



**T.C.**  
**KONYA TEKNİK ÜNİVERSİTESİ**  
**LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**



**ÇOK AMAÇLI OPTİMİZASYON  
PROBLEMLERİNİN ÇÖZÜMÜNDE PARETO  
TABANLI YENİ YAKLAŞIMLAR**

Mustafa ALTIOK

**DOKTORA TEZİ**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Temmuz-2023**  
**KONYA**  
**Her Hakkı Saklıdır**

## TEZ KABUL VE ONAYI

Mustafa ALTIOK tarafından hazırlanan “**Çok Amaçlı Optimizasyon Problemlerinin Çözümünde Pareto Tabanlı Yeni Yaklaşımlar**” adlı tez çalışması 05/07/2023 tarihinde aşağıdaki jüri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda DOKTORA TEZİ olarak kabul edilmiştir.

### Jüri Üyeleri

### İmza

#### Başkan

Prof. Dr. Halife KODAZ

.....

#### Danışman

Doç. Dr. Mesut GÜNDÜZ

.....

#### Üye

Doç. Dr. Ömer Kaan BAYKAN

.....

#### Üye

Dr. Öğr. Üyesi Ayşe Merve ACILAR

.....

#### Üye

Dr. Öğr. Üyesi Murat KARAKOYUN

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Mevlüt UYAN  
Enstitü Müdürü

## TEZ BİLDİRİMİ

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

## DECLARATION PAGE

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

İmza

Mustafa ALTIOK

Tarih:05/07/2023

# ÖZET

## DOKTORA TEZİ

### ÇOK AMAÇLI OPTİMİZASYON PROBLEMLERİNİN ÇÖZÜMÜNDE PARETO TABANLI YENİ YAKLAŞIMLAR

Mustafa ALTIOK

Konya Teknik Üniversitesi  
Lisansüstü Eğitim Enstitüsü  
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Mesut GÜNDÜZ

2023, 133 Sayfa

Jüri

Prof. Dr. Halife KODAZ

Doç. Dr. Mesut GÜNDÜZ

Doç. Dr. Ömer Kaan BAYKAN

Dr. Öğr. Üyesi Ayşe Merve ACILAR

Dr. Öğr. Üyesi Murat KARAKOYUN

Gerçek dünya problemlerinin tasarımında her amacın kusursuz olması istenir, örneğin tasarlanan bir aracın maliyetinin düşük olması beklenirken hızının veya güvenliğinin çok yüksek olması istenilebilmektedir. Halbuki kaynaklar sınırlıdır. Neredeyse bütün mühendislik problemlerinde böyle çelişkiler görmek mümkündür. Bu ve benzer problemlerin üstesinden gelmek için iki ya da daha fazla amacı eş zamanlı olarak optimize edebilecek çok amaçlı problem çözme yaklaşımları geliştirilmiştir. Bunun yanında, NFL (*No Free Lunch*) teoremine göre bir algoritma ne kadar iyi olursa olsun, her problemde mükemmel sonucu garanti edemeyeceği için her zaman yeni problem çözme yaklaşımlarına ihtiyaç vardır.

Tez çalışmasında, iki farklı problem incelenmiş, bu problemleri çözmek için hipotezler sunulmuş ve bu hipotezlere dayanarak çok amaçlı çözüm yaklaşımları geliştirilmiştir. Her iki problem içinde en köklü çok amaçlı çözüm yaklaşımlardan olan Pareto tabanlı algoritmalar tasarlanmış ve her iki çalışmada da farklı bir sistem, amaç fonksiyonu yerine kullanılmıştır. Bu amaçla, çok amaçlı algoritma ile bir sistem hibrit olarak beraber çalıştırılmıştır. Bu hibrit yaklaşımların, özellikle bir amaç fonksiyonu çıkarmanın zor olduğu problemler için yol gösterici nitelikte olduğu düşünülmektedir.

İlk çalışmada, derin öğrenme alanında yaygın olarak kullanılan bir yapay sinir ağı mimarisi olan, CNN (*Convolutional Neural Network*) tabanlı bir sistemin hiper parametre optimizasyonu ele alınmıştır. Bu çalışmada, ilk zorluk, meta-sezgisel optimizasyon türü çözüm yaklaşımlarının temelde deneme yanılmaya ve rastsallığa dayanması, yani yüzbinlerce deneme gerektirmesidir. Bu tür problemler polinomal zamanda çözülemeyen problemlerin sınıflarına girebilmektedirler. Diğer yandan, derin öğrenmenin önemli zorluklarından olan zaman zaman maliyetinin yüksek olmasıdır. Bu iki yaklaşımın birlikte kullanılması durumunda zaman maliyeti büyük ölçüde artacağından; ilgili problem için az sayıda denemeyle, birden fazla amaç için eş zamanlı, kabul edilebilir sonuçlar üretebilen bir optimizasyon algoritması geliştirmek hedeflenmiştir. Böylece F-MODE (*Fast Multi-Objective Differential Evolution*) isminde hızlı yakınsamaya sahip bir algoritma önerilmiş, literatürde iyi bilinen bazı algoritmalarla test fonksiyonları üzerinde kıyaslanmış ve çeşitli istatistik testlere tabi tutulduktan sonra CNN tabanlı sistemin hiper parametre optimizasyonu için kullanılmıştır. Sonuç olarak hiper parametrelerinden oluşan Pareto-optimal'a yakın veriler elde edilmiştir.

İkinci çalışmada ise asfalt karışımlarının oranlarını optimize etmek için D-MOGA (*Dynamic Multi-Objective Genetic Algorithm*) adında farklı bir algoritma geliştirilmiştir. Daha önce başka bir çalışma

kapsamında laboratuvar ortamında yapılan 42 adet asfalt karışımı deneyleri bu çalışmada kullanılmıştır. Bu deneylerin her biri beş girdiye ve üç çıktıya sahip verilerden oluşmaktadır. İlk önce bu veriler kullanılarak kural tabanlı bir bulanık mantık (*Fuzzy logic, FL*) sistemi tasarlanmış ve böylece, farklı girdiler için gerçek deneylere ihtiyaç duymadan çıktı sonuçlarına bakılabilecek bir sistem elde edilmiştir. Daha sonra dengeli ve rekabetçi bir algoritma olan D-MOGA geliştirilmiş ve çeşitli istatistik testler üzerinde literatürde bilinen başarılı algoritmalarla kıyaslamalar yapılmıştır. Son olarak geliştirilen bulanık mantık sistemi bir amaç fonksiyonu gibi kullanılarak D-MOGA algoritmasına entegre edilmiş ve asfalt karışım problemi için Pareto-optimale yakın veriler bulunmuştur.

Çalışmanın son bölümünde test fonksiyonları kullanılarak F-MODE ve D-MOGA algoritmaları arasında karşılaştırmalı bir analiz yapılmıştır. Başlangıçta, testler birinci algoritmaya (F-MODE) özgü koşullara göre yapılmıştır. Daha sonra aynı testler ikinci algoritmaya (D-MOGA) özgü koşullar dikkate alınarak uygulanmıştır. Bu karşılaştırmalı analiz, farklı senaryolar ve koşullar altında her bir algoritmanın performansını ve etkinliğini değerlendirmeyi amaçlamıştır. Sonuç olarak iki ayrı problem çok amaçlı optimizasyon problemi kapsamında ele alınmış ve bu problemleri çözmek için yaklaşımlar geliştirilmiştir.

**Anahtar Kelimeler:** Çok amaçlı optimizasyon, Pareto taban, mühendislik problemleri, meta-sezgisel yaklaşımlar, F-MODE, D-MOGA, MODE-CNN, MOGA-HMA



## **ABSTRACT**

### **Ph.D THESIS**

## **NEW APPROACHES ON PARETO BASED FOR SOLVING MULTI-OBJECTIVE OPTIMIZATION PROBLEMS**

**Mustafa ALTIOK**

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

**Advisor: Assoc. Prof. Dr. Mesut GÜNDÜZ**

**2023, 133 Pages**

### **Jury**

**Prof. Dr. Halife KODAZ  
Assoc. Prof. Dr. Mesut GÜNDÜZ  
Assoc. Prof. Dr. Ömer Kaan BAYKAN  
Asst. Prof. Dr. Ayşe Merve ACILAR  
Asst. Prof. Dr. Murat KARAKOYUN**

In the design of real-world problems, it is desired that every objective be flawless; for instance, while the cost of a designed vehicle is expected to be low, its speed and safety are also desired to be high. Nevertheless, it is important to acknowledge the inherent limitations imposed by finite resources. Such inherent contradictions can be observed in nearly all engineering problems. To address and surmount these challenges, multi objective problem-solving approaches have been developed, enabling the simultaneous optimization of two or more objectives. Furthermore, regardless of the efficacy of an algorithm, the pursuit of new problem-solving approaches remains imperative, as no algorithm can guarantee flawless outcomes for every problem.

The thesis study focused on the examination of two distinct problems, with the formulation of hypotheses aimed at resolving these issues. Subsequently, multi-objective solution approaches were developed based on these hypotheses. In both studies, Pareto-based algorithms, renowned as one of the most established multi-objective solution approaches, were employed. Additionally, a distinct system was utilized in lieu of the objective function for both problems. In essence, a hybrid system was implemented, combining the multi-objective algorithm with another system to work in tandem and achieve multiple objectives effectively. The hybrid approaches prove particularly beneficial for problems in which deriving a single objective function is challenging or impractical.

In the initial study, the focus was on hyperparameter optimization of a system developed based on CNN, which is a commonly used artificial neural network architecture in the field of deep learning. In this study, one of the primary challenges lies in the fact that optimization-based solution approaches primarily rely on trial and error and incorporate an element of randomness. These types of problems can fall into classes of problems that cannot be solved in polynomial time. Consequently, a substantial number of iterations, often reaching hundreds of thousands, are required to achieve satisfactory results. Conversely, one of the major obstacles encountered in deep learning is the significant time cost associated with the training and inference processes. The objective of the research is to devise an algorithm capable of yielding satisfactory outcomes with a reduced number of trials for the given problem. This approach is pursued to mitigate the substantial increase in time costs that would ensue if both methods were employed simultaneously. In light of this objective, a fast convergence algorithm called F-MODE was proposed. This algorithm was subsequently compared with several established algorithms in the literature using test functions. Afterwards, it has been utilized for hyperparameter optimization of the CNN-based system. As a result, data points that are close to the Pareto-optimal set of hyperparameters have been identified.

In the second study, a distinct algorithm called D-MOGA was developed specifically to optimize the ratios of asphalt mixtures. This algorithm aimed to find optimal combinations of asphalt mixture ratios considering multiple objectives. This study utilized forty-two asphalt mixture experiments conducted in a laboratory environment, which were originally conducted as part of another research investigation. These experimental data served as the basis for the analysis and optimization process in the current study. Each of these experiments comprises data characterized by five input variables and three corresponding output variables. Initially, a rule-based fuzzy logic system was developed utilizing the available data. This system enabled the generation of output results for various input combinations without the necessity of conducting real experiments. Subsequently, a balanced and competitive algorithm called D-MOGA was developed, specifically tailored for the optimization of asphalt mixture ratios. This algorithm was subject to rigorous comparisons with established and successful algorithms found in the literature, employing various statistical tests. Lastly, the fuzzy logic system developed earlier was employed as an objective function and seamlessly integrated into the D-MOGA algorithm. By utilizing this integrated approach, optimal Pareto solutions for the asphalt mixture problem were determined, providing a range of ideal solutions that balance multiple objectives effectively.

In the final part of the study, a comparative analysis was conducted between F-MODE and D-MOGA algorithms using test functions. Initially, the test functions were evaluated based on the conditions specific to the first algorithm (F-MODE). Subsequently, the same test functions were examined considering the conditions specific to the second algorithm (D-MOGA). This comparative analysis aimed to assess the performance and effectiveness of each algorithm under different scenarios and conditions. As a result, two distinct problems have been addressed within the scope of multi-objective optimization problem, and approaches have been developed to solve these problems.

**Keywords:** Multi-objective optimization, Pareto based, engineering problems, metaheuristic approaches, F-MODE, D-MOGA, MODE-CNN, MOGA-HMA

## ÖNSÖZ

Öncelikle, bu çalışmayı başarıyla tamamlamamda beni muvaffak eden Rabbime sonsuz şükranlarımı sunuyorum. Tez çalışmasında ve her konuda bana yardımcı olan ilk danışman hocam Doç. Dr. Barış KOÇER'i rahmetle yad ediyorum. Çalışmalarına fikirleri ile yön veren danışman hocam Doç. Dr. Mesut GÜNDÜZ'e, yapıcı eleştirilerinden dolayı tez komitesi üyesi Doç. Dr. Ömer Kaan BAYKAN'a ve açık fikirliliği ve yol göstericiliği ile Dr. Öğr. Üyesi Ayşe Merve ACILAR hocalarıma teşekkür ediyorum.

Çalışmalarım ne zaman tıkanırsa, canı gönülden, severek yardımcı olan değerli tecrübeleri ile bana yol gösteren Dr. Öğr. Üyesi Ahmet ÖZKİŞ'a teşekkür ediyorum.

Her durumda desteklerini benden esirgemeyen Dr. Öğr. Üyesi Özkan İNİK ve Arş. Gör. Ali AKDAĞ'a teşekkür ediyorum.

Araştırma görevlisi olarak çalıştığım sürece akademik gelişimime katkısı olan bütün KTÜN Bilgisayar Mühendisliği ve TOGÜ Bilgisayar Mühendisliğinin değerli öğretim elemanlarına teşekkür ediyorum.

Hayatımın her döneminde bana destek olan annem Canan ALTIOK ve sırtında taş taşıyarak bizi büyüten rahmetli babam Yaşar ALTIOK'a ve her zaman yanımda olan kardeşlerime teşekkür ediyorum.

Çalışmalarımda en büyük motivasyon kaynağım olan sevgili kızlarım Ahsen ALTIOK ve Saliha Sare ALTIOK'a ve tez çalışmalarımı yürüttüğüm dönemde fedakarane bir şekilde kendisini ailesine adayan sevgili eşim Nazik ALTIOK'a canı gönülden teşekkür ediyorum.

Mustafa ALTIOK  
KONYA-2023



# İÇİNDEKİLER

ÖZET .....	iv
ABSTRACT.....	vi
ÖNSÖZ .....	viii
İÇİNDEKİLER.....	ix
ŞEKİL LİSTESİ.....	xi
ÇİZELGE LİSTESİ.....	xiii
SİMGELER VE KISALTMALAR.....	xiv
<b>1. GİRİŞ .....</b>	<b>1</b>
1.1. Tezin Amacı ve Literatüre Katkıları .....	5
1.2. Tezin Organizasyonu .....	7
<b>2. KAYNAK ARAŞTIRMASI .....</b>	<b>8</b>
2.1. Çok Amaçlı Optimizasyon Algoritmaları ve Kaynak Araştırması .....	8
2.2. CNN Tabanlı Modellerin Hiper Parametre Optimizasyonu ve Kaynak Araştırması.....	10
2.3. Sıcak Asfalt Karışım Problemi ve Kaynak Araştırması .....	12
<b>3. MATERYALLER VE YÖNTEMLER .....</b>	<b>15</b>
3.1. Optimizasyon Nedir? .....	15
3.2. Çok Amaçlı Optimizasyon Temel Bilgiler .....	15
3.2.1. Çok amaçlı optimizasyon problemi .....	15
3.2.2. Amaç fonksiyonları.....	16
3.2.3. Karar değişkenleri.....	16
3.2.4. Problem kısıtları.....	17
3.2.5. Çok amaçlı ve tek amaçlı optimizasyon problemlerinin farkları.....	17
3.3. Pareto Tabanlı Çok Amaçlı Optimizasyon Algoritmaları .....	18
3.3.1. Hızlı-baskın olmayan sıralama operatörü .....	19
3.3.2. Kalabalıklık mesafesi operatörü .....	22
3.3.3. Seçim operatörü .....	23
3.3.4. Çaprazlama operatörü .....	24
3.3.5. Mutasyon operatörü .....	25
3.4. Kullanılan Problem Setleri.....	26
3.4.1. ZDT problem seti.....	26
3.4.2. DTLZ problem seti .....	26
3.4.3. WFG problem seti.....	26
3.4.4. Klasik problemler .....	27
3.5. Bu Çalışmada Kullanılan Metrikler .....	27
3.5.1. Hiperküp metriği.....	28
3.5.2. Nesilsel mesafe .....	29

3.5.3. Ters nesil mesafe .....	30
3.6. Performans Göstergeleri .....	30
3.6.1. Ortalama ve standart sapma .....	31
3.6.2. Pareto-optimal yüzeyler .....	31
3.6.3. Kutu grafikleri.....	32
3.6.4. Wilcoxon testi .....	33
<b>4. ÖNERİLEN YÖNTEMLER VE DENEYSEL ÇALIŞMALAR .....</b>	<b>35</b>
4.1. Hiper Parametre Optimizasyonu Problemi İçin F-MODE Yaklaşımı .....	35
4.1.1. Problemin tanımı.....	35
4.1.2. Çalışmanın literatüre katkıları .....	40
4.1.3. F-MODE algoritması .....	40
4.1.4. Önerilen algoritmanın hiper parametre optimizasyonuna uygulanması .....	58
4.2. Asfalt Karışımı Problemi İçin D-MOGA Yaklaşımı .....	63
4.2.1. Problemin tanımı.....	63
4.2.2. Motivasyon ve katkılar .....	66
4.2.3. Asfalt karışım deneyi .....	68
4.2.4. Bulanık mantık.....	71
4.2.5. D-MOGA algoritması .....	77
4.2.6. Önerilen algoritmanın asfalt karışımı problemine uyarlanması.....	98
4.3. F-MODE ve D-MOGA Karşılaştırılması. ....	102
4.4. F-MODE ve D-MOGA'nın Literatürdeki Bazı Güncel Algoritmalarla Karşılaştırılması. ....	106
<b>5. SONUÇLAR VE ÖNERİLER .....</b>	<b>108</b>
5.1. Sonuçlar .....	108
5.2. Öneriler .....	111
<b>KAYNAKLAR .....</b>	<b>113</b>
<b>EKLER .....</b>	<b>120</b>
EK-1.1. 5,000 maksFes F-MODE ve D-MOGA Pareto-optimum yüzeyler .....	120
EK-1.2. 5,000 maksFes F-MODE ve D-MOGA Kutu grafikler .....	125
EK-2.1. 30,000 maksFes F-MODE ve D-MOGA Pareto-optimum yüzeyler .....	128
EK-2.2. 30,000 maksFes F-MODE ve D-MOGA Kutu grafikler .....	133

## ŞEKİL LİSTESİ

Şekil 3. 1. Hızlı-bastırılmayan sıralama yaklaşımın sözde kodu (Özkış, 2017).....	21
Şekil 3. 2. Hızlı-bastırılmayan sıralama yaklaşımında çözümlerin yüzeylere ayrılması .....	22
Şekil 3. 3. Kalabalıklık mesafesi işlemi için 4 numaralı çözüm ve komşuları.....	23
Şekil 3. 4. Yakınsama ve iyi çeşitlilik, kötü çeşitlilik temsili gösterimi .....	28
Şekil 3. 5. Hiperküp metriğinin hesaplanmasının temsili gösterimi.....	29
Şekil 3. 6. Örnek Pareto-optimal yüzeyler (Altiok ve ark., 2023).....	32
Şekil 3. 7. Kutu grafiği temsili gösterimi .....	33
Şekil 4. 1. CNN temelli segmentasyon yönteminin ana çatısı (İnik ve ark., 2021).....	37
Şekil 4. 2. Farklı <i>GS</i> değerleri için alınan yamalar (İnik ve ark., 2021).....	38
Şekil 4. 3. Farklı <i>PA</i> değerleri (İnik ve ark., 2021) .....	38
Şekil 4. 4. MODE-CNN algoritmasının CNN tabanlı sistemin hiper parametre optimizasyonunda uygulanması.....	39
Şekil 4. 5. Temsili rulet tekerliği seçim yöntemi.....	42
Şekil 4. 6. Çözümlerin temsili kalite skorları .....	43
Şekil 4. 7. F-MODE akış diyagramı .....	45
Şekil 4. 8. F-MODE ve diğer algoritmalar tarafından ZDT1-3, DTLZ1-2 problemleri için üretilen Pareto-optimal yüzeyler.....	51
Şekil 4. 9. F-MODE ve diğer algoritmalar tarafından DTLZ3-7 problemleri için üretilen Pareto-optimal yüzeyler .....	52
Şekil 4. 10. F-MODE ve diğer algoritmalar tarafından WFG1-3, Fonseca problemleri için üretilen Pareto-optimal yüzeyler.....	53
Şekil 4. 11. F-MODE ve diğer algoritmalar tarafından Kursawe, Schaffer problemleri için üretilen Pareto-optimal yüzeyler.....	54
Şekil 4. 12. F-MODE ve diğer algoritmaların ZDT1-3, DTLZ1-5 problemleri için ürettiği performans metriklerinin kutu grafikleri.....	56
Şekil 4. 13. F-MODE ve diğer algoritmaların DTLZ6-7, WFG1-3, Fonseca, Kursawe, Schaffer problemleri için ürettiği performans metriklerinin kutu grafikleri.....	57
Şekil 4. 14. Üst sıra, parametrelerin optimizasyonu için kullanılan resimlerken; alt sıra tespit edilmesi gereken bölgelerdir ( <i>ground truth images</i> ).....	58
Şekil 4. 15. MODE-CNN ile hesaplanan segmentasyon kaybı ve test süresi için elde edilen Pareto cepheleri.....	59
Şekil 4. 16. MODE-CNN ile elde edilen sonuçlar: (a) girdi görüntüleri, (b) girdi görüntülerinin temel gerçeği, (c) minimum segmentasyon kaybıyla elde edilen ile elde edilen görüntü ile minimum test süresi.....	60
Şekil 4. 17. (a) Atık mermer alanı, (b) Atık beton alanı.....	68
Şekil 4. 18. Agregat derecelendirme eğrisi.....	69
Şekil 4. 19. Bulanık mantığın temsili akış diyagramı.....	72
Şekil 4. 20. FL modeli giriş ve çıkışları .....	73
Şekil 4. 21. FL modeli MS değerlerinin sınıflandırılması.....	73
Şekil 4. 22. MS için deneysel ve tahmini veriler arasındaki regresyon ilişkileri .....	76
Şekil 4. 23. AVC için deneysel ve tahmini veriler arasındaki regresyon ilişkileri.....	76
Şekil 4. 24. VFA için deneysel ve tahmini veriler arasındaki regresyon ilişkileri .....	77
Şekil 4. 25. D-MOGA akış diyagramı .....	79
Şekil 4. 26. D-MOGA ve diğer algoritmalar tarafından ZDT1-3, DTLZ1-2 problemleri için üretilen Pareto-optimal yüzeyler.....	89
Şekil 4. 27. D-MOGA ve diğer algoritmalar tarafından DTLZ3-7 problemleri için üretilen Pareto-optimal yüzeyler.....	90

<b>Şekil 4. 28.</b> D-MOGA ve diğer algoritmalar tarafından WFG1-4 problemleri için üretilen Pareto-optimal yüzeyler.....	91
<b>Şekil 4. 29.</b> D-MOGA ve diğer algoritmalar tarafından WFG5-8 problemleri için üretilen Pareto-optimal yüzeyler.....	92
<b>Şekil 4. 30.</b> D-MOGA ve diğer algoritmalar tarafından WFG9, Kursawe, Schaffer problemleri için üretilen Pareto-optimal yüzeyler.....	93
<b>Şekil 4. 31.</b> D-MOGA ve diğer algoritmaların ZDT1-3, DTLZ1-3 problemleri için ürettiği performans metriklerinin kutu grafikleri.....	95
<b>Şekil 4. 32.</b> D-MOGA ve diğer algoritmaların DTLZ4-7, WFG1-2 problemleri için ürettiği performans metriklerinin kutu grafikleri.....	96
<b>Şekil 4. 33.</b> D-MOGA ve diğer algoritmaların WFG3-8 problemleri için ürettiği performans metriklerinin kutu grafikleri .....	97
<b>Şekil 4. 34.</b> D-MOGA ve diğer algoritmaların WFG9, Kursawe, Schaffer problemleri için ürettiği performans metriklerinin kutu grafikleri.....	98
<b>Şekil 4. 35.</b> MOGA-HMA algoritmasının HMA problemine uyarlanması.....	99



## ÇİZELGE LİSTESİ

Çizelge 3. 1. Kalabalıklık mesafesi sözde kodu (Deb ve ark., 2002).....	23
Çizelge 3. 2. Örnek bir ortalama, standart sapma gösterimi.....	31
Çizelge 4. 1. HV metriğine göre algoritmaların performansları.....	47
Çizelge 4. 2. IGD metriğine göre algoritmaların performansları .....	48
Çizelge 4. 3. HV Metriğine göre Wilcoxon sıralı toplam testi sonuçları .....	49
Çizelge 4. 4. IGD Metriğine göre Wilcoxon sıralı toplam testi sonuçları.....	49
Çizelge 4. 5. MODE-CNN ile 100 popülasyon için hesaplanan çözüm uzayı.....	61
Çizelge 4. 6. Laboratuvar deney sonuçları (Alakara ve Agaoglu, 2020) .....	65
Çizelge 4. 7. Agregaların fiziksel özellikleri.....	69
Çizelge 4. 8. Kullanılan bitümün fiziksel özellikleri.....	69
Çizelge 4. 9. Agreg gruplarının karışım oranları .....	70
Çizelge 4. 10. Modelde kullanılan semboller .....	72
Çizelge 4. 11. Kurallar tablosu .....	73
Çizelge 4. 12. Sonuçların karşılaştırılması .....	75
Çizelge 4. 13. ADB çaprazlamasının sözde kodu .....	82
Çizelge 4. 14. HV metriği kapsamında WR değerinin farklı durumları.....	82
Çizelge 4. 15. HV metriği kapsamında ABC, BLX ve DE çaprazlamalarının karşılaştırılması.....	83
Çizelge 4. 16. D-MOGA ve diğer algoritmaların parametreleri.....	84
Çizelge 4. 17. HV metriğine göre algoritmaların performansı.....	84
Çizelge 4. 18. GD metriğine göre algoritmaların performansı.....	85
Çizelge 4. 19. HV metriğine göre Wilcoxon sıralı toplam testi .....	86
Çizelge 4. 20. GD metriğine göre Wilcoxon sıralı toplam testi .....	87
Çizelge 4. 21. Asfalt problemi için önerilen girdi ve çıktı değerleri .....	98
Çizelge 4. 22. HMA problemi için MOGA-HMA tarafından bulunan çözüm uzayı... 101	
Çizelge 4. 23. HV metriği kapsamında F-MODE ve D-MOGA kıyaslaması .....	103
Çizelge 4. 24. GD metriği kapsamında F-MODE ve D-MOGA kıyaslaması .....	103
Çizelge 4. 25. IGD metriği kapsamında F-MODE ve D-MOGA kıyaslaması.....	104
Çizelge 4. 26. MOHO'nun popülasyon büyüklüğü ve nesil sayısı .....	106
Çizelge 4. 27. IGD metriğinde MOHO, NSGA-III ve MOEA/DD'nin önerilen algoritmalar ile karşılaştırılmaları.....	106
Çizelge 4. 28. HV metriğinde MOSG, MOCeII, IBEA, MOVS algoritmalarının önerilen algoritmalar ile karşılaştırılmaları.....	107

## SİMGELER VE KISALTMALAR

### Simgeler

$\Phi$	: ABC çaprazlama sabiti
$<$	: Baskınlık ( <i>domination</i> )
$\Omega$	: Arama uzayındaki uygun çözümler kümesi
$D$	: Boyut sayısı
$F$	: DE çaprazlama sabiti
$Q$	: Çözümlerin kalitesi
$P$	: Polinomal zaman
$X$	: Karar değişkenleri vektörü

### Kısaltmalar

ABC	: Yapay Arı Kolonisi Algoritması ( <i>Artificial Bee Colony Algorithm</i> )
ADB	: ABC, DE, BLX alfa çaprazlama yöntemlerinin baş harflerinden elde edilen kısaltma (Önerilen Hibrit Çaprazlama Yöntemi)
AVC	: Hava ile Dolu Boşluk ( <i>Air Void Content</i> )
BLX- $\alpha$	: Alfa ile Karıştırılan Çaprazlama ( <i>Blend Crossover with Alpha</i> )
CD	: Kalabalıklık Mesafesi ( <i>Crowding-Distance</i> )
CR	: Çaprazlama Oranı ( <i>Crossover Rate</i> )
CNN	: Konvansiyonel Sinir Ağı ( <i>Convolutional Neural Network</i> )
DE	: Diferansiyel Evrim Algoritması ( <i>Differential evolution Algorithm</i> )
DTLZ	: Deb-Thiele-Laumanns-Zitzler Tarafından Oluşturulan Problem Seti
GA	: Genetik Algoritma ( <i>Genetic Algorithm</i> )
GS	: Genel Adım ( <i>General Stride</i> )
GD	: Nesilsel Mesafe ( <i>Generational Distance</i> )
HV	: Hiperküp ( <i>Hypervolume</i> )
HMA	: Sıcak Asfalt Karışımlar ( <i>Hot Mix Asphalt</i> )
IGD	: Terslenmiş Nesilsel Mesafe ( <i>Inverted Generational Distance</i> )
maxFes	: Maksimum Fonksiyon Değerlendirme Sayısı

		<i>(Maximum Function Evaluation Numbers)</i>
MOAAA	:	Çok Amaçlı Yapay Alg Algoritması <i>(Multi Objective Artificial Algae Algorithm)</i>
F-MODE	:	<i>Fast Multi-Objective Differential Evolution</i>
MODE-CNN	:	CNN İçin Çok Amaçlı Diferansiyel Evrim Algoritması <i>(Multi-Objective Differential Evolution For CNN)</i>
MOEA	:	Çok Amaçlı Evrimsel Algoritma <i>(Multi-Objective Evolutionary Algorithm)</i>
MOEA/D	:	Ayrıştırma Tabanlı Çok Amaçlı Evrimsel Algoritma <i>(Multi-Objective Evolutionary Algorithm Based On Decomposition)</i>
D-MOGA	:	Dinamik Çok Amaçlı Genetik Algoritma <i>(Dynamic Multi-Objective Genetic Algorithm)</i>
MOGA-HMA	:	Sıcak Asfalt Karışımlar için Çok Amaçlı Genetik Algoritma <i>(Multi-Objective Genetic Algorithm For Hot Mix Asphalt)</i>
MOO	:	Çok Amaçlı Optimizasyon <i>(Multi-Objective Optimization)</i>
MOOP	:	Çok Amaçlı Optimizasyon Problemleri <i>(Multi-Objective Optimization Problems)</i>
MC	:	Mermer Kaba Agrega <i>(Marble Coarse Aggregate)</i>
MS	:	Marshal Stabilitesi <i>(Marshall Stability)</i>
NP	:	Deterministik Olmayan Polinomal Zaman <i>(Non-Deterministic-Polynomial Time)</i>
NC	:	Doğal Kaba Agrega <i>(Natural Coarse Aggregate)</i>
ND	:	Komşu Mesafesi <i>(Neighbour Distance)</i>
NF	:	Doğal İnce Agrega <i>(Natural Fine Aggregate)</i>
NSGA-II	:	Bastırılmayan Sıralamalı Genetik Algoritma-II <i>(Nondominated Sorting Genetic Algorithm II)</i>
OMOPSO	:	Geliştirilmiş PSO <i>(Improving PSO-Based Multi-Objective Optimization Using Crowding, Mutation And <math>\epsilon</math>-Dominance)</i>
PA	:	Yama Doğruluğu <i>(Patch Accuracy)</i>
PF	:	Pareto-Optimal Yüzey <i>(Pareto-Optimal Front)</i>
PF <sub>e</sub>	:	Tahmin Edilen Pareto-Optimal Yüzey <i>(Pareto Front Estimated)</i>
PF <sub>t</sub>	:	Gerçek Pareto-Optimal Yüzey <i>(Pareto Front True)</i>
PS	:	Pareto-Optimal Set <i>(Pareto-Optimal Set)</i>
PSO	:	Parçacık Sürü Optimizasyonu <i>(Particle Swarm Optimization)</i>

RF	:	Geri Dönüştürülmüş Beton Agregasıdır ( <i>Recycled Concrete Fine Aggregate</i> )
SOOP	:	Tek Amaçlı Optimizasyon Problemi ( <i>Single-Objective Optimization Problems</i> )
TSP	:	Gezgin Satıcı Problemi ( <i>Travelling Salesman Problem</i> )
VFA	:	Asfalt ile Dolu Boşluk ( <i>Voids of Filled With Asphalt</i> )
WFG	:	Walking Fish Group Problem Seti
ZDT	:	Zitzler-Deb-Thiele Tarafından Oluşturulan Problem Seti





## 1. GİRİŞ

Optimizasyon, bir problemin kısıtlarını dikkate alarak mümkün olan en uygun çözümün ya da çözümlerin bulunma sürecidir. Optimizasyon sürecinde bir problemin çözümü için gerekli ilk adım, problemin matematiksel modelini oluşturma işlemidir. Matematiksel model, problemin amaç fonksiyonu başta olmak üzere problem değişkenlerini ve problem kısıtlarını içermelidir. Amaç fonksiyonunu maksimize veya minimize eden problem değişkenlerini (karar değişkenlerini), önceden tanımlı problem kısıtlarını da hesaba katarak elde etmek, matematiksel modelin hedefi olarak tanımlanabilir. Karar değişken sayısı az, kısıtları sade olan problemlerin matematik modelleri basit ve çözümleri de kolay olmaktadır. Bunun yanında karar değişken sayısı fazla olan karmaşık problemlerin hem matematik modellerini oluşturmak hem de onları çözüme kavuşturmak zordur. Hatta bazı problemler ya hiç çözülememekte ya da çözümün maliyeti çok yüksek olmaktadır.

Problemler karmaşıklık sınıflarına göre dört farklı gruba ayrılmaktadır. İlk sınıf, polinom zamanında çözülebilen problemlerin sınıfı olan P (*Polynomial time*)'dir. Bir problem, eğer bir algoritma kullanılarak verilen girdilerin boyutuna bağlı olarak polinom zamanında çözülebiliyorsa, P sınıfına aittir. Örnek verilecek olursa, bir dizinin elemanlarının ortalaması polinom zamanda bulunabilir. Polinom süre; bir algoritmanın, girdi boyutları ne kadar büyük olursa olsun bir problemi makul bir sürede çözmeye zamanına verilen genel isimdir (Sipser, 1996).

İkinci sınıf ise NP (*Non-deterministic-Polynomial time*) sınıfıdır ve bu sınıftaki problemler, bir çözüm olup olmadığına bakılmaksızın polinom zamanında doğrulanabilen karar problemleridir. Yani; bir problem, eğer verilen bir çözümün doğruluğunu polinom zamanında doğrulayabilen bir algoritma varsa, NP sınıfına aittir. NP sınıfı aslında P ve NP-tam sınıflarını da kapsayan bir sınıftır. Özetle NP sınıfına ait problemler; polinom sürede doğrulanabilen ancak polinom sürede çözülemeyen problemleri içerir. Yani, bir çözüm adayı verildiğinde, çözümün doğruluğu polinom sürede kontrol edilebilir, ancak problemi polinom sürede çözme garantisi yoktur (Sipser, 1996).

Üçüncü sınıf NP-tam (*NP-complete*) problem türüdür ve bu tür problemler, her adımdaki çözümleme zamanı bir önceki adımdaki çözümleme zamanlarından çok daha fazla olduğundan polinom zamanında çözülemezler. Bu problemler hem NP sınıfına ait

hem de NP-hard sınıfına ait problemlerdir. NP-tam sınıfı problemler polinom zamanda doğrulanabilir ancak polinom zamanda çözülebilirliği bilinmemektedir (Sipser, 1996).

Son sınıf ise NP-zor (*NP-hard*) sınıfıdır, NP-hard problemler, en iyi çözümü bulmak için çok fazla zaman ve kaynak gerektiren problemlerdir. NP sınıfı problemler polinom zamanda doğrulanabilirken, NP-hard sınıfı problemler polinom zamanda doğrulanamazlar. NP-hard problem sınıfı içinde NP sınıfına indirgenebilen bir problem varsa bu problem NP-hard bir problemdir. Bir örnek verilecek olursa: A problemi NP-hard bir problemdir çünkü, bu problemin çıktısı ile NP veya NP-hard sınıfına ait B problemi çözülebilir. Mesela Gezgin Satıcı Problemi (*Traveling Salesman Problem, TSP*), Hamilton Yol Problemine indirgenebiliyorsa NP-Hard problemdir. En iyi bilinen NP-hard problemler olarak: TSP, Graf Boyama Problemi (*Graph Coloring Problem*), Çanta Problemi (*Knapsack Problem*), Sıralama Problemi (*Sorting Problem*) gibi literatürde iyi bilinen problemler sayılabilirler (Charbit ve ark., 2007; Fortnow, 2009; Mahajan ve ark., 2012).

Karşılaşılan böyle zor problemler araştırmacıları farklı yollar aramaya sevk etmiştir. Doğa ve canlıların yaşamlarını model alan birçok algoritma, uzun yıllar boyu araştırmacılar tarafından zor problemleri çözmek amacı ile geliştirilmiştir. Öyle ki önerilen bu algoritmalar en zor problemlere bile makul zamanda, en doğru olmasa bile kabul edilebilir en iyiye yakın çözümleri üretmişlerdir. Bu çözüm yaklaşımları sezgisel ve meta-sezgisel algoritmalar olarak isimlendirilmiştir. Sezgisel algoritmalar, problemlerin çözümü için doğal fenomenlerden veya insan zekasından ilham alarak tasarlanan çözüm yaklaşımlarıdır. Bu algoritmalar problem alanına veya yapısına göre özelleştirilmiş çözümler sunabilirler. Meta-sezgisel algoritmalarsa, bir problemi çözmek için farklı sezgisel yöntemleri bir arada kullanabilen, bu yöntemleri koordine edebilen ve yönlendiren daha üst düzey algoritmalarlardır. Bu meta-sezgisel algoritmalar aynı zamanda son derece esnek ve her bir probleme uyarlanabilir şekilde tasarlanmışlardır (Balamurugan ve ark., 2015).

Bu algoritmaların ilk geliştirilme amaçları, tek amaçlı optimizasyon problemlerini (*Single-Objective Optimization Problems, SOOP*) çözmektir. Literatürde en yaygın olarak adı geçen ve üzerinde çalışılan SOOP'ları çözmek için kullanılan bazı algoritmalar ise şunlardır: Genetik Algoritma (*Genetic Algorithm, GA*) (Holland, 1992) Karınca Koloni Optimizasyonu (*Ant Colony Optimization, ACO*) (Dorigo ve ark., 1996), Parçacık Sürü Optimizasyonu (*Particle Swarm Optimization, PSO*) (Kennedy ve Eberhart, 1995), Yapay Arı Kolonisi (*Artificial Bee Colony, ABC*) (Karaboga, 2010), Ağaç Tohum

Algoritması (*Tree-Seed Algorithm, TSA*) (Kiran, 2015) ve Yapay Alg Algoritması (*Artificial Algae Algorithm, AAA*) (Uymaz ve ark., 2015).

Gerçek dünya problemleri genellikle çok amaçlıdır, yani birden çok amaç göz önünde bulundurularak karar değişkenlerinin elde edilmesi gerekir. Bu ihtiyaç düşünülerek son yıllarda çeşitli yollarla çok sayıda, çok amaçlı evrim algoritmaları (*Multi-Objective Evolutionary Algorithms, MOEA*) üretilmiştir. Bu tür algoritmalar ilk başlarda hedefleri çelişmeyen problemler için amaçlar birleştirilerek tek amaçlı hale getirilmiştir ancak çelişen amaçlarda bunu yapmak biraz daha zor olmuştur. Bu tür problemler için daha sonraki dönemlerde çözümler biraz daha karmaşıklaşmış ve daha başarılı algoritmalar önerilmiştir. Geçmişten bugüne sayılamayacak kadar optimizasyon algoritması üretilmiş ve tıpkı doğada olduğu gibi iyi olanlar varlıklarını sürdürmüş, başarısız olanlar ise unutulmuşlardır. Bu tez kapsamında da optimizasyon problemleri üzerinde durulmuş ve optimizasyon algoritmaları geliştirilmiştir. Geliştirilen algoritmalar öncelikle test problemlerinde sınanmıştır. Test problemleri üzerinde başarılı olan algoritmalar iki farklı dünya problemine uygulanmıştır.

Genellikle parametreleri manuel olarak belirlenen CNN modelleri, birçok problemi yüksek başarıyla çözmektedir. Ancak bu modellerin tasarımları manuel olarak yapıldığından dolayı, elde edilen başarılar bile var olan potansiyelin altında kalabilmektedir (Inik ve Ülker, 2022). Buna bağlı olarak bu çalışmanın ana motivasyonu, CNN tabanlı bir sistemin hiper parametrelerinin çok amaçlı ve otomatik olarak optimize edilmesini sağlayacak bir yöntem geliştirmektir. Manuel ayarlamalar ile hem istenilen seviyede başarılı bir model elde edilemeyebilir hem de deneme yanılma yöntemiyle yapılan ayarlamalar zaman maliyetini artırabilir. Literatür incelendiğinde bu problemleri çözmek için optimizasyon algoritmaları kullanıldığı tespit edilmiştir. Bu çalışmalarda nispeten başarılı sonuçlar da alınmıştır ancak CNN mimarileri ile nesne tespitin en büyük problemi olan zaman maliyetinin çok olması, sezgisel veya meta-sezgisel optimizasyon sürecinin en büyük avantajı olan deneme yanılma taktiği ile ters düşmektedir. Bu durumda bütün bu sorunlar hem mimarinin başarısını hem de zaman maliyetini aynı anda iyileştirecek bir yapıya ihtiyaç duyulduğunu göstermektedir. Bunun yanı sıra önerilecek algoritmanın hızlı bir şekilde sonuca yakınsama yapması gerekmektedir. Halbuki, literatürde iyi bilinen ABC ve PSO gibi algoritmaların tanıtıldığı makalelerde maksimum fonksiyon değerlendirme (*Maximum number of Function evaluations, maksFes*) sayısı 240,000-350,000 olarak önerilmiştir (Karaboga ve Basturk, 2007).

MOEA'larda ise en iyi bilinen algoritmalar; Bastırılmayan Sıralı Genetik Algoritma-II (*Non-dominated Sorting Genetic Algorithm-II, NSGA-II*) için 25,000 ve Ayrıştırma Tabanlı Çok Amaçlı Evrimsel Algoritma (*Multi-Objective Evolutionary Algorithm Based on Decomposition, MOEA/D*) için 3 amaçlı problemlerde 75,000 maksFes sayısı önerilmiştir (Deb ve ark., 2002; Zhang ve Li, 2007).

Verilen bu bilgiler ışığında; bir algoritma geliştirilirken maksFes en az 25,000 olarak kullanılabilir olduğu anlaşılmıştır. CNN tabanlı modellerin hiper parametre optimizasyonunda bu sayıya ulaşmak zordur çünkü üst düzey donanıma sahip bilgisayarla bile 1 (bir) fonksiyon değerlendirme süresinin maliyeti çok fazladır (İnik ve ark., 2021).

Bu kıstaslar göz önünde bulundurularak 5,000 maksFes'te iyi, kabul edilebilir sonuçlar üretebilecek bir algoritma tasarlamak hedeflenmiştir. Genellikle MOEA alanında literatürde yaygın olarak kullanılan Pareto tabanlı yaklaşımlardır; bu yüzden bu çalışmalarda Pareto tabanlı bir yaklaşım kullanılmıştır. Hızlı yakınsama içinse Rulet Tekerleri Seçim (*Roulette Wheel Selection, RWL*) mekanizması kullanılarak seçkinci (*elitist*) bir çözüm tekniğinin uygun olabileceğine dair hipotezler kurulmuştur. Başarılı bir optimizasyon algoritması olarak kabul gören Diferansiyel Evrim (*Differential Evolution, DE*) algoritmasının çaprazlama yaklaşımı, ilk çalışmada kullanılmıştır, böylece önerilen ilk algoritmanın ismi: Hızlı Yakınsamalı Çok Amaçlı Diferansiyel Evrim Algoritması yani F-MODE olarak belirlenmiştir. F-MODE literatürde iyi bilinen algoritmalar ile 5,000 maksFes'te test fonksiyonları (*benchmark functions*) üzerinde kıyaslanmıştır. Sonrasında geliştirilen F-MODE hızlı yakınsamaya sahip olduğundan dolayı, zamanın önemli olduğu CNN tabanlı modelin hiper parametre optimizasyonunda kullanılmıştır. Bu algoritma, CNN tabanlı bir mimariye uyarlandığından MODE-CNN (*Multi-Objective Differential Evolution for CNN*) olarak isimlendirilmiştir (İnik ve ark., 2021).

Tezde ele alınan ikinci gerçek dünya problemi ise sıcak asfalt karışım (*Hot Mix Asphalt, HMA*) konusudur. Sıcak asfalt içerisine hangi oranlarda karışımlar yapılmalı sorusunun cevabını bulmak için literatürde birçok çalışma mevcuttur. Bunun için çok sayıda laboratuvar deneyleri yapılmış ve çeşitli yöntemler denenmiştir. Karar değişkeni sayısına ve alt üst limitlerine bağlı olarak bazen milyonlarca ihtimal arasından problem girdileri manuel olarak yapılmaktadır ancak doğru girdi oranlarını el ile deneyerek bulmak neredeyse imkânsızdır. Bu sebeple literatürde var olan bu problemi çok amaçlı optimizasyon problemlerine (*Multi-Objective Optimization Problems, MOOP*) uygun olarak modelleme ihtiyacı görülmüştür. Bu tarz problemler ele alınırken literatürde FL tabanlı algoritmalar kullanıldığı tespit edilmiştir; zira gerçek deney sayıları genelde 40-

50 adet olmakta ve bu sayı etkin matematiksel bir amaç fonksiyonu formülünü üretmek için yetersiz kalmaktadır. Bu deney verileri kullanılarak bir FL uzman sistem geliştirilmiş ve ilk aşamada hangi girdilerin hangi çıktıyı ürettiği görülmüştür. İşin bu kısmında konunun uzmanı inşaat mühendisliği akademisyenlerinden yardım alınmıştır. Sonuç olarak, girdi oranları verildiğinde uygun çıktıları üreterek bir amaç fonksiyonu işlevi üstlenecek FL uzman sistemi tasarlanmıştır. Bu sistem tasarımında çeşitli problemleri çözmek gerekmiştir. Bu problemlerden birincisi, yüz binlerce girdi arasından en uygun çıktıyı sağlayacak girdi oranlarının nasıl belirleneceğidir İkinci problemse, aynı girdi oranlarına bağlı olarak problemin üç adet çıktı üretmesinin gerekliliğidir. Dolayısıyla bu problemi bir MOOP olarak ele almak kaçınılmazdır. Bütün bu problemleri çözmek için çeşitlilik yeteneği fazla olan bir MOEA geliştirmek hedeflenmiştir. Başarılı ve yaygın yöntemlerden biri olan Pareto taban yaklaşımı, geliştirilecek algoritmanın ana hatları olarak tercih edilmiştir. Yeterli çeşitliliği sağlamak amacıyla hibrit bir çaprazlama modeli kullanılmasının uygun olacağına dair hipotez geliştirilmiştir (Altiok ve ark., 2023).

D-MOGA'nın test fonksiyonları üzerinde performansının yeterli olduğu görülmüş ve sonrasında HMA problemine uygulanmıştır. Bu uygulamada D-MOGA, HMA problemine uyarlandığı için MOGA-HMA (*Multi-Objective Genetic Algorithm for the Hot Mix Asphalt Problem*) olarak isimlendirilmiştir.

Sonuç olarak önerilen yöntemle otomatik olarak üretilen girdilerle alınan sonuçlar, laboratuvar ortamında deneme yanılma ile elde edilebilecek sonuçlara göre daha başarılı olduğu anlaşılmıştır.

### **1.1. Tezin Amacı ve Literatüre Katkıları**

Bu tezin amacı, çok amaçlı problem çözme yaklaşımlarının kullanılabileceği iki farklı problemi incelemek ve bu problemleri çözmek için hipotezler oluşturarak çok amaçlı çözüm yaklaşımları geliştirmektir. Pareto tabanlı algoritmalar kullanılarak her iki problem için hibrit yaklaşımlar geliştirilmiştir. İlk çalışmada, derin öğrenme alanında yaygın olarak kullanılan CNN tabanlı bir modelin hiper parametre optimizasyonu ele alınmış ve hızlı yakınsamaya sahip F-MODE isimli bir algoritma önerilmiştir. İkinci çalışmada ise asfalt karışımlarının oranlarını optimize etmek için D-MOGA adında farklı bir algoritma geliştirilmiştir. Çalışmanın sonunda F-MODE ve D-MOGA algoritmaları arasında karşılaştırmalı bir analiz yapılmış ve elde edilen sonuçlar kapsamlı bir şekilde analiz edilerek değerlendirilmiştir. Bu çalışma, çok amaçlı problem çözme

yaklaşımlarının performansını ve etkinliğini değerlendirerek, gelecekteki araştırmalara yönelik güçlü yönlere, sınırlamalara ve potansiyel iyileştirme alanlarına ışık tutacağı düşünülmektedir (İnik ve ark., 2021; Altıok ve ark., 2023).

Bu tez çalışması kapsamında yapılan ilk çalışmanın katkıları şöyle sıralanabilir:

- F-MODE, CNN tabanlı sistemlerin hiper parametre optimizasyonunu veya benzer başka problemleri çözmek amacıyla geliştirilmiştir.
- MOEA kullanıldığında, çözümlerin kalitelerini belirlemek zor olabilir, çünkü birinci amaç için iyi kabul edilen bir sonuç, ikinci amaç için kötü olabilir. Bu nedenle, bu çalışmada kullanılan bir kalite belirleme yöntemi önerilmiştir.
- Kaliteye göre belirlenen çözümler, ebeveyn seçiminde rulet tekeri yöntemiyle kullanılmış ve çaprazlama işlemleri bu seçimlere göre gerçekleştirilmiştir. Bu şekilde, kaliteli çözümler çaprazlamada daha fazla yer almış ve nispeten hızlı bir yakınsama sağlanmıştır.
- Bu çalışmada yapılan literatür taramasına göre ilk kez çok amaçlı olarak ele alınan *GS*, *ND* ve *PA* parametreleri, test süresi ve segmentasyon hatası temel alınarak ve MODE-CNN algoritması kullanılarak optimize edilmiştir. Bu çalışmada CNN mimarisi MODE-CNN için amaç fonksiyonu görevini üstlenmiştir.

Tez çalışması kapsamında yapılan ikinci çalışmanın katkıları şöyle sıralanabilir:

- D-MOGA, literatürde yaygın olarak başarılı kabul edilen çaprazlama modellerinden ilham alarak, artırılmış rastgelelik özelliklerine sahip hibrit bir çaprazlama operatörüne sahiptir. Bu şekilde, geniş arama alanına sahip gerçek dünya problemleri için güçlü bir yaklaşım literatüre katılmıştır.
- Bu çalışmada, literatürde genellikle sabit değerlerle temsil edilen çaprazlama oranı (*Crossover Rate*, *CR*), değerlendirme sayısına bağlı olarak dinamik şekilde değişen yeni bir yaklaşım önerilmektedir. Başlangıçta yüksek bir *CR* kullanılarak global aramaya ağırlık verilirken, sürecin ilerleyen aşamalarında *CR* oranı düşürülmekte ve yerel arama faaliyetleri daha etkin hale getirilmektedir.
- Son olarak, HMA için gerçek deney verileri kullanılarak kural tabanlı bir FL uzman sistemi tasarlanmış ve bu uzman sistem, D-MOGA algoritması için amaç fonksiyonu işlevi üstlenmiştir. Bu sayede, matematiksel amaç

fonksiyonu formülünü çıkarmanın zor olduğu problemlerde uzman sistemi entegre etme yaklaşımı literatüre kazandırılmıştır.

Bu iki çalışmanın ortak özellikleri şöyle sıralanmıştır:

- Birinci olarak, iki çalışmada da çok amaçlı optimizasyon (*Multi-Objective Optimization, MOO*) yöntemlerinden yaygın olarak kullanılan Pareto taban yaklaşımının kullanılmasıdır.
- İkinci ortak nokta ise yapılan çalışmaların test fonksiyonlarında sınılandıktan sonra bir gerçek dünya problemi üzerinde uygulanmasıdır.
- Son olarak, gerçekleştirilen HMA ve hiper parametre optimizasyonu problemlerinde amaç fonksiyonu yerine başka bir sistemin (yaklaşımın) kullanılması yani hibrit bir problem çözme modelinin tasarlanmasıdır.

## 1.2. Tezin Organizasyonu

Bu tez çalışması beş bölümden oluşmaktadır. Giriş bölümünde tezin çıkış noktası, önemi ve literatüre katkılarına değinilmiştir. İkinci bölümde tez konuları ile alakalı kaynak araştırması yapılmıştır. Üçüncü bölümde tez konusu hakkında temel bilgilere yer verilmiştir. Dördüncü bölümde, problemler ayrıntılı şekilde açıklanmış ve bu problemler için geliştirilen yöntemler tanıtılmıştır. Aynı bölümde geliştirilen algoritmalar literatürde bilinen algoritmalar ile kıyaslanmış daha sonra algoritmalar ana problemlere uyarlanmıştır. Dördüncü bölümün son kısmında ise geliştirilen iki algoritma test fonksiyonları üzerinde kıyaslanmıştır. Son bölümde ise elde edilen sonuçların değerlendirilmesi ve gelecek çalışmalar için önerilere yer verilmiştir.

## 2. KAYNAK ARAŞTIRMASI

Bu bölümde kaynak araştırması yapılmış ve araştırma üç bölüme ayrılmıştır çünkü bu çalışma; çok amaçlı optimizasyon algoritmaları ve iki ayrı gerçek dünya problemi olarak üç ana kısımdan oluşmaktadır. Bunun yanında ilgili problemler bu çalışmada yapılan literatür araştırmasına göre, ilk defa bir MOOP olarak ele alınmıştır. Kaynak araştırması bölümünde kısaca MOEA'nın yıllara göre gelişimi, en önemli MOEA algoritmalarının neler olduğu, CNN tabanlı modellerin hiper parametre optimizasyonu problemi, önceki çalışmalarda bu tür problemlerin nasıl ele alındığı açıklanmıştır. Son olarak HMA probleminin ne olduğu ve daha önce bu ve benzeri problemler hakkında ne tür çalışmalar yapıldığından bahsedilmiştir.

Bu çalışma kapsamında yapılmış olan literatür taramasında görülen eksiklikler ise; hiper parametre optimizasyonu yapılan modelin, daha önce çok amaçlı optimizasyon problemi olarak ele alınmadığı anlaşılmıştır. Aynı şekilde HMA probleminin daha önce literatürde farklı yönleri ile ele alındığı ancak çok amaçlı olarak düşünülmediği görülmüştür. Bunun yanında HMA türü problemlerde yapılan deneylerin FL ile modellendiği tespit edilmiştir.

### 2.1. Çok Amaçlı Optimizasyon Algoritmaları ve Kaynak Araştırması

MOEA'larda Pareto kavramı çok önemli bir yer tuttuğundan ilk bu kavramın tarihçesi açıklanmıştır. Pareto kavramı, Pareto tabanlı MOEA'ların temel felsefesini oluşturmaktadır.

Bir iktisat kavramı olan Pareto verimliliği teorisi, bazı kişilerin durumunu iyileştirirken diğerlerinin durumunu kötüleştirmeden refahı artırmayı amaçlar. Bu teori ilk olarak Edgeworth (1881) tarafından önerilmiştir. Uygulamada ise ilk kullanan, İtalyan bir iktisatçı olan Pareto'dur (1896). Pareto'nun amacı toplumdaki üretici ve tüketici arasındaki kaynak tahsisini optimize etmektir. Pareto verimliliği teorisinin ilk defa MOEA'larında kullanılabileceği fikri ise Goldberg tarafından 1989 yılında öngörülmüştür (Coello, 2006).

