

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAZILIM MÜHENDİSLİĞİ ANABİLİM DALI

ÇOK AMAÇLI OPTİMİZASYON ALGORİTMALARININ
TASARIMI İÇİN KÜMELEME ESASLI YENİ BİR ARŞİV İNDİRGEME
YÖNTEMİNİN GELİŞTİRİLMESİ VE UYGULANMASI

YÜKSEK LİSANS TEZİ

Mustafa AKBEL

ŞUBAT 2024

TRABZON



KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAZILIM MÜHENDİSLİĞİ ANABİLİM DALI

**ÇOK AMAÇLI OPTİMİZASYON ALGORİTMALARININ TASARIMI
İÇİN KÜMELEME ESASLI YENİ BİR ARŞİV İNDİRGEME YÖNTEMİNİN
GELİŞTİRİLMESİ VE UYGULANMASI**

Mustafa AKBEL

**Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünce
"YÜKSEK LİSANS (YAZILIM MÜHENDİSLİĞİ)"
Unvanı Verilmesi İçin Kabul Edilen Tezdir.**

Tezin Enstitüye Verildiği Tarih : 28 / 12 / 2023

Tezin Savunma Tarihi : 02 / 02 / 2024

Tez Danışmanı : Prof. Dr. Hamdi Tolga KAHRAMAN

Trabzon 2024

ÖNSÖZ

Lisans eğitiminden buyana, akademik gelişimimde bana yol gösteren ve bu süreçte SCI indeksli Q1 çeyreğindeki dergilerde dört makalemizin yayınlanmasında başlıca katkıyı sağlayan çok değerli tez danışmanım sayın Prof. Dr. Hamdi Tolga KAHRAMAN'a içtenlikle teşekkür ederim. Çok amaçlı optimizasyon üzerine geliştirdiğimiz algoritmaların elektrik güç sistemleri alanındaki gerçek dünya mühendislik problemlerine tatbik edilmesinde sağladığı önemli bilgi ve katkılar için sayın Doç. Dr. Serhat DUMAN'a da çok teşekkür ederim. k-ortalamlar kümeleme yönteminin tatbik edilmesinde sağladığı katkılar için sayın Seyithan TEMEL'e teşekkürlerimi sunarım. Yüksek lisans eğitimim süresince sabır, destek ve sevgisiyle, bu yaşıma kadar benim her anımda yanımda olan ve desteklerini hiçbir zaman esirgemeyen canım aileme ve eşime çok teşekkür ederim.

Mustafa AKBEL

Trabzon 2024

TEZ ETİK BEYANNAMESİ

Yüksek Lisans Tezi olarak sunduğum “Çok Amaçlı Optimizasyon Algoritmalarının Tasarımı İçin Kümeleme Esaslı Yeni Bir Arşiv İndirgeme Yönteminin Geliştirilmesi ve Uygulanması” başlıklı bu çalışmayı baştan sona kadar danışmanım Prof. Dr. Hamdi Tolga KAHRAMAN’a sorumluluğunda tamamladığımı, verileri/örnekleri kendim topladığımı, deneyleri/analizleri ilgili laboratuvarlarda yaptığımı/yaptırdığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 10/01/2024

Mustafa AKBEL

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ.....	III
TEZ ETİK BEYANNAMESİ.....	IV
İÇİNDEKİLER.....	V
ÖZET	VII
ŞEKİLLER DİZİNİ	IX
TABLolar DİZİNİ.....	X
SEMBOLLER VE KISALTMALAR DİZİNİ	XI
1. GENEL BİLGİLER	1
1.1. Giriş	1
1.2. Çok Amaçlı Optimizasyon	4
1.2.1. Problem Tanımı.....	4
1.2.2. Çok Amaçlı Optimizasyon Problemlerinin Çözümlemesinde Kullanılan Yaklaşımlar	6
1.3. Pareto-tabanlı Çok Amaçlı Optimizasyonda Tanımlar ve Terminoloji	8
1.3.1. Karar Uzayı	8
1.3.2. Amaç Uzayı.....	8
1.3.3. Pareto-Baskınlık İlişkisi	9
1.3.4. Pareto-Optimal	10
1.3.5. Pareto-Optimal Set	11
1.3.6. Pareto-Optimal Cephe	11
1.4. Performans Metrikleri	12
1.4.1. IGDX ve IGDF	12
1.4.2. Pareto Set Proximity	13
1.4.3. Hypervolume	14
1.5. Uyarlanabilir Rehberlik Mekanizmalı Diferansiyel Evrim Algoritması.....	15
1.5.1. Popülasyon Oluşturma Evresi	16
1.5.2. Rehber Seçim Mekanizması Tasarımı	18
1.5.3. Komşuluk Araması ve Çeşitlilik	18
1.5.4. Güncelleme Mekanizması Tasarımı.....	19

1.6. Arşiv İndirgeme.....	19
1.6.1. Klasik Kalabalık Mesafesi Yöntemi	20
1.6.2. Dinamik Anahtarlama Kalabalık Yöntemi	22
1.7. K-Ortalamalar Kümeleme Yöntemi	28
2. YAPILAN ÇALIŞMALAR.....	31
2.1. Önerilen Yöntem	31
2.1.1. Dinamik Referans Uzaylar Esaslı Kümeleme ve Arşiv İndirgeme	32
2.1.2. Önerilen Yöntemin ÇAURDE Algoritmasına Tatbik Edilişi.....	36
3. BULGULAR VE TARTIŞMA	40
3.1. Deneysel Çalışma Çevresi.....	40
3.1.1. Ayarlar ve Adillik	40
3.1.2. IEEE CEC 2020 MMOP Karşılaştırma Suiiti	41
3.1.3. İstatistiksel Test Yöntemleri ve Analiz	42
3.2. DRUK-ÇAURDE Versiyonlarının Performanslarının İncelenmesi.....	43
3.2.1. Friedman Skorları.....	43
3.2.2. Yakınsama Performansları	44
3.3. DRUK-ÇAURDE'nin Rakiplerle Karşılaştırılması	54
3.3.1. Friedman Skorları.....	54
3.3.2. Yakınsama Analizi	55
4. SONUÇLAR.....	67
5. ÖNERİLER.....	69
6. KAYNAKLAR	70
ÖZGEÇMİŞ	

Yüksek Lisans Tezi

ÖZET

ÇOK AMAÇLI OPTİMİZASYON ALGORİTMALARININ TASARIMI İÇİN KÜMELEME ESASLI YENİ BİR ARŞİV İNDİRGEME YÖNTEMİNİN GELİŞTİRİLMESİ VE UYGULANMASI

Mustafa AKBEL

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Yazılım Mühendisliği Anabilim Dalı
Danışman: Prof. Dr. Hamdi Tolga KAHRAMAN
2024, 77 Sayfa

Amaç fonksiyonları çatışma halinde olan çok amaçlı optimizasyon problemlerinin çözümlenmesinde başlıca zorluk küresel en iyi çözüm setinin kararlı ve etkili bir şekilde bulunmasıdır. Literatürdeki çalışmalar kalabalık mesafesi yönteminin arama uzayındaki çeşitliliği sağlamada yetersiz kaldığını rapor etmektedir. Bu tez çalışmasında Pareto-tabanlı çok amaçlı evrimsel arama algoritmalarının çok tipli çok amaçlı optimizasyon problemleri üzerindeki arama performansının iyileştirilmesi için yeni bir arşiv indirgeme yöntemi önerilmektedir. Önerilen yöntemde arşiv indirgeme işlemi dinamik anahtarlamalı referans uzaylar esaslı kümeleme algoritması tarafından yerine getirilmektedir. Bu süreçte, kümeleme mekanizmasının görevi arşivdeki vektörlerin sayısında aşım olduğunda, vektörleri benzerliklerine göre gruplandırmak ve her kümeden sadece bir vektör seçerek arşivi güncellemektir. Kümeleme işleminin gerçekleştirilmesinde referans alınan uzay vektörünün dinamik bir şekilde belirlenmesi sayesinde karar ve amaç uzaylarında çeşitlilik sağlanabilmektedir. Test çalışmalarından elde edilen sonuçlara göre, tasarlanan algoritma hem baz algoritmadan, hem iyileştirilmiş versiyonundan hem de güçlü rakiplerinden daha iyi performans metrik değerlerine sahiptir.

Anahtar Kelimeler: Dinamik referans uzaylar esaslı kümeleme, Arşiv indirgeme

Master Thesis

SUMMARY

DEVELOPMENT AND IMPLEMENTATION OF A NEW CLUSTERING-BASED ARCHIVE REDUCTION METHOD FOR THE DESIGN OF MULTI-OBJECTIVE OPTIMIZATION ALGORITHMS

Mustafa AKBEL

Karadeniz Technical University
Institute of Science and Technology
Department of Software Engineering
Supervisor: Prof. Dr. Hamdi Tolga KAHRAMAN
2024, 77 Page

The main challenge in solving multi-objective optimization problems with conflicting objective functions is to find the global optimal solution set in a stable and efficient way. Studies in the literature report that the crowd distance method fails to provide diversity in the search space. In this thesis, a new archive reduction method is proposed to improve the search performance of Pareto-based multi-objective evolutionary search algorithms on multimodal multi-objective optimization problems. In the proposed method, the archive reduction process is performed by a clustering algorithm based on dynamic switched reference spaces. In this process, the task of the clustering mechanism is to group the vectors according to their similarity and update the archive by selecting only one vector from each cluster when the number of vectors in the archive is exceeded. By dynamically determining the reference space vector for the clustering process, the decision and objective spaces can be varied. According to the results obtained from the test runs, the algorithm designed has better performance metric values than both the base algorithm, its improved version and its strong competitors.

Keywords: Dynamic reference spaces, Archive reduction

ŞEKİLLER DİZİNİ

	<u>Sayfa No</u>
Şekil 1. Karar değişkenleri ve amaç vektörleri uzaylarının temsili	5
Şekil 2. PS ve PC Gösterimleri	10
Şekil 3. IGD metriği.....	12
Şekil 4. Kalabalık mesafe hesaplaması	21
Şekil 5. Dinamik Anahtarlama Kalabalık Mesafesi	23
Şekil 6. K-ortalamlar algoritma şeması.....	30
Şekil 7. Dinamik referans uzay vektörü esaslı kümeleme ve arşiv indirgeme süreci.....	32
Şekil 8. Kümeleme sürecinin işleyişi.....	34
Şekil 9. Kümeleme Algoritması.....	35
Şekil 10. DRUK-ÇAURDE'nin en iyi üç versiyonunun PF yakınsama performansları.....	46
Şekil 11. DRUK-ÇAURDE'nin en iyi üç versiyonunun PS yakınsama performansları.....	50
Şekil 12. DRUK-ÇAURDE'nin en iyi üç versiyonunun IGD _X ve IGD _F kutu grafikleri	53
Şekil 13. Friedman skorlarına göre rekabetçi ilk üç algoritmanın PF yakınsama performansları.....	58
Şekil 14. Friedman skorlarına göre rekabetçi ilk üç algoritmanın PS yakınsama performansları.....	62
Şekil 15. Friedman skorlarına göre rekabetçi ilk üç algoritmanın PS yakınsama performansları.....	65

TABLÖLAR DİZİNİ

	<u>Sayfa No</u>
Tablo 1. CEC 2020’de tanımlı CAO problemlerinin özellikleri.....	42
Tablo 2. Algoritmaların performans metrik değerlerine göre elde ettikleri Friedman skorları.....	44
Tablo 3. DRUK-ÇAURDE’nin ve rakiplerinin Friedman skorları.....	54



SEMBOLLER VE KISALTMALAR DİZİNİ

CR	: Cover rate
ÇAEA	: Çok amaçlı evrimsel algoritmalar
ÇAO	: Çok amaçlı optimizasyon
ÇAURDE	: Çok amaçlı uyarlanabilir rehberlik mekanizmalı diferansiyel evrim
DAK	: Dinamik anahtarlamalı kalabalık
DE	: Diferansiyel Evrim
DRU	: Dinamik referans uzaylar
DRUK	: Dinamik referans uzaylarda kümeleme
DRUK-ÇAURDE	: Dinamik referans uzaylarda kümeleme esaslı arşiv indirgeme yöntemi geliştirilmiş ve çok amaçlı uyarlanabilir rehberlik mekanizmalı diferansiyel evrim
DUK	: Dinamik uzayda kümeleme
HV	: Hypervolume
PC	: Pareto cephe
PS	: Pareto set
PSP	: Pareto set proximity
RUV	: Referans uzay vektörü
UD	: Uygunluk değeri
URDE	: Uyarlanabilir rehberlik mekanizmalı diferansiyel evrim

1. GENEL BİLGİLER

Bu bölümde tez çalışmasında yürütülen araştırmalar ve tezin kapsamı genel olarak tanıtılmakta ve çok amaçlı optimizasyon ile ilgili temel bilgiler sunulmaktadır.

1.1. Giriş

Optimizasyon, önceden belirlenen ya da anlık olarak ortaya çıkan bir amacı ve bu amacın gereksinimlerini azami kazançla ve asgari maliyetle yerine getiren şartları keşfetmektir. Optimizasyon problemleri esas olarak iki öğeden oluşur. Bu öğeler amaç fonksiyonları ve tasarım parametreleridir. Optimizasyon problemlerinin sınıflandırılmasında birçok durum etkilidir. Amaç fonksiyonu sayısının bir olması durumu tek amaçlı, birden fazla olması durumu ise çok amaçlı optimizasyon problemi olarak tanımlanır. Tasarım parametrelerinin veri türlerine bağlı olarak optimizasyon problemleri, sürekli değerli, ayrık değerli, ikili ve kombinasyon problemleri olarak tanımlanırlar. Optimizasyon probleminin tasarım parametrelerinin alabileceği değerler üzerinden tanımlanan uzaya arama uzayı denir. Arama uzayının sınırları arasında yapılan arama tüm optimizasyon problemleri için geçerli bir kısıttır. Bunun yanında, tasarım parametrelerinin arama uzayı sınırları için genellikle kısıt ifadesi kullanılmaz. Optimizasyon problemlerinde kısıt terminolojisi tasarım parametrelerinin birbirleriyle olan ilişkilerini tanımlamak için kullanılır. Tasarım parametrelerinin birbirleriyle olan ilişkilerinin matematiksel denklemlere dönüştürülmesinde eşitlik ve eşitsizlik kısıtları kullanılır. Bu iki tip kısıt dışında, ikili ve kombinasyon tipli optimizasyon problemlerinde kural tabanlı kısıtlar da tanımlanabilir.

Bir optimizasyon probleminin karmaşıklık düzeyini etkileyen çok sayıda durum ve parametre vardır. Amaç ve kısıt fonksiyonlarının sayısı, amaç ve kısıt fonksiyonlarının matematiksel ifadelerinin tipleri ve karmaşıklık düzeyleri, tasarım parametrelerinin sayısı, tasarım parametrelerinin sayısı, tasarım parametrelerinin alabilecekleri değerlerin uzayı optimizasyon problemlerinin karmaşıklık düzeyini doğrudan etkileyen parametreler ve durumlardır. Amaç fonksiyon sayısının tek ya da birden fazla oluşu problemin karmaşıklığını önemli oranda artırmaktadır. Tek amaçlı optimizasyon problemlerinin çözümlenmesinde sadece bir amaç fonksiyon için en iyi çözümü temsil eden tasarım parametreleri aranmaktadır. Buna karşın, çok amaçlı optimizasyon problemlerinde amaç

fonksiyonları çatışma halinde olmaması durumunda, diğer tüm şartların birbirine denk olduğu varsayılırsa problemin karmaşıklık düzeyinde önemli bir değişiklik olmaz. Bu tür çok amaçlı optimizasyon problemlerinin çözülmesinde, amaç fonksiyonları için önem derecesini temsil eden ağırlıklar atamak ve amaç fonksiyonlarının toplamını maksimize ya da minimize etmek mümkündür. Buna karşın, karar değişkenleri (tasarım parametreleri) aldıkları değerler ile amaç fonksiyonlarının bazıları için daha iyi çözümler sunarken bazıları için ise daha kötü çözümler sunabilirler. Bu tür durumlar amaç fonksiyonları çatışma halinde olan optimizasyon problemleri olarak tanımlanırlar. Amaç fonksiyonları çatışma halinde olan optimizasyon problemlerinin terminolojisi ve bu tür problemlerin çözülmesinde kullanılan prosedürler tek amaçlı ve amaç fonksiyonları çatışma halinde olmayan çok amaçlı optimizasyon çalışmalarından farklıdır.

Amaç fonksiyonları çatışma halinde olan çok amaçlı optimizasyon problemleri için küresel en iyi çözümden bahsedilemez. Bunun yerine, amaç uzayında birbirlerine karşı üstünlükleri ve zayıflıkları olan ve birbirlerine salt bir baskınlık kuramayan optimum çözümlerin setinden yani Pareto yüzeyden bahsedilebilir. Amaç fonksiyonları çatışma halinde olan çok amaçlı optimizasyon problemleri için karar ve amaç uzaylarının optimum Pareto setlerini bulabilmek için her iki uzayda da yerel çözüm tuzaklarından kurtulabilen ve sömürü-keşif yeteneklerini dengeli bir şekilde sürdürebilen algoritmaların geliştirilmesine ihtiyaç vardır. Literatürde bu amaçla yaygın olarak, Pareto-tabanlı arşivleme ve kalabalık mesafesi tabanlı arşiv indirgeme mekanizmasına sahip çok amaçlı evrimsel arama algoritmaları üzerine araştırmalar yürütülmektedir. Pareto-tabanlı yaklaşımlar sayesinde, en az bir amaç fonksiyonda rakiplerinden daha iyi çözüme sahip olan vektörlerin arşivlenmesi amaçlanırken, kalabalık mesafesi yöntemi sayesinde de arşivleme sürecinin çeşitliliği gözeterek yürütülmesi amaçlanmaktadır. Bu süreçte karşılaşılan başlıca problem ise arşivdeki çeşitliliğin amaç ve karar uzaylarında dengeli bir şekilde yürütülememesi ve arama sürecinin ilk evrelerinde yerel çözüm tuzaklarına yakalanılmasıdır. Bu durum arama sürecinin sonunda bulunan amaç ve karar uzayı setlerinin hedeflenen küresel en iyi setlerle örtüşmemesine ve aramanın başarısızlıkla sonuçlanmasına neden olmaktadır.

Literatürde, amaç ve karar uzaylarında çeşitliliği sağlamak için çeşitli yöntemler önerilmiştir. Bu yöntemlerden biri de dinamik referans uzaylar (DRU) esaslı kalabalık mesafesi yöntemidir. DRU yöntemi esaslı kalabalık mesafesi yöntemi sayesinde arşivin indirgendiği ve çeşitliliğin sağlandığı referans uzayın arama süreci yaşam döngüsü boyunca değişmesi sağlanmıştır. Böylelikle, arşivdeki indirgeme işlemi sadece bir referans uzayda

değil, arama sürecinde anlık olarak belirlenen amaç, karar ve bileşke referans uzaylardan her birinde gerçekleşebilmektedir. Arşiv indirgeme sürecindeki bu dinamik yapıya geçiş sayesinde kalabalık mesafesi yönteminin performansında istatistiksel olarak kayda değer bir iyileşme olduğu görüldü. Yapılan çalışmalar DRU yaklaşımının kalabalık mesafesi yöntemi üzerinde statik referans uzay yaklaşımına kıyasla daha etkili olduğunu gösterdi (Kahraman, Akbel, Duman, Kati, & Sayan, 2022).

DRU yönteminin kalabalık mesafesi esaslı arşiv indirgeme mekanizmasının tatbik edildiği dinamik anahtarlama kalabalık (DAK) mesafesi esaslı çok amaçlı uyarlanabilir rehberlik mekanizmalı diferansiyel evrim algoritması (ÇAURDE) (Kahraman, Akbel, Duman, Kati, & Sayan, 2022) üzerinde başarılı olması, bu yöntemin performansının farklı arşiv indirgeme yöntemleri üzerinde de araştırılması için güçlü bir motivasyon yaratmıştır. Bu tez çalışmasında, DRU yöntemi ilk defa kümeleme esaslı bir arşiv indirgeme sürecinde tatbik edildi. Geliştirilen arşiv indirgeme yaklaşımı dinamik referans uzaylarda kümeleme (DRUK) esaslı arşiv indirgeme yöntemi olarak adlandırıldı. DRUK arşiv indirgeme yönteminin performansının araştırılması ve bu yöntemin DAK yöntemiyle kıyaslanabilmesi için deneysel çalışmalarda ÇAURDE algoritmasının kullanılmasına karar verilmiştir. Böylelikle tatbik edildikleri algoritmalar aynı olmak üzere, DRU yöntemi kullanılarak geliştirilen DAK-esaslı arşiv indirgeme ve DRUK esaslı arşiv indirgeme algoritmalarının performansları incelenmiştir. Bu açıklamalara göre, bu tez çalışmasında dinamik referans uzaylarda kümeleme esaslı arşiv indirgeme yöntemi geliştirilmiş ve çok amaçlı uyarlanabilir rehberlik mekanizmalı diferansiyel evrim algoritması üzerinde tatbik edilerek (DRUK-ÇAURDE) algoritması geliştirilmiştir. Geliştirilen DRUK-ÇAURDE algoritmasının farklı versiyonları tasarlanmış ve bu versiyonların çoklu moda sahip çok amaçlı optimizasyon problemleri karşılaştırma problemleri üzerindeki performansları araştırılmıştır. Tez çalışmasının literatüre katkıları aşağıda maddeler halinde sıralanmıştır:

- DRU yöntemi ilk defa kümeleme esaslı arşiv indirgeme sürecinde tatbik edilmiş ve DRUK arşiv indirgeme yöntemi geliştirilmiştir.
- DRUK arşiv indirgeme yöntemi ÇAURDE algoritmasında tatbik edilmiş ve klasik kümeleme esaslı arşiv indirgeme yönteminden daha başarılı sonuçlar elde edildiği performans metriklerinden elde edilen verilerle ve istatistiksel analiz sonuçları esas alınarak doğrulanmıştır.
- Tez çalışmasında geliştirilen DRUK-ÇAURDE algoritmasının kalabalık mesafesi esaslı DAK-ÇAURDE algoritması ile karşılaştırılması yapılmış ve önerilen

algoritmanın daha başarılı sonuçlar elde edildiği performans metriklerinden elde edilen verilerle ve istatistiksel analiz sonuçları esas alınarak doğrulanmıştır.

- DRUK-ÇAURDE algoritmasının literatürdeki farklı rakip algoritmalarla karşılaştırması yapılmış ve önerilen algoritmanın rakiplerinden daha başarılı sonuçlar elde ettiği performans metriklerinden elde edilen verilerle ve istatistiksel analiz sonuçları esas alınarak doğrulanmıştır.

Bu tez çalışması sekiz bölümden oluşmaktadır. İlk bölümde, çok amaçlı optimizasyon (ÇAO) algoritmalarına ve ÇAO problemlerine ilişkin olarak temel terminolojik bilgiler sunulmaktadır. İkinci bölümde, tez çalışmasında kullanılan ÇAURDE algoritmasında arama sürecinin yürütülmesinden sorumlu olan uyarlanabilir rehberlik mekanizmalı diferansiyel evrim algoritması tanıtılmaktadır. Üçüncü bölümde, ÇAO algoritmalarında arşiv indirgeme sürecinin tasarımı için kullanılan kalabalık mesafesi yöntemi ve DRU yöntemi açıklanmaktadır. Dördüncü bölümde, K-ortalamlar kümeleme yöntemi tanıtılmaktadır. Beşinci bölümde, bu tez çalışmasında ilk defa önerilen DRUK yöntemi ve DRUK-ÇAURDE algoritması tanıtılmaktadır. Altıncı bölümde DRUK yönteminin ve DRUK-ÇAURDE algoritmasının performansı incelenmektedir. Yedinci ve sekizinci bölümlerde de sırasıyla sonuçlar ve öneriler sunulmaktadır.

1.2. Çok Amaçlı Optimizasyon

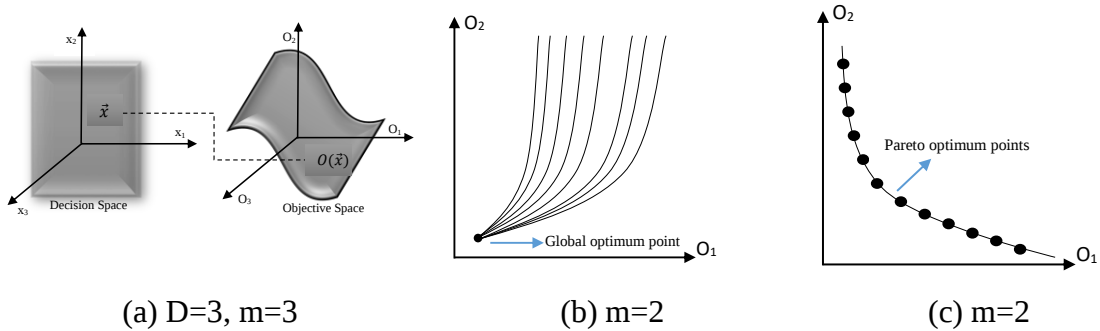
1.2.1. Problem Tanımı

Optimizasyon problemleri birçok özelliğe bağlı olarak sınıflandırılmaktadırlar. Bu problemlerin formülasyonu, tasarım değişkenlerinin sayısı, tasarım değişkenlerinin türleri ve amaç fonksiyonları ile kısıtlamaların fonksiyonlarıdır. Buna göre optimizasyon problemlerinin genel matematiksel formu denklem 1’de verilmektedir:

Amaç	$\underset{X \in R^n}{\text{minimize / maximize}} O(\vec{x}) = \{o_1(\vec{x}), o_2(\vec{x}), o_3(\vec{x}), \dots, o_m(\vec{x})\}$	
Fonksiyon		
Eşitlik Kısıtları	$\phi_j(\vec{x}) = 0, (j = 1, 2, \dots, J),$	(1)
Eşitsizlik Kısıtları	$\varphi_k(\vec{x}) \leq 0, (k = 1, 2, \dots, K)$	
Kısıtları	$\vec{x}=(x_1, x_2, x_3, \dots, x_D), \quad d=1, 2, \dots, D \quad lb_d \leq x_d \leq ub_d$	

Burada $O(\vec{x})$, m -adet amaç fonksiyonundan oluşan bir vektördür. $\vec{x}$, D -boyutlu tasarım değişkenleri vektörü, lb_d ve ub_d değerleri x_d karar değişkeni için alt ve üst sınırları, $\phi_j(\vec{x})$, eşitlik kısıtlarını ve $\varphi_k(\vec{x})$, eşitsizlik kısıtlarını temsil eder. Çok amaçlı ve D -boyutlu bir optimizasyon probleminde arama uzayının sınırları karar değişkenleri vektörü $\vec{x}$ in olası tüm değerleri ile tanımlanır. m -adet amaç fonksiyondan oluşan bir çok amaçlı optimizasyon probleminde amaç fonksiyonların olası tüm değerleri “amaç uzayının” sınırlarını tanımlar.

$m \geq 2$ olduğu durumlarda problem çok amaçlı olarak tanımlanır. Çok amaçlı optimizasyon problemlerinin çözümlenmesindeki en büyük zorluk, amaç fonksiyonların çatışma halinde olduğu durumdur. Bu durumda, çok amaçlı optimizasyon problemlerinde birbirinden üstün olmayan çok sayıda çözüm mevcuttur. Bu tanımlamalara göre $\vec{x}=(x_1, x_2, x_3)$ iken karar değişkenleri uzayı ve $m=3$ iken amaç uzayı Şekil 1-a’da verilmektedir (Özkış, 2017).



Şekil 1. Karar değişkenleri ve amaç vektörleri uzaylarının temsili

Çok amaçlı optimizasyon sürecinin zorluğu esas itibariyle probleme ait amaç fonksiyonlarının çatışma (çelişme) halinde olup olmaması belirler. Amaç fonksiyonlarının çatışma halinde olmaması durumunda problemin tek amaçlı optimizasyon çalışmalarında olduğu gibi çözümlenmesi mümkündür. Bu tür problemlerin çözümlenmesi sürecinde amaç fonksiyonları ağırlıklandırılabilir ve tek bir amaç fonksiyona indirgenebilmektedir. Bu durumda tüm amaçlar için en uygun değeri üreten tek bir çözüm noktası olur (Şekil 1-b). Çok amaçlı optimizasyon problemlerinde amaç fonksiyonlarının çatışma halinde olması durumu ise aşılması gereken önemli bir durumdur. Çelişen amaç fonksiyonlarından kastedilen şey, bir amaç fonksiyonu için optimum değer üreten bir çözüm, diğer amaç fonksiyonlar için kötü değerler üretebilmesidir. $m=2$ iken, amaç fonksiyonlarının çatışma halinde olması durumu Şekil 1-c'de verilmektedir. Amaç fonksiyonlarının çatışma halinde olduğu çok amaçlı optimizasyon problemlerinin çözümlenmesi için pareto-tabanlı yaklaşımlar kullanılmaktadır. Pareto-tabanlı yaklaşımlarda birbirlerine üstünlük kuramayan çözümlerden oluşan bir çözüm adayları kümesi (pareto set) oluşturulur.

