04	3.33E+01	5 +	3.79E+03	9.32E+02	1 +	6.78E+03	1.33E+03	4 +	4.58E+03	2.59E+02	3 +	4.23E+03	7.08E+02	2 +
	60	2.51E+04	3.79E-03	6	2.50E+04	2.72E+02	5 +	1.07E+04	2.37E+03	2 +	1.72E+04	2.07E+03	4 +	1.15E+04	3.47E+02	3 +	8.48E+03	7.36E+02	1 +
	90	3.77E+04	4.10E-03	6	3.76E+04	2.07E+02	5 +	1.91E+04	3.58E+03	2 +	2.80E+04	1.81E+03	4 +	1.96E+04	4.47E+02	3 +	1.43E+04	1.17E+03	1 +
Ackley	30	9.71E-03	5.16E-05	2	3.69E-02	5.93E-04	3 +	1.79E+00	8.91E-01	4 +	2.36E+00	5.75E-01	5 +	5.94E+00	1.81E+00	6 +	8.58E-05	5.65E-06	1 +
	60	1.00E-02	3.48E-05	1	4.82E-02	8.40E-04	2 +	3.52E+00	9.15E-01	4 +	3.57E+00	4.73E-01	5 +	1.08E+01	1.60E+00	6 +	9.74E-02	3.71E-01	3 +
	90	1.02E-02	2.40E-05	1	5.62E-02	1.11E-03	2 +	5.22E+00	9.12E-01	5 +	4.57E+00	6.18E-01	4 +	1.23E+01	1.78E+00	6 +	1.74E+00	6.05E-01	3 +
Penalized1	30	1.55E+00	4.18E-02	3	1.70E+00	8.77E-04	4 +	1.53E+00	1.58E+00	2 -	3.49E+00	3.13E+00	5 +	5.66E+00	2.81E+00	6 +	3.43E-01	7.47E-01	1 +
	60	1.34E+00	7.70E-03	2	1.46E+00	6.42E-04	3 +	1.33E+01	5.54E+00	5 +	8.81E+00	2.28E+00	4 +	1.49E+01	6.84E+00	6 +	1.32E+00	1.85E+00	1 -
	90	1.27E+00	6.84E-03	1	1.38E+00	6.87E-04	2 +	1.95E+02	3.95E+02	6 +	1.36E+01	3.56E+00	4 +	6.14E+01	1.75E+02	5 +	3.42E+00	2.77E+00	3 +
Penalized2	30	2.97E+00	3.65E-03	5	2.94E+00	4.99E-02	4 +	1.97E+00	3.46E+00	3 -	9.05E-03	9.74E-03	2 +	2.27E+01	1.10E+01	6 +	1.83E-03	4.16E-03	1 +
	60	5.95E+00	5.69E-03	3	5.92E+00	3.34E-02	2 +	7.47E+01	2.69E+01	4 +	9.83E+01	3.63E+01	5 +	5.56E+03	8.04E+03	6 +	3.83E-01	1.04E+00	1 +
	90	8.93E+00	8.19E-03	3	8.92E+00	4.54E-02	2 -	3.40E+04	7.42E+04	5 +	1.93E+02	1.98E+01	4 +	5.34E+04	5.80E+04	6 +	7.68E+00	8.21E+00	1 -
Alpine	30	4.42E-02	3.52E-02	3	3.10E-02	4.71E-04	2 +	6.55E-02	2.20E-01	4 -	9.72E-01	8.90E-01	5 +	1.01E+01	1.94E+00	6 +	3.68E-04	4.69E-04	1 +
	60	1.57E-02	6.64E-04	2	3.09E-01	5.82E-01	3 +	1.14E+00	9.41E-01	4 +	6.57E+00	2.72E+00	5 +	3.24E+01	3.04E+00	6 +	2.65E-03	1.68E-03	1 +
	90	2.36E-02	4.10E-04	2	2.60E-01	5.93E-01	3 +	7.95E+00	3.93E+00	4 +	1.54E+01	4.54E+00	5 +	5.55E+01	6.08E+00	6 +	1.50E-02	5.03E-03	1 +
Levy	30	2.97E+01	3.09E-02	6	2.54E+01	2.80E+00	5 +	5.87E-02	1.14E-01	1 +	1.05E+00	1.73E+00	3 +	3.32E+00	2.21E+00	4 +	3.77E-01	1.37E+00	2 +
	60	5.95E+01	6.95E-02	5	5.34E+01	3.39E+00	4 +	2.78E+00	4.95E+00	3 +	2.77E+00	4.08E+00	2 +	6.21E+01	1.45E+01	6 -	2.12E+00	3.96E+00	1 +
	90	8.92E+01	6.50E-02	5	8.29E+01	4.43E+00	4 +	2.66E+01	1.58E+01	3 +	2.16E+01	1.19E+01	2 +	1.68E+02	2.49E+01	6 +	2.59E+00	4.24E+00	1 +
Weierstrass	30	4.74E+01	1.77E+00	6	4.62E+01	1.50E+00	5 +	2.05E+00	1.43E+00	2 +	9.77E+00	1.89E+00	3 +	1.11E+01	1.99E+00	4 +	7.02E-01	1.07E+00	1 +
	60	9.80E+01	3.06E+00	5	9.95E+01	3.05E+00	6 -	9.81E+00	2.57E+00	2 +	2.87E+01	3.86E+00	3 +	3.41E+01	2.69E+00	4 +	3.85E+00	2.86E+00	1 +
	90	1.50E+02	3.54E+00	5	1.51E+02	3.38E+00	6 -	2.50E+01	4.26E+00	2 +	5.00E+01	3.60E+00	3 +	5.80E+01	4.20E+00	4 +	1.35E+01	5.22E+00	1 +
Schaffer	30	1.35E-04	3.16E-06	1	1.07E-03	7.60E-05	2 +	1.79E-01	5.47E-02	5 +	1.51E-01	3.18E-02	4 +	4.71E-01	1.27E-02	6 +	1.41E-01	4.35E-02	3 +
	60	2.84E-04	4.82E-06	1	2.98E-03	1.52E-04	2 +	4.83E-01	9.53E-03	5 +	4.04E-01	2.90E-02	4 +	4.97E-01	9.50E-04	6 +	3.64E-01	4.67E-02	3 +
	90	4.41E-04	4.70E-06	1	5.45E-03	3.11E-04	2 +	4.99E-01	5.34E-04	5 +	4.83E-01	6.77E-03	4 +	4.99E-01	2.48E-04	6 +	4.71E-01	1.48E-02	3 +
30	mean R					2.67		3.43		3.29		3.95		5.43		2.00			
	final R					2		4		3		5		6		1			
60	mean R					2.05		3.05		4.10		4.00		5.48		2.10			
	final R					1		3		5		4		6		2			
90	mean R					1.95		2.81		4.38		3.76		5.48		2.38			
	final R					1		3		5		4		6		2			



Şekil 4.1. 30 boyut için bazı fonksiyonlardaki yakınsama grafikleri



Şekil 4.2. 60 boyut için bazı fonksiyonlardaki yakınsama grafikleri



Şekil 4.3. 90 boyut için bazı fonksiyonlardaki yakınsama grafikleri

4.2. saFOA Algoritmasının Deneysel Sonuçları

FOA ve saFOA, Çizelge 3.1'deki 21 tanınmış sayısal kıyaslama fonksiyonu ile test edilmiştir. Testler 10, 30 ve 50 boyutlarında yapılmıştır. Testlerde, 10 boyut için 100000 yineleme, 30 boyut için 300000 yineleme ve 50 boyut için 500000 fonksiyon değerlendirme sayısı dikkate alınmaktadır. Popülasyon büyüklüğü 20, sıcaklık değeri 2000, soğutma katsayısı 0.95 olarak alınmıştır. Her bir test durumu için, algoritmaların istatistiksel olarak karşılaştırılması amacıyla tüm boyutlar için 51 bağımsız çalışma gerçekleştirilmiştir. Çizelge 4.3 testlerin sonuçlarını göstermektedir.

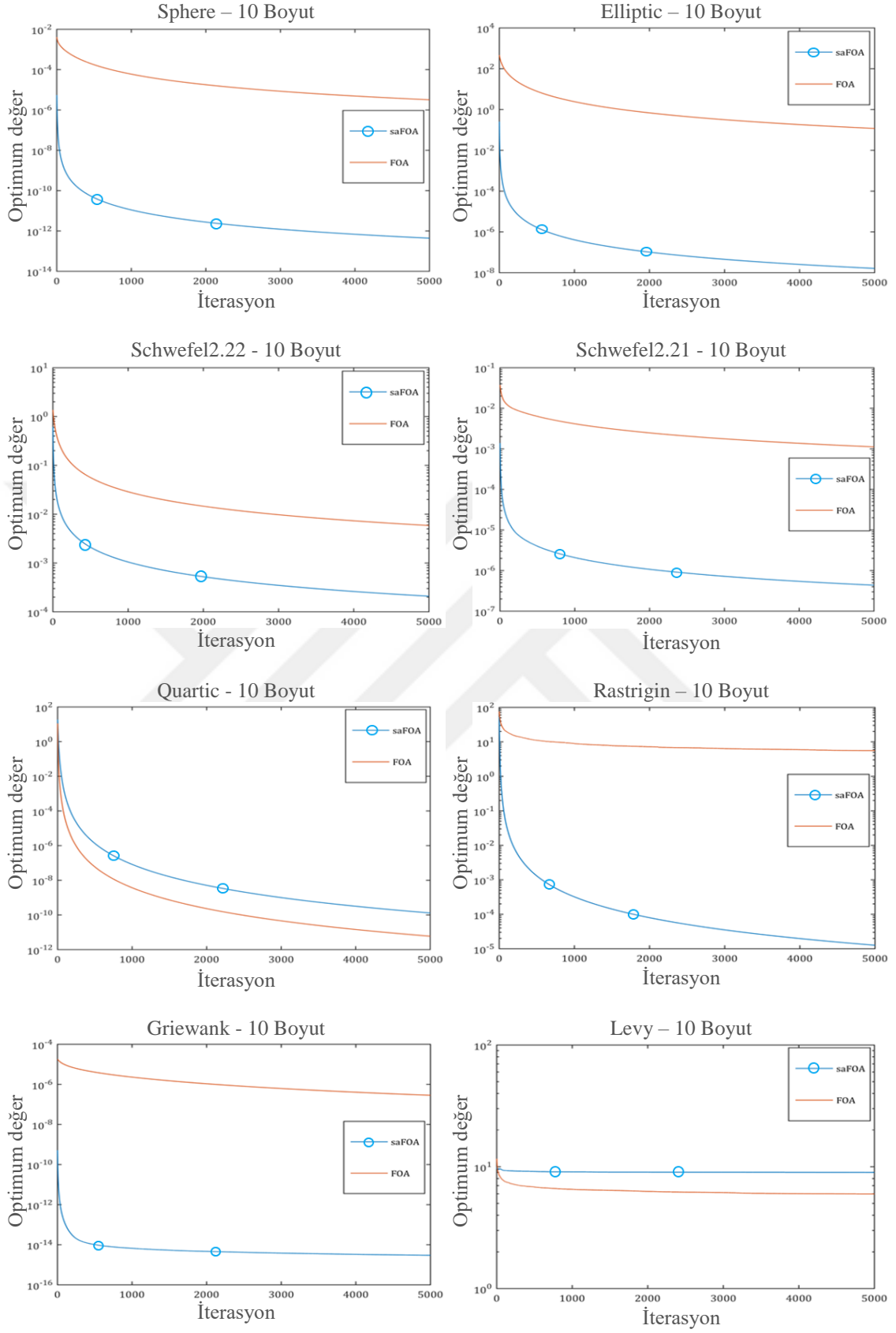
Çizelge 4.3'e göre, saFOA, her üç boyutta 13 fonksiyonda FOA'dan daha iyi sonuçlar vermiştir. 2 fonksiyonda saFOA ve FOA aynı değerleri bulmuştur. Sadece 6 fonksiyonda, FOA daha iyi sonuçlar elde etmiştir.

saFOA ve FOA'nın bazı fonksiyonlardaki yakınsama eğrileri Şekil 4.4, 4.5 ve 4.6'da verilmiştir. Her üç boyutta, yakınsama grafiklerine göre saFOA'nın yakınsama performansı FOA'dan daha iyidir.

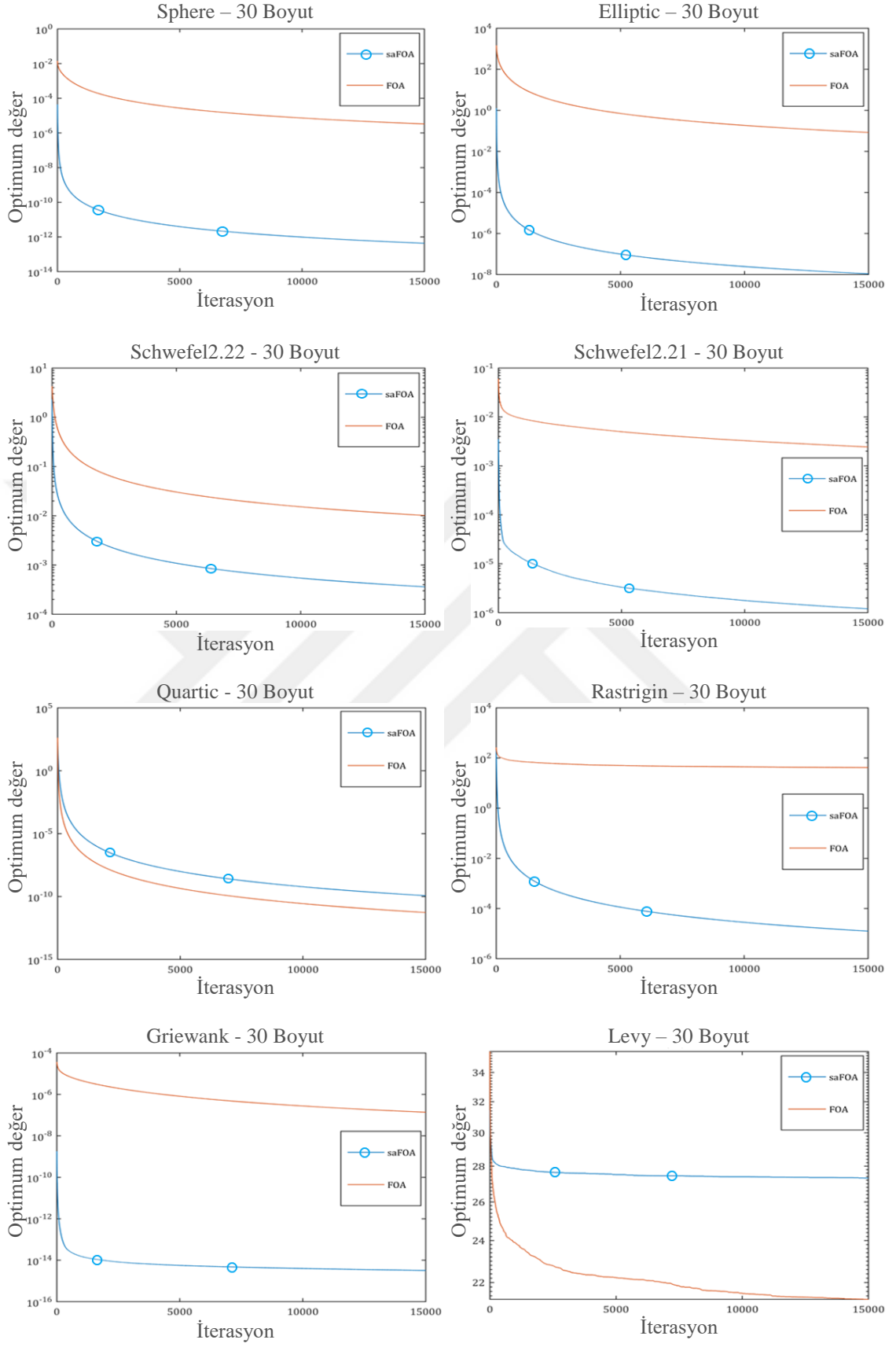
Deneysel sonuçlara göre, saFOA, belli oranda kötü çözümleride dikkate alarak çözüm kalitesi ve dayanıklılığı açısından temel FOA'nın performansını her üç boyutta da iyileştirmiştir.

Çizelge 4.3. saFOA ve temel FOA'nın 10, 30 ve 50 boyutları için karşılaştırmalı sonuçları.