Araştırmacılar çoğu gerçek dünya problemlerinin yapısal olarak MOOP sınıfına girdiğini keşfettikten sonra bu problemleri çeşitli yaklaşımlarla çözmeye çalışmışlardır. 19. yüzyılın ortalarından itibaren hedef programlama, doğrusal programlama, ağırlıklı toplam yöntemleri MOOP'ları çözmek için denenmiştir. Yapılan çalışmalardan sonra MOOP'lar için kesin matematiksel yöntemlerin hesaplama maliyeti, zaman maliyeti,



probleme özgü sorunlar gibi olumsuzluklar yüzünden uygun olmadığı anlaşılmıştır. Bu düşünce araştırmacıları sezgisel ve meta-sezgisel optimizasyon algoritmalarını araştırmaya sevk etmiştir. 1967 yılında ise Rosenberg evrimsel algoritmaların MOOP'ları çözümede kullanılabileceğini iddia etmiştir. Uygulamada ise çok amaçlı problemleri bir şekilde tek amaçlı problemlere dönüştürerek gerçekleştirmiştir (Özkış, 2017; Özkış ve Babalık, 2017; Karakoyun ve ark., 2020)

Vektörel Değerlendirmeli Genetik Algoritma (*Vector Evaluated Genetic Algorithm, VEGA*) ile problemler ilk defa çok amaçlı olarak ele alınmasına rağmen başarılı sonuçlar elde edilememiştir (*Schaffer, 1985*). VEGA sonuçları itibari ile başarısız bir algoritma olmasına rağmen, bu alanda öncü olduğu için önemli bir algoritmadır. Goldberg, VEGA'nın eksiklerini tespit etmiş ve bu eksiklerin giderilmesi için günümüzde hala kullanılan bastırılmayan sıralama (*non-dominated sorting*), seçim (*selection*) gibi çözüm önerileri sunmuştur (Goldberg ve ark., 1989). Bu öneriler ışığında Srinivas ve Deb (1994) Bastırılmayan Sıralı Genetik Algoritmayı (*Non-dominated Sorting Genetic Algorithm, NSGA*); Fonseca ve Fleming (1993), Çok Amaçlı Genetik Algoritmayı (*Multi-Objective Genetic Algorithm, MOGA*); Horn ve arkadaşları (1994) Hücrelendirilmiş Pareto Genetik Algoritmayı (*Niching Pareto Genetic Algorithm, NPGA*) önermişlerdir. Sonraki çalışmalarda hem çeşitliliği hem de bastırılmayan bireyleri dengeli şekilde korumanın ve geliştirmenin önemi keşfedilmiştir. Bundan dolayı seçkinci yaklaşım ön plana çıkmıştır. Seçkincilik mekanizması iki şekilde işler: birincisi, ebeveynlerin içinden en iyisinin seçilmesidir; diğeri ise harici bir popülasyonda iyi bireylerin tutulmasıdır ki bu yöntem "arşiv yöntemi" denir. Bu stratejiden sonra ikinci nesil denilebilecek: NSGA-II; Kuvvet Pareto Evrimsel Algoritması 1 ve 2 (*Strength Pareto Evolutionary Algorithms, SPEA 1- SPEA II*) (Zitzler ve Thiele, 1999); Pareto Arşivlenmiş Evrimsel Stratejisi (*Pareto Archived Evolution Strategy, PAES*) (Knowles ve Corne, 1999) gibi algoritmalar önerilmiştir. Bunların yanı sıra farklı bir, çok amaçlı problem çözme yaklaşımı olarak ayrıştırma tabanlı problem yaklaşımı olan MOEA/D algoritması, bu alanda ilgi duyulan bir çalışma olmuştur. MOEA/D üzerine birçok çalışmalar yapılmıştır, bu çalışmalardan MOEA/D Tabanlı İki Adımlı Evrimsel Algoritma (*Two-stage Evolutionary Strategy Based MOEA/D to Multi-Objective Problems, MOEA/D-TS*) bu tez kapsamında karşılaştırma algoritmalarından biri olarak kullanılmış yeni ve rekabetçi bir algoritmadır. Bunun gibi MOEA/D üzerine yapılmış birçok çalışma mevcuttur (Zhang ve Li, 2007; Qi ve ark., 2014; Wang ve ark., 2015; Li ve ark., 2019; Zhang ve ark., 2020).

Evrimsel tabanlı meta-sezgisel algoritmalar MOOP'larda çok başarılı olunca birçok araştırmacının ilgisi bu konu üzerine odaklanmıştır. Bu süreçten sonra ise tek amaçlı evrimsel algoritmalar yani SOOP'lar çok amaçlı algoritmalara dönüştürülmüş ve tez kapsamında sayılamayacak kadar çok çalışma yapılmıştır. Bunlardan bazıları şöyledir: *PSO* algoritmasından uyarlanan MOPSO ve türevleri (Coello ve ark., 2004; Sierra ve Coello, 2005; Peng ve Zhang, 2008; Aboud ve ark., 2022). ABC'den esinlenilerek geliştiren bazı algoritmalar (Akbari ve ark., 2012; Xiang ve Zhou, 2015; Xiang ve ark., 2015; Kishor ve ark., 2016). Bunların yanında Özkış ve ark. tarafından AAA' dan esinlenerek dizayn edilmiş Çok Amaçlı AAA (*Multi-Objective AAA, MOAAA*) (2018); Tek amaçlı TSA'nın çok amaçlı versiyonu olan (*Multi-Objective TSA, MOTSA*) (Ozcan ve ark.) yapılan birçok çalışmadan bazı önemli olanlarıdır.

Bu çalışmalar gibi daha birçok araştırmacı tarafından, SOOP'lardan esinlenerek bu algoritmaların çok sayıda, çok amaçlı türevleri geliştirilmiştir (Corne ve ark., 2000; Abbass ve ark., 2001; Corne ve ark., 2001; Zitzler ve ark., 2001; Nebro ve ark., 2009; Akay, 2013; Kishor ve ark., 2016; Mirjalili ve ark., 2016; Özkış ve Babalık, 2017; Savsani ve Tawhid, 2017; Babalik ve ark., 2018; Zapotecas-Martínez ve ark., 2019; Karakoyun ve ark., 2020).

## **2.2. CNN Tabanlı Modellerin Hiper Parametre Optimizasyonu ve Kaynak Araştırması**

Derin öğrenmenin temel mimarisini oluşturan konvansiyonel (*convolution*) sinir ağı, CNN son yıllarda sınıflandırma (*classification*), tespit (*detection*) ve bölütleme (*segmentation*) gibi problemlerde başarıyla kullanılmaktadır (Krizhevsky ve ark., 2012). CNN'de temel sorun, birçok problem türü için geliştirilmiş çok sayıda problem mimarisi olmasıdır. Bu durum, her problem ve dolayısıyla her mimari için farklı hiper parametre ayarlarının yapılmasını gerektirir ve bu da büyük bir emek ve özen gerektirir.

Bu modellerin hiper parametreleri genellikle bir uzmanın tecrübelerine dayanarak ayarlanır. Fakat son yıllarda CNN tabanlı mimarilerinin optimizasyon yoluyla tasarımı hakkında çalışmalara ilgi artmıştır. İlk parametre optimizasyonu çalışmaları Izgara Arama (*Grid Search*), Rastgele Arama (*Random Search*) (Bergstra ve Bengio, 2012) Bayes Optimizasyon (*Bayesian Optimization*) (Pelikan ve ark., 1999) algoritmaları ile yapılmıştır. Bunların yanında çok sayıda tek amaçlı meta-sezgisel optimizasyon algoritmaları da daha yüksek sınıflandırma oranlarını hedefleyerek, parametre

optimizasyonu için kullanılmışlardır (Liu ve ark., 2017; Lorenzo ve ark., 2017; Real ve ark., 2017; Suganuma ve ark., 2017; Xie ve Yuille, 2017; Liu ve ark., 2018; Miikkulainen ve ark., 2019; Sun ve ark., 2019).

Başka bir çalışmada, Dufourq ve Bassett 2017 yılında yayınladıkları bilgilere göre optimum katman sayısı ve filtre boyutu, filtre sayısı ve her katmanın aktivasyon fonksiyonunu GA ile bulmaya çalışmışlardır (Dufourq ve Bassett, 2017).

Yamasaki ve arkadaşları *Alexnet* (Krizhevsky ve ark., 2012) mimarisinin filtre sayısı, filtre boyutu ve havuz tipini optimize etmişlerdir (Yamasaki ve ark., 2017). Bu çalışmaya göre, optimize edilmiş *AlexNet*'in *CIFAR10*, *CIFAR100*, *Subset10*, *Subset30*, *Subset50* veri setlerinde orijinal *AlexNet* mimarisine göre daha iyi doğruluk değerleri elde ettiği belirtilmiştir.

Sun ve arkadaşları (2018) tarafından önerilen çalışmada, konvansiyonel ve havuzlama katmanları üzerindeki kısıtlamaları ortadan kaldırarak esnek bir konvansiyonel otomatik kodlayıcı önerilmiştir. Bu çalışma, derin öğrenme alanında önemli bir adımdır ve gelişmiş bir konvansiyonel otomatik kodlayıcı tasarımını sunmayı hedeflemektedir. Bu makaledeki yenilikçi yaklaşım, konvansiyon ve havuzlama katmanlarındaki kısıtlamaları kaldırmak suretiyle daha esnek bir kodlama süreci sunmasıdır. Bu da modelin veriye daha iyi uyum sağlamasına ve daha iyi özelliklerin öğrenilmesine olanak tanımaktadır. Önerilen yöntem, daha etkili bir özellik çıkarımı sağlayarak, örnekleme problemlerini azaltır ve genel olarak daha iyi performans sunmuştur. Bu çalışma, derin öğrenme araştırmalarında önemli bir gelişme olarak kabul edilirken, önerilen esnek konvansiyonel otomatik kodlayıcı, çeşitli uygulama alanlarında potansiyel olarak daha iyi sonuçlar elde etme imkânı sunmaktadır.

Da Silva ve arkadaşları (2017) tarafından yapılan bir çalışmada, akciğer nodüllerinin sınıflandırılması için sabit bir mimariye sahip bir CNN kullanılmıştır. Bu çalışmada, filtre sayısı ve her katmandaki nöron sayısı, GA kullanılarak optimize edilmiştir. Bu çalışmanın temel amacı, akciğer nodüllerinin sınıflandırılmasında kullanılan CNN'nin performansını artırmaktır. GA'nın kullanımı, bu çalışma ile hiper parametre optimizasyonunda etkili bir araç olarak ortaya çıkmaktadır.

Ma ve arkadaşları (2020) çalışmasında, GA kullanılarak farklı veri kümeleri üzerinde farklı CNN modellerinin tasarımı elde edilmeye çalışılmıştır. Bu çalışma, CNN modellerinin optimize edilmesi ve veri setlerine özgü ihtiyaçlara uygun hale getirilmesi üzerinde odaklanmaktadır.

Lee ve arkadaşları (2018) çalışmasında, veri kümelerine göre farklı boyutlarda CNN'ler seçilmiştir. Bu çalışmada, konvansiyon katmanındaki filtre boyutu, filtre sayısı ve adım değeri ile havuzlama katmanındaki filtre boyutu ve adım değeri, meta-sezgisel bir optimizasyon yöntemi olan Harmoni Algoritması ile optimize edilmiştir. Bu çalışmanın amacı, veri kümelerine özgü ihtiyaçları karşılamak üzere CNN mimarilerinin optimize edilmesidir. Harmoni Algoritması, müzikteki harmoni prensiplerinden esinlenerek, farklı parametre değerlerini bir arada uyumlu bir şekilde optimize etmek için kullanılan bir optimizasyon tekniğidir.

Bir CNN mimarisi için hiper parametre optimizasyonu; doğası gereği çok amaçlı bir problemdir, yani en ufak bir parametre değişimi birbirinden bağımsız en az iki sonucu eş zamanlı olarak etkilemektedir. Bu yüzden parametre optimizasyonunu ele alırken tek amaç yerine birden fazla amacın aynı anda optimize edilmesi daha doğru olur çünkü bu probleme uygun algoritmaları kullanmak daha iyi sonuçlar verecektir. Mesela en az parametre sayısı ile en yüksek doğruluk veya en kısa zamanda en yüksek doğruluk için giriş değerleri eş zamanlı olarak optimize edilebilir. Bu gibi avantajlarından dolayı MOEA, CNN mimarisinin parametre optimizasyonunda kullanılmaya başlanmıştır (Hsu ve ark., 2018; Junior ve Yen, 2019; Loni ve ark., 2020).

### **2.3. Sıcak Asfalt Karışım Problemi ve Kaynak Araştırması**

Otoyol yapımında yaygın olarak kullanılan malzeme; sıcak asfalt, agrega ve bitümden oluşmaktadır; aynı zamanda bu karışımın yaklaşık %90'ı andezit kalker ve bazalt gibi doğal taşlardan elde edilen agregalardan oluşmaktadır (Ahmedzade ve Sengoz, 2009; Sargın ve ark., 2013). Ancak doğal agrega kaynaklarının çok fazla kullanımı kaliteli agrega kaynaklarını azaltmakla kalmamış bunun yanında bazı çevre sorunlarını da beraberinde getirmiştir. Bu durum araştırmacıları farklı kaynakların kullanımını incelemeye yöneltmiştir. Bu arayışa çevre sorunlarını da dâhil edince agrega yerine geçebilecek çeşitli inşaat ve mermer atıklarını değerlendirmek üzerine birçok çalışma yapılmıştır (Akbulut ve Gürer, 2007; Mills-Beale ve You, 2010; Lee ve ark., 2012; Pourtahmasb ve ark., 2015; Jamshidi ve ark., 2019).

Akbulut ve Gürer, 2007'de yaptığı çalışmada, atık mermer parçalarının (*Waste Marble Aggregates*, WMA) asfalt kaplamada kullanılabilirliğini incelemiştir. Bu çalışma kapsamında atık mermer parçaları kaba agrega, ince agrega veya dolgu agregası olarak, yani agreganın her türü için olarak denenmiştir. Yapılan deneylere göre, sağlamlık

(*stability*) deęerleri karřılařtırıldıęında mermer agregası ile hazırlanan karıřımların doęal agrega karıřımlarına benzer sonular verdięi grlmřtr. Sonuta, mermer atıęı agregaların hafif ve orta hacimli asfalt kaplamaların baęlayıcı tabakasında kullanılmasında bir problem olmayacaęı belirlenmiřtir (Akbulut ve Grer, 2007).

Khodabakhshian ve arkadařları sıradan Portland imentosu yerine kısmen atık mermer tozu kullanılmasının sonucu olarak beton karıřımı zerindeki dayanım zelliklerini incelemiřlerdir. Sonuta %10'a kadar atık mermer tozu kullanımının beton dayanıklılıęını dřrmedięi ancak bu orandan fazla kullanımlarda beton dayanıklılıęının azaldıęı grlmřtr (Khodabakhshian ve ark., 2018).

Beale ve You bir alıřmalarında, agrega karıřımına doęal agregalar yerine %25, %35, %50 ve %75 oranlarında geri dnřtrlmř beton agregası (*Recycled Concrete Aggregate, RCA*) karıřtırmıřlardır. Bu alıřmadan edinilen tecrbeye gre, *RCA* ieren HMA numunelerinin sıkıřtırma iin klasik HMA numunelerine gre daha az enerjiye ihtiya duyduęu belirlenmiřtir. Bu ıkarıma ek olarak, kk hacimli trafięe sahip yollarda %75'e kadar *RCA* kullanılabilereęi anlařılmıřtır (Mills-Beale ve You, 2010).

Hebhoub ve arkadařları, beton retiminde doęal agrega yerine %25, %50, %75 ve %100 *WMA*'yı kullanmıřlardır. Yapılan deneylere gre *WMA* kullanılarak oluřturulan rneklerin mekanik zellikler bakımından standartlara uygunluęu anlařılmıřtır. Ayrıca *WMA* doęal agrega karıřımında %75'e kadar kullanıldıęında basıncı dayanımında artıř saęladıęı gzlemlenmiřtir (Hebhoub ve ark., 2011).

lkemiz dnya mermer rezervinin %40'ına sahiptir ve yıllık mermer retimi milyonlarca tonlara ulařmaktadır. Mermer bloklarının ıkarılması ve iřlenmesi sırasında milyonlarca ton mermer atıęının oluřtuęu tahmin edilmektedir (Gencel ve ark., 2012; Alakara ve Agaoglu, 2020). Bu mermer atıklar yerleřim yerlerinin yakınlıklarına atıldıkları ve vreysi nemli lde kirlettikleri tespit edilmiřtir. Btn bunlar gz nne alındıęında mermer atıklarını belli oranlarda inřaat alanında kullanmanın ekonomik ve vresel faktrlere gre faydalı olacaęı bunun yanı sıra doęal agregalara da alternatif olacaęı ngrlmektedir.

Bu alanda nemli olan bazı alıřmalar incelendięinde; FL yaklařımı kullanılarak beton ve bitml sıcak karıřımların bazı performans parametrelerinin yksek doęrulukla tahmin edilebildięi gzlemlenmiřtir (Siddique ve ark., 2011; Shafabakhsh ve Tanakizadeh, 2015; Zehtabchi ve ark., 2018).

Kara ve Karacasu geliřtirdikleri FL ile kiremit atıklarının Marshall parametreleri zerindeki etkisini analiz etmiřlerdir (Kara ve Karacasu, 2017). Bu model ile hangi

girdilerin hangi çıktılarını yani: Marshal Stabilitesi (*Marshall Stability, MS*), hava boşluk içeriği (*Air Void Content, AVC*) ve asfalt ile dolu boşluk (*Voids of Filled with Asphalt, VFA*) verdikleri yüksek bir başarı ile tahmin edilmiştir.

Başka bir çalışmada Topçu ve Sarıdemir, silis dumanı içeren *RCA*'lı betonların basınç ve bölünme çekme dayanımını tahmin etmek için bulanık mantık yöntemi geliştirmiştir. Bu çalışmanın sonucu olarak geliştirilen *FL*'nin basınç ve bölme çekme dayanıklılığı değerlerini tahmin etmede oldukça güçlü bir potansiyele sahip olduğu gözlemlenmiştir (Topçu ve Sarıdemir, 2008).

Özgan, asfalt örneklerinin *MS*'sini farklı sıcaklıklarda ve farklı maruz kalma sürelerinde *FL*'yi kullanarak modellemiştir (Ozgan, 2009). Oluşturulan model sonucunda asfalt numunelerinin *MS*'sinin farklı sıcaklık ve farklı maruz kalma sürelerine bağlı olarak değiştiği ve *FL* ile gerçek deneylerin sonuçlarının %99 olarak uyumlu oldukları tespit edilmiştir.

### 3. MATERYALLER VE YÖNTEMLER

Bu bölümde optimizasyon, çok amaçlı optimizasyon, Pareto tabanlı çok amaçlı optimizasyon algoritmaları, kullanılan problem setleri, kullanılan metrikler ve performans göstergeleri hakkında bilgiler verilmiştir.

#### 3.1. Optimizasyon Nedir?

Optimizasyon, Latince bir kelime olan optimum kelimesinden türetilmiştir (TDK, 2023). Optimum kelimesi ise, en uygun, en elverişli, en iyi anlamlarına gelmektedir. Literatürde ise optimizasyon, mevcut kaynakları en iyi şekilde kullanarak, olabilecek en iyi çözümü elde etme sürecidir. Örneğin bir fabrikanın maliyeti en aza indirerek maksimum seviyede kar etme hedefi bir optimizasyon süreci olarak tanımlanabilir. Buna benzer olarak o fabrika işçilerinin işe gelirken en kısa yolu tercih etmeleri mesafenin optimize edilmesi, en az maliyetle çalıştıkları yere ulaşmaları yol masrafının optimize edilmesi, en güvenli yolu tercih etmeleri ise güvenliğin optimize edilmesidir. Çok amaçlı optimizasyon ise bu sayılan amaçlardan en az iki tanesini eşzamanlı olarak optimize etme sürecidir.

#### 3.2. Çok Amaçlı Optimizasyon Temel Bilgiler

Bu bölümde çok amaçlı optimizasyonun ne olduğu ve matematik modeli açıklanmıştır. Bunun yanında çok amaçlı optimizasyon konusu için gerekli temel ifadeler; amaç fonksiyonu, karar değişkenleri, problem kısıtları hakkında bilgiler verilmiştir. Bunlara ek olarak; çok amaçlı ve tek amaçlı problemlerin farkları, Pareto tabanlı optimizasyon algoritmaları ve bu algoritmaların temel operatörleri hakkında gerekli açıklamalar yapılmıştır.

##### 3.2.1. Çok amaçlı optimizasyon problemi

Belirli kısıtlamalar dahilinde aynı anda birden fazla amacın eş zamanlı olarak optimizasyonunu gerektiren problemler, çok amaçlı optimizasyon problemi olarak bilinir. Günümüzde çoğu mühendislik probleminde, birden fazla (genellikle birbiriyle çelişen) amacı aynı anda, eş zamanlı optimize etmek gerekebilir. Örneğin, aynı koşullar altında maliyeti en aza indirmek ve kaliteyi en üst düzeye çıkarmak çok amaçlı bir problemdir. Bu problemleri optimal bir şekilde çözmek için, problemin matematiksel modeli (amaç fonksiyonları) oluşturulur ve bir çözüm üretmek için kullanılır. Amaç fonksiyonunun

hesaplanmasında kullanılan her girdiye karar değişkeni adı verilir ve karar değişkeni sayısı arttıkça problemin boyutu da artar. MOOP'lar için genel matematiksel model Denklem 3.1'de verilmiştir (Ergül, 2010; Özkış, 2017).

$$\begin{aligned}
 (Min/Max)F(X) &= [f_1(X), f_2(X), f_3(X) \dots f_m(X)]^T, \\
 X &= [x_1, x_2, x_3, \dots x_D]^T, \\
 L_d &\leq x_d \leq U_d, d = 1, 2, 3 \dots D, \\
 g_i(X) &\leq 0, i = 1, 2, 3 \dots I, \\
 h_j(X) &= 0, j = 1, 2, 3 \dots J
 \end{aligned} \tag{3.1}$$

Bu matematik modele göre en baştaki (*Min/Max*) problemin minimizasyon ya da maksimizasyon problemi olması gerektiğini ifade eder.  $F(X)$  ise  $m$  boyutlu amaç fonksiyonları vektörünü gösterir.  $f_1(X), f_2(X), f_1(X), \dots, f_m(X)$  ifadesi ise maksimize yada minimize edilecek 1. amaçtan,  $m$ . amaca kadar olan amaç fonksiyonlarının optimizasyonunu temsil eder. Matematiksel eşitliğin  $X = [x_1, x_2, x_3, \dots x_D]^T$  kısmında  $X$  1'den  $D$ . boyuta kadar olan karar değişkenleri vektörüdür.  $L_d$  karar değişkenlerinin alt limitini ifade ederken,  $U_d$  ise karar değişkenlerinin üst limitini ifade eder.  $g_i(X)$  ve  $h_j(X)$  ise sırasıyla sağlanması gereken  $i$ . eşitsizlik ve  $j$ . eşitlik kısıtlarını ifade eder (Ergül, 2010; Özkış, 2017).

### 3.2.2. Amaç fonksiyonları

Literatürde maliyet fonksiyonu olarak da geçen amaç fonksiyonu; kısaca bir optimizasyon probleminin matematiksel modeline denir. Örneğin her karar değişkeninin karelerinin toplamı şeklinde ifade edilen *Sphere* fonksiyonun, matematik modeli  $N$  doğal sayı olmak üzere, şöyledir (Coello, 2007):  $F(X) = \sum_{i=1}^N x_i^2$ .

### 3.2.3. Karar değişkenleri

$X = [x_1, x_2, x_3, \dots x_D]^T$ ,  $X$  bir karar değişkenleri vektörü olmak üzere, arama uzayındaki alt ve üst limitlere uygun değerler  $\Omega$  simgesiyle gösterilirse  $X \in \Omega$  problemin çözümlerinden birini ifade eder.  $D$  problem boyutu olmak üzere ve  $d = 1, 2, 3 \dots D$  ise, her  $x_d$  alt limit  $L_d$  ve üst limit  $U_d$  aralığında bir değer almak zorundadır. Karar değişkenleri



tam sayı, ikili, sürekli ya da permütasyon olarak problemin yapısına göre değişirler (Coello, 2007).

### 3.2.4. Problem kısıtları

Karakteristiğine bağlı olarak birçok optimizasyon probleminin kısıtları olabilir. İstenilen sonucu elde etmek için bazen fiziksel bazen de zaman kısıtı olarak görülen problem sınırlılıklarına uymak gerekmektedir (Coello, 2007).

### 3.2.5. Çok amaçlı ve tek amaçlı optimizasyon problemlerinin farkları

Gerçek dünyada mühendislik problemleri, genellikle birden çok amacın hesaba katılmasıyla ele alınırlar. Bu yüzden tek amaçlı optimizasyon algoritmaları bu tür problemler için temelde uygun değildir. MOOP'ların SOOP'lara göre bazı farklılıkları vardır. Deb (2001) bu problemlerin farklarını üç temel başlığa indirgemmiştir.

Birinci olarak, SOOP'larda, bir adet amaç fonksiyonu vardır ve bunun için en uygun değeri karşılayan tek bir çözüm noktası vardır. Gerçek dünya problemlerinde ise neredeyse her problemin birden fazla amaç fonksiyonu olmak zorundadır. Amaçlar arasında bir zıtlık, çelişki yoksa amaçlar tek bir amaca indirgenerek tek amaçlı algoritmaların çözebileceği forma getirilebilirler ancak, problemlerin amaçları birbiri ile çelişiyorsa işte o zaman, bir amaç için iyi sonuç üreten karar değişkenleri, diğer amaç için çok kötü sonuçlar üretebilirler. Bu gibi durumlarda yapılması gereken şey, en az bir amaç için en iyi çözümlerin tespitidir. Böylece birbirini bastıramayan, birbirine üstün olmayan bir dizi çözüm elde edilir. Bu çözüm kümesine Pareto-optimal küme denir. Bu çözüm kümesinden, amaç fonksiyonlarına uygun yakınsamanın yanı sıra yeterli ve dengeli çeşitlilik beklenilir. Bu paradigma ile bakılınca MOOP'ları çözmek SOOP'lara göre çok daha zordur ve maliyetlidir (Deb ve Beyer, 2001).

Optimizasyon problemlerinde, karar değişkenlerinin üst ve alt limitleri arasındaki aralığa arama uzayı veya karar değişkenleri uzayı denir. SOOP'larda amaç fonksiyonu değeri en uygun olan tek bir nokta vardır.

İkinci temel fark ise MOOP'larda iki adet uzay kavramı olmasıdır. İlk uzay, alt-üst limitler arasında rastgele dağıtılmış bireylerin olduğu arama uzayı; ikinci uzay ise, içerisinde birbirine üstün gelemeyen çözümlerin oluşturduğu amaç uzayıdır.

Son fark ise, geçmiş çalışmalarda tek amaçlı algoritmalarla MOOP'ları çözmek için bu problemler üzerlerinde yapay iyileştirmeler yapılmıştır. Dolayısıyla algoritmaları

probleme değil, problemi algoritmalara uyarlamışlardır. Oysaki çok amaçlı çözüm yaklaşımlarında böyle bir durum söz konusu değildir.

MOOP'ları çözmek için literatürde bilinen dört adet çok amaçlı problem çözme yaklaşımı vardır (Özkış, 2017).

Bunlar:

- Ağırlığa dayalı yaklaşımlar (*Weight-based approaches*);
- Ayrıştırma tabanlı yaklaşımlar (*Decomposition-based approaches*);
- Gösterge tabanlı yaklaşımlar (*Indicator-based approaches*);
- Pareto tabanlı yaklaşımlar (*Pareto-based approaches*).

Bu tez kapsamında Pareto tabanlı çözüm yaklaşımları ele alınmıştır.

### 3.3.Pareto Tabanlı Çok Amaçlı Optimizasyon Algoritmaları

İlk olarak 1881'de Edgeworth tarafından önerilen Pareto verimlilik teorisi, daha sonra 1896'da Pareto adlı bir İtalyan bilim adamı tarafından üreticiler ve tüketiciler arasında kaynak tahsisini optimize etmek için geliştirilmiştir. Bu teori kısaca şu şekilde özetlenebilir: Herhangi birinin durumu kötüleşmeden önce bazı kişilerin durumu iyileştirilebiliyorsa, refahın arttığı kabul edilir; hiçbirinin durumu kötüleşmeden, bir kişinin durumu bile düzeleliyorsa, o zaman iyilik hali maksimum noktasına, yani doyuma ulaşmıştır. Bu teoremin MOOP'larda kullanılabileceği ilk defa Goldberg tarafından öne sürülmüştür (Goldberg ve ark., 1989). Bu fikir, araştırmacılara MOGA, NPGA, NSGA ve MODE gibi birçok algoritma geliştirmeleri için ilham verilmiştir (Özkış, 2017).

Pareto tabanlı algoritmanın temel ilkeleri şöyledir: Çözümün kalitesine göre bastırılmayan bireyler sınıflara ayrılırlar, Pareto cephesi olarak bilinen ilk cephede en nitelikli bireylerin, sonra ise ikinci, üçüncü derecede nitelikli bireylerin yer aldığı sınıflara ayrılırlar. Popülasyon içinden, çaprazlama operatörüyle çaprazlama ve belirli bir olasılığa göre mutasyon yapmak için ebeveynler rastgele seçilir, böylece aday bireylerin seçilmesi işlemi tamamlanmış olur. Ana popülasyonun büyüklüğü kadar yani aynı sayıda aday birey üretilir ve her iterasyondan sonra ana popülasyon ile aday bireylerin popülasyonu birleştirilir. Birleştirilmiş yeni popülasyon, ana popülasyonun iki katı olur. Sonra; önce yüzey numaralarına göre, aynı yüzey numaralara sahip bireyler için ise kalabalık mesafe puanlarına göre en kaliteliden en kalitesize doğru bir sıralama yapılır ve sıralama işlemi

bittikten sonra ana popülasyon kadar kısım birleştirilmiş popülasyondan seçilir, birleştirilmiş popülasyonun diğer kısmı ise atılır.

Pareto tabanlı yaklaşımlarda kullanılan temel operatörler şunlardır:

- Hızlı-bastırılmayan sıralama operatörü (*Fast Non-dominated Sorting*);
- Kalabalıklık mesafesi operatörü (*Crowding-Distance*);
- Seçim operatörü (*Selection*);
- Çaprazlama operatörü (*Crossover*);
- Mutasyon operatörü (*Mutation*).

Seçim, çaprazlama ve mutasyon operatörleri bütün MOEA'larda kullanılırken, hızlı-bastırılmayan sıralama ve kalabalık mesafe operatörleri yalnızca Pareto tabanlı yaklaşımlarda kullanılmaktadır (Coello, 2007).

### 3.3.1. Hızlı-baskın olmayan sıralama operatörü

Çok amaçlı algoritmaların nihai amaçlarından biri literatürde Pareto-optimal küme olarak da bilinen birbirlerini bastıramayan bireylerden oluşan optimal kümeyi bulmaktır. Arama uzayında uygun karar değişkeni vektörleri kümesini  $\Omega$  sembolü ile ifade edilirse, bir minimizasyon problemi için şu formül elde edilir:

*Pareto baskınlığı:*  $X, Z \in \Omega$

$$\forall_i \in \{1, 2, \dots, m\} f_i(X) \leq f_i(Z)$$

Eğer  $X < Z$  (3.2)

$$\exists_j \in \{1, 2, \dots, m\} f_j(X) < f_j(Z)$$

Denklem 3.2'de  $X$  çözümü hiçbir amaç fonksiyonunda  $Z$ 'den daha kötü olmamak şartıyla, en az bir amaç fonksiyonunda  $Z$ 'den daha iyi bir değere sahip olursa  $X$  çözümü,  $Z$  çözümünü bastırır denilir ve  $X \succ Z$  şeklinde ifade edilir. Pareto baskınlık kuralının matematiksel ifadesi,  $m$  amaç sayısı olmak üzere bir minimizasyon problemi için Denklem 3.1'de tanımlanmıştır. En az birer amaç fonksiyonunda  $X$  ve  $Z$  çözümleri birbirlerinden daha iyi değer üretmişlerse bu iki çözüm için birbirlerini bastıramayan çözümlerdir denir (Babalik ve ark., 2018).

*Pareto optimal:*

Eğer  $\neg \exists Z \in \Omega: Z \succ X$  ise  $X$ , Pareto-optimaldir. Yani,  $\Omega$  kümesinin hiçbir elemanı  $X$  çözümünü bastırmıyordur.

*Pareto optimal set (PS):*

Karar değişkenleri vektörlerinden oluşan,  $\Omega$  kümesindeki Pareto-optimal çözümlerin kümeleridir. Matematik modeli Denklem 3.3'tedir.

$$PS = \{X \in \Omega \mid \neg \exists Z \in \Omega: Z \succ X\} \quad (3.3)$$

*Pareto optimal yüzey (PF):*

Amaç uzayındaki PS' ye ait çözümlerin konum vektörlerinden oluşur ve Denklem 3.4 ile ifade edilir (Babalik ve ark., 2018).

$$PF = \{F(X) \mid X \in PS\} \quad (3.4)$$

Şekil 3.1'de hızlı-bastırılmayan sıralama yaklaşımının sözde kodu açıklamaları birlikte sunulmuştur. Bu sözde kodun önemli kısımları aşağıda açıklanmıştır.

$p$ ,  $P$  popülasyonunda herhangi bir çözümdür;

$n_i$ ,  $p$  çözümünü bastıran çözümlerin sayısı;

$S_p$ :  $p$  çözümünün bastırıldığı çözümlerin kümesi olmak üzere.

---

```

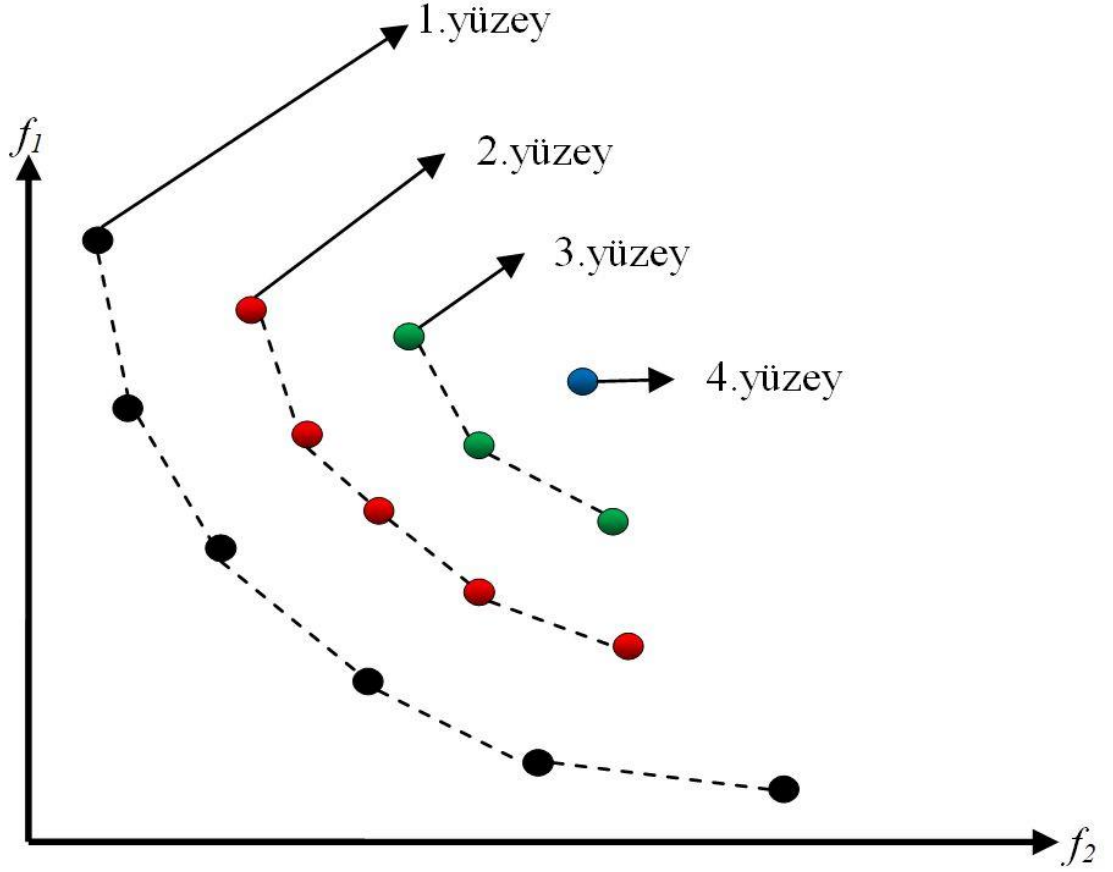
1. for each p in P
  a.  $n_p = 0, S_p = \emptyset$ 
  b. for each q in P
    i. if ( $p < q$ ) //p çözümü q çözümünü bastırıyorsa
      1.  $S_p = S_p \cup \{q\}$  // q çözümü p tarafından bastırılan çözümler kümesine eklenir
    ii. else if ( $q < p$ ) //q çözümü p çözümünü bastırıyorsa
      1.  $n_p = n_p + 1$  //p çözümünün bastırılma sayısı artırılır
    iii. end if
  c. end //for each
  d. if ( $n_p = 0$ ) //p çözümü hiç domino edilmemişse
    i.  $p_{rank} = 1$ 
    ii.  $F_1 = F_1 \cup \{p\}$ 
  e. end if
  f.  $i=1$  //yüzey sayacı başlangıç durumuna getirilir
  g. while  $F_i \neq \emptyset$ 
    i.  $Q = \emptyset$  //sonraki yüzeylerin çözümleri için boşaltılır
    ii. for each  $p \in F_i$ 
      1. for each  $q \in S_p$ 
        a.  $n_q = n_q - 1$ 
        b. if ( $n_q = 0$ ) //q çözümünün bastırılma sayacı 0 ise sonraki yüzeye aittir
          i.  $q_{rank} = i + 1$ 
          ii.  $Q = Q \cup \{q\}$ 
        c. end if
      2. end //for each
    iii. end //for each
    iv.  $i=i+1$ 
    v.  $F_i = Q$ 
  h. end //while
2. end //for each

```

---

**Şekil 3. 1.** Hızlı-bastırılmayan sıralama yaklaşımın sözde kodu (Özkış, 2017)

Hızlı-bastırılmayan sıralama stratejisi kullanılarak yapılan; yüzeylere ayırma işlemi sırasında, ilk yüzeydeki çözümlerin bastırılma sayısı  $n_i$  sıfır olur. İlk yüzey belirlendikten sonra, her bir çözümün bastırıldığı diğer çözümlerin ( $S_p$ ) bastırılma sayısı azaltılır ve bastırılma sayısı sıfır olan çözümler, sonraki yüzeye atanmak üzere  $Q$  kümesine eklenir. Üçüncü yüzeydeki çözümlerin belirlenmesi için,  $Q$  kümesindeki çözümler tarafından bastırılan çözümlerin bastırılma sayısı bir azaltılır ve bastırılma sayısı sıfır olanlar üçüncü yüzeye aktarılır. Bu işlem, tüm çözümler, ait oldukları yüzeye atanana kadar devam eder. Şekil 3.2'de, Hızlı-bastırılmayan sıralama stratejisinin çözümleri yüzeylere ayırmasının temsili bir gösterimi vardır.



Şekil 3. 2. Hızlı-bastırılmayan sıralama yaklaşımında çözümlerin yüzeylere ayrılması

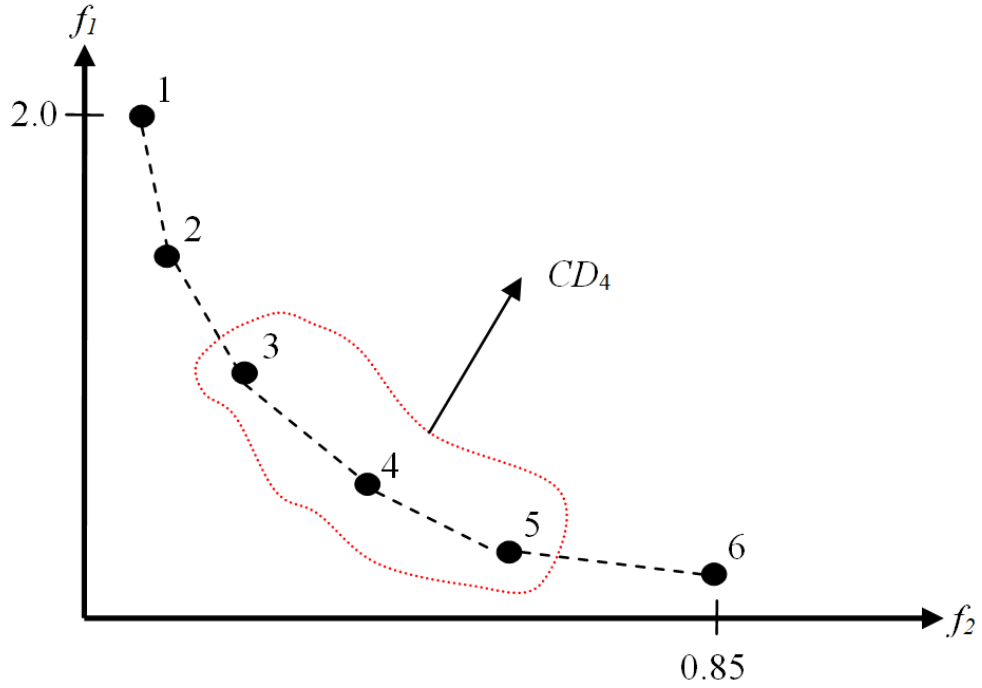
### 3.3.2. Kalabalıklık mesafesi operatörü

Hızlı-bastırılmayan sıralama operatörü, çözümlerin kalitelerine göre yüzeyleri, sınıflara artan sıralama ile ayırır; ancak aynı yüzey numarasına sahip birçok çözüm bulunmaktadır. Bundan dolayı aynı sınıfa ait çözümlerin kalitesini ayırmak için kalabalıklık mesafesi operatörü kullanılır. Bu algoritma ebeveynleri ve nitelikli çözümleri seçerken çözüm yoğunluğunun düşük olduğu yerlerden çözümler seçerek, modelin çeşitlilik performansını arttırmayı amaçlamaktadır. Ekstrem çözümler (uç noktadaki çözümler) için, kalabalıklık mesafesi değeri sonsuza ayarlanırken, geri kalan çözümler için kalabalıklık mesafesi değeri, her bir çözüm ile komşu çözümler arasındaki normalleştirme ve toplama yoluyla hesaplanan farktır. Çizelge 3.1’de kalabalıklık mesafesinin sözde kodu verilmiştir.

Çizelge 3. 1. Kalabalıklık mesafesi sözde kodu (Deb ve ark., 2002)

$l =  I $	$I$ 'daki çözüm sayısı
for each $i$ , set $I[i]_{mesafe} = 0$	mesafeleri sıfırla
for each amaç $m$	
$I = \text{sort}(I, m)$	$m$ . amaca göre çözümleri sırala
$I[1]_{mesafe} = I[l]_{mesafe} = \infty$	uç çözümlere sonsuz mesafe ata
for $i = 2$ to $(l - 1)$	aradaki çözümler için
$I[i]_{mesafe} = \frac{I[i]_{mesafe} + (I[i+1].m - I[i-1].m)}{(f_m^{\max} - f_m^{\min})}$	Kalabalıklık mesafesini hesapla
end for $i$	
end for each $m$	

Şekil 3.3'te bir örnek tasarlanmıştır. Buna göre uç noktalar yani ekstrem çözümler sonsuza ayarlanmaktadır ( $CD_1=inf$  ve  $CD_6=inf$ ), sonrasında ise 4. Çözümün komşu değerleri olan 3. çözüm ve 5. çözüme göre; her bir amaç için ayrı-ayrı olmak üzere farklar alınır ve normalize edilerek toplanır (Özkıış, 2017; Karakoyun ve ark., 2020).



Şekil 3. 3. Kalabalıklık mesafesi işlemi için 4 numaralı çözüm ve komşuları

### 3.3.3. Seçim operatörü

Ana popülasyonu sürekli geliştirmek için ebeveyn çözümleri çaprazlayarak yeni aday çözümler oluşturulur. Bu çaprazlama işleminde kullanılmak üzere, çözümleri seçerken literatürde en sık kullanılan yöntemler şöyledir (Ergül, 2010):

- Rastgele seçim mekanizması (*The random selection mechanism*);
- Turnuva seçim mekanizması (*The tournament selection mechanism*);
- Rulet tekeri seçim mekanizması (*The roulette wheel selection mechanism*);
- Sıralı seçim mekanizması (*The sequential selection mechanism*).

### 3.3.4. Çaprazlama operatörü

Çaprazlama (*crossover*) işlemi, ana popülasyondan iki ya da daha fazla çözümün belirli işlemlerden geçirilerek yeni bir çözüm elde etme sürecidir. Çaprazlamanın amacı, her aşamada popülasyonun daha kaliteli çözümlerden oluşması işlemidir. MOEA’larda yaygın şekilde kullanılan çaprazlama operatörleri şöyledir (Durillo ve Nebro, 2011):

- Simüle edilmiş ikili çaprazlama (*Simulated Binary crossover, SBX*);
- Diferansiyel evrim çaprazlaması (*Differential Evolution crossover, DE*);
- BLX alfa çaprazlama (*BLX Alpha crossover, BLX*);
- Tek noktalı çaprazlama (*Single Point crossover, SP*);
- ABC çaprazlama (*ABC crossover*).

Bu tez kapsamında kullanılan çaprazlama operatörleri için, bu bölümde gerekli bilgiler verilecektir. DE çaprazlaması, farklar yaklaşımı olarak da bilinir. Farklar yaklaşımına göre ana popülasyondan rastgele üç adet çözüm ( $x_i, x_k, x_z$ ) seçilir, daha sonra iki çözümün  $j$ . farkları alınır ve sabit bir değer (ölçek faktörü) olan  $F$  ile çarpılır ( $F$  değeri literatürde çoğunlukla 0.5 olarak kullanılmaktadır), bu değer, üçüncü çözümün  $j$ . değeri ile toplanır ve yeni aday çözümün bir karar değişkeni (*offSpring<sub>ij</sub>*) elde edilmiş olur. İlgili çaprazlama modelinin matematiksel eşitliği Denklem 3.5’te ifade edilmiştir.

$$offSpring_{ij} = x_{ij} + F(x_{kj} - x_{zj}) \quad (3.5)$$

Denklem 3.6 ve Denklem 3.7’de ise ABC algoritmasının çaprazlama modelinin matematiksel ifadeleri gösterilmiştir. Aslında ABC algoritmasında da bir çeşit farklar yaklaşımı kullanılmıştır yalnız sabit bir değer olan  $F$  yerine dinamik bir formül yani Denklem 3.6’daki ifade kullanılmıştır. Bu formüle göre 0 ve 1 arasında rastgele üretilen bir sayıdan 0.5 çıkarılıyor ve çıkan sonuç 2 ile çarpılarak çaprazlamada kullanılacak  $\Phi$  değeri üretiliyor. İlgili değer, sabit bir değer olan  $F$  değerine üstünlüğü şöyle ki;  $\Phi$  eksi değerlerde alabiliyor (-1 ve +1 arasında değişen değerler) ve böylece farklar yaklaşımına



etkisi daha fazla oluyor. Dolayısıyla daha etkili bir çeşitlilik sağlanmakta ve arama uzayı daha iyi şekilde taranmaktadır.

$$\Phi = (rand - 0.5) * 2 \quad (3.6)$$

$$offSpring_{ij} = x_{ij} + \Phi(x_{ij} - x_{kj}) \quad (3.7)$$

Dikkat edilirse Denklem 3.7 ise  $\Phi$  dışında Denklem 3.5 ile neredeyse aynı eşitliktir, yani ABC çaprazlaması da bir farklar yaklaşımı modelidir.

BLX alfa çaprazlamasında ise, rastgele iki çözümün  $j$ . değerleri seçilir ve büyük olanından küçük olanın farkları alınır ve “range” değeri bulunur. Sonraki işlemler 3.8-3.12. eşitliklerde gösterilmiştir.

$$\left[ \begin{array}{l} \text{if } (x_{k,j} > z_{z,j}) \text{ max} = x_{k,j}, \text{ min} = z_{z,j} \\ \text{else max} = z_{z,j}, \text{ min} = z_{k,j} \end{array} \right] \quad (3.8)$$

$$range = max - min \quad (3.9)$$

$$minRange = min - range * \alpha \quad (3.10)$$

$$maxRange = max + range * \alpha \quad (3.11)$$

$$offSpring_{ij} = minRange + rand(0,1) * (maxRange - minRange) \quad (3.12)$$

### 3.3.5. Mutasyon operatörü

MOEA’larda mutasyon işlemi popülasyondaki çeşitliliğe katkı sağlamak için kullanılır. Genelde çok küçük bir olasılıkla mutasyon işlemi gerçekleşir ve genelde bir çözümün sadece bir değişkeninde rastgele bir atama şeklinde gerçekleştirilir. Literatürde kullanılan en yaygın mutasyon operatörleri şöyledir (Durillo ve Nebro, 2011):

- Polinomal mutasyon (*Polynomial mutation*);
- Gauss mutasyon (*Gaussian mutation*);
- Basit rastgele mutasyon (*Simple random mutation*);
- Takas mutasyon (*Swap mutation*);
- Çevrim mutasyon (*Flip mutation*).

### 3.4. Kullanılan Problem Setleri

Bu tez kapsamında toplamda 22 adet test fonksiyonu kullanılmıştır. Bunlardan 3 tanesi *ZDT* problem ailesinden, 7 tanesi *DTLZ* ailesinden, 9 tanesi *WFG* ailesinden ve 3 adette klasik test fonksiyonlarından oluşmaktadır.

#### 3.4.1. ZDT problem seti

Yaygın şekilde kullanılan problem ailelerinden biri olan *ZDT* problem ailesi, iki amaçlı olarak tasarlanmıştır ve ismini problemi tasarlayan yazarlardan (*Zitzler, Deb ve Thiele, ZDT*) almıştır (*Zitzler ve ark., 2000*). Toplamda iyi tasarlanmış 6 adet *ZDT* fonksiyonu olmasına karşın, bu tez kapsamında *ZDT 1-3* problemleri kullanılmıştır. Bu tercihin sebebi *ZDT* problem setinin diğer yarısında bulunan *ZDT5* probleminin süreksiz problemler için tasarlanmasıdır, yani bütünlük bozulmasın diye bu problem ailesinin ilk yarısını kullanmak tercih edilmiştir.

#### 3.4.2. DTLZ problem seti

Bu problem ailesi, ölçeklenebilir uygunluk (*fitness*) boyutlarına sahip olan çok amaçlı problemler için tasarlanmıştır. Adını ise yazarlarının *Deb, Thiele, Laumanns ve Zitzler* baş harflerinden almıştır. Toplamda 7 adet *DTLZ* problemi vardır. Bu tez kapsamında tamamı kullanılan bu problem setlerinin tamamı, kutu kısıtlamalı (*box-constrained*) ve süreklidir (*Deb ve ark., 2005*).

#### 3.4.3. WFG problem seti

Bu problem ailesi 9 problemden oluşmuştur ve daha önceki tasarlanan test paketlerinin işlevlerini aşacak şekilde tasarlanmıştır. Özellikle ayrılmayan (*non-separable*), yanıltıcı (*deceptive*), dejeneratif (*truly degenerative*) problemler, aynı zamanda hem amaç hem de değişken sayısında ölçeklenebilir ve karışık Pareto cephelerine sahip problemler ayrıntılı şekilde tasarlanmıştır. *WFG* problem test paketi *Simon Huband, Luigi Barone, Lyndon While ve Phil Hingston* isimli yazarlar tarafından önerilmiştir (*Huband ve ark., 2006*).

### 3.4.4. Klasik problemler

Birçok klasik test fonksiyonu olmasına karşın bu tez çalışması kapsamında 3 adet klasik test fonksiyonu kullanılmıştır. Bunlar: *Fonseca-Fleming* fonksiyonu, *Kursawe* fonksiyonu ve *Schaffer* fonksiyonudur.

*Fonseca-Fleming* fonksiyonu, kutu kısıtlamalı ve iki amaçlı olarak tasarlanmıştır. Bu problemin amaç fonksiyonu dışbükey olmadığından (*non-convex*), dışbükey olmayan MOOP'tur (*Fonseca ve Fleming, 1998*).

*Kursawe* test fonksiyonu ise, 2 amaçlı ve 3 boyutludur. Bu problemde dışbükey değildir ve parçalı yapıda bir Pareto-optimal'a sahip olduğundan dolayı optimize edilmesi zor olan bir problemdir (*Kursawe, 1990*).

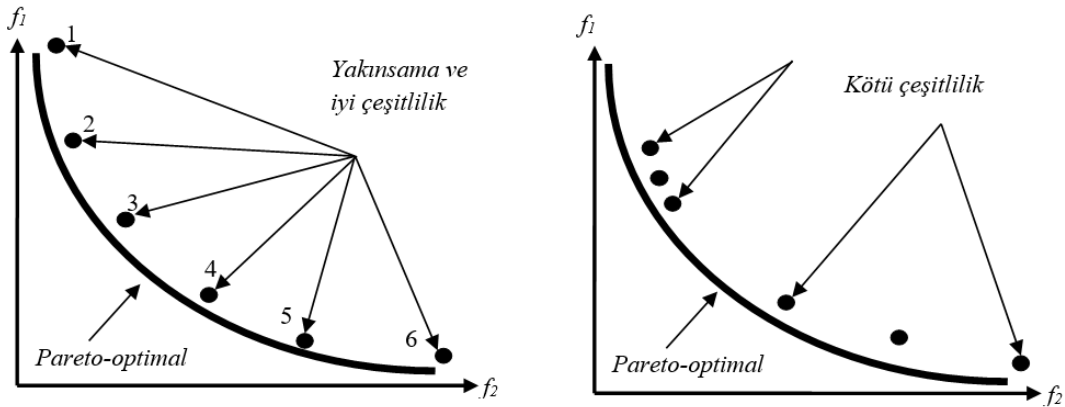
*Schaffer* problemi, karakteristik özellikleri tek modlu (*unimodal*), sürekli, dışbükey olmayan, türevlenebilir (*differentiable*), işlevsel olarak ayrılmaz (*non-separable*) iki boyutlu bir uzayda tanımlanmıştır (*Schaffer, 1985*).

### 3.5. Bu Çalışmada Kullanılan Metrikler

Tek amaçlı evrimsel algoritmalarda, herhangi bir algoritmanın sonucunun diğerleri ile karşılaştırılması zor değildir. Sadece tek bir amaca yönelik olan bu tür algoritmalarda amaç fonksiyonunun değerlendirilmesi ile ortaya çıkan uygunluk (*fitness*) değerine bakmak yeterlidir. Eğer problem minimizasyon problemi ise uygunluk değerinin daha küçük olması ilgili sonucun daha iyi olduğu anlamına gelmektedir. Eğer problem maksimizasyon problemi ise sonucun daha büyük bir değer olması daha kaliteli olduğunu gösterir. Tek amaçlılarda bu böyle iken, çok amaçlı alanında tek bir sonuç yerine, bir çözüm uzayı (Pareto-optimal) elde edildiğinden dolayı, üretilen sonuç bir değer değil, birçok değerden oluşan bir matristir. Buradaki problem bir matrisin diğer bir matris ile nasıl kıyaslanacağıdır. Bu problemi çözüme kavuşturmak amacıyla, araştırmacılar tarafından performans metrikleri geliştirilmiştir. Bu metrikler algoritmaların bazı özelliklerini ölçerler. Bunlardan iki tanesi yakınsama (*convergence*) ve çeşitliliktir (*diversity*).

MOEA'lar tarafından elde edilen matrisin, gerçek Pareto-optimal yüzeye mümkün olduğunca yakın olması hedeflenir. Bu işlemde iki duruma bakılır: Bunlardan biri yakınsamadır; yani algoritmanın tahmin ettiği yüzeyin, gerçek Pareto-optimal yüzeye ne kadar yakın olduğudur. İkincisi ise çeşitliliktir, algoritmanın tahmin ettiği yüzeyin gerçek yüzeye göre isabetli dağılım miktarıdır.

Şekil 3.4'te yakınsamanın nasıl gerçekleştiği ve iyi çeşitlilikle kötü çeşitlilik arasındaki farklar gösterilmiştir. Kötü çeşitlilik göstermiş bir yüzeyde çözümler düzensiz şekilde; yani birden çok çözüm bir arada öbikleşir ve bu gruplaşmış çözümler ise birbirinden uzak şekilde dizilmişlerdir. İyi çeşitlilik göstermiş olan bir sonuca göre ise, bütün sonuçlar birbirine eşit mesafe çok düzenli şekilde dizilirler (Ergül, 2010; Özkış, 2017)

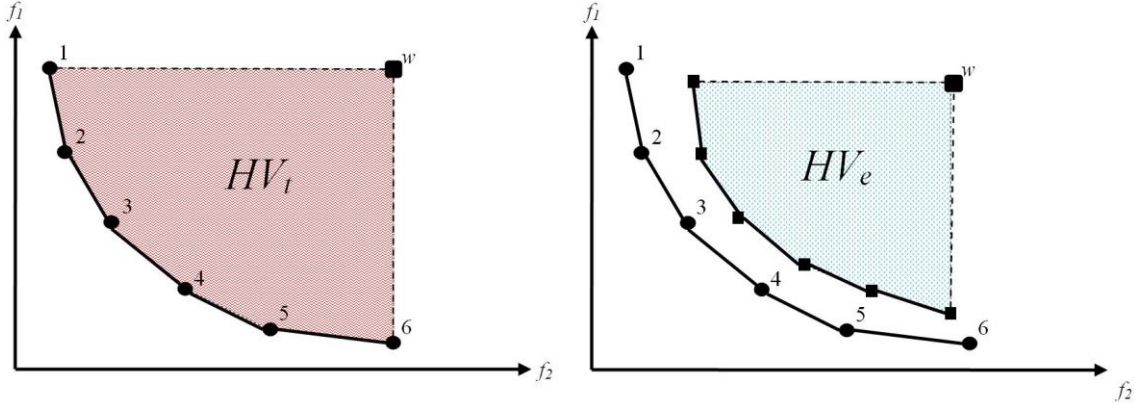


Şekil 3. 4. Yakınsama ve iyi çeşitlilik, kötü çeşitlilik temsili gösterimi

### 3.5.1. Hiperküp metriği

Hiperküp metriği (*Hypervolume*,  $HV$ ), minimizasyon problemleri için uygundur ve algoritmanın tahmin ettiği çözüm uzayının (*Pareto Front estimated*,  $PF_e$ ) çeşitlilik ve yakınsama performanslarını ölçmektedir. Kaç adet amaç fonksiyonu varsa yani, problem kaç amaçlıysa  $PF_e$  içerisinde amaçlar için en kötü sonucu veren vektörler referans noktası olarak alınır. O referans noktaları ile tüm arama uzayı arasında kalan alan hacimsel olarak toplanır. Bu alana tahmini hypervolume (*Hypervolume estimated*,  $HV_e$ ) denir. Bu işlem, gerçek  $PF_t$  (*Pareto Front true*) içinde yapılır ve bulunan değere gerçek hypervolume (*Hypervolume true*,  $HV_t$ ) denir. Sonuç olarak  $HV_t$  değerinin  $HV_e$ 'ye oranının normalize edilmesi ile nihai değer olan  $HV$  değeri elde edilir. Buna göre bu oran 1'e ne kadar yakınsa o kadar iyi kabul edilir. Oranlar 1'i geçemeyeceğinden  $HV$  ne kadar büyük bir değerse performans o kadar iyidir denilebilir (Özkış, 2017).