1.2.2. Çok Amaçlı Optimizasyon Problemlerinin Çözümlenmesinde Kullanılan Yaklaşımlar

Pareto tabanlı yaklaşımlar: Çok amaçlı problemleri çözmek için erken evrimsel algoritmaları, popülasyonu sıralama ve pareto'nun baskınlık ilişkisini entegre ederek pareto optimal cepheye olan yakınlığı değerlendirmektedir. Uygun bir pareto cephe yaklaşımı, aynı zamanda yakınsama ve çeşitliliği de sağlamalıdır. Bu sebepten dolayı çözümleri birtakım kazançlar elde etmek için bazı kayıpları göze alarak pareto eğrisi etrafında dağıtmak için pareto baskınlığı, ikinci bir faktör ile iş birliği yapılmalıdır. Çözümleri pareto cephesi boyunca dağıtmak için önerilen bazı yöntemler arasında; uygunluk değer paylaşımı, kümeleme (Tian, Shao, Xie, & Zhang, 2023; Zhou, ve diğerleri, 2023), kalabalık mesafesi (Kahraman, Akbel, Duman, Kati, & Sayan, 2022; Duman, Akbel, & Kahraman, 2021) vb. bulunmaktadır. 2000'li yılların başlarında, pareto tabanlı çok amaçlı evrimsel algoritmalar (ÇAEA), en yaygın kullanılan stratejilerden biri haline geldi. Ancak, çözümleri uygun bir şekilde yaymak ve yüksek boyutlu amaç uzaylarında ayrımcılık özelliklerini kaybetme zorluğu nedeniyle kullanımları azalmıştır.

Ayrıştırma Tabanlı Yaklaşımlar: Son on yılda, birçok evrimsel yaklaşım, ölçeklendirme fonksiyonlarını kullanarak, ayrılmaya dayalı olarak adlandırılan evrimsel çok

amaçlı optimizasyon algoritmaları ortaya çıkarmıştır (Zhu, Qin, Yang, Xu, & Zhou, 2023). Ayırıştırma tabanlı yaklaşımlar, aynı sayıda ağırlık vektörü tarafından formüle edilen bir dizi ölçeklendirme fonksiyonunu çözmeye dayanmaktadır (Yin & Cao, 2023). Bu çok amaçlı problemleri çözmek için bu strateji, karmaşık test problemleriyle başa çıkmak için faydalı olmuştur (Wang, Li, Zhang, Hu, & Shen, 2019). Bu prensibin kullanımı, çok amaçlı problemlerle başa çıkmak için uygun bir alternatif haline gelmiş olsa da performansı öncelik olarak uygun bir şekilde dağıtılmış ağırlık vektörlerine bağlıdır (Xie, Li, Li, Huang, & He, 2023; Xiong, Wang, Li, Feng, & Liu, 2024; Bao, Gao, Gu, Xu, & D.Goodman, 2023).

Performans Metriği Tabanlı Yaklaşımlar: Performans metriği tabanlı evrimsel çok amaçlı optimizasyon algoritmaları, çevresel seçim prosedürlerinde performans metriklerini kullanmaktadır (Hsieh, 2023). Evrimsel çok amaçlı optimizasyon algoritmaların performansını değerlendirmek için birçok metrik bulunmaktadır. Bu metrikler, farklı şekillerde yakınsama veya çeşitliliği veya her ikisini aynı anda değerlendirmektedir. Özellikle, bir çok amaçlı optimizasyon problemlerinin nispeten iyi bir pareto ön cephe temsili, hypervolume (Fonseca, Paquete, & Lopez-Ibanez, 2006; Beume, Fonseca, Lopez-Ibanez, Paquete, & Vahrenhold, 2009) metriği veya R2 (Liu, Liu, Li, & Li, 2019), IGD (Sun, Yen, & Yi, 2018) veya Δp (Liu, Liu, Li, & Li, 2019) gibi referans setleri üzerine kurulan performans metrikleri kullanılarak elde edilebilir. Referans setlerine dayalı performans metriği tabanlı evrimsel algoritmalar, çoğu durumda, arama öncesinde belirlemesi zor olan bir referans setinin uygun bir tanımına bağlıdır. Bununla birlikte, hypervolume tabanlı performans metriği tabanlı evrimsel algoritmalar yalnızca hypervolume metriğini hesaplamak için tek bir referans vektörüne ihtiyaç duyar. Ancak, bu yaklaşımlar, hedef sayısının artmasıyla eksponansiyel olarak artan hypervolume metriği değerlerini hesaplamak için gereken yüksek hesaplama maliyeti ile sınırlıdır. Yine de hypervolume metrik tabanlı evrimsel algoritmaların kullanım avantajı, bunların çeşitli pareto ön cepheleriyle başa çıkabilmesidir, bunlar arasında konveks, konkav, karışık, bağlantısız ve dejeneratif şekiller bulunmaktadır. Özellikle, hypervolume tabanlı evrimsel çok amaçlı optimizasyon algoritmalar lineer pareto ön cephelerinde bir düz nokta kümesi elde edebilir ve bu da düzenli karışım tasarımları oluşturmak için kullanılmasını teşvik eder.

Buna göre pareto-tabanlı yaklaşıma ait tanımlamalar ve terminoloji aşağıda açıklanmaktadır.

1.3. Pareto-tabanlı Çok Amaçlı Optimizasyonda Tanımlar ve Terminoloji

1.3.1. Karar Uzayı

Karar Uzayı, çok amaçlı optimizasyon problemlerinde kullanılan bir terimdir. Bu terim, problemdeki çeşitli hedefleri veya amaçları temsil etmek üzere tanımlanan değişkenlerin oluşturduğu bir uzayı ifade eder. Her bir değişken, bir karar noktasının belirli bir konumunu tanımlar ve bu karar noktasının objektif fonksiyonlardaki performansını etkiler.

Çok amaçlı optimizasyon, genellikle çelişen hedefleri içerir. Yani, bir hedefi iyileştirmek diğer hedefin performansını olumsuz etkileyebilir. Pareto tabanlı çok amaçlı optimizasyon algoritmaları, bu çelişkileri çözmek veya dengelemek için kullanılır. Bu algoritmalar, Pareto hakimiyeti kullanarak çözümleri sıralar ve çeşitli amaçları optimize eden bir Pareto ön cephesi oluştururlar (Zheng, Du, Zou, & Yang, 2023).

Karar Uzayı terimi, bu çok boyutlu uzayda yer alan potansiyel çözümleri ifade eder. Yani, her bir karar noktası, uzaydaki belirli bir konumu temsil eder ve bu konum, karar değişkenlerinin belirli değerlerine bağlı olarak belirlenir. Pareto tabanlı algoritmalar, bu karar uzayındaki çözümleri etkili bir şekilde tarayarak, çeşitli amaçları en iyi şekilde dengeleyen veya iyileştiren çözümleri bulmaya çalışırlar.

1.3.2. Amaç Uzayı

Amaç Uzayı, çok amaçlı optimizasyon problemlerinde, optimize edilmeye çalışılan birden fazla hedefin değerlerini içeren bir uzaydır. Her bir amaç, genellikle optimize edilmek istenen bir ölçüdür veya belirli bir kriteri temsil eder. Amaç uzayı, bu farklı hedeflerin kombinasyonlarını içerir ve her kombinasyon, bir çözümün performansını tanımlar.

Çok amaçlı optimizasyon problemlerinde, genellikle çeşitli amaçlar arasında bir çelişki vardır, yani bir amacın iyileştirilmesi diğer amaçları olumsuz etkileyebilir. Amaç uzayı, her bir amaç fonksiyonunun bir boyutunu temsil eder. Bu uzayda her bir nokta, farklı amaçların belirli bir kombinasyonunu temsil eder.

Pareto tabanlı çok amaçlı optimizasyon algoritmaları, amaç uzayında çözümleri değerlendirir ve çeşitli amaçları optimize eden bir Pareto ön cephesi oluştururlar. Bu ön

cephes, hiçbir ama aısından gerekleřtirilemeyen özömleri ierir ve özömleri birbirine göre sıralar. Ama uzayı, optimize edilmeye alıřılan hedeflerin eřitli kombinasyonlarını ifade eden, ok boyutlu bir matematiksel uzaydır (Zheng, Du, Zou, & Yang, 2023).

1.3.3. Pareto-Baskınlık İliřkisi

Pareto baskınlık iliřkisi, ok amalı optimizasyon problemlerinde özömlerin karřılařtırılmasında kullanılan bir kriterdir. Bu iliřki, bir özömlün diğesine göre üstünlüğünü belirlemek iin kullanılır ve özellikle Pareto tabanlı ok amalı optimizasyon algoritmalarında önemli bir rol oynar.

Pareto baskınlık iliřkisi, bir özömlün diğ bir özömlü "domine etmesi" veya "pareto daha iyi olma" durumunu belirtir. Bir özöml, diğlerinden en az bir hedef aısından daha iyi ve hiçbir hedef aısından daha kötü değılse, bu özöml diğelerini pareto baskınlığı ile domine eder.

Matematiksel olarak, özöml A'nın özöml B'yi Pareto domine ettiğini ifade etmek iin řu kriter kullanılır:

- 1- Her bir ama fonksiyonu iin en az birinde A, B'den daha iyi olmalıdır.
- 2- Hibir ama fonksiyonu iin A, B'den daha kötü olmamalıdır.

Yani, A ve B özömleri arasındaki Pareto baskınlık iliřkisi, özömlerin ok boyutlu ama uzayında nasıl sıralandığını belirler. Bu iliřki, bir özömlün diğesine göre üstünlüğünü değılendirerek, Pareto ön cephelelerini oluřturmak ve eřitli amaları optimize eden özömleri tanımlamak iin ok amalı optimizasyon algoritmalarında kullanılır. Pareto baskınlık iliřkisi, özömleri eřitli hedeflere göre karřılařtırarak, eřitli pareto optimal noktalarını belirleme ve ok amalı optimizasyon problemlerinde özöml eřitliliğini sağılama konusunda yardımcı olur.

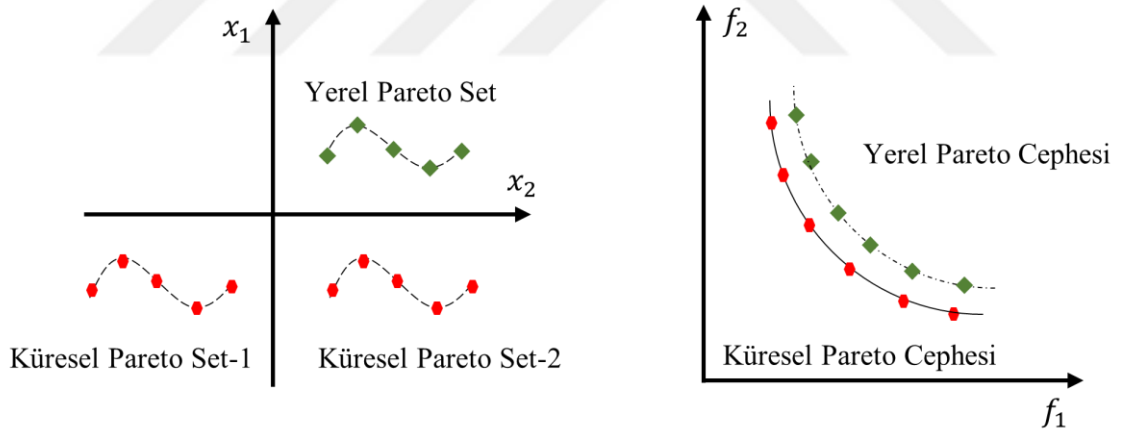
Denklem 2'de pareto baskınlığının matematiksel ifadesi verilmiştir. Farklı x ve y özöml adaylarının karřılařtırılması gösterilmiştir.

$$\begin{aligned} \vec{x} < \vec{y} &: \quad \forall i \in \{1, 2, \dots, m\}; i \leftarrow 1:m; F_i(\vec{x}) \leq F_i(\vec{y}) \wedge \exists i F_i(\vec{y}) < F_i(\vec{x}) \\ \vec{x} << \vec{y} &: \quad \forall i \in \{1, 2, \dots, m\}; F_i(\vec{x}) < F_i(\vec{y}) \end{aligned} \quad (2)$$

1.3.4. Pareto-Optimal

Pareto-optimal, birçok amaçlı optimizasyon problemi içindeki çözümler arasında bir tür üstünlük durumunu ifade eder. Bu kavram, Vilfredo Pareto'nun çalışmalarına dayanır ve çok amaçlı optimizasyon problemlerinde çeşitli amaçların birbirleriyle çeliştiği durumları ele alır (Çavuş, Sezer, & Yazıcı, 2015).

Pareto-optimal çözümler, herhangi bir hedefi diğerlerinden daha iyi yapamayan çözümlerdir. Yani, bu çözümler, herhangi bir amaç fonksiyonunu artırırken diğer amaç fonksiyonlarında bir kötüleşmeye neden olmazlar. Pareto-optimal çözümler, birbirine göre üstün olmayan ve bu nedenle çözümler arasında bir seçim yaparken kişisel tercihlere veya uygulama bağlamına bağlı olarak seçilebilen bir ön cephe oluşturur. Şekil 2'de çok modlu problemlerin gösterimi bulunmaktadır. (Liang, Suganthan, Qu, Gong, & Yue, 2019; Zhou, Zhang, & Jin, 2009; Zitzler, Thiele, Laumanns, Fonseca, & Fonseca, 2003; Yue, Qu, Yu, Liang, & Li, 2019). Aynı amaç uzayına karşılık gelen birden fazla küresel pareto set bulunmaktadır. Bu sebepten dolayı bu problemlere çok modlu problem denilmektedir.



Şekil 2. PS ve PC Gösterimleri

Pareto-optimal çözümler genellikle "Pareto ön cephe" olarak adlandırılır. Bu ön cephe, her bir Pareto-optimal çözümün bir noktasını içerir ve bu çözümler arasında baskılanamama ilişkisini gösterir. Çözümler arasında bir denge sağlamak ve çeşitli amaçları dengeli bir şekilde optimize etmek isteyen çok amaçlı optimizasyon algoritmaları, Pareto-optimal çözümleri tespit etmek ve bu ön cepheyi oluşturmak için kullanılır. Pareto-optimal, çok

amaçlı optimizasyon problemlerinde araştırmacılara ve mühendislere çözümler arasında bir seçenek yelpazesi sunma esnekliği sağlar.

1.3.5. Pareto-Optimal Set

Pareto-optimal kümesi, birçok amaçlı optimizasyon problemindeki Pareto-optimal çözümlerin oluşturduğu küme veya topluluğu ifade eder. Bu küme, herhangi bir çözümün diğerine göre üstün olamadığı Pareto-optimal çözümleri içerir.

Pareto-optimal kümesi, çok boyutlu amaç uzayında yer alan ve çeşitli amaçları optimize eden çeşitli çözümleri içerir. Bu çözümler arasında bir denge sağlar ve çelişen hedeflere ulaşma çabasında çözümleri çeşitlendirir. Pareto-optimal kümesi genellikle bir grafik veya ön cephede temsil edilir, bu ön cephe üzerindeki her nokta bir Pareto-optimal çözümü temsil eder.

Pareto-optimal kümesi, çok amaçlı optimizasyon algoritmalarının hedeflerini başarıyla dengelediği ve çözüm çeşitliliğini sağladığı bir anahtar kavramdır. Çeşitli amaçları optimize etmek isteyen bir mühendis veya araştırmacı, Pareto-optimal kümesinden çözümler arasında uygun bir seçim yapabilir ve baskınlık ilişkisine dayalı olarak kararlar alabilir. Bu, çok amaçlı optimizasyon problemlerinin çözümünde esneklik ve anlayış sağlar.

1.3.6. Pareto-Optimal Cephe

Pareto-optimal cephesi, birçok amaçlı optimizasyon problemi içindeki Pareto-optimal çözümlerin grafiksel bir temsidir. Bu cephe, her bir çözümün diğer çözümlere göre üstün olamadığı durumları ifade eder. Pareto-optimal cephesi, çok boyutlu amaç uzayında bulunan ve çeşitli amaçları optimize etmeye yönelik çözümleri içerir.

Pareto-optimal cephesi genellikle iki amaçlı bir problemde iki boyutlu bir grafik olarak düşünülse de üç veya daha fazla hedef içeren problemler için çok boyutlu bir uzayda temsil edilir. Her bir nokta, bu cephe üzerinde bir Pareto-optimal çözümü temsil eder. Bu çözümler arasında bir baskılanamama ilişkisi bulunur, yani bir hedefi iyileştirmek, diğer hedefleri olumsuz etkileyebilir.

Pareto-optimal cephesi, mühendislik ve bilim alanlarında çok amaçlı optimizasyon problemlerine yaklaşırken, çözüm çeşitliliğini anlamak ve dengelemek için kullanılır. Çeşitli amaçları optimize etme çabasında, bir çözümü seçerken bir hedefi artırmanın diğer hedefleri

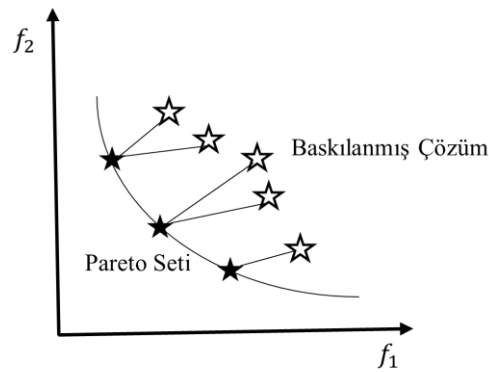
nasıl etkileyeceğini değerlendirmek için Pareto-optimal cephesi incelenir. Pareto-optimal cephesi, çok amaçlı optimizasyon algoritmalarının performansını değerlendirmek ve çeşitli baskılanamayan noktaları görselleştirmek için de kullanılır.

1.4. Performans Metrikleri

Çok amaçlı optimizasyon algoritmaları için performans metrikleri, bu algoritmaların çeşitli hedefleri ne kadar iyi optimize ettiğini değerlendirmek ve karşılaştırmak amacıyla kullanılır. Çok amaçlı optimizasyon problemleri, birden fazla amaç fonksiyonunu optimize etmeye çalışan karmaşık problemlerdir. Bu nedenle, bu tür algoritmaların performansını değerlendirmek için özel performans metrikleri kullanılır (Yue, Qu, & Liang, multi-objective particle swarm optimizer using ring topology for solving multimodal multi-objective problems, 2018; Çavuş, Sezer, & Yazıcı, 2015; Luo, ve diğerleri, 2017; Köse, 2017; Fonseca, Paquete, & Lopez-Ibanez, 2006; Liu, Liu, Li, & Li, 2019)

1.4.1. IGD_X ve IGDF

Çok amaçlı optimizasyonda, bir çözümün birden fazla hedef fonksiyonu optimize etmesi gerekir. Bu, çözümün her bir hedef fonksiyon için ne kadar iyi performans gösterdiğini ölçmek için bir ölçüm gereklidir. IGD_X ve IGDF, çok amaçlı optimizasyonda kullanılan iki yaygın ölçümdür. IGD metriğinin genel gösterimi Şekil 3'te verilmiştir (Liang, Suganthan, Qu, Gong, & Yue, 2019; Zhou, Zhang, & Jin, 2009; Zitzler, Thiele, Laumanns, Fonseca, & Fonseca, 2003; Yue, Qu, Yu, Liang, & Li, 2019).



Şekil 3. IGD metriği

IGDX (İnteraktif Çok Amaçlı Çözüm Yakınlığı), bir çözümün hedeflenen çözümden ne kadar uzakta olduğunu ölçer. Hedefli çözüm, her bir hedef fonksiyon için en iyi bilinen çözümdür.

IGDX, bir çözümün genel performansını ölçmek için kullanılır. Ancak, bir çözümün her bir hedef fonksiyon için performansını ayrı ayrı ölçmek için kullanılamaz.

IGDF (İnteraktif Çok Amaçlı Çözüm Farklılık Fonksiyonu), bir çözümün her bir hedef fonksiyon için hedeflenen çözümden ne kadar farklı olduğunu ölçer.

IGDF, IGDX'e benzer şekilde, bir çözümün genel performansını ölçmek için kullanılır. Ancak, IGDF, bir çözümün her bir hedef fonksiyon için performansını ayrı ayrı ölçmek için de kullanılabilir. Genel formülü denklem 3'te verilmiştir.

$$IGD(PF_e, PF_t) = \frac{\sum_{X \in PF_t} d(X, PF_e)}{|PF_t|} \quad (3)$$

IGDX ve IGDF, çok amaçlı optimizasyon algoritmalarında yaygın olarak kullanılan iki ölçümdür. Her iki ölçüm de bir çözümün genel performansını ölçmek için kullanılabilir.

Örnek olarak, bir fabrika, bir ürünün maliyetini ve kalitesini en üst düzeye çıkarmak isteyebilir. Bu durumda, maliyet ve kalite için iki hedef fonksiyon vardır. IGDX veya IGDF kullanılarak, bir çözümün maliyeti ve kalitesi için hedeflenen çözümden ne kadar uzakta olduğu ölçülebilir. Bu ölçümler, fabrikanın ürünlerinin performansını iyileştirmek için hangi çözümün en iyi olduğunu belirlemesine yardımcı olabilir.

1.4.2. Pareto Set Proximity

Pareto Set Proximity (PSP), bir çözümün pareto setine ne kadar yakın olduğunu ölçer. Pareto seti, tüm mümkün çözümler arasında, hiçbir diğer çözümün her iki hedef fonksiyonda da daha iyi performans göstermediği çözüm kümesidir. Benzerliği ölçmek amacıyla cover rate (CR) ve IGDX değerleri birlikte kullanılır. CR, denklem 4 ile hesaplanır. (Mohamed & Mohamed, 2019):

$$CR = \left(\prod_{i=1}^m \delta_i \right)^{1/2D} \quad (4)$$

δ_i ise i. amaç fonksiyonu için elde edilen pareto cephesi ve küresel pareto cephesindeki uç çözümlerin bulunduğu maksimum ve minimum değerleri kullanarak denklem 5'te gösterildiği gibi hesaplanmaktadır.

$$\delta_i = \left(\frac{\min(PFe_i^{max}, PFt_i^{max}) - \max(PFe_i^{min}, PFt_i^{min})}{PFt_i^{max} - PFt_i^{min}} \right)^2 \quad (5)$$

Son olarak psp hesaplamak için kullanılan denklem 6 ile gösterilmiştir.

$$PSP = \frac{CR}{IGDX} \quad (6)$$

PSP, 0 ile 1 arasında bir değer alır. 0, bir çözümün Pareto setine tamamen uzak olduğunu gösterir. 1, bir çözümün Pareto setine tamamen yakın olduğunu gösterir.

PSP, çok amaçlı optimizasyon algoritmalarında yaygın olarak kullanılan bir ölçümdür. Bu ölçüm, bir algoritmanın Pareto setine ne kadar yakın çözümler ürettiğini ölçmek için kullanılır.

PSP metriği, IGDX ve IGDF metriklerine benzer şekilde, bir çözümün genel performansını ölçmek için kullanılır. Ancak, PSP metriği, bir çözümün pareto setine ne kadar yakın olduğunu ölçerek, bir çözümün her bir hedef fonksiyon için performansını ayrı ayrı ölçmeyi de sağlar.

1.4.3. Hypervolume

Hypervolume (HV), çok amaçlı optimizasyon problemlerinde performansı ölçmek için kullanılan bir metriktir. Genellikle pareto ön cephe ile ilgili bir çözüm kümesinin kalitesini değerlendirmek için kullanılır.

Hypervolume, bir çözüm kümesinin pareto ön cephesi tarafından kapsanan hacmi ölçer. Bir çözüm kümesinin pareto ön cephesi, bir tür ideal çözüm kümesini temsil eder. Hypervolume metriği, bu ideal çözüm kümesine göre bir çözüm kümesinin kapsadığı hacmi değerlendirir (Fonseca, Paquete, & Lopez-Ibanez, 2006; Beume, Fonseca, Lopez-Ibanez, Paquete, & Vahrenhold, 2009).

Hypervolume hesaplanması aşağıda anlatılmıştır.

İdeal Noktaların Belirlenmesi: İdeal noktalar, her bir amaç fonksiyonu maksimize edildiğinde elde edilebilecek en iyi değerlerdir.

Pareto-optimal Çözümlerin Bulunması: Çözüm kümesi içindeki pareto-optimal çözümler belirlenir. Yani, bu çözümler herhangi bir hedefi diğerine göre iyileştiremeyen çözümlerdir.

Hypervolume Hesaplaması: Her bir pareto-optimal çözüm ile ideal noktalar arasındaki hacim hesaplanır ve bu hacimler toplanarak hypervolume elde edilir.

Hypervolume metriği genellikle bir çözüm kümesinin çeşitliliği ve çözümlerin Pareto ön cephe içindeki dağılımını değerlendirmek için kullanılır. Daha yüksek bir hypervolume değeri, daha iyi bir çözüm kümesini temsil eder. HV metriğinin hesaplamasını için denklem 7 kullanılır.

$$HV(S, w) = \lambda_D(\cup_{z \in S} [z; w]) \quad (7)$$

Hypervolume metriği, çok amaçlı optimizasyon algoritmalarının performansını karşılaştırmak ve çözüm çeşitliliğini ölçmek için sıklıkla kullanılır.

1.5. Uyarlanabilir Rehberlik Mekanizmalı Diferansiyel Evrim Algoritması

Uyarlanabilir rehberlik mekanizmalı diferansiyel evrim (URDE) algoritması, evrimsel hesaplama alanında kullanılan bir optimizasyon algoritması türüdür. URDE, Diferansiyel Evrim (DE) algoritmasının bir türevidir ve popülasyon tabanlı birçok amaçlı optimizasyon yaklaşımıdır. Uyarlanabilir ve rehberlik mekanizmalı kelimeleri, algoritmanın davranışının problem bağlamına uyum sağlayabilen ve çözüm alanında daha etkili bir arama gerçekleştirmeye çalışan özellikleri ifade eder.

URDE algoritmasının temel özellikleri:

DE Algoritması Temeli: URDE, temelde diferansiyel evrim algoritmasına dayanır. DE, popülasyon içindeki bireyler arasında vektörel farklar kullanarak yeni çözümler üretir.

Adaptasyon Mekanizması: URDE, popülasyonun evrimi sürecinde adaptasyon mekanizmalarını kullanır. Bu, algoritmanın parametrelerinin (örneğin, mutasyon ve

çaprazlama olasılıkları) problem bağlamına bağlı olarak otomatik olarak ayarlanabilmesini sağlar.

Rehberlik Mekanizması: URDE, bir tür rehberlik mekanizması kullanır. Bu, algoritmanın çözüm alanında daha etkili bir arama gerçekleştirmesine yardımcı olacak bilgiyi kullanmasını sağlar. Bu rehberlik mekanizması, önceki iterasyonlardan elde edilen bilgileri içerebilir veya problem bağlamına özgü olarak tasarlanmış bir rehberlik stratejisi kullanabilir.

URDE algoritmasının sözde kodu algoritma Algoritma 1’de gösterilmektedir (Duman, Akbel, & Kahraman, 2021) .

Algoritma 1. DE ve URDE algoritmalarının temel adımları	
1.	Başla
2.	P : rastgele popülasyon oluştur
3.	for $i=1: n$ (n sayısı kadar çözüm adayı üretilir)
4.	Her P çözüm adayı için fitness değeri hesaplanır
5.	end
6.	while (arama süresi yaşam döngüsü)
7.	Seçim süreci: r_1, r_2, r_3
8.	Arama operatörleri:
9.	Çaprazlama
10.	Mutasyon
11.	Güncelleme süreci: Tüm çözüm adaylarının fitness değerleri birbirleri ile kıyaslanır.
12.	sıradaki iterasyon ile devam eder eğer sonlandırma kriteri gerçekleşmediyse
13.	Bitir

1.5.1. Popülasyon Oluşturma Evresi

URDE algoritmasında popülasyon oluşturma evresi, algoritmanın başlangıcında veya bir iterasyonun başlangıcında gerçekleşir. Bu evre, çözüm uzayında rastgele veya belirli bir stratejiye göre seçilen çözüm adaylarından oluşan bir popülasyonun oluşturulmasını içerir. URDE algoritmasının popülasyon oluşturma evresinde genel adımlar:

Başlangıç Noktalarının Belirlenmesi: URDE’de, başlangıç noktaları genellikle rastgele seçilir. Başlangıç noktaları, çözüm uzayındaki potansiyel çözümleri temsil eder.

Popülasyon Boyutunun Belirlenmesi: Kaç bireyden oluşacaksa popülasyon boyutu belirlenir. Bu, genellikle kullanıcı tarafından belirlenen bir parametredir.

Başlangıç Noktalarının Oluşturulması: Belirlenen başlangıç noktaları kullanılarak popülasyon oluşturulur. Başlangıç noktaları genellikle rastgele seçilir ve her bir birey, çözüm uzayındaki bir noktayı temsil eder.