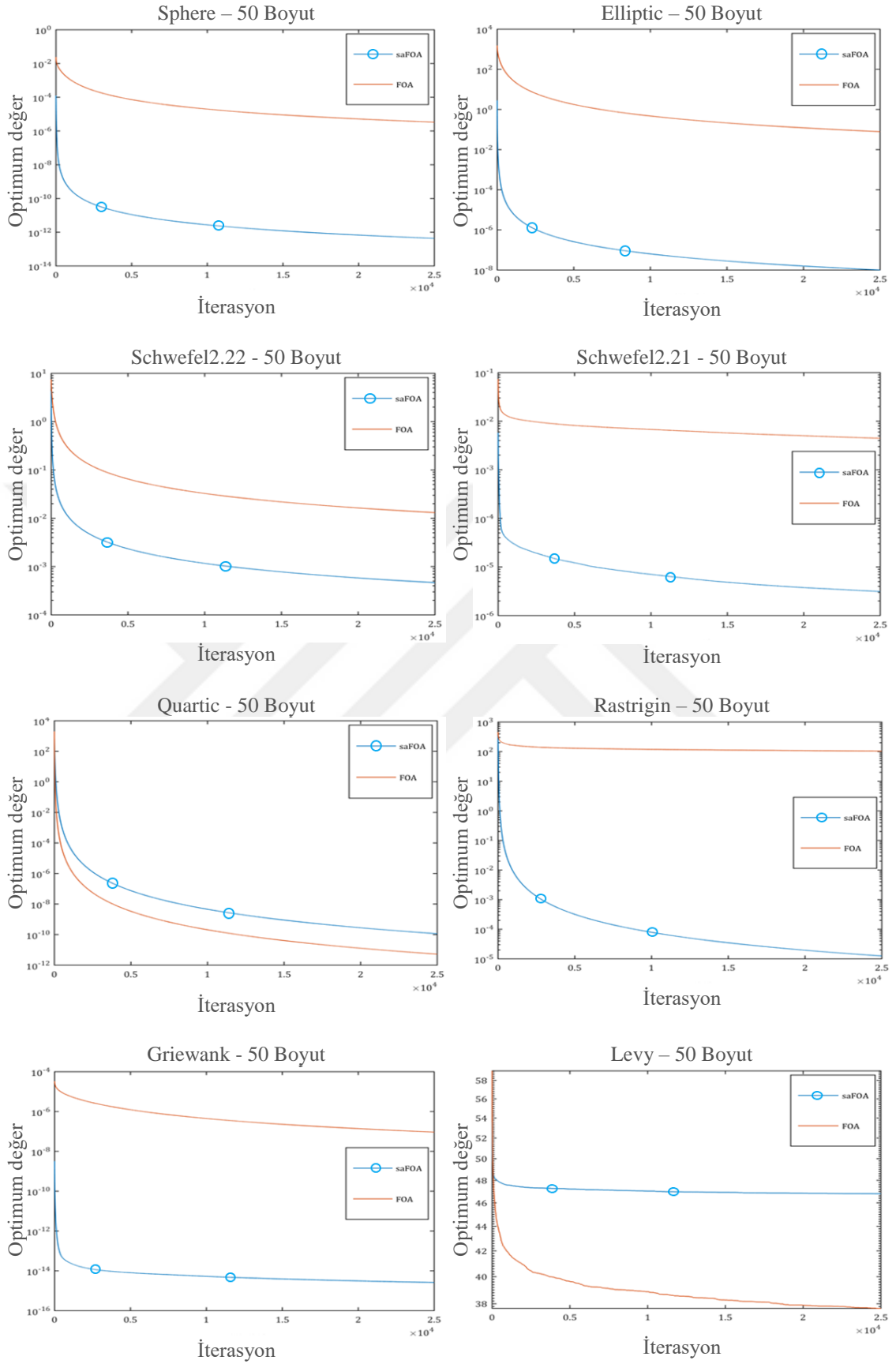
Fonk.	D	saFOA (Önerilen)		FOA		D	saFOA (Önerilen)		FOA		D	saFOA (Önerilen)		FOA	
		Mean	Std. Dev.	Mean	Std. Dev.		Mean	Std. Dev.	Mean	Std. Dev.		Mean	Std. Dev.	Mean	Std. Dev.
Sphere	10	4,39E-13	2,20E-13	3,21E-06	1,52E-06	30	4,35E-13	2,18E-13	3,30E-06	1,56E-06	50	4,35E-13	2,18E-13	3,33E-06	1,57E-06
Elliptic	10	1,63E-08	6,51E-09	1,18E-01	3,72E-02	30	1,09E-08	4,36E-09	8,36E-02	2,63E-02	50	1,01E-08	4,05E-09	7,81E-02	2,46E-02
SumSquares	10	2,22E-08	7,34E-09	1,74E-05	4,35E-06	30	6,07E-08	2,01E-08	4,80E-05	1,20E-05	50	9,93E-08	3,29E-08	7,87E-05	1,96E-05
SumPower	10	6,23E-11	1,79E-11	4,06E-08	8,66E-09	30	6,93E-12	1,99E-12	4,43E-09	9,43E-10	50	2,49E-12	7,17E-13	1,58E-09	3,37E-10
Schwefel2.22	10	2,09E-04	5,39E-05	5,87E-03	1,11E-03	30	3,61E-04	9,29E-05	1,02E-02	1,92E-03	50	4,65E-04	1,20E-04	1,31E-02	2,48E-03
Schwefel2.21	10	4,37E-07	1,03E-07	1,12E-03	1,92E-04	30	1,20E-06	2,84E-07	2,44E-03	4,18E-04	50	3,13E-06	7,42E-07	4,46E-03	7,66E-04
Step	10	0,00E+00	0,00E+00	0,00E+00	0,00E+00	30	0,00E+00	0,00E+00	0,00E+00	0,00E+00	50	0,00E+00	0,00E+00	0,00E+00	0,00E+00
Quartic	10	1,30E-10	2,65E-11	5,93E-12	8,75E-13	30	1,17E-10	2,39E-11	5,35E-12	7,89E-13	50	1,14E-10	2,34E-11	5,23E-12	7,71E-13
QuarticWN	10	7,71E-03	1,51E-03	2,77E-03	3,93E-04	30	1,59E-02	3,07E-03	5,78E-03	8,09E-04	50	2,31E-02	4,46E-03	7,82E-03	1,09E-03
Rosenbrock	10	7,94E+00	1,45E+00	6,99E+00	9,19E-01	30	2,78E+01	5,08E+00	2,75E+01	3,61E+00	50	4,76E+01	4,23E-01	4,80E+01	6,31E+00
Rastrigin	10	1,26E-05	2,20E-06	5,58E+00	1,04E+00	30	1,25E-05	2,18E-06	4,16E+01	6,21E+00	50	1,25E-05	4,59E-08	1,04E+02	1,42E+01
Non-Cont. Rastrigin	10	1,96E-02	2,37E-02	1,38E+00	2,09E-01	30	1,25E-05	2,09E-06	1,49E+01	3,32E+00	50	3,92E-02	1,39E-01	7,10E+01	1,07E+01
Griewank	10	2,96E-15	4,76E-16	2,85E-07	3,28E-08	30	3,22E-15	5,18E-16	1,38E-07	1,58E-08	50	2,60E-15	1,24E-15	9,30E-08	1,07E-08
Sewefel2.26	10	4,19E+03	6,46E+02	4,00E+03	4,41E+02	30	1,26E+04	1,94E+03	1,22E+04	1,35E+03	50	2,09E+04	9,09E+03	2,04E+04	2,25E+03
Ackley	10	8,17E-06	1,22E-06	2,35E-03	2,50E-04	30	4,70E-06	7,01E-07	1,35E-03	1,44E-04	50	3,64E-06	1,46E-06	1,05E-03	1,12E-04
Penalized1	10	2,65E+00	3,83E-01	2,65E+00	2,74E-01	30	1,67E+00	2,41E-01	1,67E+00	1,72E-01	50	1,47E+00	2,62E-09	1,47E+00	1,52E-01
Penalized2	10	9,31E-01	1,30E-01	9,54E-01	9,57E-02	30	2,88E+00	4,03E-01	2,90E+00	2,90E-01	50	4,86E+00	2,44E+00	4,85E+00	4,85E-01
Alpine	10	2,09E-05	2,85E-06	3,00E-03	1,69E-03	30	3,61E-05	4,91E-06	1,62E-02	7,50E-03	50	4,65E-05	2,20E-05	1,19E-02	7,37E-03
Levy	10	8,99E+00	1,20E+00	5,98E+00	5,82E-01	30	2,73E+01	3,62E+00	2,12E+01	2,02E+00	50	4,68E+01	5,08E-01	3,77E+01	3,57E+00
Weierstrass	10	3,86E+01	4,99E+00	1,02E+01	9,38E-01	30	1,18E+02	1,53E+01	4,43E+01	4,09E+00	50	1,98E+02	9,95E+01	8,02E+01	7,38E+00
Schaffer	10	4,43E-13	5,58E-14	3,21E-06	2,89E-07	30	4,61E-13	5,81E-14	3,30E-06	2,97E-07	50	4,94E-13	2,33E-13	1,94E-04	1,23E-04



Şekil 4.4. 10 boyut için bazı fonksiyonlardaki yakınsama grafikleri



Şekil 4.5. 30 boyut için bazı fonksiyonlardaki yakınsama grafikleri



Şekil 4.6. 50 boyut için bazı fonksiyonlardaki yakınsama grafikleri

4.3. pFOA Algoritmasının Deneysel Sonuçları

Önerilen pFOA_v1 ve pFOA_v2'nin performansı Çizelge 3.1'de gösterilen kıyaslama fonksiyonları üzerinde araştırılmıştır.

PFOA_v1 ve pFOA_v2 tarafından elde edilen sonuçlar, temel FOA, SFOA, SPSO2011, FA, Ağaç Tohum Algoritması (TSA) (Kiran, 2015), CS ve JAYA sonuçları ile karşılaştırılmıştır. Bu algoritmaların kullanımının sebepleri, önerilen algoritma ile aynı kategoride ve bu çalışmada kullanılan kıyaslama fonksiyonlarının çözülmesinde başarılı olan algoritmalar olmalarıdır. Algoritmanın kontrol parametrelerinden popülasyon büyüklüğü tüm deneyler için 20 olarak alınmıştır. SPSO2011'in sosyal ve bilişsel parametreleri ile atalet ağırlık parametreleri, sırasıyla 1.1931, 1.1931 ve 0.7213'e eşit olarak, orijinal yaklaşımda olduğu gibi kullanılmıştır (Omran, 2007). CS algoritmasında yabancı bir yumurtanın keşif hızı 0.25'e eşit olarak kullanılmıştır. Işık soğurma katsayısı, çekim katsayısı baz değeri, mutasyon katsayısı ve mutasyon katsayısı sönümlenme oranı FA'da sırasıyla 1, 2, 0.2 ve 0.95'e eşit olarak kullanılmıştır. Adil bir karşılaştırma yapmak için her bir algoritma, her bir fonksiyonda rastgele tohumlarla 51 kez çalıştırılmıştır. Fonksiyonlar 10, 30 ve 50 boyutlu fonksiyonlardır. Algoritmalarda sonlandırma koşulu için fonksiyon değerlendirme sayısı kullanılmış ve 10 boyutlu fonksiyonlar için 100000, 30 boyutlu fonksiyonlar için 300000 ve 50 boyutlu fonksiyonlar için 500000 olarak belirlenmiştir.

İlk olarak, pFOA_v1 ve pFOA_v2, temel FOA ile Çizelge 4.4'de 10, 30 ve 50 boyutlu fonksiyonlarda karşılaştırılmıştır. Çizelge 4.4'den görüldüğü gibi,

- Step fonksiyonunda her üç algortmada her üç boyutta optimum değeri bulmuştur.
- pFOA_v1, FOA'dan her üç boyutta 3 fonksiyon dışında daha iyi sonuçlar elde etmiştir.
- pFOA_v2, FOA'dan 10 boyutta 1 fonksiyon dışında, 30 ve 50 boyutta ise tüm fonksiyonlarda daha iyi sonuçlar elde etmiştir.
- pFOA_v2, pFOA_v1'den 10 ve 30 boyutta 4 fonksiyon dışında, 50 boyutta ise 5 fonksiyon dışında daha iyi sonuçlar elde etmiştir.

İkinci olarak, pFOA_v1, temel FOA, SFOA, SPSO2011, FA, TSA, CS, JAYA algoritmaları ile 10 boyutlu fonksiyonlar için Çizelge 4.5'te, 30 boyutlu fonksiyonlar için Çizelge 4.6'da ve 50 boyutlu fonksiyonlar için Çizelge 4.7'de karşılaştırılmıştır.

Karşılaştırma tabloları göz önüne alındığında her üç boyutta, pFOA_v1'in sırası 1'dir ve bu nedenle, pFOA_v1'in sıra ortalaması diğer algoritmalarından daha iyidir. Fonksiyonların boyutları arttığında, algoritmaların performansı azalır, ancak pFOA_v1, karşılaştırılan algoritmalarından daha iyi performans göstermiştir.

Üçüncü olarak, pFOA_v2, 10 boyutlu fonksiyonlar için Çizelge 4.8'de, 30 boyutlu fonksiyonlar için Çizelge 4.9'da ve 50 boyutlu fonksiyonlar için Çizelge 4.10'da aynı algoritmalarla karşılaştırılmıştır.

Karşılaştırma tablolarına göre her üç boyutta pFOA_v2'nin sırası 1'dir, ortalama sırası ise hem pFOA_v1'den hem de karşılaştırılan algoritmalarından daha iyidir.

pFOA_v1 ve pFOA_v2, JAYA'dan 10 ve 30 boyutta 3 fonksiyon dışında, 50 boyutta ise 4 fonksiyon dışında daha iyi sonuçlar elde etmiştir.

Karşılaştırma tablolarından görülmektedir ki, pFOA_v1 ve pFOA_v2, güncelleme stratejilerindeki iyileştirme sayesinde optimum değerini bulunmasında daha etkili bir yaklaşımla FOA ve JAYA'dan daha iyi sonuçlar elde etmiştir.

Bu tür algoritmalarda bir başka önemli karşılaştırma metriği yakınsama özellikleridir. pFOA_v1 ve FOA'nın yakınsama karşılaştırması, Şekil 4.7'de 10 boyutlu, Şekil 4.8'de 30 boyutlu ve Şekil 4.9'da 50 boyutlu fonksiyonlar için verilmiştir.

Şekil 4.7, Şekil 4.8 ve Şekil 4.9'a dayanarak, neredeyse tüm durumlarda, pFOA_v1'in yakınsama özellikleri, optimum veya neredeyse optimum çözüme ulaşması açısından FOA'dan daha iyidir.

Çizelge 4.4. pFOA versiyonlarının FOA ile 10, 30 ve 50 boyutlar için karşılaştırmalı sonuçları.

Fonk.	D	pFOA_v1 (Önerilen)		pFOA_v2 (Önerilen)		FOA		D	pFOA_v1 (Önerilen)		pFOA_v2 (Önerilen)		FOA		D	pFOA_v1 (Önerilen)		pFOA_v2 (Önerilen)		FOA	
		Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.		Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.		Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
Sphere	10	1,46E-154	1,00E-153	2,32E-158	1,62E-157	3,21E-06	1,52E-06	30	3,41E-146	1,23E-145	4,53E-197	0,00E+00	3,30E-06	1,56E-06	50	6,63E-116	3,07E-115	1,97E-164	0,00E+00	3,33E-06	1,57E-06
Elliptic	10	4,89E-153	2,09E-152	2,09E-155	1,05E-154	1,18E-01	3,72E-02	30	9,73E-142	4,64E-141	2,97E-198	0,00E+00	8,36E-02	2,63E-02	50	1,82E-111	7,21E-111	3,50E-228	0,00E+00	7,81E-02	2,46E-02
SumSquares	10	1,79E-142	8,95E-142	6,89E-156	2,69E-155	1,74E-05	4,35E-06	30	8,95E-142	2,40E-144	2,84E-179	0,00E+00	4,80E-05	1,20E-05	50	2,18E-115	6,28E-115	1,56E-163	0,00E+00	7,87E-05	1,96E-05
SumPower	10	1,66E-273	0,00E+00	2,09E-271	0,00E+00	4,06E-08	8,66E-09	30	2,08E-227	0,00E+00	3,02E-257	0,00E+00	4,43E-09	9,43E-10	50	1,48E-161	7,25E-161	6,88E-236	0,00E+00	1,58E-09	3,37E-10
Schwefel2.22	10	2,53E-81	4,99E-81	7,59E-78	8,91E-78	5,87E-03	1,11E-03	30	5,46E-79	2,47E-78	1,39E-97	9,91E-97	1,02E-02	1,92E-03	50	4,15E-51	2,90E-50	7,93E-137	5,66E-136	1,31E-02	2,48E-03
Schwefel2.21	10	5,93E-46	4,15E-45	2,01E-47	3,48E-47	1,12E-03	1,92E-04	30	6,86E-05	2,12E-04	1,38E-09	3,16E-09	2,44E-03	4,18E-04	50	1,43E-02	1,86E-03	2,56E-07	3,94E-07	4,46E-03	7,66E-04
Step	10	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	30	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	50	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
Quartic	10	1,29E-234	0,00E+00	2,38E-265	0,00E+00	5,93E-12	8,75E-13	30	4,61E-184	0,00E+00	5,80E-258	0,00E+00	5,35E-12	7,89E-13	50	3,28E-128	2,12E-127	7,75E-180	0,00E+00	5,23E-12	7,71E-13
QuarticWN	10	9,01E-04	4,80E-04	8,08E-04	4,53E-04	2,77E-03	3,93E-04	30	1,15E-01	7,50E-01	3,86E-03	1,45E-03	5,78E-03	8,09E-04	50	1,96E-01	1,13E+00	5,05E-03	3,22E-03	7,82E-03	1,09E-03
Rosenbrock	10	1,60E-03	1,98E-03	5,69E+00	2,92E-01	6,99E+00	9,19E-01	30	9,34E-09	4,47E-08	2,67E+01	2,95E-01	2,75E+01	3,61E+00	50	5,79E-01	9,82E-01	4,69E+01	1,41E+01	4,80E+01	6,31E+00
Rastrigin	10	7,22E-01	4,48E-01	6,15E-01	6,45E-01	5,58E+00	1,04E+00	30	4,76E+00	4,30E+00	2,46E+00	2,50E+00	4,16E+01	6,21E+00	50	8,23E+00	4,02E+00	4,34E+00	3,61E+00	1,04E+02	1,42E+01
Non-Cont. Rastrigin	10	9,80E-01	7,07E-01	3,49E-01	2,61E-01	1,38E+00	2,09E-01	30	6,56E+00	8,64E+00	1,30E+01	5,37E+00	1,49E+01	3,32E+00	50	1,31E+01	1,01E+01	2,40E+01	9,76E+00	7,10E+01	1,07E+01
Griewank	10	0,00E+00	0,00E+00	0,00E+00	0,00E+00	2,85E-07	3,28E-08	30	8,49E-17	1,61E-16	0,00E+00	0,00E+00	1,38E-07	1,58E-08	50	8,51E-16	7,04E-16	0,00E+00	0,00E+00	9,30E-08	1,07E-08
Schwefel2.26	10	4,15E+03	5,43E-12	3,96E+03	1,03E+03	4,00E+03	4,41E+02	30	1,25E+04	3,21E+00	8,96E+03	2,34E+03	1,22E+04	1,35E+03	50	1,98E+04	3,24E+03	1,55E+04	4,00E+03	2,04E+04	2,25E+03
Ackley	10	3,69E-15	6,96E-16	1,67E-15	6,20E-16	2,35E-03	2,50E-04	30	2,31E-14	1,16E-14	3,62E-15	9,13E-16	1,35E-03	1,44E-04	50	1,90E-13	1,86E-13	4,67E-15	1,24E-15	1,05E-03	1,12E-04
Penalized1	10	6,88E-02	1,95E-02	9,15E-03	2,39E-03	2,65E+00	2,74E-01	30	4,45E-01	1,88E-01	7,56E-02	1,88E-02	1,67E+00	1,72E-01	50	6,26E-01	1,05E-01	1,52E-01	2,29E-02	1,47E+00	1,52E-01
Penalized2	10	9,75E-01	2,56E-02	9,67E-01	2,29E-01	9,54E-01	9,57E-02	30	2,82E+00	4,99E-01	2,81E+00	6,64E-01	2,90E+00	2,90E-01	50	4,45E+00	1,18E+00	4,73E+00	1,88E-01	4,85E+00	4,85E-01
Alpine	10	4,31E-05	1,55E-04	1,26E-03	5,10E-04	3,00E-03	1,69E-03	30	5,53E-31	1,65E-30	3,43E-04	1,98E-04	1,62E-02	7,50E-03	50	1,23E-52	5,41E-52	2,25E-05	7,15E-05	1,19E-02	7,37E-03
Levy	10	3,37E+00	2,81E+00	1,00E+00	2,68E-01	5,98E+00	5,82E-01	30	7,23E-01	6,77E-01	8,43E+00	1,92E+00	2,12E+01	2,02E+00	50	9,50E-01	6,87E-01	2,40E+01	5,39E+00	3,77E+01	3,57E+00
Weierstrass	10	3,99E+01	5,60E-01	0,00E+00	0,00E+00	1,02E+01	9,38E-01	30	1,20E+02	4,90E-01	0,00E+00	0,00E+00	4,43E+01	4,09E+00	50	2,00E+02	1,83E-01	0,00E+00	0,00E+00	8,02E+01	7,38E+00
Schaffer	10	0,00E+00	0,00E+00	0,00E+00	0,00E+00	3,21E-06	2,89E-07	30	1,15E-16	4,82E-17	0,00E+00	0,00E+00	3,30E-06	2,97E-07	50	3,13E-16	1,56E-16	1,20E-17	5,60E-18	1,94E-04	1,23E-04