Şekil 3.5'te Hiperküp metriği hesabının yapılabilmesi için gerekli olan, İki uç değerler arasındaki referans noktası ( $w$ ) ile gerçek çözümler arasında kalan alan  $HV_t$  ve referans noktası ile tahmini çözümlerin arasında kalan alan  $HV_e$  alanlarının hacimsel olarak hesaplanması temsili olarak gösterilmiştir.



Şekil 3. 5. Hiperküp metriğinin hesaplanmasının temsili gösterimi

Denklem 3.13- Denklem 3.15'te ise ilgili hacim alanlarının hesaplanmasının matematiksel modeli görünmektedir. Kısaca HV; tahmini değerlerin kapladığı hacmin, gerçek değerlerin kapladığı hacme oranıdır denilebilir (Özkış, 2017).

$$HV_t = volume \left( \bigcup_{i=1}^{|PF_t|} v_i \right) \quad (3.13)$$

$$HV_e = volume \left( \bigcup_{i=1}^{|PF_e|} v_i \right) \quad (3.14)$$

$$HV = \frac{HV_e}{HV_t} \quad (3.15)$$

### 3.5.2. Nesilsel mesafe

Nesilsel mesafe (*Generational Distance*, GD) metriği ile, algoritmaların yakınsama performansı değerlendirilir. Algoritmanın bulduğu her bir çözüm ile gerçek çözüm arasındaki Öklid mesafesini hesaplanır. Öklid mesafelerinin ortalamaları alınır ve nihai sonuç elde edilir. GD metriği ne kadar küçük olursa gerçek sonuca o kadar yakın olacağından, GD değerinin daha küçük olması daha iyi bir performans anlamına gelir. GD metriğine göre en ideal sonuç 0'dır.

Denklem 3.16 ile M amaçlı bir problem için  $i \in PF_e$  çözümünün  $PF_t$ 'de en yakın olduğu çözümle arasındaki Öklid mesafesi hesaplanır. Bu denklemde  $f_m^{e,i}$   $PF_e$ 'deki  $i$ . çözümün  $m$ . amaç fonksiyonu değerini,  $f_m^{t,n}$   $PF_t$ 'deki  $n$ . çözümün  $m$ . amaç fonksiyon

değerini belirtir. Her  $i \in PF_e$  çözümü için  $d_i$  değerleri hesaplandıktan sonra Denklem 3.17'ye göre GD metriği hesaplanır.

GD hesaplanırken güvenilir bir metrik hesabı için  $PF_t$ 'deki çözüm sayısının fazla olması önemlidir. GD metriğinin bir dezavantajı, az sayıda ve belirli yerde toplanmış bazı çözümler çok iyi sonuçlar verebilirler ve sanki bütün sonuçlar iyiymiş gibi bir yanılsama oluşabilir. Bu eksikliği gidermek için ters nesilsel mesafe metriği önerilmiştir (Karakoyun ve ark., 2020).

$$d_i = \min_{n=1}^{PF_t} \sqrt{\sum_{m=1}^M (f_m^{e,i} - f_m^{t,n})^2} \quad (3.16)$$

$$GD = \frac{\left(\sum_{i=1}^{|PF_e|} d_i^M\right)^{1/M}}{|PF_e|} \quad (3.17)$$

### 3.5.3. Ters nesilsel mesafe

Denklem 3.18'de matematiksel eşitliği verilen, ters nesilsel mesafe (*Inverted Generational Distance*, IGD) metriği, GD metriğinin terslenmiş versiyonudur. Bu denkleme göre her  $PF_t$  değeri ile kendisine en yakın olan algoritma tarafından bulunmuş  $PF_e$  değerinin arasındaki Öklid mesafesi hesaplanır. GD metriğinde  $PF_e$  ile en yakın  $PF_t$  arası hesaplanırken, IGD metriğinde  $PF_t$  referans alınarak en yakın  $PF_e$  ile arasındaki öklit mesafesinin hesaplanması aralarındaki önemli farkı oluşturmaktadır (Özkış, 2017).

$$IGD(PF_e, PF_t) = \frac{\sum_{X \in PF_t} d(X, PF_e)}{|PF_t|} \quad (3.18)$$

### 3.6. Performans Göstergeleri

Bu bölümde, tez boyunca yapılan bütün çalışmalarda kullanılan; ortalama ve standart sapma, Pareto-optimal yüzey, kutu grafikleri, Wilcoxon testleri hakkında genel bilgiler verilmiştir.

### 3.6.1. Ortalama ve standart sapma

Bir optimizasyon algoritması geliştirildiğinde, performansının diğer algoritmalara göre ne kadar iyi olduğunu belirlemek için istatistiksel analizler yapılır. Bu analizlerde, özellikle algoritmanın başarısını ölçmek için kullanılan metriklerin ortalama ve standart sapma gibi istatistiksel ölçümlerinin kullanılması yaygındır.

Çizelge 3.2’de bir örneği gösterilen bu ölçümler, algoritmanın performansının objektif bir şekilde ölçülmesine yardımcı olur. Öncelikle ilgili veriler algoritmaların en az 20 bağımsız çalışması sonucu elde edilirler. Örneğin HV metriği kapsamında birbirinden bağımsız çalışmaların ortalaması, algoritmanın performansının bir özetidir ve diğer algoritmalara göre ne kadar iyi olduğunu belirlemek için kullanılır.

**Çizelge 3. 2.** Örnek bir ortalama, standart sapma gösterimi

	NSGA-II		MOAAA		MOEA/D		MOEA/D-TS		Önerilen Alg.	
	Ort.	S.s.	Ort.	S.s.	Ort.	S.s.	Ort.	S.s.	Ort.	S.s.
ZDT1	6.58E-01	8.20E-04	6.58E-01	2.60E-03	2.91E-01	6.10E-02	6.63E-01	2.80E-04	6.65E-01	4.40E-05
ZDT2	3.21E-01	1.60E-03	3.22E-01	3.80E-03	1.38E-02	3.20E-02	3.29E-01	2.40E-04	3.31E-01	5.80E-05
ZDT3	5.11E-01	8.20E-04	5.11E-01	1.50E-03	2.41E-01	3.90E-02	5.16E-01	1.10E-04	5.17E-01	7.90E-05

Bağımsız çalışmaların sonuçlarından elde edilen standart sapma ise, algoritmanın performansının ne kadar tutarlı olduğunu ve diğer algoritmalara göre ne kadar değişken olduğunu belirlemek için kullanılır. Bu istatistiksel ölçümler, algoritmanın geliştirilmesi veya iyileştirilmesi için de önemli ipuçları sağlayabilir. Bu nedenle, yeni bir optimizasyon algoritması önerildiğinde, ortalama ve standart sapma gibi istatistiksel ölçümlerinin kullanılması yaygın bir uygulamadır.

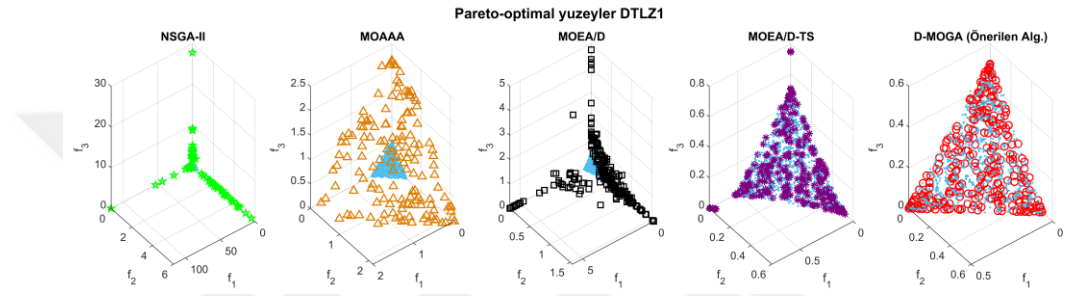
### 3.6.2. Pareto-optimal yüzeyler

Pareto-optimal yüzey, MOOP’larda gerçek çözümlerin oluşturduğu yüzeydir. Bu yüzeyde, bir amacın değerinin artırılması diğer amaçların değerini azaltır, dolayısıyla optimal çözümler, hiçbir amaç fonksiyonunun diğerlerine zarar vermediği, yani etkileşimsiz olduğu noktalardır. Bu noktaların tümü Pareto-optimal olarak adlandırılır. Pareto-optimal yüzeyler, çoğu zaman karar vericilere farklı amaçlar arasındaki dengeleme noktalarını (*trade-off*, takas ya da ticaret noktaları da denir; iki ya da daha fazla seçenek arasından karar vermeyi gerektiren durumlar) göstermek için kullanılır.

Örneğin, bir araç üreticisi, bir aracın performansını ve yakıt tüketimini optimize etmek isteyebilir. Bu durumda, Pareto-optimal yüzey, aracın performansı ve yakıt

tüketimi arasındaki dengeleme noktalarını gösterir ve üretici, kararını Pareto-optimal yüzeydeki bir noktadan seçebilir. Pareto-optimal yüzey ayrıca, farklı çözüm önerilerinin karşılaştırılması ve tercih edilen amaçlara göre en uygun olanların seçilmesi için kullanılabilir. Pareto-optimal yüzey, karar vericilerin, bir veya birden fazla amaç fonksiyonunun değerini optimize etmek için farklı stratejiler kullanmalarına olanak tanır.

Şekil 3.6'da örnek bir Pareto-optimal yüzey çizimleri vardır. Turkuaz olan kısımlar gerçek Pareto-optimal yüzeylerken, diğer renklerle gösterilen çizimler algoritmaların performanslarını ifade etmektedir.



Şekil 3. 6. Örnek Pareto-optimal yüzeyler (Altiok ve ark., 2023)

Pareto-optimal yüzeyler bu tezde iki ayrı hedef için kullanılmıştır. Birincisi, Algoritmanın bulunduğu  $PF_e$ 'nin, gerçek yüzey olan  $PF_i$ 'ye ne kadar yakınsadığını görmek için, ikinci olarak önerilen algoritmanın diğer algoritmalara kıyasla ne kadar yakınsama yaptığını tespit için kullanılmıştır.

### 3.6.3. Kutu grafikleri

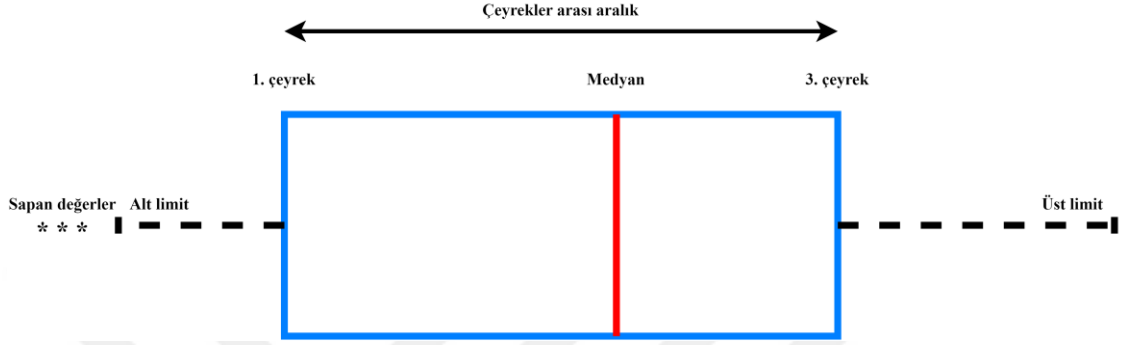
Kutu grafikleri (*Box-plot*), veri analizi sırasında literatürde sıkça kullanılan bir görselleştirme aracıdır. Verilerin merkezi eğilimi, yayılımı ve aykırı değerleri hakkında bilgi edinmek için kullanılır. Beş sayı özetlemesi adı verilen en küçük değer, en büyük değer, alt çeyrek, üst çeyrek ve medyan değerleri kutu grafikleriyle görselleştirilir.

Kutu grafikleri, veri dağılımının simetrikliği veya çarpıklığı, medyan değerinin kullanımının uygunluğu, aykırı değerlerin sıklığı ve yoğunluğu gibi konularda karar vermeyi kolaylaştırır. Ayrıca, farklı gruplar arasındaki veri dağılımını karşılaştırmak veya bir grup içindeki farklı değişkenlerin dağılımını analiz etmek için de kullanılabilir.

Şekil 3.7'de görülen kutu grafiği dağılımlarına göre: Alt limit, sapan (aykırı) değerler hariç, veri setinde bulunan en düşük değerdir ve sol ucun sonunda gösterilir. Üst limit, sapan değerler hariç, veri setinde bulunan en yüksek değerdir ve sağ ucun sonunda



gösterilir. Medyan, ortanca değerdir. Birinci çeyrek, birinci dörtte birlik ve üçüncü çeyrek üçüncü dörtte birlik değerlerdir. Bu hesaplamalar yapılabilmesi için veri kümesindeki ilgili özellik değerleri küçükten büyüğe doğru sıralanması gerekmektedir. Kutunun sol ve sağ sınırları, sırasıyla ilk %25'lik değer ve %75'lik değer temsil eder. Kutunun içindeki çizgi (medyan) %50'lik değerde çizilir (Güçlü, 2012).



Şekil 3. 7. Kutu grafiği temsili gösterimi

Kutu grafikleri, verilerin hızlı ve kolay bir şekilde görselleştirilmesini sağlar, böylece analiz süreci daha anlaşılır ve verimli hale gelir. Bu nedenle, araştırmacılar ve diğer veri kullanıcıları tarafından sıkça tercih edilen bir görselleştirme aracıdır. Ayrıca, sapan değerleri belirlemek için de kullanılabilirler.

#### 3.6.4. Wilcoxon testi

Wilcoxon testi (WT), istatistiksel veri analizinde kullanılan bir hipotez testidir. ve normal dağılıma sahip olmayan örneklem verilerinin karşılaştırılması için kullanılır. Bu test, bağımsız iki örneklemin sıralama verilerine dayanır. WT verilerin normal dağılıma sahip olma varsayımı gerektirmediği için non-parametrik bir testtir. WT, parametrik testlerin güçsüzlüğü durumunda veya verilerin normal dağılımdan uzak olduğu durumlarda kullanılabilir. WT, iki grup arasındaki farkların anlamlılığını belirlemek için ve ayrıca, aynı grup içindeki farklı örneklemelerin karşılaştırılmasında da kullanılabilir. Bu test, veri analizi ve karar verme süreçlerinde önemli bir araçtır ve özellikle küçük örneklemeler veya normal dağılıma sahip olmayan veriler ile çalışan araştırmacılar için faydalıdır (Conover, 1999).

Bir örnek verilecek olursa  $N_{ds}$  veri kümesinin  $i$ . çıkışı için iki algoritmanın ürettiği sonuçları arasındaki fark  $d_i$  olsun. Bu farkların mutlak değeri alınarak sıralanır ve her farka bir işaret değeri atanır. İlk algoritmanın, ikinci algoritmayı geçtiği durumlardaki sıra

değeri toplamı  $R^+$ ; ikinci algoritmanın, ilk algoritmayı geçtiği durumlardaki sıra değerlerinin toplamı ise  $R^-$  olarak kabul edilirse;  $d_i=0$  olduğu durumlar varsa ve sayısı çiftse ikiye bölünüp toplamlara eklenir, tekse bir durum ihmal edilir ve sayısının çift olması sağlanır. Denklem 3.19'da  $R^+$  ve Denklem 3.20'de  $R^-$  değerlerinin hesaplanması verilmiştir (Alcalá ve ark., 2011; Acılar, 2013).

$$R^+ = \sum_{d_i > 0} \text{sıra}(d_i) + \frac{1}{2} \sum_{d_i = 0} \text{sıra}(d_i) \quad (3.19)$$

$$R^- = \sum_{d_i < 0} \text{sıra}(d_i) + \frac{1}{2} \sum_{d_i = 0} \text{sıra}(d_i) \quad (3.20)$$

$$T = \min(R^+, R^-) \quad (3.21)$$

Denklem 3.21'de verilen  $T$  değeri Wilcoxon işaretli sıra testi kritik değerinden küçük yada eşitse, sıfır hipotezi reddedilir, değilse kabul edilir (Kartal, 2006; Acılar, 2013).

Bu çalışmada WT; HV, GD ve IGD metrikleri kapsamında, toplamda 22 test problemi üzerinde 100 bağımsız çalışmada önerilen algoritma ile karşılaştırılan algoritmaların ürettiği değerler arasında anlamlı bir fark olup olmadığını tespit etmek için kullanılmıştır.

## 4. ÖNERİLEN YÖNTEMLER VE DENEYSEL ÇALIŞMALAR

Bu bölümde ele alınan problemlerin tanımları ve onları çözmek için geliştirilen algoritmalar tanıtılmıştır. Bunun yanında önerilen algoritmaların literatürde bilinen bazı önemli algoritmalarla karşılaştırılmaları ve bu algoritmalar tarafından üretilen sonuçların istatistiksel verileri sunulmuştur. Son bölümde ise problemleri çözmek için geliştirilen iki algoritma F-MODE ve D-MOGA karşılaştırılmış, test problemlerinin çözümündeki performansları ve etkinlikleri dikkate alınarak dikkatlice incelenmiş ve yorumlanmıştır.

### 4.1. Hiper Parametre Optimizasyonu Problemi İçin F-MODE Yaklaşımı

Bu bölümde, CNN tabanlı mimarilerinin hiper parametre optimizasyonu probleminin tanımı yapılmış ve gerekli ayrıntılar verilmiştir. Bunların yanında, bu problemi çözmek için geliştirilen F-MODE algoritması açıklanmıştır. Bu algoritmanın geliştirildiği süreçte önerilen ve performansı olumlu yönde etkileyen orijinal katkılar ayrıntılı şekilde açıklanmıştır. Sonra algoritmanın test fonksiyonları üzerindeki performansı diğer algoritmalarla kıyaslanmış ve gerekli istatistiksel testler yapıldıktan sonra ana problem üzerinde algoritma yürütülmüştür. Bu bölümün son kısmında ise sonuçlar hakkında gerekli değerlendirmeler yapılmıştır.

#### 4.1.1. Problemin tanımı

Kaynak araştırması bölümünde CNN mimarilerinin ne olduğu hakkında gerekli bilgiler verildiğinden, bu kısımda daha önce verilen açıklamalara ek olarak problemin tanımı derinlemesine açıklanmıştır.

İlk başlarda sadece görüntü sınıflandırma problemlerinde kullanılan CNN mimarileri, daha sonra segmentasyon ve algılama problemlerine de uyarlanmıştır (Girshick, 2015; Redmon ve ark., 2016; He ve ark., 2017; Ren ve ark., 2017). Bu mimarilerde temel CNN mimarilerine ek olarak daha fazla parametre olmasına karşılık, aynı şekilde optimize edilebildiği anlaşılmıştır. Bu gerçeği göz önünde bulundurarak, CNN tabanlı yöntemler için hiper parametre optimizasyonu yapmayı amaçlayan çalışmalardan kaynak araştırması kısmında bahsedilmiştir.

Bu çalışmanın temelleri İnik ve arkadaşlarının (2019) yılında yapmış olduğu ovaryum dokusuna ait bir çalışmasına dayanmaktadır. İlgili çalışmada ovaryum görüntülerinin segmentasyonu ve sınıflandırılması için Derin Öğrenme tabanlı yeni bir yöntem önerilmiştir. Önerilen yöntemde ovaryum dokusuna ait primordial, primer,

preantrel, sekonder ve tersiyerden oluşan beş farklı follükülün otomatik sayılması gerçekleştirilmiştir. Bu tez çalışmasında ise bu çalışmanın parametreleri çok amaçlı olarak yeniden tasarlanmıştır. Bu çalışmada iki önemli amaç ön plana çıkmaktadır. Bunlar CNN tabanlı modelin ne kadar doğru sonuç üretebildiği ve bu modelin ne kadar hızla sonuç ürettiğidir. Burada bu iki amacı doğrudan etkileyen üç parametre eş zamanlı olarak optimize edilmiştir. Bu parametreler: Alt görüntüler için adım değerini temsil eden genel adım (*General Stride, GS*), Alt görüntü sınıflandırma olasılığını temsil eden yama doğruluğu (*Patch Accuracy, PA*) ve düşük olasılıkla sınıflandırılan alt görüntü komşu yakınlığını temsil eden komşu mesafesi (*Neighbour Distance, ND*)'dir.

İnik ve arkadaşları ilk defa *GS, ND, PA* parametrelerini önerdikleri CNN tabanlı bir sistemde kullanılmışlardır (İnik ve ark., 2019). Bu tez çalışmasını literatürdeki diğer çalışmalardan ayıran özellik ise ilgili problemin çok amaçlı olarak düşünülmesidir.

Bu parametreler genellikle tecrübelerle bağlı olarak manüel ayarlanmaktadır. Ancak teknolojinin ve bilimin bu denli geliştiği bu günlerde manüel yöntem ilkel kalmakta ve beklenen başarı için uygun değerleri karşılamamaktadır. Bu yüzden yapılan çalışma gereklidir ve özellikle de çok daha fazla parametreye sahip olan mimarilerde kullanılabileceği varsayılmaktadır.

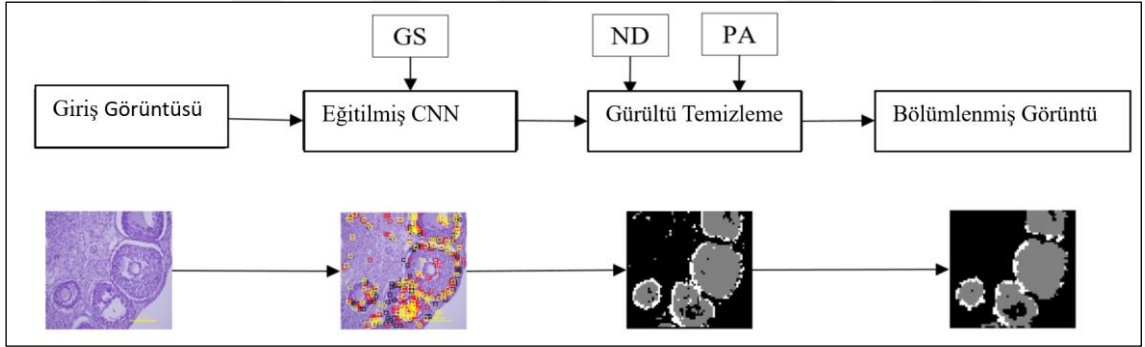
Bu problemin en büyük zorluklarından biri ise, CNN mimarilerinin, en üstün teknolojiler kullanılsa bile çok yavaş çalışmalarınıdır. Bu yüzden bu tarz mimariler, meta-sezgisel optimizasyon algoritmaları gibi temel felsefesi deneme yanılmaya dayanan, algoritmalar için uygun değildir. Normalde bir optimizasyon algoritması test fonksiyonları üzerinde denenirken literatüre göre maksFes 25,000-500,000 arası ayarlanır. Böyle bir maksFes sayısı CNN tabanlı bir mimariye uygulandığında sonuç almak için aylarca beklemek gerekebilir. Tez çalışmasında bu zorluktan ilham alarak hızlı yakınsama yapabilen ve aynı anda zaman ve doğruluğu optimize edebilen bir algoritma geliştirmek hedeflenmiştir. Hipotez ise; eğer rulet tekeri seçim sistemi çaprazlama için ebeveyn seçiminde kullanırsa, yani seçkin bir yaklaşımla çaprazlama yapılırsa çok daha az maksFes ile kabul edilebilir bir sonuç üretilebilir olacaktır. Doğal olarak tezin bu kısmı da ikiye ayrılmıştır: Birinci kısımda hızlı bir yakınsamaya sahip orijinal bir algoritma geliştirilmiştir, test fonksiyonlarında sınanmıştır ve popüler algoritmalarla karşılaştırılmıştır. Bu bölümün ikinci kısımda ise: Geliştirilen algoritma CNN tabanlı bir sisteme uyarlanıp doğruluk ve zaman performansı açısından ele alınan *GS, ND, PA* parametrelerinin optimize edildiği bir çözüm uzayı elde edilmiştir (İnik ve ark., 2021).

#### 4.1.1.1. CNN tabanlı modelin hiper parametreleri

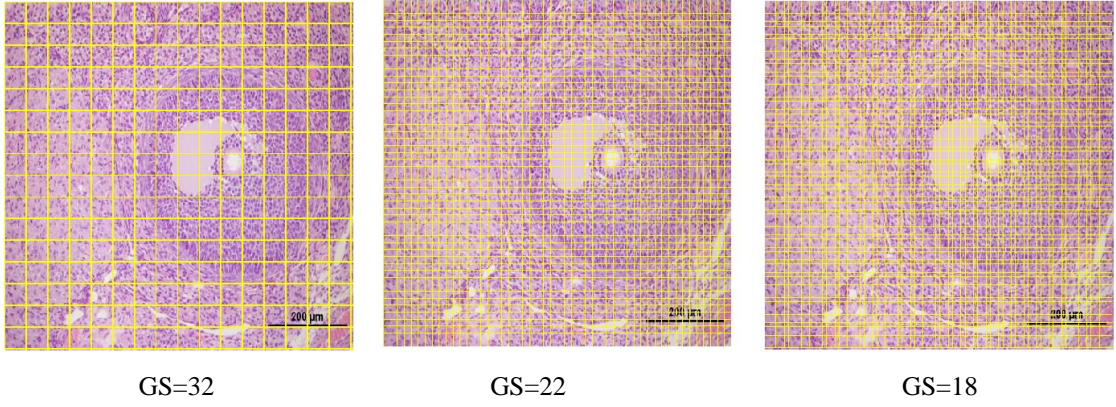
Şekil 4.1’de bu çalışmada optimize edilecek olan *GS*, *ND* ve *PA* parametrelerinin sonuca etkileri görülmektedir. CNN için giriş görüntüleri taranırken, tarama işleminde aranılan bölgeden sonra kaç piksel kayılacağını belirten adım değeri *GS* ile temsil edilir. *GS* değeri 1 ila 32 arasında değişen değerlerden seçilmelidir. Örnek olarak Şekil 4.2’de *GS* değerlerinin sırası ile 32, 22 ve 18 olarak alınması temsil edilmiştir. Bu işlemde özetle, görüntü girdilerinden alınan her bir yama sınıflandırılır.

Önerilen yöntemde, *PA* değerinden daha düşük olasılıkla sınıflandırılan yamaların yeniden değerlendirilmesi gerekmektedir. *PA* parametresi bu düşük olasılık değerini temsil etmektedir. Örneğin *PA* parametresi 0.70 olarak ayarlanırsa, 0.70’den küçük bir olasılıkla sınıflandırılmış her yama, parça yeniden değerlendirilecektir. Şekil 4.3’te *PA* parametresi 0.70 olarak ayarlandığında, 93 yama; aynı parametre, 0.80 olarak kabul ayarlandığında 143 yama, aynı parametre 0.85 olarak ayarlandığında ise 190 yamanın yeniden değerlendirilip sınıflandırılması gerekmektedir.

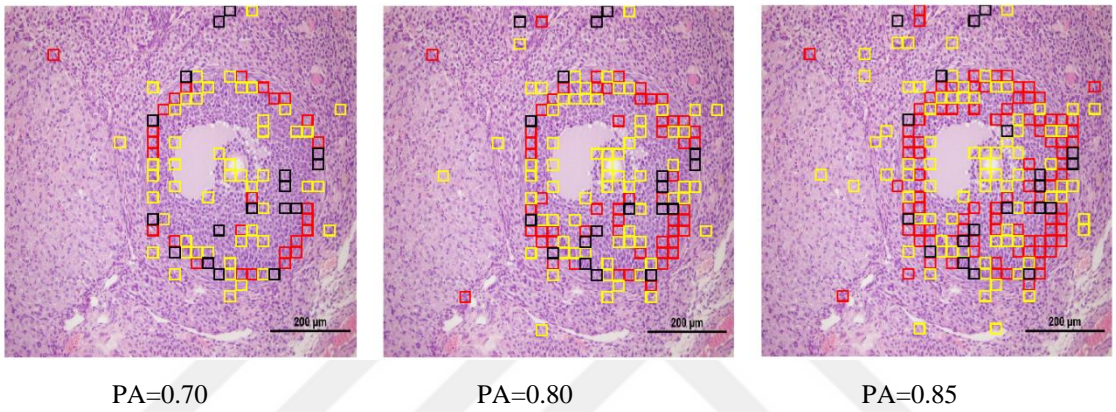
*ND* 1-32 aralığında değer almaktadır ve yeniden değerlendirilme işleminde etrafındaki komşuları ne oranda tarayacağını ifade etmektedir.



Şekil 4. 1. CNN temelli segmentasyon yönteminin ana çatısı (İnik ve ark., 2021)



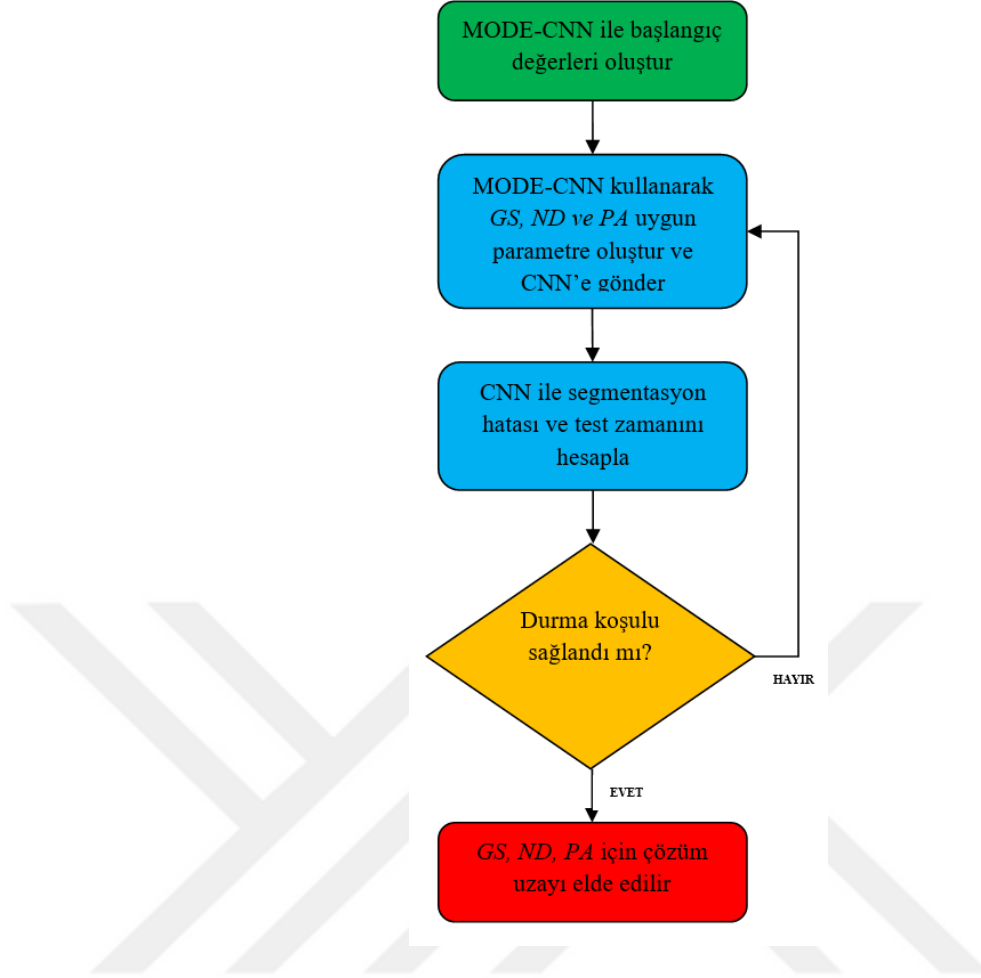
Şekil 4. 2. Farklı  $GS$  değerleri için alınan yamalar (İnik ve ark., 2021)



Şekil 4. 3. Farklı  $PA$  değerleri (İnik ve ark., 2021)

#### 4.1.1.2. $GS$ , $ND$ ve $PA$ 'nın optimizasyonu

Bu çalışmada F-MODE kullanmanın amacı, CNN tabanlı yöntemler için doğruluk ve test süresi değerlerini optimize eden, en iyi parametre değerlerini bulmaktır. Bu uygulamada F-MODE algoritması CNN mimarisine uyarlandığı için ismi MODE-CNN olarak kullanılmıştır. MODE-CNN algoritmasının, CNN mimarileri ile hibrit şekilde nasıl kullanıldığı Şekil 4.4'te şematize edilmiştir. Şekilden de anlaşılacağı üzere  $GS$ ,  $ND$  ve  $PA$  parametreleri için alt ve üst limitler dikkate alınarak rastgele değerler oluşturulur, bu değerlerle CNN mimarisi segmentasyon hatası ve test süresini hesaplar. Bu veriler dikkate alınarak MODE-CNN tarafından daha uygun parametre değerleri üretilir ve tekrar CNN mimarisi yeni parametrelerle yeni hesaplamalar yapar. Sonlandırma koşulu sağlanana kadar bu döngü böyle devam eder. Sonuçta uygun test süresi ve segmentasyon hatası değerlerini üretecek parametrelerden oluşan bir çözüm uzayı elde edilmiştir.



Şekil 4. 4. MODE-CNN algoritmasının CNN tabanlı sistemin hiper parametre optimizasyonunda uygulanması

Segmentasyon hatası ya da segmentasyon kaybı olarak ifade edilen değer eşitlik 4.1’de gösterilmiştir. Segmentasyon işlemi için hesaplanan toplam süre, ikinci amaç olan test süresidir.

$$Segmentation Loss = \left( 1 - \frac{1}{n} \sum_{i=1}^n \frac{1}{m} \sum_{j=1}^m (IoU)_j \right) \quad (4.1)$$

Bu formülde  $m$ , görüntüdeki nesnelerin sayısıdır. Aynı formüldeki  $n$  ise toplam görüntü sayısıdır. Denklem 4.2 yardımıyla hesaplanan  $IoU$  değeri ise, doğru değer (*ground truth*) ile tahmin edilen nesnenin kesiştiği piksel sayısının, toplam piksel sayısına oranını ifade etmektedir.

$$IoU = \frac{TP}{TP + FP + FN} \quad (4.2)$$

Bu formülde *TP* doğru pozitifleri (*true positives*) temsil eder, yani tahmin edilen nesnenin doğru şekilde sınıflandırılmış piksellerinin sayısıdır. *FP* ise yanlış pozitifleri (*false positives*) gösterir; bunun anlamı, nesneye ait olmayan piksellerin nesneye aitmiş gibi sınıflandırılmasını ifade eder. *FN* ise yanlış negatifler (*false negatives*) anlamına gelir, yani gerçekte nesneye ait olan piksellerin yanlışlıkla nesneye ait değilmiş gibi sınıflandırılmasını temsil etmektedir.

#### 4.1.2. Çalışmanın literatüre katkıları

Bu çalışmanın literatüre katkılarından önemli olan bazıları şöyle sıralanmıştır.

1. CNN tabanlı modellerinin hiper parametrelerini optimize etmek amacıyla, hızlı yakınsamaya sahip F-MODE adında bir algoritma geliştirilmiştir.
2. MOEA'larda çözümlerin kalitelerini belirlemek zordur, yani birinci amaç için iyi kabul edilen bir sonuç ikinci amaç için kötü olabilir. Bu sebepten dolayı 4.1.3.2'de ayrıntısı verilen bir kalite belirleme yöntemi önerilmiş ve bu çalışmada kullanılmıştır.
3. Kalitesi belirlenen çözümler rulet tekeri yöntemi ile ebeveyn seçiminde kullanılmış ve çaprazlamalar bu seçimler doğrultusunda yapılmıştır. Böylece kaliteli çözümler çaprazlamada daha çok kullanılmıştır ve nispeten hızlı yakınsama sağlanmıştır.
4. Bu çalışmada yapılan literatür taraması kapsamında bilindiği kadarıyla literatürde ilk defa olarak CNN tabanlı yöntemde kullanılan *GS*, *ND* ve *PA* parametreleri, test süresi ve segmentasyon hatasına dayalı olarak, çok amaçlı bir problem olarak, optimize edilmiştir.

#### 4.1.3. F-MODE algoritması

Optimizasyon alanında araştırmacılar yeni bir algoritma geliştirirken genellikle problemin amaç fonksiyonunu en az 25,000 defa hesaplarlar başka bir ifade ile değerlendirirler (*evaluation*). Geliştirilen algoritma maksFes kadar değerlendirildikten sonra ancak diğer iyi algoritmalarla performans açısından karşılaştırılabilir hale gelir. Giriş bölümü ve problemin tanımı bölümünde bahsedildiği gibi CNN mimarilerinin optimizasyonunda en büyük problemlerinden biri zaman maliyetidir. CNN mimarileri hiper parametre optimizasyonu kapsamında bir örnek verilecek olursa, 1 adet fotoğraftan belirli bir nesne tespiti için, bu çalışmadaki edilen tecrübelerle göre, *GPU* tabanlı



işlemciye sahip bir bilgisayarla bile 55 saniyelere çıkan bir zaman maliyeti oluşmuştur. 1000 fotoğraf kullanıldığı düşünüldüğünde 55,000 saniye zaman maliyeti oluşmaktadır.

Bunun yanında meta-sezgisel optimizasyon algoritmaları gibi 25,000 maksFes ile algoritma çalıştırılrsa 5 fotoğraf kullanılarak hiper parametre optimizasyonu yapılsa  $5 \times 55 \times 25,000 = 6,875,000$  saniye gibi bir zaman maliyeti ortaya çıkmaktadır. Bu süre hiçbir algoritma ya da çalışma için verimli veya kabul edilebilir değildir. Bu nedenle daha düşük maksFes’lerde hızlı şekilde yakınsama yeteneğine sahip yeni algoritmalara ihtiyaç vardır. Bu problemin getirdiği motivasyonun ışığında, bu çalışma kapsamında maksFes 5,000 seviyesine kadar azaltılarak etkili bir algoritma tasarlamak amaçlanmıştır. Dolayısıyla bu çalışmanın amacı 5,000 maksFes’te muadillerine göre daha iyi bir performans sergileyecek bir algoritma geliştirmektir.

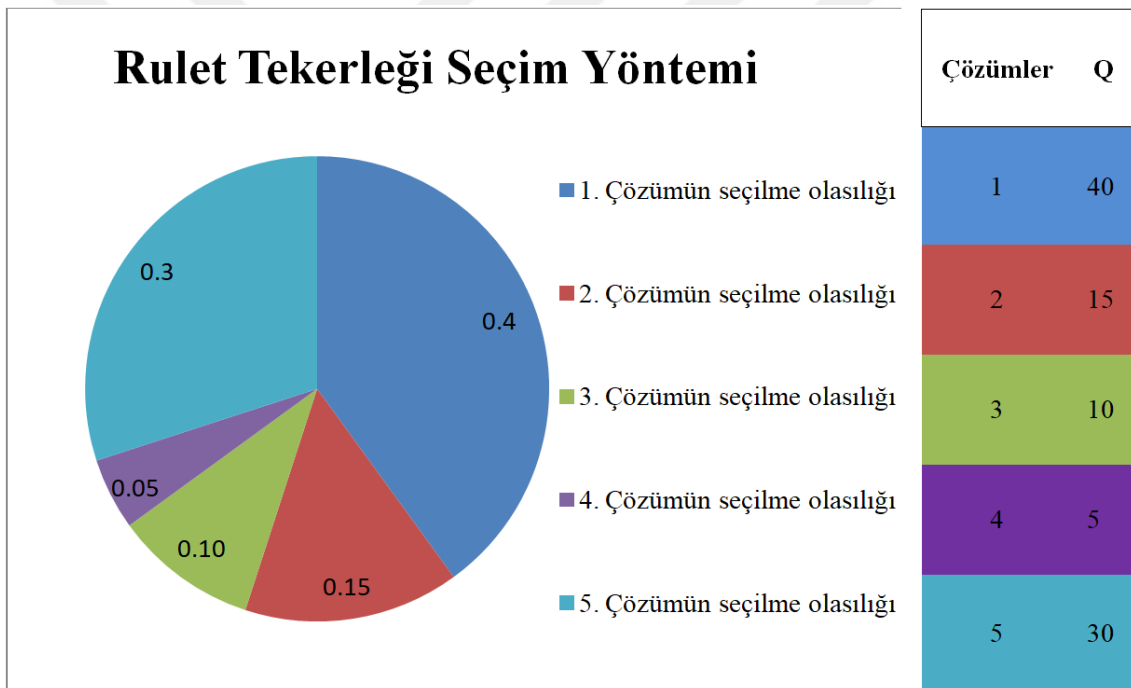
Bu amacın gerçekleştirilmesi için seçkinci, elitist bir yaklaşım gerektiğine dair hipotezler geliştirilmiştir. Bu bağlamda literatür incelendiğinde çaprazlama yapmak için popülasyonun kaliteli bireylerinden ebeveyn seçiminde rulet tekeri seçim yöntemi ya da mekanizması kullanıldığı görülmüştür. Her ne kadar rulet tekeri seçim yöntemi literatürde geçse de genellikle tek amaçlı optimizasyon algoritmalarında kullanıldığı, yapılan literatür araştırmasına göre maksFes sayısını azaltmak için daha önce MOEA’larda kullanılmadığı tespit edilmiştir. Rulet tekeri seçim yöntemi kullanmak için çözümlerin kalite değerlerini sembolize edecek göstergelere ihtiyaç vardır. Pareto tabanlı algoritmalarda ise çözümlerin doğrudan kalitelerini gösteren bir gösterge yoktur; çünkü her bir çözüm için iki amaçlı bir problemse iki sonuç, üç amaçlı bir problemse üç sonuç vardır. Bundan dolayı hızlı-bastırılmayan sıralama operatörünün sonucu olan cephe numarası (*Front Number, FNum*) ve kalabalıklık mesafesi (*Crowding Distance, CD*) operatörünün sonucu kullanılarak yeni bir kalite hesaplama modeli bu çalışma kapsamında önerilmiştir (İnik ve ark., 2021).

Önerilen F-MODE birçok operatörden oluşmaktadır. Bunlar: Bastırılmayan sıralama, kalabalıklık mesafesi, seçim, çaprazlama ve mutasyon operatörleridir. Bu bölümde F-MODE’un gelişimine katkı sağlayan seçim ve kalite başlıkları ayrıntıları ile açıklanmıştır.

#### 4.1.3.1. Seçim

MODE’de çaprazlama için ebeveyn seçilirken tüm popülasyondan çözümler rastgele seçilir. DE çaprazlama yöntemi kullanıldığından her bir aday çözüm üretimi için

üç adet ebeveyn seçimi yapılması gereklidir. DE çaprazlama yöntemine göre özetle iki çözümün farkları alınır sabit bir değer olan  $F$  ile çarpılır ve üçüncü bireye eklenerek aday birey oluşturulur. F-MODE 'de rulet tekeri seçim yöntemi ise; seçilecek üç ebeveynden, sadece biri için rulet tekeri seçimi kullanılmıştır. Tek bir ebeveynin seçiminde bile %100'lük rulet tekeri kullanımı, yerel aramaya takılma problemine sebep olabileceği için sadece aday çözümlerin %10'u oranında bu seçim yöntemi kullanılmıştır. Şekil 4.5'te rulet tekerliği seçim yöntemi temsilen gösterilmiştir. Buna göre 1 numaralı çözümün kalitesi 40'dır ve seçilme ihtimalide %40 olarak belirlenmiştir. Rulet tekerliği seçiminin başarılı olmasının bir sebebi de her çözümün seçilme ihtimalinin olmasından dolayı çeşitlik oranının korunması ve aynı zamanda kaliteli çözümlerin daha fazla seçilmesiyle hızlı yakınsama sağlamasıdır.



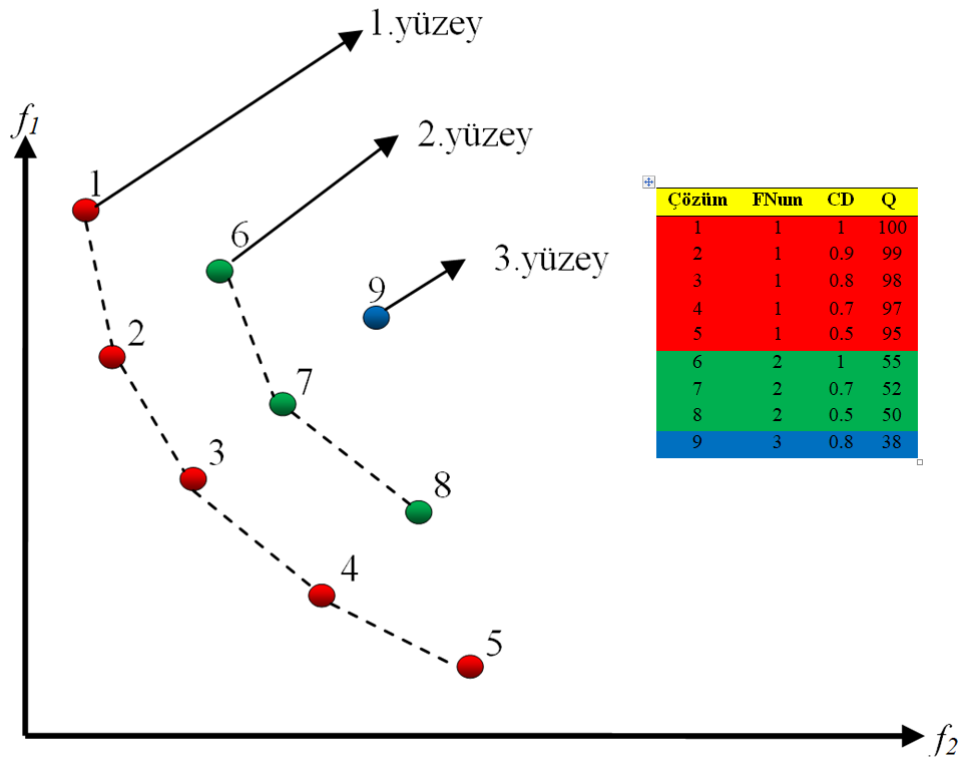
Şekil 4. 5. Temsili rulet tekerliği seçim yöntemi

#### 4.1.3.2. Kalite

Pareto tabanlı algoritmalarda çözümlerin kalite göstergesi olarak her ne kadar  $FNum$  ve  $CD$  değerleri hesaplanıyorsa da rulet tekerliği seçim mekanizması için tek bir kalite göstergesi olmak zorundadır. Bundan dolayı Denklem 4.3'te bir matematik eşitlik önerilmiştir.

$$Q = \frac{90}{FNum} + 10 * CD \quad (4.3)$$

Burada  $Q$  her çözüm için kalite değerini ifade etmektedir.  $FNum$  cephe numarasını ve  $CD$  ise kalabalıklık mesafe değerini temsil etmektedir.  $FNum$ 'da kalite, artan sıralama yaklaşımına göre organize edilmiştir yani herhangi bir çözümün cephe değeri  $FNum=1$  ise o çözüm en kaliteli sınıftadır, başka bir deyişle Pareto cephesindedir. Bu durumda Pareto cephesini oluşturan çözümlerin birden fazla olması problemi bulunmaktadır. Bunun için eşit olan  $FNum$ 'a sahip çözümlerde hangisinin daha kaliteli olduğuna bakmada  $CD$  değeri kullanılır.  $CD$  değeri ise azalan sıralamaya göre tasarlanmıştır. Örneğin  $CD=100$ ,  $CD=99$ 'dan daha kaliteli bir çözümdür.  $Fnum$  ve  $CD$  değerleri Denklem 4.3'te doğrudan kullanılabilir. Şekil 4.6'da kalite hesabı için temsili bir örnek verilmiştir (İnik ve ark., 2021).

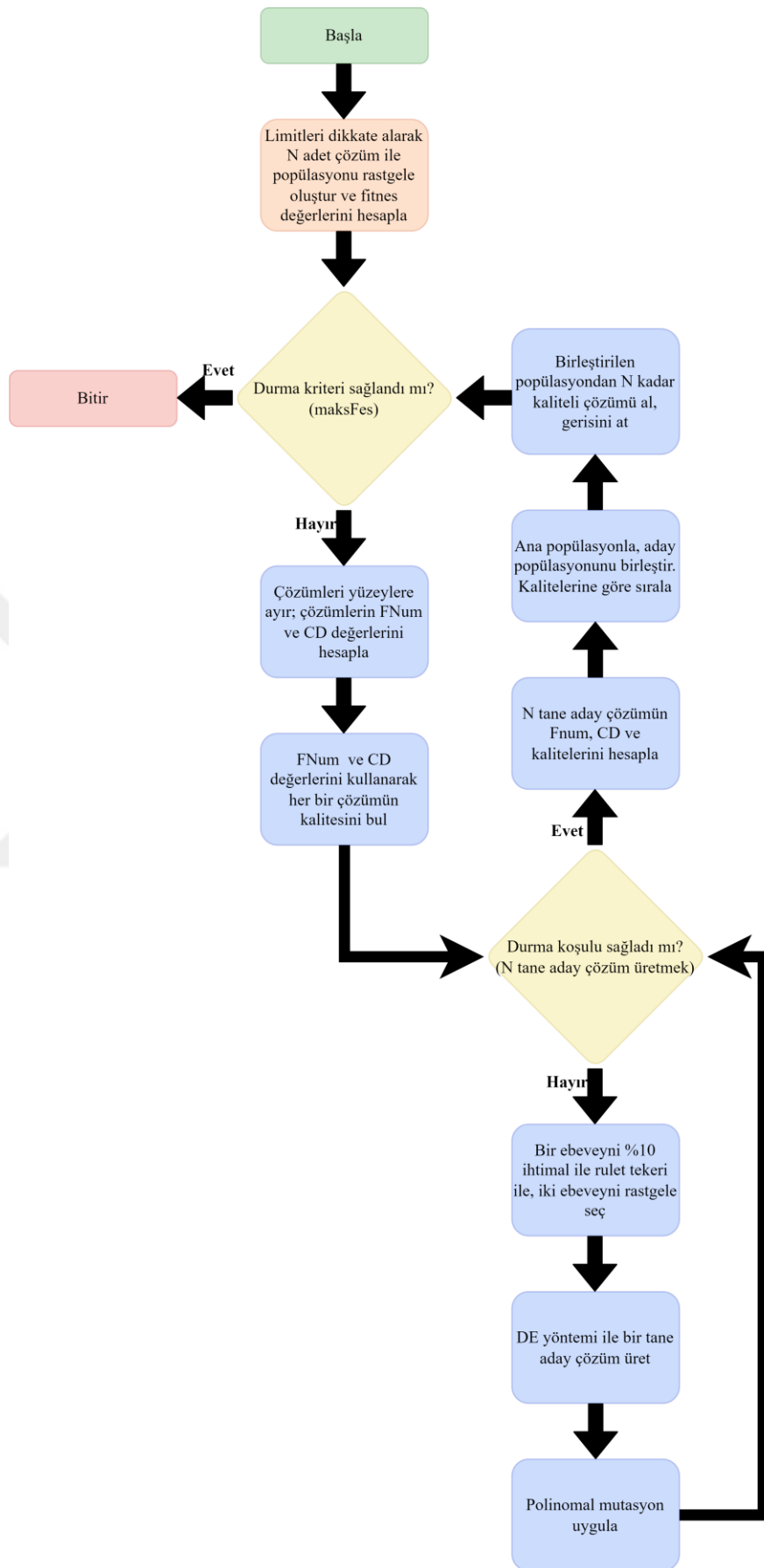


Şekil 4. 6. Çözümlerin temsili kalite skorları

Şekil 4.6'ya göre Pareto cephesi kırmızı ile iki numaralı cephe yeşil ile üç numaralı cephe mavi ile gösterilmiştir. Şekilden de görüldüğü gibi  $FNum$ ' de değer arttıkça kalite düşerken,  $CD$ ' de değer azaldıkça kalite de azalmaktadır.

Akış diyagramı Şekil 4.7'de görülen F-MODE'nin adımları şöyledir:

1. Ana popülasyonu  $P_0$  temsil eden  $N$  adet çözüm, üst ve alt sınırlar dikkate alınarak rastgele oluşturulur.
2.  $P_0$ 'a ait tüm çözümler için amaç fonksiyonu hesaplanır.
3. maksFes sayısı kadar ana döngü başlatılır.
  - a) Ana popülasyondaki her çözüm, "hızlı-bastırılmayan sıralama" yöntemi kullanılarak kalitelerine göre yüzeylere ayrılır ( $FNum$ ). Aynı cephedeki çözümlerin kalitesini ayırmak için ise kalabalıklık mesafesi operatörü kullanılır ( $CD$ ).
  - b)  $FNum$  ve  $CD$  kullanılarak, popülasyondaki her çözümün kalitesi göre Denklem 4.3'e göre hesaplanır.
  - c) Popülasyon sayısı kadar bir döngü daha başlatılır.
    - I. Ana popülasyondan üç tane ebeveyn seçilir. Bunlardan ilkinin %10 rulet tekerliği seçimi ile seçilir. Kalanı ve diğer iki ebeveyn rastgele olarak popülasyondan seçilir. DE çaprazlaması ile yeni aday birey üretilir.
    - II. 0.001 oranında polinomal mutasyon uygulanır.
    - III.  $N$  tane aday çözüm üretilerek  $A_0$  popülasyonu oluşturuldu mu? Hayırsa, 3.c.I. adıma git. Evetse, devam et.
  - d) Ana popülasyon  $P_0$  ve aday popülasyonu  $A_0$  birleştirilir. Bu yeni büyük popülasyon  $B_0=P_0+A_0$  "hızlı-bastırılmayan sıralama operatörü" ile yüzeylere ayrılır.
  - e) Büyük popülasyondaki çözümler,  $FNum$  ve  $CD$  değerlerine göre iyiden kötüye doğru sıralanır.
  - f) En iyi kaliteye sahip  $N$  çözüm, yeni ana popülasyon olur.  $B_0$ 'ın geri kalan kısmı atılır.
4. Sonlandırma kriteri ( $maksFes$ ) karşılanmadıysa 3a. adımına gidilir. Sonlandırma kriteri karşılandığında algoritma durur ve sonuçta bastırılmayan çözümlerden oluşan Pareto optimal çözüm kümesi elde edilir.



Şekil 4. 7. F-MODE akış diyagramı

### 4.1.3.3. Deneyler

Bu bölümdeki deneyler iki aşamada gerçekleşmiştir. Birincisi F-MODE'un performansı bu alanda iyi bilinen 4 algoritma ile 16 test fonksiyonu üzerinde yapılan deneylerle ve karşılaştırmalarla gösterilmiştir. İlgili deneyler *Intel® Core™ i7-3630QM CPU@2.40GHz* işlemciye sahip ve *12GB RAM*'i bulunan bir bilgisayarda gerçekleştirilmiştir. Bunun yanı sıra F-MODE algoritması *JMETAL 4.5* çok amaçlı yazılım paketi üzerine eklenmiştir ve performans karşılaştırmaları ilgili paket dâhilinde yapılmıştır (Durillo ve Nebro, 2011).

İkinci aşama ise F-MODE uyarlanarak ile hiper parametre optimizasyonu uygulaması gerçekleştirilmiştir. Bu uygulama, derin öğrenme uygulaması olduğu için; *Intel® Core™ i9-7900X 3.30GHz×20* işlemciye sahip, *64GB RAM'e* ve iki adet *GeForce RTX2080Ti* ekran kartına sahip bilgisayarda yapılmıştır. Yazılım aracı olarak, *Matlab R2020a 64bit (Win64)*'nin derin öğrenme kütüphanesi kullanılmıştır. Kodlamada gerçek değerli kodlama yapılmıştır. Önemli parametreler olarak *CR* ve çaprazlama sabiti olan *F* değeri çaprazlama bölümünde açıldığı gibi 0.5 olarak kullanılmıştır. Bütün algoritmaların popülasyon sayısı test fonksiyonları üzerinde çalışırken 100 olarak belirlenmiştir.

F-MODE eşit koşullardaki NSGA-II (Deb ve ark., 2002), MOAAA (Babalik ve ark., 2018), OMOPSO (Sierra ve Coello, 2005) ve klasik MODE (Xue ve ark., 2003) algoritması ile karşılaştırılmıştır. Bu algoritmalar: NSGA-II literatürde en çok geçen algoritma, MOAAA son zamanlarda önerilmiş ve performans açısından çoğu algoritmadan önde olan bir algoritma, OMOPSO literatürdeki en hızlı yakınsamaya sahip çok amaçlı algoritmalarından biri ve klasik MODE algoritması da F-MODE algoritması ile ortak yönleri olduğundan karşılaştırma algoritmaları olarak kullanılmıştır.

Algoritmaların performans kıyaslamalarında HV (Riquelme ve ark., 2015; Karakoyun ve ark., 2020) ve IGD metrikleri kullanılmıştır (Durillo ve Nebro, 2011; Karakoyun ve ark., 2020).

Çalışmanın bu kısmında kullanılan test fonksiyonları ise İki amaçlı *ZDT* problem seti, üç amaçlı *DTLZ* problem seti, *WFG* problem seti ve bunların yanında *Fonseca*, *Kursawe*, *Schaffer* fonksiyonlarıdır.