Özellikle URDE’de, popülasyon oluşturma aşamasında çeşitli başlangıç stratejileri kullanılabilir. Bu stratejiler, popülasyonun çeşitliliğini artırmak ve çözüm uzayında daha geniş bir alanı keşfetmek için tasarlanmış olabilir. Denklem 8’de çözüm adaylarının uygunluk değerleri (UD) hesaplanma formülü verilmiştir. n adet tasarım değişkeni ve m adet çözüm adayından oluşmaktadır.

$$P \equiv [x_{11} \cdots x_{1n} \vdots \vdots x_{m1} \cdots x_{mn}], \quad UD \equiv [f_1 \vdots f_m] \quad (8)$$

Çözüm adaylarının popülasyondaki uygunluk değeri ile en yüksek çözüm adayına mesafesi hesaplanır. Bu mesafeyi hesaplamak amacıyla öklid kullanılmaktadır. Denklem 9 kullanarak hesaplanır.

$$i = 1m \forall P_i, D_{P_i} = \sqrt{(x_{1[i]} - x_{1[best]})^2 + (x_{2[i]} - x_{2[best]})^2 + \cdots + (x_{n[i]} - x_{n[best]})^2} \quad (9)$$

Mesafe skorları ve uygunluk değerleri kullanılarak çözüm adaylarının skorları hesaplanmaktadır. Denklem 10 UD ve D_p kullanılarak skor hesabının matematiksel denklemi verilmiştir.

$$i = 1m \forall P_i, \quad S_{P_i} = w * normFV_i + (1 - w) * normD_{P_i} \quad (10)$$

Popülasyon oluşturma evresi, URDE’nin evrim sürecinin başlangıcını oluşturur ve çözüm uzayında etkili bir arama başlatılmasına katkıda bulunur. Bu aşamada iyi bir

başlangıç popülasyonu oluşturmak, algoritmanın daha hızlı ve daha etkili bir şekilde çözümler üretmesine yardımcı olabilir.

1.5.2. Rehber Seçim Mekanizması Tasarımı

URDE algoritması, çok amaçlı optimizasyon problemlerini çözmek için kullanılan bir genetik algoritmadır. Rehber seçim mekanizması, bir çözümün yeni bir çözüm üretmek için kullanılacak rehberini seçmek için kullanılır.

Seçilen rehber, mevcut çözümle birlikte kullanılır. Rehber ve mevcut çözüm, bir genetik operatör kullanılarak yeni bir çözüm üretmek için kullanılır.

DE algoritmasında kullanılan referans vektörleri (r_1, r_2, r_3) P-popülasyonundan rastgele seçilir ($r_1 \neq r_2 \neq r_3$). URDE algoritmasında ise r_1 vektörü P-popülasyonundan rastgele seçilir, ancak r_2 ve r_3 vektörleri aç gözlü bir yöntemle belirlenir.

Aç gözlü seçim yöntemi şu şekilde işler: P-popülasyonundaki çözüm adayları, uygunluk değerlerine göre sıralanır ve en iyi aday P_{best} olarak seçilir, ardından r_2 olarak atanır. Aynı şekilde, P-popülasyonundaki çözüm adaylarından uygunluk değeri en kötü olan aday P_{worst} olarak seçilir ve r_3 olarak atanır (Mohamed & Mohamed, 2019; Duman, Akbel, & Kahraman, 2021).

URDE algoritmasında kullanılan rehber seçim mekanizması, algoritmanın verimliliğini ve performansını iyileştirmek için tasarlanmıştır. Bu mekanizma, algoritmanın pareto setine daha yakın çözümler üretmesine ve daha geniş bir çözüm alanı keşfetmesine yardımcı olur.

1.5.3. Komşuluk Araması ve Çeşitlilik

URDE algoritmasının diferansiyel evrim algoritmasından ayrıldığı ikinci aşama, çaprazlama operatörünün işleyiş biçimidir. Çaprazlama operatörünün arama sürecindeki etkisi, CR kontrol parametresi kullanılarak ayarlanır. CR'nin değeri değiştirilerek çeşitlilik-komşuluk araması arasında geçiş yapılır. Büyük CR değerleri, algoritmanın keşfi artırmasına yönelikken, aynı zamanda kullanılan çözümleri sınırlar. CR'nin değerinin düşürülmesi, hassas bir komşuluk aramasını destekler ancak çeşitliliği azaltabilir. Bu durum, CR parametresinin etkili bir şekilde ayarlanmasının önemini vurgular. Denklem 11'de,

çaprazlama oranı iterasyon sayısına ve p_1 ve p_2 olasılıklarına bağlı olarak dinamik olarak ayarlanmaktadır.

$$\begin{aligned} \{if\ G = 1, \quad CR_i^1 &= \{CR_1, if\ rand(0,1) \leq 0.5\ CR_2, \\ else, CR_i^G &= \{CR_1, if\ rand(0,1) \leq p_1\ CR_2, \\ if\ p_1 &\leq rand(0,1) \leq p_1 + p_2 \end{aligned} \quad (11)$$

URDE algoritmasında, CR parametresi arama sürecinde denklem 12’de görüldüğü gibi dinamik olarak adapte edilir bu da algoritmanın çeşitli görevlere uyum sağlama yeteneğini artırır (Mohamed & Mohamed, 2019).

URDE algoritmasında her arama yaşam döngüsünde, her amaç vektörü mutasyona uğrayacaktır denklem 12 ile URDE algoritmasında kullanılan mutant denklemi verilmiştir.

$$v_i^G = X_{r1}^G + F \cdot (X_{r2}^G - X_{r3}^G) \quad (12)$$

1.5.4. Güncelleme Mekanizması Tasarımı

Yeni oluşturulan çözüm adayları, mevcut popülasyondaki uygun adaylarla karşılaştırılır. Eğer yeni çözüm adayları daha iyi bir uygunluk değerine sahipse, popülasyon güncellenir. Aç gözlü bir seçim yöntemi kullanılarak, en iyi ve en kötü çözümler arasında karşılaştırma yapılır ve uygun olanlar popülasyona dahil edilir (Mohamed & Mohamed, 2019).

1.6. Arşiv İndirgeme

Çok amaçlı optimizasyon algoritmalarındaki arşiv indirgeme, genellikle çözüm kümesindeki çeşitliliği artırmak ve önemli çözümleri korumak amacıyla kullanılan bir stratejidir. Bu strateji, algoritmanın bulduğu çözüm kümesini, çeşitli ve önemli çözümleri içerecek şekilde düzenleme ve sınırlama işlemidir. Arşiv indirgeme, özellikle çok amaçlı optimizasyon problemlerinde kullanılır (Zitzler, Deb, & Thiele, 2001).

Arşiv indirgeme şu temel prensiplere dayanır:

Çeşitlilik Koruma: Çözüm kümesinde çeşitliliği artırmak için arşivde farklı ve çeşitli çözümleri tutar. Bu, algoritmanın yakınsamayı artırabilir ve daha geniş bir çözüm uzayını keşfetmesine olanak tanır.

İlgi Çözümlerin Korunması: Arşiv, önceki iterasyonlarda elde edilen önemli veya ilginç çözümleri tutar. Bu, algoritmanın daha önce keşfedilen başarılı çözümleri unutmamasını ve bu çözümleri geliştirmesini sağlar.

Arşiv Boyutu Kontrolü: Arşiv, sınırlı bir boyutta tutularak gereksiz çözümlerin elenmesiyle kontrol edilir. Bu, algoritmanın hafızasının sınırlı kalmasını sağlar ve kaynakları etkili bir şekilde kullanmasına yardımcı olur.

Bu strateji, algoritmanın optimizasyon sürecinde, çözüm kümesini sadece en iyi ve çeşitli çözümlerle güncellemesine olanak tanır. Bu şekilde, algoritma daha etkili bir şekilde çözüm uzayını keşfeder ve problemi çözer.

1.6.1. Klasik Kalabalık Mesafesi Yöntemi

Çok amaçlı optimizasyonda kullanılan klasik kalabalık mesafe yöntemi, çözüm uzayındaki çeşitliliği ve çözümlerin dağılımını artırmak için kullanılan bir stratejidir. Bu yöntem, birçok algoritma tarafından benimsenmiş ve özellikle çözüm kümesinin Pareto önünde homojen bir dağılımını sağlamak için etkili bir araç olarak kullanılmıştır.

Kalabalık mesafe yöntemi, çözümlerin birbirine olan "kalabalıklık" veya "yoğunluk" durumunu ölçerek, çözümleri birbirinden ayıran bir ölçüdür. Temel prensip, çözümlerin birbirlerine ne kadar yakın olduğunu değerlendirerek yoğun bölgelerdeki çözümleri ölçmektir. Bu yöntem, çözüm kümesindeki yoğun bölgelerde daha fazla çözüm bulunmasını teşvik eder ve bu sayede çözüm kümesinin çeşitliliği artar (Sushmita Sharma, Gharehchopogh, & Mirjalili, 2022).

Kalabalık mesafe yönteminin temel adımları şunlardır:

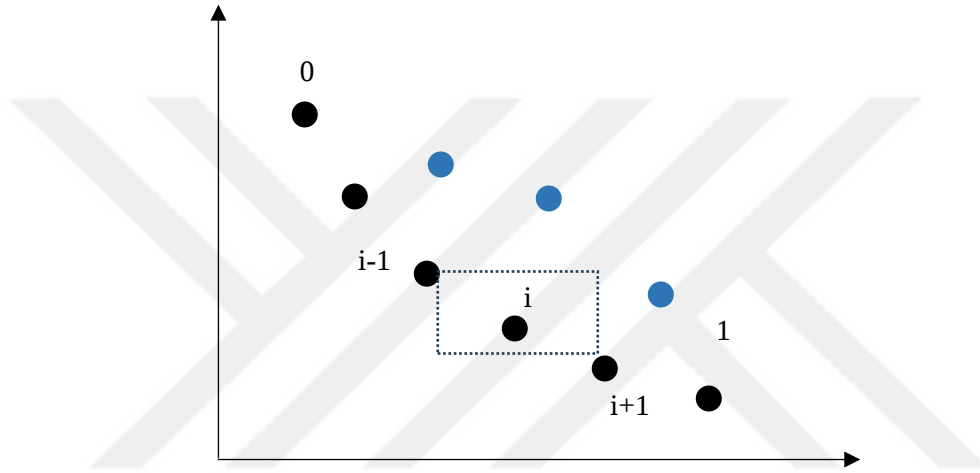
Mesafe Hesaplama: Çözümler arasındaki mesafeler genellikle öklid formülü kullanılarak hesaplanır. Ancak, problem bağlamına bağlı olarak farklı mesafe metrikleri de kullanılabilir.

Kalabalık Mesafe Hesaplama: Çözümlerin birbirine olan kalabalık mesafeleri, genellikle her çözümün çevresindeki diğer çözümlere olan uzaklıkların ortalaması veya bir tür normalleştirilmiş mesafe hesaplamasıyla belirlenir (Sağ, 2015).

Sıralama: Çözümler, kalabalık mesafelerine göre sıralanır. Daha düşük kalabalık mesafesine sahip çözümler, daha yoğun bölgelerde yer almaktadır.

Seçim: Kalabalık mesafe sıralamasına göre seçim yapılır. Genellikle, daha düşük kalabalık mesafesine sahip çözümler, çeşitliliği artırmak için tercih edilir.

Kalabalık mesafesinin hesaplanmasının görselleştirilmiş halini Şekil 4’de görebilirsiniz (Sağ, 2015).



Şekil 4. Kalabalık mesafe hesaplaması

Bu yöntem, çok amaçlı optimizasyon problemlerinde popülasyonun homojen bir şekilde dağılmasına yardımcı olabilir ve Pareto önünde geniş bir çeşitlilik sağlayarak çözüm kümesini iyileştirebilir.

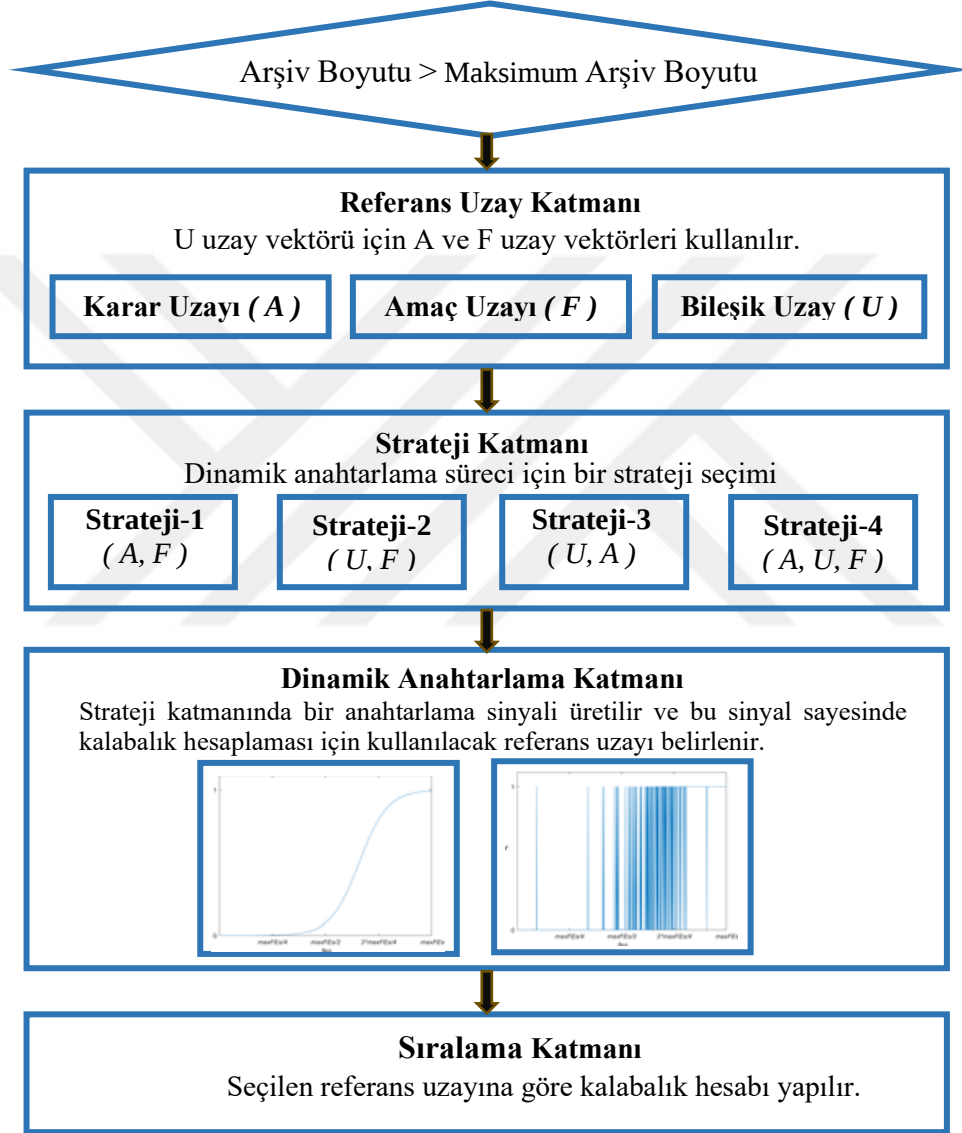
Çok amaçlı uyarlanabilir rehberlik mekanizmalı diferansiyel evrim (ÇAURDE) algoritmasında kullanılan genel yöntem olan kalabalık mesafe yönteminin sözde kodu Algoritma 2’de anlatılmıştır (Kahraman, Akbel, Duman, Kati, & Sayan, 2022).

Algoritma 2. ÇAURDE Algoritmasında Kullanılan Kalabalık Mesafe Yönteminin Sözde Kodu	
Parametreler : ruv (referans uzay vektörü), arşiv_boyut (arşiv maksimum boyutu), r (mesafe limiti), m (uzay amaç fonksiyon sayısı)	
Çıktı : sıralamalar	
1.	Başla
2.	for (i=1:size_of(ruv))
3.	sıralamalar(i)=0;
4.	for (j=1:size_of(ruv))
5.	durum=false;
6.	if ($\forall_{k=1}^m \text{mesafe}(RSV_{[j,k]}, RSV_{[i,k]} < r_k)$) then durum=true; end if
7.	end for
8.	if (durum) then sıralamalar(i)= sıralamalar (i)+1; end if
9.	end for
10.	return sıralamalar
11.	Bitir

1.6.2. Dinamik Anahtarlama Kalabalık Yöntemi

Kalabalık mesafesi hesaplaması, sıralama ve arşiv indirgeme işlemlerinde gerçekleşmektedir. Arşiv boyutu aşılsa arşiv indirgeme süreci başlar. Sıralama sürecinde, çözüm adayları etrafındaki yoğunluk, kalabalık mesafesi kullanılarak hesaplanır. Bu süreçte arşiv, referans olarak alınan uzayda (karar uzayı veya amaç uzayı) yoğunluğa göre azaltılır. Ancak, kalabalık mesafesi hesaplamasında referans alınan uzay hangisi olursa olsun, seçilen uzayda çeşitlilik sağlanır. Referans alınmayan uzayda, yani ne karar uzayında ne de amaç uzayında çeşitliliği sağlamak risklidir, çünkü bu durum yerel çözüm tuzağına düşmeye yol açabilir. Bu nedenle, çeşitliliği yalnızca bir referans uzayında değil, her iki uzayda da korumak önemlidir. Ayrıca, arama süreci boyunca zaman zaman her iki uzayda da çeşitliliğe ihtiyaç duyulabilir. Literatürdeki çalışmalar, birçok problemin, en etkili algoritmaların bile PFe ve PSe kümelerini PFT ve PSt kümeleriyle uygun bir şekilde örtüştüremediğini göstermiştir. Bu durum, arşiv işleme sürecinin tamamen gözden geçirilmesini ve daha etkili yöntemler kullanılarak tasarlanmasını gerektirir. Sonuç olarak, sıralama sürecinde

tanımlanan kalabalık mesafesi hesaplama yöntemini güçlendirmek hem karar hem de amaç uzaylarında çeşitliliği korumak ve her iki uzayda da dengeli bir arama yapabilen yöntemleri geliştirmek önemlidir. Bu sebepten dolayı dinamik anahtarlama kalabalık yöntemi geliştirilmiştir (Kahraman, Akbel, Duman, Kati, & Sayan, 2022).



Şekil 5. Dinamik Anahtarlama Kalabalık Mesafesi

Arama süreci yaşam döngüsünün herhangi bir anındaki uygunluk değerlendirme numarasının güncel değeri, bir referans sinyali ve rasgele bir sayı kullanılarak, geçiş süreci dinamik bir şekilde gerçekleştirilir.

Şekil 5. Dinamik Anahtarlama Kalabalık Mesafesi dinamik anahtarlama kalabalık mesafe yönteminin akış diyagramı verilmiştir (Kahraman, Akbel, Duman, Kati, & Sayan, 2022).

Şekil 5’te belirtildiği gibi bir referans sinyal ile kullanılacak referans uzay belirlenir. Belirlenen uzay ile sıralama işlemi yapılır. Algoritma 3’te dinamik anahtarlama kalabalık mesafe yönteminin sözde kodu bulunmaktadır (Kahraman, Akbel, Duman, Kati, & Sayan, 2022).

Algoritma 3. Dinamik Anahtarlama Kalabalık Yönteminin Sözde Kodu	
Parametreler: strateji, referans sinyal, fes, A, F _A , U	
Çıktı: sıralamalar	
1.	Başla
2.	<i>ref = referans_sinyal(fes)</i>
3.	<i>Kalabalık hesaplama ve sıralama süreci</i>
4.	<i>if</i> (“strateji-1” veya “strateji-2” veya “strateji-3”)
5.	<i>if</i> (rand()<ref) then Algoritma 2 çalıştır (1. ruv) return { sıralamalar }
6.	<i>else then</i> Algoritma 2 çalıştır (2. ruv) return { sıralamalar } end if
7.	<i>end if</i>
8.	<i>if</i> (“strateji-4”)
9.	<i>if</i> (fes<maxFes/2)
10.	<i>if</i> (rand()<ref) then Algoritma 2 çalıştır (1. ruv) return { sıralamalar }
11.	<i>else then</i> Algoritma 2 çalıştır (2. ruv) return { sıralamalar } end if
12.	<i>else</i>
13.	<i>if</i> (rand()<ref) then Algoritma 2 çalıştır (2. ruv) return {sıralamalar}
14.	<i>else then</i> Algoritma 2 çalıştır (3. ruv) return { sıralamalar } end if
15.	<i>end if</i>
16.	<i>end if</i>
17.	Bitir

Meta-sezgisel optimizasyon algoritmalarının stokastik yapısı, rastgele sayı dizilerini yaygın bir şekilde kullanmalarına neden olmaktadır. Bu alanda sıkça tercih edilen rastgele üretilen sayı dizilerinin dağılımları, algoritmaların global optimuma yakınsama yeteneklerini belirlemede önemli bir etkiye sahiptir. Bu bağlamda, kaotik haritaların rastgele sayı üretici olarak kullanılması, meta-sezgisel optimizasyon algoritmalarının performansını artırabilir (Kahraman, Akbel, Duman, Kati, & Sayan, Unified space approach-based Dynamic Switched Crowding (DSC): A new method for designing Pareto-based multi/many-objective algorithms, 2022).

Kaotik haritalar, algoritmaların performansını artırma konusunda oldukça etkili stratejilerden biridir. Bu haritalar hem yerel optimumlardan kaçınma hem de yakınsama hızını optimize etme açısından başarılı sonuçlar sağlar. Sezgisel algoritmaların performansını geliştirmek amacıyla yapılan çalışmalarda, kaotik haritaların kullanımı literatürde önemli bir yer tutmaktadır. Örneğin, Kaotik Arı Kolonisi Algoritmaları (Toktas, Erkan, Ustun, & Wang, 2023; Li & Yin, 2014), Kaotik Haritalı Parçacık Sürü Optimizasyon Algoritmaları (Xiang, Liao, & Wong, 2007) Kaotik Haritalı Hibrit Parçacık Sürü Optimizasyon Algoritması (Xu, Geng, Zhu, & Gu, 2013), Kaotik Ateş Böceği Algoritması (Gandomi, Yang, Talatahari, & Alavi, 2013), Kaotik Gri Kurt Optimizasyon Algoritması (Kohli & Arora, 2018), Kaotik Big-Bang Optimizasyonu (Alatas, 2011), Kaotik Armoni Arama Algoritmaları (Alatas, Chaotic harmony search algorithms, 2010), Kaotik Haritalı Balina Optimizasyon Algoritması (Kaur & Arora, 2018) gibi çalışmalar, kaotik haritaların kullanılmasının algoritmaların performansını artırmada başarılı bir yol olduğunu göstermektedir.

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

Lojistik Harita: Lojistik harita, basitlik ve yaygınlığıyla bilinen önemli matematiksel yapılarından biridir. Çoğunlukla ayrık zamanlı dinamik sistemlerde kullanılan bu harita, tek boyutlu olup doğrusal olmayan özelliklere sahiptir. Lojistik haritanın matematiksel ifadesi, denklem 13'te açıklanmıştır. Bu harita, genellikle basitlik ve etkili sonuçları nedeniyle birçok uygulama alanında tercih edilmektedir.

$$X_{n+1} = a X_n(1 - X_n) \quad (13)$$

Çadır Harita: Çadır haritası, Lojistik haritaya benzer özellikleri ile tanınmaktadır. Matematiksel ifadesi denklem 14'te sunulmaktadır. Bu harita, Lojistik haritaya benzer şekilde dinamik ve karmaşık sayı dizileri üretebilen bir yapıya sahiptir ve bu özellikleriyle birçok uygulama alanında ilgi çekmektedir.

$$X_{n+1} = \begin{cases} \frac{X_n}{0.7} & X_n < 0.7 \\ \frac{X_n}{p} (1 - X_n) & 0.7 \leq X_n \end{cases} \quad (14)$$

Çember Harita: Andrey Kolmogorov tarafından ilk geliştirilen Çember harita denklemi, aynı zamanda elektronikteki faz kilitlemeli döngü denklemini de ifade etmektedir (Tavazoei & Haeri, 2007). Çember haritanın matematiksel formülü, denklem 15'te sunulmuştur. Bu denklem hem matematikte hem de elektronik sistemlerde kullanılan bir model olarak önem taşımaktadır ve özellikle faz kilitlemeli döngü konseptini açıklamak için yaygın bir şekilde kullanılmaktadır.

$$X_{n+1} = X_n + b - \left(\frac{a}{2\pi}\right) \sin \sin (2\pi X_n) \bmod(1) \quad (15)$$

Chebyshev Harita: Matematikte, Pafnuty Chebyshev'in adını taşıyan Chebyshev polinomları, Moivre formülü ile ilişkilendirilebilen ve iteratif bir şekilde tanımlanabilen ortogonal polinomlar dizisidir (Chatterjee, ve diğerleri, 2016). Chebyshev haritasının matematiksel ifadesi denklem 16'da sunulmuştur. Bu harita, Chebyshev polinomlarının benzersiz özelliklerini ve iteratif yapısını vurgulayarak matematiksel modelleme ve sayısal analiz bağlamında önemli bir rol oynamaktadır.

$$X_{n+1} = \cos \cos (n * (X_n)) \quad (16)$$

Gaus Harita: Gauss haritasının matematiksel formülü, denklem 17'de ifade edilmiştir. Gauss haritası, matematikte önemli bir yere sahip olan bir modeldir ve bu ifade, haritanın

iteratif özelliklerini vurgulayarak sayısal analiz ve matematiksel modelleme bağlamında kullanışlı bir araçtır.

$$X_{n+1} = \begin{cases} 0 & X_n = 0 \\ \frac{1}{X_n} \bmod(1) & X_n \in (0,1) \end{cases} \quad (17)$$

Sinüs Harita: Sinüs haritasının matematiksel ifadesi, denklem 18'de tanımlanmıştır. Bu denklem, sinüs haritasının matematiksel modelini açıklar. Sinüs haritası, matematik ve fizikte geniş bir uygulama alanına sahip olan ve sinüs fonksiyonuyla ilişkilendirilen bir harita türüdür. Denklem 18, sinüs haritasının karakteristik özelliklerini ifade eder ve bu ifade, haritanın matematiksel yapısını anlamak ve uygulamak için bir temel sağlar.

$$X_{n+1} = \left\{ \frac{a}{4} \sin \sin (\pi X_n) \right\} \quad 0 < a \leq 4 \quad (18)$$

Sinüsoidal Harita: Sinüsoidal haritanın matematiksel formülü, denklem 19'da belirtilmiştir. Bu denklem, sinüsoidal haritanın matematiksel modelini tanımlar. Sinüsoidal haritalar genellikle matematiksel analizlerde, sinyal işlemede ve diğer uygulamalarda kullanılan bir türdür. Denklem 19, sinüsoidal haritanın özelliklerini ve davranışını açıklar, bu da haritanın nasıl çalıştığını ve ne tür sonuçlar üretebileceğini anlamak için bir referans noktası sağlar.

$$X_{n+1} = aX_n^2 \sin \sin (\pi X_n) \quad (19)$$

Singer Harita: Singer haritasının matematiksel formülü, denklem 20'de ifade edilmiştir. Bu denklem, Singer haritasının matematiksel modelini temsil eder. Singer haritası, matematiksel dinamik sistemlerde ve karmaşık sayı teorisinde kullanılan bir türdür. Denklem 20, Singer haritasının benzersiz özelliklerini ve davranışını anlatarak, bu haritanın nasıl çalıştığını ve matematiksel analizlerde nasıl kullanılabileceğini açıklar.

$$X_{n+1} = \mu(7.86X_n - 23.31X_n^2 + 28.75X_n^3 - 13.302875X_n^4) \quad (20)$$

Parçalı Harita: Parçalı haritanın matematiksel formülü, denklem 21'de sunulmuştur. Bu denklem, parçalı haritanın matematiksel modelini ifade eder. Parçalı haritalar, karmaşık dinamik sistemlerde ve kaotik yapıların incelenmesinde kullanılan önemli matematiksel araçlardır. Denklem 21, parçalı haritanın karakteristik özelliklerini ve davranışını açıklar, bu da matematiksel analizlerde ve uygulamalarda bu haritanın nasıl kullanılabileceğine dair bir anlayış sağlar.

$$\begin{aligned}
 X_{n+1} = & \begin{cases} \frac{X_n}{P} & 0 \leq X_n < P \\ \frac{X_n - P}{0.5 - P} & P \leq X_n \\ < 0.5 \end{cases} \quad \frac{1 - P - X_n}{0.5 - P} \quad 0.5 \leq X_n \\
 & < 1 - P \quad \frac{1 - X_n}{P} \quad 1 - P \leq X_n < 1
 \end{aligned} \tag{21}$$

İteratif Harita: İteratif haritanın matematiksel formülü, denklem 22'de açıklanmıştır. Bu denklem, iteratif haritanın matematiksel modelini ifade eder. İteratif haritalar, matematikte ve bilgisayar bilimlerinde geniş bir uygulama alanına sahip olan ve ardışık iterasyonlarla belirli bir mantıksal kurala göre değerler üreten dinamik sistemlerdir. Denklem 22, iteratif haritanın özelliklerini ve davranışını anlatarak, bu haritanın nasıl çalıştığını ve matematiksel analizlerde nasıl kullanılabileceğini açıklar.

$$X_{n+1} = \sin \sin \left(\frac{a * \pi}{X_n} \right) \tag{22}$$

1.7. K-Ortalamalar Kümeleme Yöntemi

K-ortalama algoritması, denetimsiz öğrenmenin yaygın bir yöntemidir. Bu algoritma, n adet gözlem içeren bir veri kümesini, önceden belirlenen k adet kümeye ayırmayı hedefler. Her bir gözlem sadece bir kümeye atanır (Belhor, El-Amraoui, Jemai, & Delmotte, 2023). Temel amaç, algoritmanın uygulanması sonucunda elde edilen kümelerin, birbirlerine olan benzerliklerini maksimize etmek ve kümeler arasındaki benzerlikleri minimize etmektir (Jain, 2010; Koşuta, 2018).

K-Ortalamalar algoritması, ilk aşamada rastgele seçilen k adet merkez noktası ile başlar. Her bir veri noktası, kendisine en yakın olan küme merkezine atanır. Ardından, her

küme için merkez noktalarının değeri, o kümeye dahil edilen noktaların aritmetik ortalaması alınarak hesaplanır. Bu süreç, küme merkezlerinin değerleri değişmeyinceye kadar tekrarlanır (Kanungo, ve diğerleri, 2002; Koşuta, 2018).