Çizelge 4.5. pFOA_v1'in 10 boyut için diğer algoritmalarla karşılaştırmalı sonuçları

Fonk.	D	pFOA_v1 (Önerilen)			SFOA (Önerilen)			FOA			SPSO2011			FA			TSA			CS			JAYA																
		Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R														
Sphere	10	1,46E-154	1,00E-153	2	5,05E-07	2,54E-07	7	3,21E-06	1,52E-06	8	2,50E-280	0,00E+00	1	7,01E-88	2,70E-88	6	2,89E-109	8,34E-109	4	1,09E-90	2,49E-90	5	1,99E-151	7,85E-151	3														
Elliptic	10	4,89E-153	2,09E-152	1	4,19E-02	1,39E-02	6	1,18E-01	3,72E-02	7	9,98E+03	3,66E+03	8	1,89E-83	5,81E-84	5	1,24E-106	3,23E-106	3	5,81E-88	4,72E-88	4	5,10E-148	6,63E-148	2														
SumSquares	10	1,79E-142	8,95E-142	3	2,75E-06	7,09E-07	7	1,74E-05	4,35E-06	8	3,21E-188	0,00E+00	1	3,43E-89	8,16E-90	6	1,15E-110	2,59E-110	4	5,59E-92	5,85E-92	5	6,44E-152	5,21E-152	2														
SumPower	10	1,66E-273	0,00E+00	1	2,68E-08	5,85E-09	7	4,06E-08	8,66E-09	8	5,22E-09	1,17E-09	6	6,31E-23	9,19E-23	5	4,07E-169	0,00E+00	3	8,20E-140	1,25E-139	4	8,63E-267	0,00E+00	2														
Schwefel2.22	10	2,53E-81	4,99E-81	1	2,28E-03	4,38E-04	7	5,87E-03	1,11E-03	8	1,93E-71	2,55E-71	3	6,61E-45	1,20E-45	5	3,40E-69	2,35E-69	4	7,53E-42	2,15E-42	6	3,50E-79	2,69E-79	2														
Schwefel2.21	10	5,93E-46	4,15E-45	2	2,44E-04	4,25E-05	7	1,12E-03	1,92E-04	8	1,27E-116	1,45E-116	1	1,51E-44	2,49E-45	4	9,54E-20	8,23E-20	6	3,17E-26	1,60E-26	5	6,28E-45	3,77E-45	3														
Step	10	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1	1,57E-01	6,82E-02	8	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1														
Quartic	10	1,29E-234	0,00E+00	3	1,40E-13	2,08E-14	7	5,93E-12	8,75E-13	8	0,00E+00	0,00E+00	1	1,63E-182	0,00E+00	4	1,66E-161	2,41E-161	6	8,92E-170	0,00E+00	5	4,46E-249	0,00E+00	2														
QuarticWN	10	9,01E-04	4,80E-04	4	2,52E-04	3,67E-05	2	2,77E-03	3,93E-04	8	2,55E-04	3,94E-05	3	1,31E-04	2,13E-05	1	1,03E-03	3,49E-04	6	1,01E-03	1,55E-04	5	1,08E-03	2,18E-04	7														
Rosenbrock	10	1,60E-03	1,98E-03	3	8,90E+00	1,18E+00	8	6,99E+00	9,19E-01	7	2,56E-01	1,27E-01	4	1,05E+00	2,55E-01	5	1,83E+00	5,96E-01	6	2,55E-19	2,39E-19	1	1,10E-03	1,05E-03	2														
Rastrigin	10	7,22E-01	4,48E-01	3	1,03E-04	1,30E-05	2	5,58E+00	1,04E+00	6	8,92E+00	1,40E+00	7	1,20E+01	1,64E+00	8	1,81E+00	7,33E-01	4	0,00E+00	0,00E+00	1	3,81E+00	7,77E-01	5														
N-C. Rastrigin	10	9,80E-01	7,07E-01	2	6,55E+00	1,89E+00	6	1,38E+00	2,09E-01	3	7,10E+00	8,59E-01	7	1,20E+01	1,56E+00	8	5,97E+00	1,71E+00	5	2,94E-01	7,28E-02	1	5,77E+00	9,95E-01	4														
Griewank	10	0,00E+00	0,00E+00	1	5,99E-08	6,92E-09	3	2,85E-07	3,28E-08	4	4,16E-02	5,46E-03	6	6,39E-02	8,14E-03	8	4,60E-02	1,81E-02	7	2,82E-03	5,94E-04	5	0,00E+00	0,00E+00	1														
Schwefel2.26	10	4,15E+03	5,43E-12	6	4,19E+03	4,66E+02	8	4,00E+03	4,41E+02	5	1,04E+03	1,17E+02	4	9,06E+02	1,02E+02	3	1,16E+01	9,88E+00	2	1,62E-11	1,28E-11	1	4,15E+03	6,33E+02	6														
Ackley	10	3,69E-15	6,96E-16	3	9,10E-04	9,76E-05	6	2,35E-03	2,50E-04	7	2,26E-02	1,71E-02	8	8,43E-15	9,72E-16	5	3,55E-15	8,87E-16	2	3,34E-15	3,65E-16	1	3,76E-15	5,68E-16	4														
Penalized1	10	6,88E-02	1,95E-02	6	2,48E+00	2,57E-01	7	2,65E+00	2,74E-01	8	6,10E-03	4,47E-03	3	1,83E-02	7,70E-03	4	4,71E-32	1,14E-32	1	4,71E-32	4,83E-33	1	2,07E-02	3,11E-03	5														
Penalized2	10	9,75E-01	2,56E-02	7	9,89E-01	9,94E-02	8	9,54E-01	9,57E-02	6	6,46E-04	2,65E-04	5	1,35E-32	1,33E-33	1	1,35E-32	3,18E-33	1	1,35E-32	1,34E-33	1	2,15E-04	2,15E-04	4														
Alpine	10	4,31E-05	1,55E-04	3	1,33E-02	1,50E-03	7	3,00E-03	1,69E-03	5	5,49E-03	1,67E-03	6	5,13E-16	8,57E-17	1	7,93E-13	1,33E-12	2	2,36E-02	3,59E-03	8	5,29E-05	4,51E-05	4														
Levy	10	3,37E+00	2,81E+00	6	1,00E+01	9,53E-01	8	5,98E+00	5,82E-01	7	3,80E-02	1,99E-02	5	2,15E-03	1,44E-03	3	1,35E-31	3,02E-32	1	1,35E-31	1,27E-32	1	2,15E-03	2,04E-03	3														
Weierstrass	10	3,99E+01	5,60E-01	8	1,39E+01	1,29E+00	7	1,02E+01	9,38E-01	5	3,48E-01	5,22E-02	4	4,88E-16	1,36E-16	3	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1	1,04E+01	1,61E+00	6														
Schaffer	10	0,00E+00	0,00E+00	1	5,05E-07	4,56E-08	2	3,21E-06	2,89E-07	3	1,03E-02	9,75E-04	7	1,03E-02	9,71E-04	8	9,72E-03	2,07E-03	5	9,72E-03	8,69E-04	6	9,72E-03	1,21E-03	4														
Mean Rank				3,19					5,86					6,19					4,33					4,81					3,52					3,24					3,43
Final Rank				1					7					8					5					6					4					2					3

Çizelge 4.6. pFOA_v1'in 30 boyut için diğer algoritmalarla karşılaştırmalı sonuçları

Fonk.	D	pFOA_v1 (Önerilen)			SFOA (Önerilen)			FOA			SPSO2011			FA			TSA			CS			JAYA		
		Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R
Sphere	30	3,41E-146	1,23E-145	3	2,03E-07	1,02E-07	7	3,30E-06	1,56E-06	8	1,29E-303	0,00E+00	1	4,99E-262	0,00E+00	2	1,65E-56	2,72E-56	6	3,79E-94	1,12E-93	5	9,97E-146	2,13E-145	4
Elliptic	30	9,73E-142	4,64E-141	2	1,31E-02	4,34E-03	6	8,36E-02	2,63E-02	7	3,96E+04	1,28E+04	8	8,13E-257	0,00E+00	1	2,99E-54	3,64E-54	5	8,64E-92	6,38E-92	4	5,83E-141	1,49E-140	3
SumSquares	30	8,95E-142	2,40E-144	3	3,09E-06	7,96E-07	7	4,80E-05	1,20E-05	8	4,30E-41	2,04E-41	6	6,92E-263	0,00E+00	1	1,81E-57	2,00E-57	5	3,09E-96	1,89E-96	4	2,56E-144	3,51E-144	2
SumPower	30	2,08E-227	0,00E+00	1	2,99E-09	6,51E-10	5	4,43E-09	9,43E-10	6	1,02E-08	2,16E-09	7	1,12E-16	1,02E-16	4	4,70E-42	1,20E-41	3	1,00E+10	2,08E+09	8	1,69E-223	0,00E+00	2
Schwefel2.22	30	5,46E-79	2,47E-78	3	2,48E-03	4,77E-04	6	1,02E-02	1,92E-03	7	3,29E+00	6,95E-01	8	9,38E-132	1,69E-132	1	1,14E-42	6,51E-43	5	3,02E-52	8,49E-53	4	1,08E-81	1,15E-81	2
Schwefel2.21	30	6,86E-05	2,12E-04	1	8,92E-05	1,55E-05	2	2,44E-03	4,18E-04	4	7,25E+00	1,31E+00	6	1,25E+01	2,18E+00	8	6,28E-01	2,51E-01	5	8,78E+00	1,61E+00	7	1,81E-04	6,62E-05	3
Step	30	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1	4,84E+00	9,77E-01	8	2,94E-01	9,88E-02	7	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1	1,37E-01	8,06E-02	6
Quartic	30	4,61E-184	0,00E+00	3	2,07E-14	3,09E-15	7	5,35E-12	7,89E-13	8	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1	4,60E-55	8,18E-55	6	1,91E-107	1,29E-107	5	4,15E-183	0,00E+00	4
QuarticWN	30	1,15E-01	7,50E-01	8	2,35E-04	3,36E-05	1	5,78E-03	8,09E-04	4	1,53E-03	2,31E-04	2	2,09E-03	3,77E-04	3	8,42E-03	2,75E-03	5	1,17E-02	1,78E-03	7	9,21E-03	2,28E-03	6
Rosenbrock	30	9,34E-09	4,47E-08	2	2,87E+01	3,81E+00	8	2,75E+01	3,61E+00	7	1,84E+01	2,95E+00	5	1,76E+01	3,47E+00	4	2,15E+01	6,88E+00	6	6,25E-01	2,07E-01	3	1,17E-09	8,47E-10	1
Rastrigin	30	4,76E+00	4,30E+00	3	4,08E-05	5,14E-06	1	4,16E+01	6,21E+00	7	3,35E+01	4,29E+00	6	7,54E+01	9,61E+00	8	2,68E+01	1,09E+01	5	4,19E+00	6,48E-01	2	2,03E+01	3,61E+00	4
N-C. Rastrigin	30	6,56E+00	8,64E+00	3	4,08E-05	4,91E-06	1	1,49E+01	3,32E+00	4	4,59E+01	5,73E+00	6	1,04E+02	1,26E+01	8	6,03E+01	1,85E+01	7	5,06E+00	6,47E-01	2	2,94E+01	5,46E+00	5
Griewank	30	8,49E-17	1,61E-16	2	1,16E-08	1,34E-09	3	1,38E-07	1,58E-08	4	9,75E-03	1,56E-03	8	4,83E-03	9,02E-04	7	1,45E-04	2,87E-04	5	2,13E-03	6,74E-04	6	4,79E-17	1,61E-17	1
Schwefel2.26	30	1,25E+04	3,21E+00	7	1,26E+04	1,40E+03	8	1,22E+04	1,35E+03	6	4,89E+03	5,41E+02	4	4,16E+03	4,58E+02	3	1,30E+03	3,48E+02	2	3,13E+02	4,27E+01	1	1,05E+04	1,72E+03	5
Ackley	30	2,31E-14	1,16E-14	2	3,31E-04	3,55E-05	4	1,35E-03	1,44E-04	5	2,31E+00	2,53E-01	8	2,48E-14	2,78E-15	3	6,41E-15	1,64E-15	1	7,41E-01	1,07E-01	7	2,22E-01	7,60E-02	6
Penalized1	30	4,45E-01	1,88E-01	4	1,56E+00	1,61E-01	7	1,67E+00	1,72E-01	8	9,54E-01	2,06E-01	6	4,74E-01	9,65E-02	5	1,57E-32	3,80E-33	1	2,03E-03	1,50E-03	2	2,71E-01	3,99E-02	3
Penalized2	30	2,82E+00	4,99E-01	6	2,97E+00	2,99E-01	8	2,90E+00	2,90E-01	7	1,79E+00	9,23E-01	5	1,29E-03	3,73E-04	2	2,15E-04	3,73E-04	1	1,03E-01	5,51E-02	4	2,98E-03	8,95E-04	3
Alpine	30	5,53E-31	1,65E-30	1	2,03E-02	3,76E-03	6	1,62E-02	7,50E-03	5	7,53E-01	1,11E-01	8	8,67E-15	9,23E-16	3	2,09E-05	3,50E-05	4	2,57E-01	3,91E-02	7	1,20E-17	1,16E-17	2
Levy	30	7,23E-01	6,77E-01	5	2,97E+01	2,82E+00	8	2,12E+01	2,02E+00	7	9,30E-01	1,49E-01	6	5,45E-01	2,06E-01	4	1,35E-31	3,02E-32	1	2,15E-03	1,45E-03	2	4,31E-02	1,35E-02	3
Weierstrass	30	1,20E+02	4,90E-01	8	4,70E+01	4,35E+00	6	4,43E+01	4,09E+00	5	9,80E+00	9,10E-01	4	5,22E-01	9,33E-02	3	0,00E+00	0,00E+00	1	4,83E-01	7,01E-02	2	5,15E+01	6,60E+00	7
Schaffer	30	1,15E-16	4,82E-17	1	2,03E-07	1,83E-08	2	3,30E-06	2,97E-07	3	4,99E-02	5,28E-03	6	1,36E-01	1,28E-02	7	3,67E-02	7,86E-03	5	1,56E-01	1,48E-02	8	9,72E-03	1,21E-03	4
		Mean Rank		3,29			4,95			5,76			5,67			4,05			3,81			4,33			3,62
		Final Rank		1			6			8			7			4			3			5			2

Çizelge 4.7. pFOA_v1'in 50 boyut için diğer algoritmalarla karşılaştırmalı sonuçları

Fonk.	D	pFOA_v1 (Önerilen)			SFOA (Önerilen)			FOA			SPSO2011			FA			TSA			CS			JAYA		
		Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R
Sphere	50	6,63E-116	3,07E-115	4	1,29E-07	6,51E-08	7	3,33E-06	1,57E-06	8	7,92E-295	0,00E+00	2	0,00E+00	0,00E+00	1	9,06E-25	8,22E-25	6	1,55E-90	2,25E-90	5	4,79E-116	1,36E-115	3
Elliptic	50	1,82E-111	7,21E-111	1	8,15E-03	2,70E-03	6	7,81E-02	2,46E-02	7	8,55E+04	2,75E+04	8	1,41E-28	2,90E-28	4	9,40E-23	9,03E-23	5	5,60E-87	5,90E-87	3	3,62E-110	6,13E-110	2
SumSquares	50	2,18E-115	6,28E-115	2	3,24E-06	8,35E-07	7	7,87E-05	1,96E-05	8	2,69E-21	1,16E-21	6	0,00E+00	0,00E+00	1	2,34E-25	2,35E-25	5	1,07E-90	1,00E-90	4	4,15E-114	3,92E-114	3
SumPower	50	1,48E-161	7,25E-161	1	1,08E-09	2,34E-10	3	1,58E-09	3,37E-10	4	7,81E-09	1,62E-09	5	1,84E+03	2,63E+03	7	4,56E-03	1,07E-02	6	1,00E+10	2,08E+09	8	4,38E-152	6,15E-152	2
Schwefel2.22	50	4,15E-51	2,90E-50	2	2,55E-03	4,91E-04	6	1,31E-02	2,48E-03	7	1,02E+01	1,98E+00	8	4,68E-219	0,00E+00	1	1,13E-22	5,95E-23	4	5,94E-16	8,01E-16	5	3,43E-50	6,30E-50	3
Schwefel2.21	50	1,43E-02	1,86E-03	3	5,47E-05	9,52E-06	1	4,46E-03	7,66E-04	2	1,66E+01	2,80E+00	5	4,41E+01	7,35E+00	8	2,33E+01	8,76E+00	7	2,15E+01	3,69E+00	6	5,25E-01	1,21E-01	4
Step	50	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1	2,85E+01	4,82E+00	8	2,16E+00	4,35E-01	7	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1	1,41E+00	3,81E-01	6
Quartic	50	3,28E-128	2,12E-127	4	8,28E-15	1,23E-15	7	5,23E-12	7,71E-13	8	4,90E+324	0,00E+00	2	0,00E+00	0,00E+00	1	6,83E-18	4,92E-18	6	4,37E-89	2,58E-89	5	4,69E-130	6,35E-130	3
QuarticWN	50	1,96E-01	1,13E+00	8	2,30E-04	3,25E-05	1	7,82E-03	1,09E-03	3	3,37E-03	4,78E-04	2	9,92E-03	1,54E-03	4	4,26E-02	1,38E-02	6	4,63E-02	6,68E-03	7	3,05E-02	6,54E-03	5
Rosenbrock	50	5,79E-01	9,82E-01	2	4,86E+01	6,43E+00	8	4,80E+01	6,31E+00	6	3,52E+01	4,64E+00	5	3,31E+01	5,69E+00	4	4,81E+01	1,57E+01	7	2,88E+00	6,24E-01	3	1,01E-01	5,21E-02	1
Rastrigin	50	8,23E+00	4,02E+00	2	2,59E-05	3,26E-06	1	1,04E+02	1,42E+01	7	6,19E+01	7,81E+00	6	1,68E+02	2,10E+01	8	4,38E+01	1,35E+01	5	1,65E+01	2,22E+00	3	3,48E+01	6,11E+00	4
N-C. Rastrigin	50	1,31E+01	1,01E+01	3	2,59E-05	3,11E-06	1	7,10E+01	1,07E+01	5	1,03E+02	1,30E+01	7	2,04E+02	2,44E+01	8	9,65E+01	2,94E+01	6	1,19E+01	1,48E+00	2	5,29E+01	9,99E+00	4
Griewank	50	8,51E-16	7,04E-16	2	5,03E-09	5,81E-10	4	9,30E-08	1,07E-08	5	6,91E-03	1,29E-03	8	3,14E-03	7,26E-04	6	0,00E+00	0,00E+00	1	3,38E-03	9,31E-04	7	1,08E-15	3,41E-16	3
Schwefel2.26	50	1,98E+04	3,24E+03	6	2,09E+04	2,33E+03	8	2,04E+04	2,25E+03	7	9,00E+03	9,89E+02	5	7,67E+03	8,37E+02	4	3,77E+03	9,88E+02	2	1,20E+03	1,42E+02	1	4,74E+03	9,15E+02	3
Ackley	50	1,90E-13	1,86E-13	1	2,04E-04	2,19E-05	3	1,05E-03	1,12E-04	4	3,37E+00	3,63E-01	8	6,91E-02	3,68E-02	5	1,93E-13	5,45E-14	2	2,27E+00	2,51E-01	7	7,10E-01	1,38E-01	6
Penalized1	50	6,26E-01	1,05E-01	3	1,38E+00	1,43E-01	5	1,47E+00	1,52E-01	6	5,45E+00	7,83E-01	8	1,84E+00	2,52E-01	7	1,54E-01	2,52E-01	2	3,78E-02	9,88E-03	1	6,35E-01	9,26E-02	4
Penalized2	50	4,45E+00	1,18E+00	5	4,95E+00	4,98E-01	7	4,85E+00	4,85E-01	6	6,79E+01	7,58E+00	8	2,25E-01	7,28E-02	3	7,84E-08	5,46E-08	1	3,45E-01	1,06E-01	4	9,20E-03	2,16E-03	2
Alpine	50	1,23E-52	5,41E-52	1	2,58E-04	2,52E-05	4	1,19E-02	7,37E-03	6	6,01E+00	6,55E-01	8	4,27E-05	2,93E-05	3	1,31E-03	2,16E-03	5	1,47E-01	4,36E-02	7	2,83E-17	1,97E-17	2
Levy	50	9,50E-01	6,87E-01	4	4,95E+01	4,70E+00	8	3,77E+01	3,57E+00	7	1,96E+00	2,45E-01	5	2,34E+00	4,22E-01	6	2,90E-26	1,25E-26	1	1,17E-02	6,60E-03	2	1,50E-01	3,72E-02	3
Weierstrass	50	2,00E+02	1,83E-01	8	8,13E+01	7,52E+00	6	8,02E+01	7,38E+00	5	2,44E+01	2,24E+00	4	4,18E+00	4,63E-01	3	5,57E-16	6,29E-16	1	2,75E+00	2,89E-01	2	8,90E+01	1,14E+01	7
Schaffer	50	3,13E-16	1,56E-16	1	1,30E-07	1,17E-08	2	1,94E-04	1,23E-04	3	1,07E-01	9,94E-03	6	3,90E-01	3,50E-02	8	8,02E-02	1,72E-02	5	3,27E-01	2,98E-02	7	1,08E-02	1,51E-03	4
		Mean Rank		3,05				4,57		5,48		5,90		4,71		4,00		4,29				3,52			
		Final Rank		1				5		7		8		6		3		4				2			