Çizelge 4.1'de F-MODE, HV metriğine göre 100 bağımsız çalışmadaki, 5,000 maksFes üretilerek elde edilen sonuçların ortalama ve standart sapma değerleri

verilmiştir. Gri olan hücreler HV metriğine göre daha başarılı olan sonuçları göstermektedir.

**Çizelge 4. 1.** HV metriğine göre algoritmaların performansları

	NSGAIH		MOAAA		MODE		OMOPSO		F-MODE	
	Ort.	S. s.	Ort.	S. s.	Ort.	S. s.	Ort.	S. s.	Ort.	S. s.
<b>ZDT1</b>	4.53E-01	3.30E-02	3.27E-01	6.20E-02	1.05E-01	3.90E-02	6.05E-01	3.60E-02	6.34E-01	4.30E-03
<b>ZDT2</b>	5.60E-02	3.50E-02	1.56E-02	2.10E-02	0.00E+00	0.00E+00	3.06E-01	1.80E-02	2.98E-01	5.30E-03
<b>ZDT3</b>	3.79E-01	2.30E-02	3.00E-01	3.60E-02	1.15E-01	3.50E-02	3.94E-01	6.60E-02	4.84E-01	5.70E-03
<b>DTLZ1</b>	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.12E-01	1.80E-01
<b>DTLZ2</b>	3.80E-01	8.40E-03	3.64E-01	9.40E-03	3.46E-01	7.80E-03	3.43E-01	9.70E-03	3.82E-01	6.80E-03
<b>DTLZ3</b>	1.04E-04	7.30E-04	2.74E-04	2.70E-03	0.00E+00	0.00E+00	9.71E-03	8.40E-02	2.62E-02	5.90E-02
<b>DTLZ4</b>	3.65E-01	6.00E-02	3.72E-01	1.10E-02	3.58E-01	1.30E-02	2.98E-01	3.80E-02	3.88E-01	6.80E-03
<b>DTLZ5</b>	9.03E-02	6.60E-04	8.38E-02	2.40E-03	8.39E-02	1.60E-03	8.95E-02	1.20E-03	9.01E-02	5.70E-04
<b>DTLZ6</b>	0.00E+00	0.00E+00	9.19E-02	5.90E-04	1.93E-02	2.90E-02	2.05E-02	3.70E-02	9.26E-02	1.80E-04
<b>DTLZ7</b>	9.47E-02	2.20E-02	1.47E-01	3.80E-02	9.38E-03	9.90E-03	1.99E-01	5.90E-02	2.73E-01	1.00E-02
<b>WFG1</b>	1.01E-01	5.30E-02	2.44E-01	4.60E-02	1.78E-01	4.10E-03	1.56E-01	6.90E-03	3.28E-01	4.90E-02
<b>WFG2</b>	5.50E-01	2.50E-03	5.46E-01	2.50E-03	5.41E-01	2.80E-03	5.47E-01	2.30E-03	5.53E-01	8.60E-04
<b>WFG3</b>	4.87E-01	1.70E-03	4.87E-01	1.40E-03	4.87E-01	7.80E-04	4.88E-01	9.20E-04	4.91E-01	5.00E-04
<b>Fonseca</b>	3.08E-01	4.30E-04	3.08E-01	3.90E-04	3.09E-01	3.20E-04	3.12E-01	1.70E-04	3.10E-01	2.70E-04
<b>Kursawe</b>	4.00E-01	4.60E-04	3.97E-01	1.40E-03	3.96E-01	1.10E-03	3.92E-01	1.70E-03	4.00E-01	5.00E-04
<b>Schaffer</b>	1.55E-01	2.30E-01	1.32E-01	2.20E-01	1.64E-01	2.50E-01	8.30E-01	3.30E-05	8.29E-01	3.00E-04

HV metriği, daha önce bahsedildiği gibi literatürde en çok kullanılan metriklerden biridir. Bu metrikte, diğer metriklerin tersine büyük skorlar küçük skorlara göre daha iyi sonuçları temsil ederler çünkü genel anlamda 1'e en yakın skor en iyi skordur, dolayısıyla skorun daha yüksek olması gerçek sonuçlara (olabilecek en doğru sonuçlara) daha yakın olduğu anlamına gelir. İlgili çizelgeye göre geliştirilen algoritmanın *ZDT2*, *DTLZ5*, *Fonseca* ve *Schaffer* dışındaki bütün test fonksiyonlarında diğer algoritmalarından üstün olduğu görülmektedir. Performans olarak geri kaldığı dört fonksiyonda da F-MODE'un kabul edilebilir değerler ürettiği görülmektedir. Hızlı yakınsaması ile ön plana çıkan OMOPSO beklendiği gibi performans açısından ikinci sırayı almıştır. Bu deneylerden çıkan sonuç, diğer algoritmaların kötü olduğu değildir. Hatırlatmak gerekir ki meta-sezgisel optimizasyon alanında az bir deneme değeri olarak görülen 5,000 maksFes'te bu sonuçlar alınmaktadır. Bu sonuçlar, sadece yakınsama hakkındaki yapılan hipotezin beklendiği gibi doğrulandığını göstermektedir.

IGD metriği, algoritmanın performansını doğruluk, yakınsama ve çeşitlilik açısından ölçer. Testlerden elde edilen IGD değerleri Çizelge 4.2'de sunulmuştur.

Çizelge 4. 2. IGD metriğine göre algoritmaların performansları

	NSGAIH		MOAAA		MODE		OMOPSO		F-MODE	
	Ort.	S.s.	Ort.	S.s.	Ort.	S.s.	Ort.	S.s.	Ort.	S.s.
<b>ZDT1</b>	8.57E-03	1.40E-03	1.39E-02	3.40E-03	2.98E-02	3.80E-03	2.19E-03	1.40E-03	1.08E-03	1.60E-04
<b>ZDT2</b>	1.62E-02	7.50E-03	1.99E-02	6.50E-03	3.92E-02	5.40E-03	7.06E-04	4.90E-04	9.00E-04	1.60E-04
<b>ZDT3</b>	4.52E-03	1.20E-03	8.85E-03	2.30E-03	2.29E-02	3.30E-03	4.96E-03	3.30E-03	7.81E-04	1.30E-04
<b>DTLZ1</b>	1.05E+00	3.30E-01	1.25E+00	5.60E-01	9.00E-01	3.30E-01	3.20E+00	9.40E-01	7.75E-02	4.80E-02
<b>DTLZ2</b>	1.02E-03	5.50E-05	1.05E-03	4.80E-05	1.08E-03	4.00E-05	1.06E-03	4.60E-05	9.46E-04	3.50E-05
<b>DTLZ3</b>	3.52E-01	9.90E-02	5.18E-01	1.00E-01	5.16E-01	8.80E-02	8.14E-01	2.00E-01	2.11E-01	4.90E-02
<b>DTLZ4</b>	1.80E-03	2.40E-03	1.08E-03	5.20E-05	1.71E-03	4.60E-04	3.98E-03	1.00E-03	9.90E-04	4.00E-05
<b>DTLZ5</b>	2.17E-04	2.00E-05	4.33E-04	7.70E-05	4.12E-04	5.10E-05	2.32E-04	4.50E-05	2.17E-04	1.80E-05
<b>DTLZ6</b>	3.66E-02	1.70E-03	6.68E-05	1.00E-05	5.13E-03	3.10E-03	6.91E-03	4.60E-03	5.65E-05	3.20E-06
<b>DTLZ7</b>	4.73E-03	8.80E-04	3.47E-03	8.70E-04	1.29E-02	2.60E-03	3.02E-03	1.30E-03	1.56E-03	1.00E-03
<b>WFG1</b>	5.11E-02	6.60E-03	3.06E-02	6.20E-03	4.03E-02	9.10E-04	4.46E-02	6.60E-04	1.95E-02	5.50E-03
<b>WFG2</b>	2.61E-03	2.50E-03	4.82E-04	8.80E-05	6.58E-04	1.10E-04	4.61E-04	8.70E-05	2.55E-04	5.40E-05
<b>WFG3</b>	1.59E-04	1.70E-05	1.60E-04	1.50E-05	1.66E-04	9.40E-06	1.45E-04	1.10E-05	1.36E-04	8.60E-06
<b>Fonseca</b>	9.19E-05	3.50E-06	8.69E-05	3.00E-06	8.09E-05	2.90E-06	6.16E-05	5.60E-07	8.07E-05	3.20E-06
<b>Kursawe</b>	1.06E-04	5.10E-06	1.35E-04	1.50E-05	1.41E-04	1.50E-05	1.93E-04	2.40E-05	1.00E-04	6.10E-06
<b>Schaffer</b>	2.14E-02	3.00E-02	2.98E-02	5.00E-02	2.86E-02	4.60E-02	3.41E-05	3.00E-07	4.77E-05	3.70E-06

IGD metriği kapsamında F-MODE algoritması 16 fonksiyondan 13'ünde daha iyi performans göstermiştir. Çizelge 4.2'de de görülebileceği gibi, F-MODE, IGD metriği çerçevesinde düşünüldüğünde de başarılı sayılabilmektedir. Önerilen algoritma en iyi performansı vermediği sonuçlarda da kabul edilebilir sonuçlar üretebildiği gözlenmiştir.

Çizelge 4.3 ve Çizelge 4.4'e göre: F-MODE algoritması, diğer algoritmalarla karşılaştırıldığında anlamsal olarak farklı sonuçlar ürettiği, rakip algoritmalarla Wilcoxon sıralı toplam testi sonucuna göre de anlaşılmaktadır. Görüldüğü üzere  $p$  değerinin 0.05'ten daha küçük olan değerler "+" ile belirtilmiştir. Bu istatistiksel veriler, F-MODE algoritmasının sonuçlarının tesadüfe dayalı olarak daha iyi olmadığı aksine her bağımsız çalışmada anlamlı sonuçlar ürettiğini göstermiştir. Algoritmanın anlamlı değerler üretmediği durumlar, yani  $p$  değerinin 0.05'ten daha büyük olduğu durumlar "-" işareti ile belirtilmiştir.



Çizelge 4. 3. HV Metriğine göre Wilcoxon sıralı toplam testi sonuçları

F-MODE	HV METRİĞİNE GÖRE SIRA TOPLAMI TESTİ								
	NSGA-II		OMOPSO		MODE		MOAAA		
	vs	p-değeri	işaret	p-değeri	işaret	p-değeri	işaret	p-değeri	işaret
<b>ZDT1</b>		2.56E-34	+	1.50E-13	+	2.56E-34	+	2.56E-34	+
<b>ZDT2</b>		2.54E-34	+	5.14E-10	+	5.64E-39	+	2.21E-34	+
<b>ZDT3</b>		2.56E-34	+	1.48E-30	+	2.56E-34	+	2.56E-34	+
<b>DTLZ1</b>		9.58E-16	+	9.58E-16	+	9.58E-16	+	9.58E-16	+
<b>DTLZ2</b>		1.34E-01	-	2.64E-34	+	2.89E-34	+	5.64E-27	+
<b>DTLZ3</b>		1.06E-07	+	8.17E-07	+	7.31E-09	+	3.13E-08	+
<b>DTLZ4</b>		1.86E-06	+	2.56E-34	+	9.56E-34	+	4.85E-23	+
<b>DTLZ5</b>		8.13E-02	+	2.70E-06	+	2.56E-34	+	2.72E-34	+
<b>DTLZ6</b>		5.64E-39	+	1.80E-24	+	5.48E-34	+	1.40E-30	+
<b>DTLZ7</b>		2.56E-34	+	7.70E-23	+	2.56E-34	+	1.33E-33	+
<b>WFG1</b>		4.40E-34	+	2.72E-34	+	2.56E-34	+	1.19E-20	+
<b>WFG2</b>		6.87E-20	+	6.48E-34	+	2.56E-34	+	2.56E-34	+
<b>WFG3</b>		2.00E-29	+	5.42E-34	+	2.56E-34	+	3.67E-34	+
<b>Fonseca</b>		2.56E-34	+	2.56E-34	+	1.18E-13	+	2.72E-34	+
<b>Kursawe</b>		1.50E-05	+	2.56E-34	+	2.56E-34	+	4.02E-34	+
<b>Schaffer</b>		2.55E-35	+	2.56E-34	+	2.01E-35	+	1.19E-35	+

Çizelge 4. 4. IGD Metriğine göre Wilcoxon sıralı toplam testi sonuçları

F-MODE	IGD METRİĞİNE GÖRE SIRA TOPLAMI TESTİ								
	NSGA-II		OMOPSO		MODE		MOAAA		
	vs	p-değeri	işaret	p-değeri	işaret	p-değeri	işaret	p-değeri	işaret
<b>ZDT1</b>		2.56E-34	+	1.21E-15	+	2.56E-34	+	2.56E-34	+
<b>ZDT2</b>		2.56E-34	+	3.00E-09	+	2.56E-34	+	2.56E-34	+
<b>ZDT3</b>		2.56E-34	+	5.81E-33	+	2.56E-34	+	2.56E-34	+
<b>DTLZ1</b>		2.56E-34	+	2.56E-34	+	2.56E-34	+	2.72E-34	+
<b>DTLZ2</b>		5.66E-21	+	2.84E-31	+	4.59E-33	+	3.89E-29	+
<b>DTLZ3</b>		6.04E-23	+	1.24E-31	+	3.36E-34	+	1.28E-32	+
<b>DTLZ4</b>		1.86E-18	+	2.56E-34	+	3.36E-34	+	5.95E-27	+
<b>DTLZ5</b>		9.54E-01	-	6.61E-03	+	2.56E-34	+	2.64E-34	+
<b>DTLZ6</b>		2.56E-34	+	4.77E-24	+	3.96E-33	+	5.83E-26	+
<b>DTLZ7</b>		5.02E-33	+	3.89E-31	+	6.29E-34	+	5.02E-33	+
<b>WFG1</b>		3.07E-34	+	2.56E-34	+	2.56E-34	+	6.93E-24	+
<b>WFG2</b>		2.83E-06	+	2.26E-31	+	2.64E-34	+	2.17E-32	+
<b>WFG3</b>		6.82E-23	+	4.61E-10	+	4.63E-32	+	1.66E-26	+
<b>Fonseca</b>		9.58E-33	+	2.56E-34	+	4.61E-01	-	3.36E-25	+
<b>Kursawe</b>		2.86E-12	+	2.56E-34	+	3.46E-34	+	2.07E-33	+
<b>Schaffer</b>		2.56E-34	+	2.56E-34	+	2.56E-34	+	2.56E-34	+

İlgili istatistik verisine göre, HV metriğinde sadece *DTLZ2* probleminde NSGA-II ile benzer sonuçlar verdiği anlaşılmıştır. Bunun sebebi de her iki algoritmanın da Pareto tabanlı algoritmalar olmasından dolayıdır denilebilir. Hatta aynı tabana sahip olmalarına rağmen bu kadar farklı sonuçlar üretmesi F-MODE için başarı göstergesi sayılabilir.

IGD metriğine göre ise NSGA-II ile *DTLZ5* ve MODE algoritmasına göre *Fonseca* probleminde benzer sonuçlar üretilmiştir. Bu sonuçlar üretmesinin sebebi *Fonseca* ve *DTLZ5* problemlerini ilgili algoritmaların çok yüksek performansla çözmeleri

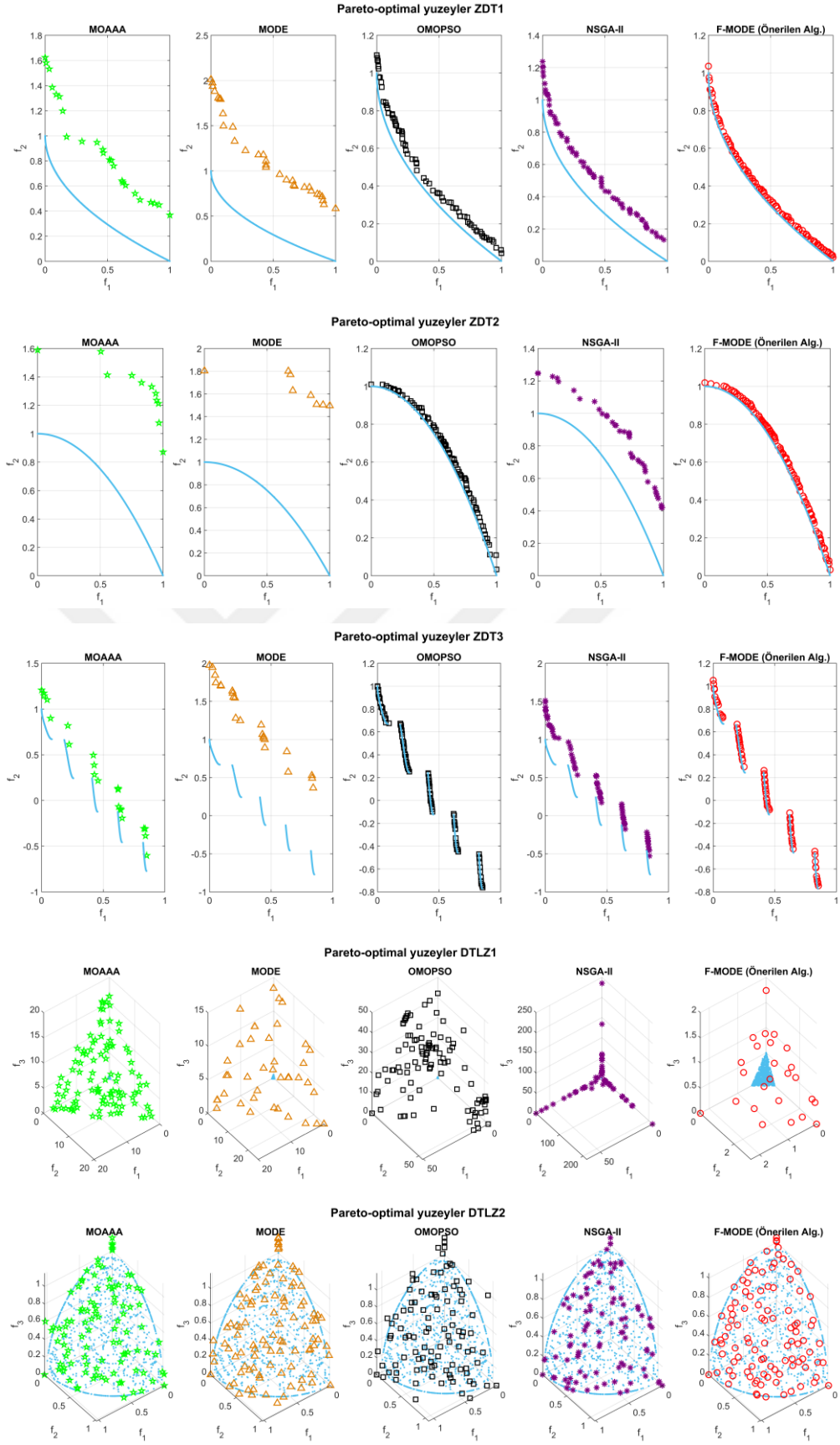
ve neredeyse gerçek Pareto-optimal yüzeyi yakalamış olmalarındandır. F-MODE, geri kalan diğer bütün problemlerde diğer algoritmalarından farklı sonuçlar üretilmiştir.

Şekil 4.8'den itibaren, F-MODE ve karşılaştırıldığı diğer algoritmaların, 16 test fonksiyonu için Pareto-optimal yüzey performansları gösterilmiştir. İlgili problemlerin gerçek yüzeyleri turkuaz renk ile belirtilmiştir. MOAAA'nın tahmin ettiği yüzeyler içi boş yeşil yıldız; MODE'nin tahmin ettiği yüzeyler sarı üçgen, OMOPSO'nun tahmini siyah kare, NSGA-II'nin tahmin ettiği yüzeyler ise mor yıldız; F-MODE tarafından tahmin edilen yüzeyler ise kırmızı halkalar ile temsil edilmektedir.

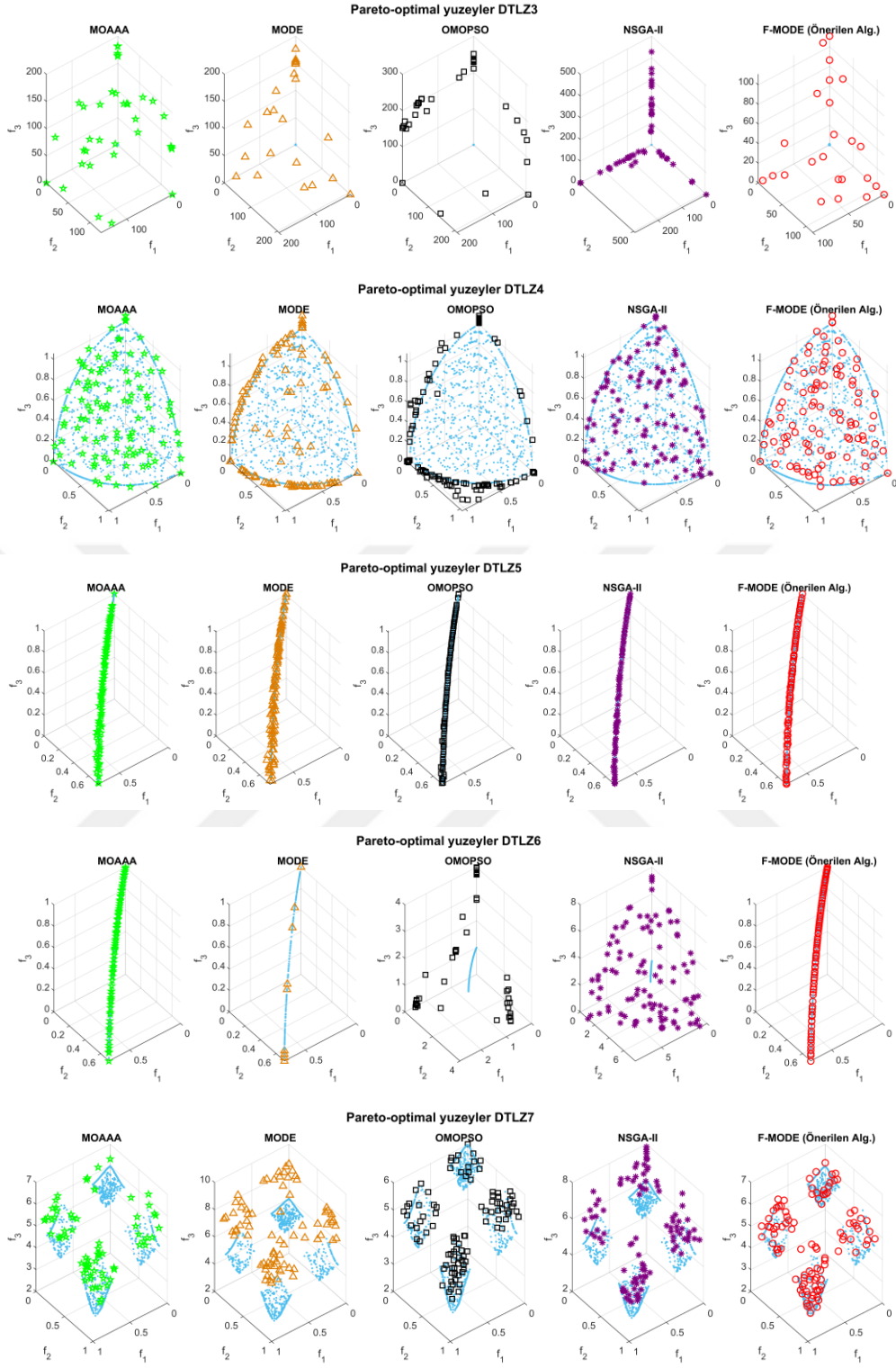
Şekil 4.8- Şekil 4.11'e göre ilk problem olan *ZDT1* için açıkça görüldüğü gibi en iyi performans F-MODE'e aittir. Bunun yanında OMOPSO ve NSGA-II kabul edilebilir bir Pareto-optimal yüzey bulmuşlardır. *ZDT2* probleminde ise en iyi yüzeyler F-MODE ve OMOPSO'ya aittir. Bu problemde diğer algoritmalar, özellikle MOAAA ve MODE kötü bir performans göstermişlerdir. *ZDT3* probleminde yine iyi ve kötü performans gösteren algoritmalar değişmemiştir. Genel anlamda 5,000 maksFes'te *ZDT* problemlerinde gerçek Pareto-optimal yüzeylere en çok yaklaşan algoritmalar F-MODE ve OMOPSO olmuştur. Bu iki algoritmada hızlı yakınsamaları ile ön plana çıkan algoritmalarlardır.

*DTLZ1* ve *DTLZ3* probleminde hiçbir algoritma iyi bir grafik çizememiştir. *DTLZ2* ve *DTLZ4* probleminde ise neredeyse bütün algoritmalar kabul edilir bir sonuç üretmişlerdir. *DTLZ5*'te bütün algoritmalar kusursuz bir performans gösterirlerken *DTLZ6*'da F-MODE ve MOAAA dışında diğer algoritmalar kötü bir Pareto-yüzey bulmuşlardır. *DTLZ7*'de MOAAA, OMOPSO ve F-MODE gerçek Pareto-optimal yüzeylere diğer algoritmalara göre daha iyi bir şekilde yakınsamışlardır.

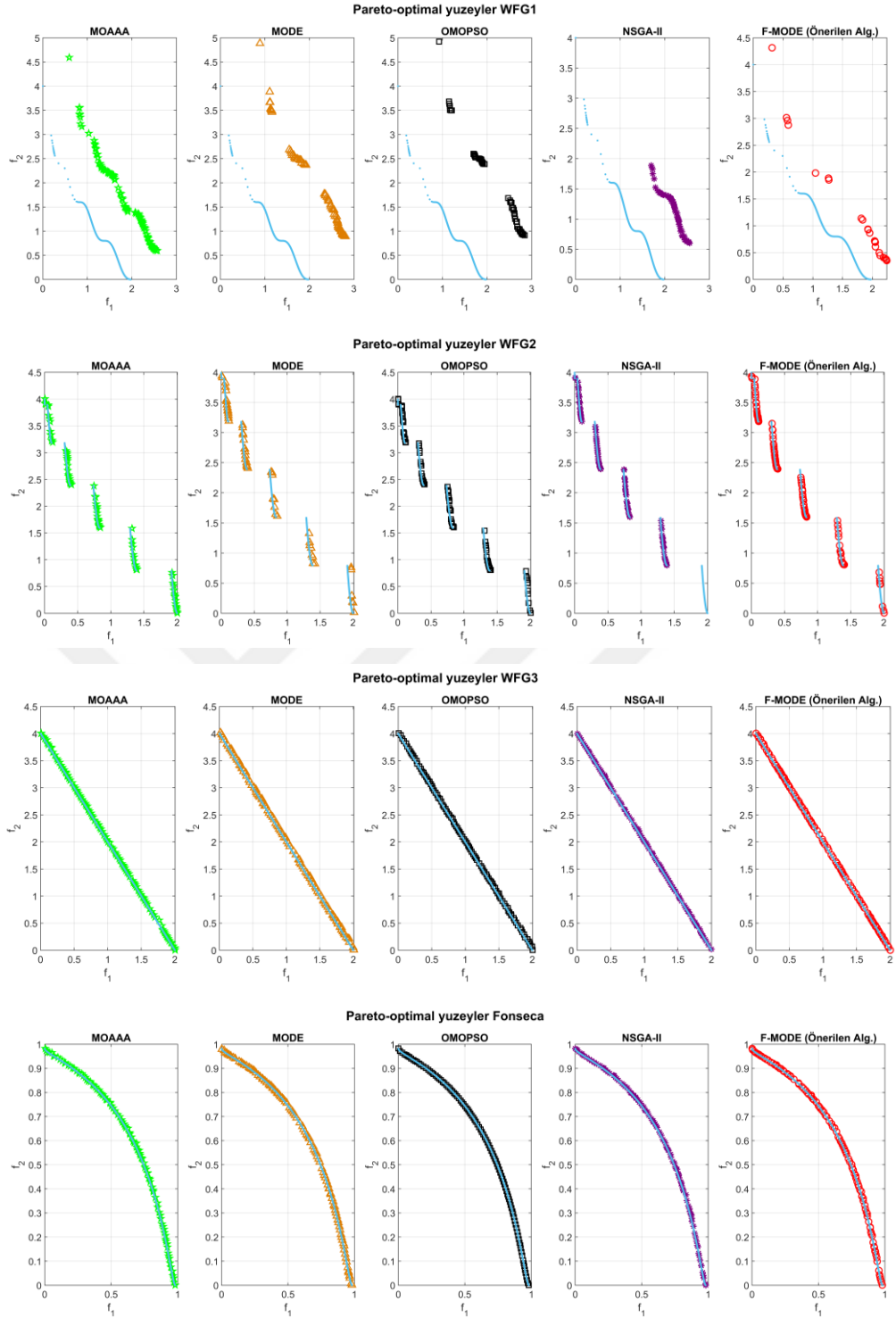
*WFG1* probleminde bütün algoritmalar neredeyse aynı oranda kötü bir yakınsama sağlamışlardır. *WFG2* probleminde ise yine yakın bir performans göstermelerine rağmen, en iyi performansı OMOPSO en kötü performansı ise NSGA-II göstermiştir. *WFG3*, *Fonseca*, *Kursawe* fonksiyonlarında bütün algoritmalar neredeyse kusursuz bir Pareto-optimal yüzey çizmişlerdir. *Schaffer* probleminde ise F-MODE ve OMOPSO gerçek yüzeylere çok iyi şekilde yakınsama sağlarken, diğer algoritmalar gerçek Pareto-optimal yüzeylere yaklaşamamışlardır.



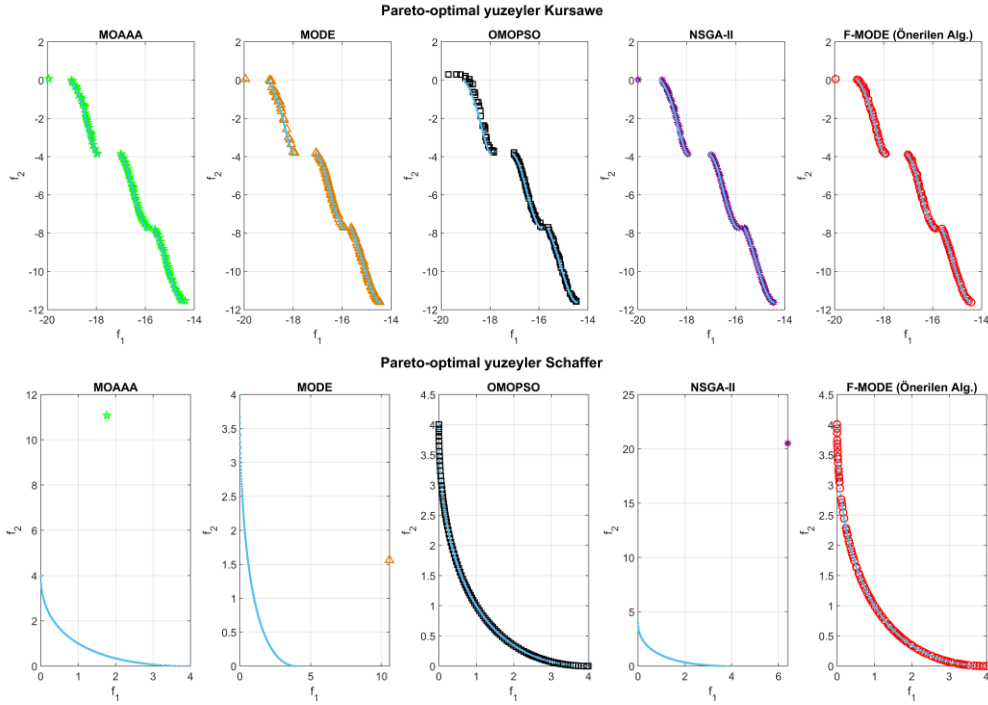
**Şekil 4. 8.** F-MODE ve diğer algoritmalar tarafından ZDT1-3, DTLZ1-2 problemleri için üretilen Pareto-optimal yüzeyler



Şekil 4. 9. F-MODE ve diğer algoritmalar tarafından DTLZ3-7 problemleri için üretilen Pareto-optimal yüzeyler



Şekil 4. 10. F-MODE ve diğer algoritmalar tarafından WFG1-3, Fonseca problemleri için üretilen Pareto-optimal yüzeyler



Şekil 4. 11. F-MODE ve diğer algoritmalar tarafından Kursawe, Schaffer problemleri için üretilen Pareto-optimal yüzeyler

Şekil 4.12'den itibaren kutu grafikleri gösterilmiştir ve sonrasında analizleri yapılmıştır. Kutu grafikleri hakkında ayrıntılı bilgi “Materyaller ve Yöntemler” bölümünde verilmiştir. Bu bölümde F-MODE ve karşılaştırılan algoritmaların HV ve IGD metrikleri üzerindeki kutu grafikleri verilmiştir.

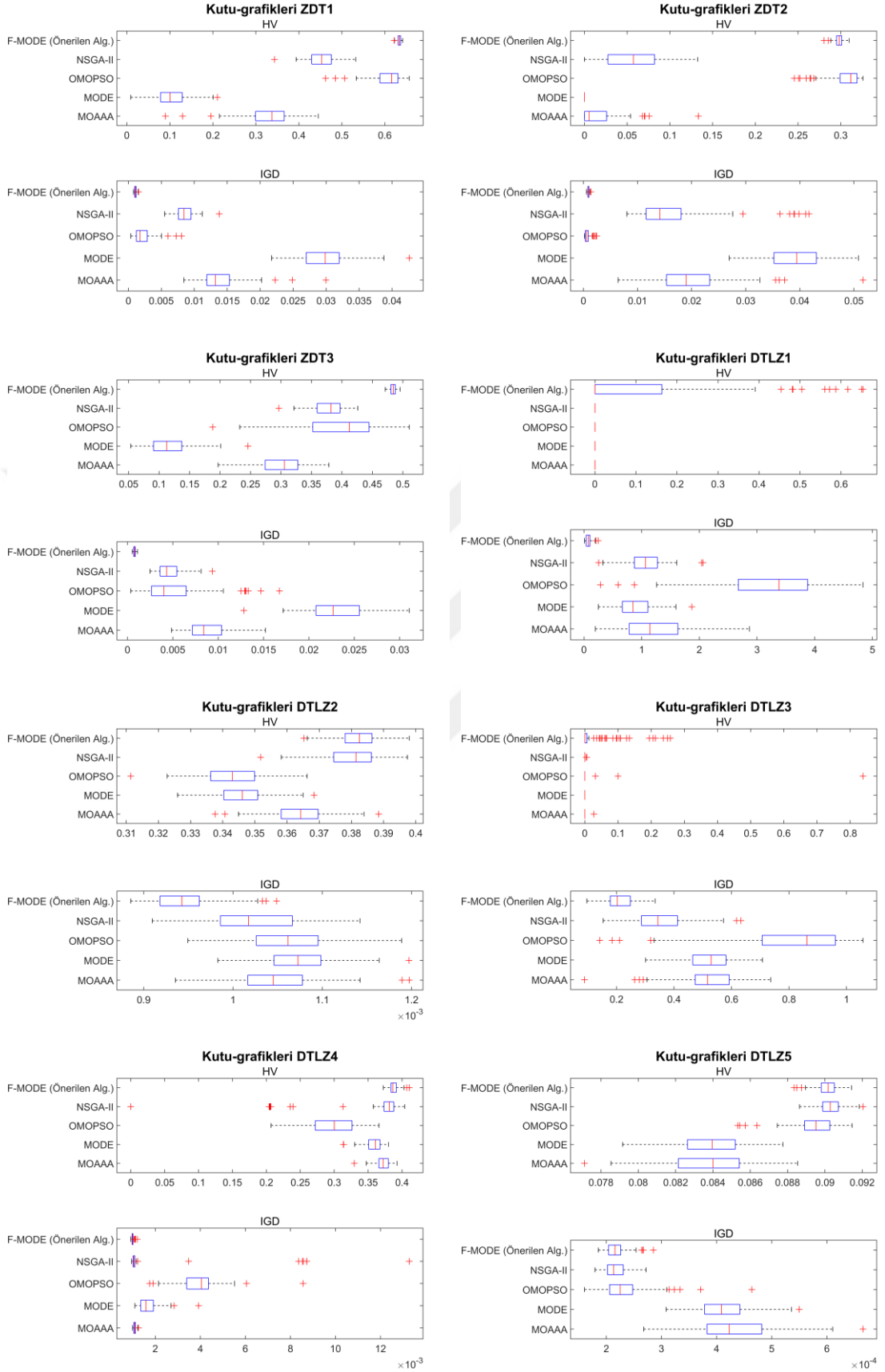
Şekil 4.12- Şekil 4.13'te gösterilen kutu grafiğini incelerken dikkat edilmesi gereken en önemli hususlardan birisi medyan değeridir. HV metriğine göre büyük değerler daha iyi kabul edildiği için sağa yakın bir medyan en iyi performans olarak kabul edilebilir. IGD metriğine göre ise küçük olan değer iyi sayıldığından sola yakın bir medyan değeri ilgili skorun daha iyi olduğunu gösterir.

ZDT1 probleminde F-MODE'un ürettiği kutu grafiğinin diğerlerine göre çok daha küçük bir kutu olduğuna dikkati çekmektedir. Çünkü kutu grafiği karşılaştırıldıkları diğer grafiklere nazaran ne kadar büyük olursa, sonuçlar o kadar dağınıktır denilebilir. Örneğin, ZDT1 problemi için diğer algoritmalar her bağımsız çalışmada farklı-farklı sonuçlar üretmişken, F-MODE'un ürettiği değerler daha tutarlıdır denilebilir. Başka bir ifadeyle standart sapmaları diğer sonuçlardan küçüktür. ZDT2 probleminde ise HV metriğine göre en küçük kutu grafiğini yine F-MODE üretmiştir. MODE algoritması ise bir sonuç üretememiş her seferinde sıfır değeri vermiştir. Bu yüzden bir kutu grafiği oluşmamıştır. Aynı problemin IGD metriğine göre ise F-MODE ve OMOPSO küçük grafikler üretmiştir. Dikkat edilecek olursa en çok aykırı değer üreten NSGA-II olmuştur. ZDT3

probleminde yine önerilen algoritma en iyi performansı gösterirken, özellikle IGD metriğine göre en çok aykırı değer üreten OMOPSO olmuştur.

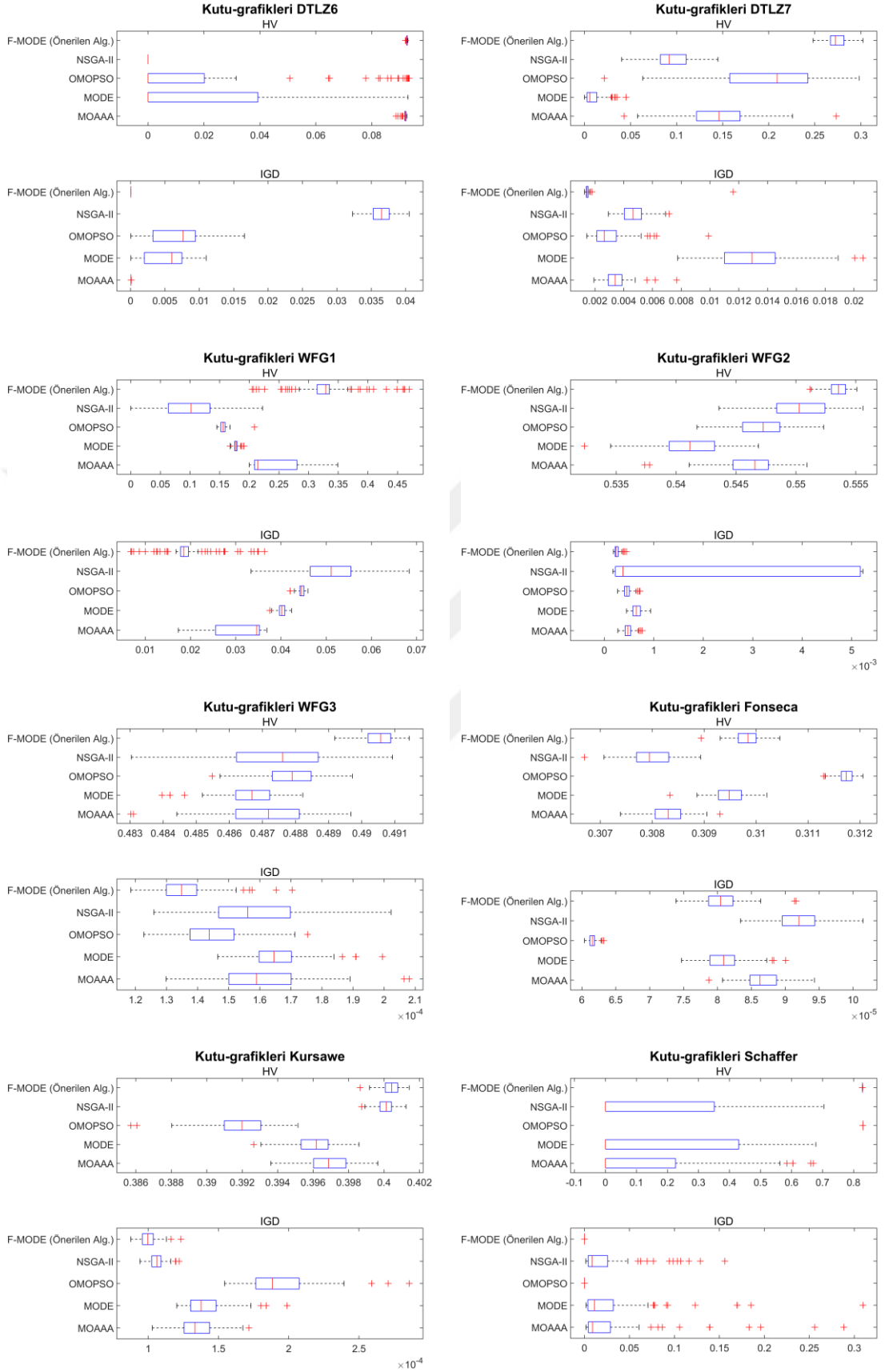
*DTLZ1* probleminde HV metriğine göre kutu grafiğini tam çizdirebilen tek algoritma F-MODE olmuştur. IGD metriğine göre ise en küçük kutu aynı algoritma tarafından üretilmiştir. *DTLZ2* probleminde tüm algoritmalar yakın performans göstermişlerdir ve nispeten hepsi de başarılı olmuşlardır. *DTLZ3* probleminde HV metriğine göre, neredeyse bütün algoritmalar kutu çizdirmek için gerekli verileri üretememişlerdir, bunun yanında F-MODE'un fazlasıyla aykırı veri üretmesi bu problem için yeterli değerlendirme sayısına ulaşılmadığını göstermektedir. Aynı problemin IGD metriğinde ise yine F-MODE en küçük kutu grafiğini çizdirmiştir ama medyanı simetrik değildir. *DTLZ4* probleminde OMOPSO hariç diğer algoritmalar başarılı bir kutu grafiği performansı göstermişlerdir ancak NSGA-II'nin aykırı değerleri oldukça fazladır. IGD'de yine benzer bir durum söz konusudur. *DTLZ5* probleminde; her iki metriğe göre de, F-MODE, NSGA-II, OMOPSO belirgin şekilde daha iyi sonuçlar üretmiştir. *DTLZ6* probleminde algoritmalar çok farklı sonuçlar ürettiğinden belirgin bir şekil oluşmamıştır ancak; ortalama, standart sapma grafiklerine ve Pareto-optimal yüzey grafiklerine bakıldığında MOAAA ve F-MODE'un mükemmel performans gösterdiği, NSGA-II'nin mantıklı herhangi bir değer üretmediği gözlemlenmiştir. IGD metriğinde sonuçlar biraz daha belirginleşmiştir. *DTLZ7*'de ise belirgin bir kutu grafiği performansı vardır.

*WFG1* probleminde bütün algoritmalar kötü performans göstermesine rağmen, OMOPSO ve MODE algoritmaları daha kararlı sonuçlar ürettiklerinden kutu grafikleri daha küçük görünmektedir. F-MODE'de aykırı verilerin çokluğu, aslında daha aramanın bitmediği anlamını taşımaktadır. *WFG2* probleminde HV metriğine göre bütün algoritmalar güzel grafik çizdirse de en iyi performansı F-MODE göstermiştir. IGD'ye göre ise NSGA-II bütün algoritmalar başarılıdır denilebilir. *WFG3* probleminde belirgin şekilde F-MODE performansının iyi olduğu görünmektedir; bunun yanında diğer algoritmalarda başarılı kutu grafiği üretmişlerdir. *Fonseca* probleminde açık ara en iyi performans OMOPSO algoritmasına aittir. *Kursewa*'da F-MODE ve NSGA-II rekabet halinde başarılı bir performans göstermişlerdir. *Schaffer* probleminde ise açık ara en iyi performanslar OMOPSO ve F-MODE ait olmasına karşın, diğer algoritmalarla aralarında çok fark olduğundan kutu grafiği belirgin bir şekilde görünmemektedir.



Şekil 4. 12. F-MODE ve diğer algoritmaların ZDT1-3, DTLZ1-5 problemleri için ürettiği performans metriklerinin kutu grafikleri





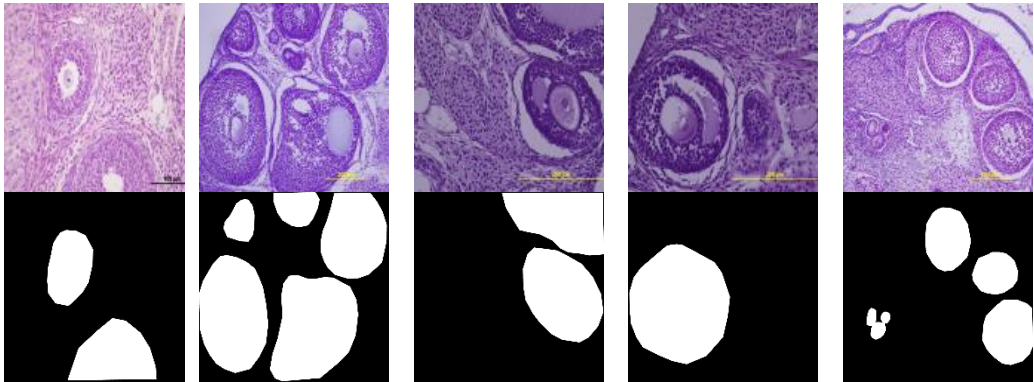
Şekil 4. 13. F-MODE ve diğer algoritmaların DTLZ6-7, WFG1-3, Fonseca, Kursawe, Schaffer problemleri için ürettiği performans metriklerinin kutu grafikleri

#### 4.1.4. Önerilen algoritmanın hiper parametre optimizasyonuna uygulanması

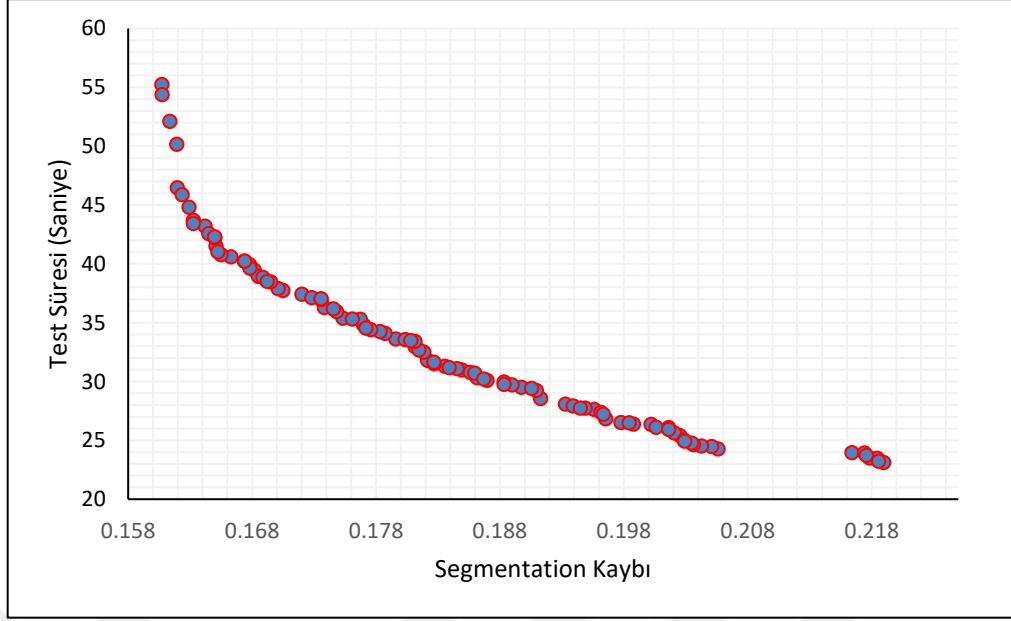
Bu çalışmanın temelleri İnik ve arkadaşlarının (2019) yılında yapmış olduğu ovaryum dokusuna ait bir çalışmasına dayanmaktadır. Bu çalışmada, ovaryum dokusunun beş farklı görüntüsü farklı boyamalarla oluşturulmuştur. Bu görüntüler, enine kesitlerden elde edilmiş ve farklı mikroskop büyütme seviyelerinde alınmıştır. İlgili çalışmada ovaryum görüntülerinin segmentasyonu ve sınıflandırılması için Derin Öğrenme tabanlı yeni bir yöntem önerilmiştir. Önerilen yöntemde ovaryum dokusuna ait primordial, primer, preantrel, sekonder ve tersiyerden oluşan beş farklı follikülün otomatik sayılması gerçekleştirilmiştir (İnik ve ark., 2019).

Bu tez çalışmasında ise, F-MODE algoritması kullanılarak hiper parametreler optimize edilmiştir. F-MODE bu probleme uyarlandığı için, ismi bu uygulamada MODE-CNN olarak güncellenmiştir. MODE-CNN algoritması hiper parametre optimizasyonu uygulamasında 100 popülasyon ve 50 iterasyon ile 5,000 maksFeste çalıştırılmıştır. Kodlamada gerçek değerli kodlama kullanılmıştır. Açıklanan parametreler dışındaki diğer parametreler test fonksiyonunda kullanılan aynı parametrelerdir.

Şekil 4.14'te, bu görüntüler ve bir uzman tarafından üretilen segmentasyonlar gösterilmektedir. MODE-CNN'deki test süresi ile segmentasyon kaybı arasındaki ilişkiyi gösteren Şekil 4.15'teki diyagram, CNN tabanlı yöntemin parametreleri için en uygun olanları göstermektedir. En uygun parametre değerleri,  $GS=31$ ,  $ND=32$  ve  $PA=0.864$ 'tir. Bu değerler, en düşük segmentasyon hatası ve en yüksek test süresi sağlamaktadır. Ayrıca, en yüksek segmentasyon hatası ve en düşük test süresi için uygun parametre değerleri  $GS=32$ ,  $ND=28$  ve  $PA=0.408$ 'dir (İnik ve ark., 2021).



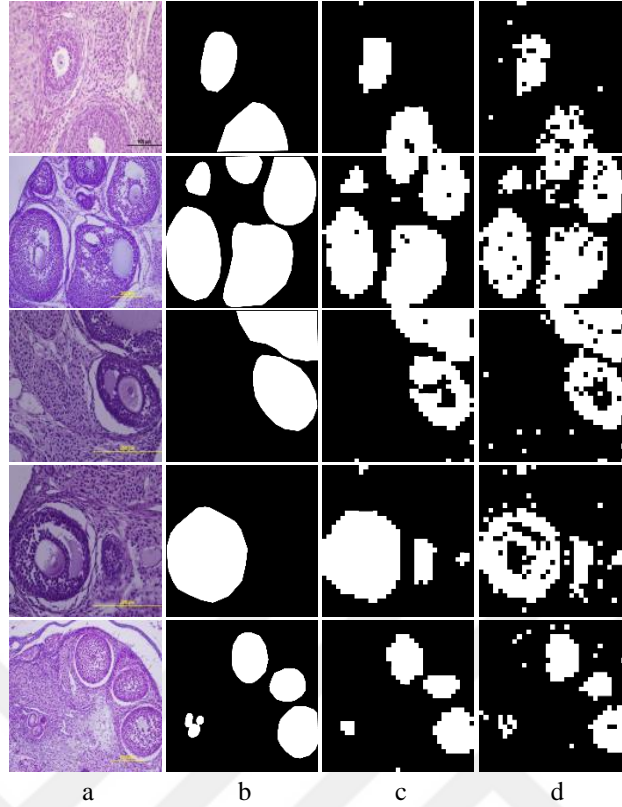
Şekil 4. 14. Üst sıra, parametrelerin optimizasyonu için kullanılan resimlerken; alt sıra tespit edilmesi gereken bölgelerdir (*ground truth images*).



**Şekil 4.15.** MODE-CNN ile hesaplanan segmentasyon kaybı ve test süresi için elde edilen Pareto cepheleri

Optimizasyon sonrası parametreler için, minimum test süresi ve segmentasyon kaybı veren görüntüler elde edilmiştir. Bu görüntüler Şekil 4.16'da sunulmuştur. Şekil 4.16.c'de en az segmentasyon hatası olan görüntülerin temel gerçeğe çok yakın olduğu görülmektedir. Ayrıca, Şekil 4.16.d'de test süresi minimum olan görüntüler için segmentasyon kaybının 0.2191 olduğu, ancak nesnelerin konumlarının çoğunlukla doğru bir şekilde tespit edildiği gözlemlenmektedir (İnik ve ark., 2021).

İlgili şekil dikkatlice incelendiğinde sadece 5,000 maksFes'te bulunan parametre değerleri başarılıdır denilebilir. Özellikle c bölümü, en kötü zamanda, en iyi sonucu temsil etmektedir. Kaliteden fedakârlık yapıldığında ise zaman performansında neredeyse üç katlara varan bir iyileşme olduğu Çizelge 4.5'te görülebilmektedir.



**Şekil 4. 16.** MODE-CNN ile elde edilen sonuçlar: (a) girdi görüntüleri, (b) girdi görüntülerinin temel gerçeği, (c) minimum segmentasyon kaybıyla elde edilen ile elde edilen görüntü ile minimum test süresi

Çizelge 4.5'te 100 adet popülasyon için segmentasyon kaybı ve test sürelerine göre optimize edilen  $GS$ ,  $ND$  ve  $PA$  değerleri verilmiştir. İlgili çizelgeden de anlaşılacağı üzere segmentasyon kaybı azalırca, test süresi artmaktadır yani amaçları çakışan çok amaçlı bir optimizasyon problemi ele alınmıştır. Bu çizelgeden de anlaşılacağı üzere  $GS$ , 31 de sabitlenmişken  $ND$  ve  $PA$  değerleri çok farklı aralıklarda seyretmektedir. Zaten veriler ayrıntılı şekilde incelendiğinde  $ND$  ve  $PA$  parametrelerinin her iki amacı da doğrudan etkilediği gözlemlenmiştir.  $ND$  ve  $PA$  parametrelerinin daha büyük değerler almasıyla test zamanı artmış ve segmentasyon kaybı azalmıştır.

Çizelge 4. 5. MODE-CNN ile 100 popülasyon için hesaplanan çözüm uzayı

Çözüm	GS	ND	PA	Segmentasyon kaybı	Test Süresi (Sn)	Çözüm	GS	ND	PA	Segmentasyon kaybı	Test Süresi (Sn)
1	31	32	5.98	0.18	29.74	51	31	21	5.17	0.20	25.56
2	31	32	5.66	0.19	28.44	52	31	21	5.34	0.19	26.42
3	32	9	4.70	0.22	23.03	53	31	32	5.03	0.20	25.01
4	31	25	4.14	0.21	23.08	54	31	32	8.41	0.16	48.41
5	31	32	8.36	0.16	47.90	55	31	23	4.57	0.20	23.53
6	31	32	8.91	0.16	55.69	56	31	32	6.73	0.17	34.42
7	31	32	8.72	0.16	53.13	57	31	32	5.71	0.19	28.70
8	31	32	8.08	0.16	44.84	58	31	32	6.10	0.18	30.67
9	31	32	7.87	0.16	42.94	59	31	32	6.66	0.18	34.19
10	31	32	8.70	0.16	52.46	60	31	21	5.08	0.20	25.31
11	31	32	8.56	0.16	50.44	61	31	32	6.58	0.18	33.28
12	31	32	6.94	0.17	35.85	62	31	23	4.42	0.20	23.45
13	31	32	8.10	0.16	44.69	63	31	32	7.40	0.17	39.45
14	31	32	8.50	0.16	49.82	64	31	32	7.28	0.17	38.17
15	31	21	5.35	0.19	26.53	65	32	9	4.48	0.22	22.30
16	31	21	5.48	0.19	27.26	66	31	32	6.20	0.18	31.13
17	31	32	6.87	0.17	35.31	67	31	32	7.67	0.16	41.45
18	31	32	7.12	0.17	37.17	68	31	32	6.17	0.18	30.92
19	31	32	6.66	0.18	33.96	69	31	21	5.13	0.20	25.48
20	31	32	6.40	0.18	32.30	70	31	32	5.77	0.19	29.21
21	31	32	6.97	0.17	36.05	71	31	32	6.32	0.18	31.96
22	31	32	8.44	0.16	48.60	72	31	23	4.59	0.20	23.87
23	31	32	7.09	0.17	36.91	73	31	32	5.32	0.19	26.41
24	31	21	5.58	0.19	28.01	74	<b>31</b>	<b>32</b>	<b>8.97</b>	<b>0.16</b>	<b>56.76</b>
25	31	32	6.43	0.18	32.61	75	32	9	4.27	0.22	22.26
26	31	32	7.01	0.17	36.43	76	31	32	6.76	0.17	34.75
27	31	32	5.64	0.19	28.27	77	31	32	6.78	0.17	34.95
28	32	31	4.62	0.22	22.72	78	31	32	5.07	0.20	25.04
29	31	32	6.56	0.18	33.41	79	31	32	5.90	0.19	29.38
30	31	32	7.32	0.17	38.62	80	31	32	6.07	0.18	30.34
31	31	32	7.35	0.17	38.74	81	31	21	5.23	0.20	26.07
32	31	32	7.79	0.16	42.19	82	31	32	5.80	0.19	29.08
33	31	32	6.45	0.18	32.75	83	31	32	7.69	0.16	41.33
34	31	32	7.22	0.17	37.93	84	31	21	5.47	0.19	27.30
35	31	22	5.92	0.19	29.53	85	31	32	6.24	0.18	31.45
36	31	32	7.40	0.17	40.14	86	31	19	5.04	0.20	25.03
37	31	32	8.94	0.16	56.41	87	31	21	5.22	0.20	25.93
38	<b>32</b>	<b>25</b>	<b>4.12</b>	<b>0.22</b>	<b>21.95</b>	88	31	21	5.13	0.20	25.34
39	31	32	7.75	0.16	41.71	89	31	21	5.20	0.20	25.78
40	31	21	4.80	0.20	23.98	90	31	29	5.08	0.20	25.17
41	31	32	6.37	0.18	32.04	91	31	23	4.13	0.21	23.17
42	31	32	5.75	0.19	28.79	92	31	32	7.40	0.17	39.40
43	31	32	7.63	0.16	40.94	93	31	32	6.77	0.17	34.85
44	31	32	4.98	0.20	24.73	94	31	18	4.29	0.20	23.18
45	31	28	5.00	0.20	24.84	95	31	32	5.70	0.19	28.56
46	31	32	6.01	0.18	30.16	96	32	25	4.28	0.22	22.11
47	31	32	7.61	0.17	40.65	97	31	21	5.07	0.20	25.06
48	31	32	5.98	0.18	29.75	98	31	21	5.30	0.20	26.39
49	31	32	7.36	0.17	38.97	99	31	32	6.27	0.18	31.64
50	31	21	5.56	0.19	27.80	100	31	32	4.44	0.20	23.35

Bu CNN tabanlı modelinin parametre optimizasyonu yapılması ile, hem segmentasyon kaybı en aza indirilmeye çalışılmış eş zamanlı olarak test süreside mümkün olan en kabul edilebilir şekilde optimize edilmiştir. Bu çalışmanın ışığında benzer durumlarda çok daha fazla parametreye sahip ya da üçten daha fazla amacı aynı anda optimize eden (*many-objective*) optimizasyon alanında çalışmalar yapılacağı öngörülmektedir.