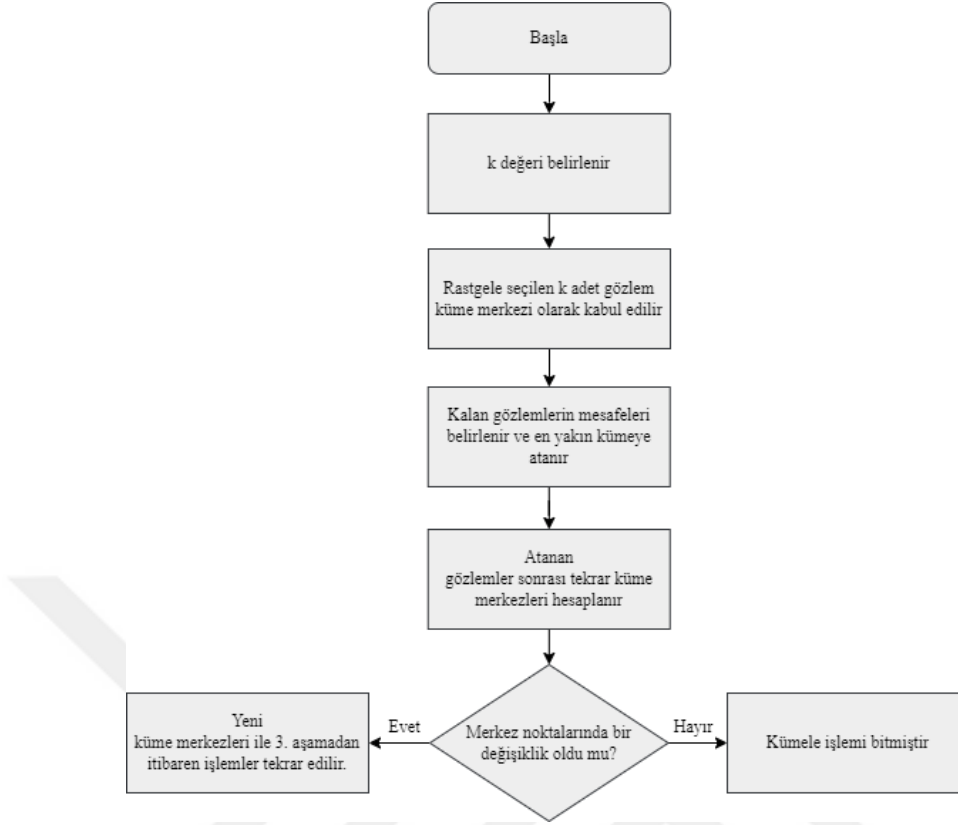
K-Ortalamlar algoritmasının uygulanması için aşağıdaki 5 adımı izlemek gerekmektedir (Ray & Turi, 1999; Koşuta, 2018):

- 1- İlk olarak, önceden belirlenmiş olan kümelerin sayısı olan k değeri seçilir.
- 2- Veri kümesinden rastgele seçilen k adet gözlem, küme merkezleri olarak kabul edilir.
- 3- Kalan gözlemlerin her biri için, en yakın olduğu küme merkezi belirlenir ve ilgili gözlem o kümeye atanır.
- 4- Her kümeye atanmış olan gözlemlerle birlikte, yeni küme merkezleri hesaplanır.
- 5- Eğer yeni küme merkezleri, bir önceki merkez noktaları ile aynı ise, kümeleme tamamlanır.
- 6- Hesaplanan yeni merkez noktaları ile 3. adımdan itibaren işlemler tekrarlanır.

Yukarıdaki adımları anlatılan k-ortalamlar algoritmasının sözde kodu Algoritma 4'te verilmiştir (Haraty, Dimishkieh, & Masud, 2015).

Algoritma 4. K-Ortalamlar Sözde Kodu	
	Parametreler: k (küme sayısı), d (veri seti içerisinde n nesne)
	Çıktı: k adet küme
1.	Başla
2.	D'den k nesneyi rastgele seçerek başlangıç küme merkezlerini belirle
3.	tekrarla
4.	Her nesneyi, nesnenin bulunduğu kümenin ortalamasına dayanarak en benzer olduğu kümeye (yeniden) atayın
5.	Küme ortalamalarını güncelle, yani her küme için nesnelerin ortalamasını hesapla
6.	Değişiklik olmadığı duruma kadar tekrarla
7.	Bitir

K-Ortalamlar algoritmasının akış diyagramı Şekil 6'te gösterilmiştir (Koşuta, 2018).



Şekil 6. K-ortalamlar algoritma şeması

2. YAPILAN ÇALIŞMALAR

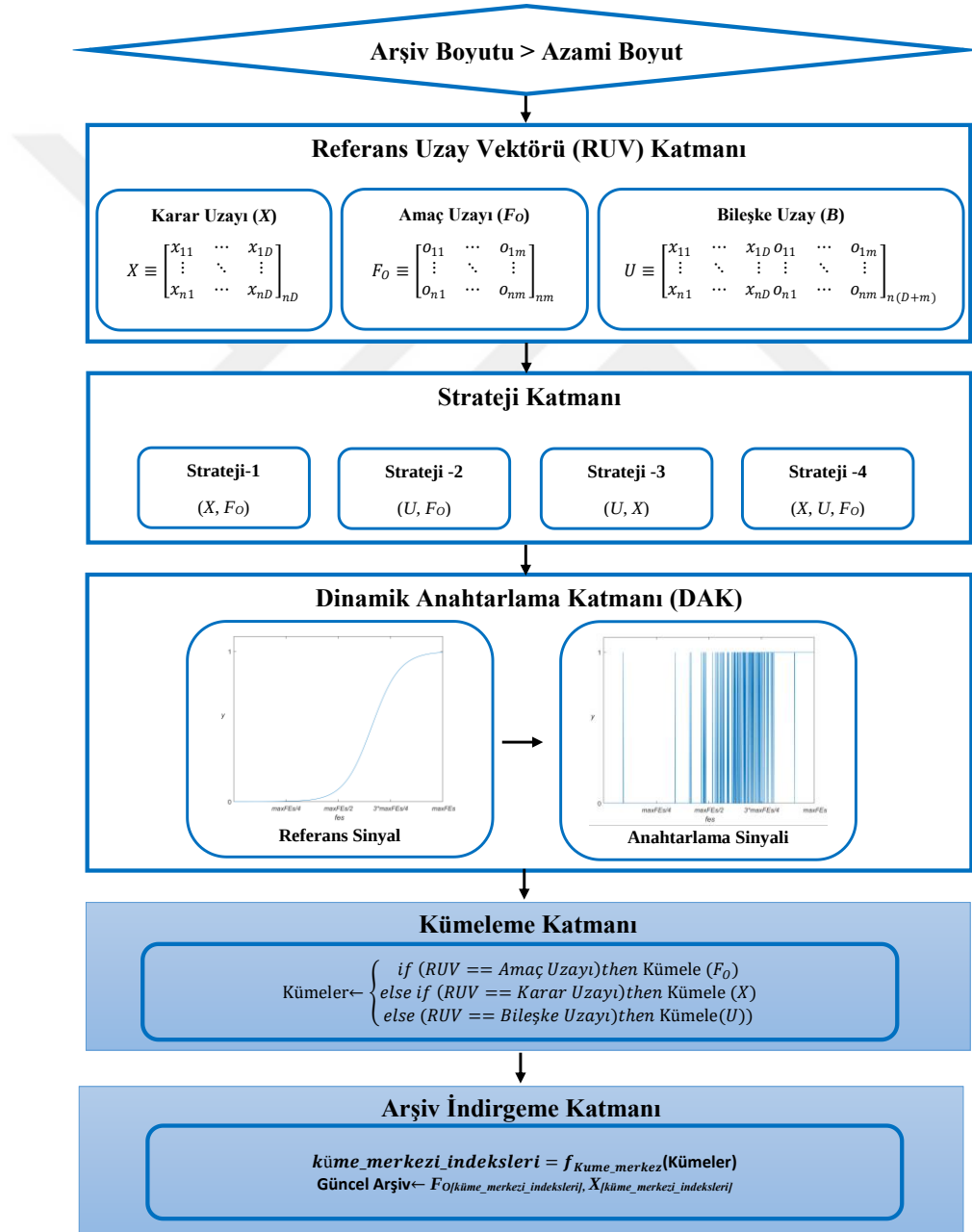
Bu bölümde tez çalışmasında yürütülen araştırmanın sonucunda geliştirilen dinamik referans uzay vektörü esaslı kümeleme ve arşiv indirgeme yöntemi tanıtılmaktadır. Kümeleme esaslı arşiv indirgeme algoritmasının geliştirilmesinde daha önce Kahraman ve diğerleri tarafından tanıtılan (Kahraman, Akbel, Duman, Kati, & Sayan, 2022) dinamik anahtarlamalı referans uzay vektörü yaklaşımı kullanılmıştır. Dinamik anahtarlamalı referans uzay vektörü yaklaşımı, kümelemenin yapılacağı uzayın belirlenmesi için kullanılmaktadır. Önerilen yöntemde, literatürdeki kümeleme esaslı arşiv indirgeme yöntemlerinden farklı olarak arşiv indirgeme sürecinde kümelemenin yapıldığı uzay vektörü arama süreci yaşam döngüsü boyunca dinamik bir şekilde değişmektedir. Bu sayede karar, amaç ve bileşke uzaylarda benzerliklerine göre gruplandırılan vektörlerin her üç uzayda da çeşitliliği sağlayacak şekilde seyreltilmesi (indirgenmesi) için güçlü bir teorik temele dayanan yöntem geliştirilmiştir. Bu kısa tanıtımdan sonra, önerilen yöntemin detayları takip eden alt bölümlerde açıklanmaktadır.

2.1. Önerilen Yöntem

Dinamik anahtarlamalı kalabalık (DAK) ve dinamik uzayda kümeleme (DUK) yöntemlerinin her ikisinde de arşiv indirgeme sürecinde ortak olarak karar, amaç ve bileşke referans uzaylar dinamik bir şekilde kullanılır. DAK yönteminin tatbik edildiği arşiv indirgeme sürecinin ilk evresinde referans uzay vektörü (RUV) belirlenmekte, ikinci evrede ise kalabalık mesafesi tabanlı boyut indirgeme işlemi yerine getirilmektedir. Benzer şekilde, DUK yönteminin tatbik edildiği arşiv boyutu indirgeme sürecinin ilk evresinde kümelemenin tatbik edileceği RUV belirlenmektedir. DUK yönteminin ikinci evresinde ise boyut indirgeme işlemi gerçekleştirilmek için kalabalık mesafesi yerine, hiyerarşik olmayan kümeleme çözümlemesi yöntemi kullanılmaktadır. Bu amaçla DUK yönteminin tasarımında kullanılan k-ortalamlar yöntemi önceki bölümlerde tanıtıldı. Bu açıklamalara göre, takip eden bölümde dinamik referans uzaylar esaslı kümeleme yöntemi açıklanmaktadır.

2.1.1. Dinamik Referans Uzaylar Esaslı Kümeleme ve Arşiv İndirgeme

Önerilen kümeleme tabanlı arşiv indirgeme yönteminin ilk evresi dinamik anahtarlama kalabalık mesafesi esaslı arşiv indirgeme yöntemi ile aynıdır. Her iki arşiv indirgeme sürecinde ortak olan süreç referans uzay vektörünün dinamik olarak üretildiği adımlardır. Buna göre, önerilen dinamik referans uzaylar esaslı kümeleme ve arşiv indirgeme sürecinin işleyişi Şekil 7’de sunulmaktadır.



Şekil 7. Dinamik referans uzay vektörü esaslı kümeleme ve arşiv indirgeme süreci

Şekil 7’de görüldüğü gibi, arşiv boyutunda aşım olması halinde arşiv indirgeme süreci başlatılır. Şekil 7 incelendiğinde, kümeleme esaslı arşiv indirgeme algoritması ile kalabalık mesafesi esaslı arşiv indirgeme algoritmasının ilk üç adımının ortak olduğu anlaşılmaktadır. Buna göre referans uzay katmanı, strateji katmanı ve dinamik anahtarlama katmanı her iki algoritmada ortaktır. Dinamik referans uzaylar esaslı kümeleme ve arşiv indirgeme katmanları ise ilk defa bu tez çalışmasında önerilen yöntemlerdir. Bu bilgilere göre, Şekil 7’de sunulan sürece ait her bir katman ve bu katmanlardaki algoritmalar aşağıda açıklanmaktadır.

Referans Uzay Vektörü Katmanı: Şekil 7’de görüldüğü gibi, kümeleme esaslı arşiv indirgeme sürecinde karar, amaç ve bileşke olmak üzere üç farklı referans uzay vektörü kullanılmaktadır. Bu üç referans uzay vektörü sayesinde, arama süreci yaşam döngüsü boyunca farklı uzaylarda çeşitlilik ve yerel arama yapmak mümkün hale gelmektedir.

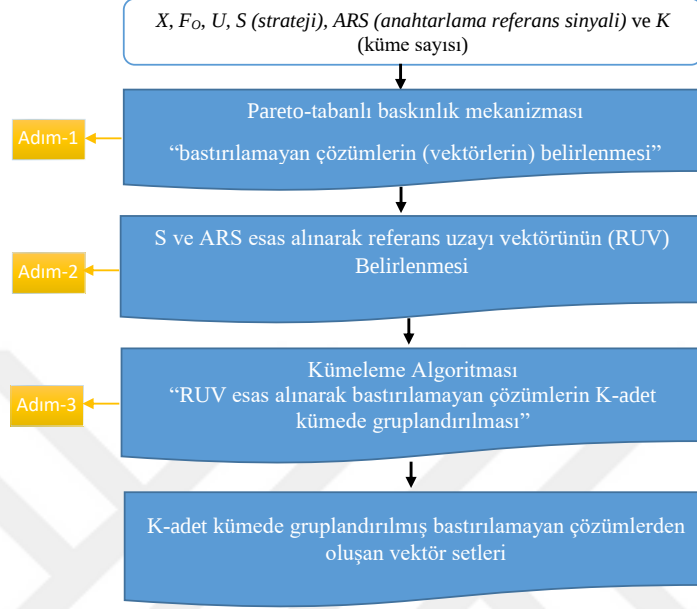
Strateji ve Dinamik Anahtarlama Katmanı: Bu katmanın kümeleme tabanlı arşiv indirgeme sürecindeki görevi karar, amaç ve bileşke referans uzayları arasından birini seçmektir. Bu süreçte arşiv indirgeme işleminin gerçekleştirileceği referans uzayı dinamik bir şekilde belirlenmektedir. Bu amaçla, referans uzay vektörünün (RUV) dinamikleştirilmesi için kaotik haritalar başta olmak üzere dinamik bir anahtarlama referans sinyali üretilmekte ve Algoritma 3’teki prosedür kullanılmaktadır. Buna göre, kümeleme esaslı arşiv indirgeme mekanizmasının her işleyişinde, kümelemenin gerçekleştirileceği RUV’u belirlemek için bu prosedür kullanılır. Algoritma-3’teki işleyişe göre X , U , F_O referans uzaylarından biri RUV olarak atanır. RUV’un dinamikleştirilmesi sayesinde, A (karar), U (bileşke), F_O (amaç) uzaylarının üçünde de çeşitlilik sağlanması mümkün hale gelmektedir. Böylelikle, RSV’nin statik olarak belirlenmesinden kaynaklanan prematüre yakınsama problemine karşı güçlü bir çözüm üretilmektedir.

Kümeleme Katmanı: Bu katmanın görevi, arşiv boyutunda bir aşım olduğunda bastırılmayan çözümlerin K -adet kümede gruplandırılmasıdır. Kümeleme sürecinin adımları Şekil 8’de sunulmaktadır.

Adım 1: Şekil 8’de ilk olarak arşivdeki ve popülasyondaki bastırılmayan çözümler tespit edilir. Bu süreçte, “1.3.3. Pareto-Baskınlık İlişkisi” bölümündeki denklem 2’de açıklanan Pareto-baskınlık mekanizmasının kuralları işletilir.

Adım 2: Bastırılmayan çözümlerin sayısı arşiv için belirlenen azami sayıyı aşması halinde, kümeleme esaslı arşiv indirgeme sürecinin ilk adımı olarak referans uzay vektörü belirlenir. Bu aşamada, Şekil 8’de sunulan dört strateji arasından seçilen strateji (S) olarak

atanır. Arama sürecinin herhangi bir t -anında anahtarlama referans sinyalinin değerine ve stratejiye bağlı olarak referans uzay vektörü Algoritma 5’de sunulduğu gibi belirlenir (Kahraman, Akbel, Duman, Kati, & Sayan, 2022).



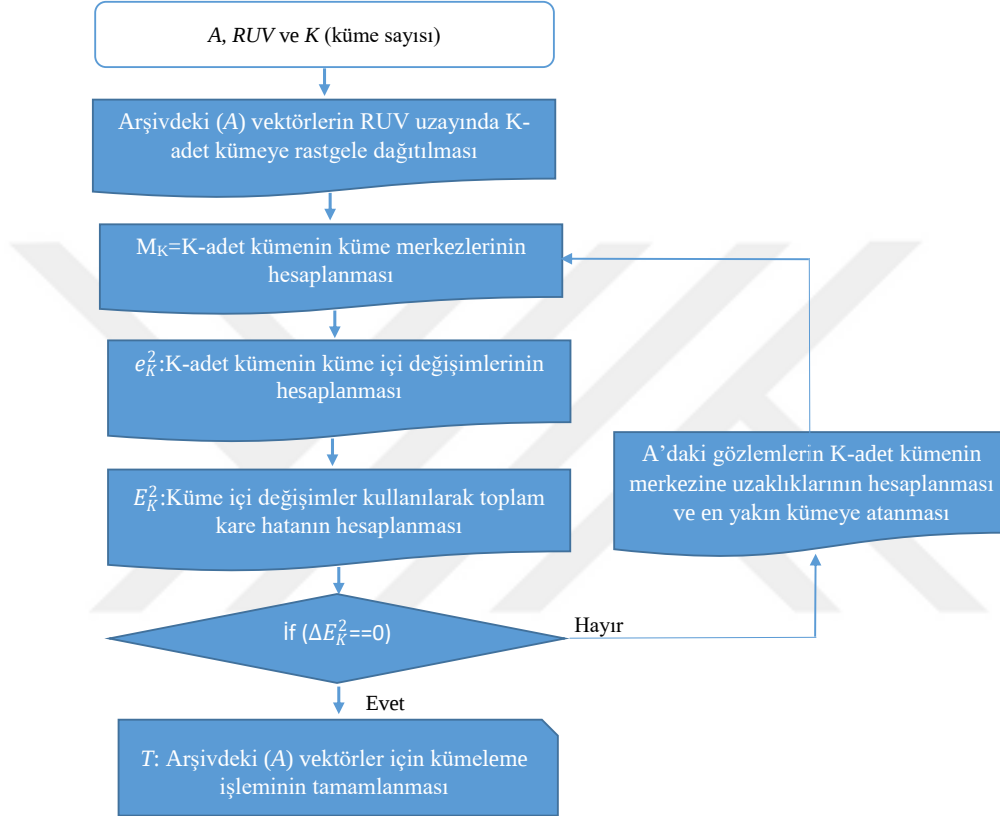
Şekil 8. Kümeleme sürecinin işleyişi

Algoritma 5. RUV’nin belirlenmesi sürecinin sözde kodu

	Parametreler: S (Şekil 7’deki dört stratejiden biri), ARS, fes, maxFEs Çıktı: RUV
1.	ref=reference_signal(fes);
2.	if (“S” is “strateji-1” or “strateji -2” or “strateji -3”)
3.	if (rand()<ref) then return (1 st RUV)
4.	else then return (2 nd RUV) end if
5.	end if
6.	if (“S” is “strateji-4”)
7.	if (fes<maxFEs/2)
8.	if (rand()<ref) then return (1 st RUV) else then return (2 nd RUV) end if
9.	else
10.	if (rand()<ref) then return (2 nd RUV) else then return (3 rd RUV) end if
11.	end if end if

Adım 3: bu adımda referans uzay vektörüne bağlı olarak kümeleme işlemi gerçekleştirilir. Kümeleme işlemi

Şekil 9’da sunulan akış diyagramına göre yürütülür. Kümeleme işlemi sonucunda arşivdeki bastırılmayan çözümler K-adet kümede gruplandırılır ve T matrisinde tutulur. Bu tez çalışmasında K-küme sayısı arşiv için belirlenen azami arşiv boyutu ile eşit tutulmuştur.



Şekil 9. Kümeleme Algoritması

İndirgeme Katmanı: bu katmanın görevi, K-adet kümede gruplandırılmış olan ve T -setinde tutulan vektörleri arşiv için belirlenen boyuta indirgemektir. Bu işlemi gerçekleştiren algoritmanın sözde kodu Algoritma 6’da sunulmaktadır.

Algoritma 6. İndirgeme katmanının sözde kodu

Parametreler: T Çıktı : $Y = \{y_1, y_2, \dots, y_k\}$, Y, küme merkezlerine en yakın vektörlerin setidir.	
1. for $T_i (i = 1, \dots, K)$ $m_i = \frac{1}{r_i} \sum_{l=1}^{r_i} T_{li}$ $jj = \text{index}(\text{argmin}_{l=1}^{r_i} \ x_{r \in C^i} - h_i\ ^2)$ $y_i = T_{li[jj]}$ end for	Küme merkezlerinin hesaplanması $H = \{h_1, h_2, \dots, h_K\}$, H kümelerin merkezlerini temsil eden vektördür, r_i , i-inci kümedeki bastırılmayan vektörlerin sayısıdır.
2. return Y	

Algoritma 6 incelendiğinde, K-adet kümede gruplandırılmış olan vektörlerin seti T'nin giriş olarak alındığı ve her bir küme merkezine en yakın vektörün bulunduğu görülmektedir. K-adet küme için her küme merkezine en yakın vektör bulunduğu K-boyutlu bir vektör seti elde edilmektedir. Y-matrisinde tutulan bu vektörler arşiv indirgeme süreci sonucunda elde edilen bastırılmayan çözümleri temsil etmektedir.

2.1.2. Önerilen Yöntemin ÇAURDE Algoritmasına Tatbik Edilişi

ÇAURDE (Duman, Akbel, & Kahraman, 2021) çoklu moda sahip çok amaçlı optimizasyon problemlerinin çözümlenmesinde kullanılan literatürdeki en güçlü birkaç algoritmadan biridir. Bu tez çalışmasında ise kümeleme esaslı arşiv indirgeme yönteminin ÇAURDE algoritmasında tatbik edilmesi sayesinde, çoklu moda sahip çok amaçlı optimizasyon problemleri için karar ve amaç uzaylarında daha etkili bir şekilde çeşitlilik sağlayabilen kümeleme esaslı ÇAURDE (DRUK-ÇAURDE) algoritması tasarlanmıştır. Tasarlanan algoritmanın sözde kodu Algoritma-Z'de sunulmaktadır. DRUK-ÇAURDE'nin ÇAURDE (Duman, Akbel, & Kahraman, 2021) ve DAK-ÇAURDE (Kahraman, Akbel, Duman, Kati, & Sayan, 2022) algoritmalarından farklılaştığı işlem adımları Algoritma 7'de mavi renkli olarak gösterilmiştir.

Algoritma 7. DRUK-ÇAURDE algoritmasının sözde kodu	
	Parametreler: S (strateji), K (küme sayısı), ARS (anahtarlama referans sinyali), maxFES (amaç fonksiyonu azami değerlendirme sayısı) Çıktı: PS _e , PF _e
	Birinci Evre: Arşiv ve Popülasyon Oluşturma Süreci
1	X, F _O : Arşivin Karar ve Amaç Uzayı vektörlerinin oluşturulması
2	P, F _P : URDE algoritmasında n-adet çözüm adayından oluşan popülasyon oluşturulması ve amaç fonksiyon değerlerinin hesaplanması.
3	İkinci Evre: DRUK-ÇAURDE Algoritması Arama Süreci Yaşam Döngüsü
4	while (fes<maxFES)
5	for i=1: n
6	F _{P[i]} =f_objective(P _[i])
7	Pareto baskınlık ilişkisini kullanarak P'deki seçkin/en iyi çözümü güncelleyin if dominates (F _{P[i]} , elite_fitness) then elite_fitness = F _{P[i]} and elite_position= P _[i] end if
8	end for
	Arşiv Güncelleme Süreci
9	[M, F _M]←Arşiv ve popülasyonu birleştir (A, F _A , P, F _P)
10	for i=1: n (n, A ve P' matrislerindeki toplam çözüm adayı sayısı)
11	for j=1:n-1
12	Pareto-baskınlık ilişkisini kullanın ve [M, F _M]'de baskın olmayan çözümler kümesini belirleyerek arşivi (PS ve PF) oluşturun. end for end for
13	if (arşiv aşımı)
14	RUV ←Algoritma- 5(S, ARS, fes, M, F _M , MUF _M)
15	T← Kümeleme_Algoritması (A, RUV, K)
16	Y←Algoritma-6 [M, F _M]
17	end if
18	URDE algoritmasının evrimsel arama süreci
19	Rehber seçimi: r ₁ , r ₂ , r ₃ vektörlerinin popülasyondan ve arşivden seçilmesi
20	Arama operatörleri: çaprazlama ve mutasyon operatörleri ile komşuluk araması ve çeşitlilik sağlanması.
21	fes++ (amaç fonksiyonu değerlendirme sayısının güncellenmesi)
22	Arama süreci sonlandırma kriteri sağlanıncaya kadar arama sürecine devam edilmesi
23	return PS _e ← A, PF _e ← F _A

Algoritma 7'deki sözde kod incelendiğinde, önerilen DRUK-ÇAURDE'nin iki evreden oluştuğu görülmektedir.

Birinci Evre (Arşiv ve Popülasyon Oluşturma Süreci): Bu evrede bastırılmayan çözümlerin tutulacağı Arşiv (A, F_o) ve URDE algoritmasının popülasyonunu temsil eden (P, F_P) matrisleri oluşturulur.

İkinci Evre (DRUK-ÇAURDE Algoritması Arama Süreci Yaşam Döngüsü): Bu evrede dinamik referans uzaylarda kümeleme esaslı ÇAURDE algoritmasının arama süreci yaşam döngüsü işletilmektedir. Arama süreci yaşam döngüsünde sonlandırma kriteri olarak literatürde standart olarak kullanılan amaç fonksiyonu azami değerlendirme sayısı ($\max FE$) benimsenmiştir. Algoritma 7 üzerinden açıklanacak olursa, arama süreci yaşam döngüsü boyunca sırasıyla, (i) 5-8 numaralı satırlar arasında popülasyondaki bastırılmayan vektörleri (çözüm adaylarını) bulma ve bu vektörler arasından varsa diğer tüm vektörlerden daha iyi olanını belirleme, (ii) 9-12 numaralı satırlar arasında popülasyon ve arşiv birleştirilip bu iki matristeki bastırılmayan çözümler tespit edilmektedir. Bastırılmayan çözümlerin tespit edilmesinde “1.3.3. Pareto-Baskınlık İlişkisi” başlıklı bölümdeki denklem 2'de tanımlanan Pareto-baskınlık kuralları kullanılmaktadır. (iii) 13-16 numaralı satırlarda ise arşivde bir aşım olma durumu kontrol edilmekte ve dinamik referans uzaylarda kümeleme esaslı arşiv indirgeme işlemi gerçekleştirilmektedir. Buna göre, arşiv aşımı 13-numaralı satırda, referans uzay vektörünün belirlenmesi 14-numaralı satırda, A 'daki bastırılmayan vektörlerin K -adet kümede gruplandırılması işlemi 15-numaralı satırda, K -adet kümenin her birinde sadece bir vektör kalmak üzere kümelerin indirgenmesi işlemi 16-numaralı satırda gerçekleştirilmektedir. (iv) Algoritma 7'de arama işlemi ise URDE algoritması tarafından 19-20 numaralı satırlar arasında gerçekleştirilmektedir. URDE algoritmasının işleyişi “1.5. Uyarlanabilir Rehberlik Mekanizmalı Diferansiyel Evrim Algoritması” başlıklı bölümde detaylı bir şekilde açıklanmıştır.