Çizelge 4.8. pFOA_v2'nin 10 boyut için diğer algoritmalarla karşılaştırmalı sonuçları

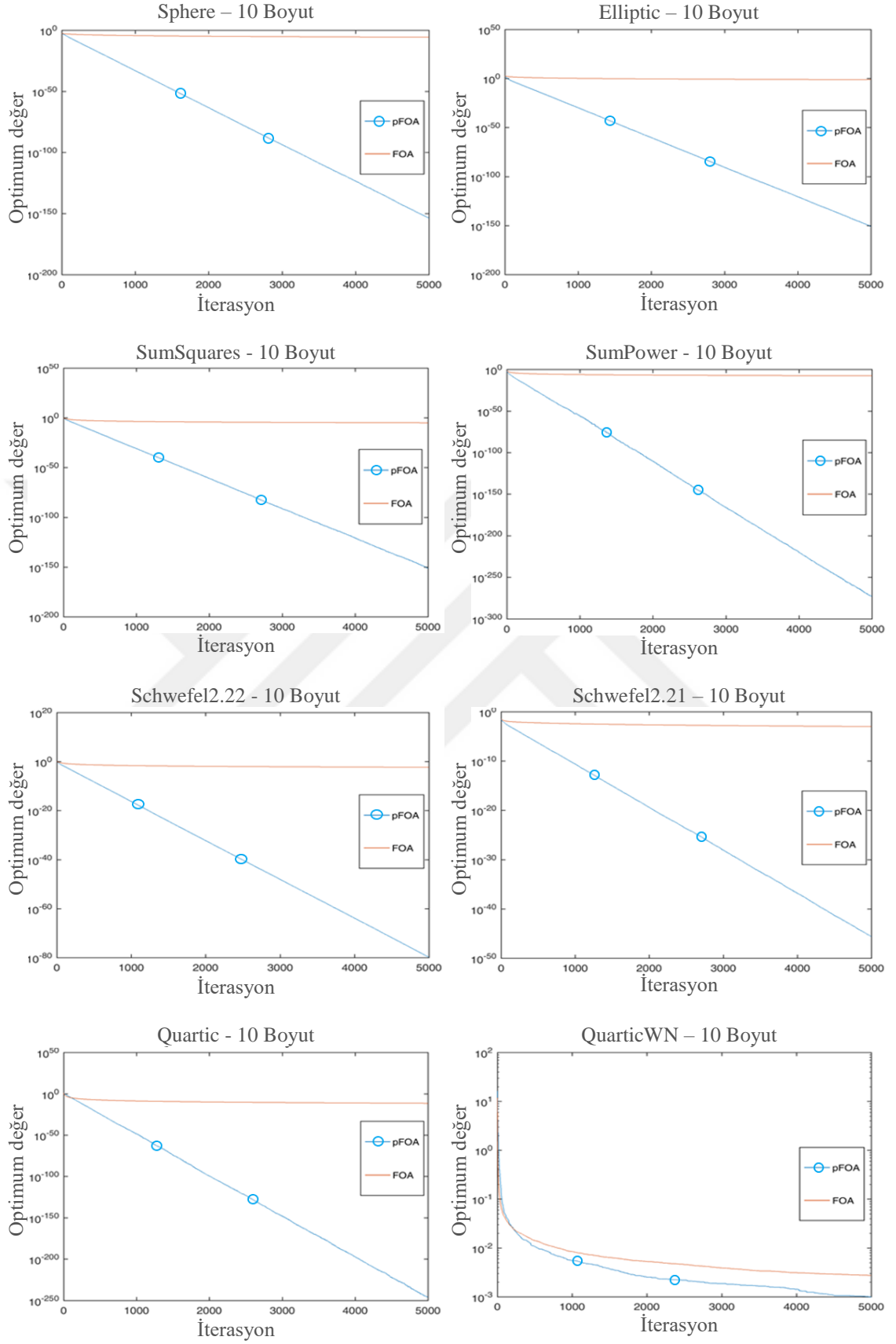
Fonk.	D	pFOA_v2 (Önerilen)			SFOA (Önerilen)			FOA			SPSO2011			FA			TSA			CS			JAYA		
		Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R
Sphere	10	2,32E-158	1,62E-157	2	5,05E-07	2,54E-07	7	3,21E-06	1,52E-06	8	2,50E-280	0,00E+00	1	7,01E-88	2,70E-88	6	2,89E-109	8,34E-109	4	1,09E-90	2,49E-90	5	1,99E-151	7,85E-151	3
Elliptic	10	2,09E-155	1,05E-154	1	4,19E-02	1,39E-02	6	1,18E-01	3,72E-02	7	9,98E+03	3,66E+03	8	1,89E-83	5,81E-84	5	1,24E-106	3,23E-106	3	5,81E-88	4,72E-88	4	5,10E-148	6,63E-148	2
SumSquares	10	6,89E-156	2,69E-155	2	2,75E-06	7,09E-07	7	1,74E-05	4,35E-06	8	3,21E-188	0,00E+00	1	3,43E-89	8,16E-90	6	1,15E-110	2,59E-110	4	5,59E-92	5,85E-92	5	6,44E-152	5,21E-152	3
SumPower	10	2,09E-271	0,00E+00	1	2,68E-08	5,85E-09	7	4,06E-08	8,66E-09	8	5,22E-09	1,17E-09	6	6,31E-23	9,19E-23	5	4,07E-169	0,00E+00	3	8,20E-140	1,25E-139	4	8,63E-267	0,00E+00	2
Schwefel2.22	10	7,59E-78	8,91E-78	2	2,28E-03	4,38E-04	7	5,87E-03	1,11E-03	8	1,93E-71	2,55E-71	3	6,61E-45	1,20E-45	5	3,40E-69	2,35E-69	4	7,53E-42	2,15E-42	6	3,50E-79	2,69E-79	1
Schwefel2.21	10	2,01E-47	3,48E-47	2	2,44E-04	4,25E-05	7	1,12E-03	1,92E-04	8	1,27E-116	1,45E-116	1	1,51E-44	2,49E-45	4	9,54E-20	8,23E-20	6	3,17E-26	1,60E-26	5	6,28E-45	3,77E-45	3
Step	10	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1	1,57E-01	6,82E-02	8	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1
Quartic	10	2,38E-265	0,00E+00	2	1,40E-13	2,08E-14	7	5,93E-12	8,75E-13	8	0,00E+00	0,00E+00	1	1,63E-182	0,00E+00	4	1,66E-161	2,41E-161	6	8,92E-170	0,00E+00	5	4,46E-249	0,00E+00	3
QuarticWN	10	8,08E-04	4,53E-04	4	2,52E-04	3,67E-05	2	2,77E-03	3,93E-04	8	2,55E-04	3,94E-05	3	1,31E-04	2,13E-05	1	1,03E-03	3,49E-04	6	1,01E-03	1,55E-04	5	1,08E-03	2,18E-04	7
Rosenbrock	10	5,69E+00	2,92E-01	6	8,90E+00	1,18E+00	8	6,99E+00	9,19E-01	7	2,56E-01	1,27E-01	3	1,05E+00	2,55E-01	4	1,83E+00	5,96E-01	5	2,55E-19	2,39E-19	1	1,10E-03	1,05E-03	2
Rastrigin	10	6,15E-01	6,45E-01	3	1,03E-04	1,30E-05	2	5,58E+00	1,04E+00	6	8,92E+00	1,40E+00	7	1,20E+01	1,64E+00	8	1,81E+00	7,33E-01	4	0,00E+00	0,00E+00	1	3,81E+00	7,77E-01	5
N-C. Rastrigin	10	3,49E-01	2,61E-01	2	6,55E+00	1,89E+00	6	1,38E+00	2,09E-01	3	7,10E+00	8,59E-01	7	1,20E+01	1,56E+00	8	5,97E+00	1,71E+00	5	2,94E-01	7,28E-02	1	5,77E+00	9,95E-01	4
Griewank	10	0,00E+00	0,00E+00	1	5,99E-08	6,92E-09	3	2,85E-07	3,28E-08	4	4,16E-02	5,46E-03	6	6,39E-02	8,14E-03	8	4,60E-02	1,81E-02	7	2,82E-03	5,94E-04	5	0,00E+00	0,00E+00	1
Schwefel2.26	10	3,96E+03	1,03E+03	5	4,19E+03	4,66E+02	8	4,00E+03	4,41E+02	6	1,04E+03	1,17E+02	4	9,06E+02	1,02E+02	3	1,16E+01	9,88E+00	2	1,62E-11	1,28E-11	1	4,15E+03	6,33E+02	7
Ackley	10	1,67E-15	6,20E-16	1	9,10E-04	9,76E-05	6	2,35E-03	2,50E-04	7	2,26E-02	1,71E-02	8	8,43E-15	9,72E-16	5	3,55E-15	8,87E-16	3	3,34E-15	3,65E-16	2	3,76E-15	5,68E-16	4
Penalized1	10	9,15E-03	2,39E-03	4	2,48E+00	2,57E-01	7	2,65E+00	2,74E-01	8	6,10E-03	4,47E-03	3	1,83E-02	7,70E-03	5	4,71E-32	1,14E-32	1	4,71E-32	4,83E-33	1	2,07E-02	3,11E-03	6
Penalized2	10	9,67E-01	2,29E-01	7	9,89E-01	9,94E-02	8	9,54E-01	9,57E-02	6	6,46E-04	2,65E-04	5	1,35E-32	1,33E-33	1	1,35E-32	3,18E-33	1	1,35E-32	1,34E-33	1	2,15E-04	2,15E-04	4
Alpine	10	1,26E-03	5,10E-04	4	1,33E-02	1,50E-03	7	3,00E-03	1,69E-03	5	5,49E-03	1,67E-03	6	5,13E-16	8,57E-17	1	7,93E-13	1,33E-12	2	2,36E-02	3,59E-03	8	5,29E-05	4,51E-05	3
Levy	10	1,00E+00	2,68E-01	6	1,00E+01	9,53E-01	8	5,98E+00	5,82E-01	7	3,80E-02	1,99E-02	5	2,15E-03	1,44E-03	3	1,35E-31	3,02E-32	1	1,35E-31	1,27E-32	1	2,15E-03	2,04E-03	3
Weierstrass	10	0,00E+00	0,00E+00	1	1,39E+01	1,29E+00	8	1,02E+01	9,38E-01	6	3,48E-01	5,22E-02	5	4,88E-16	1,36E-16	4	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1	1,04E+01	1,61E+00	7
Schaffer	10	0,00E+00	0,00E+00	1	5,05E-07	4,56E-08	2	3,21E-06	2,89E-07	3	1,03E-02	9,75E-04	7	1,03E-02	9,71E-04	8	9,72E-03	2,07E-03	5	9,72E-03	8,69E-04	6	9,72E-03	1,21E-03	4
Mean Rank				2,76			5,90			6,29			4,33			4,86			3,52			3,29			3,57
Final Rank				1			7			8			5			6			3			2			4

Çizelge 4.9. pFOA_v2'nin 30 boyut için diğer algoritmalarla karşılaştırmalı sonuçları

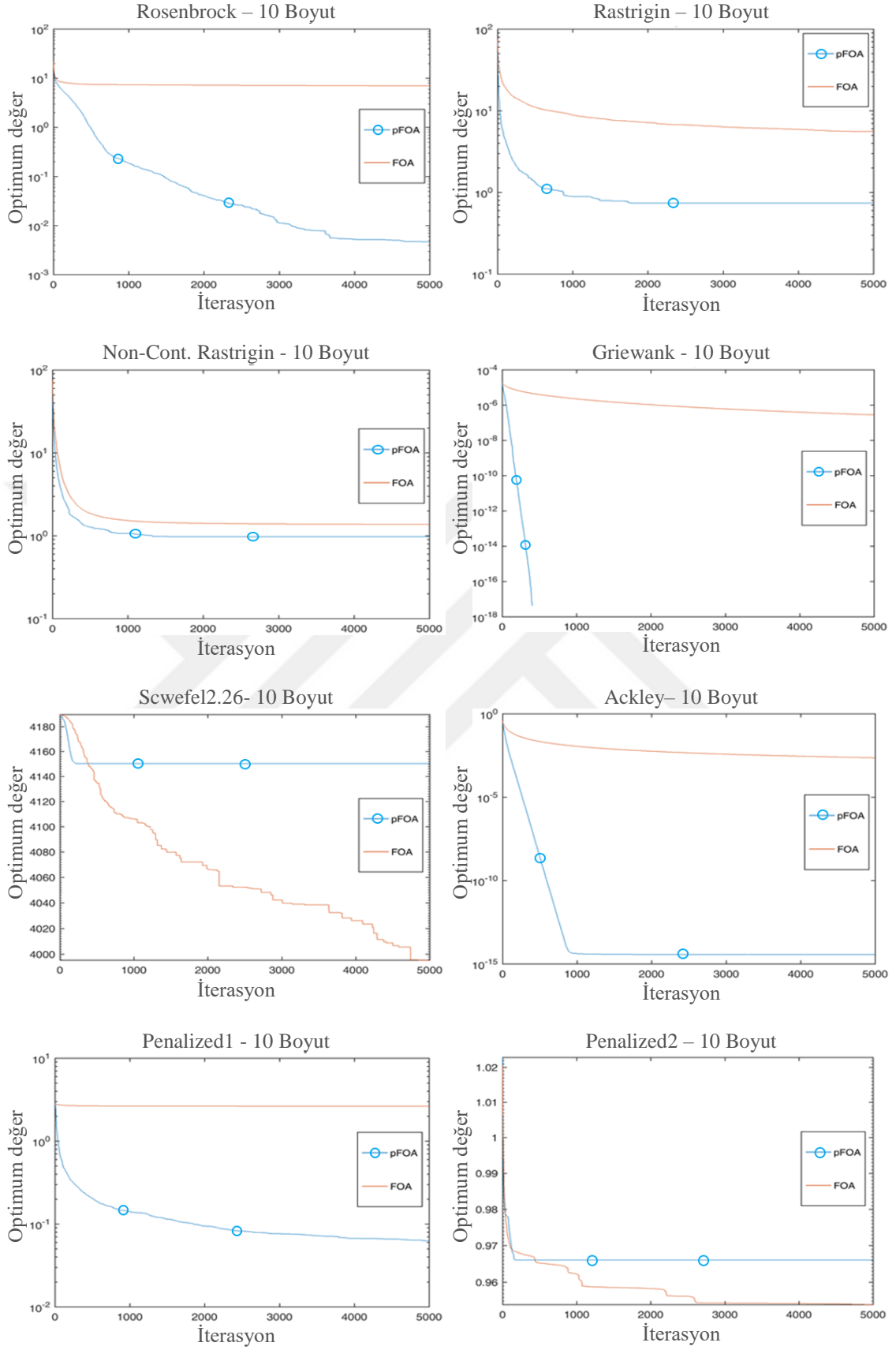
Fonk.	D	pFOA_v2 (Önerilen)			SFOA (Önerilen)			FOA			SPSO2011			FA			TSA			CS			JAYA			
		Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	
Sphere	30	4,53E-197	0,00E+00	3	2,03E-07	1,02E-07	7	3,30E-06	1,56E-06	8	1,29E-303	0,00E+00	1	4,99E-262	0,00E+00	2	1,65E-56	2,72E-56	6	3,79E-94	1,12E-93	5	9,97E-146	2,13E-145	4	
Elliptic	30	2,97E-198	0,00E+00	2	1,31E-02	4,34E-03	6	8,36E-02	2,63E-02	7	3,96E+04	1,28E+04	8	8,13E-257	0,00E+00	1	2,99E-54	3,64E-54	5	8,64E-92	6,38E-92	4	5,83E-141	1,49E-140	3	
SumSquares	30	2,84E-179	0,00E+00	2	3,09E-06	7,96E-07	7	4,80E-05	1,20E-05	8	4,30E-41	2,04E-41	6	6,92E-263	0,00E+00	1	1,81E-57	2,00E-57	5	3,09E-96	1,89E-96	4	2,56E-144	3,51E-144	3	
SumPower	30	3,02E-257	0,00E+00	1	2,99E-09	6,51E-10	5	4,43E-09	9,43E-10	6	1,02E-08	2,16E-09	7	1,12E-16	1,02E-16	4	4,70E-42	1,20E-41	3	1,00E+10	2,08E+09	8	1,69E-223	0,00E+00	2	
Schwefel2.22	30	1,39E-97	9,91E-97	2	2,48E-03	4,77E-04	6	1,02E-02	1,92E-03	7	3,29E+00	6,95E-01	8	9,38E-132	1,69E-132	1	1,14E-42	6,51E-43	5	3,02E-52	8,49E-53	4	1,08E-81	1,15E-81	3	
Schwefel2.21	30	1,38E-09	3,16E-09	1	8,92E-05	1,55E-05	2	2,44E-03	4,18E-04	4	7,25E+00	1,31E+00	6	1,25E+01	2,18E+00	8	6,28E-01	2,51E-01	5	8,78E+00	1,61E+00	7	1,81E-04	6,62E-05	3	
Step	30	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1	4,84E+00	9,77E-01	8	2,94E-01	9,88E-02	7	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1	1,37E-01	8,06E-02	6	
Quartic	30	5,80E-258	0,00E+00	3	2,07E-14	3,09E-15	7	5,35E-12	7,89E-13	8	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1	4,60E-55	8,18E-55	6	1,91E-107	1,29E-107	5	4,15E-183	0,00E+00	4	
QuarticWN	30	3,86E-03	1,45E-03	4	2,35E-04	3,36E-05	1	5,78E-03	8,09E-04	5	1,53E-03	2,31E-04	2	2,09E-03	3,77E-04	3	8,42E-03	2,75E-03	6	1,17E-02	1,78E-03	8	9,21E-03	2,28E-03	7	
Rosenbrock	30	2,67E+01	2,95E-01	6	2,87E+01	3,81E+00	8	2,75E+01	3,61E+00	7	1,84E+01	2,95E+00	4	1,76E+01	3,47E+00	3	2,15E+01	6,88E+00	5	6,25E-01	2,07E-01	2	1,17E-09	8,47E-10	1	
Rastrigin	30	2,46E+00	2,50E+00	2	4,08E-05	5,14E-06	1	4,16E+01	6,21E+00	7	3,35E+01	4,29E+00	6	7,54E+01	9,61E+00	8	2,68E+01	1,09E+01	5	4,19E+00	6,48E-01	3	2,03E+01	3,61E+00	4	
N-C. Rastrigin	30	1,30E+01	5,37E+00	3	4,08E-05	4,91E-06	1	1,49E+01	3,32E+00	4	4,59E+01	5,73E+00	6	1,04E+02	1,26E+01	8	6,03E+01	1,85E+01	7	5,06E+00	6,47E-01	2	2,94E+01	5,46E+00	5	
Griewank	30	0,00E+00	0,00E+00	1	1,16E-08	1,34E-09	3	1,38E-07	1,58E-08	4	9,75E-03	1,56E-03	8	4,83E-03	9,02E-04	7	1,45E-04	2,87E-04	5	2,13E-03	6,74E-04	6	4,79E-17	1,61E-17	2	
Schwefel2.26	30	8,96E+03	2,34E+03	5	1,26E+04	1,40E+03	8	1,22E+04	1,35E+03	7	4,89E+03	5,41E+02	4	4,16E+03	4,58E+02	3	1,30E+03	3,48E+02	2	3,13E+02	4,27E+01	1	1,05E+04	1,72E+03	6	
Ackley	30	3,62E-15	9,13E-16	1	3,31E-04	3,55E-05	4	1,35E-03	1,44E-04	5	2,31E+00	2,53E-01	8	2,48E-14	2,78E-15	3	6,41E-15	1,64E-15	2	7,41E-01	1,07E-01	7	2,22E-01	7,60E-02	6	
Penalized1	30	7,56E-02	1,88E-02	3	1,56E+00	1,61E-01	7	1,67E+00	1,72E-01	8	9,54E-01	2,06E-01	6	4,74E-01	9,65E-02	5	1,57E-32	3,80E-33	1	2,03E-03	1,50E-03	2	2,71E-01	3,99E-02	4	
Penalized2	30	2,81E+00	6,64E-01	6	2,97E+00	2,99E-01	8	2,90E+00	2,90E-01	7	1,79E+00	9,23E-01	5	1,29E-03	3,73E-04	2	2,15E-04	3,73E-04	1	1,03E-01	5,51E-02	4	2,98E-03	8,95E-04	3	
Alpine	30	3,43E-04	1,98E-04	4	2,03E-02	3,76E-03	6	1,62E-02	7,50E-03	5	7,53E-01	1,11E-01	8	8,67E-15	9,23E-16	2	2,09E-05	3,50E-05	3	2,57E-01	3,91E-02	7	1,20E-17	1,16E-17	1	
Levy	30	8,43E+00	1,92E+00	6	2,97E+01	2,82E+00	8	2,12E+01	2,02E+00	7	9,30E-01	1,49E-01	5	5,45E-01	2,06E-01	4	1,35E-31	3,02E-32	1	2,15E-03	1,45E-03	2	4,31E-02	1,35E-02	3	
Weierstrass	30	0,00E+00	0,00E+00	1	4,70E+01	4,35E+00	7	4,43E+01	4,09E+00	6	9,80E+00	9,10E-01	5	5,22E-01	9,33E-02	4	0,00E+00	0,00E+00	1	4,83E-01	7,01E-02	3	5,15E+01	6,60E+00	8	
Schaffer	30	0,00E+00	0,00E+00	1	2,03E-07	1,83E-08	2	3,30E-06	2,97E-07	3	4,99E-02	5,28E-03	6	1,36E-01	1,28E-02	7	3,67E-02	7,86E-03	5	1,56E-01	1,48E-02	8	9,72E-03	1,21E-03	4	
		Mean Rank		2,76				5,00			5,90			5,62			4,00			3,81			4,43			3,90
		Final Rank		1				6			8			7			4			2			5			3