Bu çalışma verileri sayesinde CNN mimarilerinin hiper parametre optimizasyonu manuel değil otomatik olarak gerçekleştirme imkânı elde edilmiştir; böylece elde edilen çözüm uzayından, zamanın ön plana çıktığı durumlarda en iyi zamana göre optimum segmentasyon kaybının olduğu parametreler kullanılacakken, segmentasyon kaybının önemsendiği durumlarda en az segmentasyon kaybına göre optimum zamanı sağlayan parametreler kullanılacaktır.

## 4.2. Asfalt Karışımı Problemi İçin D-MOGA Yaklaşımı

Bu bölümde, HMA probleminin ayrıntılarına yer verilecektir. Bu çalışmanın motivasyonunun ne olduğu ve neden böyle bir çalışmaya ihtiyaç olduğu yine bu bölümde açıklanacaktır. Ayrıca, hangi deney çıktıları ile bu çalışmaya başlandığı ve deneyin malzemeleri ve yapılışı hangi koşullarda gerçekleştirildiği üzerinde durulmuştur. Bundan sonra FL hakkında kısaca bilgi verildikten sonra, başarılı bir uzman sistemin doğrulanmasının nasıl sağlandığı tartışılmıştır. Bu bölümün son kısmında; D-MOGA algoritması tanıtılmış, test fonksiyonları üzerinde testler yapılmış ve diğer algoritmalarla çeşitli alanlarda karşılaştırılmış, son olarak HMA problemine uyarlanarak sonuçlar bulunmuş ve bulunan sonuçlar üzerinde analizler yapılmıştır.

### 4.2.1. Problemin tanımı

Agrega, sıcak asfalt karışımlarında kullanılan çakıl taşı benzeri kayaçların literatürde geçen teknik adıdır. Giriş bölümünde tanımına yer verilen HMA problemi için bu çalışmada üç tip agrega kullanılmıştır. Bu üç tip agrega sınıfı: Doğal agrega, mermer agregası ve geri dönüştürülmüş beton agregasıdır. Bunlar asfalt karışımına şu formlarda eklenmişlerdir: Doğal kaba agrega (*Natural coarse aggregate, NC*), doğal ince agrega (*Natural fine aggregate, NF*), mermer kaba agrega (*Marble coarse aggregate, MC*) ve geri dönüştürülmüş beton agregasıdır (*Recycled concrete fine aggregate, RF*). Bu 4 agrega formu ve bitümlerle birlikte toplam 5 girdiye sahip bir problem oluşturulmuştur. Bitüm (*Bitümen*), asfaltlamada yaygın olarak kullanılır, petrolün damıtılması sırasında arta kalan yüksek kaynama noktalı hidrokarbonlardan oluşan, doğal ısıda katı formda olan, koyu kestane renginde bir karışımdır.

Çalışmanın bu kısmında amaç; asfaltın kalitesini bozmadan, mermer ve beton atıklarının doğal agregalar yerine kullanılabilmesinin önerilmesidir. Genel olarak, az miktarda bitüm kullanılması asfaltta dayanıklılığı olumsuz olarak etkiler; ancak kabul edilebilir seviyelerde bitüm seviyesinin azaltılması asfalt maliyetini düşüreceğinden, bitüm miktarının kabul edilebilir düzeyde farklı seviyelerini bulmanın başta maliyet olmak üzere birçok avantajı vardır.

Çizelge 4.6'da Alakara ve arkadaşları tarafından, laboratuvar ortamında gerçekleştirilen bir deneyde; çeşitli agreg türlerinin, farklı oranda kullanıldığı da *MS*, *AVC*, *VFA* çıktılarına etkileri gözlemlenmiştir (Alakara ve Agaoglu, 2020). İlgili deney seti, toplamda 42 farklı deneyden oluşmuştur. Bu tarz çalışmalarda, verilerin yetersiz

olması veya girdi ve çıktılar arasında doğrusal bir bağlantı kurulamaması sebebiyle doğru sonuçlar üretebilen bir amaç fonksiyonu oluşturmak oldukça zordur. Literatür incelendiğinde bu tür problemler ele alınırken bir amaç fonksiyonu üretmek yerine bulanık mantık tabanlı uzman sistem ve benzeri sistemlerin kullanıldığı görülmüştür (Kara ve Karacasu, 2017). Bu ve benzeri çalışmalardan esinlenilerek bu çalışma kapsamında önerilen D-MOGA için amaç fonksiyonu gibi işlev üstelenecek ve D-MOGA ile hibrit şekilde çalışabilecek bir FL modeli oluşturulmuştur. Böylece HMA için kullanılacak girdi değerleri en uygun çıktı değerlerine göre optimize edilecektir. Çizelge 5.1'deki ki; bitüm oranı (*Bitümen Rate, BR*), toplam asfalt karışımının %3.5 ila %6.5 oranlarında değişen bitüm değerlerini göstermektedir. Agrega karışımları içinde temelde ise kaba agrega ve ince agregalardan oluşan iki grup vardır. Kaba agrega *NC* veya *MC*' den oluşmaktadır. Benzer şekilde ince agregalar *NF* ve *RF* karışımlarından oluşmaktadır. İlgili tabloda görüldüğü gibi *NC* ve *MC* oranlarının toplamları ile aynı şekilde *NF* ve *RF* oranlarının toplamları 100'dür. *RF* bina yıkımlarından arta kalan beton atığı iken, *MC* ise mermer ocaklarından çıkan atıkların öğütülmesi ile elde edilmektedir. *NC* ve *NF* ise adlarından da anlaşılacağı üzere asfaltta kullanılan doğal agregalardır. Çıkış değerleri olarak kullanılan *MS*, dayanıklılığı temsil ettiği için ne kadar yüksek bir *MS* değeri elde edilirse o kadar iyi olacağı kabul edilir. Uygun asfaltlama için oluşturulmuş, Türk Karayolları Yapım Şartnamesine (KGM, 2013) göre *MS*'nin 750'den yüksek olduğu durumlarda kabul edilebilir bir dayanıklılığa sahiptir. *MS* değeri için ilgili koşul sağlandığından, bu yönden bakarak deneyin başarılı olduğu söylenebilir. Aynı şartnamede ise *AVC* değerinin 4-6 arasında ve *VFA* değerinin 60-75 arasında olduğu durumlarda kabul edilebilir olduğu belirtilmiştir. Bu üç şartı aynı anda ele alınırsa 42 deneyden sadece 7 deney bu üç şarta da uyduğu görülmektedir. Bu üç şartı sağlayan sonuçlar kalınlaştırılarak gösterilmiştir. Buna göre tabloya ilk bakışta asfaltın içine doğal agregalar yerine kullanılan mermer ve beton atığı beklenen sonucu vermiştir denilebilir, çünkü şartnameye uygun olan 7 sonucun 6 tanesinde bu atıklardan yararlanılmıştır.



Çizelge 4. 6. Laboratuvar deney sonuçları (Alakara ve Agaoglu, 2020)

Parametreler					Deney Sonuçları		
BR (%)	NC (%)	NF (%)	MC (%)	RF (%)	MS (kg)	AVC (%)	VFA (%)
3.5	100	100	-	-	1367	7.68	48.34
4	100	100	-	-	1450	5.67	59.61
4.5	100	100	-	-	<b>1343</b>	<b>4.41</b>	<b>68.32</b>
5	100	100	-	-	1329	2.58	80.59
5.5	100	100	-	-	1260	2.13	84.66
6	100	100	-	-	1204	1.77	87.8
6.5	100	100	-	-	1156	1.54	89.89
3.5	-	100	100	-	1054	6.48	53.05
4	-	100	100	-	<b>1137</b>	<b>5.09</b>	<b>62.44</b>
4.5	-	100	100	-	1126	3.7	72.21
5	-	100	100	-	1123	2.93	78.52
5.5	-	100	100	-	1081	1.94	85.9
6	-	100	100	-	1056	1.49	89.6
6.5	-	100	100	-	954	0.96	93.53
3.5	-	50	100	50	1122	8.98	39.07
4	-	50	100	50	1165	7.1	49.26
4.5	-	50	100	50	1207	6.26	55.96
5	-	50	100	50	<b>1242</b>	<b>5.49</b>	<b>62.11</b>
5.5	-	50	100	50	<b>1277</b>	<b>4.02</b>	<b>71.53</b>
6	-	50	100	50	1288	2.15	84
6.5	-	50	100	50	1214	1.65	88.16
3.5	100	50	-	50	1408	8.98	38.5
4	100	50	-	50	1548	8.1	45.21
4.5	100	50	-	50	1557	5.29	59.86
5	100	50	-	50	<b>1533</b>	<b>4.49</b>	<b>66.61</b>
5.5	100	50	-	50	1496	3.2	75.89
6	100	50	-	50	1389	2.61	80.9
6.5	100	50	-	50	1377	1.68	87.88
3.5	-	75	100	25	1153	7.62	46.11
4	-	75	100	25	1190	6.08	55.71
4.5	-	75	100	25	<b>1210</b>	<b>4.78</b>	<b>64.71</b>
5	-	75	100	25	1281	3.52	73.75
5.5	-	75	100	25	1267	1.95	85.02
6	-	75	100	25	1178	1.49	89
6.5	-	75	100	25	1115	0.96	93.14
3.5	100	75	-	25	1212	8.04	44.26
4	100	75	-	25	1227	5.97	55.87
4.5	100	75	-	25	<b>1337</b>	<b>4.79</b>	<b>64.41</b>
5	100	75	-	25	1347	3.07	76.22
5.5	100	75	-	25	1351	2.28	82.69
6	100	75	-	25	1232	1.96	85.8
6.5	100	75	-	25	1199	1.44	89.98

İlgili deney, kendi alanında başarılı sayılabilen bir deneydir. Çünkü önemli bir çıktı olan *MS* çıktısı deneyin tamamında şartnameye uygun olarak 750'den büyüktür. Bunun yanında parametrelerin yüzbinlerce ihtimal arasından tecrübelerle dayanarak yani bir anlamda rastgele seçilerek en iyi çıktıyı beklemek olanaksızdır. Bu tür deneylerin daha başarılı olabilmesi için; araştırmacıya yol gösterecek çok amaçlı bir problem çözme yaklaşımının gerekli olduğu düşünülmektedir. Bu tespit, çalışmanın bu kısmının ana motivasyonlarından biridir.

Kısaca problem şu şekilde tanımlanabilir; laboratuvar ortamlarında gerçekleştirilen deneylerde, optimum girdi miktarlarını bulma ihtimali, deneylerin parametre sayılarına bağlı olarak; yüz binde bir, hatta milyonda birlere kadar değişebilmektedir. Bu bakış açısı ile iki problemle daha karşılaşılmaktadır.

Bunlardan birincisi; gerçek deney ortamının modellenmesidir, çünkü girdilerin sonuçları nasıl etkilediğini anında görmek gerekmektedir. Bu modelleme için işlevsel bir matematiksel amaç fonksiyonu üretmek gereklidir ancak, yeterli sayıda veri olmaması ve giriş çıkışlar arasında doğrudan bir bağ kurulamaması sebepleri ile amaç fonksiyonu kullanmak tercih edilmemiştir. Bu yüzden ilgili çalışmaların literatürünün ışığında FL uzman sistemi, deneyin modellenmesi için tasarlanmıştır. FL'nin işlevi, HMA problemine ait beş giriş parametresini kullanarak gerçek deney verilerine uygun şekilde çıktılar üretilmesidir. Bu deney için 82 adet kural oluşturularak FL sistemi gerçek deney sonuçlarına mümkün olduğunca yaklaştırılmıştır.

İkinci problem ise FL sistemi için, manüel olarak girildiği takdirde hala yüz binlerce giriş ihtimali var olmasıdır. Parametre sayısı çok daha fazla olan ve alt-üst limitlerin arası daha geniş olan veya çözüm uzayı sürekli olan problemlerde, doğru girdilerin manüel olarak bulunması imkânsız hale gelmektedir. Hiç şüphesiz, bunun için optimizasyon algoritmalarına ihtiyaç vardır. Problem çıktısı, yani amacı üç tane olduğundan ve üç çıktının hepsi eş zamanlı olarak beş girdiye bağlı olduğundan bu problem, çok amaçlı bir optimizasyon problemidir. Dolayısıyla çok amaçlı bir optimizasyon algoritması kullanmak bu problemin çözümü için gereklidir denilebilir.

#### 4.2.2. Motivasyon ve katkılar

Çalışmanın bu kısmında iki ana motivasyon etkili olmuştur; bunlardan ilki, doğal agregaların kullanılması yerine çevreye atık olarak ekstra yük getiren mermer ve beton atıklarının asfalt karışımlarında değerlendirilebilme ihtimalidir. Böylece çevre temizliğine bir katkı sağlanacak, aynı zamanda asfalt maliyeti az da olsa iyileştirilebilecektir.

Bu tür gerçek dünya problemlerinde mümkün olduğunca çeşitliliği ve yakınsamayı dengeleyebilen ve rakipleriyle rekabet edebilen bir MOEA'nın gerekliliği, bu çalışmanın ikinci motivasyonunu oluşturmaktadır. Çalışma bu yönüyle ele alınırsa, geliştirilecek olan algoritmanın çaprazlama operatörüne odaklanmalıdır çünkü çaprazlama işlemi çeşitliliğe ve yakınsamaya doğrudan etki ettiği için çok önemlidir. Başka bir deyişle diğer tüm koşullar eşit olduğunda çaprazlama operatörü ne kadar iyi olursa, algoritma o kadar başarılıdır denilebilir. Bu problemi çözmek için şöyle bir hipotez kurulmuştur: İlgili problemi çözecek algoritmanın özellikle çeşitlilik performansını iyileştirmek için literatürde sıklıkla geçen üç adet çaprazlama modeli

rastsallığa bağlı olarak hibrit şekilde kullanılmıştır. Aynı zamanda NFL teoremine göre bir algoritma ne kadar başarılı olursa olsun her problemde aynı başarıyı sürdürmez bu yüzden her zaman yeni algoritmalara, yeni problem çözme yaklaşımlarına ihtiyaç vardır (Wolpert ve Macready, 1997) (Adam ve ark., 2019). Bu kapsam da D-MOGA olarak isimlendirilen Pareto tabanlı yeni bir algoritma önerilmiştir. D-MOGA için ilgili testler yapıldıktan ve gerekli karşılaştırmalara tabi tutulduktan sonra performans analizleri yapılmış ve HMA problemine uyarlanmıştır.

Sonuç olarak, yol yapımında sıklıkla kullanılan asfalt karışımı oranları için yüz binlerce ihtimal arasından en uygun çıktıları sağlayan girdilerin olduğu bir çözüm uzayı elde edilmiştir. Bu sonucu elde etmek için ilk adımda gerçek deney verilerini baz alarak 82 adet kurala dayalı bir FL uzman sistemi geliştirilmiştir. Bu uzman sistemin performansı test edildikten sonra D-MOGA ile entegre çalışacak şekilde algoritmaya eklenmiştir. Bu çalışma kapsamındaki literatür taramasına göre, bu tez çalışmasında ilk defa, HMA problemi FL ile hibrit şekilde çalışan çok amaçlı bir optimizasyon algoritması kullanılarak ele alınmıştır.

Bu çalışmanın literatüre katkıları kısaca şöyledir:

- D-MOGA literatürde sıklıkla geçen başarılı çaprazlama modellerden ilham alan, artırılmış rastgeleliğe sahip hibrit bir çaprazlama operatörüne sahiptir. Bu çalışma ile geniş arama uzayına sahip gerçek dünya problemleri için güçlü bir yaklaşım literatüre kazandırılmıştır.
- Literatürde genel olarak sabit değerlerden oluşan çaprazlama oranı (*Crossover Rate, CR*) bu çalışmada sabit değer yerine, değerlendirme (*Evaluation*) sayısına bağlı olarak dinamik bir şekilde değişen yeni bir yaklaşım önerilmiştir. Böylece fonksiyon değerlendirmenin başlarında yüksek bir *CR* ile çaprazlama yapılmış ve daha hızlı bir yakınsama hedeflenmiştir, sürecin sonlarına doğru ise *CR* oranı düşük tutulmuş ve yerel arama daha etkin hale getirilmiştir.
- Son olarak, gerçek deney verileri ile kural tabanlı FL uzman sistemi tasarlanmış ve bu uzman sistem D-MOGA içerisinde amaç fonksiyonu işlevi üstlenmiştir.

### 4.2.3. Asfalt karışım deneyi

Önceki bölümlerde bahsedildiği gibi; Çizelge 5.1'deki, Alakara ve arkadaşları (2020) tarafından gerçekleştirilen bir deney verileri ile bulanık uzman sistem geliştirmiş, daha sonra bu uzman sistemle D-MOGA'nın entegre şekilde çalışması sağlanmıştır. Bu yüzden bu bölümde, ilgili deneyden gerekli ayrıntıların verilmesi uygun olacaktır.

#### 4.2.3.1. Deney malzemeleri

Deneyde kullanılan bitümün penetrasyon sınıfı 50/70 olarak belirlenmiştir. Kullanılan bitümün karayolları şartnamesine (KGM, 2013) uygunluğu gerekli testler yapılarak araştırılmıştır. Daha önce belirtildiği gibi karışımda agrega olarak; doğal agrega (*Natural Aggregate, NA*), atık mermer (*Waste Marble, WM*) ve geri dönüştürülmüş beton agregası (*Recycled Concrete Aggregate, RCA*) kullanılmıştır.

Mermer atıkları Tokat şehrindeki (Türkiye) bir mermer tesisinden elde edilmişlerdir ve Şekil 4.17.a'da kullanılan mermerlerin yapıları görünmektedir. Aynı şekilde ince agrega yerlerine kullanılan RCA'lar (Şekil 4.17.b) Tokat şehri kentsel dönüşüm atık alanından elde edilmiştir. Atık mermerler ve kentsel dönüşüm beton atıkları, laboratuvar tipi çeneli kırıcı kullanarak istenilen boyutlara getirilmiştir. Bu agregaların fiziksel özellikleri Çizelge 4.7'de, kullanılan bitümün özellikleri ise Çizelge 4.8'de gösterilmiştir. Deneyler kapsamında hazırlanan, sıcak bitümlü karışımların bağlayıcı tabakası için agrega gradasyonu Şekil 4.18'de gösterilmiştir (Altıok ve ark., 2023).



a



b

Şekil 4. 17. (a) Atık mermer alanı, (b) Atık beton alanı

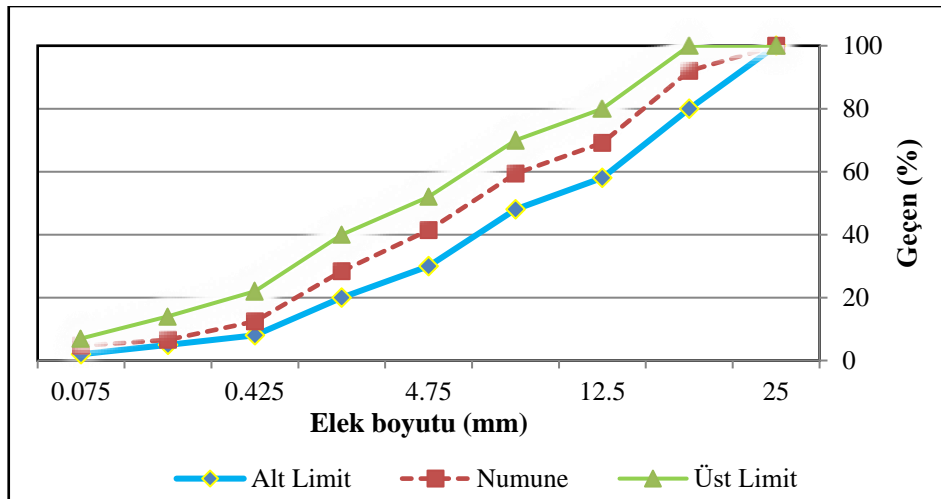
Çizelge 4. 7. Agregaların fiziksel özellikleri

Test	Yöntem	Değer			Standartların gerekliliği
		NA	WMA	RCA	
Özgül ağırlık (kaba agrega)	ASTM C 127				
Toplu		2.68	2.689	-	-
SSD					
Görünür		2.714	2.710	-	-
Özgül ağırlık (ince agrega)	ASTM C 128				
Toplu		2.684	-	2.269	-
SSD					
Görünür		2.734	-	2.681	-
Özgül ağırlık (dolgu)	ASTM D854	2.724	-	-	-
Su emme (kaba)	ASTM C 127	0.45	0.30	-	-
Su emme (para cezaları)	ASTM C 128	0.67	-	6.82	-
Los Angeles aşınması (%)	ASTM C131	20.65	20.78	-	< 30%
Yassılık indeksi (%)	ASTM D 4791	17.62	23.07	-	< 35%

Çizelge 4. 8. Kullanılan bitümün fiziksel özellikleri

Test	Sonuçlar	Standart
Penetrasyon (25°C, 1/10 mm)	58.4	ASTM D5
Yumuşama noktası (°C)	50	ASTM D36
Özgül ağırlık (g/cm <sup>3</sup> )	1.036	ASTM D70
Parlama noktası (°C)	296	ASTM D92
Yumuşaklık (25°C, 5 cm/dk)	> 100	ASTM D113

Şekil 4.18'deki agrega gradasyon (sınıflandırma) eğrisine göre; agrega karışımının %58.6'sının kaba agrega, %36.7'sinin ince agrega ve %4.7'sinin dolgu agregadan oluştuğu görülmektedir.



Şekil 4. 18. Agrega derecelendirme eğrisi

#### 4.2.3.2. Marshall tasarımı

Bitümlü sıcak karışım örneklerinin hazırlanmasında Marshall tasarımı metodu kullanılmıştır. İlgili deneyde altı farklı agrega karışımı denenmiş ve bağlayıcı tabakanın sınıflandırılmasına göre hazırlanmıştır. Çizelge 4.9'da gösterilen agrega gruplarının karışımında 7 farklı bitüm oranı denenmiştir. Agrega karışımlarının ağırlıklarına bağlı olarak bitüm miktarı kullanılmış olup, her bir bitüm yüzdesi için 3 örnek hazırlanmıştır. Bitümlü karışım numuneleri hazırlanırken, bitüm ve 1,150 gram agrega karışımı 155°C sıcaklığındaki etüv cihazında 24 saat bekletilmiştir. Sonraki adımda 1,150 g agrega karışımına farklı oranlarda bitüm ilave edilerek karıştırılmıştır. Karıştırma işlemi bittiğinde, karışımlar numune kalıplarına dökülmüştür. Son adım olarak karışım örnekleri sıkıştırma çekici yardımı ile her iki tarafına 75'er darbe uygulanarak iyice preslenmiş ve soğumaya bırakılmıştır. Soğutulan asfalt örneklerinin *AVC* ve *VFA* değerleri belirlendikten sonra, en son *MS* değerlerinin ölçülmesi için numuneler 60°C sıcaklığındaki su banyosunda 30 dakika bekletilmiş ve *MS* cihazı ile ölçüm gerçekleştirilmiştir (Altiok ve ark., 2023).

Çizelge 4.9'a göre agrega karışım oranları, toplam kaba agrega oranı %58.6 iken, toplam ince agrega oranı %36.7 ve dolgu agrega oranı %4.7'dir. Tüm gruplardaki dolgu agrega oranları sabit tutulmuştur.

Çizelge 4. 9. Agrega gruplarının karışım oranları

Grup adı	Kaba Agrega (%)		İnce Agrega (%)		Filler (%)	Toplam (%)
	NC	MC	NF	RF		
1 (NC - NF )	58.6	-	36.7	-	4.7	100
2 (MC- NF )	-	58.6	36.7	-	4.7	100
3 (MC- NF50RF50)	-	58.6	18.35	18.35	4.7	100
4 (NC - NF50RF50)	58.6	-	18.35	18.35	4.7	100
5 (MC- NF75RF25)	-	58.6	27.53	9.17	4.7	100
6 (NC- NF75RF25)	58.6	-	27.53	9.17	4.7	100

Karışım gruplarının içerikleri: *NC-NF* grubu, tamamı doğal agregadan alınan kaba, ince ve dolgu agregaları. *MC-NF* grubu, kaba agreganın tamamı mermerden, ince agregalar ise tamamen doğal agregadan alınmıştır. *MC-NF50RF50* grubu, kaba agreganın tamamı mermerden, ince agreganın %50'si doğal agregadan ve %50'si *RCA*'dan alınmıştır. *NC-NF50RF50* grubu, kaba agreganın tamamı doğal agregadan, ince agreganın %50'si doğal agregadan, %50'si *RCA*'dan alınmıştır. *MC-NF75RF25* grubu, kaba agreganın tamamı mermerden, ince agreganın %75'i doğal agregadan, %25'i kentsel

dönüşüm atıklarından alınmıştır. Son olarak; *NC-NF75RF25* grubu ise, kaba agreganın tamamı doğal agregadan, ince agreganın %75'i doğal agregadan ve %25'i kentsel dönüşüm atıklarından alınmıştır (Altiok ve ark., 2023).

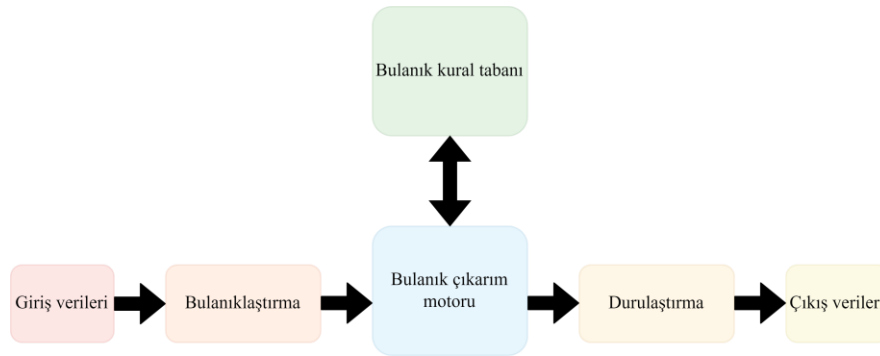
#### 4.2.4. Bulanık mantık

FL, Lotfi Zadeh tarafından 1965 yılında geliştirilen belirsiz durumlar için geliştirilmiş matematiksel bir araçtır. Bu model, kelime tabanlı hesaplama kavramını tanıtmıştır. Bulanık teoride, "çok", "düşük", "orta", "sıklıkla", "az" gibi dil yapılarını, temsil eden mekanizmalar bulunur. Gerçek hayatta sıklıkla karşılaşılan belirsizlikleri modelleme yeteneği, bulanık kümelerin en önemli özelliklerinden biridir. FL sistemleri, hızlı, sorunsuz ve kontrol karmaşıklığı çok az olan modellerdir. FL modelinin işleyişi, öncelikle girdi değişkenlerinin belirlenmesi ve bu değişkenlerin bulanık kümelerle ifade edilmesi ile başlar. Ardından, uygun üyelik fonksiyonları seçilerek, her bir girdi değerinin kümeye ne kadar uyduğu ölçülür. Belirlenen girdi değişkenleri arasındaki ilişkileri ifade eden kurallar belirlenir. Bu kurallar, üyelik fonksiyonları ve belirlenen girdi değişkenleri kullanılarak sonuçlar çıkarılır. Elde edilen sonuçlar gerektiğinde yeniden bulanıklaştırılır ve belirli bir amaca uygun olarak yorumlanır.

FL modeli, belirsizliğin yüksek olduğu karmaşık sistemlerde kullanılan etkili bir araçtır. Bu model, hava durumu tahmini, endüstriyel kontrol sistemleri, yapay zekâ ve robotik gibi birçok alanda yaygın olarak kullanılmaktadır.

Modelin temel akış şeması şöyledir:

1. Girdi değişkenleri belirlenir ve bulanık kümelerle ifade edilir.
2. Girdi değişkenleri için uygun bir üyelik fonksiyonu seçilir.
3. Kurallar belirlenir.
4. Kuralların çıkarımı yapılır ve sonuçlar elde edilir.
5. Sonuçlar, gerekirse, yeniden bulanıklaştırılır.
6. Sonuçlar, belirli bir amaç için yorumlanır (Alakara ve Agaoglu, 2020).



Şekil 4. 19. Bulanık mantığın temsili akış diyagramı

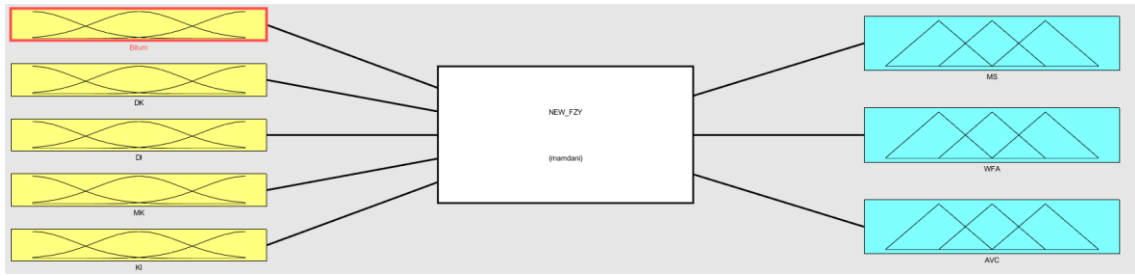
Bu çalışmada, Marshall deneyi sonuçları Mamdani tipi bir Fuzzy Logic çıkarım sistemi (FIS) kullanılarak değerlendirilmiştir. Modelde, agrega karışım oranları ve bitüm oranı girdi verileri olarak kullanılmıştır ve çıkış olarak MS, AVC ve VFA değerleri belirlenmiştir. Giriş ve çıkış sembolleri Çizelge 4.10'da gösterilmiştir. Modelde, FL sistemi için 82 "if-then" kuralı oluşturulmuştur ve bu kurallar Çizelge 4.11'de sunulmuştur. Deneysel sonuçlara ve model çıktılarına dayalı olarak elde edilen sonuçlar Çizelge 4.12'da yer almaktadır.

Çizelge 4. 10. Modelde kullanılan semboller

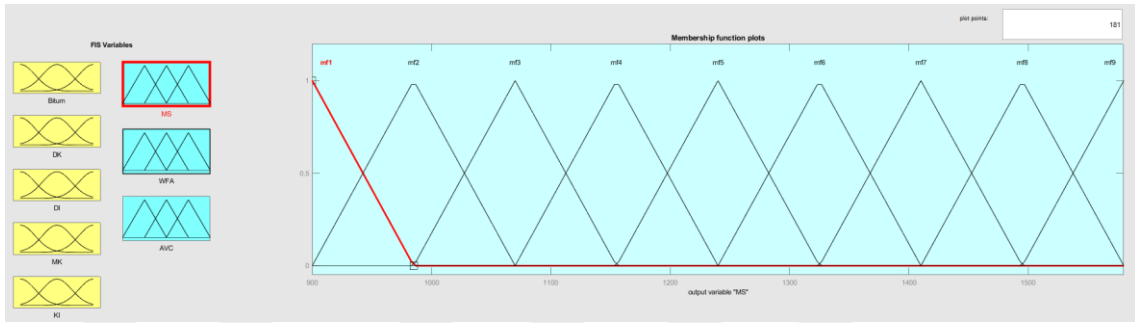
Girdiler					Çıktılar		
BR (%)	NC (%)	NF (%)	MC (%)	RF(%)	MS (kg)	AVC (%)	VFA(%)
B1	NC1	NF1	MC1	RF1	MS1	AVC 1	VFA 1
B2	NC2	NF2	MC2	RF2	MS2	AVC 2	VFA 2
B3		NF3		RF3	MS3	AVC 3	VFA 3
B4					MS4	AVC 4	VFA 4
B5					MS5	AVC 5	VFA 5
B6					MS6	AVC 6	VFA 6
B7					MS7	AVC 7	VFA 7
B8					MS8	AVC 8	VFA 8
B9					MS9	AVC 9	VFA 9

Şekil 4.20 ve Şekil 4.21'de tasarlanan FL modelinin ekran görüntüleri verilmiştir. Bu modelde Mamadani yöntemi kullanılmıştır. Mamadani tipi FL modeli en yaygın kullanılan yöntemlerden biridir. Mamadani tipi bulanık mantık modeli, bir giriş setine dayanan belirli bir çıktıyı hesaplamak için bir dizi kural kullanmaktadır. Bu kural dizisi, dilsel ifadeleri ve belirli bir kurala uyan giriş değerlerini içermektedir. Her bir kural, giriş değerlerine dayanarak bir çıktı değeri belirlemektedir. Sonrasında, tüm kural çıktıları bir araya getirilmekte ve birleştirilen çıktı değerine göre sonuç tespit edilmektedir (Altiok ve ark., 2023).





Şekil 4. 20. FL modeli girişi ve çıkışları



Şekil 4. 21. FL modeli MS değerlerinin sınıflandırılması

Çizelge 4.11; Çizelge 4.10'daki verilere dayanarak oluşturulan Mamdani tipi bir FL çıkarım sistemi için kullanılan kuralları içermektedir. Bu tablo, agrega karışım oranları ve bitüm oranı gibi beş girdi değişkeni ile MS, AVC ve VFA gibi üç çıktı değişkeni kombinasyonları için toplam 82 adet "if-then" kuralını göstermektedir (Altioğ ve ark., 2023). Bu çalışmada FL kapsamında üyelik fonksiyonu olarak TRIMF (*Triangular Membership Function*) kullanılmıştır.

Çizelge 4. 11. Kurallar tablosu

Girdiler					Çıktılar		
BR (%)	NC (%)	NF (%)	MC (%)	RF (%)	MS (kg)	AVC (%)	VFA (%)
2	2	3	1	1	6	3	7
2	2	3	1	1	7	3	7
3	2	3	1	1	7	4	5
3	2	3	1	1	7	4	6
3	2	3	1	1	8	4	5
3	2	3	1	1	8	4	6
4	2	3	1	1	6	5	4
4	2	3	1	1	6	5	5
4	2	3	1	1	6	6	4
4	2	3	1	1	6	6	5
5	2	3	1	1	6	7	3
6	2	3	1	1	5	8	3
7	2	3	1	1	4	8	2
7	2	3	1	1	4	8	3
7	2	3	1	1	5	8	2
7	2	3	1	1	5	8	3
8	2	3	1	1	4	8	2
8	2	3	1	1	4	8	3
2	1	3	2	1	3	3	6
2	1	3	2	1	3	4	6
3	1	3	2	1	4	5	5
4	1	3	2	1	4	6	4
5	1	3	2	1	4	7	3
5	1	3	2	1	3	7	3

5	1	3	2	1	4	7	4
6	1	3	2	1	3	8	2
6	1	3	2	1	3	8	3
7	1	3	2	1	3	8	2
7	1	3	2	1	3	9	2
8	1	3	2	1	2	9	2
8	1	3	2	1	1	9	2
8	1	3	2	1	2	9	1
2	1	1	2	3	4	1	8
2	1	1	2	3	4	2	8
2	1	1	2	3	3	2	8
3	1	1	2	3	4	3	7
4	1	1	2	3	5	4	6
4	1	1	2	3	4	4	6
5	1	1	2	3	5	5	5
5	1	1	2	3	5	4	5
6	1	1	2	3	5	6	4
6	1	1	2	3	6	6	4
7	1	1	2	3	5	7	3
7	1	1	2	3	5	8	3
7	1	1	2	3	6	7	3
7	1	1	2	3	6	8	3
8	1	1	2	3	5	8	2
2	2	1	1	3	7	2	8
3	2	1	1	3	9	2	7
3	2	1	1	3	9	2	8
4	2	1	1	3	9	4	5
5	2	1	1	3	8	5	5
5	2	1	1	3	9	5	4
6	2	1	1	3	8	6	4
6	2	1	1	3	8	7	4
7	2	1	1	3	7	7	3
8	2	1	1	3	7	8	2
8	2	1	1	3	6	8	2
8	2	1	1	3	6	8	3
2	1	2	2	2	4	2	7
2	1	2	2	2	4	3	7
3	1	2	2	2	4	4	6
3	1	2	2	2	5	4	6
4	1	2	2	2	5	5	5
5	1	2	2	2	5	6	4
5	1	2	2	2	6	6	4
6	1	2	2	2	5	8	2
6	1	2	2	2	5	8	3
7	1	2	2	2	4	8	2
8	1	2	2	2	3	9	2
8	1	2	2	2	4	9	2
2	2	2	1	2	5	2	7
2	2	2	1	2	5	2	8
3	2	2	1	2	5	4	6
4	2	2	1	2	6	5	5
5	2	2	1	2	6	6	3
5	2	2	1	2	6	7	3
6	2	2	1	2	6	7	3
6	2	2	1	2	6	8	3
7	2	2	1	2	5	8	3
8	2	2	1	2	4	8	2
8	2	2	1	2	5	8	2

Çizelge 4.12’de görüldüğü üzere, gerçek deneylerle, tasarlanan uzman sistemin tahmin ettiği sonuçlar birbirlerine çok yakın olarak elde edilmiştir. Literatür incelendiğinde oluşturulan FL uzman sisteminin performansının iyi olup olmadığını anlamak için gerçek deney verileri ile uzman sistemin tahmin ettiği veriler arasında doğrusal regresyon analizi yapıldığı görülmüştür (Özcan ve ark., 2009; Ozgan, 2009; Özgan, 2010; Metin ve ark., 2012; Gencel ve ark., 2013; Serin ve ark., 2013; Subaşı ve

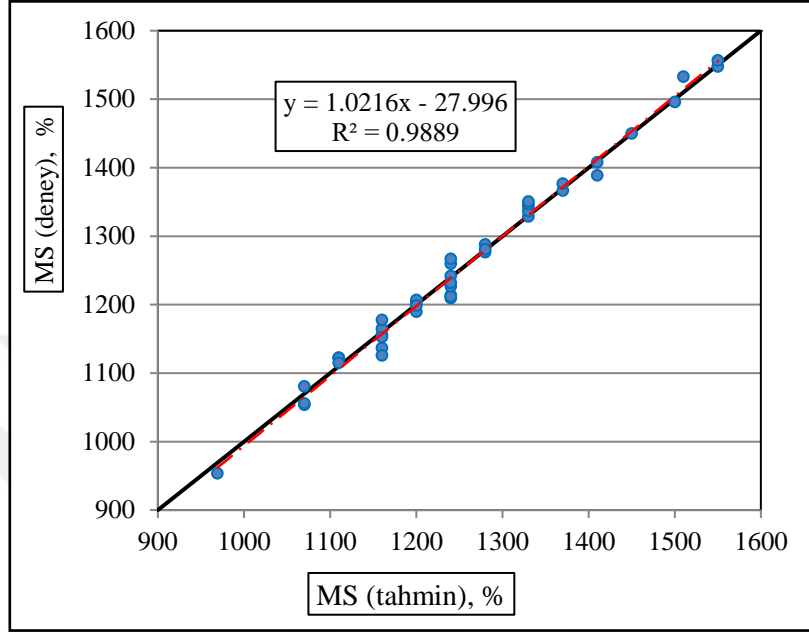
ark., 2013; Beycioglu ve ark., 2017; Kara ve Karacasu, 2017; Zehtabchi ve ark., 2018; Gilani ve ark., 2020).

Çizelge 4. 12. Sonuçların karşılaştırılması

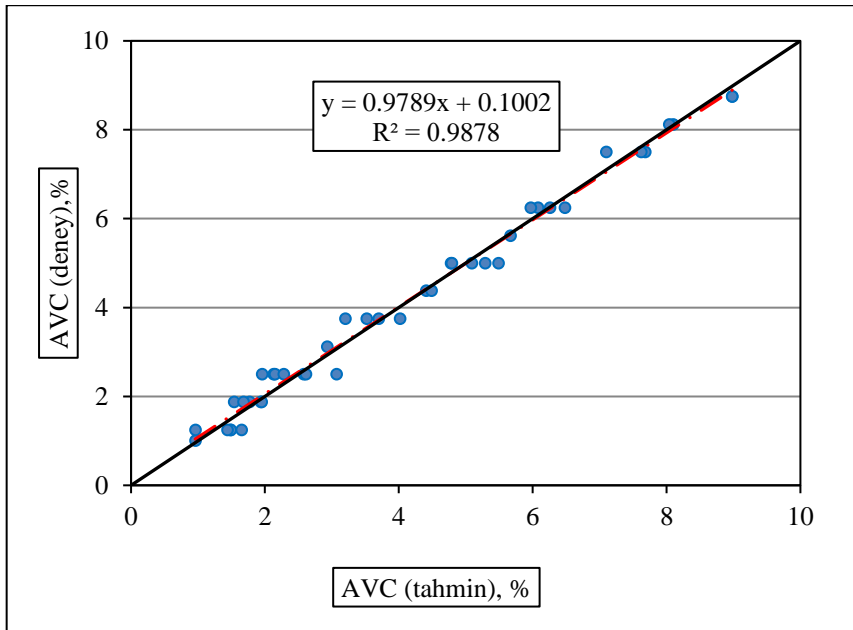
Girdiler					Deney sonuçları			Tahmin sonuçları		
BR (%)	NC (%)	NF (%)	MC (%)	RF (%)	MS (kg)	AVC (%)	VFA (%)	MS (kg)	AVC (%)	VFA (%)
3.5	100	100	-	-	1367	7.68	48.34	1370	7.50	50.0
4.0	100	100	-	-	1450	5.67	59.61	1450	5.62	57.5
4.5	100	100	-	-	1343	4.41	68.32	1330	4.38	68.7
5.0	100	100	-	-	1329	2.58	80.59	1330	2.50	80.0
5.5	100	100	-	-	1260	2.13	84.66	1240	2.50	87.5
6.0	100	100	-	-	1204	1.77	87.80	1200	1.88	87.5
6.5	100	100	-	-	1156	1.54	89.89	1160	1.88	87.5
3.5	-	100	100	-	1054	6.48	53.05	1070	6.25	53.7
4.0	-	100	100	-	1137	5.09	62.44	1160	5.00	65.0
4.5	-	100	100	-	1126	3.70	72.21	1160	3.75	72.5
5.0	-	100	100	-	1123	2.93	78.52	1110	3.12	80.0
5.5	-	100	100	-	1081	1.94	85.90	1070	1.88	87.5
6.0	-	100	100	-	1056	1.49	89.60	1070	1.25	89.0
6.5	-	100	100	-	954	0.96	93.53	969	1.01	92.7
3.5	-	50	100	50	1122	8.98	39.07	1110	8.75	41.0
4.0	-	50	100	50	1165	7.10	49.26	1160	7.50	50.0
4.5	-	50	100	50	1207	6.26	55.96	1200	6.25	57.5
5.0	-	50	100	50	1242	5.49	62.11	1240	5.00	61.3
5.5	-	50	100	50	1277	4.02	71.53	1280	3.75	72.5
6.0	-	50	100	50	1288	2.15	84.00	1280	2.50	83.7
6.5	-	50	100	50	1214	1.65	88.16	1240	1.25	87.5
3.5	100	50	-	50	1408	8.98	38.50	1410	8.75	42.5
4.0	100	50	-	50	1548	8.10	45.21	1550	8.12	42.5
4.5	100	50	-	50	1557	5.29	59.86	1550	5.00	57.5
5.0	100	50	-	50	1533	4.49	66.61	1510	4.38	65.0
5.5	100	50	-	50	1496	3.20	75.89	1500	3.75	76.3
6.0	100	50	-	50	1389	2.61	80.90	1410	2.50	80.0
6.5	100	50	-	50	1377	1.68	87.88	1370	1.88	87.5
3.5	-	75	100	25	1153	7.62	46.11	1160	7.50	46.3
4.0	-	75	100	25	1190	6.08	55.71	1200	6.25	57.5
4.5	-	75	100	25	1210	4.78	64.71	1240	5.00	65.0
5.0	-	75	100	25	1281	3.52	73.75	1280	3.75	72.5
5.5	-	75	100	25	1267	1.95	85.02	1240	1.88	87.5
6.0	-	75	100	25	1178	1.49	89.00	1160	1.25	87.5
6.5	-	75	100	25	1115	0.96	93.14	1110	1.25	92.7
3.5	100	75	-	25	1212	8.04	44.26	1240	8.12	42.5
4.0	100	75	-	25	1227	5.97	55.87	1240	6.25	57.5
4.5	100	75	-	25	1337	4.79	64.41	1330	5.00	65.0
5.0	100	75	-	25	1347	3.07	76.22	1330	2.50	76.3
5.5	100	75	-	25	1351	2.28	82.69	1330	2.50	83.7
6.0	100	75	-	25	1232	1.96	85.80	1240	2.50	87.5
6.5	100	75	-	25	1199	1.44	89.98	1200	1.25	87.5

Şekil 4.22-Şekil 4.24'te, FL modelinden elde edilen tahmin verileri ile deneysel çalışmadan elde edilen veriler arasındaki regresyon ilişkileri gösterilmiştir.  $R^2$  değeri, deneysel ve tahmin edilen veriler arasındaki ilişkiyi belirtmek için kullanılır ve

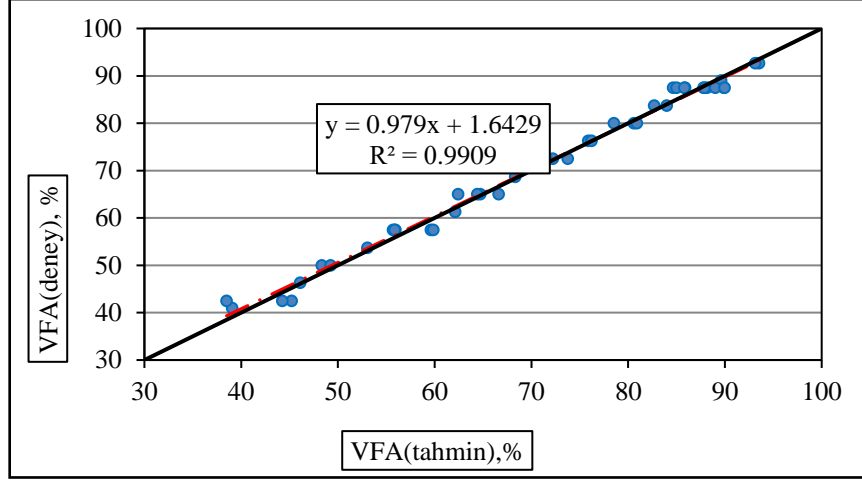
maksimum değeri bir olur. Grafiklerden, *MS* (%98.89), *AVC* (%98.78) ve *VFA* (%99.09) için  $R^2$  değerlerinin oldukça yüksek olduğu görülmektedir. Bu bilgilere dayanarak, bu makalede sunulan 82 kurala dayalı FL uzman sisteminin yüksek doğrulukta tahminler yapabileceği söylenebilir.



Şekil 4. 22. MS için deneysel ve tahmini veriler arasındaki regresyon ilişkileri



Şekil 4. 23. AVC için deneysel ve tahmini veriler arasındaki regresyon ilişkileri



Şekil 4. 24. VFA için deneysel ve tahmini veriler arasındaki regresyon ilişkileri

#### 4.2.5. D-MOGA algoritması

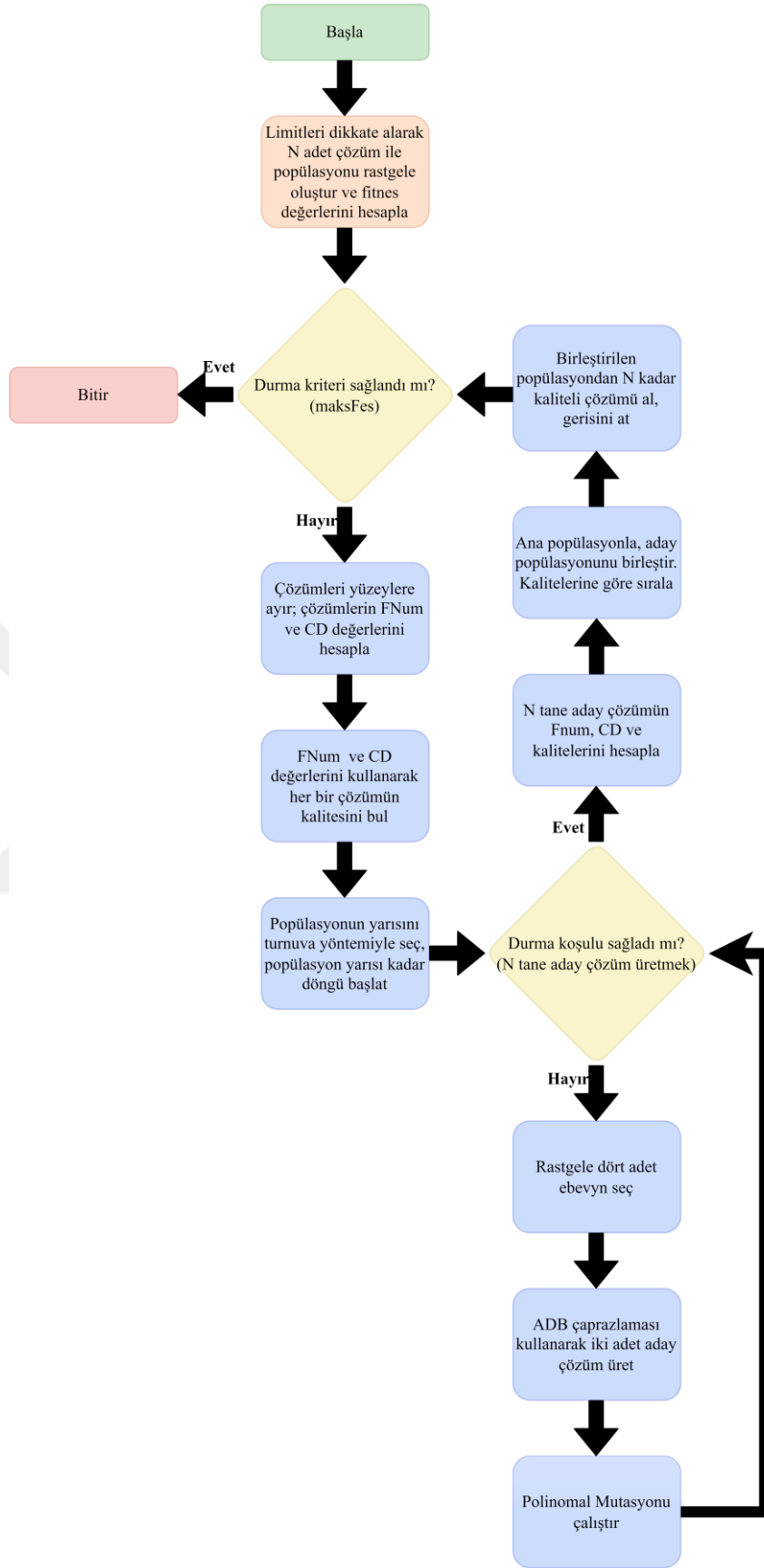
Çok amaçlı optimizasyon problemlerini çözmek için pek çok algoritma önerilmesine rağmen, NFL teoremi her algoritmanın her probleme tam olarak başarılı bir çözüm sunamayacağını belirtmektedir. Bu nedenle, her zaman yeni algoritmalara ihtiyaç duyulmaktadır. Bu bağlamda, asfalt karışımları için Pareto-optimal yüzeylere ulaşmak amacıyla önerilen D-MOGA, literatürdeki algoritmaların eksikliklerini gidermek amacıyla farklı çaprazlama yöntemleri kullanarak ve arama uzayındaki çeşitliliği artırmak için ekstra rastgelelik ile desteklenmiş bir çaprazlama yaklaşımı ile tasarlanmıştır.

Bu önerilen algoritma, literatürdeki iyi bilinen algoritmalarla karşılaştırmalı değerlendirmelerde test edilmiştir ve rekabetçi olduğu kanıtlanmıştır. Asfalt karışımları probleminin yani HMA'nın çözümü için de bu yaklaşım kullanılmıştır. Elde edilen sonuçlar, algoritmanın uzman yardımcısı olarak kullanılabileceğini ve asfalt karışımları için yüksek performanslı çözümler sunabileceğini göstermektedir.

Akış diyagramı Şekil 4.25'te görülen D-MOGA'nın adımları şöyledir:

1. Ana popülasyonu  $P_0$  temsil eden  $N$  adet çözüm, üst ve alt sınırlar dikkate alınarak rastgele oluşturulur.
2.  $P_0$ 'a ait tüm çözümler için amaç fonksiyonu hesaplanır.
3. maksFes sayısı kadar ana döngü başlatılır.
  - a) Ana popülasyondaki her çözüm, "hızlı-bastırılmayan sıralama" yöntemi kullanılarak kalitelerine göre yüzeylere ayrılır ( $FNum$ ). Aynı cephedeki çözümlerin kalitesini ayırmak için ise kalabalıklık mesafesi operatörü kullanılır ( $CD$ ).

- b)  $FNum$  ve  $CD$  kullanılarak, popülasyondaki her çözümün kalitesi yukarıda verilen eşitliğe göre hesaplanır (Denklem 4.3).
- c) Popülasyonun daha kaliteli yarısını seçmek ve bir aday havuzu oluşturmak için turnuva seçim yöntemi kullanılır. Bu seçim çözümlerin kalitelerine göre yapılır.
- d) Popülasyonun yarısı kadar bir döngü daha başlatılır.
- I. Kaliteli aday havuzundan iki adet farklı çözüm seçilir. Aday havuzunun sıradaki çözümü ile ADB çaprazlama operatörüne aday çözümünü oluşturmak için gönderilir. Bu işlem her döngüde iki defa gerçekleştirilir, yani bir döngüde **2 aday çözüm** oluşturulur.
  - II. Belirli bir ihtimalle (0.001) mutasyon yapılabilir.
  - III.  $N$  tane aday çözüm üretilerek  $A_0$  popülasyonu oluşturuldu mu? Hayırsa, 3.d.I. adıma git. Evetse, devam et.
- e) Ana popülasyon  $P_0$  ve aday popülasyonu  $A_0$  birleştirilir. Bu yeni büyük popülasyon  $B_0 = P_0 + A_0$  "hızlı-bastırılmayan sıralama operatörü" ile yüzeylere ayrılır.
- f) Büyük popülasyondaki ( $B_0$ ) çözümler,  $FNum$  ve  $CD$  değerlerine göre iyiden kötüye doğru sıralanır.
- g) En iyi kaliteye sahip  $N$  çözüm, yeni ana popülasyon olur.  $B_0$ 'ın geri kalan kısmı atılır.
4. Sonlandırma kriteri ( $maksFes$ ) karşılanmadıysa 3.a. adıma gidilir. Sonlandırma kriteri karşılandığında algoritma durur ve sonuçta bastırılmayan çözümlerden oluşan Pareto optimal çözüm kümesi elde edilir.



Şekil 4. 25. D-MOGA akış diyagramı

#### 4.2.5.1. Seçim

D-MOGA’da, seçim mekanizması iki ayrı yerde kullanılır. İlk mekanizmada, turnuva seçim yöntemiyle popülasyonun yüksek nitelikli yarısı seçilmektedir. Bu seçim işleminde turnuva seçim sisteminin çalışması şöyledir:

- Turnuva boyutu ya da derinliği 10 olarak belirlenir.
- Popülasyondan bu boyutta rastgele örneklem seçilir.
- Bu örneklem arasında en iyi performans gösteren birey seçilir.
- Buradaki performansta çözümlerin kaliteleri dikkate alınır.