DSC-ÇAURDE ve DRUK-ÇAURDE algoritmaları arasındaki farklılıklar: İki algoritma arasındaki başlıca fark, arşiv indirgeme sürecinin tasarımıdır. DSC-ÇAURDE algoritmasında arşiv indirgeme sürecinde kalabalık mesafesi yöntemi kullanılmaktadır. DRUK-ÇAURDE algoritmasında ise arşiv indirgeme sürecinde kümeleme algoritması kullanılmaktadır. Buna göre, bu iki algoritmanın farklılaştığı işlem adımları Algoritma 7'de mavi olarak işaretlenen 15-16 numaralı satırlarda gerçekleştirilmektedir. DSC-ÇAURDE algoritmasında bastırılmayan vektörlerin bir r -mesafesi komşuluğundaki kalabalık hesaplanır. Kalabalık hesabı esasında her bir vektörün yakın çevresindeki vektörlerin

sayısının bulunması ve yoğun bölgelerin tespit edilmesini sağlar. Kalabalık hesabı sonucunda bastırılmayan tüm vektörler için komşu sayısına bağlı olarak bir kalabalık skoru hesaplanır. Kalabalık skoru esaslı arşiv indirgeme sürecinde ise sıklıkla vektörlerin skorlarına bağlı olarak çalışan olasılık içeren bir seçim yöntemi kullanılır. Bu yöntem kalabalığın, yani yoğunluğun çok olduğu bölgelerden vektör seçilmesi olasılığının fazla olmasını sağlar. Böylelikle, yoğun bölgelerin seyreltilmesi ve arşivin dağılımının çeşitlendirilmesi amaçlanmaktadır. DRUK-ÇAURDE algoritmasında ise arşivdeki çeşitliliği sağlamanın yolu vektörleri benzerliklerine göre gruplandırmaktan geçmektedir. Buna göre, arşivdeki vektörler K-ortalamalar yöntemi kullanılarak K-adet kümede gruplandırılırlar. Arşiv indirgeme sürecinde ise her kümeden küme merkezine en yakın olan vektör seçilir. Bu süreçte, kümeleme algoritmasının vektörleri benzerliklerine göre gruplandırması ve arşiv indirgeme sürecinde her gruptan sadece bir vektörün seçilmesi arşivdeki çeşitliliğin en üst düzeyde sağlanması bakımından kritik bir etki yaratmaktadır. Kalabalık mesafesi yöntemi bu düzeyde bir çeşitlilik sağlayacak şekilde vektörleri birbirinden ayıramamaktadır.

3. BULGULAR VE TARTIŞMA

Bu bölümde tez çalışmasında geliştirilen DRUK-ÇAURDE'nin performansını test etmek ve doğrulamak için yürütülen araştırmalar ve bu araştırmalardan elde edilen istatistiksel analiz sonuçları sunulmaktadır. Buna göre, DRUK-ÇAURDE'nin performansını test etmek ve doğrulamak için yürütülen araştırmalar üç alt bölümde sunulmaktadır. İlk alt bölümde, deneysel çalışma çevresi tanıtılmaktadır. İkinci alt bölümde

3.1. Deneysel Çalışma Çevresi

Bu bölümde sırasıyla tez çalışmasında geliştirilen deneysel ayarlar, DRUK-ÇAURDE'nin performansının araştırılmasında kullanılan karşılaştırma problemleri seti ve rakip algoritmalar ve son olarak deneysel çalışmalardan elde edilen verilerin test edilmesinde ve algoritmaların performanslarının analiz edilmesinde kullanılan istatistiksel analiz yöntemleri hakkında bilgiler verilmektedir.

3.1.1. Ayarlar ve Adillik

Deneysel çalışmaların yürütülmesinde başlıca gereksinim adilliğin ve objektifliğin sağlanmasıdır. Bu amaçla literatürde standart olarak kabul edilen ayarların ve kuralların benimsenmesi gerekir. Tez çalışmasında adilliği ve objektifliği sağlamak amacıyla IEEE CEC 2020 (Congress on Evolutionary Computation) kongresinde (Liang J. J., Suganthan, Qu, Gong, & Yue, 2019) evrimsel çok amaçlı optimizasyon çalışmaları için tanımlanan kurallar ve ayarlar referans alınmıştır. Buna göre, tez çalışması kapsamında yürütülen araştırmalarda aşağıdaki ayarlar benimsenmiştir:

- Deneysel çalışmaların tekrar sayısı 21'dir. Buna göre her bir algoritmanın IEEE CEC 2020 karşılaştırma problemleri setindeki her bir problem için yirmi bir adet PF ve PS setleri elde edilmiştir.
- Tüm rakip algoritmalar için popülasyon büyüklüğü ($200 * N_{ops}$ dir. N_{ops} ise çoklu moda sahip problemler için elde edilmesi gereken PS sayısıdır.
- Arama süreci sonlandırma kriteri olarak, diğer tüm IEEE CEC dokümanlarında tanımlandığı gibi amaç fonksiyonu azami değerlendirme sayısı (MaxFEs)

benimsenmiştir. Sonlandırma parametresinin değeri, $10.000 * N_{ops}$ olarak atanmıştır.

- Deneysel çalışmaların yürütüldüğü programlama ve işlemci çevresinin özellikleri şunlardır: MATLAB®R2018a, Intel(R) Core(TM) i7-4770K CPU @ 3.50 GHz, 16 GB RAM, x64-based işlemci.

3.1.2. IEEE CEC 2020 MMOP Karşılaştırma Suiti

Önerilen yöntemin rakip algoritmalarla karşılaştırılması ve performansının doğrulanması için kapsamlı bir araştırma yürütüldü. Bu araştırmalarda literatürdeki en güncel çok amaçlı optimizasyon algoritmalarının yanı sıra, iyi bilinen ve gerçek dünya problemlerinin çözülmesinde de sıklıkla kullanılan on sekiz rakip yöntem kullanıldı. Bunlar sırasıyla, DAK-ÇAURDE (Kahraman, Akbel, Duman, Kati, & Sayan, 2022), MOMRFO (Kahraman, Akbel, & Duman, Optimization of Optimal Power Flow Problem Using Multi-Objective Manta Ray Foraging Optimizer, 2022), MOCS (Yang & Deb, 2013), MO-Ring-PSO-SCD (Yue, Qu, & Liang, 2017), Omni-Opt (Deb & Tiwari, 2005), MOMGA (Nouhi, Khodadadi, Azizi, Talatahari, & Gandomi, 2022), MOFPA (Yang, Karamanoglu, & He, Multi-objective Flower Algorithm for Optimization, 2013), MOBA (Yang, Bat algorithm for multi-objective optimisation, 2011), NSMFO (Savsani & Tawhid, 2017), DN_NSGAII (Liang, Yue, & Qu, 2016), MOSMA (Premkumar, ve diğerleri, 2020), NSGA2 (Deb, Pratap, Agarwal, & Meyarivan, 2002), SPEA2 (Zitzler, Laumanns, & Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm, 2001), MOMVO (Jangir, Mirjalili, Saremi, & Trivedi, 2017), MOGWO (Mirjalili, Saremi, Mirjalili, & Coelho, 2016), MODA (Mirjalili, 2016), MOALO (Mirjalili, Jangir, & Saremi, Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems, 2017), MAOA (Khodadadi, Abualigah, El-Kenawy, Snasel, & Mirjalili, 2022), MOEA-D (Zhang & Li, 2007) algoritmalarıdır. Deneysel çalışmalarda bu algoritmaların tamamında bu algoritmaların tanıtıldıkları makalelerde tasarımcıları tarafından önerilen ayarlar kullanıldı. Deneysel çalışmalarda rakipler arasındaki adillik ise bir önceki başlıkta açıklanan deneysel ayarlar ile sağlandı. Tablo 1’de, kullanılan problem süitinin özellikleri sunulmaktadır.

Tablo 1. CEC 2020’de tanımlı CAO problemlerinin özellikleri

	Problem İsmi	Ölçeklenebilir değişken sayısı	Ölçeklenebilir hedef sayısı	Pareto en iyi bilinen	Pareto cephe geometrisi	Pareto set geometrisi	Ölçeklenebilir Pareto seti sayısı	N_Ops (N_global + N_local)
1	MMF1	✗	✗	✓	Dışbükey	Doğrusal Olm.	✗	2 + 0
2	MMF2	✗	✗	✓	Dışbükey	Doğrusal Olm.	✗	2 + 0
3	MMF4	✗	✗	✓	İçbükey	Doğrusal Olm.	✗	2 + 0
4	MMF5	✗	✗	✓	Dışbükey	Doğrusal Olm.	✗	2 + 0
5	MMF7	✗	✗	✓	Dışbükey	Doğrusal Olm.	✗	2 + 0
6	MMF8	✗	✗	✓	İçbükey	Doğrusal Olm.	✗	2 + 0
7	MMF10	✗	✗	✓	Dışbükey	Doğrusal	✗	1 + 0
8	MMF11	✗	✗	✓	Dışbükey	Doğrusal	✓	1 + 0
9	MMF12	✗	✗	✓	Dışbükey	Doğrusal	✓	1 + 0
10	MMF13	✗	✗	✓	Dışbükey	Doğrusal Olm.	✓	1 + 0
11	MMF14	✓	✓	✓	İçbükey	Doğrusal	✓	2 + 0
12	MMF15	✓	✓	✓	İçbükey	Doğrusal	✓	1 + 0
13	MMF1_e	✗	✗	✓	Dışbükey	Doğrusal Olm.	✗	2 + 0
14	MMF14_a	✓	✓	✓	İçbükey	Doğrusal Olm.	✓	2 + 0
15	MMF15_a	✓	✓	✓	İçbükey	Doğrusal Olm.	✓	1 + 0
16	MMF10_1	✗	✗	✓	Dışbükey	Doğrusal	✗	1 + 1
17	MMF11_1	✗	✗	✓	Dışbükey	Doğrusal	✓	1 + 1
18	MMF12_1	✗	✗	✓	Dışbükey	Doğrusal	✓	1 + 1
19	MMF13_1	✗	✗	✓	Dışbükey	Doğrusal Olm.	✓	1 + 1
20	MMF15_1	✓	✓	✓	İçbükey	Doğrusal	✓	1 + 1
21	MMF15_a_1	✓	✓	✓	İçbükey	Doğrusal Olm.	✓	1 + 1
22	MMF16_11	✓	✓	✓	İçbükey	Doğrusal	✓	2 + 1
23	MMF16_12	✓	✓	✓	İçbükey	Doğrusal	✓	1+2
24	MMF16_13	✓	✓	✓	İçbükey	Doğrusal	✓	2+2

3.1.3. İstatistiksel Test Yöntemleri ve Analiz

Deneysel çalışmalardan elde edilen verilerin analiz edilmesi için istatistiksel test yöntemlerinin kullanılmasına ihtiyaç vardır. Bu amaçla, bu tez çalışmasında algoritmalara

ait verilerin analiz edilmesi ve algoritmaların performanslarının karşılaştırılması için parametrik olmayan istatistiksel test yöntemleri kullanılmıştır. Bu süreçte, algoritmalar arasındaki ikili karşılaştırmalar için Wilcoxon (Kruskal & Wallis, 1952) ikili karşılaştırma test yöntemi ve algoritmaların tümünün birbirlerine kıyasla skorlarını araştırmak için de Friedman test yöntemi (Friedman, 1937) kullanılmıştır.

3.2. DRUK-ÇAURDE Versiyonlarının Performanslarının İncelenmesi

3.2.1. Friedman Skorları

Bu bölümde, farklı stratejiler ve anahtarlama sinyalleri kullanılarak tasarlanan dokuz farklı DRUK-ÇAURDE'nin versiyonunun performansı incelenmektedir. Tablo 2'de sunulan dokuz farklı DRUK-ÇAURDE versiyonlarının tasarlanmasında kullanılan stratejiler ve anahtarlama sinyalleri "1.6.2. Dinamik Referans Uzaylar Esaslı Kümeleme ve Arşiv İndirgeme" başlıklı bölümde ve Şekil 7 üzerinde açıklanmıştır.

DRUK-ÇAURDE'nin ve DAK-ÇAURDE'nin performansları Tablo 2'de sunulmaktadır. Tablo 2'de sunulan bilgiler algoritmaların, dört farklı performans metriği (PSP, HV, IGD_X ve IGD_F) için elde ettikleri değerlerdir. DRUK-ÇAURDE versiyonlarının performanslarının araştırılmasında, IEEE CEC MMOP karşılaştırma setinde yer alan yirmi dört problem kullanıldı. Algoritmaların yirmi dört problem için dört farklı performans metriğinin değerleri Friedman test yöntemi kullanılarak analiz edildi. Bu analizin sonucunda algoritmaların dört performans metriğinin her biri için Friedman skorları elde edilerek bu skorlar ve dört skorun ortalaması Tablo 2'de sunuldu. Bu açıklamalara göre, DRUK yönteminin performansının doğrulanması için ÇAURDE algoritmasının iyileştirilmiş farklı bir versiyonu olan DAK-ÇAURDE ile performansları karşılaştırılmıştır.

Tablo 2'de sunulan Friedman skorlarına göre, DRUK-ÇAURDE'nin tüm versiyonları DAK-ÇAURDE'den daha iyi performans metrik değerleri ve Friedman skorları elde etmeyi başarmıştır. Bu sonuca göre, aralarındaki tek fark arşiv indirgeme mekanizmalarının tasarımı olan bu iki algortimadan DRUK-ÇAURDE'nin daha iyi metrik değerlerine ve Friedman skorlarına sahip olmasının nedeni, dinamik anahtarlama referans uzaylar yaklaşımının kalabalık mesafesi yöntemi yerine kümeleme esaslı arşiv indirgeme yöntemi üzerinde daha iyi bir arama performansı etkisi oluşturmastır. Kümeleme esaslı yöntem, arşivdeki

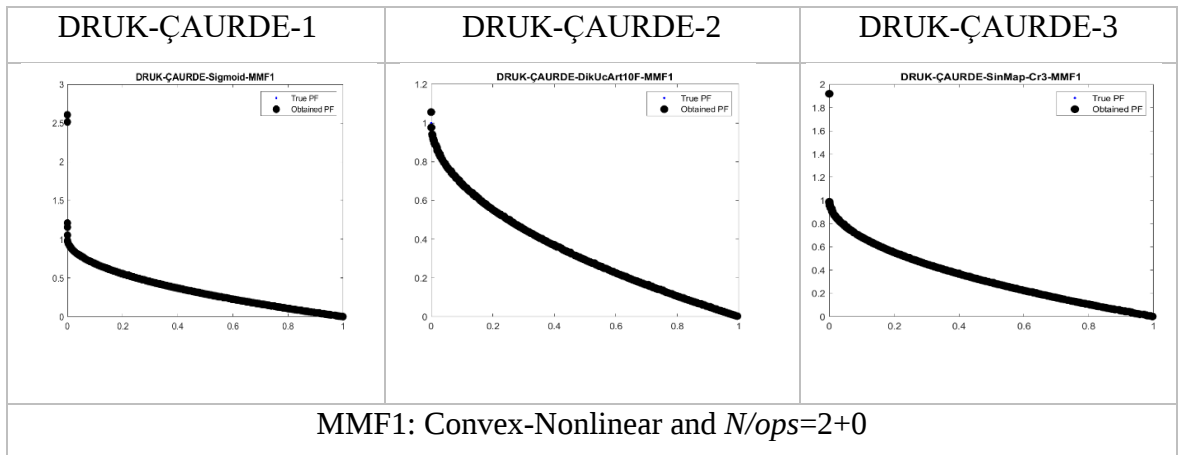
çeşitliliği ve komşu vektörleri belirleme konusunda kalabalık mesafesi yönteminde daha etkili olmuştur.

Tablo 2. Algoritmaların performans metrik değerlerine göre elde ettikleri Friedman skorları

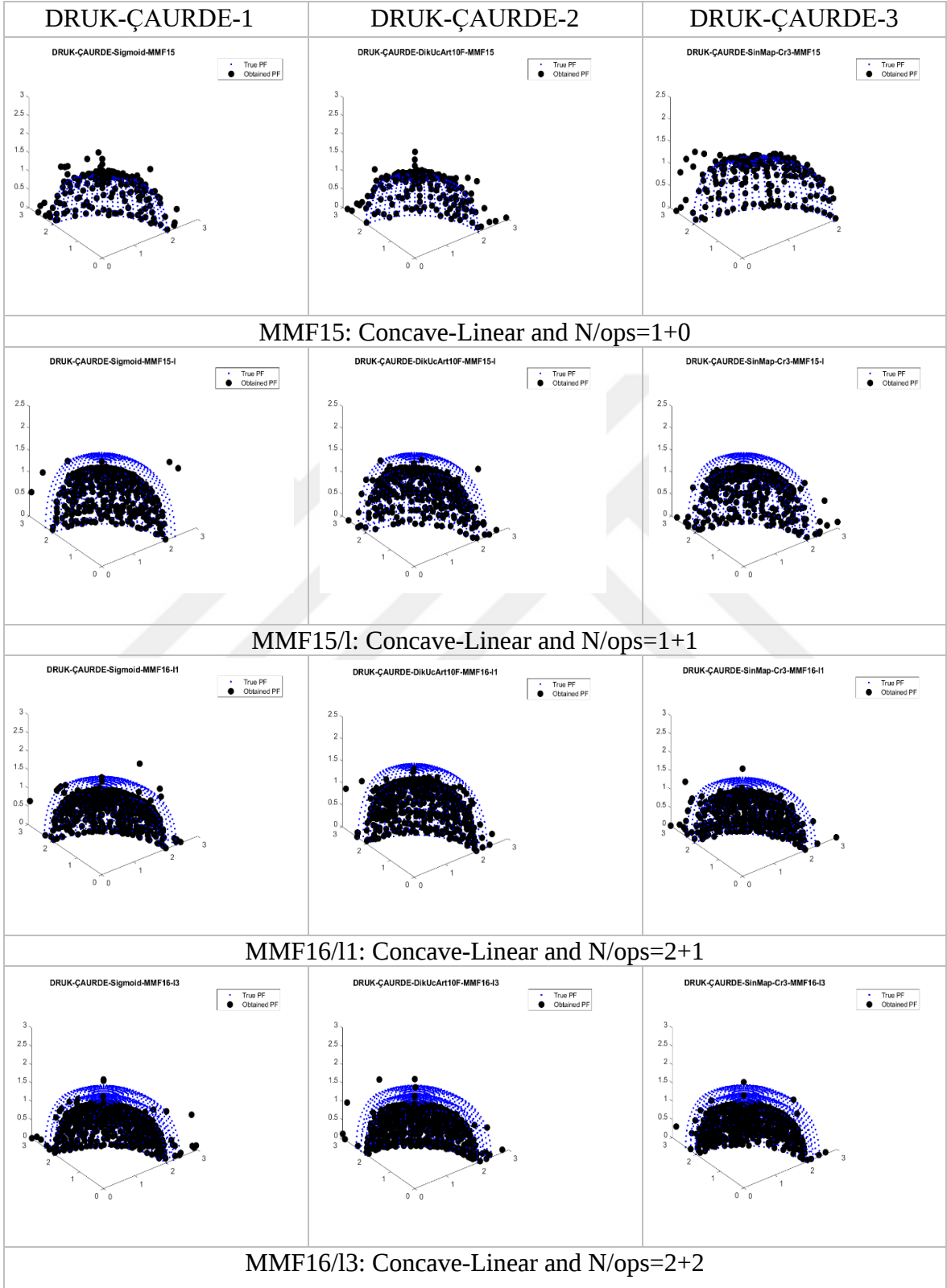
Algoritmalar	PSP	HV	IGDF	IGDX	Ortalama
DRUK-ÇAURDE -1 (Strateji-3/Sigmoid)	6.107	5.272	5.407	6.067	5.713
DRUK-ÇAURDE -2 (Strateji-3/Sawtooth)	5.675	6.234	6.345	5.530	5.946
DRUK-ÇAURDE -3 (Strateji-3/ Sinusoidal)	6.008	6.081	5.855	5.915	5.965
DRUK-ÇAURDE -4 (Strateji-3/ Piecewise)	5.808	6.393	6.353	5.754	6.077
DRUK-ÇAURDE -5 (Strateji-3/Singer)	6.127	6.190	6.060	6.075	6.113
DRUK-ÇAURDE -6 (Strateji-2/ Chebyshev)	6.095	6.395	6.558	6.065	6.278
DRUK-ÇAURDE -7 (Strateji-2/ Logistic)	6.194	6.450	6.690	6.260	6.399
DRUK-ÇAURDE -8 (Strateji-4/ Sinusoidal)	7.841	5.397	4.444	7.935	6.404
DRUK-ÇAURDE -9 (Strateji-3/ Iterative)	6.014	6.742	7.179	6.105	6.510
DAK-ÇAURDE (DSC-MOAGDE)	8.690	6.929	9.605	8.873	8.524

3.2.2. Yakınsama Performansları

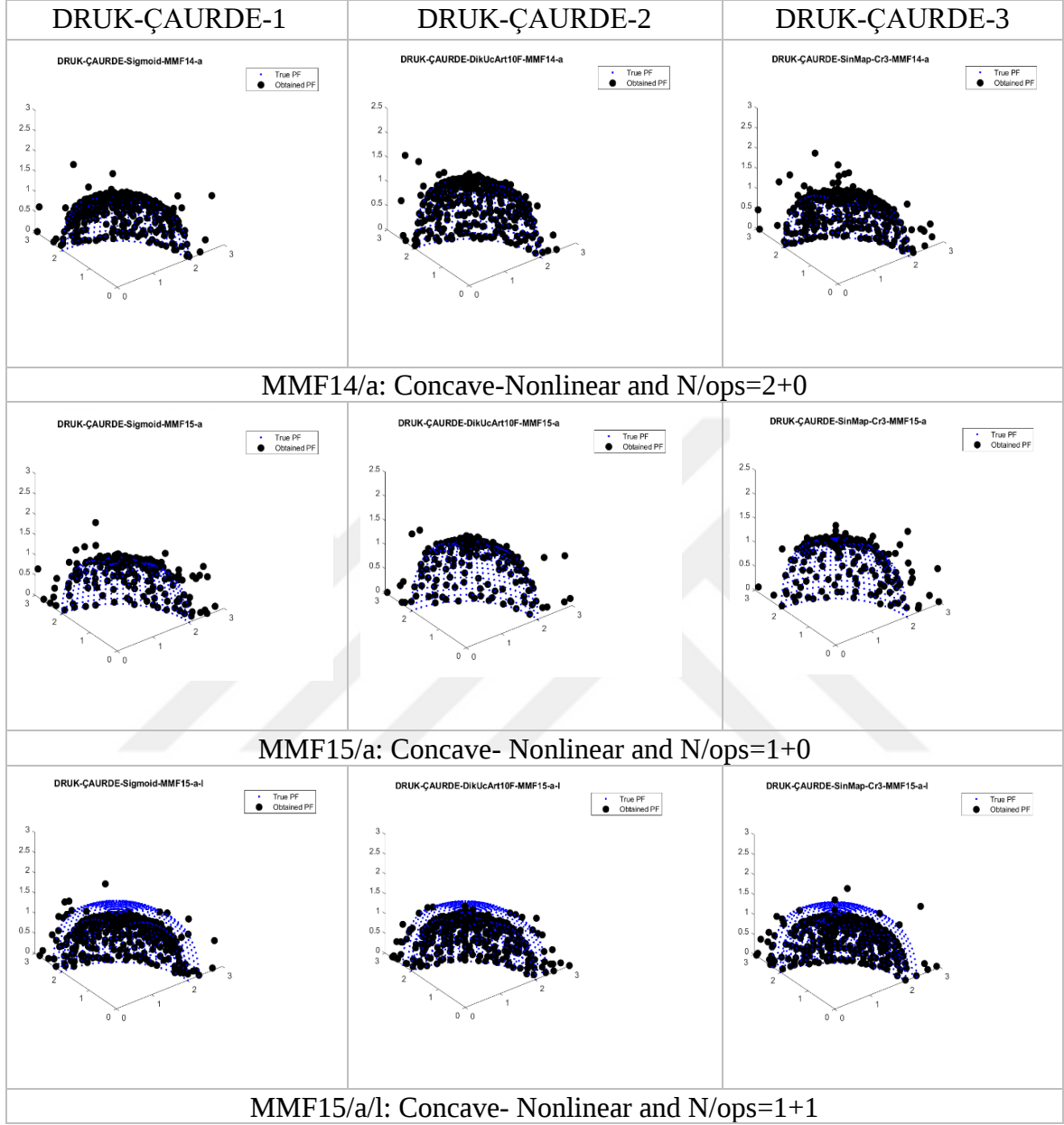
Tablo 2’de sunulan Friedman skorlarına göre, DRUK-ÇAURDE’nin en iyi ilk üç versiyonu belirlenmiş ve bu üç algoritmanın yirmi dört karşılaştırma problemi için amaç uzayındaki yakınsama performansları (PF) Şekil 10’da sunulmuştur.



Şekil 10'un devamı



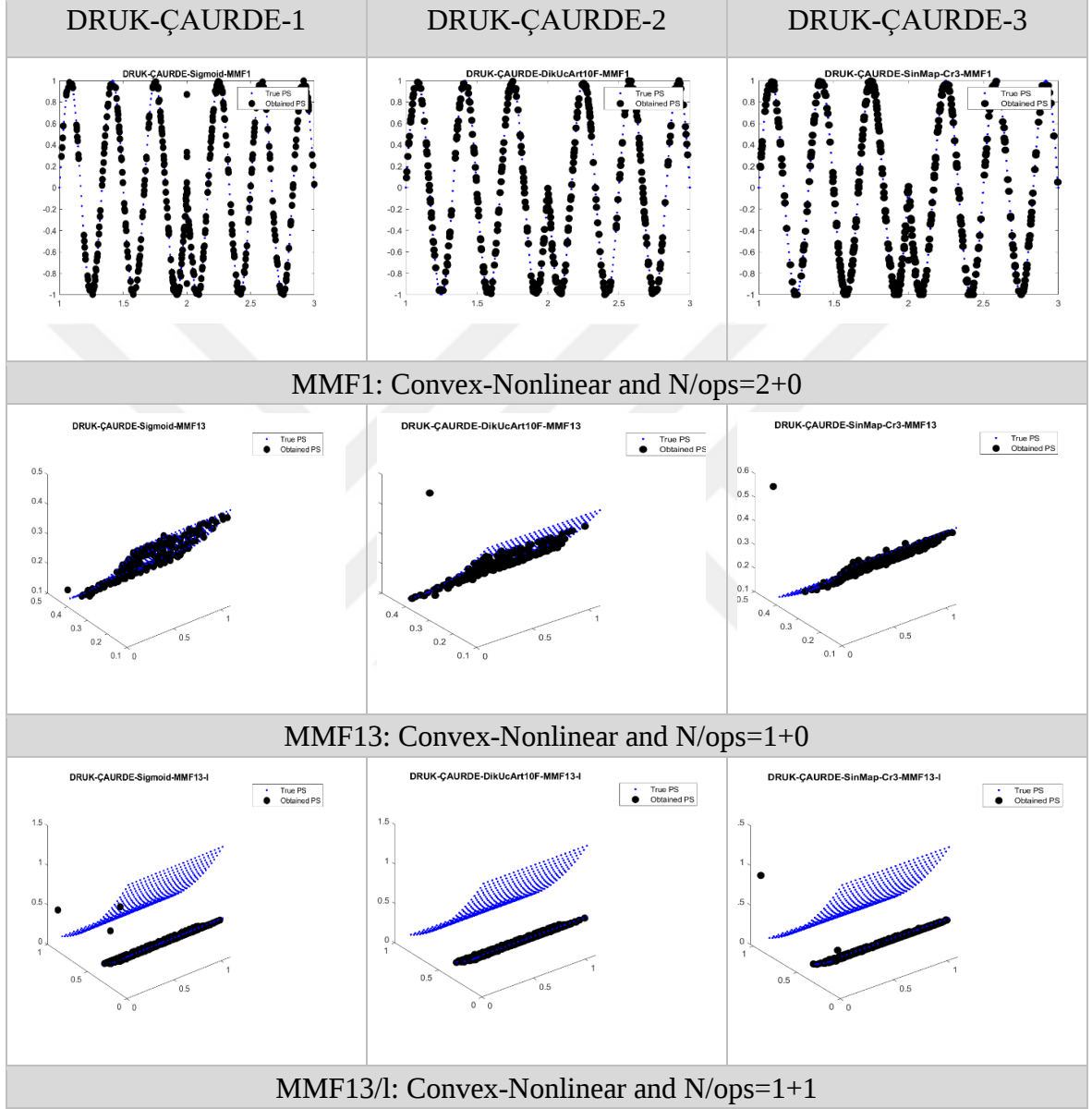
Şekil 10'un devamı



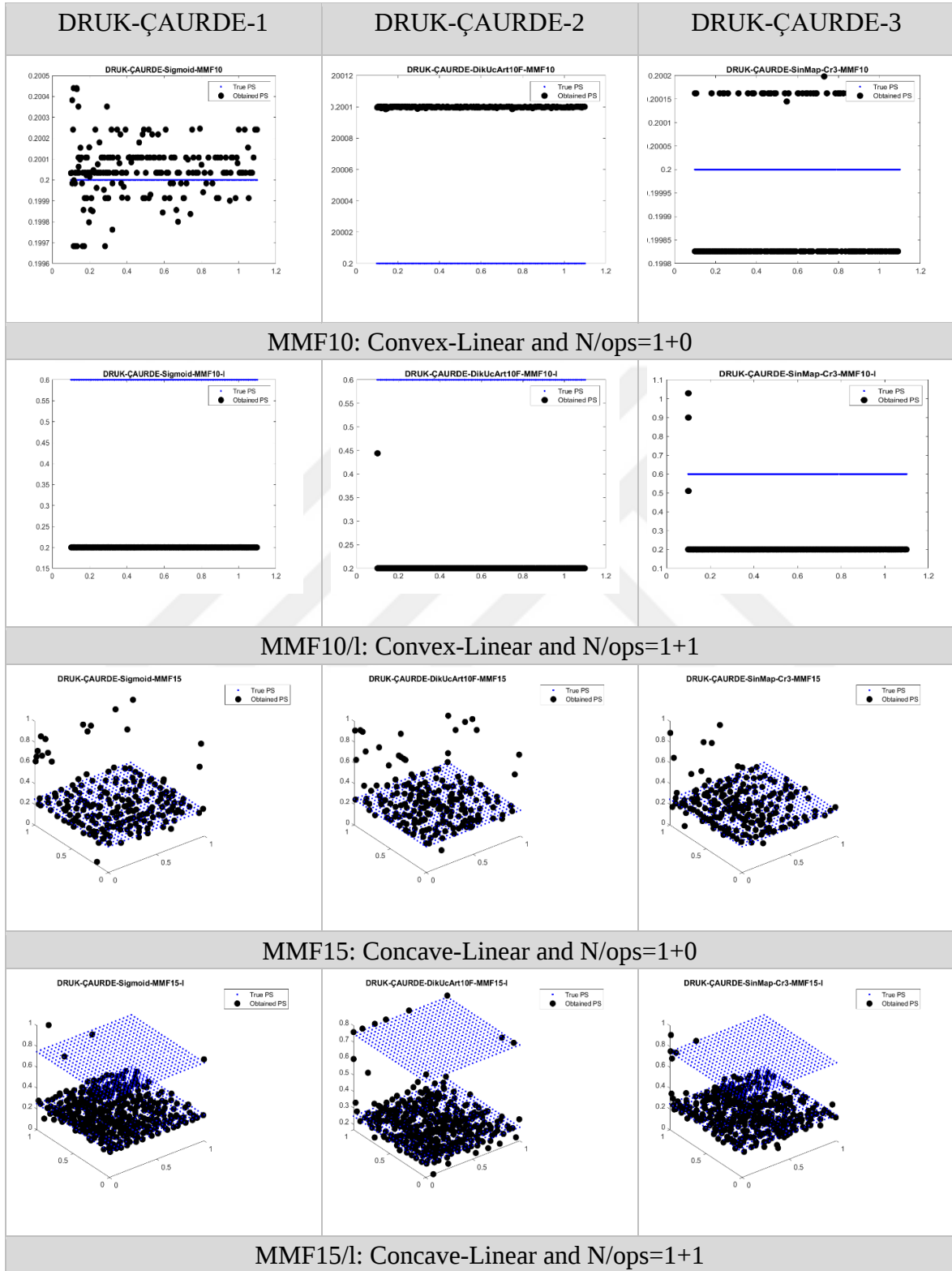
Şekil 10. DRUK-ÇAURDE'nin en iyi üç versiyonunun PF yakınsama performansları

Şekil 10 incelendiğinde, DRUK-ÇAURDE'nin en iyi üç versiyonunun PF yakınsama performanslarının birbirlerine yakın olduğu görülmektedir. Her üç algoritma da karşılaştırma setindeki problemlerin tümü için optimum PF setlerini büyük ölçüde bulabilmektedirler. Şekil 10'daki grafikler DRUK-ÇAURDE versiyonlarının amaç fonksiyonları çatışma halinde olan çok tipli çok amaçlı optimizasyon problemlerinin çözülmesinde etkili bir şekilde kullanılabileceklerini göstermektedir.