Çizelge 4.10. pFOA_v2'nin 50 boyut için diğer algoritmalarla karşılaştırmalı sonuçları

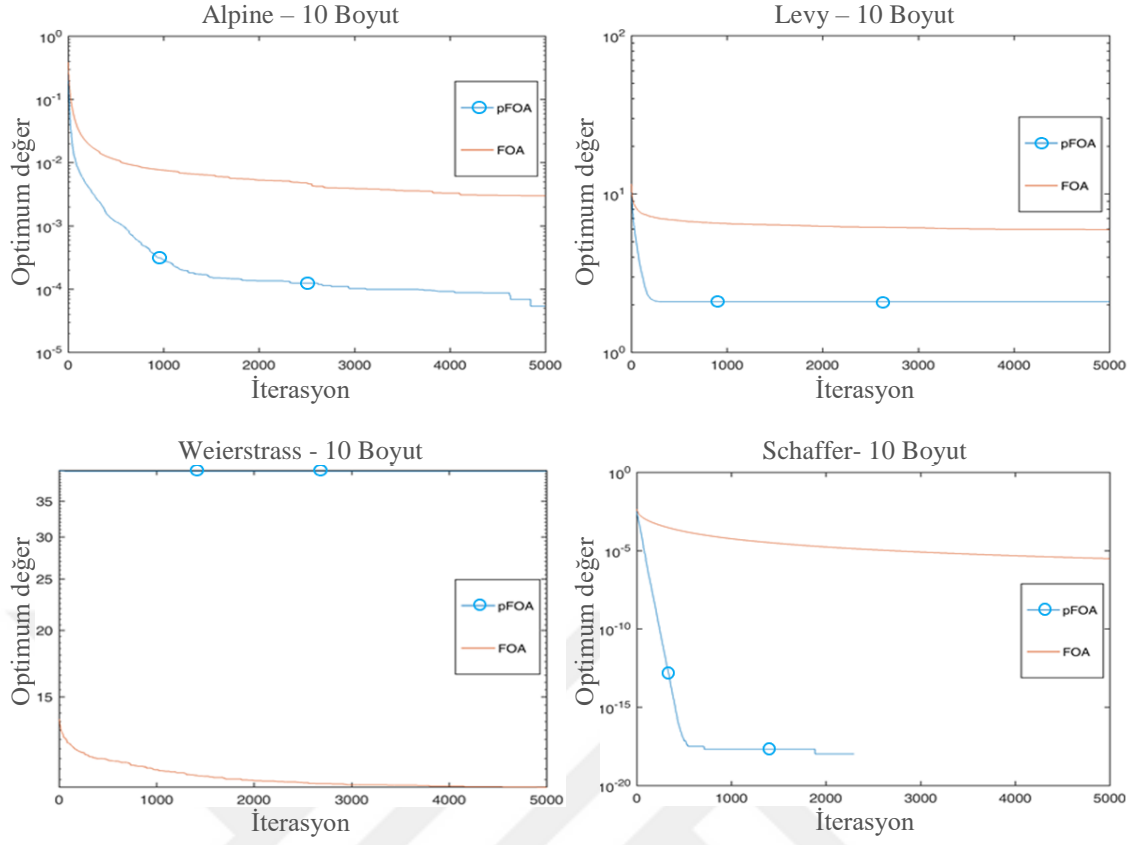
Fonk.	D	pFOA_v2 (Önerilen)			SFOA (Önerilen)			FOA			SPSO2011			FA			TSA			CS			JAYA		
		Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R	Mean	Std. Dev.	R
Sphere	50	1,97E-164	0,00E+00	3	1,29E-07	6,51E-08	7	3,33E-06	1,57E-06	8	7,92E-295	0,00E+00	2	0,00E+00	0,00E+00	1	9,06E-25	8,22E-25	6	1,55E-90	2,25E-90	5	4,79E-116	1,36E-115	4
Elliptic	50	3,50E-228	0,00E+00	1	8,15E-03	2,70E-03	6	7,81E-02	2,46E-02	7	8,55E+04	2,75E+04	8	1,41E-28	2,90E-28	4	9,40E-23	9,03E-23	5	5,60E-87	5,90E-87	3	3,62E-110	6,13E-110	2
SumSquares	50	1,56E-163	0,00E+00	2	3,24E-06	8,35E-07	7	7,87E-05	1,96E-05	8	2,69E-21	1,16E-21	6	0,00E+00	0,00E+00	1	2,34E-25	2,35E-25	5	1,07E-90	1,00E-90	4	4,15E-114	3,92E-114	3
SumPower	50	6,88E-236	0,00E+00	1	1,08E-09	2,34E-10	3	1,58E-09	3,37E-10	4	7,81E-09	1,62E-09	5	1,84E+03	2,63E+03	7	4,56E-03	1,07E-02	6	1,00E+10	2,08E+09	8	4,38E-152	6,15E-152	2
Schwefel2.22	50	7,93E-137	5,66E-136	2	2,55E-03	4,91E-04	6	1,31E-02	2,48E-03	7	1,02E+01	1,98E+00	8	4,68E-219	0,00E+00	1	1,13E-22	5,95E-23	4	5,94E-16	8,01E-16	5	3,43E-50	6,30E-50	3
Schwefel2.21	50	2,56E-07	3,94E-07	1	5,47E-05	9,52E-06	2	4,46E-03	7,66E-04	3	1,66E+01	2,80E+00	5	4,41E+01	7,35E+00	8	2,33E+01	8,76E+00	7	2,15E+01	3,69E+00	6	5,25E-01	1,21E-01	4
Step	50	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1	2,85E+01	4,82E+00	8	2,16E+00	4,35E-01	7	0,00E+00	0,00E+00	1	0,00E+00	0,00E+00	1	1,41E+00	3,81E-01	6
Quartic	50	7,75E-180	0,00E+00	3	8,28E-15	1,23E-15	7	5,23E-12	7,71E-13	8	4,90E-324	0,00E+00	2	0,00E+00	0,00E+00	1	6,83E-18	4,92E-18	6	4,37E-89	2,58E-89	5	4,69E-130	6,35E-130	4
QuarticWN	50	5,05E-03	3,22E-03	3	2,30E-04	3,25E-05	1	7,82E-03	1,09E-03	4	3,37E-03	4,78E-04	2	9,92E-03	1,54E-03	5	4,26E-02	1,38E-02	7	4,63E-02	6,68E-03	8	3,05E-02	6,54E-03	6
Rosenbrock	50	4,69E+01	1,41E+01	5	4,86E+01	6,43E+00	8	4,80E+01	6,31E+00	6	3,52E+01	4,64E+00	4	3,31E+01	5,69E+00	3	4,81E+01	1,57E+01	7	2,88E+00	6,24E-01	2	1,01E-01	5,21E-02	1
Rastrigin	50	4,34E+00	3,61E+00	2	2,59E-05	3,26E-06	1	1,04E+02	1,42E+01	7	6,19E+01	7,81E+00	6	1,68E+02	2,10E+01	8	4,38E+01	1,35E+01	5	1,65E+01	2,22E+00	3	3,48E+01	6,11E+00	4
N-C. Rastrigin	50	2,40E+01	9,76E+00	3	2,59E-05	3,11E-06	1	7,10E+01	1,07E+01	5	1,03E+02	1,30E+01	7	2,04E+02	2,44E+01	8	9,65E+01	2,94E+01	6	1,19E+01	1,48E+00	2	5,29E+01	9,99E+00	4
Griewank	50	0,00E+00	0,00E+00	1	5,03E-09	5,81E-10	4	9,30E-08	1,07E-08	5	6,91E-03	1,29E-03	8	3,14E-03	7,26E-04	6	0,00E+00	0,00E+00	1	3,38E-03	9,31E-04	7	1,08E-15	3,41E-16	3
Scwefel2.26	50	1,55E+04	4,00E+03	6	2,09E+04	2,33E+03	8	2,04E+04	2,25E+03	7	9,00E+03	9,89E+02	5	7,67E+03	8,37E+02	4	3,77E+03	9,88E+02	2	1,20E+03	1,42E+02	1	4,74E+03	9,15E+02	3
Ackley	50	4,67E-15	1,24E-15	1	2,04E-04	2,19E-05	3	1,05E-03	1,12E-04	4	3,37E+00	3,63E-01	8	6,91E-02	3,68E-02	5	1,93E-13	5,45E-14	2	2,27E+00	2,51E-01	7	7,10E-01	1,38E-01	6
Penalized1	50	1,52E-01	2,29E-02	2	1,38E+00	1,43E-01	5	1,47E+00	1,52E-01	6	5,45E+00	7,83E-01	8	1,84E+00	2,52E-01	7	1,54E-01	2,52E-01	3	3,78E-02	9,88E-03	1	6,35E-01	9,26E-02	4
Penalized2	50	4,73E+00	1,88E-01	5	4,95E+00	4,98E-01	7	4,85E+00	4,85E-01	6	6,79E+01	7,58E+00	8	2,25E-01	7,28E-02	3	7,84E-08	5,46E-08	1	3,45E-01	1,06E-01	4	9,20E-03	2,16E-03	2
Alpine	50	2,25E-05	7,15E-05	2	2,58E-04	2,52E-05	4	1,19E-02	7,37E-03	6	6,01E+00	6,55E-01	8	4,27E-05	2,93E-05	3	1,31E-03	2,16E-03	5	1,47E-01	4,36E-02	7	2,83E-17	1,97E-17	1
Levy	50	2,40E+01	5,39E+00	6	4,95E+01	4,70E+00	8	3,77E+01	3,57E+00	7	1,96E+00	2,45E-01	4	2,34E+00	4,22E-01	5	2,90E-26	1,25E-26	1	1,17E-02	6,60E-03	2	1,50E-01	3,72E-02	3
Weierstrass	50	0,00E+00	0,00E+00	1	8,13E+01	7,52E+00	6	8,02E+01	7,38E+00	5	2,44E+01	2,24E+00	4	4,18E+00	4,63E-01	3	5,57E-16	6,29E-16	2	2,75E+00	2,89E-01	2	8,90E+01	1,14E+01	7
Schaffer	50	1,20E-17	5,60E-18	1	1,30E-07	1,17E-08	2	1,94E-04	1,23E-04	3	1,07E-01	9,94E-03	6	3,90E-01	3,50E-02	8	8,02E-02	1,72E-02	5	3,27E-01	2,98E-02	7	1,08E-02	1,51E-03	4
		Mean Rank		2,48			4,62			5,57			5,81			4,67			4,14			4,29			3,62
		Final Rank		1			5			7			8			6			3			4			2



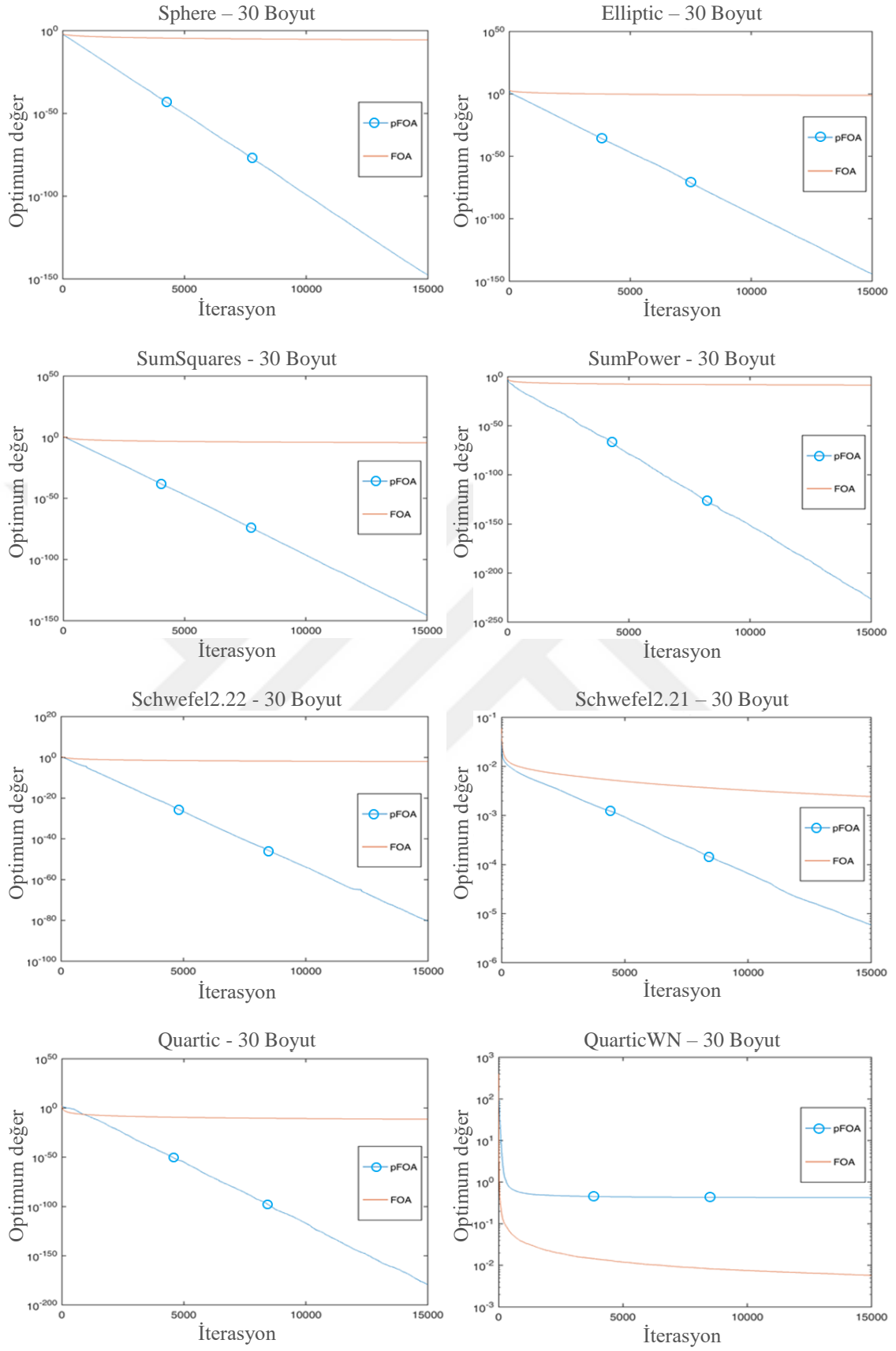
Şekil 4.7.a. pFOA_v1 ve temel FOA'nın 10 boyut için yakınsama grafikleri