İkinci seçim ise klasik şekilde ilk seçim aşamasında elde edilen popülasyonun kaliteli yarısından çaprazlama için ebeveyn seçimidir. Farklı olarak her jenerasyonda iki aday çözüm üretilir. Her aday birey için iki ebeveyn olmak üzere toplam dört ebeveyn seçilir.

#### 4.2.5.2. Kalite

Bu bölümde, çözümlerin kalitesini hesaplamak için F-MODE tanıtılırken önerilen,  $FNum$  ve  $CD$  kullanılarak kalitenin hesaplandığı (Denklem 4.3) bir yaklaşım, D-MOGA algoritmasında da kullanılmıştır. Şekil 4.6 ise her bir çözümün kalitesinin nasıl hesaplandığını göstermektedir.

#### 4.2.5.3. ADB çaprazlaması

Önerilen çaprazlama yöntemi, ABC, DE ve BLX alfa (ADB) adı verilen üç farklı çaprazlama yöntemini birleştirerek oluşturulmuştur. Bu yeni yöntem, mantıksal rastgele yaklaşımı kullanarak, bu üç yöntemin avantajlarını bir araya getirir. Ancak, ADB çaprazlaması kullanılırken dikkat edilmesi gereken bir önemli husus vardır. İlk aday çözümler üretilirken, ABC, DE ve BLX alfa yöntemlerinden sadece biri rastgele seçilir. Daha sonra, bu seçim sonraki çaprazlama değişkenleri için de aynı şekilde uygulanır. Buna göre, herhangi bir aday çözümde bir karar değişkeni ABC yöntemiyle çaprazlanırken, aynı aday çözümün diğer karar değişkeni DE yöntemiyle çaprazlanırsa, ADB çaprazlaması beklenen sonucu vermeyebilir. Çizelge 4.14’te bu durum test edilmiştir; buna göre Çizelge 4.13’te gösterilen ADB çaprazlamasının sözde kodundaki bir değişken olan  $WR$  değerinin  $for$  satırının içerisine alınması ile bir aday çözümün her bir karar değişkeni rastgele şekilde başka bir eşitlikle çaprazlanmış ve sonuçta  $ZDT$  ve  $DTLZ$  problem setlerine göre;  $WR$  değerinin  $for$  satırının içine alınması durumunda



algoritma %90 daha başarısız olmuştur. Çizelge 4.14'te 30,000 maksFes'te 100 bağımsız çalışma sonuçları HV metriği kapsamında verilmiştir.

Çizelge 4.13'te sözde kodu, verilen ADB çaprazlama işleminin ayrıntılı adımları aşağıdaki gibidir:

1. Çaprazlama işlemi başlamadan önce, üç çaprazlama yönteminden birini seçmek için hangi oran (*Which Rate, WR*) adı verilen sıfır ile bir arasında rastgele rasyonel (*double*) bir sayı oluşturulur.
2. *WR*, 0.4'ten küçükse, Denklem 3.6 kullanılır; 0.4 ile 0.8 arasında ise Denklem 3.4 kullanılır ve 0.8'den büyükse, BLX alfa çaprazlama denklemi kullanılır.
3. Karar değişkeni sayısı kadar bir döngü başlatılır.
4.  $CR = 0.9 - (evaluation/max\_evaluation)$  Yani; değerlendirme sayısına göre hesaplanır *CR* değeri hesaplanır. Buna göre fonksiyon değerlendirmenin başlarında *CR* 1'e yakın bir değer olacak, sonlarda ise 0'a yakın bir değer olacaktır. Bu da başlarda daha fazla oranda çaprazlama yapılması ve arama uzayının daha iyi taranması anlamına gelirken, sonlara doğru zaten doğru sonuca yaklaşıldığından, sadece sonucun etrafı aranması için *CR* oranı son derece küçük değerlere çekilmektedir. *CR* değeri 0.1'in altına düştüğü durumlarda 0.1'e sabitlenir.
5. 0-1 arasında rastgele bir sayı üretilir; bu sayı *CR* değerinden küçükse çaprazlama işlemi yapılır yoksa o karar değişkeni için çaprazlama yapılmaz.
6. Böylece her karar değişkeni için aynı işlemler gerçekleştirilir ve döngünün sonucunda aday bireyin ilk aşaması tamamlanmış olur.
7. Bir sonraki aşamada algoritma, aday çözümün değişkenlerinin üst ve alt limitleri aşıp aşmadığı kontrol edilir. Bir değer, bu limitlerden herhangi birini aşması durumunda, değer hangi limite yakınsa ona çekilir.

Çizelge 4. 13. ADB çaprazlamasının sözde kodu

---

**Define parameters:**  
WR (which rate)=rand(0,1) //0-1 arası rastgele double sayılar üretilir  
n=karar değişkeni sayısı  
CR=0.9-(evaluation/max\_evaluation)\* Buradaki amaç, iterasyonun başında daha fazla çaprazlama uygulayarak daha hızlı yakınsama sağlamak, sonunda ise çaprazlama oranını düşürerek kalite bozulmasını önlemektir.\*/  
if (CR<0.1)  
    CR=0.1  
for i=1:n  
    if (rand(0,1)< CR)  
        if (WR <0.4)  
            Child<sub>ij</sub> = Quality\_solution<sub>ij</sub> + Φ(Quality\_solution<sub>ij</sub> – Random<sub>kj</sub>)  
  
            else if(WR >=0.4&&WR<0.8)  
                Child<sub>ij</sub> = Quality\_solution<sub>ij</sub> + F(Random<sub>kj</sub> – Random<sub>zj</sub>)  
  
        else  
            BLX alpha çaprazlamasını kullan (Denklem 3.8–3.12)  
  
    end if  
    else  
        Sıradaki çözümün i. değerini al  
    end if  
end for  
if Üretilen değer limit dışında ise  
Değeri limitlere çek  
end if

---

Klasik çaprazlama süreçlerinde aday çözüm için aynı çaprazlama operatörü kullanılırken, önerilen yöntemde ana çözümler farklı çaprazlama işlemlerine tabi tutulmaktadır. Bu sayede yeni oluşan çözümlerde tekdüzelik ortadan kaldırılır ve çeşitlilik artırılmış olur.

ABC, DE, BLX-alfa çaprazlamalarının kullanılma ihtimalleri %40, %40, %20 seçilmiş olsa da 10 bağımsız çalıştırmaların ortalama ve standart sapmalarını gösteren Çizelge 4.15 verilerine bakılarak, elde edilen sonuçlar yakın olduğundan bu oranların eşit şekilde dağıtılmasının algoritmanın performansına bir zararı olmayacağı görülmüştür.

Çizelge 4. 14. HV metriği kapsamında WR değerinin farklı durumları

	D-MOGA WR içte		D-MOGA WR dışta	
	Ort.	S.s.	Ort.	S.s.
ZDT1	6.64E-01	7.1E-05	6.65E-01	4.1E-05
ZDT2	3.31E-01	6.8E-05	3.31E-01	4.5E-05
ZDT3	5.16E-01	1.3E-04	5.17E-01	6.9E-05
DTLZ1	7.98E-01	1.7E-03	8.01E-01	1.1E-03
DTLZ2	4.36E-01	2.0E-03	4.38E-01	2.0E-03
DTLZ3	4.56E-03	3.8E-02	4.89E-02	9.7E-02
DTLZ4	4.35E-01	2.1E-03	4.41E-01	2.0E-03
DTLZ5	9.47E-02	3.9E-05	9.47E-02	4.4E-05
DTLZ6	9.46E-02	4.0E-05	9.46E-02	3.5E-05
DTLZ7	3.28E-01	1.2E-03	3.28E-01	1.0E-03

Çizelge 4. 15. HV metriği kapsamında ABC, BLX ve DE çaprazlamalarının karşılaştırılması

	D-MOGA ABC		D-MOGA BLX		D-MOGA DE	
	Ort.	S.s.	Ort.	S.s.	Ort.	S.s.
ZDT1	6.65E-01	5.4E-05	6.64E-01	8.7E-05	6.64E-01	1.9E-04
ZDT2	3.31E-01	6.9E-05	3.31E-01	6.2E-05	3.30E-01	1.9E-04
ZDT3	5.17E-01	8.0E-05	5.17E-01	1.4E-04	5.16E-01	3.7E-04
DTLZ1	2.62E-01	3.5E-01	3.11E-01	3.1E-01	7.91E-01	1.8E-03
DTLZ2	4.41E-01	2.7E-03	4.49E-01	1.7E-03	4.36E-01	2.1E-03
DTLZ3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.92E-02	8.8E-02
DTLZ4	4.40E-01	2.6E-03	4.50E-01	1.9E-03	4.34E-01	2.0E-03
DTLZ5	9.48E-02	5.7E-05	9.48E-02	8.0E-05	9.45E-02	9.0E-05
DTLZ6	9.48E-02	2.5E-05	9.48E-02	3.3E-05	9.48E-02	3.9E-05
DTLZ7	3.34E-01	7.2E-04	3.33E-01	1.3E-03	3.24E-01	1.3E-03

#### 4.2.5.4. Deneyler

Bu bölümde D-MOGA'nın performansının hangi ortamlarda test edildiği, nasıl ve hangi tür kıyaslamalar yapıldığı hakkında bilgi verilmiştir. D-MOGA bu alanda iyi bilinen 4 algoritma ile 21 test fonksiyonu üzerinde yapılan deneylerle ve karşılaştırmalarla gösterilmiştir. İlgili deneyler *Intel® Core™ i7-3630QM CPU@2.40GHz* işlemciye sahip ve *12GB RAM*'i bulunan bir bilgisayarda gerçekleştirilmiştir. Bunun yanı sıra D-MOGA algoritması da çalışmanın ilk kısmıyla birlikte, *JMETAL 4.5* çok amaçlı yazılım paketi üzerine eklenmiştir ve performans karşılaştırmaları ilgili paket dahilinde yapılmıştır (Durillo ve Nebro, 2011).

D-MOGA'nın karşılaştırıldığı algoritmalar; çok amaçlı optimizasyon alanında literatürde sık geçen NSGA-II, yapay alg algoritması AAA'nın başarılı bir, çok amaçlı versiyonu olan MOAAA, ayrıştırma tabanlı MOEA'ları temsilen ve yine onunla alakalı literatürde çok çalışma olan MOEA/D ve son olarak MOEA/D'in yeni ve Pareto taban ile güncel hibrit versiyonu olan MOEA/D-TS ile karşılaştırılmıştır.

Bu kıyaslama işlemleri ZDT ailesinden ilk üçü, DTLZ ailesinin tamamı, WFG ailesinin tamamı Kursawe ve Schaffer olmak üzere toplam 21 adet test fonksiyonu üzerinde gerçekleştirilmiştir. Performansları karşılaştırmak için HV ve GD metrikleri kullanılmıştır.

Karşılaştırma işlemlerinde Çizelge 4.16'te görüldüğü gibi 100 bağımsız çalıştırma yapılmıştır. Bütün algoritmaların popülasyon büyüklüğü 300 olarak ayarlanmıştır ve maksFes 30.000 olarak belirlenmiştir.

Çizelge 4. 16. D-MOGA ve diğer algoritmaların parametreleri

	D-MOGA	MOEA/D-TS	MOEA/D	MOAAA	NSGA-II
<b>Bağımsız çalışma</b>	100	100	100	100	100
<b>Popülasyon sayısı</b>	300	300	300	300	300
<b>MaksFes</b>	30,000	30,000	30,000	30,000	30,000
<b>Çaprazlama</b>	ADB	SBX	DE	AAA	SBX
<b>CR</b>	<i>Dinamik CR</i>	0.5	0.5	-	0.9

Çizelge 4.17'deki daha koyu hücreler, diğer bölgelerden daha iyi sonuçları ifade etmektedir. Bu tabloda yer alan ortalama ve standart sapma değerleri, 100 bağımsız çalışma sonucunda elde edilmiştir. HV metriği, diğer metriklerden farklı olarak daha büyük bir sayının daha iyi sonuçları temsil ettiği bir metriktir. Eşit ortalama sonuçları için standart sapmalar dikkate alınarak, daha düşük standart sapmalı değerler daha kararlı olduğu için daha başarılı olarak kabul edilmiştir. Önerilen D-MOGA, HV metriği temel alındığında, diğer algoritmalara göre daha iyi bir performans sergilemiştir. Yapılan testler sonucunda D-MOGA yaklaşımı, 21 problemin 18'inde en iyi sonucu elde ederek rekabetçi ve başarılı bir algoritma olduğunu kanıtlamıştır.

Çizelge 4. 17. HV metriğine göre algoritmaların performansı

HV	NSGA-II		MOAAA		MOEA/D		MOEA/D-TS		D-MOGA (Ö.A.)	
	Ort.	S.s.	Ort.	S.s.	Ort.	S.s.	Ort.	S.s.	Ort.	S.s.
ZDT1	6.58E-01	7.8E-04	6.58E-01	2.3E-03	6.50E-01	2.8E-03	6.63E-01	2.6E-04	6.65E-01	4.1E-05
ZDT2	3.21E-01	1.3E-03	3.21E-01	4.2E-03	3.13E-01	3.3E-03	3.29E-01	2.4E-04	3.31E-01	4.6E-05
ZDT3	5.11E-01	7.5E-04	5.11E-01	1.9E-03	5.05E-01	2.6E-03	5.16E-01	1.1E-04	5.17E-01	8.7E-05
DTLZ1	2.89E-02	7.8E-02	1.17E-01	2.2E-01	5.74E-01	2.7E-01	7.86E-01	7.6E-03	8.01E-01	1.1E-03
DTLZ2	4.16E-01	2.5E-03	4.15E-01	2.6E-03	4.22E-01	1.3E-03	4.18E-01	3.1E-03	4.29E-01	2.2E-03
DTLZ3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.71E-01	1.7E-01	6.25E-02	1.2E-01
DTLZ4	4.30E-01	1.9E-03	4.26E-01	2.1E-03	4.37E-01	1.1E-03	4.10E-01	6.3E-02	4.38E-01	2.0E-03
DTLZ5	9.45E-02	6.4E-05	9.41E-02	1.2E-04	9.30E-02	3.9E-05	9.39E-02	7.0E-04	9.47E-02	4.6E-05
DTLZ6	0.00E+00	0.00E+00	9.46E-02	3.3E-05	9.33E-02	1.4E-05	4.73E-02	1.7E-02	9.46E-02	3.9E-05
DTLZ7	2.97E-01	3.1E-03	3.16E-01	2.3E-03	2.57E-01	4.4E-03	2.71E-01	2.4E-03	3.26E-01	1.0E-03
WFG1	4.81E-01	5.8E-02	4.14E-01	3.7E-02	3.32E-01	4.0E-02	5.00E-01	8.0E-02	6.26E-01	9.6E-03
WFG2	5.57E-01	2.0E-03	5.57E-01	2.6E-04	5.57E-01	1.4E-04	5.55E-01	2.5E-03	5.58E-01	2.9E-05
WFG3	4.97E-01	3.8E-04	4.97E-01	1.3E-04	4.97E-01	8.6E-05	4.97E-01	3.5E-04	4.98E-01	4.6E-05
WFG4	2.13E-01	1.6E-04	2.12E-01	4.4E-04	2.09E-01	8.0E-04	2.13E-01	1.5E-04	2.13E-01	1.3E-04
WFG5	2.16E-01	1.2E-03	2.17E-01	3.5E-03	2.16E-01	1.4E-03	2.16E-01	4.8E-05	2.17E-01	3.6E-03
WFG6	2.09E-01	3.6E-03	2.12E-01	2.6E-04	2.12E-01	1.0E-04	2.02E-01	1.2E-02	2.13E-01	7.6E-05
WFG7	2.13E-01	8.8E-05	2.12E-01	6.5E-05	2.12E-01	5.1E-05	2.13E-01	8.9E-05	2.13E-01	4.6E-05
WFG8	2.22E-01	1.9E-02	2.08E-01	1.0E-02	2.37E-01	2.6E-02	2.20E-01	1.8E-02	2.02E-01	2.8E-03
WFG9	2.25E-01	7.5E-04	2.25E-01	6.1E-04	2.23E-01	4.4E-04	2.25E-01	1.2E-03	2.26E-01	4.5E-04
Kursawe	4.04E-01	9.1E-05	4.04E-01	1.4E-04	4.04E-01	6.9E-05	4.04E-01	9.9E-05	4.05E-01	4.4E-05
Schaffer	5.31E-01	2.0E-01	5.76E-01	1.5E-01	5.72E-01	2.0E-01	5.36E-01	2.2E-01	8.32E-01	7.1E-05

GD metriği, bir algoritmanın hesapladığı Pareto cephesinin gerçek Pareto cephesine olan uzaklığını ölçer. Tüm popülasyon için hesaplanan bu metrik, popülasyonun tüm değerleri toplanarak popülasyon büyüklüğüne bölünür ve sonuç olarak nihai GD değeri elde edilir. Araştırmacılar tarafından sıklıkla kullanılan GD metriği, gerçekçi bir performans değerlendirme yöntemi olduğu için tercih edilir. GD metriği ortalama değer üzerinden hesaplandığından, daha küçük bir değer daha iyi bir sonuç olduğu kabul edilir ve Çizelge 4.18'deki koyu hücreler en iyi sonuçları gösterir. D-MOGA algoritması, 21 kıyaslama işlevi üzerinde yapılan testlerde, ortalama ve standart sapma temel alınarak değerlendirildiğinde 16 fonksiyonda daha başarılı olduğunu kanıtlamıştır. Bu nedenle D-MOGA, GD metriğine göre de başarılı ve rekabetçi bir algoritmadır denilebilir.

Çizelge 4. 18. GD metriğine göre algoritmaların performansı

GD	NSGA-II		MOAAA		MOEA/D		MOEA/D-TS		D-MOGA (Ö.A.)	
	Ort.	S.s.	Ort.	S.s.	Ort.	S.s.	Ort.	S.s.	Ort.	S.s.
ZDT1	2.88E-04	3.4E-05	2.89E-04	1.1E-04	5.85E-04	1.1E-04	9.12E-05	1.4E-05	1.13E-05	2.0E-06
ZDT2	4.38E-04	5.6E-05	5.06E-04	2.7E-04	7.50E-04	1.3E-04	8.36E-05	1.3E-05	1.18E-05	2.2E-06
ZDT3	1.47E-04	1.8E-05	2.33E-04	7.5E-05	3.58E-04	8.0E-05	3.80E-05	1.1E-04	1.61E-05	2.4E-06
DTLZ1	1.31E+00	1.20E+00	1.86E-01	1.5E-01	5.60E-02	6.2E-02	1.45E-02	5.1E-02	1.39E-04	4.9E-05
DTLZ2	5.01E-04	5.6E-05	7.45E-04	5.1E-05	1.52E-04	1.3E-05	4.36E-04	9.5E-05	7.02E-05	1.5E-05
DTLZ3	1.11E+01	2.90E+00	6.14E+00	2.20E+00	3.19E+00	7.1E-01	6.11E-02	5.5E-02	2.38E-01	2.3E-01
DTLZ4	4.66E-04	4.0E-05	7.41E-04	6.7E-05	2.04E-04	2.4E-05	4.12E-04	1.8E-04	9.06E-05	1.7E-05
DTLZ5	8.48E-05	6.3E-05	1.45E-04	2.0E-05	7.16E-05	8.2E-06	3.60E-05	3.0E-05	3.30E-06	8.1E-07
DTLZ6	1.94E-01	7.6E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.43E-02	5.7E-03	0.00E+00	0.00E+00
DTLZ7	1.44E-03	1.3E-04	7.98E-04	8.1E-05	7.14E-04	1.2E-04	1.61E-04	1.8E-04	5.31E-04	8.1E-05
WFG1	5.83E-04	1.3E-03	1.24E-02	3.9E-03	1.23E-02	2.2E-03	1.01E-03	2.2E-03	3.69E-05	3.6E-04
WFG2	2.93E-05	1.3E-05	5.93E-05	1.1E-05	2.81E-05	5.4E-06	2.79E-05	1.7E-05	9.40E-06	1.3E-06
WFG3	4.73E-05	1.5E-05	5.15E-05	5.4E-06	2.98E-05	3.0E-06	4.67E-05	1.3E-05	5.47E-06	1.3E-06
WFG4	3.06E-05	5.7E-06	7.82E-05	1.9E-05	1.39E-04	3.2E-05	2.84E-05	5.5E-06	1.39E-05	4.4E-06
WFG5	6.73E-04	4.6E-05	6.16E-04	1.3E-04	6.90E-04	5.7E-05	6.78E-04	2.8E-06	6.02E-04	1.4E-04
WFG6	1.74E-04	1.5E-04	5.38E-05	1.1E-05	2.48E-05	3.5E-06	4.88E-04	5.6E-04	9.37E-06	2.3E-06
WFG7	2.42E-05	3.8E-06	3.18E-05	3.9E-06	1.96E-05	2.0E-06	2.30E-05	4.0E-06	2.39E-06	6.4E-07
WFG8	5.20E-03	2.9E-03	8.94E-03	3.3E-03	5.13E-03	4.3E-03	6.76E-03	3.3E-03	1.29E-02	1.7E-03
WFG9	6.16E-05	2.7E-05	8.94E-05	2.3E-05	1.23E-04	1.7E-05	8.42E-05	4.4E-05	5.75E-05	1.5E-05
Kursawe	4.13E-05	3.8E-06	6.28E-05	8.0E-06	2.23E-05	4.8E-06	3.96E-05	3.9E-06	3.15E-05	3.0E-06
Schaffer	9.54E-02	1.4E-01	1.33E-01	3.6E-01	5.69E+01	8.30E+00	9.45E-02	2.0E-01	3.23E-07	3.6E-07

D-MOGA algoritması Çizelge 4.19- Çizelge 4.20'ye göre, diğer algoritmalarla karşılaştırıldığında istatistiksel olarak anlamlı bir farklılık göstermektedir. Bu farklılığı belirlemek için, rakip algoritmaların sonuçlarına Wilcoxon sıra toplamı testi uygulanarak p değerleri hesaplanmıştır.

Eğer p değeri 0.05'ten küçükse (veya %5'ten azsa), "+" sembolü ile gösterilir ve bunun anlamı; D-MOGA algoritması diğer algoritmalarla karşılaştırıldığında anlamlı

sonular retir demektir. Tersine p deęeri 0.05'ten bykse "-" deęeri, herhangi bir nedenden dolayı hesaplanamamıřsa "?" ile belirtilmiřtir.

Bu izelgelere gre D-MOGA: HV metrięi kapsamında sadece NSGA-II ile WFG9 probleminde benzer sonu retmiřtir. GD metrięi kapsamında ise; MOEA/D-TS ile WFG1 ve NSGA-II ile WFG1 ve WFG9 problemlerinde benzerlik gstermiřtir.

Bu istatistiksel veriler, D-MOGA algoritmasının sonularının tesadfi deęil, istatistiksel olarak anlamlı olduęunu gstermektedir.

**izelge 4. 19.** HV metrięine gre Wilcoxon sıralı toplam testi

D-MOGA vs	HV METRİęİNE GRE SIRA TOPLAMI TESTİ							
	MOEA/D-TS		MOEA/D		MOAAA		NSGA-II	
	p-deęeri	iřaret	p-deęeri	iřaret	p-deęeri	iřaret	p-deęeri	iřaret
ZDT1	2.56E-34	+	2.56E-34	+	2.56E-34	+	2.56E-34	+
ZDT2	2.56E-34	+	2.56E-34	+	2.56E-34	+	2.56E-34	+
ZDT3	2.56E-34	+	2.56E-34	+	2.56E-34	+	2.56E-34	+
DTLZ1	2.56E-34	+	2.56E-34	+	3.19E-35	+	1.85E-36	+
DTLZ2	5.42E-34	+	3.85E-33	+	2.56E-34	+	2.56E-34	+
DTLZ3	2.89E-05	+	2.68E-12	+	2.68E-12	+	2.68E-12	+
DTLZ4	1.05E-32	+	3.05E-13	+	2.56E-34	+	4.67E-34	+
DTLZ5	1.62E-32	+	2.56E-34	+	2.56E-34	+	2.64E-34	+
DTLZ6	2.56E-34	+	2.56E-34	+	1.09E-03	+	5.64E-39	+
DTLZ7	2.56E-34	+	2.56E-34	+	2.56E-34	+	2.56E-34	+
WFG1	7.53E-34	+	2.56E-34	+	2.56E-34	+	5.42E-34	+
WFG2	2.56E-34	+	2.56E-34	+	2.56E-34	+	8.23E-34	+
WFG3	2.56E-34	+	2.56E-34	+	2.56E-34	+	2.56E-34	+
WFG4	7.60E-28	+	2.56E-34	+	2.56E-34	+	2.08E-30	+
WFG5	5.19E-09	+	9.22E-07	+	3.47E-11	+	4.68E-09	+
WFG6	2.56E-34	+	2.56E-34	+	2.56E-34	+	2.72E-34	+
WFG7	4.40E-34	+	2.56E-34	+	2.56E-34	+	3.16E-34	+
WFG8	3.74E-33	+	1.41E-33	+	6.08E-22	+	2.40E-33	+
WFG9	2.51E-07	+	2.56E-34	+	4.50E-18	+	5.10E-01	-
Kursawe	2.56E-34	+	2.56E-34	+	2.56E-34	+	2.56E-34	+
Schaffer	2.54E-34	+	2.56E-34	+	2.56E-34	+	2.56E-34	+

Çizelge 4. 20. GD metriğine göre Wilcoxon sıralı toplam testi

D-MOGA vs	GD METRİĞİNE GÖRE SIRA TOPLAMI TESTİ							
	MOEA/D-TS		MOEA/D		MOAAA		NSGA-II	
	p-değeri	işaret	p-değeri	işaret	p-değeri	işaret	p-değeri	işaret
ZDT1	2.56E-34	+	2.56E-34	+	2.56E-34	+	2.56E-34	+
ZDT2	2.56E-34	+	2.56E-34	+	2.56E-34	+	2.56E-34	+
ZDT3	3.76E-28	+	2.56E-34	+	2.56E-34	+	2.56E-34	+
DTLZ1	3.30E-29	+	2.56E-34	+	2.56E-34	+	2.56E-34	+
DTLZ2	1.52E-30	+	2.56E-34	+	2.56E-34	+	2.56E-34	+
DTLZ3	2.53E-11	+	3.26E-34	+	2.56E-34	+	2.56E-34	+
DTLZ4	1.52E-22	+	2.56E-34	+	2.56E-34	+	2.56E-34	+
DTLZ5	4.65E-24	+	2.56E-34	+	2.56E-34	+	2.56E-34	+
DTLZ6	5.64E-39	+	-	?	-	?	5.64E-39	+
DTLZ7	1.04E-31	+	1.52E-22	+	3.22E-33	+	2.56E-34	+
WFG1	8.46E-01	-	2.56E-34	+	2.56E-34	+	6.24E-01	-
WFG2	4.15E-26	+	2.56E-34	+	2.56E-34	+	7.76E-34	+
WFG3	2.56E-34	+	2.56E-34	+	2.56E-34	+	2.56E-34	+
WFG4	1.13E-31	+	2.56E-34	+	2.56E-34	+	2.55E-33	+
WFG5	2.75E-24	+	5.25E-30	+	6.83E-07	+	2.22E-27	+
WFG6	2.56E-34	+	2.72E-34	+	2.56E-34	+	2.56E-34	+
WFG7	2.56E-34	+	2.56E-34	+	2.56E-34	+	2.56E-34	+
WFG8	2.37E-29	+	1.57E-30	+	1.28E-22	+	7.79E-32	+
WFG9	6.50E-06	+	2.72E-34	+	1.05E-21	+	5.83E-01	-
Kursawe	1.49E-28	+	3.94E-26	+	2.64E-34	+	1.20E-31	+
Schaffer	1.96E-10	+	2.13E-34	+	8.91E-16	+	7.54E-10	+

Şekil 4.26-Şekil 4.30'da D-MOGA ve karşılaştırıldığı diğer algoritmaların, ilgili 21 test fonksiyonu için Pareto-optimal yüzey performansları gösterilmiştir. İlgili problemlerin gerçek yüzeyleri turkuaz renkte gösterilmiştir. NSGA-II'nin tahmin ettiği yüzeyler ise boş yeşil yıldız; MOAAA'nın tahmin ettiği yüzeyler içi boş turuncu üçgen; MOEA/D'nin tahmin ettiği yüzeyler siyah içi boş kare, MOEA/D-TS'nin tahmini mor yıldız ve D-MOGA tarafından tahmin edilen yüzeyler ise kırmızı halkalar ile temsil edilmiştir.

Kullanılan *ZDT* problemlerinin tamamında bütün algoritmalar ideal Pareto-optimal yüzeye çok iyi şekilde yaklaşmış olduğu görülmektedir.

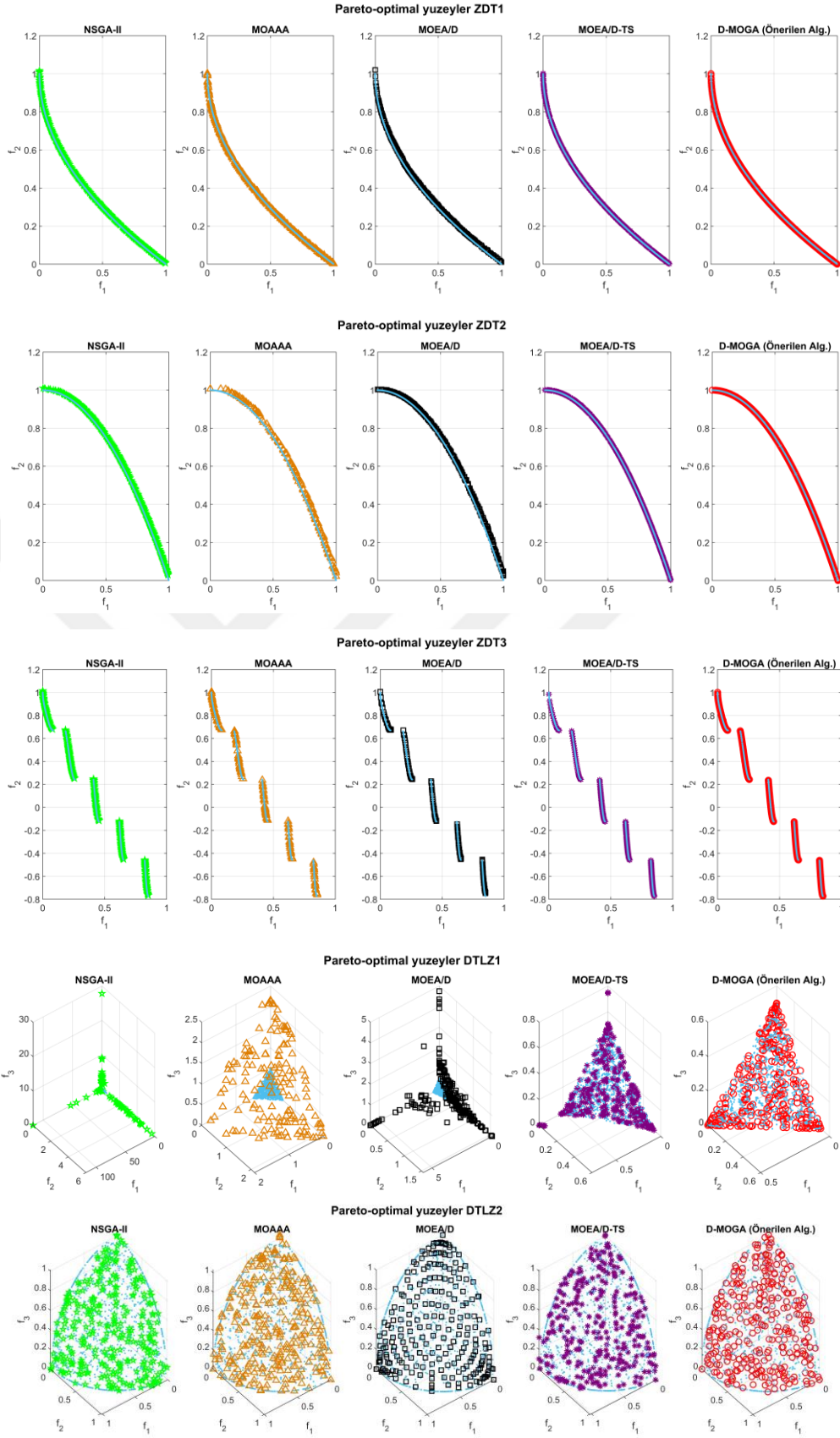
*DTLZ1* probleminde ise en iyi yüzey performansı MOEA/D-TS ve D-MOGA tarafından gösterilmiştir. *DTLZ2* probleminde bütün algoritmalar iyi bir performans göstermişlerdir. *DTLZ3*'te ise sadece MOEA/D-TS gerçek yüzeye biraz yaklaşmıştır. *DTLZ4* probleminde bütün algoritmalar kabul edilebilir şekilde gerçek yüzeylere yaklaşmışlardır. *DTLZ5* probleminde bütün algoritmaların yakınsaması iyi iken, *DTLZ6* probleminde NSGA-II ve MOEA/D-TS kötü performans göstermişlerdir. *DTLZ7* probleminde ise NSGA-II, MOAAA, D-MOGA kabul edilebilir yüzeyler elde etmişlerdir.

*WFG1* probleminde en iyi yüzey D-MOGA'ya aittir. *WFG2* probleminde ise bütün algoritmalar gerçek yüzeye makul bir yakınsama yapmıştır, yalnız MOEA/D ve MOEA/D-TS algoritmalarının yakınsamasında ufak hatalar gözlemlenmektedir. *WFG3-WFG7*'ye kadar bütün algoritmalar iyi yüzeyler elde etmişlerdir. *WFG8*'de ise doğru yüzeye en iyi yaklaşan MOEA/D olsa da yeteri kadar yakınsayamamıştır. *WFG9* probleminde de bütün algoritmalar doğru yakınsamayı başarmıştır.

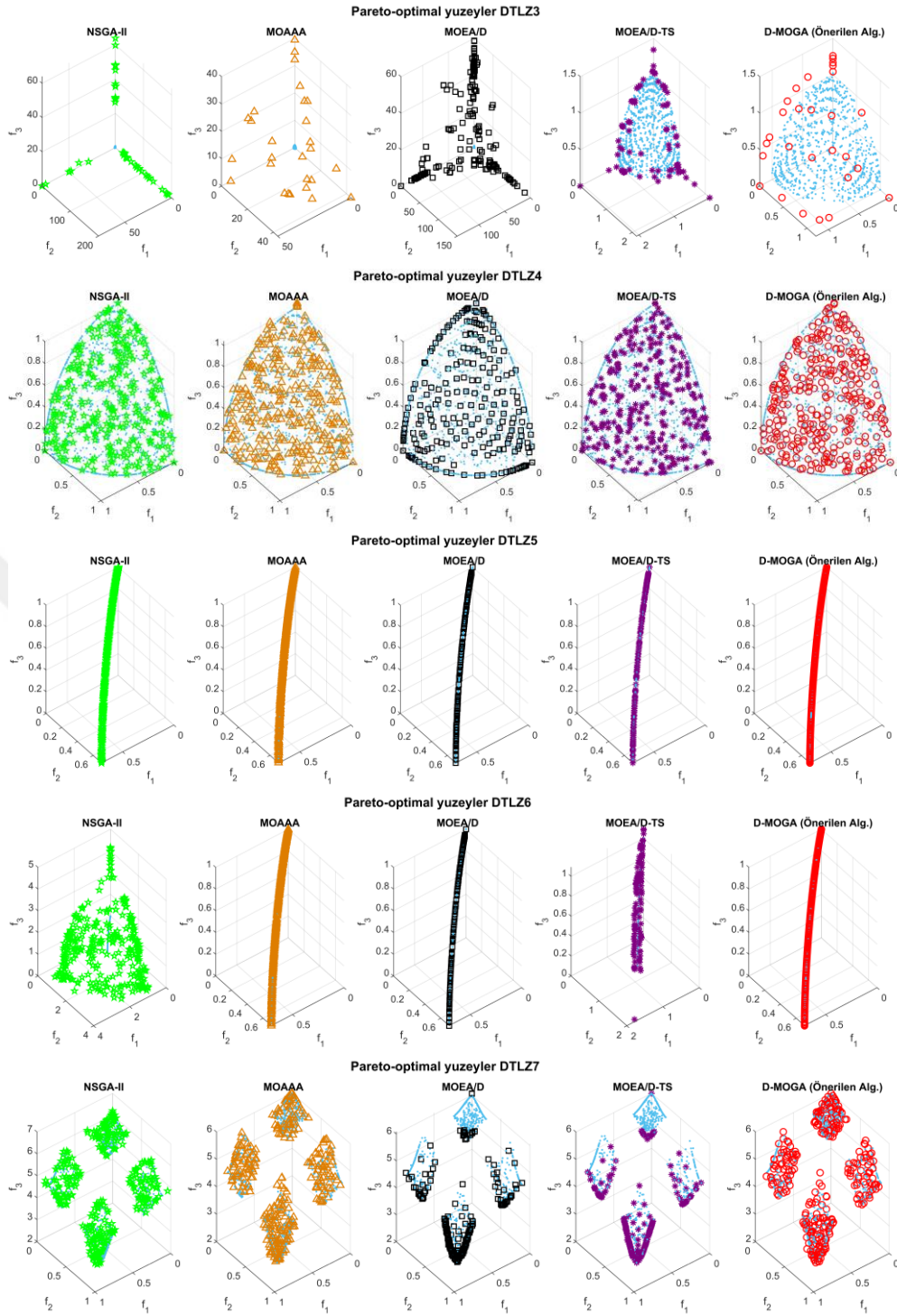
*Kursawe* probleminde bütün algoritmalar ideal yüzeyleri elde etmişlerdir. Ancak *Schaffer* probleminde sadece D-MOGA algoritması doğru bir çizim yakalamıştır.



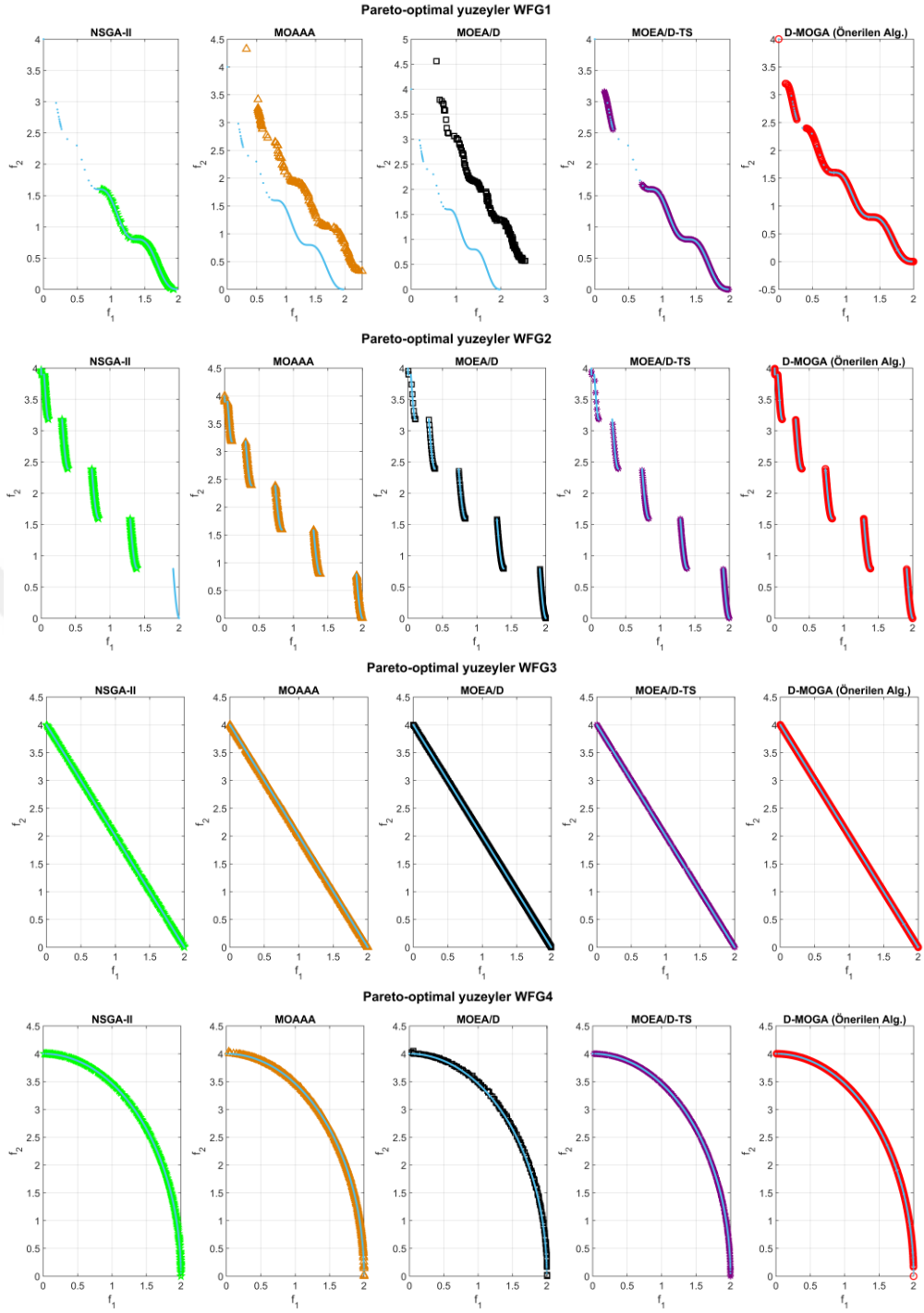




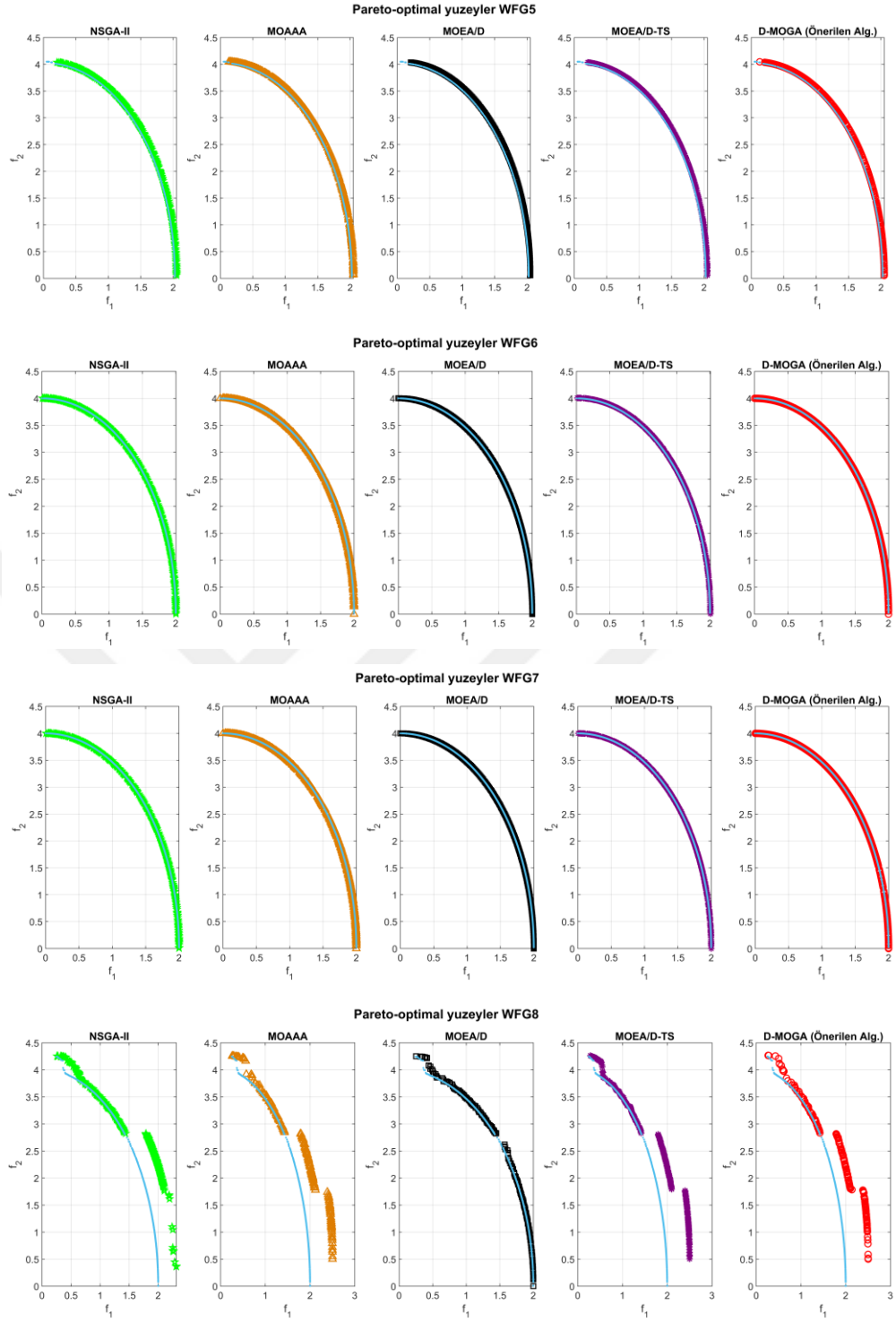
**Şekil 4. 26.** D-MOGA ve diğer algoritmalar tarafından ZDT1-3, DTLZ1-2 problemleri için üretilen Pareto-optimal yüzeyler



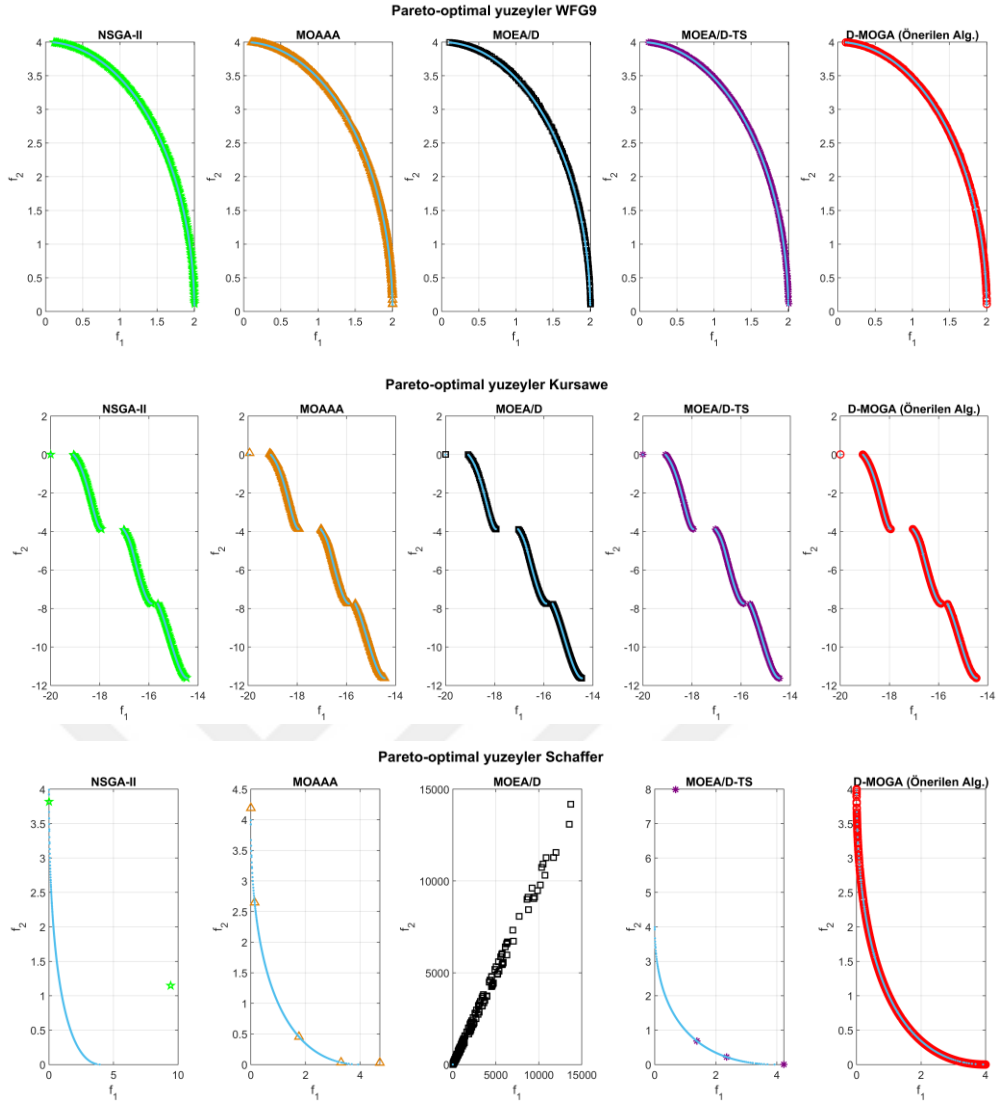
Şekil 4. 27. D-MOGA ve diğer algoritmalar tarafından DTLZ3-7 problemleri için üretilen Pareto-optimal yüzeyler



Şekil 4. 28. D-MOGA ve diğer algoritmalar tarafından WFG1-4 problemleri için üretilen Pareto-optimal yüzeyler



**Şekil 4. 29.** D-MOGA ve diğer algoritmalar tarafından WFG5-8 problemleri için üretilen Pareto-optimal yüzeyler



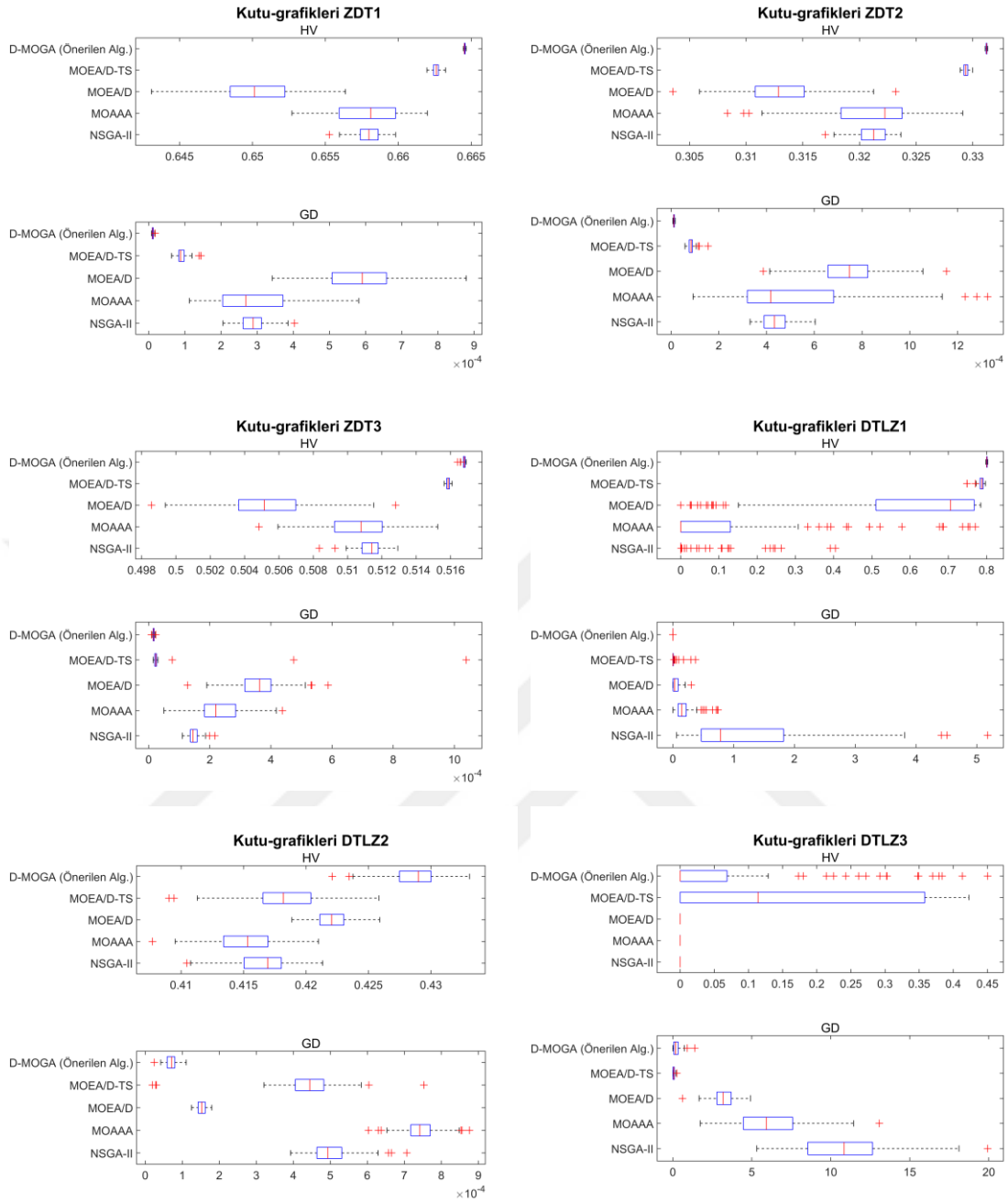
Şekil 4. 30. D-MOGA ve diğer algoritmalar tarafından WFG9, Kursawe, Schaffer problemleri için üretilen Pareto-optimal yüzeyler

Şekil 4.31-Şekil 4.34'te *ZDT* problemlerinde D-MOGA'nın standart sapma değeri çok küçük olduğundan küçük bir kutu-grafiği çizdiği görülmektedir.

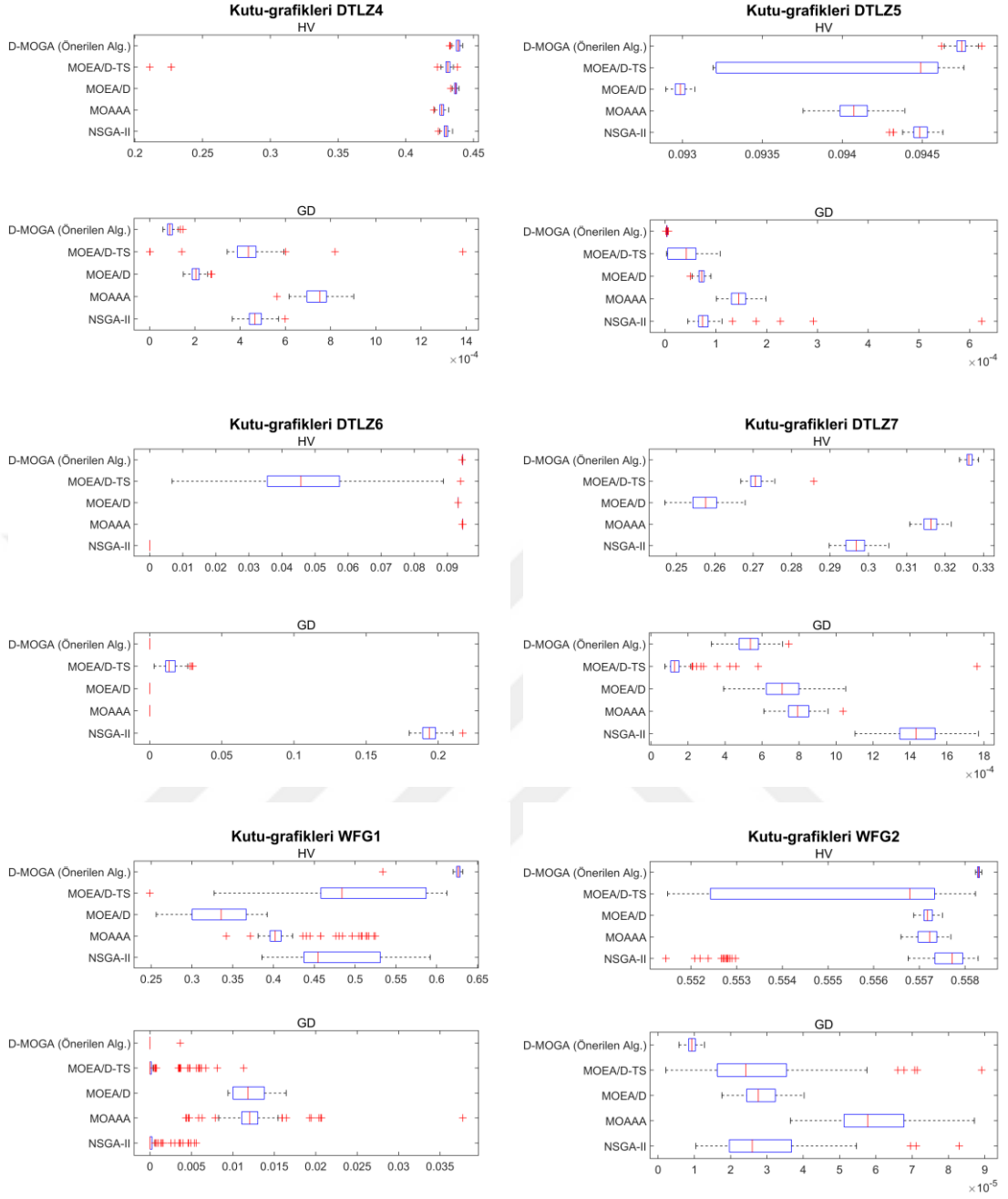
*DTLZ1* probleminde, HV metriğine göre D-MOGA'nın grafiği çok küçük kalmıştır ve aykırı değerlerinin fazlalığına bakarak diğer algoritmaların aramasını bitirmediği görülmektedir. GD metriğinde ise en iyi medyan değerinin D-MOGA'ya ait olduğu görülmektedir ve standart sapması çok küçük olduğundan kutu görünmemektedir. Böyle bir sonuç beklenmektedir çünkü D-MOGA standart sapmada  $4.9E-05$  gibi değerler üretirken, diğer algoritmalar  $1.0E-01$ 'lerde kalmışlardır. *DTLZ2* probleminde HV metriğine göre medyan değerlerine bakıldığında D-MOGA'nın diğer algoritmalara göre daha iyi sonuçlar ürettiği görülmektedir. GD metriğine göre ise, neredeyse bütün kutu grafiklerinin simetrik olması sonuçların kararlı olduğunu göstermektedir. *DTLZ3*

probleminde HV metriğine göre D-MOGA ve MOEA/D-TS dışında diğer algoritmaların sonuç üretilmedikleri görülmektedir. GD metriğine göre ise en azından algoritmaların performansının değerlendirileceği nispeten kutu grafikleri çizilebilmiştir. *DTLZ4*, HV metriğinde olduğu gibi bütün kutu grafiklerinin görülebilir ve aynı boyutlarda olduğu durumlarda algoritmaların yakın sonuçlar ürettiği gözlemlenmektedir. GD metriğinde performanslar arasındaki fark daha belirgin hale gelmiştir. *DTLZ5*, HV metriğinde MOEA/D-TS dışındaki algoritmaların kararlı sonuçlar ürettiği görülmektedir. GD metriğinde ise bütün algoritmalar kararlı sonuçlar üretse de D-MOGA'nın standart sapması diğerlerinden daha küçük görünmektedir. *DTLZ6* probleminde çoğu algoritma grafik çizdirilebilecek bir sonuç üretilmemiştir. Bunun yanında *DTLZ7* probleminde bütün algoritmalar simetrik kutu grafikleri çizdirebilmişlerdir.