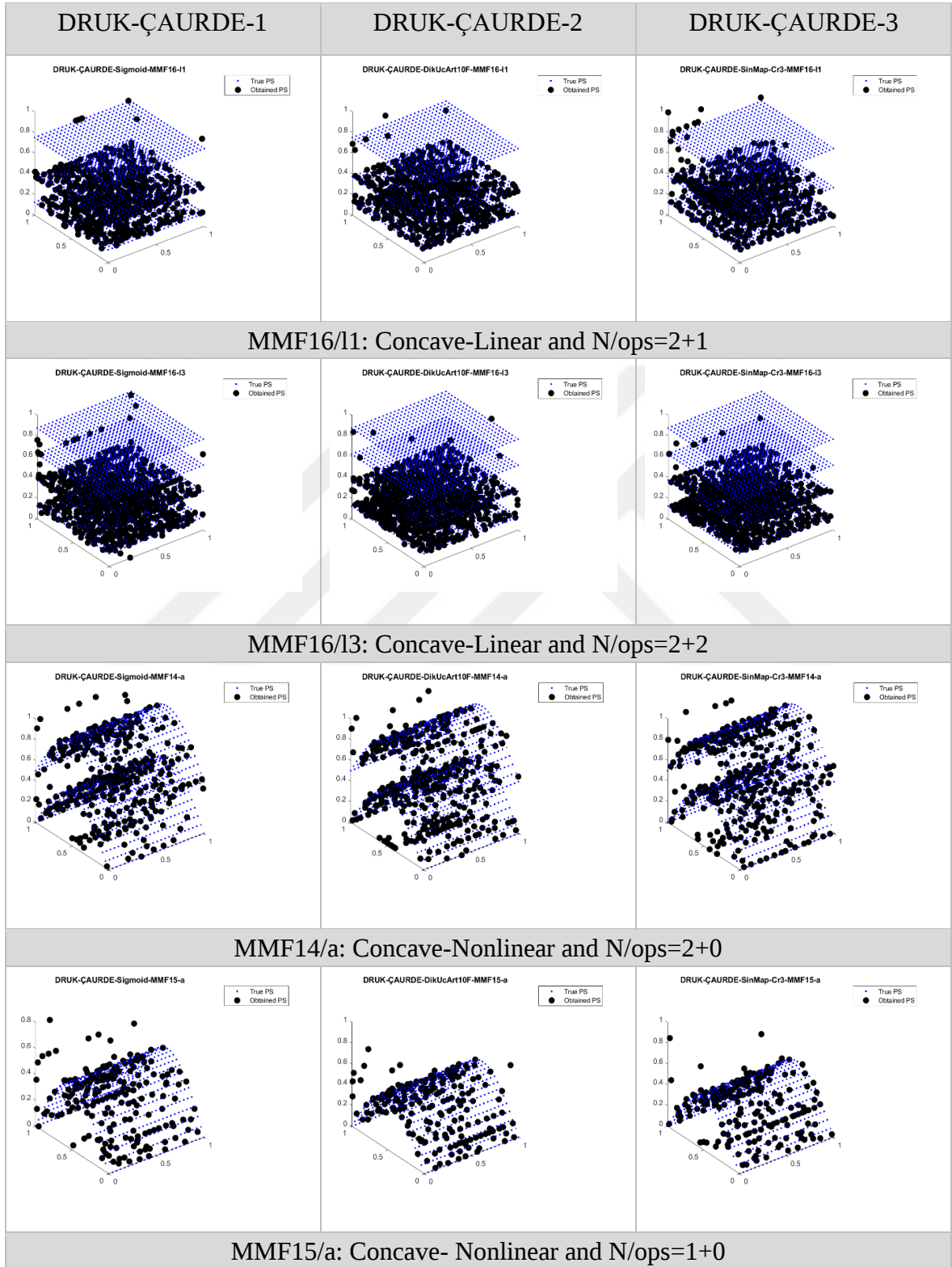
Tablo 2’de sunulan Friedman skorlarına göre, DRUK-ÇAURDE’nin en iyi ilk üç versiyonunun yirmi dört karşılaştırma problemi için karar uzayındaki yakınsama performansları (PS) Şekil 11’de görülmektedir.



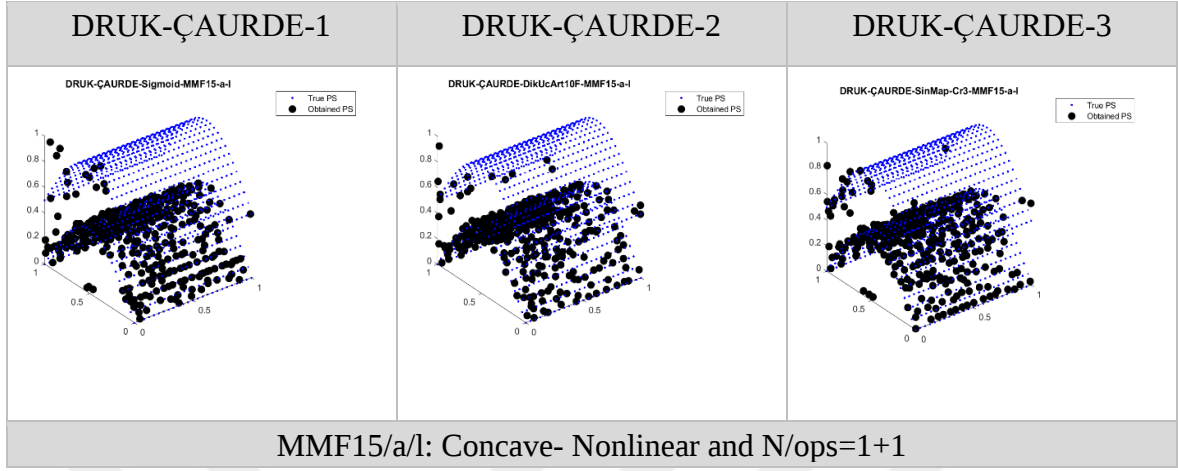
Şekil 11'in devamı



Şekil 11'in devamı



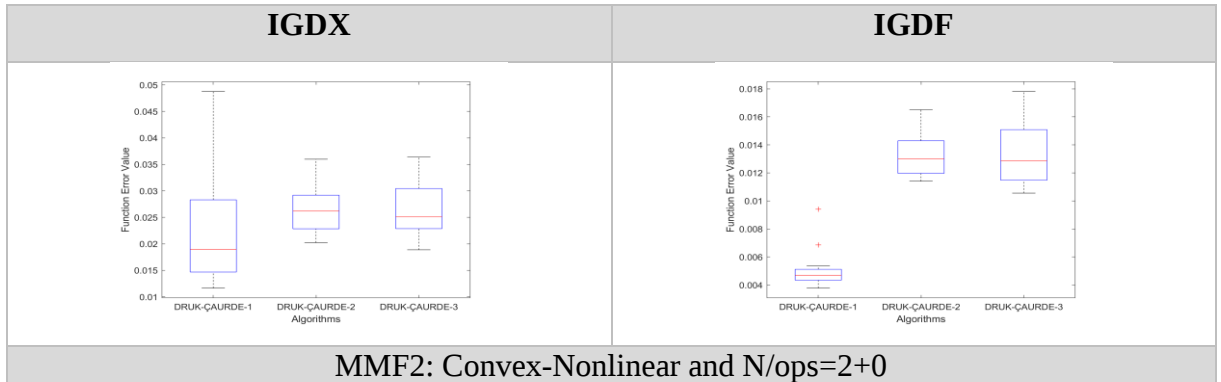
Şekil 11'in devamı



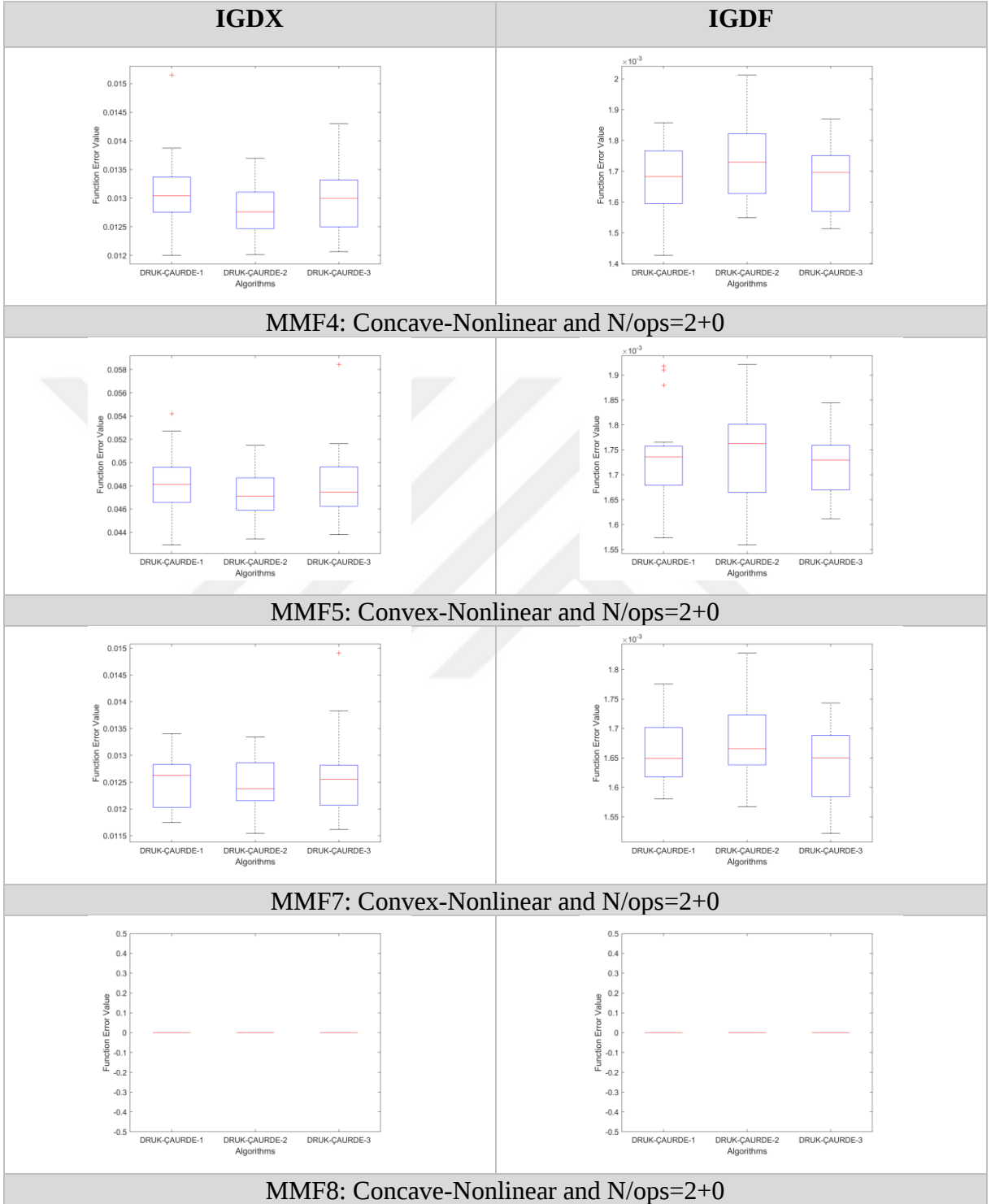
Şekil 11. DRUK-ÇAURDE'nin en iyi üç versiyonunun PS yakınsama performansları

Şekil 11 incelendiğinde, DRUK-ÇAURDE'nin en iyi üç versiyonunun PS yakınsama performanslarının da birbirlerine yakın olduğu görülmektedir. Her üç algoritma da karşılaştırma setindeki problemlerin tümü için optimum PS setlerini büyük ölçüde bulabilmektedirler. Şekil 11'deki grafikler DRUK-ÇAURDE versiyonlarının amaç fonksiyonları çatışma halinde olan çok tipli çok amaçlı optimizasyon problemlerinin çözülmesinde etkili bir şekilde kullanılabileceklerini göstermektedir.

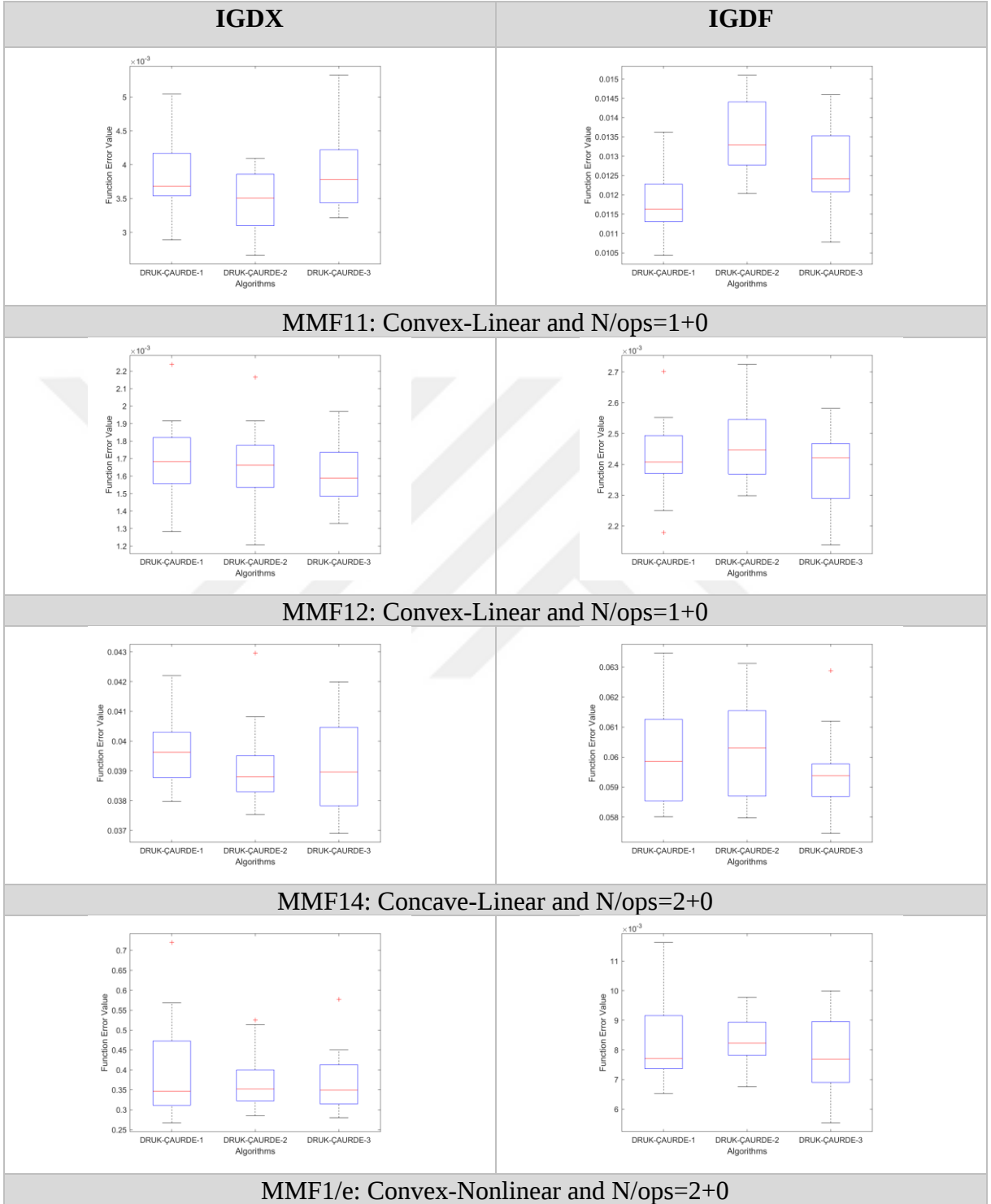
DRUK-ÇAURDE'nin en iyi üç versiyonunun karar ve amaç uzayındaki performanslarını bir de performans metrikleri üzerinden göstermek amacıyla bu üç algoritmanın IGDX (karar uzayı performans işaretçisi) ve IGDF (amaç uzayı performans işaretçisi) performans metrikleri için elde ettikleri değerler kutu grafikleri kullanılarak Şekil 12'de sunulmuştur.



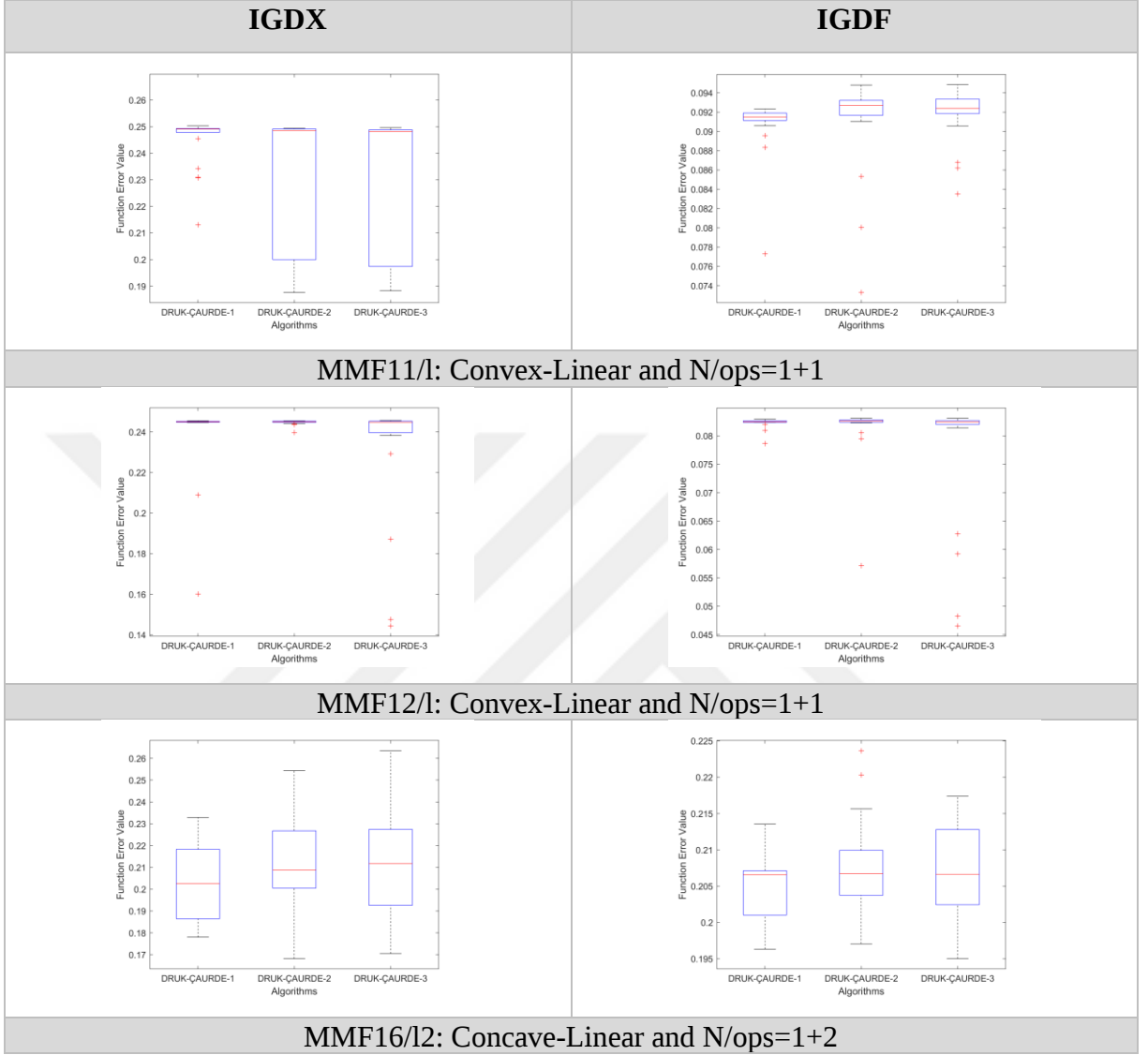
Şekil 12'nin devamı



Şekil 12'nin devamı



Şekil 12'nin devamı



Şekil 12. DRUK-ÇAURDE'nin en iyi üç versiyonunun IGDX ve IGDF kutu grafikleri

Şekil 12 incelendiğinde, DRUK-ÇAURDE'nin üç versiyonunun problemlerin büyük bölümünde birbirlerine benzer performanslar sergiledikleri görülmektedir. Bunun yanında, DRUK-ÇAURDE'nin ilk versiyonunun problemlerin bir bölümünde rakiplerine kıyasla daha iyi bir IGDX ve IGDF performansı sergilediği görülmektedir. Bunun nedeni, dinamik referans uzaylar yaklaşımının tatbik edilme sıklığının bu versiyonda daha etkili sonuçlara yol açmasıdır. Haliyle, DRUK tabanlı bir arşiv indirgeme algoritması tasarımı anahtarlama referans sinyalinin algoritmanın performansı üzerinde önemli bir etkiye sahip olduğu ortaya çıkmaktadır.

3.3. DRUK-ÇAURDE'nin Rakiplerle Karşılaştırılması

Bu bölümde literatürdeki güncel çok amaçlı optimizasyon algoritmaları ile karşılaştırmalar yapılmakta ve DRUK-ÇAURDE'nin rakip algoritmalara kıyasla performansı araştırılmaktadır. Bu amaçla, literatürde iyi bilinen ve güncel on sekiz rakip algoritma deneysel çalışmalarda kullanıldı. Bu açıklamalara göre, takip eden üç alt bölümde sırasıyla rakip algoritmaların Friedman skorları ve rekabetçi ilk üç algoritma için yakınsama analizi sunulmaktadır.

3.3.1. Friedman Skorları

Bu bölümde DRUK-ÇAURDE'nin ve rakiplerinin Friedman skorları incelenmektedir. On sekiz rakip algoritmanın karşılaştırıldığı istatistiksel analiz sonuçlarına göre rakiplerin Friedman skorları Tablo 3'de sunulmaktadır.

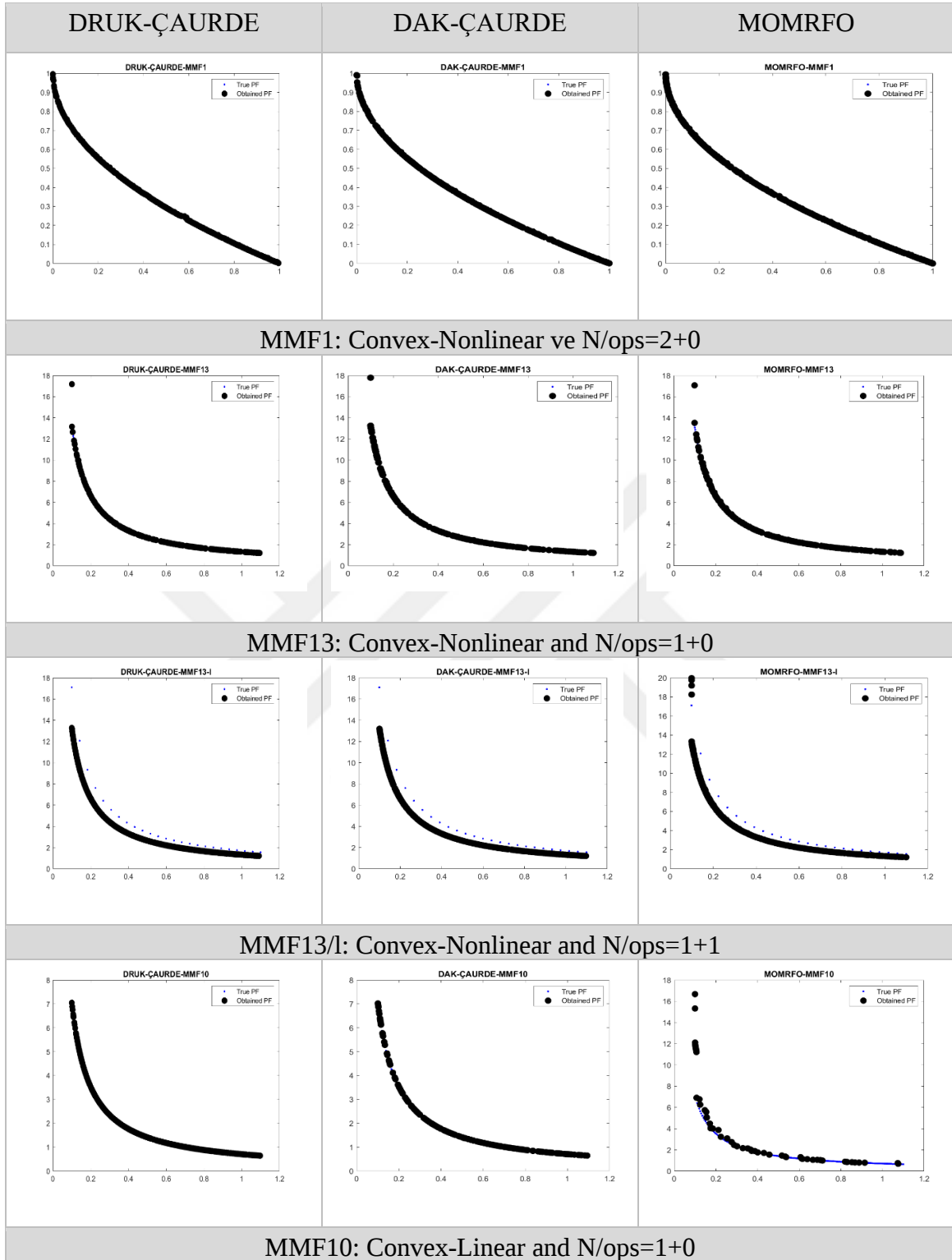
Tablo 3'te sunulan bilgiler algoritmaların, dört farklı performans metriği (PSP, HV, IGDX ve IGDF) için elde ettikleri değerlerdir. Tablo 3'te sunulan Friedman skorlarına göre, dört performans metriğinin ortalaması üzerinden değerlendirildiğinde DRUK-ÇAURDE'nin tüm rakiplerinden daha iyi bir değere sahip olduğu görülmektedir. Hatta bu dört performans metriğinden en güncel ve gürbüz olanı PSP için DRUK-ÇAURDE'nin rakiplerine kıyasla bariz bir üstünlüğe sahip olduğu görülmektedir. Bunun yanında, IGDX ve IGDF metrikleri için de DRUK-ÇAURDE'nin rakiplerinden açık bir şekilde üstün bir performans sergilediği anlaşılmaktadır. HV metriği üzerinden değerlendirildiğinde, MOBA, Omni-Opt ve MOCS algoritmalarının rekabetçi bir performans sergiledikleri DRUK-ÇAURDE'nin bu rekabetçi yöntemlere yakın bir performans sergilediği anlaşılmaktadır. Bu sonuçlara göre önerilen DRUK-ÇAURDE'nin literatürdeki güçlü ve güncel rakip yöntemlerle rekabetçi bir performans sergilediği açıkça görülmektedir.