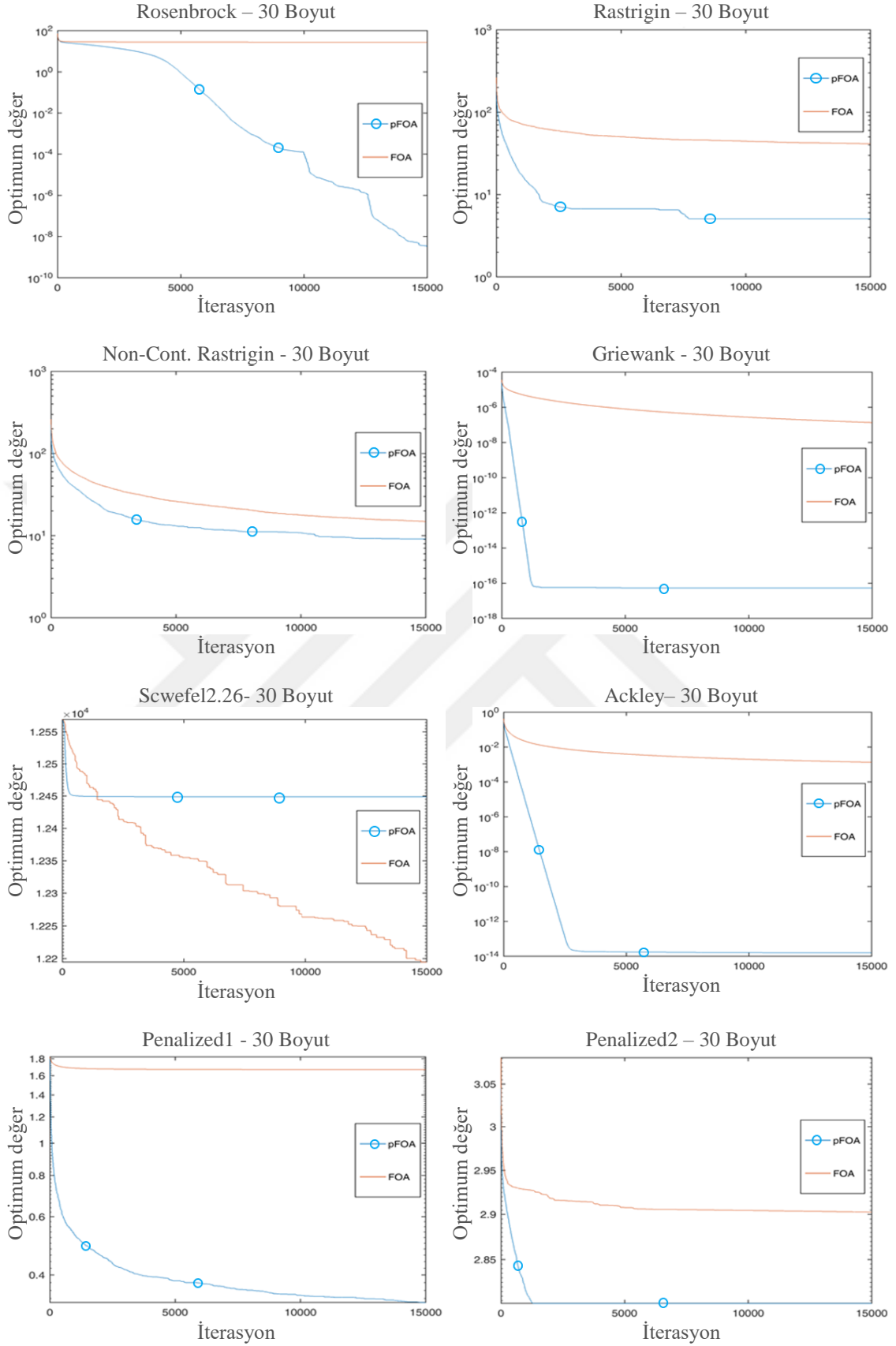
Şekil 4.7.b. pFOA_v1 ve temel FOA'nın 10 boyut için yakınsama grafikleri



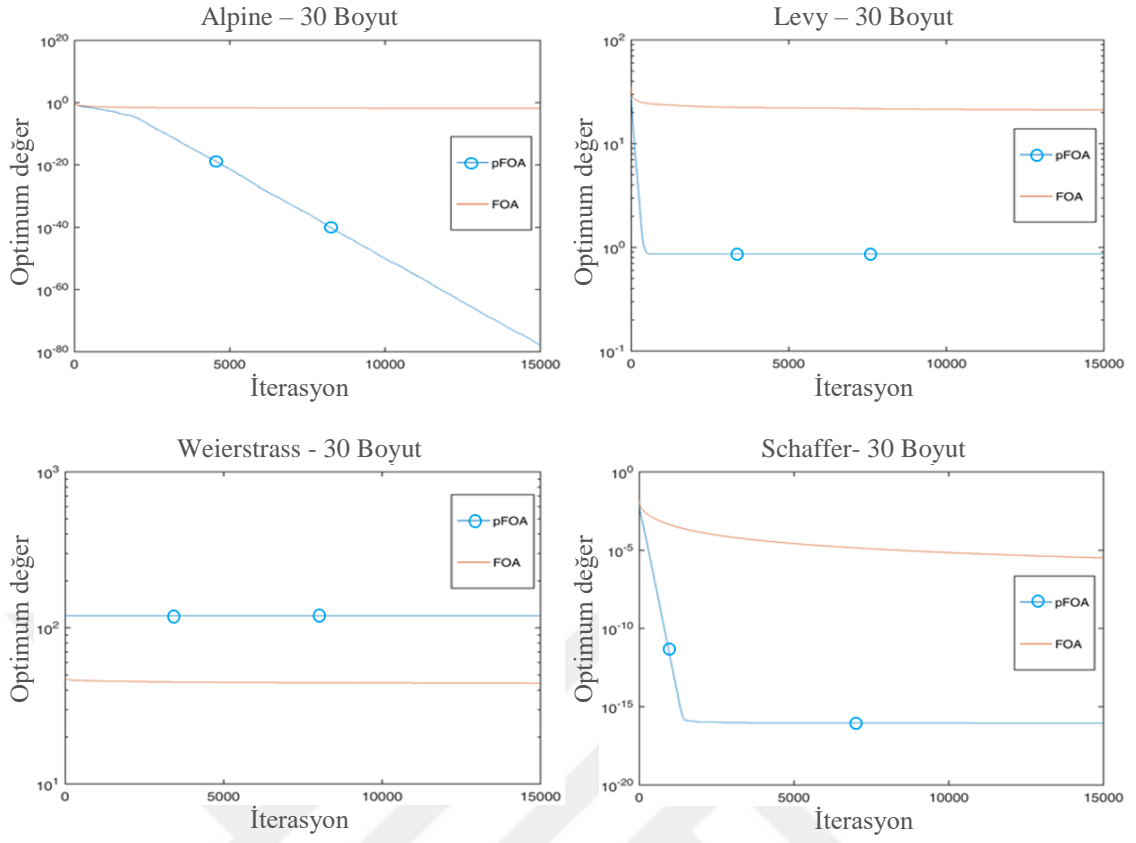
Şekil 4.7.c. pFOA_v1 ve temel FOA'nın 10 boyut için yakınsama grafikleri



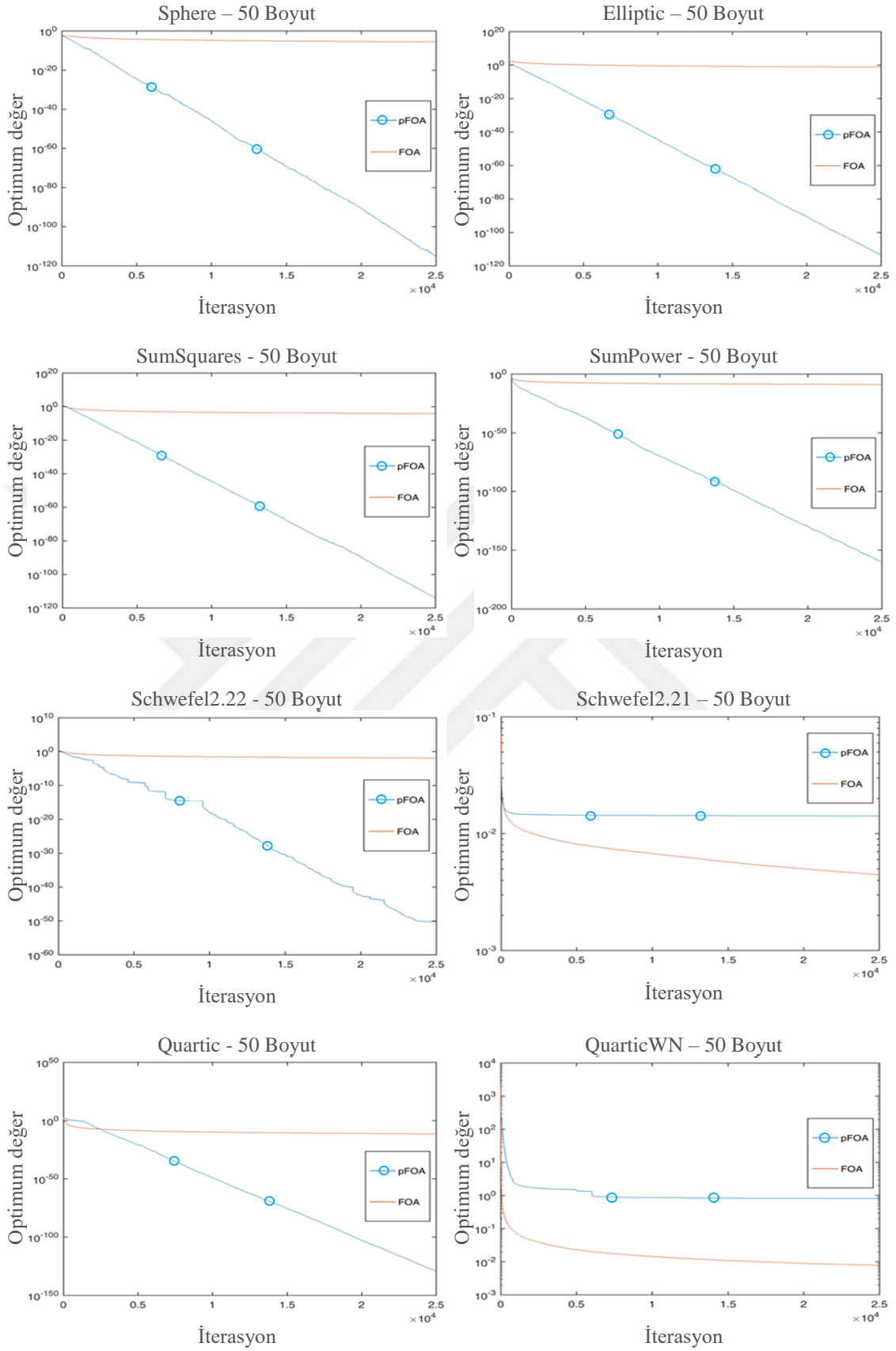
Şekil 4.8.a. pFOA_v1 ve temel FOA'nın 30 boyut için yakınsama grafikleri



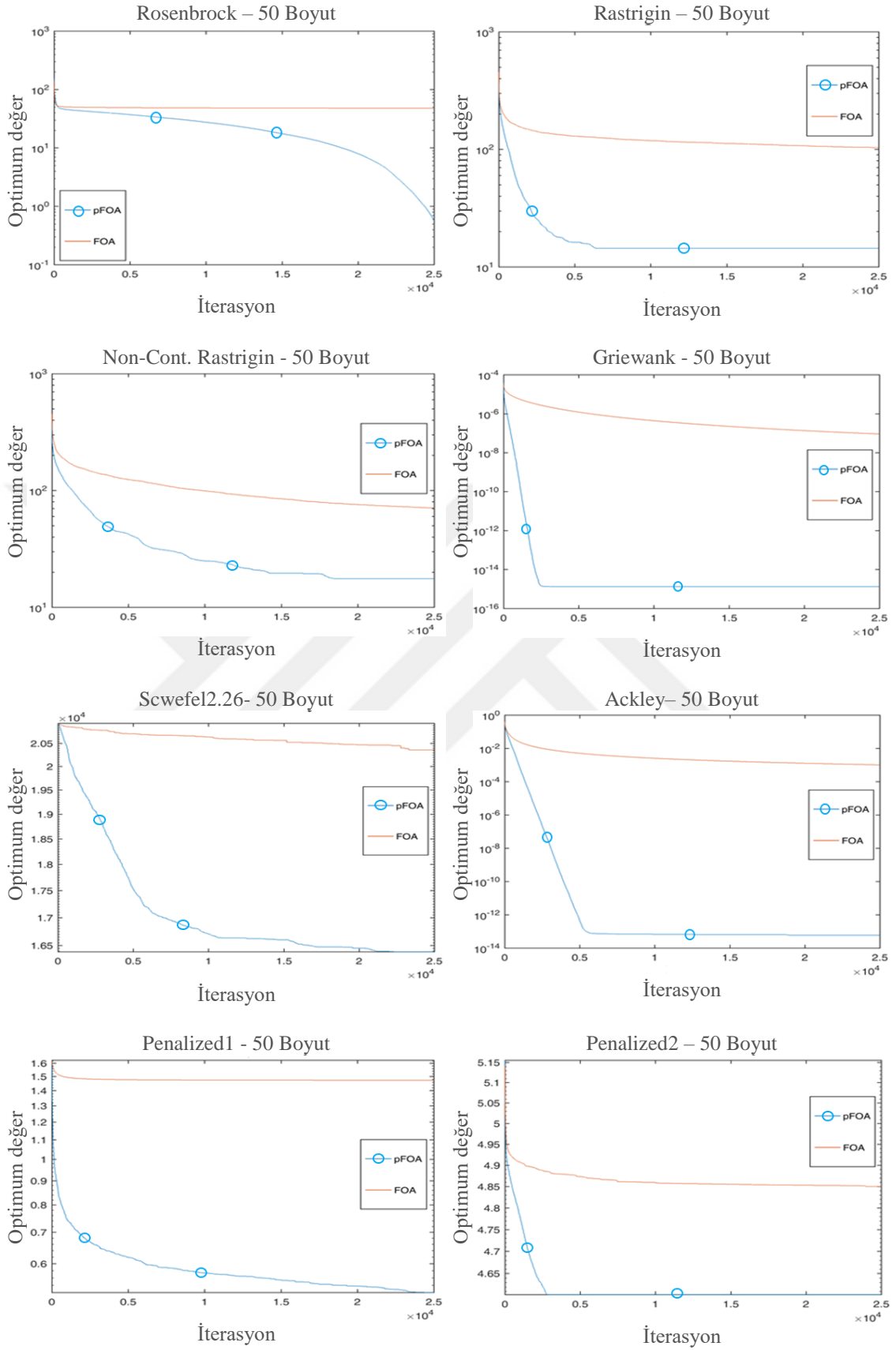
Şekil 4.8.b. pFOA_v1 ve temel FOA'nın 30 boyut için yakınsama grafikleri



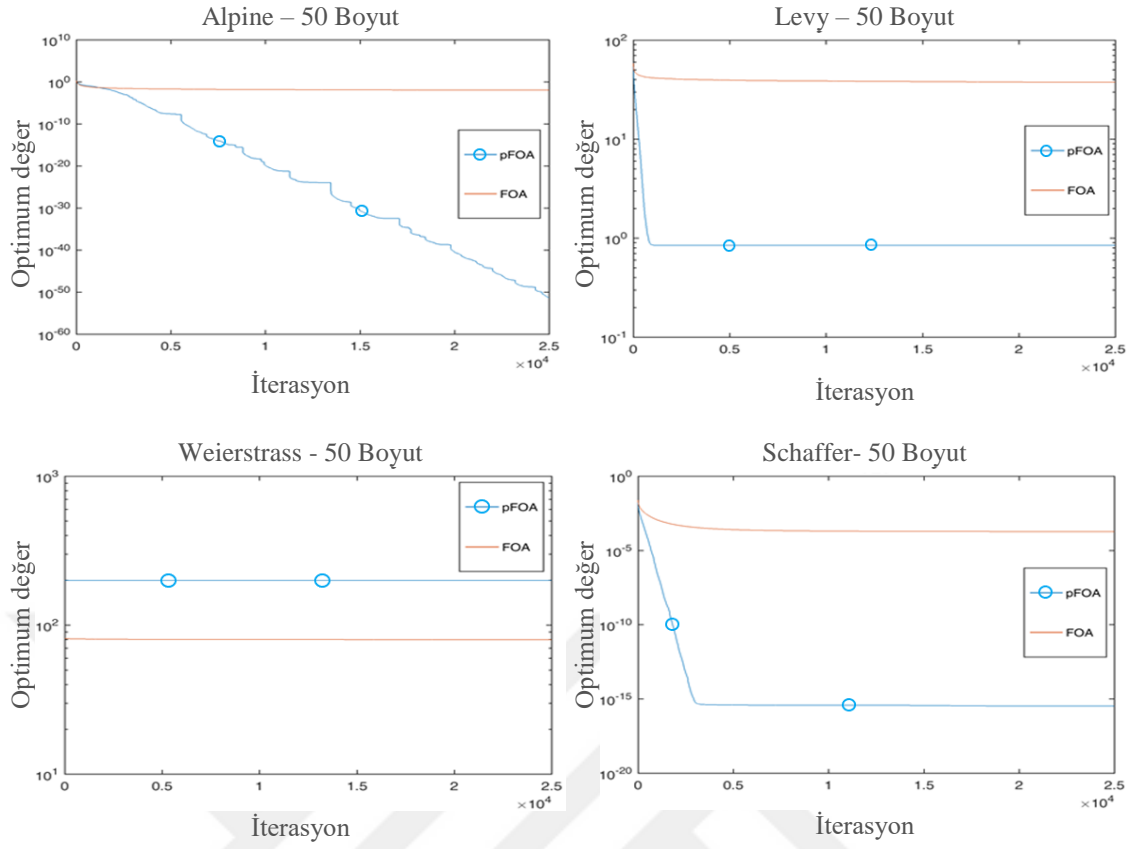
Şekil 4.8.c. pFOA_v1 ve temel FOA'nın 30 boyut için yakınsama grafikleri