*WFG1* probleminde göze çarpan ilk durum, algoritmalar tarafından üretilen çok sayıda aykırı veridir yani bu problem için D-MOGA ve MOEA/D dışındaki algoritmaların aramaları bitmemiştir, bir kararlılığa ulaşamamışlardır. *WFG2*'de HV metriğine göre MOEA/D-TS diğer algoritmalara göre kararsız sonuçlar ürettiği görülmektedir. Bu durum yüzey grafiklerinde de göze çarpmaktadır. *WFG5* ve *WFG8* problemlerinde kararsızlıklar ve aykırı sonuçlar ön plana çıkmaktadır. Diğer *WFG* problemlerinin grafikleri her algoritma için başarılıdır denilebilir. *Kursawe* probleminde bütün algoritmalar yakın değerler üretmiştir, bunun yanında D-MOGA'nın performansı ön plana çıkmıştır. *Schaffer* probleminde ise D-MOGA dışındaki algoritmaların aramalarını tamamlamadıkları anlaşılmaktadır.

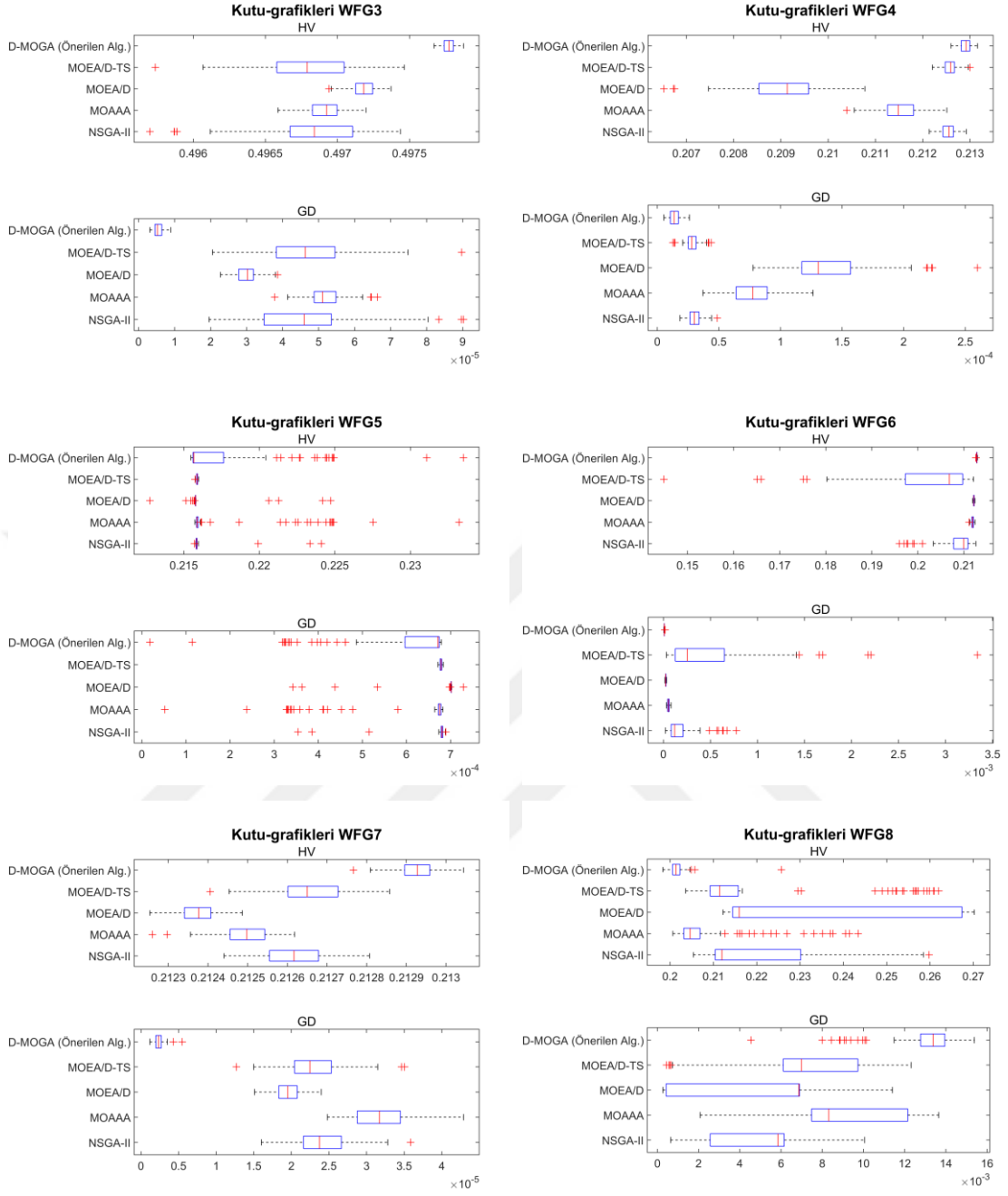


Şekil 4.31. D-MOGA ve diğer algoritmaların ZDT1-3, DTLZ1-3 problemleri için ürettiği performans metriklerinin kutu grafikleri

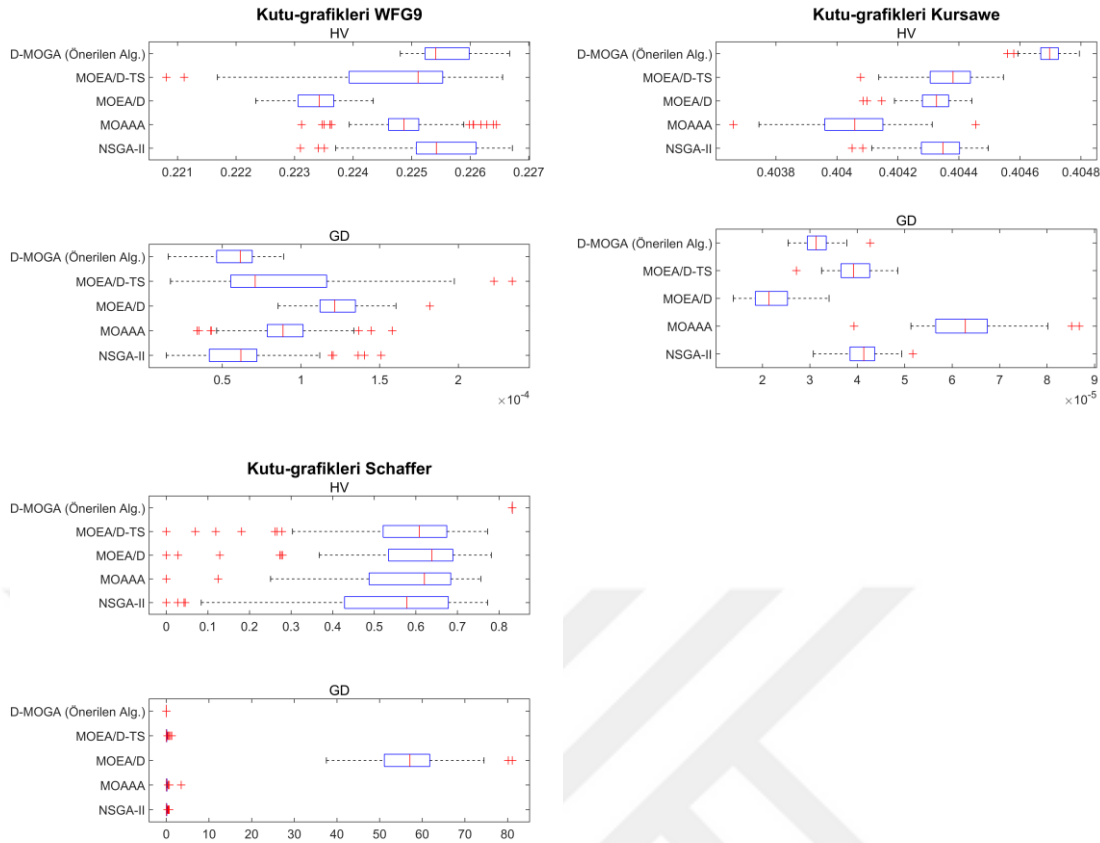


**Şekil 4. 32.** D-MOGA ve diğer algoritmaların DTLZ4-7, WFG1-2 problemleri için ürettiği performans metriklerinin kutu grafikleri





Şekil 4. 33. D-MOGA ve diğer algoritmaların WFG3-8 problemleri için ürettiği performans metriklerinin kutu grafikleri



Şekil 4. 34. D-MOGA ve diğer algoritmaların WFG9, Kursawe, Schaffer problemleri için ürettiği performans metriklerinin kutu grafikleri

#### 4.2.6. Önerilen algoritmanın asfalt karışımı problemine uyarlanması

Bu bölümde D-MOGA algoritması asfalt karışım problemi için üretilen sonuçlar verilmiştir. D-MOGA, HMA problemine uyarlandığından dolayı ismi bu uygulamada MOGA-HMA olarak kullanılmıştır.

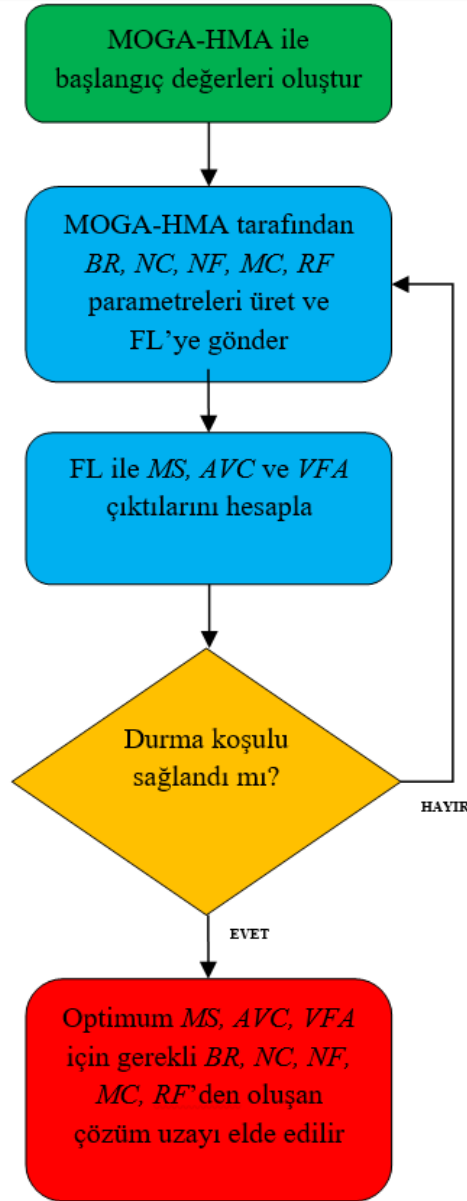
Elde edilecek sonuçların Türk Karayolları Şartnamesine (KGM, 2013) uygun olması için giriş parametreleri belirli aralıklarda ayarlanmıştır. Örneğin, bitümlü sıcak asfalt oranı için tavsiye edilen oran %3.5 ile %6.5 arasındadır ve *NC* ve *MC* toplamlarının toplamı 100 olacak şekilde her biri 0-100 aralığında değerler alacaktır. Benzer şekilde, *NF* ve *RF* toplamı 100 olacak şekilde ayarlanmıştır, ancak *NF* 50–100 aralığında, *RF* ise 0–50 aralığında değerler alabilir (Çizelge 4.21).

Çizelge 4. 21. Asfalt problemi için önerilen girdi ve çıktı değerleri

Girdiler	BR=3.5–6.5	NC=0–100	MC=0–100	NF=50–100	RF=0–50
Çıktılar	MS≥750		AVC=4–6	VFA=60–75	

Böylece girişler şartnamenin yanında gerçek deney verilerine de uygun hale getirilmiştir, çünkü geliştirilen uzman sistemdeki kural tabanları bu gerçek deney sonuçlarından çıkarılmıştır.

Şekil 4.35'te MOGA-HMA'nın daha önce tasarlanan FL uzman sistemi ile hibrit şekilde çalışarak HMA problemine uyarlanmasının akış diyagramı temsili olarak gösterilmiştir.



Şekil 4. 35. MOGA-HMA algoritmasının HMA problemine uyarlanması

Algoritma, Pareto tabanlı optimizasyon algoritmaları kullanılarak tasarlanmıştır. Varsayılan olarak, bu algoritmalar minimizasyon problemlerine göre tasarlanmıştır, ancak bu problemde özel bir durum vardır; çıktılardan biri olan  $MS$  değeri daha büyük olduğunda daha iyi kabul edileceği için, (yani problemin  $MS$  değeri maksimizasyon

problemidir)  $MS$  değeri  $1/MS$  olarak değiştirilmiştir. Bu şekilde, algoritma, daha büyük  $MS$  değerlerine odaklanmak üzere ayarlanmıştır.

Son olarak, algoritmanın ana popülasyonuna yeni bireyler eklerken  $AVC$  değeri 4–6 aralığında,  $VFA$  değeri ise 60–75 aralığında olacak şekilde seçilmiştir. Bu sayede, her sonuç Türk Karayolları Şartnamesi tarafından önerilen değerlerle uyumlu hale getirilmiştir.  $VFA$  ve  $AVC$  değerlerinde sonuçların ilgili aralıkta olması şartname için yetmektedir fakat şartnameye uygun farklı aralıklardaki değerlerin elde edilmesi uygun parametre değerlerinin seçim yelpazesini genişletmiştir.

Çizelge 4.22'deki her bir çözüm dikkatlice incelendiğinde, tüm girdi ve çıktı değerlerinin Türkiye Karayolları Şartnamesine ( $MS \geq 750$  kg,  $AVC$  %4–6 ve  $VFA$  %60–75) uygun olduğu görülmektedir. Önerilen problem çözüme yaklaşımı ile optimum çıktılar için gerekli girdi oranlarının çözüm uzayı bulunmuştur. Bu sonuçlardan hareketle geliştirilen modelin bitümlü sıcak karışımlar alanında çalışan araştırmacılar için yardımcı uzman sistem olarak kullanılabilceği düşünülmektedir.

İlgili çizelge dikkatlice incelendiğinde %40-%50 arası değişen aralıklarla kullanılan geri dönüştürülmüş beton kullanmanın istenilen çıktılar için pozitif katkıları olduğu anlaşılmıştır. Aynı zamanda bir seviyeye kadar mermer atığını doğal kaba agrega yerine kullanmanın istenilen üç çıktı için kabul edilebilir olduğu tespit edilmiştir.

Özellikle alt ve üst limitleri arasında sürekli değerler alabilen ve çok fazla parametrelere sahip olan benzer problemler için bu çözüm yaklaşımını kullanmanın literatüre ve bu alanda çalışan araştırmacılara katkıları olacağı düşünülmektedir.

Bu çalışmanın katkıları olduğu gibi eksik ve geliştirilmesi gereken yönleri de vardır. Katkıları; ilgili şartnameye uygun şekilde üretilmiş çok sayıda  $MS$ ,  $AVC$ ,  $VFA$  çıktılarının elde edileceği giriş değerlerin bulunmasıdır. Bu çalışmadan elde edilen veriler halihazırda asfalt çalışmalarında kullanılabilir durumdadır.

Çizelge 4. 22. HMA problemi için MOGA-HMA tarafından bulunan çözüm uzayı

	<b>BR</b> (%)	<b>NC</b> (%)	<b>NF</b> (%)	<b>MC</b> (%)	<b>RF</b> (%)	<b>MS</b> (kg)	<b>AVC</b> (%)	<b>VFA</b> (%)
1	4.7	100	50	0	50	1512.0	60.7	4.5
2	5.1	100	50	0	50	1510.5	69.2	4.4
3	5.3	100	50	0	50	1502.8	73.3	4.3
4	5.0	100	51	0	49	1500.8	66.1	4.3
5	5.4	100	50	0	50	1497.4	74.7	4.1
6	4.8	100	51	0	49	1495.8	63.0	4.3
7	5.0	100	52	0	48	1491.2	67.0	4.3
8	4.9	100	52	0	48	1489.8	65.2	4.3
9	5.0	100	53	0	47	1483.0	67.9	4.2
10	5.1	100	53	0	47	1481.7	70.9	4.2
11	5.1	99	53	1	47	1479.7	70.8	4.2
12	4.9	99	53	1	47	1477.3	66.0	4.2
13	5.0	100	54	0	46	1475.3	68.6	4.2
14	5.3	99	52	1	48	1475.1	73.9	4.1
15	5.1	100	54	0	46	1474.5	71.5	4.2
16	5.1	99	54	1	46	1472.6	71.3	4.1
17	5.3	98	52	2	48	1472.5	73.8	4.1
18	5.3	100	53	0	47	1468.4	74.4	4.1
19	5.2	100	54	0	46	1467.8	73.3	4.1
20	5.3	96	52	4	48	1467.4	73.6	4.0
21	5.1	92	53	8	47	1467.0	70.1	4.1
22	4.9	100	56	0	44	1461.8	67.9	4.1
23	5.1	92	54	8	46	1460.8	70.6	4.0
24	5.0	87	53	13	47	1459.3	66.6	4.2
25	5.0	100	57	0	43	1455.8	70.3	4.0
26	5.1	99	57	1	43	1454.2	72.6	4.0
27	4.7	100	56	0	44	1449.7	65.8	4.1
28	4.7	100	57	0	43	1444.2	66.4	4.0
29	5.0	92	57	8	43	1443.9	69.4	4.0
30	4.6	100	58	0	42	1436.2	64.9	4.3
31	5.0	73	50	27	50	1432.8	62.7	4.4
32	4.5	100	58	0	42	1431.3	60.1	5.0
33	5.0	70	52	30	48	1423.5	64.5	4.3
34	4.5	100	60	0	40	1417.3	60.7	5.0
35	5.0	69	57	31	43	1414.4	67.7	4.0
36	5.0	66	57	34	43	1409.1	67.5	4.0
37	5.0	59	50	41	50	1403.8	61.8	4.4
38	5.0	62	57	38	43	1402.1	67.3	4.0
39	5.0	57	51	43	49	1398.9	62.8	4.3
40	5.0	56	53	44	47	1395.2	64.5	4.2
41	5.0	52	50	48	50	1389.4	61.4	4.4
42	5.0	51	51	49	49	1386.8	62.4	4.3
43	5.0	48	51	52	49	1380.4	62.4	4.3
44	5.0	48	52	52	48	1380.0	63.3	4.2
45	5.0	47	50	53	50	1378.7	61.3	4.4
46	5.0	47	51	53	49	1378.4	62.3	4.3
47	5.0	47	53	53	47	1377.6	64.1	4.2
48	5.0	44	50	56	50	1372.3	61.3	4.4
49	5.0	44	52	56	48	1371.9	63.2	4.3
50	5.0	44	54	56	46	1371.3	64.7	4.2

Tezde kullanılan deney verilerinin eksik ve geliştirilmesi gereken yönleri ise güncel asfalt karışım deneylerinde kullanılan karışımın akma deneyi, yoğunluk ölçümü, nem içeriği, tükenme direnci ölçümleri ile gerçek deneylerin çıktılarını artırılabilir. Asfalt

karışımının giriş parametreleri olarak, mineral dolgu malzemesinin farklı oranları kullanılabilir. Bunun yanında, katkı maddeleri, örneğin polimer katkı maddeleri, köpük stabilizatörleri, yüzey aktif maddeler, reoloji modifiye edicileri bitüm maddesini geliştirmek için kullanılabilir. Son olarak geri dönüştürülmüş asfalt parçaları belirli oranda asfalt karışımına katılabilir (KGM, 2013; AASHTO, 2023; ASTM, 2023). Bu gibi yollarla deneylerin giriş parametreleri artırılıp yine MOGA-HMA algoritması ile optimizasyon yapılabilir. Amaçları artırılarak, üç amaçtan daha fazla amaçlı problemler tasarlanabilir böylece problem many-objective optimizasyon problemine evrilir ve MOGA-HMA bu kapsamda geliştirilebilir.

#### **4.3. F-MODE ve D-MOGA Karşılaştırılması.**

Bu bölümde F-MODE ve D-MOGA algoritmalarının performansları karşılaştırılmıştır. Ancak karşılaştırma yapılırken bu algoritmaların hangi problemleri çözmek için geliştirildikleri göz önüne alınması gerekmektedir. Örneğin, F-MODE 5,000 maksFes'te kabul edilebilir bir optimizasyon yapmak için tasarlanmıştır bu gerçeği göz önünde bulundurmadan her iki algoritmayı da 30,000 maksFes'te karşılaştırılırsa yanıltıcı sonuçlar oluşabilir. Bu sebepten dolayı her iki algoritma içinde diğer algoritmalar ile karşılaştırıldığı ve önerildiği parametrelerle karşılaştırılmıştır. Başka bir deyişle ilk karşılaştırmada her iki algoritma içinde popülasyon sayısı 100 ve maksFes 5,000 olarak ayarlanmıştır. İkinci karşılaştırmada ise popülasyon sayısı 300 ve maksFes 30,000 olarak ayarlanmıştır. Kıyaslamamanın doğru şekilde yapılması adına 100 bağımsız çalışma yapılmıştır.

Literatürde yaygın olarak kullanılan HV, GD, IGD metrikleri kapsamında yapılmış olan 100 bağımsız çalıştırmanın sonuçları Çizelge 4.23 -Çizelge 4.25'te elde edilen sonuçlar karşılaştırılmıştır. Bu tez çalışması kapsamında önerilen iki algoritma karşılaştırılmıştır. Bu karşılaştırma ile sonraki çalışmalarda kullanmak için faydalı, araştırmacılara yol gösterebilecek bilgiler elde edilmiştir.

Çizelge 4. 23. HV metriği kapsamında F-MODE ve D-MOGA kıyaslaması

HV	MaksFes:5000				MaksFes:30000			
	F-MODE		D-MOGA		F-MODE		D-MOGA	
	Ort.	S.S.	Ort.	S.S.	Ort.	S.S.	Ort.	S.S.
ZDT1	6.35E-01	4.2E-03	6.49E-01	3.1E-03	6.63E-01	1.4E-04	6.65E-01	4.1E-05
ZDT2	3.00E-01	5.1E-03	3.15E-01	3.3E-03	3.30E-01	1.3E-04	3.31E-01	4.5E-05
ZDT3	4.82E-01	6.0E-03	5.02E-01	3.5E-03	5.15E-01	2.5E-04	5.17E-01	6.9E-05
DTLZ1	1.13E-01	1.8E-01	3.40E-01	3.0E-01	7.93E-01	1.2E-03	8.01E-01	1.1E-03
DTLZ2	3.76E-01	6.0E-03	3.97E-01	4.8E-03	4.25E-01	2.2E-03	4.38E-01	2.0E-03
DTLZ3	0.00E+00	0.00E+00	9.01E-03	5.3E-02	0.00E+00	0.00E+00	4.89E-02	9.7E-02
DTLZ4	3.78E-01	7.0E-03	3.98E-01	4.9E-03	4.26E-01	2.2E-03	4.41E-01	2.0E-03
DTLZ5	8.91E-02	5.9E-04	9.16E-02	3.6E-04	9.43E-02	5.8E-05	9.47E-02	4.4E-05
DTLZ6	9.25E-02	1.8E-04	9.21E-02	1.7E-04	9.48E-02	4.2E-05	9.46E-02	3.5E-05
DTLZ7	2.55E-01	1.0E-02	2.95E-01	5.8E-03	3.15E-01	1.6E-03	3.28E-01	1.0E-03
WFG1	3.00E-01	5.3E-02	4.22E-01	6.1E-02	5.95E-01	2.9E-02	6.26E-01	2.4E-03
WFG2	5.53E-01	7.9E-04	5.54E-01	1.1E-03	5.58E-01	3.4E-05	5.58E-01	3.0E-05
WFG3	4.90E-01	4.3E-04	4.92E-01	4.5E-04	4.97E-01	7.8E-05	4.98E-01	4.9E-05
WFG4	2.05E-01	9.9E-04	2.09E-01	6.9E-04	2.12E-01	1.8E-04	2.13E-01	1.1E-04
WFG5	2.14E-01	4.4E-04	2.13E-01	1.1E-03	2.12E-01	1.3E-03	2.13E-01	3.8E-03
WFG6	2.06E-01	8.8E-04	2.06E-01	2.0E-03	2.12E-01	6.2E-05	2.13E-01	8.4E-05
WFG7	2.08E-01	4.4E-04	2.09E-01	2.9E-04	2.13E-01	7.7E-05	2.13E-01	4.8E-05
WFG8	2.74E-01	2.6E-03	2.76E-01	2.7E-03	2.17E-01	1.4E-03	2.18E-01	2.9E-03
WFG9	2.20E-01	6.4E-04	2.22E-01	1.0E-03	2.30E-01	3.2E-04	2.31E-01	4.2E-04
Fonseca	3.10E-01	3.1E-04	3.09E-01	3.1E-04	3.15E-01	5.8E-05	3.15E-01	5.8E-05
Kursawe	4.00E-01	5.0E-04	4.01E-01	3.6E-04	4.04E-01	7.2E-05	4.05E-01	5.5E-05
Schaffer	8.29E-01	2.6E-04	8.29E-01	2.7E-04	8.32E-01	5.7E-05	8.32E-01	5.0E-05

Çizelge 4. 24. GD metriği kapsamında F-MODE ve D-MOGA kıyaslaması

GD	MaksFes:5000				MaksFes:30000			
	F-MODE		D-MOGA		F-MODE		D-MOGA	
	Ort.	S.S.	Ort.	S.S.	Ort.	S.S.	Ort.	S.S.
ZDT1	1.74E-03	3.1E-04	7.00E-04	2.1E-04	5.89E-05	6.1E-06	1.15E-05	1.7E-06
ZDT2	1.93E-03	4.5E-04	7.42E-04	2.4E-04	5.87E-05	5.1E-06	1.18E-05	2.0E-06
ZDT3	1.44E-03	3.4E-04	5.67E-04	1.8E-04	5.72E-05	5.2E-06	1.59E-05	2.8E-06
DTLZ1	4.03E-01	2.6E-01	9.35E-02	9.8E-02	6.57E-04	5.8E-05	1.43E-04	5.2E-05
DTLZ2	2.40E-03	2.8E-04	8.38E-04	1.5E-04	7.52E-04	4.7E-05	7.37E-05	1.6E-05
DTLZ3	1.83E+00	2.8E-01	8.00E-01	2.7E-01	7.30E+00	1.10E+00	2.84E-01	2.40E-01
DTLZ4	2.21E-03	3.9E-04	7.75E-04	1.7E-04	7.66E-04	5.0E-05	8.09E-05	1.8E-05
DTLZ5	6.57E-04	1.3E-04	2.26E-04	7.2E-05	5.00E-05	4.6E-06	3.59E-06	1.3E-06
DTLZ6	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
DTLZ7	4.58E-03	6.7E-04	2.27E-03	4.3E-04	8.22E-04	7.0E-05	5.01E-04	7.9E-05
WFG1	3.47E-02	1.3E-02	1.62E-02	7.4E-03	2.97E-04	1.3E-03	4.32E-07	4.1E-07
WFG2	2.87E-04	5.4E-05	2.02E-04	6.4E-05	1.67E-05	1.5E-06	9.22E-06	1.3E-06
WFG3	2.33E-04	3.1E-05	1.17E-04	2.9E-05	2.67E-05	2.0E-06	5.45E-06	1.2E-06
WFG4	4.69E-04	7.4E-05	1.99E-04	4.9E-05	6.03E-05	8.0E-06	1.37E-05	4.4E-06
WFG5	5.05E-04	2.3E-05	4.98E-04	6.3E-05	6.25E-04	4.9E-05	5.55E-04	1.4E-04
WFG6	2.50E-04	6.1E-05	2.43E-04	1.3E-04	2.02E-05	2.0E-06	1.11E-05	2.7E-06
WFG7	1.33E-04	2.0E-05	5.55E-05	1.5E-05	1.64E-05	1.7E-06	2.78E-06	1.1E-06
WFG8	5.12E-03	6.8E-04	4.76E-03	1.1E-03	8.90E-03	1.8E-03	9.25E-03	1.6E-03
WFG9	4.76E-04	3.8E-05	3.48E-04	5.1E-05	1.09E-04	1.2E-05	5.64E-05	1.3E-05
Fonseca	2.04E-04	2.9E-05	2.64E-04	3.5E-05	3.53E-05	3.5E-06	4.63E-05	3.1E-06
Kursawe	3.01E-04	5.1E-05	2.04E-04	3.5E-05	3.82E-05	5.4E-06	3.00E-05	3.0E-06
Schaffer	5.31E-06	6.4E-06	6.47E-06	8.0E-06	2.58E-07	3.4E-07	3.35E-07	3.6E-07

Çizelge 4. 25. IGD metriği kapsamında F-MODE ve D-MOGA kıyaslaması

IGD	MaksFes:5000				MaksFes:30000			
	F-MODE		D-MOGA		F-MODE		D-MOGA	
	Ort.	S.S.	Ort.	S.S.	Ort.	S.S.	Ort.	S.S.
ZDT1	7.24E-04	1.1E-04	4.13E-04	6.8E-05	2.00E-05	1.0E-06	1.84E-05	1.0E-06
ZDT2	7.95E-04	1.5E-04	4.52E-04	8.2E-05	2.03E-05	7.9E-07	1.80E-05	4.5E-07
ZDT3	6.12E-04	1.1E-04	4.54E-04	9.7E-05	1.71E-05	5.0E-07	1.57E-05	6.8E-07
DTLZ1	3.24E-02	2.1E-02	1.61E-02	1.4E-02	2.05E-04	4.7E-06	1.93E-04	4.5E-06
DTLZ2	9.46E-04	3.4E-05	8.97E-04	3.4E-05	2.54E-04	4.8E-06	2.43E-04	5.1E-06
DTLZ3	1.27E+00	2.9E-01	5.74E-01	2.4E-01	1.02E+00	2.3E-01	5.99E-02	4.9E-02
DTLZ4	9.15E-04	3.4E-05	8.73E-04	3.2E-05	2.55E-04	4.6E-06	2.43E-04	6.1E-06
DTLZ5	2.07E-04	2.0E-05	1.60E-04	1.0E-05	2.03E-05	8.7E-07	1.85E-05	6.3E-07
DTLZ6	6.10E-05	3.9E-06	6.78E-05	3.4E-06	1.17E-05	5.6E-07	1.30E-05	3.9E-07
DTLZ7	1.18E-03	8.3E-05	9.87E-04	4.4E-05	2.05E-04	5.9E-06	1.95E-04	7.8E-06
WFG1	1.97E-02	4.7E-03	1.15E-02	4.4E-03	1.22E-04	7.3E-05	1.27E-05	3.8E-06
WFG2	2.21E-04	3.6E-05	2.38E-04	8.1E-05	1.54E-05	3.9E-07	1.51E-05	2.8E-07
WFG3	1.24E-04	7.4E-06	1.15E-04	7.7E-06	1.66E-05	7.3E-07	1.53E-05	5.4E-07
WFG4	2.17E-04	2.6E-05	1.63E-04	1.2E-05	2.83E-05	3.3E-06	2.28E-05	9.3E-07
WFG5	2.92E-04	2.4E-05	3.15E-04	2.9E-05	6.07E-04	5.0E-05	5.49E-04	1.2E-04
WFG6	1.73E-04	1.5E-05	1.70E-04	2.3E-05	2.11E-05	9.5E-07	1.97E-05	6.7E-07
WFG7	1.16E-04	9.7E-06	1.11E-04	7.1E-06	1.61E-05	9.2E-07	1.54E-05	6.0E-07
WFG8	1.40E-03	7.0E-05	1.32E-03	1.2E-04	2.11E-03	8.7E-05	2.07E-03	1.9E-04
WFG9	2.23E-04	9.6E-06	2.07E-04	1.3E-05	5.02E-05	3.0E-06	4.17E-05	1.8E-06
Fonseca	1.04E-04	4.3E-06	1.05E-04	3.8E-06	1.89E-05	5.2E-07	1.88E-05	3.9E-07
Kursawe	1.03E-04	6.7E-06	9.77E-05	5.9E-06	1.50E-05	3.7E-07	1.51E-05	9.7E-07
Schaffer	4.77E-05	3.3E-06	4.81E-05	3.8E-06	9.22E-06	4.8E-07	9.11E-06	3.6E-07

Örneğin, Çizelge 4.23 ve Çizelge 4.1'e bakılarak 5,000 maksFes'te F-MODE algoritmasının Çizelge 4.1'de karşılaştırıldığı algoritmalarından farklı olarak MODE ve OMOPSO algoritmaları ile de karşılaştırılma imkanları doğmuştur. Dikkat edilirse Çizelge 4.1'deki F-MODE ile Çizelge 4.23'teki F-MODE değerleri birbirlerine çok yakındır çünkü aynı parametreler ile elde edilmişlerdir. Başka bir bakış açısı ile bakılırsa parametreleri aynı olmasına rağmen iki çizelgedeki F-MODE'lerin az da olsa farklı sonuçlar üretmesidir. Bunun muhtemel sebebi 5,000 maksFes sayısının test problemlerini çözmeye yetersiz kalmasıdır. Bir ikinci sebep ise optimizasyon algoritmalarının temelde rastsalığa dayanmalarındandır.

Aynı şekilde Çizelge 4.23 ile Çizelge 4.17'deki 30,000 maksFes'teki D-MOGA değerleri çok benzerdir çünkü aynı parametreler kullanarak elde edilmişlerdir. Burada farklılığın daha az olması maksFes sayısının daha yüksek olmasındandır. Çizelge 4.23'teki veriler sayesinde F-MODE'un sonuçları farklı olarak MOEA/D ve MOEA/D-TS ile kıyaslanabilmektedir.

Çizelge 4.23'te başka önemli nokta ise, F-MODE algoritmasının maksFes değerinin 30,000 olarak ayarlanması ile az da olsa performansında bir artma olduğu gözlemlenmektedir. Aynı şekilde D-MOGA algoritmasının maksFes değeri 5,000 olduğunda performansının düştüğü gözlemlenmektedir.



Son olarak D-MOGA'nın her durumda F-MODE'e göre daha üstün bir algoritma olduğu görülmüştür. 22 Test fonksiyonunun ilk durumda 18, ikinci durumda 20 tanesinde daha başarılı olduğu anlaşılmıştır. Her iki durumda da *DTLZ6*, *Fonseca* fonksiyonlarında F-MODE'un başarılı olması NFL teoremini doğrular niteliktedir.

Çizelge 4.23- Çizelge 4.25'e göre D-MOGA'nın daha başarılı sonuçlar üretmesinin pek çok sebeplerinden önemli bazıları şunlardır; Öncelikle D-MOGA algoritmasında Pareto tabanlı yaklaşım çerçevesinde daha kapsamlı geliştirmeler yapılmıştır. Örneğin, yeni bir çaprazlama yaklaşımı sunulmuş, dinamik bir CR değeri kullanılmış ve aynı zamanda bütün aday çözümler kaliteli bireylerden seçilerek elitizm etkili şekilde kullanılmıştır. F-MODE'de ise amaç sadece az bir maksFes sayısında kabul edilebilir bir yakınsama yapması olduğundan belirli oranda Rulet tekerleği seçim mekanizması kullanılmıştır ve çözümlerin rulet tekeri seçim mekanizması ile seçilebilmesi için kaliteleri hesaplanmıştır. F-MODE, D-MOGA'nın gerisinde kalsa da Çizelge 4.23'e göre rakibine yakın değerler üretmiştir ve Çizelge 4.1 ve Çizelge 4.2'ye göre karşılaştırıldığı diğer algoritmalarından üstün bir performans göstermiştir.

#### 4.4. F-MODE ve D-MOGA'nın Literatürdeki Bazı Güncel Algoritmalarla Karşılaştırılması.

Reddy ve arkadaşları (2022) yılında, MOHO (*Many objective hybrid optimizer*) algoritmasını önerdikleri çalışmalarında test fonksiyonu olarak *DTLZ1-4* ve *WFG1-9*'u kullanmışlardır. Bu çalışmada *WFG* seti 3 amaçlı kullanıldığından dolayı *WFG* seti ile karşılaştırma yapılamamıştır. Çizelge 4.26'da maksFes sayısının çıkarılabileceği bilgiler verilmiştir. Bu çalışmada MOHO algoritmasının karşılaştırıldığı algoritmalarından bilinen iki algoritma NSGA-III (Mkaouer ve ark., 2015) ve MOEA/DD (*The Many-Objective Evolutionary Algorithm Based on Dominance and Decomposition*) (Li ve ark., 2014)'dır. En az maksFes olan *DTLZ1* bile 36,800 defa fonksiyon değerlendirmesi yapmıştır. F-MODE ve D-MOGA ise 30,000 maksFes'te *DTLZ 1-4* algoritmaları ile kıyaslanmışlardır.

**Çizelge 4. 26.** MOHO'nun popülasyon büyüklüğü ve nesil sayısı

Popülasyon Sayısı	Nesil Sayısı				
	DTLZ1	DTLZ2	DTLZ3	DTLZ4	WFG
92	400	250	1000	600	400

Çizelge 4.27'de bu algoritmalar F-MODE ve D-MOGA ile karşılaştırılmıştır. D-MOGA, rakiplerine göre fonksiyon değerlendirme sayısı az olduğu halde 100 bağımsız çalışma ortalamasında dört test fonksiyonun ikisine daha başarılı olmuştur.

**Çizelge 4. 27.** IGD metriğinde MOHO, NSGA-III ve MOEA/DD'nin önerilen algoritmalar ile karşılaştırılmaları

IGD	NSGA-III	MOEA/DD	MOHO	F-MODE	D-MOGA
DTLZ1	2.29E-3	1.09E-2	1.72E-4	2.05E-04	1.93E-04
DTLZ2	3.27E-3	3.59E-2	4.41E-3	2.54E-04	2.43E-04
DTLZ3	5.99E-3	3.67E-2	4.21E-1	1.02E+00	5.99E-02
DTLZ4	5.27E-4	3.60E-2	3.00E-3	2.55E-04	2.43E-04

Karakoyun (2020) ve arkadaşları tarafından yapılan bir çalışmada 25,000 MaksFes'te IBEA (*Indicator Based Evolutionary Algorithm*), MOCcell (*Multi-Objective Cellular Genetic Algorithm*), MOVS (*Multi-objective vortex search*) ve MOSG (*Multi-Objective, SFLA, GWO*) Algoritmaları ile HV metriği kapsamında karşılaştırmalar yapılmıştır.

Bu verilerle, yani literatürdeki doğrudan sonuçlarla, önerilen F-MODE ve D-MOGA algoritmalarını karşılaştırmak için 25,000 maksFeste de 100 defa birbirinden bağımsız olarak çalıştırılmıştır. Çizelge 4.28'de ilgili karşılaştırmalar yapılmıştır. Bu

karşılaştırmalara göre D-MOGA'nın performansı ön plana çıkmıştır. İlgili çizelgeye göre D-MOGA 20 test fonksiyonunun 16 tanesinde rakiplerine göre daha başarılı olmuştur.

**Çizelge 4. 28.** HV metriğinde MOSG, MOCcell, IBEA, MOVS algoritmalarının önerilen algoritmalar ile karşılaştırılmaları

HV	MaksFes:25,000											
	IBEA		MOCcell		MOVS		MOSG		F-MODE		D-MOGA	
	Ort.	S.s.	Ort.	S.s.	Ort.	S.s.	Ort.	S.s.	Ort.	S.s.	Ort.	S.s.
ZDT1	6.62E-01	9.00E-05	6.61E-01	2.30E-04	6.59E-01	2.40E-04	6.62E-01	3.10E-05	6.62E-01	2.4E-04	6.64E-01	7.3E-05
ZDT2	3.27E-01	1.40E-04	3.28E-01	4.10E-04	3.26E-01	3.20E-04	3.29E-01	2.20E-05	3.29E-01	2.4E-04	3.31E-01	9.1E-05
ZDT3	5.09E-01	6.20E-03	5.12E-01	8.80E-03	5.15E-01	2.90E-04	5.16E-01	3.60E-05	5.13E-01	4.4E-04	5.16E-01	1.7E-04
DTLZ1	1.74E-01	8.10E-02	3.40E-01	3.00E-01	5.21E-01	3.30E-01	5.56E-02	1.20E-01	7.90E-01	2.1E-03	7.99E-01	1.8E-03
DTLZ2	4.12E-01	6.00E-04	3.75E-01	5.50E-03	3.73E-01	5.00E-03	3.82E-01	3.80E-03	4.23E-01	2.3E-03	4.37E-01	1.9E-03
DTLZ3	-	-	-	-	-	-	-	-	0.00E+00	0.00E+00	6.66E-03	4.0E-02
DTLZ4	2.42E-01	1.30E-01	2.95E-01	1.60E-01	3.82E-01	4.50E-03	3.83E-01	4.90E-03	4.25E-01	2.3E-03	4.40E-01	2.2E-03
DTLZ5	9.17E-02	1.80E-04	9.40E-02	3.50E-05	9.28E-02	2.40E-04	9.32E-02	1.30E-04	9.42E-02	6.4E-05	9.47E-02	4.5E-05
DTLZ6	7.22E-02	1.40E-02	0.00E+00	0.00E+00	9.41E-02	1.70E-04	9.50E-02	2.80E-05	9.48E-02	3.3E-05	9.46E-02	3.6E-05
DTLZ7	2.34E-01	4.60E-02	2.56E-01	2.00E-02	2.69E-01	5.10E-03	2.88E-01	3.50E-02	3.13E-01	2.3E-03	3.28E-01	1.2E-03
WFG1	4.73E-01	1.10E-01	3.71E-01	9.20E-02	4.95E-01	6.20E-02	1.41E-01	8.40E-03	5.33E-01	3.9E-02	6.25E-01	2.5E-03
WFG2	5.50E-01	8.30E-04	5.52E-01	7.00E-03	5.57E-01	1.10E-04	5.53E-01	9.00E-04	5.58E-01	5.3E-05	5.58E-01	4.3E-05
WFG3	4.94E-01	3.70E-04	4.94E-01	5.50E-04	4.93E-01	3.10E-04	4.91E-01	6.70E-04	4.97E-01	7.6E-05	4.98E-01	5.5E-05
WFG4	2.09E-01	1.80E-04	2.10E-01	2.10E-04	2.09E-01	2.70E-04	2.08E-01	6.20E-04	2.12E-01	2.4E-04	2.13E-01	1.3E-04
WFG5	2.17E-01	8.00E-04	2.19E-01	3.00E-05	2.19E-01	3.80E-03	2.20E-01	2.90E-03	2.17E-01	9.8E-04	2.18E-01	3.3E-03
WFG6	1.97E-01	1.10E-02	1.83E-01	2.90E-02	2.06E-01	3.20E-03	2.06E-01	1.20E-03	2.12E-01	7.1E-05	2.13E-01	1.1E-04
WFG7	2.08E-01	1.80E-04	2.10E-01	1.00E-04	2.09E-01	2.50E-04	2.08E-01	3.80E-04	2.13E-01	5.3E-05	2.13E-01	4.4E-05
WFG8	-	-	-	-	-	-	-	-	2.72E-01	1.6E-03	2.73E-01	2.1E-03
WFG9	2.24E-01	1.30E-03	2.24E-01	2.70E-03	2.23E-01	1.10E-03	2.24E-01	1.00E-03	2.27E-01	3.8E-04	2.28E-01	3.4E-04
Fonseca	3.11E-01	1.10E-04	3.12E-01	9.70E-05	3.08E-01	4.00E-04	3.11E-01	1.40E-04	3.15E-01	5.9E-05	3.15E-01	5.9E-05
Kursawe	3.94E-01	9.20E-04	4.01E-01	9.70E-05	4.00E-01	2.30E-04	4.01E-01	2.40E-04	4.05E-01	9.0E-05	4.05E-01	4.2E-05
Schaffer	5.30E-01	2.10E-01	5.09E-01	2.10E-01	8.27E-01	2.80E-04	8.30E-01	2.80E-05	8.32E-01	4.0E-05	8.32E-01	3.9E-05

## 5. SONUÇLAR VE ÖNERİLER

### 5.1. Sonuçlar

Bu tez çalışması iki ana bölüme ayrılmaktadır; İlk çalışmada CNN tabanlı bir mimaride hiper parametre optimizasyonu yapabilecek çok amaçlı bir optimizasyon algoritması geliştirmek istenmiştir. CNN mimarilerinin zaman maliyetinin fazla olması problemi için literatürdeki muadillerine göre daha az maksFes'lerde makul yakınsama yapabilecek bir algoritmaya ihtiyaç duyulmuştur. Bu problemi çözmek için rulet tekerliği sistemini kullanmanın yeterli olacağı hipotezleri kurulmuştur. Rulet tekerliği seçimi için gerekli zemin hazırlanmıştır.

Bu kapsamda F-MODE adlı çok amaçlı bir optimizasyon algoritması geliştirilmiştir. Bu yaklaşımda, HV ve IGD metrikleri kapsamında 16 test fonksiyonu üzerinde NSGA-II, MOAAA, OMOPSO ve MODE algoritmalarıyla karşılaştırılmıştır. Bu önerilen algoritmada test fonksiyonları: *ZDT1*, *ZDT2*, *ZDT3*, *DTLZ1*, *DTLZ2*, *DTLZ3*, *DTLZ4*, *DTLZ5*, *DTLZ6*, *DTLZ7*, *WFG1*, *WFG2*, *WFG3*, *Fonseca*, *Kursawe* ve *Schaffer* olarak belirlenmiştir. Bu kıyaslama fonksiyonlarının sonuçlarına bağlı olarak algoritmaların 5,000 maksFes'teki performansları incelendiğinde, F-MODE'nin çalışmada karşılaştırılan algoritmalara göre hem HV hem de IGD metrikleri açısından çok yüksek performans sergilediği görülmüştür.

F-MODE, araştırmalarda çok yaygın olarak kullanılan HV metriği kapsamında yapılan 100 bağımsız çalışmanın ortalama ve standart sapma değerlerine göre *ZDT2*, *DTLZ5*, *Fonseca* ve *Schaffer* dışındaki tüm test fonksiyonlarında diğer algoritmalarından daha üstün bir performans göstermiştir. Bunun yanında F-MODE, performans olarak geri kaldığı bu dört fonksiyonda da kabul edilebilir sonuçlar ürettiği görülmüştür. Çok amaçlı problemler üzerinde hızlı yakınsama yapabilen OMOPSO beklenen bir sonuç olarak ikinci sırayı almıştır.

Algoritmanın performansını; doğruluk, yakınsama ve çeşitlilik açısından değerlendiren IGD metriği kapsamında F-MODE algoritması: 16 test fonksiyonundan 13'ünde daha iyi performans göstermiştir. Bu sonuçlara göre; F-MODE, IGD metriği çerçevesinde de başarılı sayılabilmektedir.

Ortalama ve standart sapma sonuçlarının yanında algoritmaların ürettiği Pareto-optimal yüzey sonuçlarında da F-MODE açıkça görülen başarılı bir performansa sahiptir. Mesela *ZDT1*, *ZDT2*, *ZDT3*, *DTLZ6* ve *Schaffer* problemlerinde, 5,000 maksFes'te diğer

algoritmalarla göre doğru yüzeylere çok daha iyi bir yakınsama yapmıştır. Bunun yanında kutu grafiği ile F-MODE'un diğer algoritmalarla göre performansının daha iyi olduğu görülmüştür.

Bu verilerin yanında Wilcoxon sıralı toplam testi ile F-MODE ile diğer algoritmalarla benzer sonuçlar üretip üretmediği kontrol edilmiştir. Bu istatistik verilerine göre, sadece HV metriği kapsamında NSGA-II algoritması ile *DTLZ2* probleminde; IGD metriği kapsamında ise NSGA-II ile *DTLZ5*; MODE algoritması ile *Fonseca* problemlerinde benzer sonuçlar üretildiği, diğer bütün problemlerde bütün algoritmalarla sonuçlarının farklı olduğu gözlemlenmiştir. Bu durumda 2 metrik, 4 algoritma ve 16 test fonksiyonu ile kıyaslanmış olan F-MODE 128 kıyaslamasının sadece 3 tanesinde benzer sonuçlar üretmiştir.

Kısaca bu çalışmanın hedefi olan 5,000 maksFes'te rakiplerine göre üstün olan F-MODE algoritmasında rulet tekeri seçim mekanizması kullanma fikrinin başarılı olduğu ispatlanmıştır.

Test fonksiyonları üzerinde F-MODE ile hızlı ve kabul edilebilir sonuçlar aldıktan sonra, tıbbi görüntülerin segmentasyonu için CNN tabanlı bir modelin hiper parametre optimizasyonu için uygulanmıştır. Bu uyarlamadan dolayı ismi bu uygulamada MODE-CNN olarak güncellenmiştir. Bu CNN tabanlı yöntemin *GS*, *ND* ve *PA* parametreleri, segmentasyon kaybı ve test süresine göre optimize edilmiştir. Minimum segmentasyon kaybı 55.2504 sn'de elde edilen 0.1607 olarak bulunmuştur. Bu uygulamanın genel sonuçlarına göre; segmentasyon kaybı azaldıkça test süresi artmaktadır, bu yüzden amaçları çakışan bir optimizasyon problemi olduğu anlaşılmıştır. Parametrelerden *GS* bir değerde sabitlenirken, *ND* ve *PA* değerleri farklı aralıklarda seyretmektedir. *ND* ve *PA* parametreleri her iki amacı da etkilemektedir. Test zamanı, *ND* ve *PA* değerleri arttıkça artmış ve segmentasyon kaybı azalmıştır. Bu çalışma, CNN mimarisi modelinin parametre optimizasyonunu, hem segmentasyon kaybını en aza indirmek hem de test süresini optimize etmek için yapılmıştır.

Bu çalışmanın ışığında, çok daha fazla parametreye sahip problemler veya üçten daha fazla amacı olan (*many-objective*) optimizasyon alanında çalışmalar yapılacağı öngörülmektedir. Elde edilen çözüm uzayından, zamanın ön plana çıktığı durumlarda en iyi zamana göre en optimum segmentasyon kaybının olduğu parametreler kullanılabilirken, segmentasyon kaybının önemsendiği durumlarda en az segmentasyon kaybına göre en optimum zamanı sağlayan parametreler kullanılabilir ve sonuçta bu alanda çalışanlara kolaylık sağlanmış olacaktır.

Çalışmanın ikinci kısmında ise sıcak asfalt karışım problemlerini çözmek amacıyla D-MOGA isimli bir algoritma önerilmiştir. HMA problemi için global aramada etkili, çeşitlilik performansı iyi olan bir algoritma geliştirmek istenmiştir. Bu problemi çözmek adına bir çaprazlama kullanmak yerine literatürde bilinen üç çaprazlama yöntemi (ABC, DE, BLX) hibrit şekilde kullanmanın algoritmanın performansına olumlu katkı sağlayacağı hipotezi geliştirilmiştir. İkinci bir hipotez ise, deney verilerinin amaç fonksiyonu çıkarmada yetersiz kaldığı durumlarda, FL modeli gibi sistemler tasarlanarak amaç fonksiyonu görevini üstlenmesi ile HMA gibi tüm karışım problemlerinin çok amaçlı bir problem olarak ele alınmasının yolu açabileceği düşüncesidir.

Bu kapsamda önerilen algoritma, HV ve GD metrikleri kullanılarak 21 test fonksiyonu üzerinde NSGA-II, MOAAA, MOEA/D ve MOEA/D-TS algoritmalarıyla karşılaştırılmıştır. D-MOGA ve karşılaştırıldığı algoritmalar test fonksiyonları: *ZDT1*, *ZDT2*, *ZDT3*, *DTLZ1*, *DTLZ2*, *DTLZ3*, *DTLZ4*, *DTLZ5*, *DTLZ6*, *DTLZ7*, *WFG1*, *WFG2*, *WFG3*, *WFG4*, *WFG5*, *WFG6*, *WFG7*, *WFG8*, *WFG9*, *Kursawe* ve *Schaffer* üzerinde 100 bağımsız çalışma gerçekleştirilmiştir.

Eşit şartlarda yapılan bu kıyaslamada HV metriğine göre 21 test fonksiyonun 18 tanesinde önerilen algoritma başarılı bir sonuç vermiştir. Algoritma tarafından tahmin edilen Pareto cephesinin, gerçek Pareto cephesine uzaklığını ölçen GD metriğine göre ise 21 test fonksiyonun 16 tanesinde D-MOGA daha iyi sonuç vermiştir. Bu veriler ışığında ortalama ve standart sapma verilerine bakılarak D-MOGA rekabetçi ve başarılı bir algoritmadır denilebilir.

Algoritmaların tahmin ettiği Pareto-optimal yüzeylere göre de D-MOGA'nın performansı ön plana çıkmaktadır. Diğer algoritmalarından farklı olarak özellikle *DTLZ1*, *WFG1*, *Schaffer* problemlerinde birebir gerçek Pareto-optimal yüzeyleri yakalamıştır. D-MOGA algoritmasının kutu grafiklerinde de diğer algoritmalarından daha başarılı olduğu gözlenmiştir.

Son istatistiksel test olarak sonuçlar Wilcoxon sıralı toplam testine tabi tutulmuş; 4 algoritma ile 21 test fonksiyonu 2 metrik üzerinde karşılaştırılma yapılmış ve 168 karşılaştırmanın sadece 4 tanesinde benzer sonuçlar vermiş böylece D-MOGA'nın rasgele değil anlamlı sonuçlar ürettiği ispatlanmıştır.

İstatistiksel testlerden sonra performansı ispatlanan D-MOGA algoritması, HMA problemine uyarlanmıştır. Bu uyarlanmadan dolayı ilgili uygulama için D-MOGA'nın ismi MOGA-HMA olarak kullanılmıştır. MOGA-HMA 50 popülasyon için 50 adet tam istenilen kriterlere uygun çözüm uzayı elde edilmiştir. Normalde elle yapılan bir deneyde

42 deneyden sadece 6 tanesi istenilen kriterleri tam karşıladığı düşünülürken elde edilen bu verilerin bu alanda çalışan araştırmacılara yardımcı bir uzman sistem olarak faydalı olacağı düşünülmektedir. Bulunan optimal veriler halihazırda asfalt çalışmalarında kullanılabilir durumda olduğu düşünülmektedir.

Bu verileri elde etmek için ayrıca bu çalışma kapsamında geliştirilen 82 kurallı uzman FL sistemi ile entegre çalışan MOGA-HMA ile; amaç fonksiyonun çıkarılması zor olan veya amaç fonksiyonunu formüleştirmeye uygun olmayan problemlerin çözümü için, bir çözüm yolu olarak literatüre girmiştir.

Son olarak D-MOGA ve F-MODE her ikisi için önerilen parametreler ile yaygın şekilde kullanılan HV, GD, IGD metriklerinde 5,000 ve 30,000 maksFes'lerde karşılaştırılmış sonuçta karşılaştırmaların çoğunda D-MOGA'nın başarılı olduğu gözlemlenmiştir; ancak sonuçlara dikkatli bakıldığında F-MODE'un her ne kadar D-MOGA'nın gerisinde kalsa da yakın sonuçlar ürettiği anlaşılmıştır.

Birbirinden ayrı her iki problemde önerilen algoritmaların ürettiği sonuçlara bakarak, çok amaçlı optimizasyon yaklaşımının gerçek dünya problemleri için ne kadar gerekli olduğu anlaşılmaktadır. Aynı zamanda esnek yapıları ile birçok probleme uyarlanabilecekleri ve bilimin birçok alanına katkı sağlayabilecek bir yaklaşım olduğu görülmektedir.

## 5.2. Öneriler

Her ne kadar D-MOGA ve F-MODE algoritmaları özel problemler için geliştirilmiş olsalar da benzer problemlere kolayca uyarlanabilecek esnekliğe sahiptirler. Özellikle, üç ya da daha fazla amacı aynı anda optimize etmeyi amaçlayan many-objective problemler önerilen algoritmalarla ele alınabilirler.

F-MODE segmentasyon kaybı ve ram kullanımı veya segmentasyon kaybı ve parametre sayısı gibi amaçları optimize etmek için veya buna benzer derin öğrenmenin farklı hiper parametrelerini optimize etmek için rahatça kullanılabilir dinamiklikte bir algoritmadır.