Tablo 3. DRUK-ÇAURDE'nin ve rakiplerinin Friedman skorları

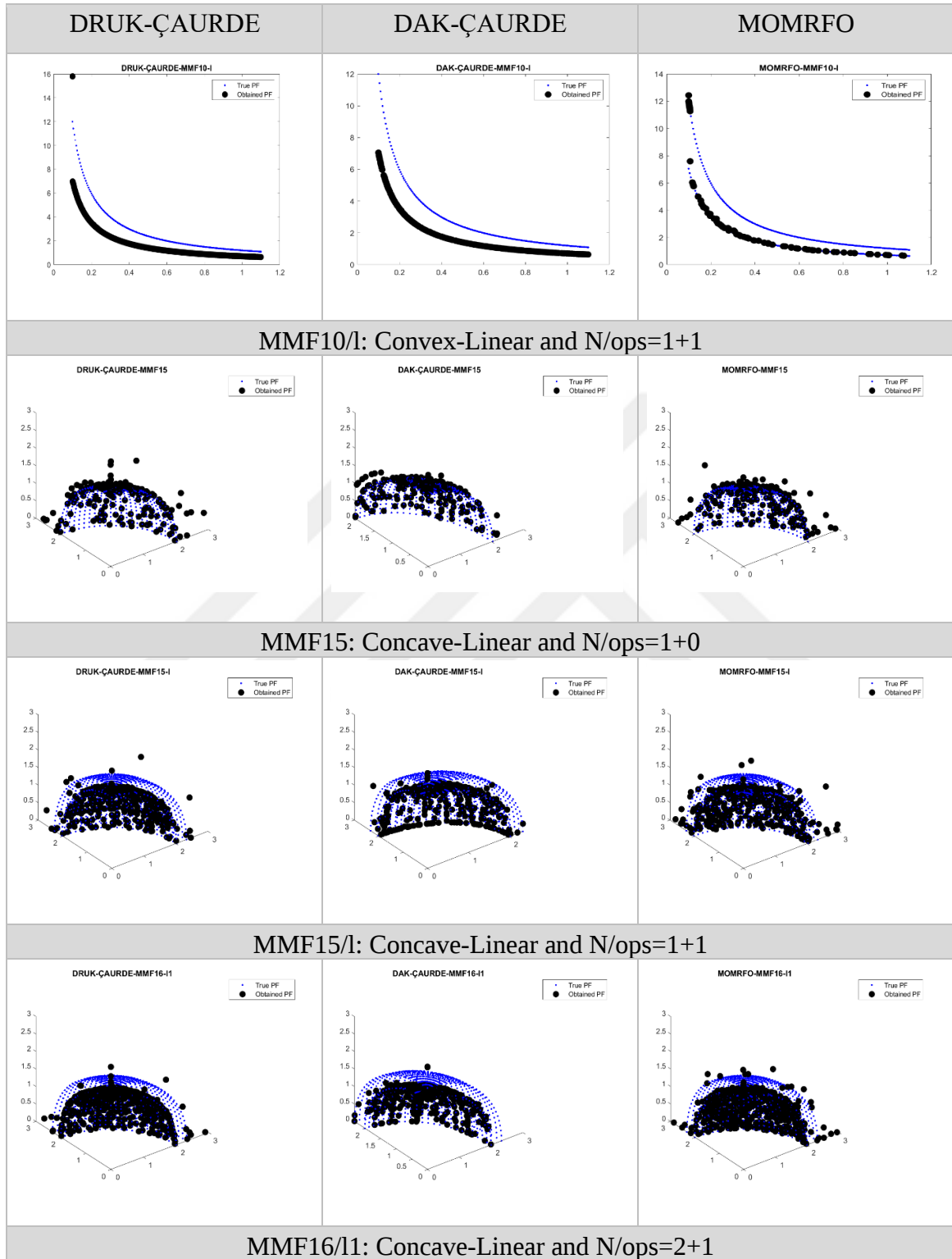
Algoritmalar	1/PSP	1/HV	IGDF	IGDX	Ortalama
DRUK-ÇAURDE	3.950	7.599	5.149	3.855	5.138
DAK-ÇAURDE	4.946	7.895	7.456	5.026	6.331
MOMRFO	5.397	11.093	5.839	5.633	6.991
MOCS	8.621	6.306	5.466	7.762	7.039
MO-Ring-PSO-SCD	4.829	11.254	7.446	5.139	7.167
Omni-Opt	11.139	6.687	8.079	11.190	9.274
MOMGA	8.772	9.272	14.480	7.284	9.952
MOFPA	13.212	7.204	7.218	12.692	10.082
MOBA	13.530	6.546	8.121	13.119	10.329
NSMFO	8.101	14.006	11.194	8.105	10.352
DN_NSGAII	11.442	7.484	11.135	11.843	10.476
MOSMA	9.085	13.315	11.020	9.008	10.607
NSGA2	14.111	7.133	8.873	14.226	11.086
SPEA2	12.740	10.865	12.718	13.637	12.490
MOMVO	13.308	12.647	12.708	12.917	12.895
MOGWO	14.167	12.046	11.343	14.202	12.939
MODA	11.585	12.528	15.571	12.413	13.024
MOALO	12.744	13.853	15.246	13.500	13.836
MAOA	12.873	19.865	13.954	12.919	14.903
MOEA-D	15.446	12.403	16.980	15.530	15.090

3.3.2. Yakınsama Analizi

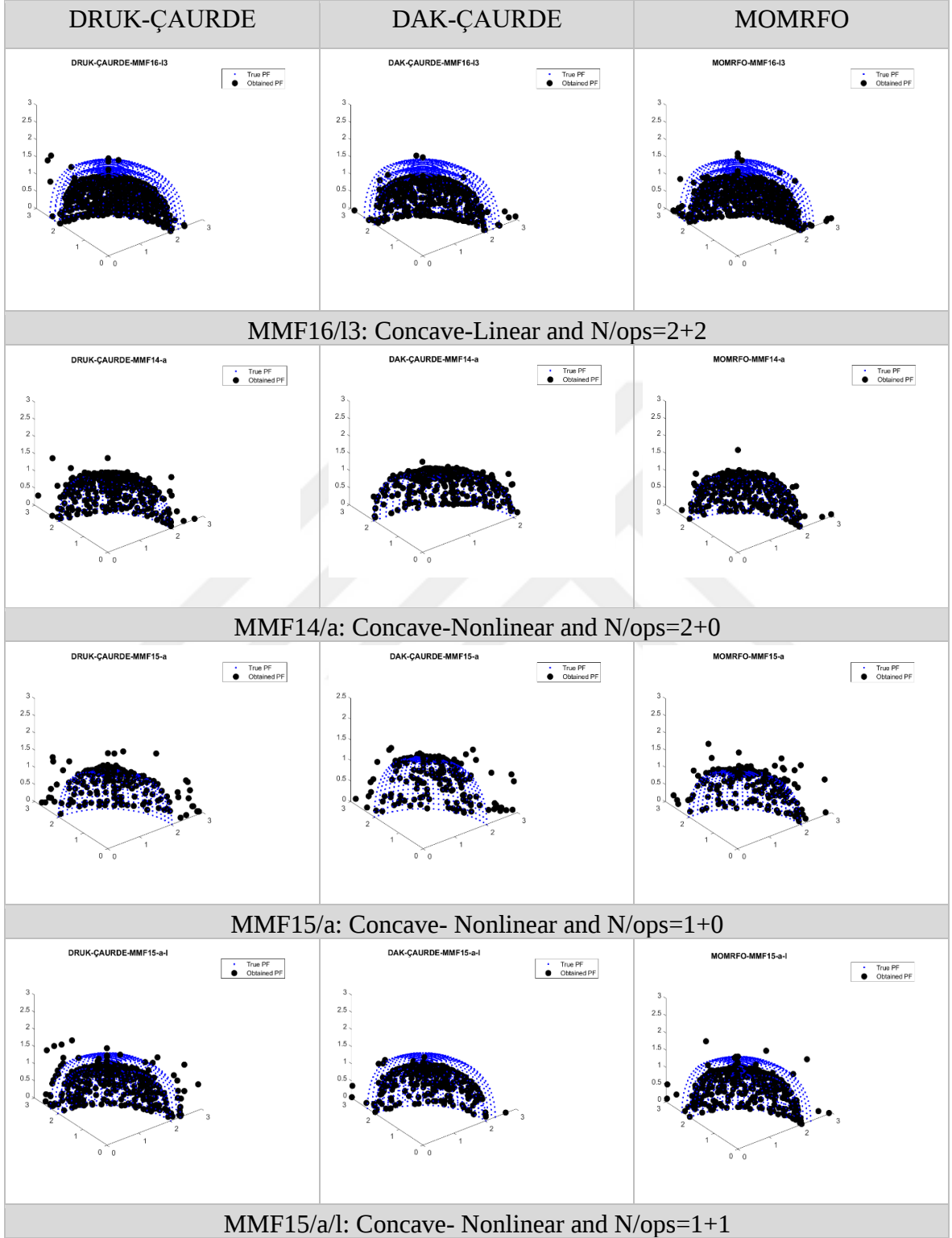
Tablo 3'te sunulan Friedman skorlarına göre, ilk üç sırada yer alan algoritmanın amaç uzayındaki ve karar uzayındaki optimum çözümlerin setlerine (PF ve PS) yakınsama grafikleri çizdirildi. Buna göre, ilk üç sırada yer alan algoritmalar sırasıyla, DRUK-ÇAURDE, DAK-ÇAURDE ve MOMRFO dur. Bu üç algoritmanın on iki problem için amaç uzayındaki optimum çözüm setine yakınsama grafikleri Şekil 13'te sunulmaktadır.



Şekil 13'ün devamı



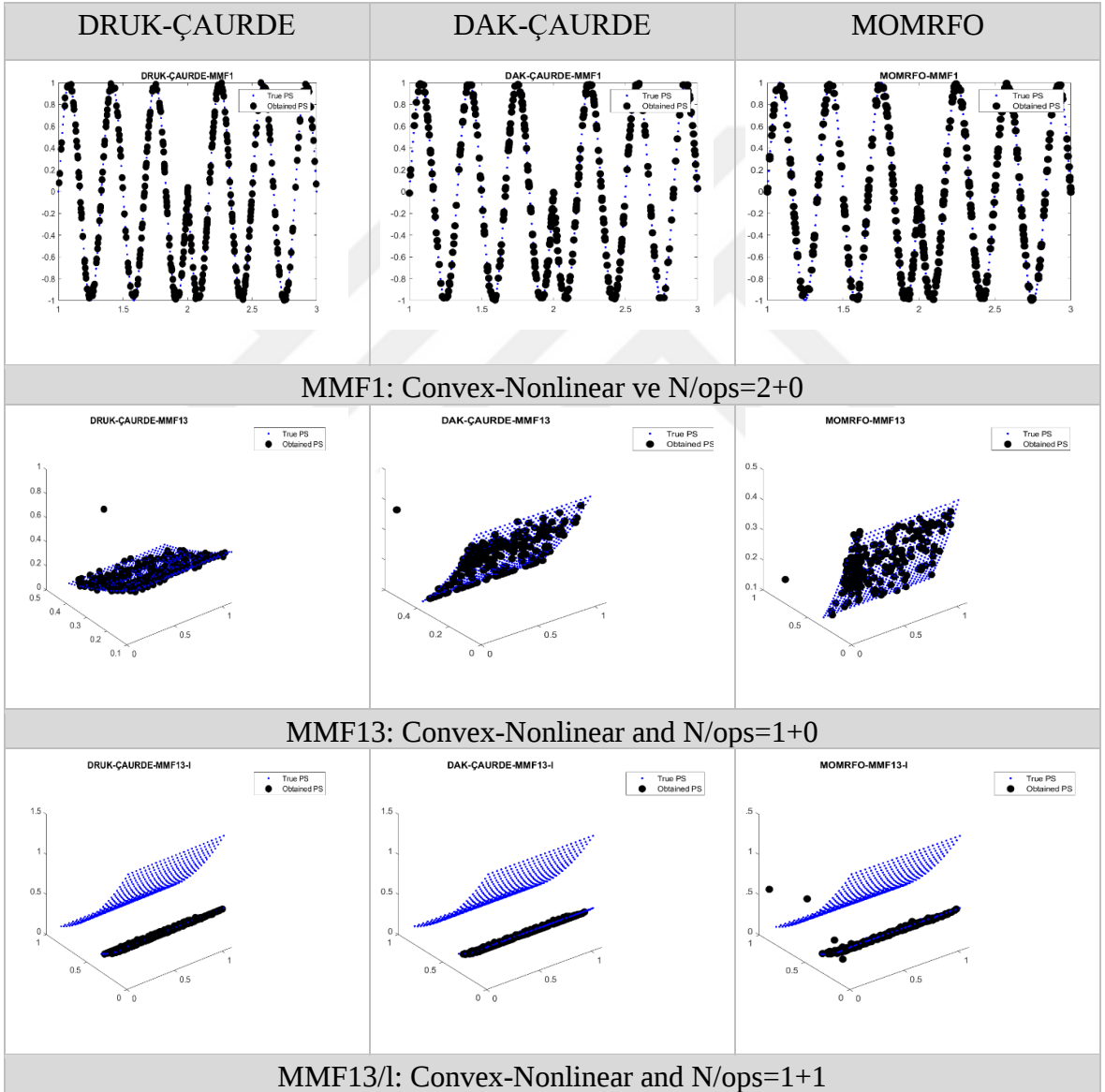
Şekil 13'ün devamı



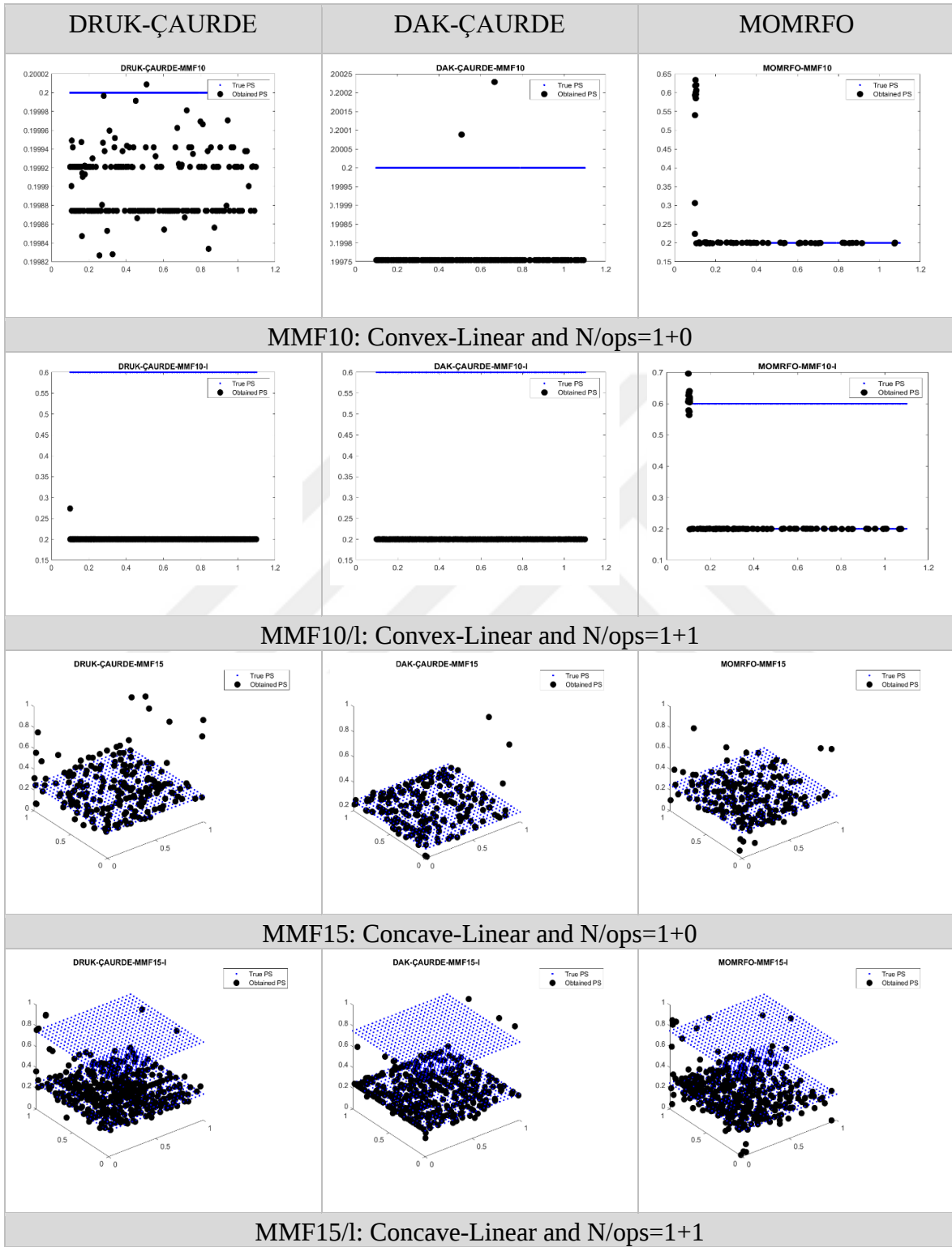
Şekil 13. Friedman skorlarına göre rekabetçi ilk üç algoritmanın PF yakınsama performansları

Şekil 13 incelendiğinde, rekabetçi üç algoritmanın da birbirlerine problemler için optimum amaç uzayı vektör setlerini büyük ölçüde buldukları görülmektedir. Bunun yanında özellikle DRUK-ÇAURDE'nin ve DAK-ÖAURDE'nin MOMRFO'dan daha başarılı bir şekilde PF setlerini bulabildikleri anlaşılmaktadır.

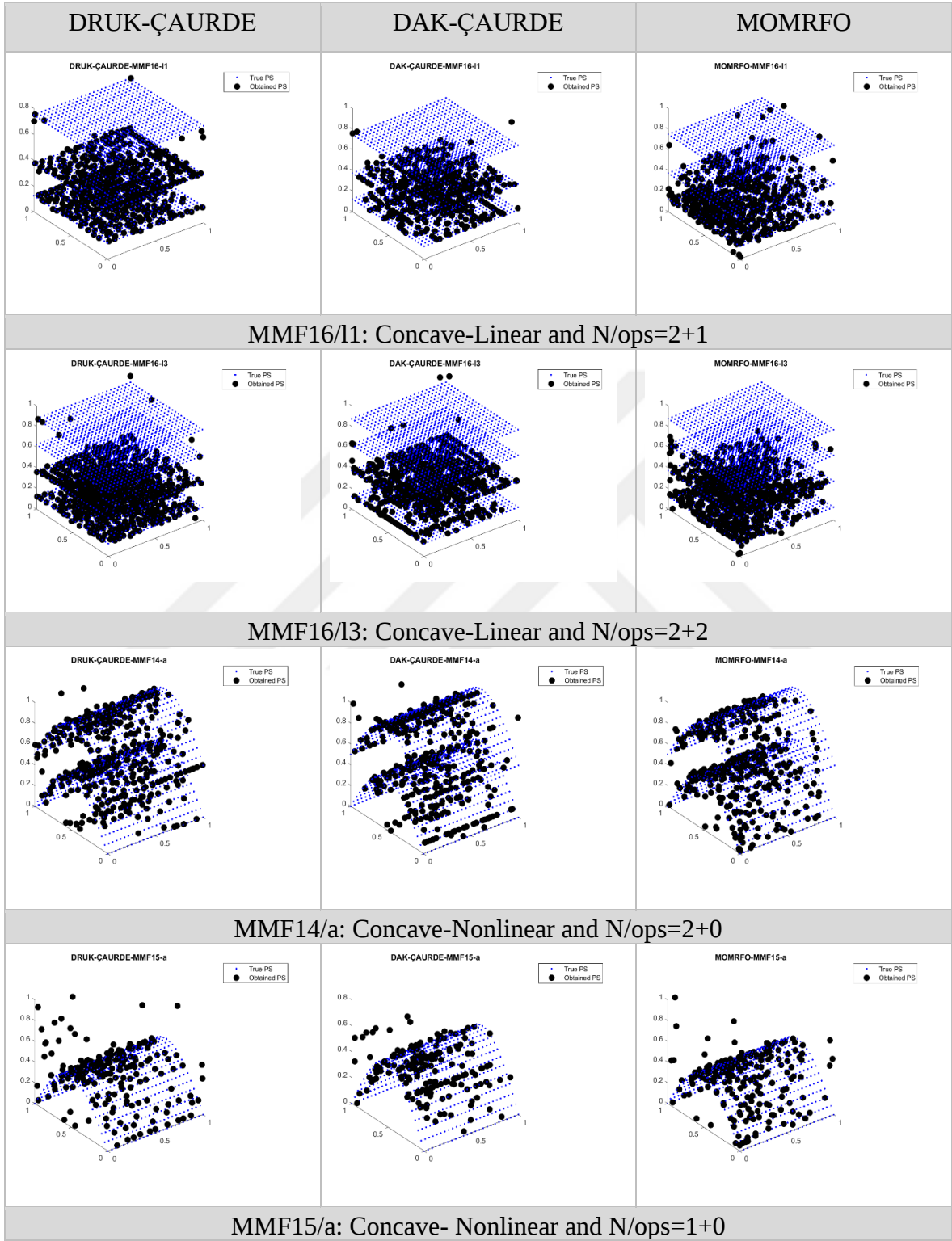
Tablo 3'te sunulan Friedman skorlarına göre ilk üç sırada yer alan algoritmanın karar uzayındaki optimum çözümlerin setlerine (PS) yakınsama grafikleri Şekil 14'de sunulmaktadır.



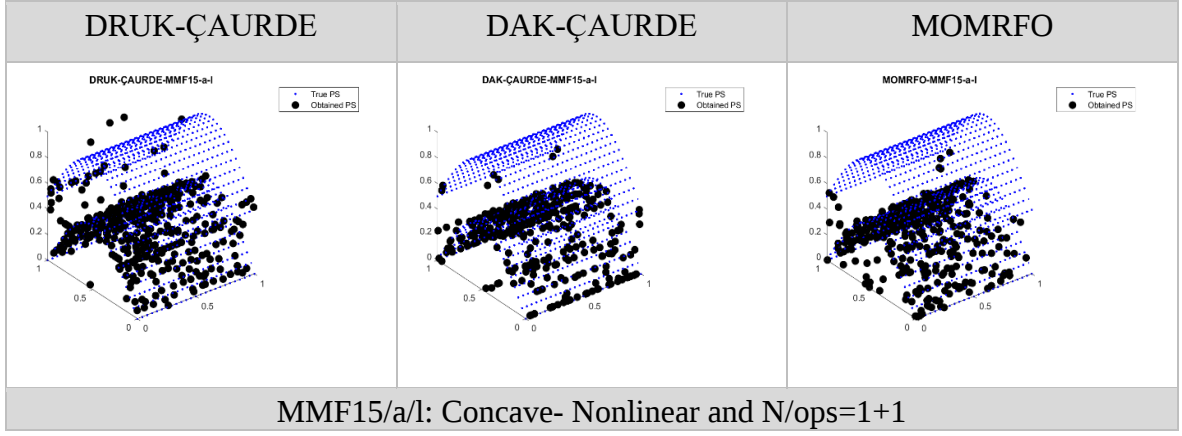
Şekil 14'ün devamı



Şekil 14'ün devamı



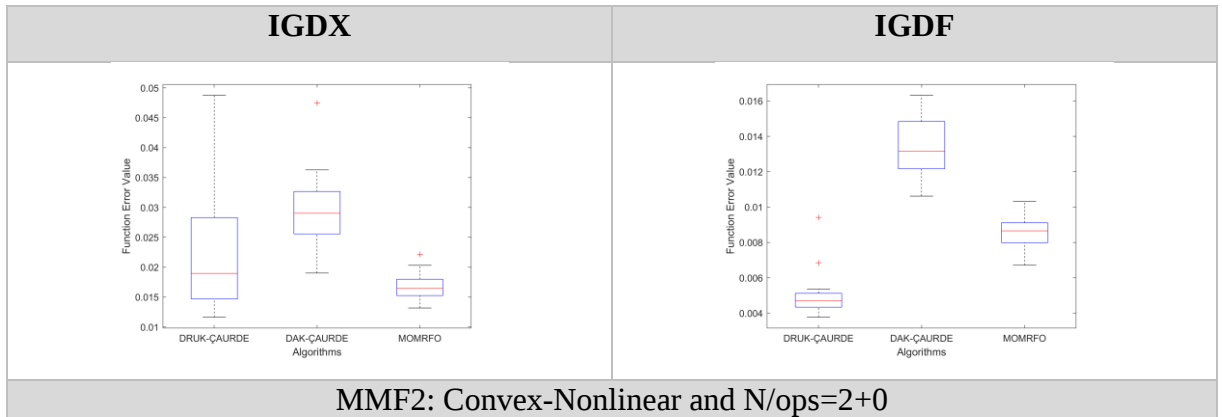
Şekil 14'ün devamı



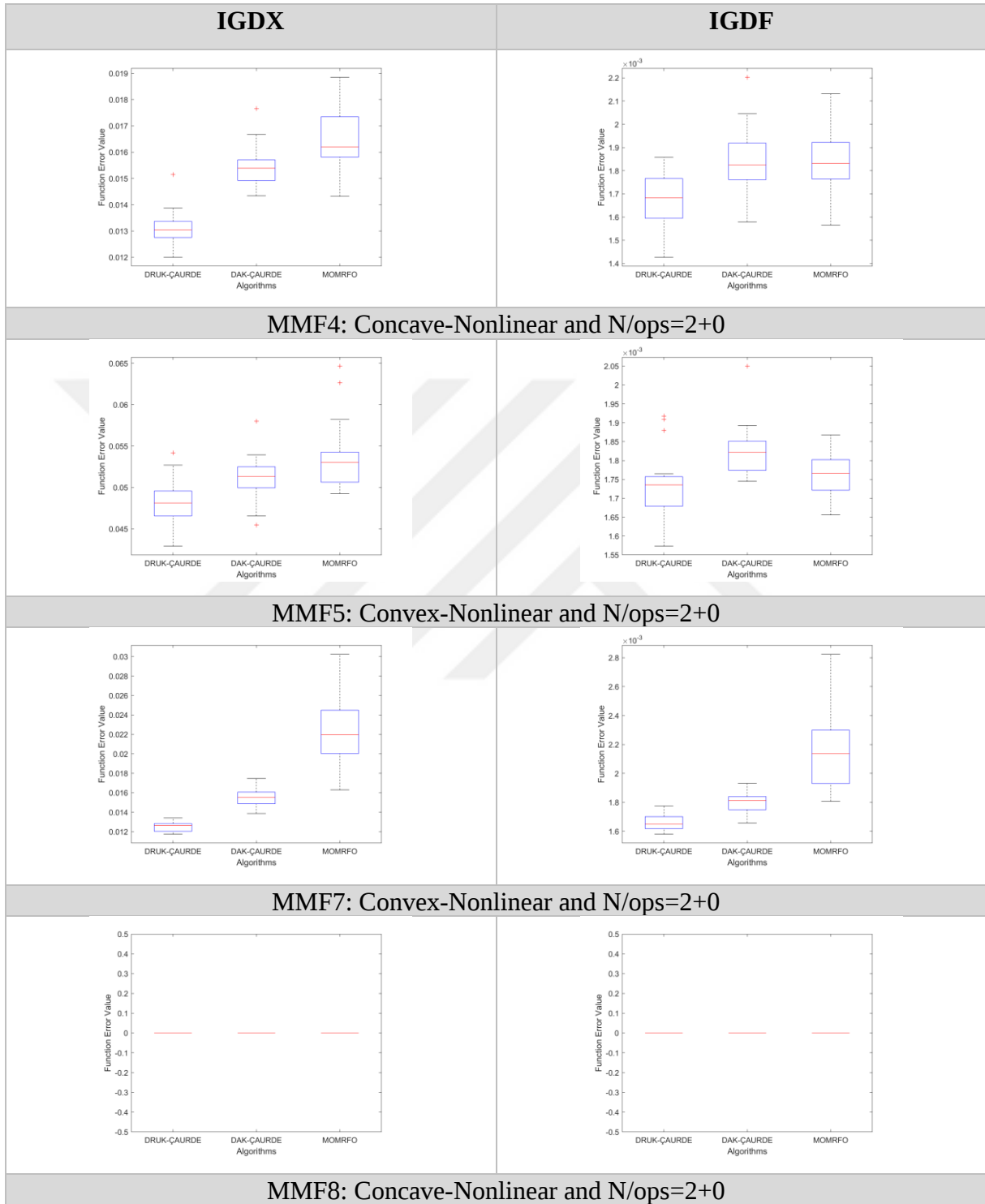
Şekil 14. Friedman skorlarına göre rekabetçi ilk üç algoritmanın PS yakınsama performansları

Şekil 14 incelendiğinde, ilk sırada yer alan DRUK-ÇAURDE'nin rakiplerine kıyasla karar uzayı optimum çözüm setlerin DAK-ÇAURDE ve MOMRFO algoritmalarından daha başarılı bir şekilde bulduğu görülmektedir. Bu sonuç, DRUK-ÇAURDE'nin amaç uzayında olduğu gibi karar uzayında da çeşitliliği sağlama konusunda güçlü bir yeteneğe sahip olduğuna işaret etmektedir.