Şekil 4.9.a. pFOA_v1 ve temel FOA'nın 50 boyut için yakınsama grafikleri

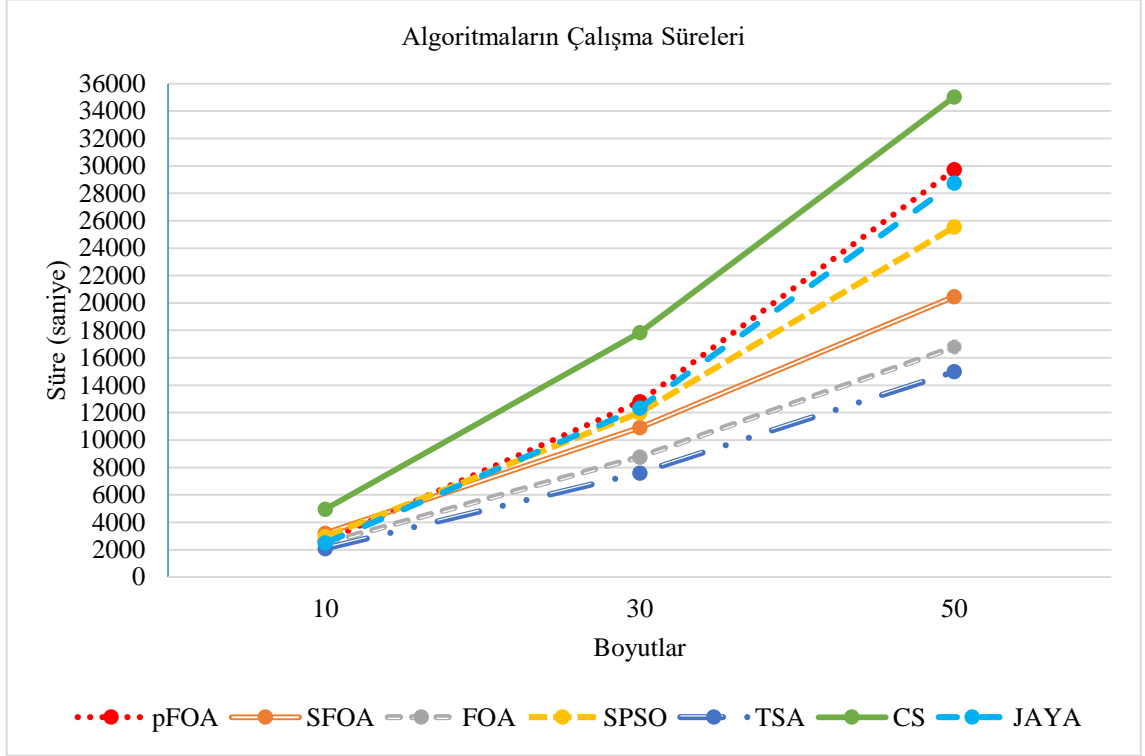


Şekil 4.9.b. pFOA_v1 ve temel FOA'nın 50 boyut için yakınsama grafikleri



Şekil 4.9.c. pFOA_v1 ve temel FOA'nın 50 boyut için yakınsama grafikleri

Ayrıca, karşılaştırılan algoritmaların 10, 30 ve 50 boyutlarındaki uygulama zamanları Şekil 4.10'de verilmiştir. Şekil 4.10'da raporlanan zamanlar 21 kıyaslama fonksiyonunun ortalama çalışma zamanlarıdır. Uygulamalar Intel Coffee Lake Core i7-8700K, 6GB GDDR5 Nvidia GTX1060 OC 192 Bit, 17.3 "FHD 1920x1080 G-SYNC IPS Mat LED Ekran, 16GB (2x8GB) DDR4 2666MHz, 512GB SAMSUNG PM981 M.2 SSD PCIe 3.0 x 4 özelliklerindeki dizüstü bilgisayarda gerçekleştirilmiştir.



Şekil 4.10. Algoritmaların Çalışma Süreleri

Her algoritma için; başlangıç zamanı ve yineleme ayrı ayrı olacak şekilde 51 kez hesaplanmıştır. Hesaplanan zamanların ortalamaları elde edilmiştir. Bildirilen çözümler, başlatma zamanının ve yineleme zamanının ortalamasının toplamıdır.

FA'nin çalışma süresi, karşılaştırılan tüm algoritmalarından daha yüksektir ve bu nedenle, ölçeklendirme sorunlarını önlemek için zaman grafiğine dâhil değildir. Bununla birlikte, FA çalışma süreleri 10 boyutlu 26469 saniye, 30 boyutlu 61200 saniye ve 50 boyutlu 21 problemlerde 79081 saniyedir.

Son olarak pFOA_v1 ve pFOA_v2, Çizelge 4.11’de verilen literatürdeki Chen ve Pi, 2019, Liu ve ark., 2019 ve Wang ve ark., 2019 çalışmalarındaki 16 tanınmış sayısal kıyaslama fonksiyonlarının sonuçları ile karşılaştırılmıştır.

Çizelge 4.11’de verilen 16 tanınmış sayısal kıyaslama fonksiyonları bahsedilen literatürdeki üç çalışmada da ortak kullanılan fonksiyonlardır. Bu sebeple pFOA versiyonları da bu fonksiyonlar üzerinde test edilmiştir.

Bahsedilen literatürdeki üç çalışmada da fonksiyon testlerinde, boyut 30, fonksiyon değerlendirme sayısı ise 50000 olarak alınmış ve 30 kez bağımsız çalışma gerçekleştirilmiştir. pFOA versiyonları da bu parametreler dikkate alınarak test edilmiştir. Popülasyon boyutu ise 10 olarak belirlenmiştir.

Literatürdeki üç çalışmada da FOA’nın geliştirilmiş bir versiyonu sunulmuş, FOA ve varyantları ile karşılaştırılmıştır.

Chen ve Pi, 2019’da geliştirilen SEDMFOA algoritması, IFFO, LGMS-FOA, FOA ve DSLCFOA ile karşılaştırılmıştır.

Liu ve ark., 2019’da geliştirilen IPGS-FFO algoritması, IFFO, LGMS-FOA, FOA ve SIGN-FOA ile karşılaştırılmıştır.

Wang ve ark., 2019’da geliştirilen JS-FOA algoritması, IFFO, IFOA ve HFOA ile karşılaştırılmıştır.

IFFO, LGMS-FOA ve FOA algoritmaları bu üç çalışmada karşılaştırmalar için ayrı ayrı kullanılmıştır. IFFO, LGMS-FOA ve FOA algoritmalarının sonuçları en iyi Chen ve Pi, 2019’da elde edilmiştir. Bu nedenle karşılaştırmalarda, IFFO, LGMS-FOA ve FOA algoritmalarının sadece Chen ve Pi, 2019’daki sonuçları dikkate alınmıştır.

Çizelge 4.12’de pFOA versiyonları ve bahsedilen üç çalışmadaki FOA varyantlarının karşılaştırmaları verilmiştir.

Çizelge 4.12’deki sonuçlara göre, pFOA_v2’nin sıra ortalaması değeri 1’dir ve karşılaştırmadaki diğer FOA varyantlarına göre daha iyi sonuçlar elde ettiği görülmektedir. pFOA_v1’in ise sıra ortalaması değerine bakılarak karşılaştırmadaki diğer FOA varyantlarına göre karşılaştırılabilir ve rekabetçi sonuçlar elde ettiği görülmektedir.

Çizelge 4.11. Kıyaslama Fonksiyonları

NumAdı	SR	C	n	Fonksiyon
1 Sphere	[-100,100]	US		$f_1(\vec{X}) = \sum_{i=1}^n x_i^2$
2 Dixon Price	[-10,10]	UN		$f_2(\vec{X}) = \sum_{i=2}^n i(2x_i^2 - x_{i-1}^2)^2 + (x_1 - 1)^2$
3 Exponential	[-1,1]	UN		$f_3(\vec{X}) = -\exp(-0.5 \sum_{i=1}^n x_i^2)$
4 Rosenbrock	[-30,30]	UN		$f_4(\vec{X}) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
5 Schwefel1.2	[-100,100]	UN		$f_5(\vec{X}) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$
6 Schwefel2.22	[-10,10]	UN		$f_6(\vec{X}) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $
7 SumPower	[-1,1]	MS		$f_7(\vec{X}) = \sum_{i=1}^n x_i ^{i+1}$
8 SumSquares	[-10,10]	US		$f_8(\vec{X}) = \sum_{i=1}^n ix_i^2$
9 Ackley	[-32,32]	MN		$f_9(\vec{X}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$
10 Alpine	[-10,10]	MS		$f_{10}(\vec{X}) = \sum_{i=1}^n x_i \sin(x_i) + 0.1x_i $
11 Schaffer F6	[-100,100]	UN		$f_{11}(\vec{X}) = f(x) = f_1(x_1, x_2) + f_1(x_2, x_3) + \dots + f_1(x_n, x_1)$ $f_1(x, y) = 0.5 + (\sin^2(\sqrt{x^2 + y^2}) - 0.5) / (1 + 0.001(x^2 + y^2))^2$
12 Rastrigin	[-5.12, 5.12]	MS		$f_{12}(\vec{X}) = \sum_{i=1}^{n-1} (x_i^2 - 10 \cos(2\pi x_i) + 10)$
13 Griewank	[-600, 600]	MN		$f_{13}(\vec{X}) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
14 Inverted Cosine Wave	[-5, 5]	MN		$f_{14}(\vec{X}) = -\sum_{i=1}^{n-1} \left(\exp\left(-\frac{x_i^2 + x_{i+1}^2 + 0.5x_i x_{i+1}}{8}\right) \cos\left(4\sqrt{x_i^2 + x_{i+1}^2 + 0.5x_i x_{i+1}}\right) \right)$
15 Pathological	[-100, 100]	MN		$f_{15}(\vec{X}) = \sum_{i=1}^{n-1} \left(0.5 + \sin^2 \frac{\sqrt{100x_i^2 + x_{i+1}^2} - 0.5}{1 + 0.001(x_i^2 - 2x_i x_{i-1} + x_{i+1}^2)^2} \right)^2$
16 Salomon	[-100, 100]	MN		$f_{16}(\vec{X}) = 1 - \cos\left(2\pi \left(\sqrt{\sum_{i=1}^n x_i^2}\right) + 0.1 \sqrt{\sum_{i=1}^n x_i^2}\right)$

5. SONUÇLAR VE ÖNERİLER

5.1 Sonuçlar

Bu tez çalışmasında FOA'nın temel yapısı ve ruhu bozulmadan üç farklı iyileştirme yapılmış ve bu iyileştirmeler sırasıyla SFOA, saFOA ve pFOA şeklinde adlandırılarak sunulmuştur. SFOA, saFOA ve pFOA, Çizelge 3.1 de verilen 21 adet sayısal kıyaslama fonksiyonunda test edilmiştir. Ayrıca, pFOA, literatürdeki üç çalışmanın sonuçları ile Çizelge 4.11 de verilen 16 adet sayısal kıyaslama fonksiyonunda test edilmiştir.

Önerilen her üç algoritmada, sürünün başlangıç konumları için, kıyaslama fonksiyon değişkenlerinin alt ve üst sınırları arasındaki rastgele bir değer kullanılmıştır. Bu sayede FOA'daki sürünün ilk konum çeşitliliğinin zayıflığı giderilmiştir.

SFOA, sineğin koku konsantrasyonu değer vektörünün her bir boyutu için bir işaret elde etmek amacıyla iki işaret değişkeninin uygulanmasıyla geliştirilen FOA'nın geliştirilmiş bir versiyonudur. SFOA algoritması arama alanının sadece pozitif tarafını değil, aynı zamanda bütünü de değerlendirebilir. SFOA'nın işaret değerleri, meyve sineklerinin optimizasyon probleminin olası çözümünü temsil eden negatif koku konsantrasyon değerlerini almasına olanak tanır ve algoritma, negatif karar değerleriyle elde edilen en uygun çözüme sahip olan problemleri çözmek için diğer algoritmalarla rekabet edebilir. Sonuçlar analiz edildiğinde, önerilen yöntemin standart sapmalara dayanan çözüm kalitesi ve sağlamlığı açısından karşılaştırılabilir ve mâkul sonuçlar ürettiği görülmüştür. Buna ek olarak, karar değişkenlerinin sayısı yani problemin boyutu arttıkça, yöntemlerin performansları azalır, çünkü arama alanı, problemin boyutluluğuna bağlı olarak katlanarak büyür. Wilcoxon İşaretli Sıra testi sonuçlarına dayanarak, önerilen algoritma 30 çalışmanın sonuçlarını kullanarak istatistiksel olarak iyi bir çözüm üretmiştir. Şekil 4.1, 4.2 ve 4.3'de gösterilen, önerilen yöntemin ve temel algoritmanın yakınsama grafiklerinden görüleceği üzere, önerilen algoritmanın yakınsama hızı algoritmanın temel sürümünden daha iyidir. Önerilen yöntem, temel FOA, PSO'nun gelişmiş versiyonlarından SPSO2007 ve SPSO2011 algoritmaları, CS algoritması ve FA'nın sonuçlarına göre neredeyse tüm durumlarda, çözüm kalitesi ve sağlamlık açısından daha iyi performans sergilemiştir. Karşılaştırmalı sonuçlar göz önüne alındığında, önerilen SFOA daha iyi ve başarılı sonuçlar elde etmiştir. Ayrıca, SFOA,

temel FOA'ya göre algoritmayı daha uygun ve adil hale getiren negatif arama alanındaki değerleri elde etmesini sağlamıştır.

saFOA, temel FOA'nın karar verme mekanizmasına kendi orijinal yapısına ilave olarak SA algoritmasının çalışma yapısının eklendiği geliştirilmiş bir sürümüdür. Yapılan iyileştirme sadece en iyi çözümler değil belli oranda kötü çözümlerde dikkate alınmıştır. Bu sayede temel FOA'ya nispeten algoritmanın yerel minimuma takılma ihtimali azaltılmıştır. Deneysel sonuçlar saFOA'nın performansının çözüm kalitesi açısından FOA'nın performansından daha iyi olduğunu göstermiştir. saFOA, 21 kıyaslama fonksiyonunun 13'ünde tüm boyutlarda temel FOA'dan daha iyi sonuçlar vermiştir. Önerilen yöntem temel FOA'nın genel arama yeteneğini geliştirmiştir. Şekil 4.4, 4.5 ve 4.6'dan görüleceği üzere her üç boyuttaki yakınsama grafiklerine göre saFOA'nın yakınsama performansı temel FOA'dan daha iyidir.

pFOA_v1 ve pFOA_v2, en iyi ve en kötü çözümlere dayalı yeni bir çözüm güncelleme kuralı önererek temel FOA'nın geliştirilmesine odaklanmıştır. Önerilen pFOA sürümleri, etkinliklerinin doğrulanması için, ilk olarak temel FOA, SFOA adı verilen değiştirilmiş FOA ve diğer akıllı optimizasyon algoritmalarından olan SPSO2011, FA, TSA, CS ve JAYA ile, ikinci olarak ise literatürde üç çalışmadaki FOA varyantları ile karşılaştırılmıştır. Çizelge 4.4'de açıkça görülmektedir ki pFOA'lar kıyaslama fonksiyonlarını çözüme temel FOA'dan daha etkili ve verimlidir. pFOA_v1, 3 fonksiyon dışında, neredeyse tüm durumlarda tüm boyutlarda en iyi değerleri elde etmiştir. pFOA_v2 ise hemen hemen her durumda tüm boyutlarda en iyi değerleri elde etmiştir. Bu nedenle, pFOA'lar temel FOA'dan daha yüksek bir optimizasyon hassasiyetine sahiptir. Yakınsama grafiklerine bakarak, pFOA'ların temel FOA'dan daha hızlı optimum veya neredeyse optimum çözümlere yaklaştığı anlaşılmaktadır. Çizelge 4.12'ye bakıldığında pFOA_v2'nin ortalama sıra değerine göre FOA varyantlarından daha etkili ve verimli olduğu, pFOA_v1'in ise karşılaştırılabilir ve rekabetçi sonuçlar elde ettiği görülmektedir. Sonuç olarak, pFOA sürümlerinin kıyaslama fonksiyonlarındaki performansı, çözüm kalitesi ve sağlamlığı açısından, temel FOA, SFOA, SPSO2011, FA, TSA, CS, JAYA ve literatürde bahsedilen üç çalışmadaki FOA varyantlarına göre daha iyidir. Tüm sonuçları değerlendirince, daha iyi, rekabetçi veya karşılaştırılabilir sonuçların pFOA versiyonları ile elde edildiği belirtilmelidir. Bu sonucun nedeni, çözüm güncelleme kuralının en iyi en kötü temelli çözüm stratejisiyle değiştirilmiş olmasıdır ve temsilciler sadece en iyi çözümü değil aynı zamanda en kötü çözümü de düşünürler. Aynı zamanda temel FOA'daki çözümlerin negatif değerleri alamama problemi bu kuralla yeni

versiyonlarda çözülmüştür. Bu nedenle, çözüm alanı etkili bir şekilde aranır ve bu kural kullanılarak yeni çözümler kolayca keşfedilebilir. Bu, genel arama yeteneğini geliştirir ancak hızlı yakınsamayı azaltır. Genel arama ve yakınsama arasındaki bu değişim, temel FOA'nın aksine önerilen algoritmalarda kabul edilebilir düzeydedir, çünkü temel FOA, sürekli olarak en iyi çözümleri takip ettiğinden dolayı yerel arama yeteneğini ve hızlı yakınsamayı iyileştirme eğilimindedir. pFOA sürümlerinin performansı, sıralama tabanlı sıralama ve yakınsama özellikleri açısından diğer algoritmalarından daha iyidir. Özellikle, önerilen algoritmaların, fonksiyonların sonuçları ve yakınsama grafikleri analiz edildiğinde, çok modlu problemlerin çözümünde temel FOA'dan daha etkili olduğu görülmektedir.

5.2 Öneriler

Önerilen yöntemler, basit bir algoritmik yapıya, az kontrol parametresine ve bu çalışmada kısıtsız sürekli optimizasyondaki yüksek performansa sahip olmalarından dolayı, kısıtlı problemlerin veya gerçek dünya problemlerinin çözümünde uygulanabilir. Önerilen yöntemler, uyarlamalı adım büyüklüğü eklenerek veya güncelleme kuralı değiştirilerek geliştirilebilir. Literatürdeki diğer yerel arama algoritmalar veya farklı sezgisel algoritmalar ile hibrit yaklaşımlar geliştirilerek sürekli, kısıtlı, ikili ve çok amaçlı optimizasyon problemlerini çözmek için kullanılabilir.