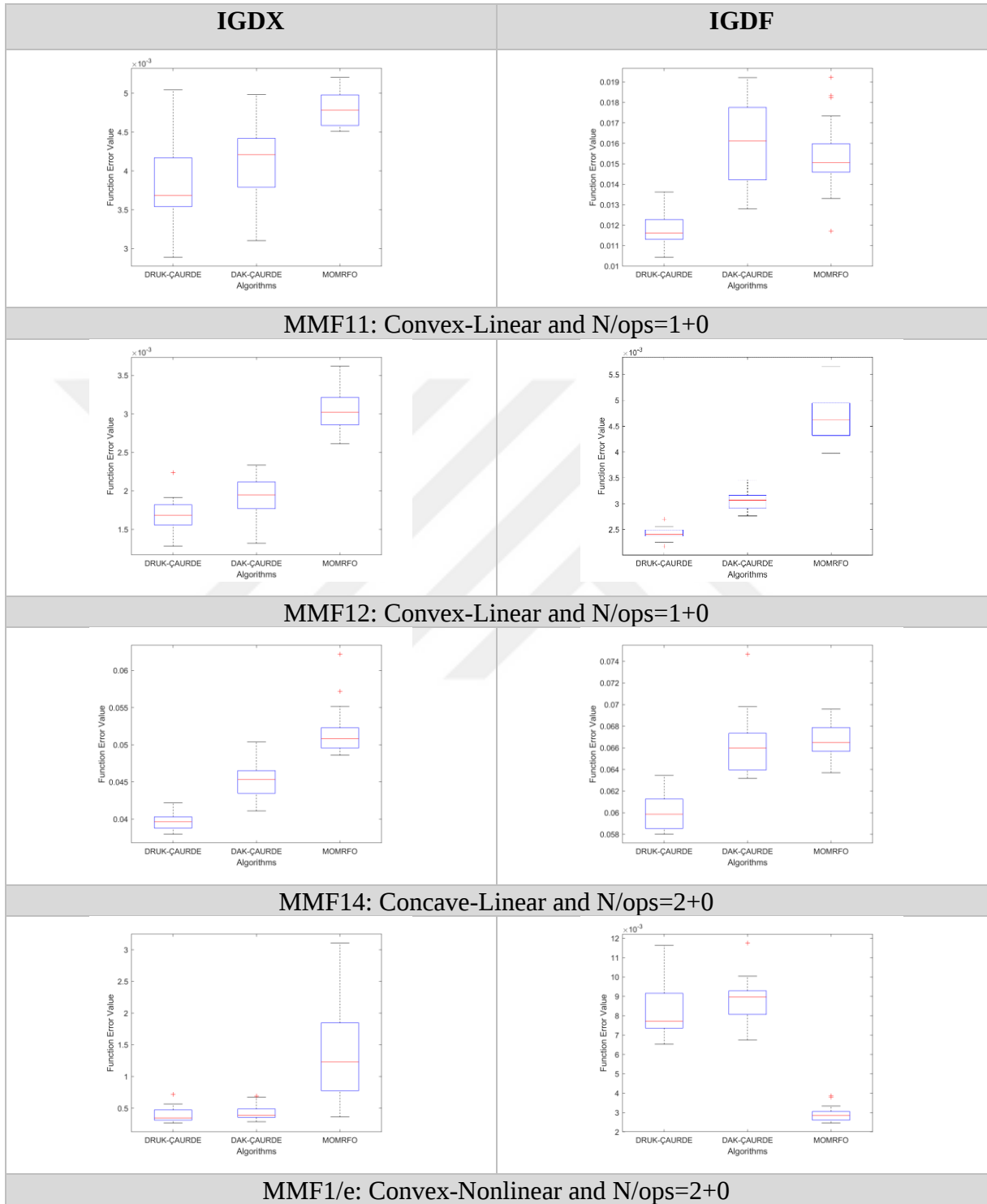
Tablo 3'te sunulan Friedman skorlarına göre ilk üç sırada yer alan algoritmanın IGDX ve IGDF performans metrikleri için değerleri Şekil 15'de sunulmaktadır.



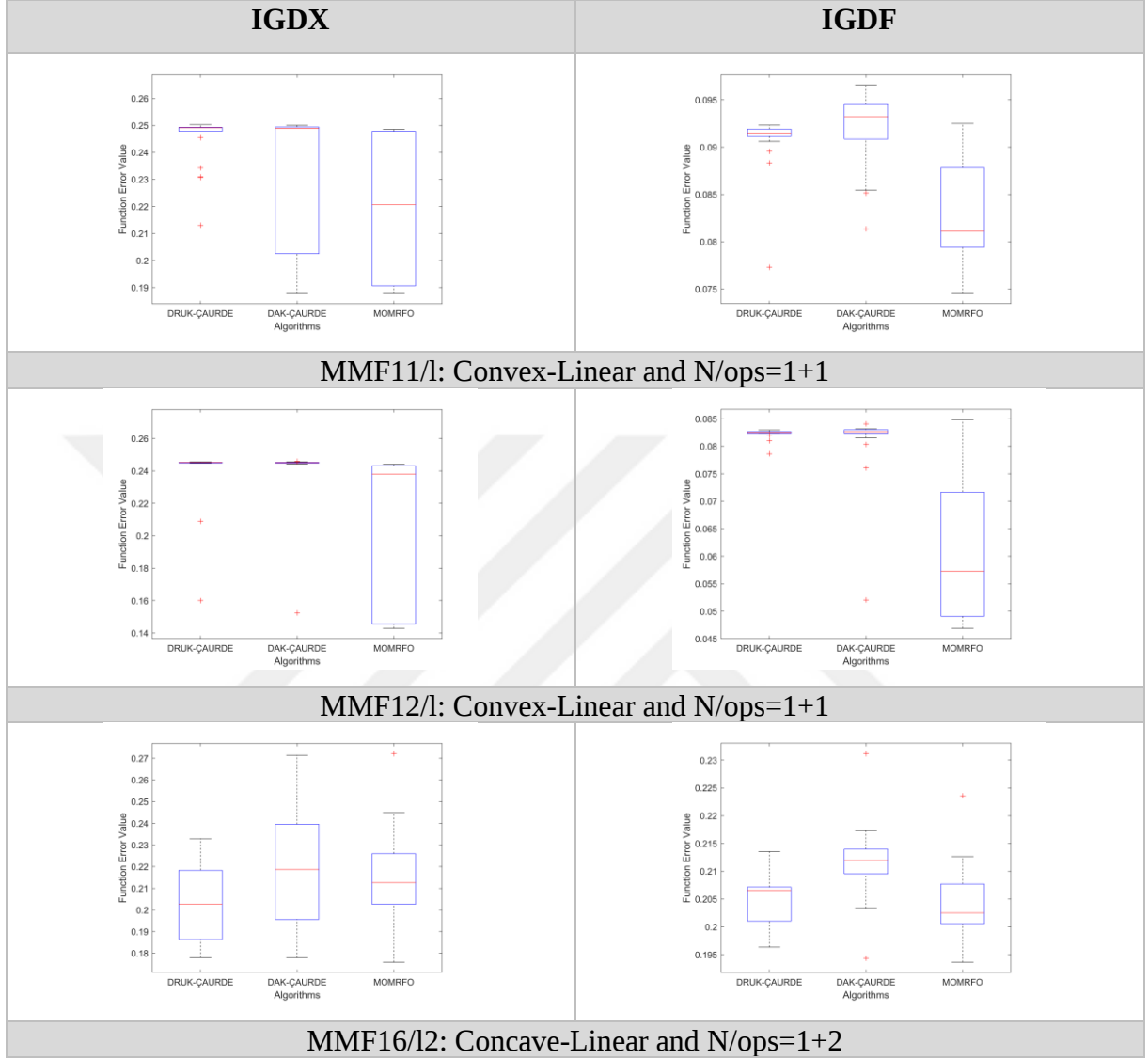
Şekil 15'in devamı



Şekil 15'in devamı



Şekil 15'in devamı



Şekil 15. Friedman skorlarına göre rekabetçi ilk üç algoritmanın PS yakınsama performansları

Şekil 15 incelendiğinde, DRUK-ÇAURDE'nin IGDX ve IGDF metrikleri için on iki problemin sekizinde rakiplerinden daha iyi bir performans sergilediği görülmektedir. Bu problemler sırasıyla MMF2, MMF4, MMF5, MMF7, MMF11, MMF12, MMF14, MMF16/12. MMF1/e, MMF11/l, ve MMF12/l problemlerinde ise MOMRFO algoritmasının IGDX ve IGDF metrikleri için daha iyi değerler elde etmeyi başardığı, MMF8 için üç algoritmanın da optimum performans metrik değerine sahip oldukları görülmektedir. Bu sonuca göre, DRUK tabanlı bir arşiv indirgeme algoritması tasarımında anahtarlar referans

sinyalinin algoritmanın performansı üzerinde önemli bir etkiye sahip olduğu ortaya çıkmaktadır.



4. SONUÇLAR

Bu tez çalışmasında, çok amaçlı evrimsel arama algoritmalarında arşiv indirgeme mekanizmasının tasarımı üzerine kümeleme esaslı yeni bir yöntem önerilmiştir. Önerilen yöntemde arşiv indirgeme süreci dinamik referans uzaylarda kümeleme esaslı olarak gerçekleştirilmiştir. Tezde önerilen yöntemin çok amaçlı optimizasyon çalışmaları açısından orijinal yönü arşiv indirgeme sürecinde kümeleme işleminin sabit bir referans uzayda değil, dinamik anahtarlama mekanizması tarafından belirlenen uzayda gerçekleşmesi olmuştur. Kümeleme sürecinde referans alınan uzayın dinamik bir şekilde değişiyor olması, indirgeme sürecinin işleyişini ve arşiv üzerindeki etkilerini de değiştirmiştir. Bu işleyişe göre, dinamik bir şekilde belirlenen referans uzay sayesinde karar, amaç ve bileşke uzaylardan herhangi birinde arşiv indirgeme işlemi gerçekleştirilebilmektedir. Bu durum, çok amaçlı evrimsel arama sürecinde algoritmanın karar, amaç ya da bileşke uzaydan sadece birinde değil tümünde indirgeme yapmasını ve arama sürecinde sürdürülebilir bir çeşitliliğin elde edilmesini sağlamıştır.

Tez çalışmasında geliştirilen DRUK yönteminin çok amaçlı evrimsel arama sürecindeki performansını araştırmak için çok tipli (multi-modal) yirmi dört adet test probleminin yer aldığı bir set kullanılmıştır. Referans uzay vektörlerini elde etmek için farklı stratejilerin tatbik edildiği bu süreçte, DRUK yönteminin farklı versiyonları geliştirildi. DRUK versiyonlarının tatbik edildiği arşiv indirgeme yönteminin ve standart olarak kullanılan kümeleme esaslı arşiv indirgeme yönteminin (karar ve amaç uzaylarının kullanıldığı) performanslarını araştırmak ve karşılaştırmak için tüm yöntemler ÇAURDE algoritması üzerinde tatbik edildi. Deneysel çalışmalardan elde edilen veriler DRUK yönteminin tatbik edildiği ÇAURDE algoritmasının klasik kümeleme yöntemlerinin tatbik edildiği ÇAURDE algoritmasından daha iyi PS ve PF setleri elde ettiğini gösterdi. Önerilen DRUK yönteminin başarısı, istatistiksel analiz sonuçları ve dört farklı performans metriğinden elde edilen verilerle doğrulandı. DRUK esaslı arşiv indirgeme yönteminin tatbik edildiği ÇAURDE algoritması çok amaçlı evrimsel diğer rakipler karşısında da bariz bir üstünlük sergilemeyi başardı.

Özet olarak, istatistiksel test yöntemlerinden elde edilen sonuçlara göre, bu tez çalışmasında önerilen DRUK esaslı arşiv indirgeme yöntemi başarılı bir şekilde geliştirildi ve ÇAURDE algoritmasının performansı üzerinde kayda değer bir iyileşme elde edildi.

Geliştirilen algoritmanın gelecekte birçok yeni araştırmada kullanılması ve çok amaçlı gerçek dünya optimizasyon problemleri üzerinde de test edilmesi beklenmektedir. DRUK-ÇAURDE algoritmasının elektrik güç sistemleri problemlerinin optimizasyonu için tatbik edilmesine başlanmıştır. Bu araştırmanın kısa süre içerisinde tamamlanması ve nitelikli bir dergide yayına dönüştürülmesi planlanmaktadır.



5. ÖNERİLER

Amaç fonksiyonları çatışma halinde olan problemlerin çözümlenmesinde Pareto-tabanlı baskınlık mekanizmasına sahip kalabalık mesafesi esaslı arşiv indirgeme yönteminin tatbik edildiği çok amaçlı evrimsel arama algoritmaları yaygın bir şekilde kullanılmaktadır. Bu çalışmalarda başlıca araştırma konusu optimum çözümlerin setinin etkili bir şekilde bulunabilmesi için karar ve amaç uzaylarında çeşitliliğin sağlanması ve arama uzayındaki yerel çözüm tuzaklarından kurtulabilecek mekanizmaların geliştirilmesidir. Bu tez çalışmasında arşiv indirgeme mekanizmasının tasarımı üzerine yürütülen araştırmalar, arama sürecinde yerel çözüm tuzaklarından kurtulma konusunda kayda değer bir iyileşme elde edilebileceğini göstermiştir. Tez çalışmasından elde edilen istatistiksel analiz sonuçlarına göre, kümeleme esaslı arşiv indirgeme algoritmasının kalabalık mesafesi esaslı arşiv indirgeme algoritmasına kıyasla arama uzayındaki çeşitliliği sağlama konusunda daha başarılı olduğu görüldü. Bunun yanında, tez çalışmasında ilk defa tatbik edilen dinamik referans uzaylarda kümeleme esaslı arşiv indirgeme algoritmasının geleneksel kümeleme esaslı arşiv indirgeme algoritmalarından daha başarılı olduğu performans metrik sonuçlarından açıkça görüldü. Bu sonuçlara göre, çok amaçlı optimizasyon algoritmalarının tasarımı üzerine gelecekte yürütülecek araştırmalarda DRUK yönteminin tatbik edilmesi ve daha da geliştirilmesi önerilmektedir. Arşiv indirgeme yöntemi olarak DRUK'un kullanılması, bunun yanında algoritmanın rehber mekanizmasının çevresel seçim yöntemleri kullanılarak tasarlanması bir başka araştırma konusu olarak önerilmektedir.

6. KAYNAKLAR

- Alatas, B. (2010). Chaotic harmony search algorithms. *Applied Mathematics and Computation*, s. 2687-2699.
- Alatas, B. (2011). Uniform Big Bang–Chaotic Big Crunch optimization. *Communications in Nonlinear Science and Numerical Simulation*, s. 3696-3703.
- Bao, C., Gao, D., Gu, W., Xu, L., & D.Goodman, E. (2023). A new adaptive decomposition-based evolutionary algorithm for multi- and many-objective optimization. *Expert Systems with Applications*.
- Belhor, M., El-Amraoui, A., Jemai, A., & Delmotte, F. (2023). Multi-objective evolutionary approach based on K-means clustering for home health care routing and scheduling problem. *Expert Systems with Applications*.
- Beume, N., Fonseca, C. M., Lopez-Ibanez, M., Paquete, L., & Vahrenhold, J. (2009). On the Complexity of Computing the Hypervolume Indicator. *IEEE*.
- Chatterjee, S., Roy, S., Das, A. K., Chattopadhyay, S., Kumar, N., & Vasilakos, A. V. (2016). Secure Biometric-Based Authentication Scheme Using Chebyshev Chaotic Map for Multi-Server Environment. *IEEE Transactions on Dependable and Secure Computing*, s. 824-839.
- Çavuş, M., Sezer, A., & Yazıcı, B. (2015). A Simulation Study on Generalized Pareto Mixture Model. *Computational Problems in Science and Engineering*, s. 249-259.
- Deb, K., & Tiwari, S. (2005). Omni-optimizer: A Procedure for Single and Multi-objective Optimization. *Evolutionary Multi-Criterion Optimization*.
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*.
- Duman, S., Akbel, M., & Kahraman, H. T. (2021). Development of the Multi-Objective Adaptive Guided Differential Evolution and optimization of the MO-ACOPF for wind/PV/tidal energy sources. *Applied Soft Computing*.
- Fonseca, C., Paquete, L., & Lopez-Ibanez, M. (2006). An Improved Dimension-Sweep Algorithm for the Hypervolume Indicator. *IEEE*.
- Friedman, M. (1937). The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *Journal of the American Statistical Association*, s. 675-701.
- Gandomi, A., Yang, X.-S., Talatahari, S., & Alavi, A. (2013). Firefly algorithm with chaos. *Communications in Nonlinear Science and Numerical Simulation*, s. 89-98.

- Haraty, R. A., Dimishkieh, M., & Masud, M. (2015). An Enhanced k-Means Clustering Algorithm for Pattern Discovery in Healthcare Data. *International Journal of distributed sensor networks*.
- Hsieh, T.-J. (2023). Performance indicator-based multi-objective reliability optimization for multi-type production systems with heterogeneous machines. *Reliability Engineering & System Safety*.
- Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, s. 651-666.
- Jangir, S. M., Mirjalili, S. Z., Saremi, S., & Trivedi, I. N. (2017). Optimization of problems with multiple objectives using the multi-verse optimization algorithm. *Knowledge-Based Systems*, s. 50-71.
- Kahraman, H. T., Akbel, M., & Duman, S. (2022). Optimization of Optimal Power Flow Problem Using Multi-Objective Manta Ray Foraging Optimizer. *Applied Soft Computing*.
- Kahraman, H. T., Akbel, M., Duman, S., Kati, M., & Sayan, H. H. (2022). Unified space approach-based Dynamic Switched Crowding (DSC): A new method for designing Pareto-based multi/many-objective algorithms. *Swarm and Evolutionary Computation*.
- Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., & Wu, A. Y. (2002). An efficient k-means clustering algorithm: Analysis and. *IEEE transactions on pattern analysis and machine*, s. 881-892.
- Kaur, G., & Arora, S. (2018). Chaotic whale optimization algorithm . *Journal of Computational Design and Engineering*, s. 275–284.
- Khodadadi, N., Abualigah, L., El-Kenawy, E.-S. M., Snasel, V., & Mirjalili, S. (2022). An Archive-Based Multi-Objective Arithmetic Optimization Algorithm for Solving Industrial Engineering Problems. *IEEE Access* .
- Kohli, M., & Arora, S. (2018). Chaotic grey wolf optimization algorithm for constrained optimization problems. *Journal of Computational Design and Engineering*, s. 458–472.
- Koşuta, K. (2018). K-ortalamalar algoritması ile ileriye dönük modellemeler. *Master's thesis, Fen Bilimleri Enstitüsü*, s. 29-24.
- Köse, U. (2017). Yapay zeka tabanlı optimizasyon algoritmaları geliştirilmesi. *Doktora Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü*.
- Kruskal, W. H., & Wallis, W. A. (1952). Use of Ranks in One-Criterion Variance Analysis. *Journal of the American Statistical Association*, s. 583-621.

- Li, X., & Yin, M. (2014). Parameter estimation for chaotic systems by hybrid differential evolution algorithm and artificial bee colony algorithm. *Nonlinear Dynamics*, s. 61–71.
- Liang, J. J., Suganthan, P. N., Qu, B. Y., Gong, D. W., & Yue, C. T. (2019). Problem definitions and evaluation criteria for the CEC 2019 special session on multimodal multiobjective optimization. *Computational Intelligence Laboratory, Zhengzhou University*.
- Liang, J. J., Yue, C. T., & Qu, B. Y. (2016). Multimodal multi-objective optimization: A preliminary study. In *2016 IEEE Congress on Evolutionary Computation*, s. 2454–2461.
- Liang, J., Suganthan, P. N., Qu, B. Y., Gong, D. W., & Yue, C. T. (2019). Problem Definitions and Evaluation Criteria for the CEC 2020 Special Session on Multimodal Multiobjective Optimization. *Zhengzhou University*.
- Liu, Y., Liu, J., Li, T., & Li, Q. (2019). An R2 indicator and weight vector-based evolutionary algorithm for multi-objective optimization. *Soft Computing*.
- Luo, J., Qiqi, L., Yang, Y., Li, X., Chen, M.-R., & Wenming, C. (2017). An artificial bee colony algorithm for multi-objective optimisation. *Applied Soft Computing*.
- Mirjalili, S. (2016). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*.
- Mirjalili, S., Jangir, P., & Saremi, S. (2017). Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems. *Applied Intelligence*.
- Mirjalili, S., Saremi, S., Mirjalili, S. M., & Coelho, L. d. (2016). Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization. *Expert Systems with Applications*, s. 106–119.
- Mohamed, A. W., & Mohamed, A. W. (2019). Adaptive guided differential evolution algorithm with novel mutation for numerical optimization. *International Journal of Machine Learning and Cybernetics*, s. 253–277.
- Nouhi, B., Khodadadi, N., Azizi, M., Talatahari, S., & Gandomi, A. H. (2022). Behnaz Nouhi; Nima Khodadadi; Mahdi Azizi; Siamak Talatahari; Amir H. Gandomi. *IEEE Access*.
- Özkış, A. (2017). Girdap arama ve yapay alg algoritmalarının çok amaçlı optimizasyon problemlerine uyarlanması. *Doctoral dissertation, Doktora Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü*.
- Premkumar, M., Jangir, P., Sowmya, R., Alhelou, H. H., Heidari, A. A., & Chen, H. (2020). MOSMA: Multi-objective slime mould algorithm based on elitist non-dominated sorting. *IEEE Access*.

- Ray, S., & Turi, R. H. (1999). Determination of number of clusters in k-means clustering and application in colour image segmentation. *Proceedings of the 4th international conference on advances in pattern recognition and digital techniques*, s. 137-143.
- Sağ, T. (2015). SÜRÜ ZEKÂSI KULLANARAK RENKLİ GÖRÜNTÜ SEGMENTASYON TEKNİKLERİNİN GELİŞTİRİLMESİ. *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*.
- Savsani, V., & Tawhid, M. A. (2017). Non-dominated sorting moth flame optimization (NS-MFO) for multi-objective problems. *Engineering Applications of Artificial Intelligence*, s. 20-32.
- Sun, Y., Yen, G. G., & Yi, Z. (2018). IGD Indicator-Based Evolutionary Algorithm for Many-Objective Optimization Problems. *IEEE Transactions on Evolutionary Computation*, s. 173-187.
- Sushmita Sharma, N. K., Gharehchopogh, F. S., & Mirjalili, S. (2022). Non-dominated Sorting Advanced Butterfly Optimization Algorithm for Multi-objective Problems. *Journal of Bionic Engineering*, s. 819–843.
- Tavazoei, M. S., & Haeri, M. (2007). Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms. *Applied Mathematics and Computation*, s. 1076-1085.
- Tian, Y., Shao, S., Xie, G., & Zhang, X. (2023). A multi-granularity clustering based evolutionary algorithm for large-scale sparse multi-objective optimization. *Swarm and Evolutionary Computation*.
- Toktas, A., Erkan, U., Ustun, D., & Wang, X. (2023). Parameter optimization of chaotic system using Pareto-based triple objective artificial bee colony algorithm. *Neural Computing and Applications*, s. 13207-13223.
- Wang, F., Li, Y., Zhang, H., Hu, T., & Shen, X. (2019). An adaptive weight vector guided evolutionary algorithm for preference-based multi-objective optimization. *Swarm and Evolutionary Computation*, s. 220-233.
- Xiang, T., Liao, X., & Wong, K.-w. (2007). An improved particle swarm optimization algorithm combined with piecewise linear chaotic map. *Applied Mathematics and Computation*, s. 1637-1645.
- Xie, F., Li, L., Li, L., Huang, Y., & He, Z. (2023). A decomposition-based multi-objective Jaya algorithm for lot-streaming job shop scheduling with variable sublots and intermingling setting. *Expert Systems with Applications*.
- Xiong, Z., Wang, X., Li, Y., Feng, W., & Liu, Y. (2024). A problem transformation-based and decomposition-based evolutionary algorithm for large-scale multiobjective optimization. *Applied Soft Computing*.

- Xu, W., Geng, Z., Zhu, Q., & Gu, X. (2013). A piecewise linear chaotic map and sequential quadratic programming based robust hybrid particle swarm optimization. *Information Sciences*, s. 85-102.
- Yang, X.-S. (2011). Bat algorithm for multi-objective optimisation. *International Journal of Bio-Inspired Computation*, s. 267-274.
- Yang, X.-S., & Deb, S. (2013). Multiobjective cuckoo search for design optimization. *Computers & Operations Research*, s. 1616-1624.
- Yang, X.-S., Karamanoglu, M., & He, X. (2013). Multi-objective Flower Algorithm for Optimization. *Procedia Computer Science*, s. 861-868.
- Yin, F., & Cao, B. (2023). A two-space-decomposition-based evolutionary algorithm for large-scale multiobjective optimization. *Swarm and Evolutionary Computation*.
- Yue, C. T., Qu, B., & Liang, J. (2018). multi-objective particle swarm optimizer using ring topology for solving multimodal multi-objective problems. *IEEE Trans. Evol. Comput.*
- Yue, C. T., Qu, B., Yu, K., Liang, J., & Li, X. (2019). A novel scalable test problem suite for multimodal multiobjective optimization. *Swarm and Evolutionary Computation*, s. 62-71.
- Yue, C., Qu, B., & Liang, J. (2017). A Multiobjective Particle Swarm Optimizer Using Ring Topology for Solving Multimodal Multiobjective Problems. *IEEE Transactions on Evolutionary Computation*.
- Zhang, Q., & Li, H. (2007). MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on evolutionary computation*, s. 712-731.
- Zheng, J., Du, Z., Zou, J., & Yang, S. (2023). A weight vector generation method based on normal distribution for preference-based multi-objective optimization. *Swarm and Evolutionary Computation*.
- Zhou, A., Zhang, Q., & Jin, Y. (2009). Approximating the set of pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm. *IEEE Trans. Evol. Comput.*, s. 1167–1189.
- Zhou, S., Chen, Z., Li, Q., Gu, M., Bao, Z., He, W., & Sheng, W. (2023). A Multi-Objective Evolutionary Algorithm With Hierarchical Clustering-Based Selection. *Institute of Electrical and Electronics Engineers*.
- Zhu, Y., Qin, Y., Yang, D., Xu, H., & Zhou, H. (2023). An enhanced decomposition-based multi-objective evolutionary algorithm with a self-organizing collaborative scheme. *Expert Systems with Applications*.
- Zitzler, E., Deb, K., & Thiele, L. (2001). Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *IEE Trans On Evolutionary Computation*, s. 173-195.

- Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm. *TIK report*.
- Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Trans. Evol. Comput.*, s. 117–132.



ÖZGEÇMİŞ

Mustafa AKBEL, İlköğrenimini Cebeci Sultançiftliği İlköğretim Okulu'nda bitirdikten sonra lise eğitimini 2011 yılında İnönü Mesleki Ve Teknik Anadolu Lisesi'nde tamamladı. 2015-2016 öğretim yılında Karadeniz Teknik Üniversitesi Of Teknoloji Fakültesinde Yazılım Mühendisliği bölümünde lisans eğitimine başladı. 2020 yılında bu bölümden mezun oldu. Aynı yıl Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yazılım Mühendisliği Anabilim dalında tezli yüksek lisans programına başladı. 2024 yılında “Çok Amaçlı Optimizasyon Algoritmalarının Tasarımı İçin Kümeleme Esaslı Yeni Bir Arşiv İndirgeme Yönteminin Geliştirilmesi Ve Uygulanması” başlıklı yüksek lisans tezini sundu. Özel sektörde yazılım mühendisi olarak çalışmaktadır. Mustafa AKBEL yüksek lisans tez dönemi boyunca çok amaçlı optimizasyon algoritmalarının tasarımı konusunda tez danışmanı ile birlikte araştırmalar yürüttü. Bu araştırmalar neticesinde çok amaçlı evrimsel arama algoritmalarında arşiv indirgeme mekanizmasının tasarımı için yeni bir yöntem geliştirdiler. Geliştirdikleri bu yöntem JCR kategorisi “Computer Science, Theory & Methods” olan ve bu kategoride dünya sıralamasında en iyi 111 bilimsel dergi arasında 7. Sırada yer alan “Swarm and Evolutionary Computation” isimli dergide “Unified space approach-based Dynamic Switched Crowding (DSC): A new method for designing Pareto-based multi/many-objective algorithms” başlıklı makale olarak yayınlanmıştır. Mustafa AKBEL'in ortak yazarları arasında yer aldığı SCI indeksli Q1 çeyreğindeki bilimsel dergilerdeki yayınlarının referansları aşağıda verilmiştir.

Kahraman, H. T., Akbel, M., Duman, S., Kati, M., & Sayan, H. H. (2022). Unified space approach-based Dynamic Switched Crowding (DSC): A new method for designing Pareto-based multi/many-objective algorithms. *Swarm and Evolutionary Computation*, 75, 101196.

Kahraman, H. T., Akbel, M., & Duman, S. (2022). Optimization of optimal power flow problem using multi-objective manta ray foraging optimizer. *Applied Soft Computing*, 116, 108334.

Duman, S., Akbel, M., & Kahraman, H. T. (2021). Development of the multi-objective adaptive guided differential evolution and optimization of the MO-ACOPF for wind/PV/tidal energy sources. *Applied Soft Computing*, 112, 107814.

Ozkaya, B., Kahraman, H. T., Duman, S., Guvenc, U., & Akbel, M. (2024). Combined heat and power economic emission dispatch using dynamic switched crowding based multi-objective symbiotic organism search algorithm. *Applied Soft Computing*, 151, 111106.