KAYNAKLAR

- Abidin, Z. Z., Arshad, M. R. ve Ngah, U. K., 2011, A simulation based fly optimization algorithm for swarms of mini autonomous surface vehicles application.
- Babalık, A., İşcan, H., Babaoğlu, İ. ve Gündüz, M., 2018, An improvement in fruit fly optimization algorithm by using sign parameters, *Soft Computing*, 22 (22), 7587-7603.
- Bonyadi, M. R. ve Michalewicz, Z., 2014, SPSO 2011: Analysis of stability; local convergence; and rotation sensitivity, *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, 9-16.
- Chen, P.-W., Lin, W.-Y., Huang, T.-H. ve Pan, W.-T., 2013, Using fruit fly optimization algorithm optimized grey model neural network to perform satisfaction analysis for e-business service, *Applied Mathematics & Information Sciences*, 7 (2L), 459-465.
- Chen, Y. ve Pi, D., 2019, Novel fruit fly algorithm for global optimisation and its application to short-term wind forecasting, *Connection Science*, 1-23.
- Cheng, X., Chen, L. ve Xu, B., 2018, An Improved Fruit Fly Optimization Algorithm for Solving High Dimensional Problems, *2018 37th Chinese Control Conference (CCC)*, 2310-2316.
- Chikh, R. ve Chikhi, S., 2019, Clustered negative selection algorithm and fruit fly optimization for email spam detection, *Journal of Ambient Intelligence and Humanized Computing*, 10 (1), 143-152.
- Dai, H., Zhao, G., Lu, J. ve Dai, S., 2014, Comment and improvement on “A new fruit fly optimization algorithm: taking the financial distress model as an example”, *Knowledge-Based Systems*, 59, 159-160.
- Dong-qin, J. ve Bu-hai, S., 2013, Based on the FOA algorithm research of ocean-going vessels economy speed, *2013 8th International Conference on Electrical and Electronics Engineering (ELECO)*, 472-476.
- Gunduz, M., Iscan, H. ve Kodaz, H., 2018, Improving the performance of the fruit fly algorithm using simulated annealing, *Journal Of Industrial Engineering Research*, 4 (4), 50-56.
- Han, J., Wang, P. ve Yang, X., 2012, Tuning of PID controller based on fruit fly optimization algorithm, *2012 IEEE International Conference on Mechatronics and Automation*, 409-413.
- Iscan, H., Kiran, M. S. ve Gunduz, M., 2019, A Novel Candidate Solution Generation Strategy for Fruit Fly Optimizer, *IEEE Access*, 7, 130903-130921.
- Karaboga, D. ve Akay, B., 2009, A comparative study of artificial bee colony algorithm, *Applied mathematics and computation*, 214 (1), 108-132.
- Kiran, M. S., 2015, TSA: Tree-seed algorithm for continuous optimization, *Expert Systems with Applications*, 42 (19), 6686-6698.
- Kiran, M. S., Hakli, H., Gunduz, M. ve Uguz, H., 2015, Artificial bee colony algorithm with variable search strategy for continuous optimization, *Information Sciences*, 300, 140-157.
- Kirkpatrick, S., Gelatt, C. D. ve Vecchi, M. P., 1983, Optimization by simulated annealing, *science*, 220 (4598), 671-680.
- Li, C., Xu, S., Li, W. ve Hu, L., 2012a, A novel modified fly optimization algorithm for designing the self-tuning proportional integral derivative controller, *Journal of Convergence Information Technology*, 7 (16).

- Li, H.-Z., Guo, S., Li, C.-J. ve Sun, J.-Q., 2013, A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm, *Knowledge-Based Systems*, 37, 378-387.
- Li, H., Guo, S., Zhao, H., Su, C. ve Wang, B., 2012b, Annual electric load forecasting by a least squares support vector machine with a fruit fly optimization algorithm, *Energies*, 5 (11), 4430-4445.
- Li, S. ve Lu, Z., 2015, Multi-swarm fruit fly optimization algorithm for structural damage identification, *Struct Eng Mech*, 56 (3), 409-422.
- Li, X., Sun, L., Li, J. ve Piao, H., 2018, An Improved Fruit Fly Optimization Algorithm and Its Application in Heat Exchange Fouling Ultrasonic Detection, *Mathematical problems in engineering*, 2018.
- Lin, S.-M., 2013, Analysis of service satisfaction in web auction logistics service using a combination of fruit fly optimization algorithm and general regression neural network, *Neural Computing and Applications*, 22 (3-4), 783-791.
- Lin, W.-Y., 2016, A novel 3D fruit fly optimization algorithm and its applications in economics, *Neural Computing and Applications*, 27 (5), 1391-1413.
- Liu, A., Deng, X., Ren, L., Liu, Y. ve Liu, B., 2019, An Inverse Power Generation Mechanism Based Fruit Fly Algorithm for Function Optimization, *Journal of Systems Science and Complexity*, 32 (2), 634-656.
- Liu, Y., Wang, X. ve Li, Y., 2012, A modified fruit-fly optimization algorithm aided PID controller designing, *Proceedings of the 10th World Congress on Intelligent Control and Automation*, 233-238.
- Liu, Z., Wang, Y. ve Zhang, Y., 2014, Multiple population fruit fly optimization algorithm for automatic warehouse order picking operation scheduling problem, *Journal of Wuhan University of Technology*, 36 (3), 71-77.
- Meng, T. ve Pan, Q.-K., 2017, An improved fruit fly optimization algorithm for solving the multidimensional knapsack problem, *Applied Soft Computing*, 50, 79-93.
- Niu, D. X., Chen, T. T., Wang, P. ve Chen, Y. C., 2014, Forecasting Residential Electricity Based on FOAGMNN, *Advanced Materials Research*, 2513-2517.
- Niu, J., Zhong, W., Liang, Y., Luo, N. ve Qian, F., 2015, Fruit fly optimization algorithm based on differential evolution and its application on gasification process operation optimization, *Knowledge-Based Systems*, 88, 253-263.
- Omran, M., 2007, SPSO 2007 Matlab.
- Pan, Q.-K., Sang, H.-Y., Duan, J.-H. ve Gao, L., 2014, An improved fruit fly optimization algorithm for continuous function optimization problems, *Knowledge-Based Systems*, 62, 69-83.
- Pan, W.-T., 2012, A new fruit fly optimization algorithm: taking the financial distress model as an example, *Knowledge-Based Systems*, 26, 69-74.
- Pan, W.-T., 2013a, Using modified fruit fly optimisation algorithm to perform the function test and case studies, *Connection Science*, 25 (2-3), 151-160.
- Pan, W.-T., 2014, Mixed modified fruit fly optimization algorithm with general regression neural network to build oil and gold prices forecasting model, *Kybernetes*, 43 (7), 1053-1063.
- Pan, W., 2013b, Estimate the Applications of Modified Fruit Fly Optimization Algorithm Feasibility, *2nd International Conference on Industrial Design and Mechanics Power (ICIDMP 2013) Applied Mechanics and Materials*, 845-848.
- Ramachandran, B. ve Bellarmine, G. T., 2014, Improving observability using optimal placement of phasor measurement units, *International Journal of Electrical Power & Energy Systems*, 56, 55-63.

- Rana, A. ve Sharma, A., 2014, Resolving set-streaming stream-shop scheduling in distributed system by mean of an a FOA, *International Journal of Computer Science & Engineering Technology*, 5 (4), 394-403.
- Rao, R., 2016, Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems, *International Journal of Industrial Engineering Computations*, 7 (1), 19-34.
- Rao, R., More, K., Taler, J. ve Ocloń, P., 2016, Dimensional optimization of a micro-channel heat sink using Jaya algorithm, *Applied Thermal Engineering*, 103, 572-582.
- Shan, D., Cao, G. ve Dong, H., 2013, LGMS-FOA: an improved fruit fly optimization algorithm for solving optimization problems, *Mathematical problems in engineering*, 2013.
- Si, L., Wang, Z., Liu, X., Tan, C., Liu, Z. ve Xu, J., 2016, Identification of shearer cutting patterns using vibration signals based on a least squares support vector machine with an improved fruit fly optimization algorithm, *Sensors*, 16 (1), 90.
- Song, R., Xu, Z. ve Liu, Y., 2018, Wireless Sensor Network Coverage Optimization Based on Fruit Fly Algorithm, *International Journal of Online Engineering*, 14 (6).
- Sun, W. ve Ye, M., 2015, Short-term load forecasting based on wavelet transform and least squares support vector machine optimized by fruit fly optimization algorithm, *Journal of Electrical and Computer Engineering*, 2015, 68.
- Tu, C.-S., Chang, C.-T., Chen, K.-K. ve Lu, H.-A., 2012, A study on business performance with the combination of Z-score and FOAGRNN hybrid model, *African Journal of Business Management*, 6 (26), 7788-7798.
- Wang, G., Ma, L. ve Chen, J., 2017, A bilevel improved fruit fly optimization algorithm for the nonlinear bilevel programming problem, *Knowledge-Based Systems*, 138, 113-123.
- Wang, L., Zheng, X.-l. ve Wang, S.-y., 2013, A novel binary fruit fly optimization algorithm for solving the multidimensional knapsack problem, *Knowledge-Based Systems*, 48, 17-23.
- Wang, L., Xiong, Y., Li, S. ve Zeng, Y.-R., 2019, New fruit fly optimization algorithm with joint search strategies for function optimization problems, *Knowledge-Based Systems*, 176, 77-96.
- Wang, X.-G. ve Zou, Z.-J., 2012, Identification of ship manoeuvring response model based on fruit fly optimization algorithm, *Journal of Dalian Maritime University*, 38 (3).
- Wang, X. ve Zou, Z., 2013, FOA-based SVM parameter optimization and its application in ship manoeuvring prediction, *Journal of Shanghai Jiaotong University*, 47 (6), 884-888.
- Wen-Tsao, P., 2011, A new evolutionary computation approach: fruit fly optimization algorithm, *Proceedings of the 11th Conference on Digital Technology and Innovation Management. Taipei, China*, 570.
- Wu, L., Zuo, C. ve Zhang, H., 2015, A cloud model based fruit fly optimization algorithm, *Knowledge-Based Systems*, 89, 603-617.
- Xu, F. Q. ve Tao, Y. T., 2013, The improvement of fruit fly optimization algorithm-using bivariable function as example, *Advanced Materials Research*, 2952-2957.
- Yang, H., Wang, S., Li, G. ve Mao, T., 2018, A new hybrid model based on fruit fly optimization algorithm and wavelet neural network and its application to underwater acoustic signal prediction, *Mathematical problems in engineering*, 2018.

- Yang, S.-C., Lee, C.-S. ve Lee, H.-S., 2012a, Evaluation of logistic flow service satisfaction using the evolutionary computation technique and general regression neural network technique, *International Journal of Advancements in Computing Technology*, 4 (11).
- Yang, S.-C., Lee, C.-S. ve Lee, H.-S., 2012b, Construction of the prediction model of business operation performance in the electronic industry, *J. Business Management and Economics*, 3 (2), 079-089.
- Yang, X.-S., 2009, Firefly algorithms for multimodal optimization, *International symposium on stochastic algorithms*, 169-178.
- Yang, X.-S. ve Deb, S., 2009, Cuckoo search via Lévy flights, *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, 210-214.
- Zhang, P. ve Wang, L., 2014, Grouped fruit-fly optimization algorithm for the no-wait lot streaming flow shop scheduling, *International Conference on Intelligent Computing*, 664-674.
- Zhang, Y., Li, D. ve Lao, D., 2012, Smith predictor in the DDE application, *2012 24th Chinese Control and Decision Conference (CCDC)*, 2346-2351.
- Zhang, Y., Cui, G., Wu, J., Pan, W.-T. ve He, Q., 2016a, A novel multi-scale cooperative mutation fruit fly optimization algorithm, *Knowledge-Based Systems*, 114, 24-35.
- Zhang, Y., Cui, G., Zhu, E. ve He, Q., 2016b, AFOA: An adaptive fruit fly optimization algorithm with global optimizing ability, *International Journal on Artificial Intelligence Tools*, 25 (06), 1650032.
- Zheng, X.-L., Wang, L. ve Wang, S.-Y., 2014, A novel fruit fly optimization algorithm for the semiconductor final testing scheduling problem, *Knowledge-Based Systems*, 57, 95-103.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Hazim İŞCAN
Uyruğu : T.C.
Doğum Yeri ve Tarihi : Konya 1973
Telefon : 0 332 2238646
Faks : 0 332 2410635
E-Posta : hiscan@ktun.edu.tr

EĞİTİM

Derece	Adı, İlçe, İl	Bitirme Yılı
Lise	: Fatih Teknik Lisesi, Selçuklu, Konya	1991
Üniversite	: Gazi Üniversitesi, Teknik Eğitim Fakültesi Bilgisayar Sistemleri Eğitimi Bölümü, Teknikokullar, Ankara	1996
Yüksek Lisans	: Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği A.B.D., Selçuklu, Konya	2004
Doktora	: Konya Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Bilgisayar Mühendisliği A.B.D., Selçuklu, Konya	2019

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2018-Devam Ediyor	Konya Teknik Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Bilgisayar Mühendisliği	Öğr.Gör.
2017-2018	Selçuk Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği	Öğr.Gör.
2012-2015	Kop Bölge Kalkınma İdaresi Başkanlığı, Bilgi İşlem Departmanı	Danışman
1997-2017	Selçuk Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği	Uzman
1996-1997	Selçuk Üniversitesi, Bilgi İşlem Dairesi Başkanlığı	Geçici Gör.

UZMANLIK ALANI

Sürü Zekası Algoritmaları, Yapay Zeka Optimizasyon Yöntemleri, Makine Öğrenmesi

YABANCI DİLLER

İngilizce

PROJE GÖREVLERİ

Bluetooth ve Sosyal Ağ Tabanlı Mobil Uygulamalı Panik Butonu, Diğer kamu kuruluşları (Yükseköğretim Kurumları hariç), Danışman, 04/10/2015 - 03/10/2016 (ULUSAL)

Meyve Sineği Algoritmasının Performansını İyileştirmek İçin Yeni Yaklaşımlar, Yükseköğretim Kurumları tarafından destekli bilimsel araştırma projesi, Araştırmacı: Hazim İŞCAN, Yürütücü: Mesut GÜNDÜZ, 30/04/2018 – 15/10/2019 (ULUSAL)

YAYINLAR

Uluslararası hakemli dergilerde yayımlanan makaleler:

Iscan Hazim, Kiran Mustafa Servet, Gunduz Mesut (2019). A Novel Candidate Solution Generation Strategy for Fruit Fly Optimizer, IEEE Access, 7, 130903-130921. **(Doktora tezinden yapılmıştır)**

Babalik Ahmet, **İşcan Hazim**, Babaoğlu İsmail, Gündüz Mesut (2018). An improvement in fruit fly optimization algorithm by using sign parameters. Soft Computing, 22(22), 7587-7603., Doi: 10.1007/s00500-017-2733-1 **(Doktora tezinden yapılmıştır)**

Gunduz, M., **Iscan, H.** ve Kodaz, H., 2018, Improving the performance of the fruit fly algorithm using simulated annealing, Journal Of Industrial Engineering Research, 4 (4), 50-56. **(Doktora tezinden yapılmıştır)**

Çınar Ahmet Cevahir, **İşcan Hazim**, Kiran Mustafa Servet (2018). Tree-Seed Algorithm for Large-Scale Binary Optimization. KnE Social Sciences, 3(1), 48, Doi: 10.18502/kss.v3i1.1396

İşcan Hazim, Gündüz Mesut (2014). Parameter Analysis on Fruit Fly Optimization Algorithm. Journal of Computer and Communications, 02(04):137-141, Doi: 10.4236/jcc.2014.24018

Kiran Mustafa Servet, **İşcan Hazim**, Gündüz Mesut (2013). The analysis of discrete artificial bee colony algorithm with neighborhood operator on traveling salesman problem. Neural Computing and Applications, 23(1), 9-21., Doi: 10.1007/s00521-011-0794-0

Uluslararası bilimsel toplantılarda sunulan ve bildiri kitaplarında (proceedings) basılan bildiriler :

İşcan Hazim, Kamal Lubna Luay, Kodaz Halife (2018). A New Hybrid Classifier based on Bat Algorithm and Artificial Neural Networks. 4th International Conference on Engineering Science and Technology (ICEST 2018) (Tam Metin Bildiri/Sözlü Sunum)

Gündüz Mesut, **İşcan Hazim**, Kodaz Halife (2018). Improving the performance of the fruit fly algorithm using simulated annealing. 4th International Conference on Engineering Science and Technology (ICEST 2018) (Tam Metin Bildiri/Sözlü Sunum) **(Doktora tezinden yapılmıştır)**

İşcan Hazim, Gündüz Mesut (2016). An application of fruit fly optimization algorithm for traveling salesman problem. 8th International Conference on Advances in Information Technology, IAIT2016, 111, 58-63., Doi: 10.1016/j.procs.2017.06.010 (Tam Metin Bildiri/Sözlü Sunum)

İşcan Hazim, Gündüz Mesut (2015). A Survey on Fruit Fly Optimization Algorithm. 2015 11th International Conference on Signal-Image Technology Internet-Based Systems (SITIS), Doi: 10.1109/SITIS.2015.55 (Tam Metin Bildiri/Sözlü Sunum)

İşcan Hazim, Gündüz Mesut (2014). Parameter Analysis on Fruit Fly Optimization Algorithm. 2nd Conference on Artificial Intelligence and Data Mining (AIDM 2014), 2(04), 137-141., Doi: 10.4236/jcc.2014.24018 (Tam Metin Bildiri/Sözlü Sunum)

İnan Onur, **İşcan Hazim**, Yıldız Ferruh (2013). The Analysis of the Impact of Socio-Economic Status of Students with Decision Tree and Cluster Algorithms. Advanced Computer Systems and Networks: Design and Application, 211-214. (Tam Metin Bildiri/Sözlü Sunum)

İşcan Hazim, Kıran Mustafa Servet, Gündüz Mesut (2011). Supply Chain Optimization Using Ant System. The International Conference on Computing and Information Technology, 17-20. (Tam Metin Bildiri/Sözlü Sunum)

Babalık Ahmet, Baykan Ömer Kaan, **İşcan Hazim**, Babaoğlu İsmail, Fındık Oğuz (2010). Effects of Feature Selection Using Binary Particle Swarm Optimization on Wheat Variety Classification. International Conference on Advances in Information Technology, 11-17., Doi: 10.1007/978-3-642-16699-0_2 (Tam Metin Bildiri/Sözlü Sunum)(Yayın No:4709306)

İşcan Hazim, Fındık Oğuz, Kodaz Halife, Erdi Ali (2009). Designing a special purpose e-commerce website. Proceedings of the 2nd International Conference on Interaction Sciences Information Technology, Culture and Human - ICIS 'xx09, 508-513., Doi: 10.1145/1655925.1656018 (Tam Metin Bildiri/Sözlü Sunum) **(Yüksek Lisans tezinden yapılmıştır)**

Kodaz Halife, Babaoğlu İsmail, **İşcan Hazim** (2009). Thyroid disease diagnosis using artificial immune recognition system AIRS. 2nd International Conference on Interaction Sciences Information Technology, Culture and Human (Tam Metin Bildiri/Sözlü Sunum)

Ulusal bilimsel toplantılarda sunulan ve bildiri kitaplarında basılan bildiriler:

Ülker Erkan, Fındık Oğuz, Arslan Ahmet, **İşcan Hazim** (2006). Selçuk Üniversitesinde Sayısal İmza Uygulaması. Ulusal Elektronik İmza Sempozyumu, 123-129. (Tam Metin Bildiri/Sözlü Sunum)