Hem CNN mimarilerinde hem de asfalt karışım problemlerinde daha çok parametrelili veya daha çok amacı optimize edecek şekilde tasarlanmış problemler için her iki algoritmada kullanılabilir bir şekilde tasarlanmıştır. İleri ki çalışmalarda D-MOGA daha güncel ve kabul gören teknolojik test verileri ile, sıcak asfalt karışım çalışmalarında

kullanılabilir. Bunun yanında D-MOGA ile sadece asfalt değil, beton gibi malzeme mühendisliği alanına ait karışımların, karışım oranları tespit edilebilir.

Bu tez kapsamında olmayan; akma deneyi, yoğunluk ölçümü, nem içeriği ve tükenme direnci gibi ölçümlerle yapılan asfalt karışım deneyleri geliştirilebilir ve MOGA-HMA many-objective optimizasyon alanına evrilebilir.

Ayrıca, asfalt karışımlarının giriş parametreleri olarak farklı mineral dolgu malzemesi oranlarının kullanılması, katkı maddelerinin (örneğin polimer katkı maddeleri, köpük stabilizatörleri, yüzey aktif maddeler, reoloji modifiye edicileri) bitüm performansını geliştirmek için kullanılması ve geri dönüştürülmüş asfalt parçalarının belirli oranlarda karışıma katılması gibi yöntemlerle deneylerin giriş parametreleri artırılarak MOGA-HMA ile yeniden optimize edilebilir.





**KAYNAKLAR**

- AASHTO, 2023, American Association of State Highway and Transportation Officials, <https://transportation.org/>: [15/07/2023].
- Abbass, H. A., Sarker, R. ve Newton, C., 2001, PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems, *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546)*, 971-978.
- About, A., Rokbani, N., Fdhila, R., Qahtani, A. M., Almutiry, O., Dhahri, H., Hussain, A. ve Alimi, A. M., 2022, DPb-MOPSO: A dynamic Pareto bi-level Multi-objective Particle Swarm Optimization Algorithm, *Applied Soft Computing*, 109622.
- Acılar, A., 2013, Yapay bağışıklık algoritmaları kullanılarak bulanık sistem tasarımı, Konya, Turkey, *Ph. D. thesis*.
- Adam, S. P., Alexandropoulos, S.-A. N., Pardalos, P. M. ve Vrahatis, M. N., 2019, No free lunch theorem: A review, *Approximation and optimization*, 57-82.
- Ahmedzade, P. ve Sengoz, B., 2009, Evaluation of steel slag coarse aggregate in hot mix asphalt concrete, *Journal of hazardous materials*, 165 (1-3), 300-305.
- Akay, B., 2013, Synchronous and asynchronous Pareto-based multi-objective Artificial Bee Colony algorithms, *Journal of Global Optimization*, 57 (2), 415-445.
- Akbari, R., Hedayatzadeh, R., Ziarati, K. ve Hassanizadeh, B., 2012, A multi-objective artificial bee colony algorithm, *Swarm and Evolutionary Computation*, 2, 39-52.
- Akbulut, H. ve Güreer, C., 2007, Use of aggregates produced from marble quarry waste in asphalt pavements, *Building and environment*, 42 (5), 1921-1930.
- Alakara, E. H. ve Agaoglu, M. N., 2020, Tokat İli Mermer İşletmelerinde Ortaya Çıkan Atık Mermerlerin ve Kentsel Dönüşüm Beton Atıklarının Bitümlü Sıcak Karışımlarda Kullanımının İncelenmesi, Master Degree, *Tokat Gaziosmanpaşa University*, Tokat.
- Alcalá, R., Nojima, Y., Herrera, F. ve Ishibuchi, H., 2011, Multiobjective genetic fuzzy rule selection of single granularity-based fuzzy classification rules and its interaction with the lateral tuning of membership functions, *Soft Computing*, 15, 2303-2318.
- Altıok, M., Alakara, E. H., Gündüz, M. ve Ağaoğlu, M. N., 2023, A multi-objective genetic algorithm for the hot mix asphalt problem, *Neural Computing and Applications*, 35 (11), 8197-8225.
- ASTM, 2023, American Society for Testing and Materials, <https://www.astm.org/>: [15/07/2023].
- Babalik, A., Ozkis, A., Uymaz, S. A. ve Kiran, M. S., 2018, A multi-objective artificial algae algorithm, *Applied Soft Computing*, 68, 377-395.
- Balamurugan, R., Natarajan, A. ve Premalatha, K., 2015, Stellar-mass black hole optimization for biclustering microarray gene expression data, *Applied Artificial Intelligence*, 29 (4), 353-381.
- Bergstra, J. ve Bengio, Y., 2012, Random search for hyper-parameter optimization, *The Journal of Machine Learning Research*, 13 (1), 281-305.
- Beycioglu, A., Gultekin, A., Aruntas, H. Y., Gencil, O., Dobiszewska, M. ve Brostow, W., 2017, Mechanical properties of blended cements at elevated temperatures predicted using a fuzzy logic model, *Comput. Concr*, 20 (2), 247-255.
- Charbit, P., Thomassé, S. ve Yeo, A., 2007, The minimum feedback arc set problem is NP-hard for tournaments, *Combinatorics, Probability and Computing*, 16 (1), 1-4.

- Coello, C. A. C., Pulido, G. T. ve Lechuga, M. S., 2004, Handling multiple objectives with particle swarm optimization, *IEEE Transactions on Evolutionary Computation*, 8 (3), 256-279.
- Coello, C. A. C., 2007, Evolutionary algorithms for solving multi-objective problems, Springer, p.
- Coello, C. C., 2006, Evolutionary multi-objective optimization: a historical view of the field, *IEEE computational intelligence magazine*, 1 (1), 28-36.
- Conover, W. J., 1999, Practical nonparametric statistics, John Wiley & Sons, p.
- Corne, D. W., Knowles, J. D. ve Oates, M. J., 2000, The Pareto envelope-based selection algorithm for multiobjective optimization, *International conference on parallel problem solving from nature*, 839-848.
- Corne, D. W., Jerram, N. R., Knowles, J. D. ve Oates, M. J., 2001, PESA-II: Region-based selection in evolutionary multiobjective optimization, *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, 283-290.
- da Silva, G. L., da Silva Neto, O. P., Silva, A. C., de Paiva, A. C. ve Gattass, M., 2017, Lung nodules diagnosis based on evolutionary convolutional neural network, *Multimedia Tools and Applications*, 76, 19039-19055.
- Deb, K. ve Beyer, H.-G., 2001, Self-adaptive genetic algorithms with simulated binary crossover, *Evolutionary computation*, 9 (2), 197-221.
- Deb, K., Pratap, A., Agarwal, S. ve Meyarivan, T., 2002, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, 6 (2), 182-197.
- Deb, K., Thiele, L., Laumanns, M. ve Zitzler, E., 2005, Scalable test problems for evolutionary multiobjective optimization, In: Evolutionary multiobjective optimization, Eds: Springer, p. 105-145.
- Dorigo, M., Maniezzo, V. ve Coloni, A., 1996, Ant system: optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26 (1), 29-41.
- Dufourq, E. ve Bassett, B. A., 2017, Eden: Evolutionary deep networks for efficient machine learning, *2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech)*, 110-115.
- Durillo, J. J. ve Nebro, A. J., 2011, jMetal: A Java framework for multi-objective optimization, *Advances in Engineering Software*, 42 (10), 760-771.
- Ergül, E., 2010, Çok amaçlı genetik algoritmalar: Temelleri ve Uygulamaları, ph.D., *Ondokuz Mayıs University, SAMSUN*.
- Fonseca, C. M. ve Fleming, P. J., 1993, Genetic Algorithms for Multiobjective Optimization: Formulation Discussion and Generalization, *Icga*, 416-423.
- Fonseca, C. M. ve Fleming, P. J., 1998, Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation, *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 28 (1), 26-37.
- Fortnow, L., 2009, The status of the P versus NP problem, *Communications of the ACM*, 52 (9), 78-86.
- Gencel, O., Ozel, C., Koksall, F., Erdogmus, E., Martínez-Barrera, G. ve Brostow, W., 2012, Properties of concrete paving blocks made with waste marble, *Journal of cleaner production*, 21 (1), 62-70.
- Gencel, O., Brostow, W., del Coz Diaz, J., Martínez-Barrera, G. ve Beycioglu, A., 2013, Effects of elevated temperatures on mechanical properties of concrete containing haematite evaluated using fuzzy logic model, *Materials Research Innovations*, 17 (6), 382-391.

- Gilani, V. N. M., Hosseinian, S. M., Behbahani, H. ve Hamedi, G. H., 2020, Prediction and pareto-based multi-objective optimization of moisture and fatigue damages of asphalt mixtures modified with nano hydrated lime, *Construction and Building Materials*, 261, 120509.
- Girshick, R., 2015, Fast r-cnn, *Proceedings of the IEEE international conference on computer vision*, 1440-1448.
- Goldberg, R. B., Barker, S. J. ve Perez-Grau, L., 1989, Regulation of gene expression during plant embryogenesis, *Cell*, 56 (2), 149-160.
- Güçlü, M., 2012, Yapay bağışıklık sistem tabanlı algoritma ile aykırı değer tespiti.
- He, K., Gkioxari, G., Dollár, P. ve Girshick, R., 2017, Mask r-cnn, *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2980-2988.
- Hebhoub, H., Aoun, H., Belachia, M., Houari, H. ve Ghorbel, E., 2011, Use of waste marble aggregates in concrete, *Construction and Building Materials*, 25 (3), 1167-1171.
- Holland, J. H., 1992, Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, MIT press, p.
- Horn, J., Nafpliotis, N. ve Goldberg, D. E., 1994, A niched Pareto genetic algorithm for multiobjective optimization, *Proceedings of the first IEEE conference on evolutionary computation. IEEE world congress on computational intelligence*, 82-87.
- Hsu, C.-H., Chang, S.-H., Liang, J.-H., Chou, H.-P., Liu, C.-H., Chang, S.-C., Pan, J.-Y., Chen, Y.-T., Wei, W. ve Juan, D.-C., 2018, Monas: Multi-objective neural architecture search using reinforcement learning, *arXiv preprint arXiv:1806.10332*.
- Huband, S., Hingston, P., Barone, L. ve While, L., 2006, A review of multiobjective test problems and a scalable test problem toolkit, *IEEE transactions on evolutionary computation*, 10 (5), 477-506.
- İnik, Ö. ve Ülker, E., 2022, Optimization of deep learning based segmentation method, *Soft Computing*, 26 (7), 3329-3344.
- İnik, Ö., Ceyhan, A., Balcioğlu, E. ve Ülker, E., 2019, A new method for automatic counting of ovarian follicles on whole slide histological images based on convolutional neural network, *Computers in biology and medicine*, 112, 103350.
- İnik, Ö., Altıok, M., Ülker, E. ve Koçer, B., 2021, MODE-CNN: A fast converging multi-objective optimization algorithm for CNN-based models, *Applied Soft Computing*, 109, 107582.
- Jamshidi, A., White, G., Hosseinpour, M., Kurumisawa, K. ve Hamzah, M. O., 2019, Characterization of effects of reclaimed asphalt pavement (RAP) source and content on dynamic modulus of hot mix asphalt concrete, *Construction and Building Materials*, 217, 487-497.
- Junior, F. E. F. ve Yen, G. G., 2019, Particle swarm optimization of deep neural networks architectures for image classification, *Swarm and Evolutionary Computation*, 49, 62-74.
- Kara, Ç. ve Karacasu, M., 2017, Investigation of waste ceramic tile additive in hot mix asphalt using fuzzy logic approach, *Construction and Building Materials*, 141, 598-607.
- Karaboga, D. ve Basturk, B., 2007, Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems, *Foundations of Fuzzy Logic and Soft Computing: 12th International Fuzzy Systems Association World Congress, IFSA 2007, Cancun, Mexico, June 18-21, 2007. Proceedings 12*, 789-798.
- Karaboga, D., 2010, Artificial bee colony algorithm, *scholarpedia*, 5 (3), 6915.

- Karakoyun, M., Ozkis, A. ve Kodaz, H., 2020, A new algorithm based on gray wolf optimizer and shuffled frog leaping algorithm to solve the multi-objective optimization problems, *Applied Soft Computing*, 96, 106560.
- Kartal, M., 2006, Bilimsel arařtırmalarda hipotez testleri, *Nobel Yayın Dağıtım, Ankara*.
- Kennedy, J. ve Eberhart, R., 1995, Particle swarm optimization, *Proceedings of ICNN'95-international conference on neural networks*, 1942-1948.
- KGM, K. G. M., 2013, Karayolu Teknik Şartnamesi, *Ankara, Türkiye, KTŞ*.
- Khodabakhshian, A., Ghalehnovi, M., De Brito, J. ve Shamsabadi, E. A., 2018, Durability performance of structural concrete containing silica fume and marble industry waste powder, *Journal of cleaner production*, 170, 42-60.
- Kiran, M. S., 2015, TSA: Tree-seed algorithm for continuous optimization, *Expert Systems with Applications*, 42 (19), 6686-6698.
- Kishor, A., Singh, P. K. ve Prakash, J., 2016, NSABC: Non-dominated sorting based multi-objective artificial bee colony algorithm and its application in data clustering, *Neurocomputing*, 216, 514-533.
- Knowles, J. ve Corne, D., 1999, The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation, *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*, 98-105.
- Krizhevsky, A., Sutskever, I. ve Hinton, G. E., 2012, Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems*, 1097-1105.
- Kursawe, F., 1990, A variant of evolution strategies for vector optimization, *International Conference on Parallel Problem Solving from Nature*, 193-197.
- Lee, C.-H., Du, J.-C. ve Shen, D.-H., 2012, Evaluation of pre-coated recycled concrete aggregate for hot mix asphalt, *Construction and Building Materials*, 28 (1), 66-71.
- Lee, W.-Y., Park, S.-M. ve Sim, K.-B., 2018, Optimal hyperparameter tuning of convolutional neural networks based on the parameter-setting-free harmony search algorithm, *Optik*, 172, 359-367.
- Li, H., Deb, K., Zhang, Q., Suganthan, P. N. ve Chen, L., 2019, Comparison between MOEA/D and NSGA-III on a set of novel many and multi-objective benchmark problems with challenging difficulties, *Swarm and Evolutionary Computation*, 46, 104-117.
- Li, K., Deb, K., Zhang, Q. ve Kwong, S., 2014, An evolutionary many-objective optimization algorithm based on dominance and decomposition, *IEEE Transactions on Evolutionary Computation*, 19 (5), 694-716.
- Liu, H., Simonyan, K., Vinyals, O., Fernando, C. ve Kavukcuoglu, K., 2017, Hierarchical representations for efficient architecture search, *arXiv preprint arXiv:1711.00436*.
- Liu, H., Simonyan, K. ve Yang, Y., 2018, Darts: Differentiable architecture search, *arXiv preprint arXiv:1806.09055*.
- Loni, M., Sinaei, S., Zoljodi, A., Daneshtalab, M. ve Sjödin, M., 2020, DeepMaker: A multi-objective optimization framework for deep neural networks in embedded systems, *Microprocessors and Microsystems*, 73, 102989.
- Lorenzo, P. R., Nalepa, J., Kawulok, M., Ramos, L. S. ve Pastor, J. R., 2017, Particle swarm optimization for hyper-parameter selection in deep neural networks, *Proceedings of the genetic and evolutionary computation conference*, 481-488.
- Ma, B., Li, X., Xia, Y. ve Zhang, Y., 2020, Autonomous deep learning: A genetic DCNN designer for image classification, *Neurocomputing*, 379, 152-161.

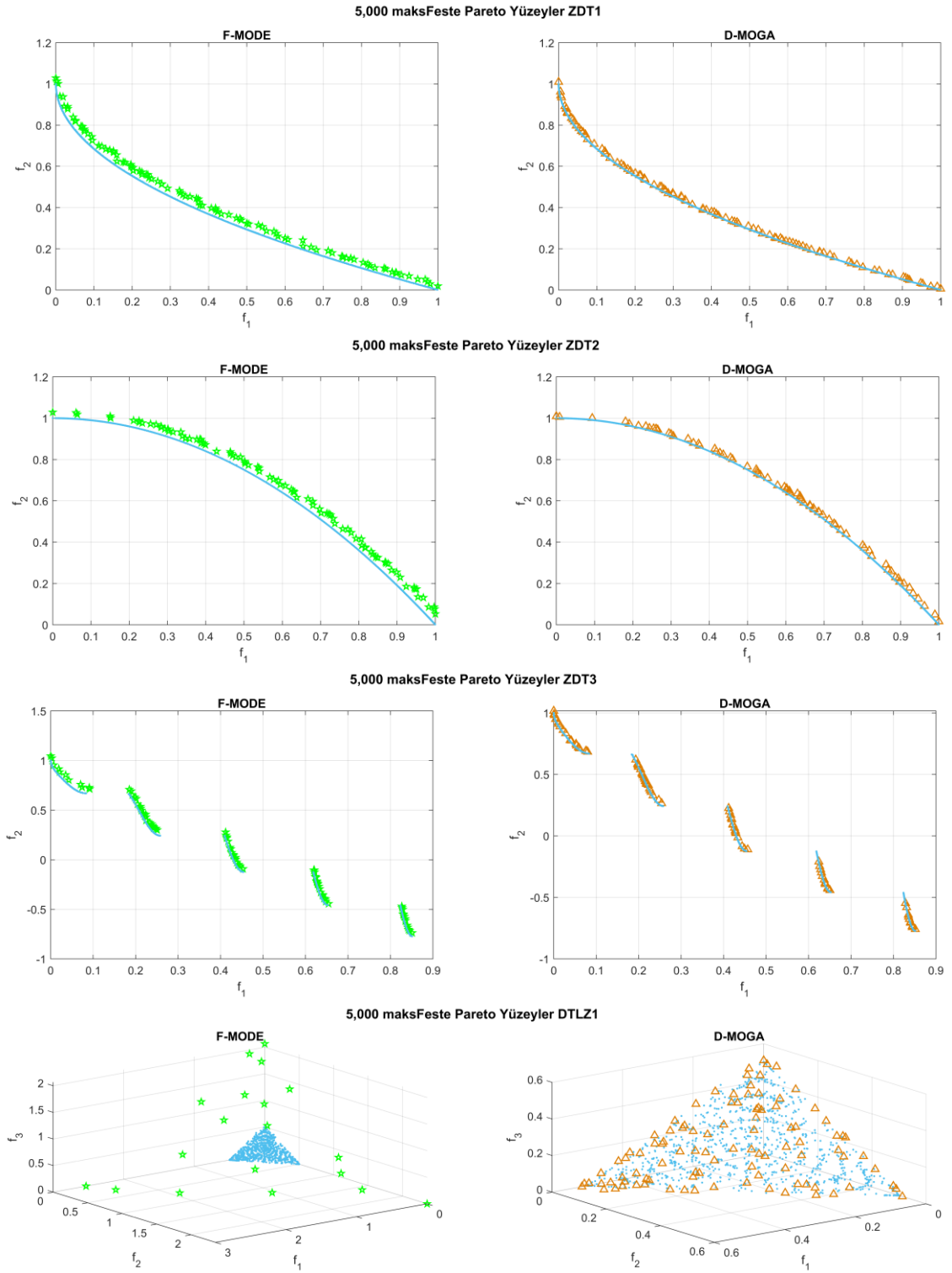
- Mahajan, M., Nimbhorkar, P. ve Varadarajan, K., 2012, The planar k-means problem is NP-hard, *Theoretical Computer Science*, 442, 13-21.
- Metin, U., Ercan, O., zgan ve Tuncay, K., 2012, Prediction of concrete compressive strength in buildings that would be reinforced by fuzzy logic, *International Journal of Physical Sciences*, 7 (31), 5193-5201.
- Miikkulainen, R., Liang, J., Meyerson, E., Rawal, A., Fink, D., Francon, O., Raju, B., Shahrzad, H., Navruzyan, A. ve Duffy, N., 2019, Evolving deep neural networks, In: *Artificial Intelligence in the Age of Neural Networks and Brain Computing*, Eds: Elsevier, p. 293-312.
- Mills-Beale, J. ve You, Z., 2010, The mechanical properties of asphalt mixtures with recycled concrete aggregates, *Construction and Building Materials*, 24 (3), 230-235.
- Mirjalili, S., Saremi, S., Mirjalili, S. M. ve Coelho, L. d. S., 2016, Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization, *Expert Systems with Applications*, 47, 106-119.
- Mkaouer, W., Kessentini, M., Shaout, A., Koligheu, P., Bechikh, S., Deb, K. ve Ouni, A., 2015, Many-objective software modularization using NSGA-III, *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 24 (3), 1-45.
- Nebro, A. J., Durillo, J. J., Garcia-Nieto, J., Coello, C. C., Luna, F. ve Alba, E., 2009, SMPSO: A new PSO-based metaheuristic for multi-objective optimization, *2009 IEEE Symposium on computational intelligence in multi-criteria decision-making (MCDM)*, 66-73.
- Özcan, F., Atiş, C. D., Karahan, O., Uncuoğlu, E. ve Tanyildizi, H., 2009, Comparison of artificial neural network and fuzzy logic models for prediction of long-term compressive strength of silica fume concrete, *Advances in Engineering Software*, 40 (9), 856-863.
- Ozcan, G., Ozkis, A. ve Kiran, M., MOTSA: A Multi-Objective Tree-Seed Algorithm, *International Conference on Advanced Technologies, Computer Engineering and Science*.
- Ozgan, E., 2009, Fuzzy logic and statistical-based modelling of the Marshall Stability of asphalt concrete under varying temperatures and exposure times, *Advances in Engineering Software*, 40 (7), 527-534.
- Özgan, E., 2010, Modelling the Stability of Asphalt Concrete with Fuzzy Logic and Statistical Methods for Various Freezing and Thawing Cycles, *Mathematical and Computational Applications*, 15 (2), 176-186.
- Özkıış, A., 2017, Girdap arama ve yapay alg algoritmalarının çok amaçlı optimizasyon problemlerine uyarlanması, *Doktora Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü*.
- Özkıış, A. ve Babalık, A., 2017, A novel metaheuristic for multi-objective optimization problems: The multi-objective vortex search algorithm, *Information Sciences*, 402, 124-148.
- Pelikan, M., Goldberg, D. E. ve Cantú-Paz, E., 1999, BOA: The Bayesian optimization algorithm, *Proceedings of the genetic and evolutionary computation conference GECCO-99*, 525-532.
- Peng, W. ve Zhang, Q., 2008, A decomposition-based multi-objective particle swarm optimization algorithm for continuous optimization problems, *2008 IEEE international conference on granular computing*, 534-537.
- Pourtahmasb, M. S., Karim, M. R. ve Shamshirband, S., 2015, Resilient modulus prediction of asphalt mixtures containing recycled concrete aggregate using an

- adaptive neuro-fuzzy methodology, *Construction and Building Materials*, 82, 257-263.
- Qi, Y., Ma, X., Liu, F., Jiao, L., Sun, J. ve Wu, J., 2014, MOEA/D with adaptive weight adjustment, *Evolutionary computation*, 22 (2), 231-264.
- Real, E., Moore, S., Selle, A., Saxena, S., Suematsu, Y. L., Tan, J., Le, Q. ve Kurakin, A., 2017, Large-scale evolution of image classifiers, *arXiv preprint arXiv:1703.01041*.
- Reddy, S. ve Dulikravich, G. S., 2022, A self-adapting algorithm for many-objective optimization, *Applied Soft Computing*, 129, 109484.
- Redmon, J., Divvala, S., Girshick, R. ve Farhadi, A., 2016, You only look once: Unified, real-time object detection, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779-788.
- Ren, S., He, K., Girshick, R. ve Sun, J., 2017, Faster R-CNN: towards real-time object detection with region proposal networks, *IEEE Transactions on Pattern Analysis & Machine Intelligence* (6), 1137-1149.
- Riquelme, N., Von Lücken, C. ve Baran, B., 2015, Performance metrics in multi-objective optimization, *2015 Latin American computing conference (CLEI)*, 1-11.
- Sargin, Ş., Saltan, M., Morova, N., Serin, S. ve Terzi, S., 2013, Evaluation of rice husk ash as filler in hot mix asphalt concrete, *Construction and Building Materials*, 48, 390-397.
- Savsani, V. ve Tawhid, M. A., 2017, Non-dominated sorting moth flame optimization (NS-MFO) for multi-objective problems, *Engineering Applications of Artificial Intelligence*, 63, 20-32.
- Schaffer, J. D., 1985, Multiple objective optimization with vector evaluated genetic algorithms, *Proceedings of the first international conference on genetic algorithms and their applications*, 1985.
- Serin, S., Morova, N., Sargin, Ş., Terzi, S. ve Saltan, M., 2013, The fuzzy logic model for the prediction of marshall Stability of lightweight asphalt concretes fabricated using expanded clay aggregate, *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 17 (1), 163-172.
- Shafabakhsh, G. ve Tanakizadeh, A., 2015, Investigation of loading features effects on resilient modulus of asphalt mixtures using Adaptive Neuro-Fuzzy Inference System, *Construction and Building Materials*, 76, 256-263.
- Siddique, R., Aggarwal, P. ve Aggarwal, Y., 2011, Prediction of compressive strength of self-compacting concrete containing bottom ash using artificial neural networks, *Advances in engineering software*, 42 (10), 780-786.
- Sierra, M. R. ve Coello, C. A. C., 2005, Improving PSO-based multi-objective optimization using crowding, mutation and  $\epsilon$ -dominance, *International conference on evolutionary multi-criterion optimization*, 505-519.
- Sipser, M., 1996, Introduction to the Theory of Computation, *ACM Sigact News*, 27 (1), 27-29.
- Srinivas, N. ve Deb, K., 1994, Multiobjective optimization using nondominated sorting in genetic algorithms, *Evolutionary computation*, 2 (3), 221-248.
- Subaşı, S., Beycioğlu, A., Sancak, E. ve Şahin, İ., 2013, Rule-based Mamdani type fuzzy logic model for the prediction of compressive strength of silica fume included concrete using non-destructive test results, *Neural Computing and Applications*, 22 (6), 1133-1139.
- Suganuma, M., Shirakawa, S. ve Nagao, T., 2017, A genetic programming approach to designing convolutional neural network architectures, *Proceedings of the genetic and evolutionary computation conference*, 497-504.

- Sun, Y., Xue, B., Zhang, M. ve Yen, G. G., 2018, A particle swarm optimization-based flexible convolutional autoencoder for image classification, *IEEE transactions on neural networks and learning systems*, 30 (8), 2295-2309.
- Sun, Y., Xue, B., Zhang, M. ve Yen, G. G., 2019, Evolving deep convolutional neural networks for image classification, *IEEE Transactions on Evolutionary Computation*, 24 (2), 394-407.
- TDK, T. D. K., 2023, Güncel Türkçe Sözlük, [http, sozluk.gov.tr](http://sozluk.gov.tr).
- Topçu, İ. B. ve Sarıdemir, M., 2008, Prediction of mechanical properties of recycled aggregate concretes containing silica fume using artificial neural networks and fuzzy logic, *Computational Materials Science*, 42 (1), 74-82.
- Uymaz, S. A., Tezel, G. ve Yel, E., 2015, Artificial algae algorithm (AAA) for nonlinear global optimization, *Applied Soft Computing*, 31, 153-171.
- Wang, Z., Zhang, Q., Zhou, A., Gong, M. ve Jiao, L., 2015, Adaptive replacement strategies for MOEA/D, *IEEE transactions on cybernetics*, 46 (2), 474-486.
- Wolpert, D. H. ve Macready, W. G., 1997, No free lunch theorems for optimization, *IEEE Transactions on Evolutionary Computation*, 1 (1), 67-82.
- Xiang, Y. ve Zhou, Y., 2015, A dynamic multi-colony artificial bee colony algorithm for multi-objective optimization, *Applied Soft Computing*, 35, 766-785.
- Xiang, Y., Zhou, Y. ve Liu, H., 2015, An elitism based multi-objective artificial bee colony algorithm, *European journal of operational research*, 245 (1), 168-193.
- Xie, L. ve Yuille, A., 2017, Genetic cnn, *Proceedings of the IEEE international conference on computer vision*, 1379-1388.
- Xue, F., Sanderson, A. C. ve Graves, R. J., 2003, Pareto-based multi-objective differential evolution, *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.*, 862-869.
- Yamasaki, T., Honma, T. ve Aizawa, K., 2017, Efficient optimization of convolutional neural networks using particle swarm optimization, *2017 IEEE third international conference on multimedia big data (BigMM)*, 70-73.
- Zapotecas-Martínez, S., García-Nájera, A. ve López-Jaimes, A., 2019, Multi-objective grey wolf optimizer based on decomposition, *Expert Systems with Applications*, 120, 357-371.
- Zehtabchi, A., Hashemi, S. A. H. ve Asadi, S., 2018, Predicting the strength of polymer-modified thin-layer asphalt with fuzzy logic, *Construction and Building Materials*, 169, 826-834.
- Zhang, Q. ve Li, H., 2007, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, *IEEE Transactions on Evolutionary Computation*, 11 (6), 712-731.
- Zhang, Y., Wang, G.-G., Li, K., Yeh, W.-C., Jian, M. ve Dong, J., 2020, Enhancing MOEA/D with information feedback models for large-scale many-objective optimization, *Information Sciences*, 522, 1-16.
- Zitzler, E. ve Thiele, L., 1999, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, *IEEE Transactions on Evolutionary Computation*, 3 (4), 257-271.
- Zitzler, E., Deb, K. ve Thiele, L., 2000, Comparison of multiobjective evolutionary algorithms: Empirical results, *Evolutionary computation*, 8 (2), 173-195.
- Zitzler, E., Laumanns, M. ve Thiele, L., 2001, SPEA2: Improving the strength Pareto evolutionary algorithm, *TIK-report*, 103.

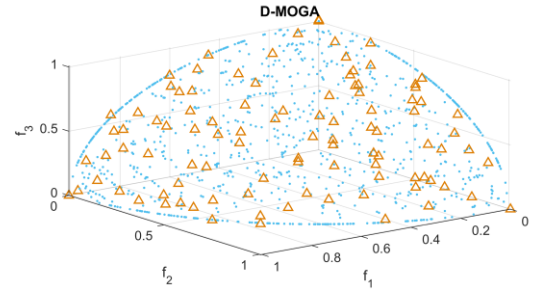
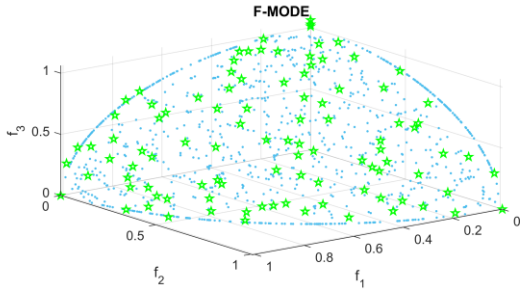
## EKLER

## EK-1.1. 5,000 maksFes F-MODE ve D-MOGA Pareto-optimum yüzeyler

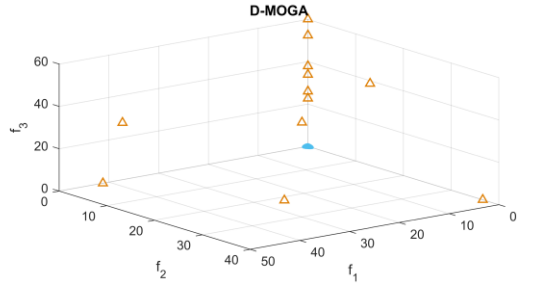
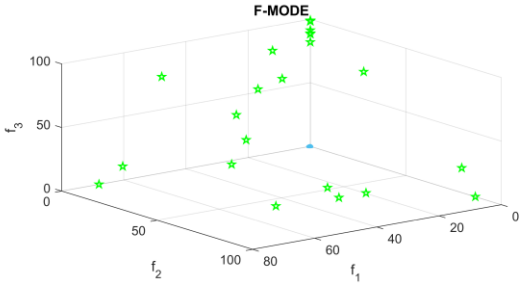




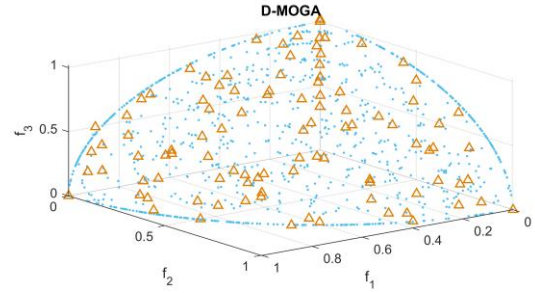
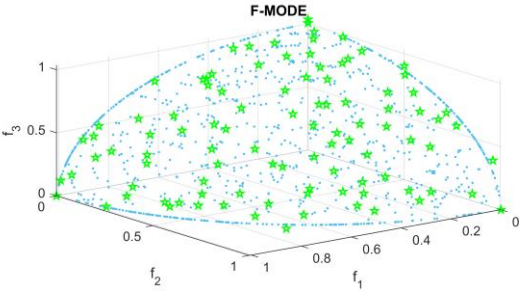
5,000 maksFeste Pareto Yüzeyler DTLZ2



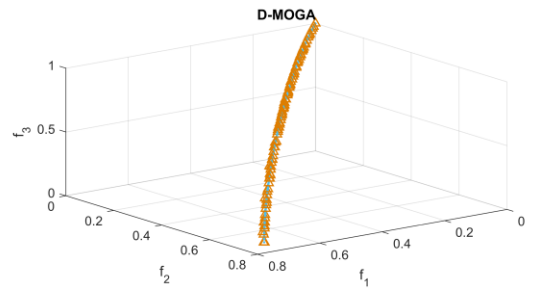
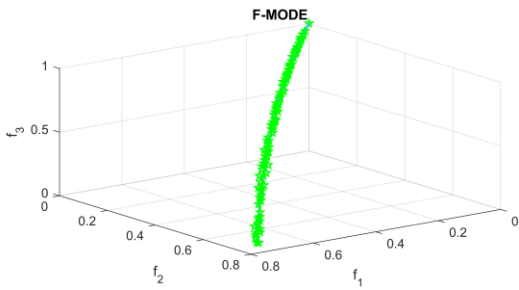
5,000 maksFeste Pareto Yüzeyler DTLZ3



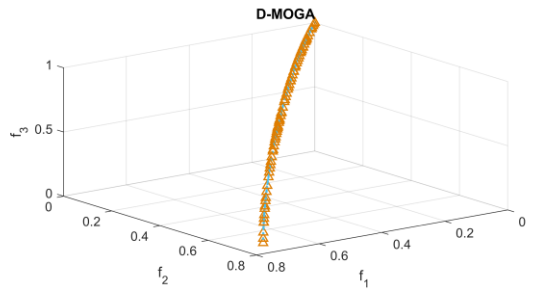
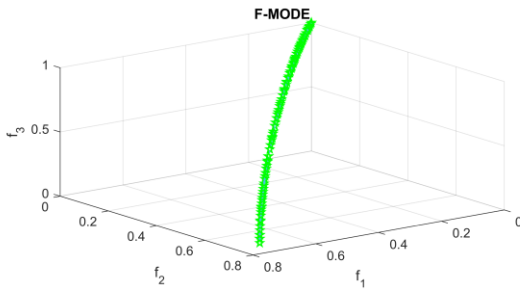
5,000 maksFeste Pareto Yüzeyler DTLZ4



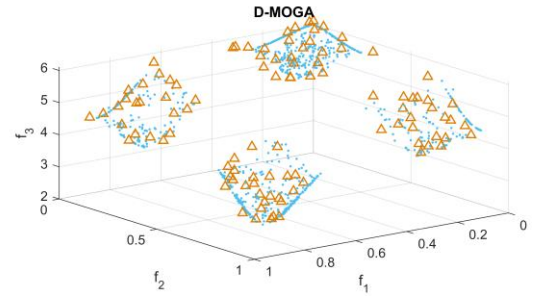
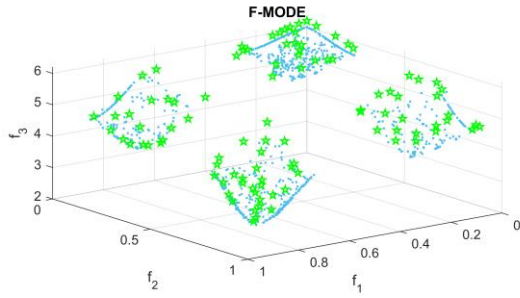
5,000 maksFeste Pareto Yüzeyler DTLZ5



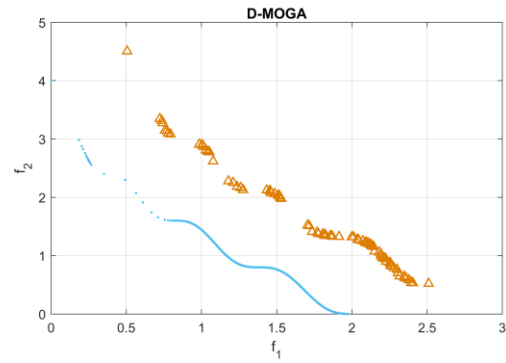
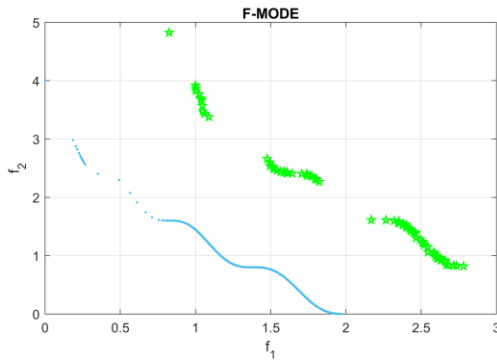
5,000 maksFeste Pareto Yüzeyler DTLZ6



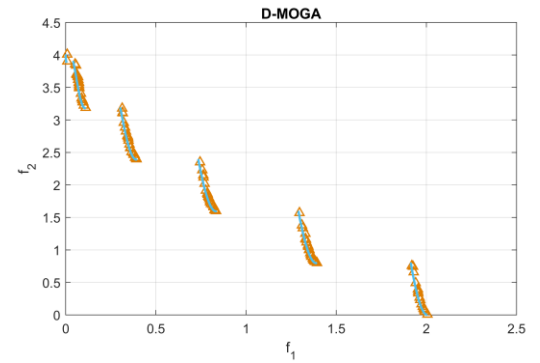
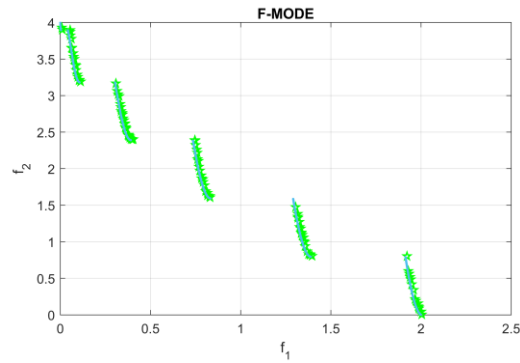
5,000 maksFeste Pareto Yüzeyler DTLZ7



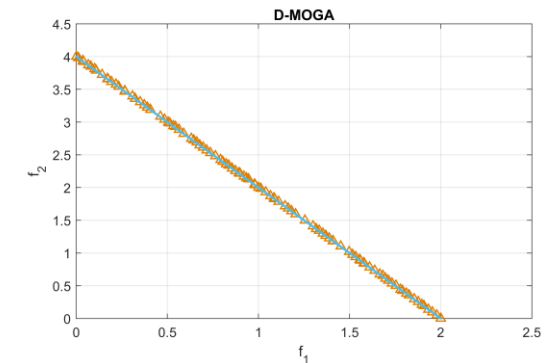
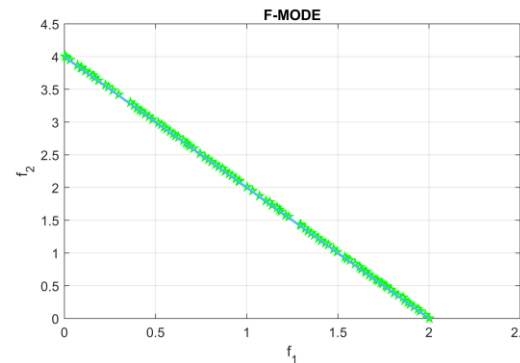
5,000 maksFeste Pareto Yüzeyler WFG1



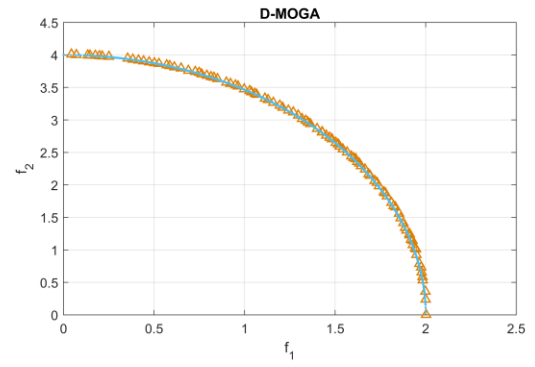
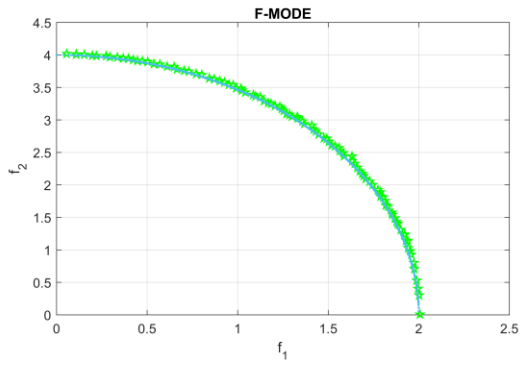
5,000 maksFeste Pareto Yüzeyler WFG2



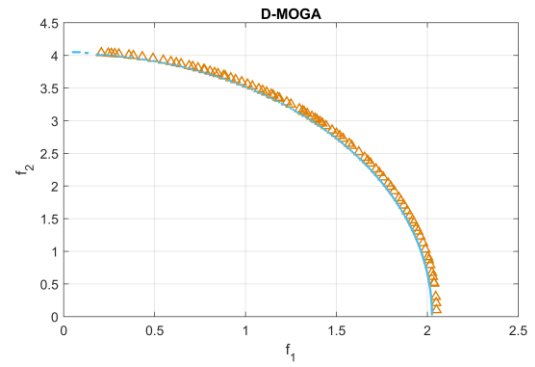
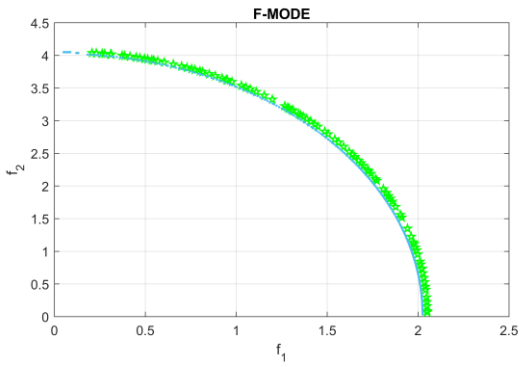
5,000 maksFeste Pareto Yüzeyler WFG3



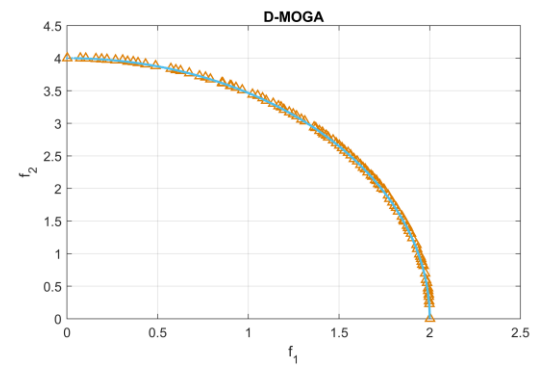
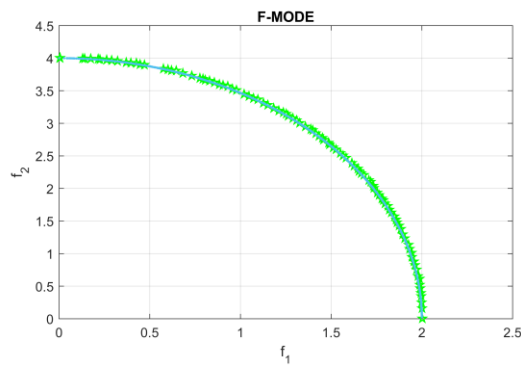
5,000 maksFeste Pareto Yüzeyler WFG4



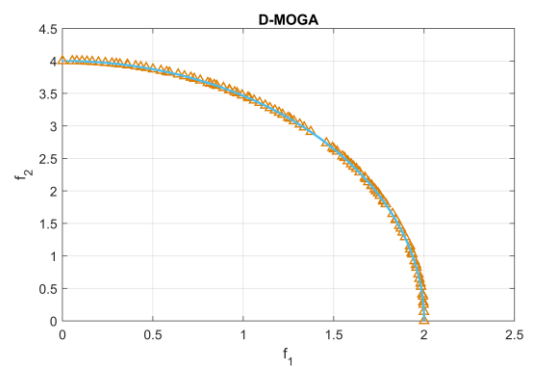
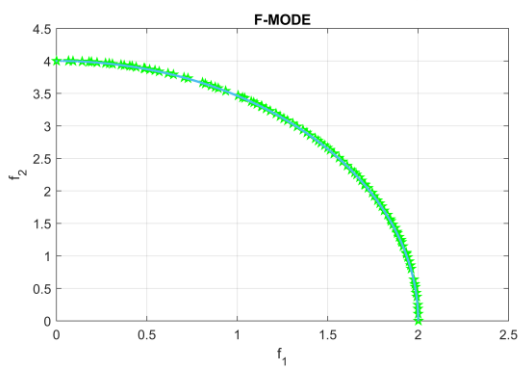
5,000 maksFeste Pareto Yüzeyler WFG5



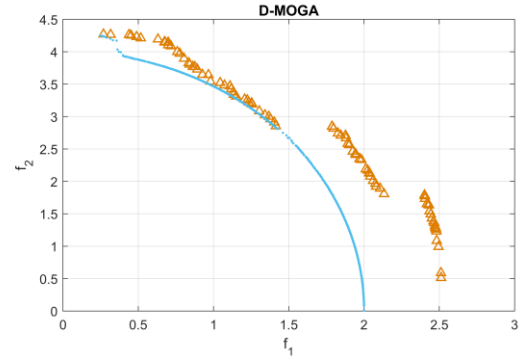
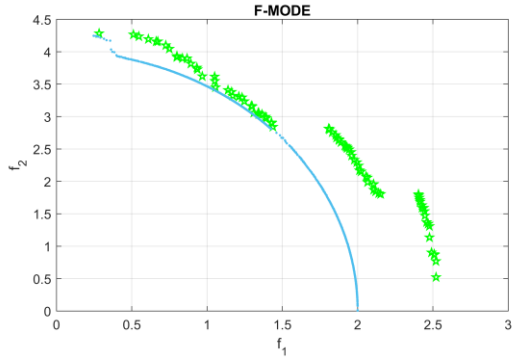
5,000 maksFeste Pareto Yüzeyler WFG6



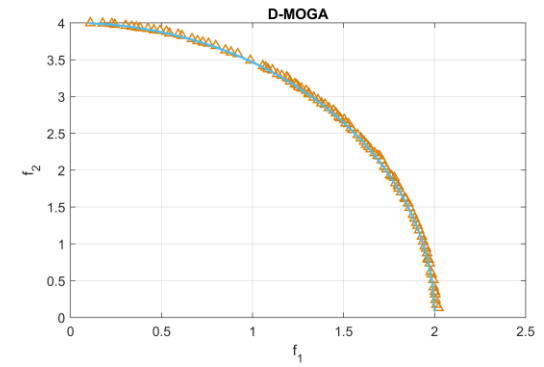
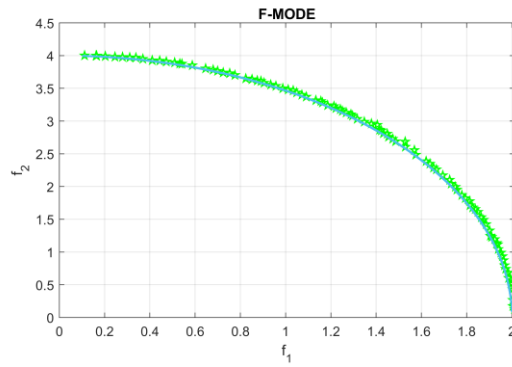
5,000 maksFeste Pareto Yüzeyler WFG7



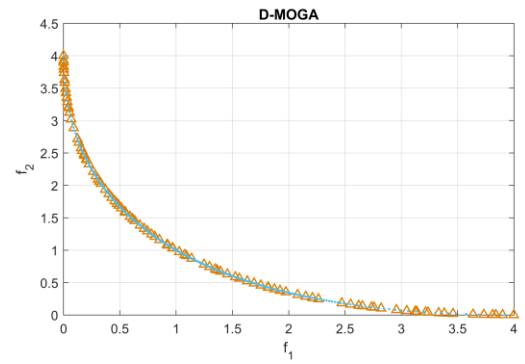
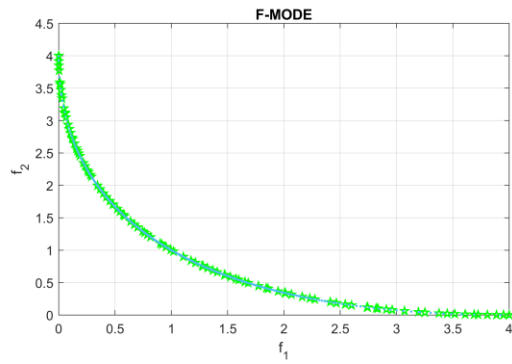
5,000 maksFeste Pareto Yüzeyler WFG8



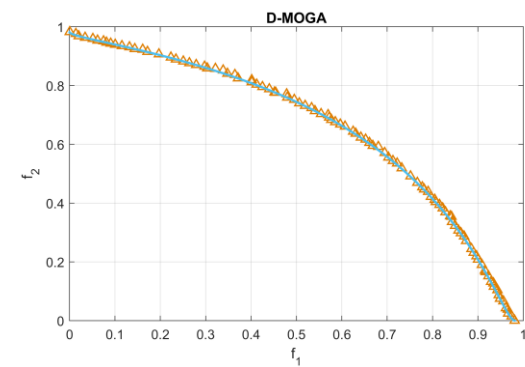
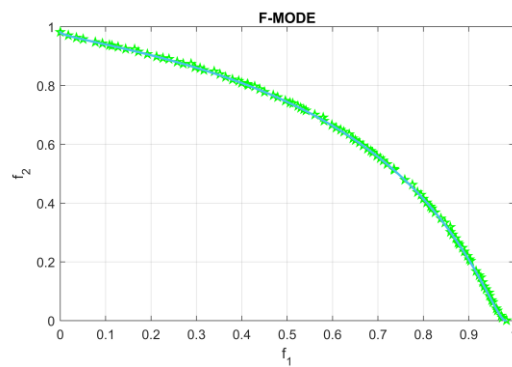
5,000 maksFeste Pareto Yüzeyler WFG9



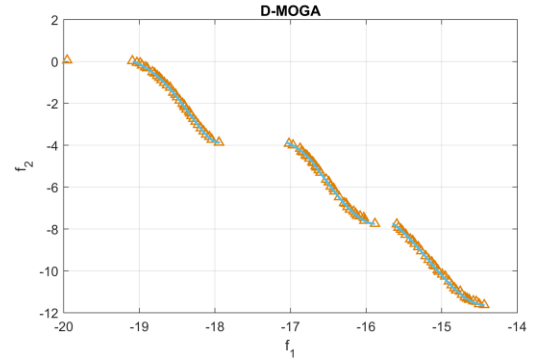
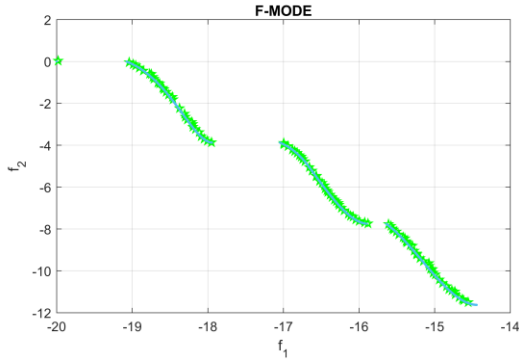
5,000 maksFeste Pareto Yüzeyler Schaffer



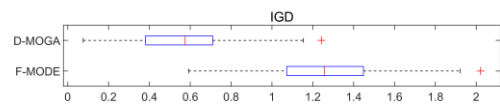
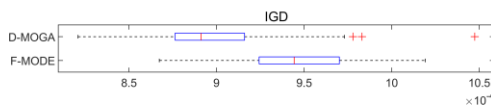
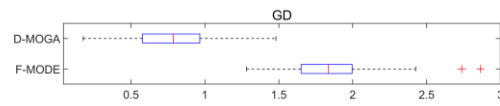
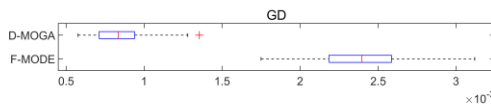
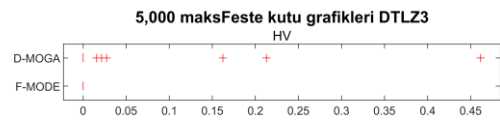
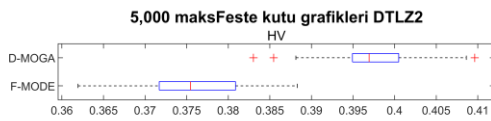
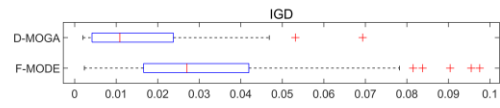
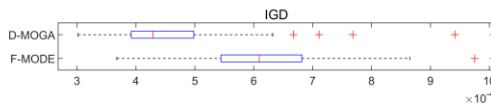
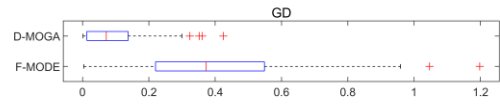
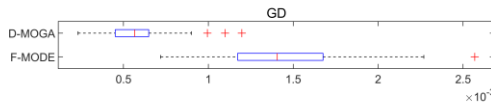
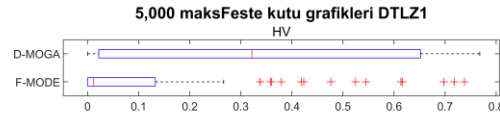
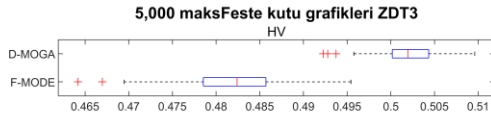
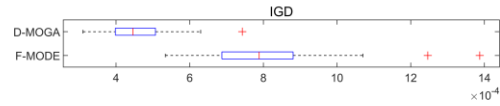
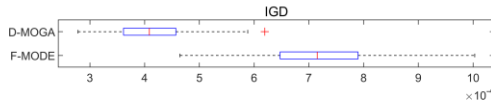
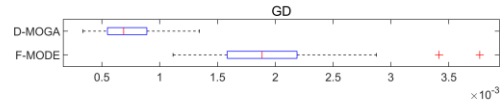
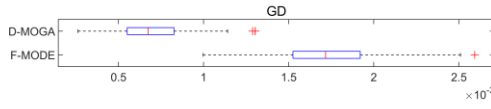
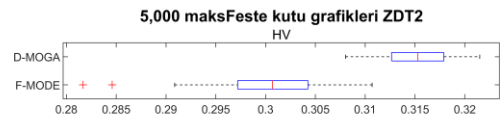
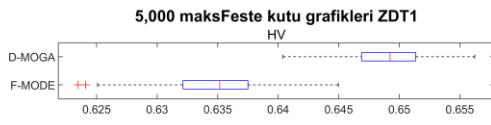
5,000 maksFeste Pareto Yüzeyler Fonseca

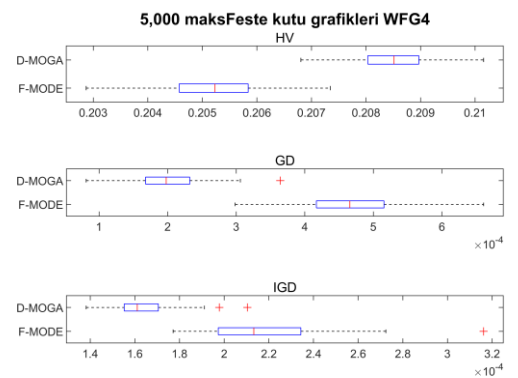
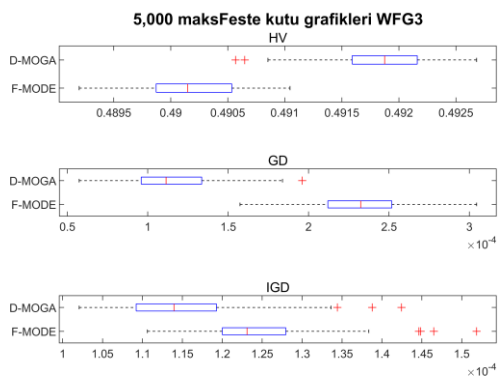
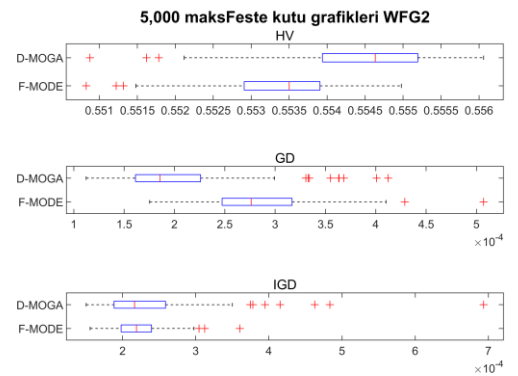
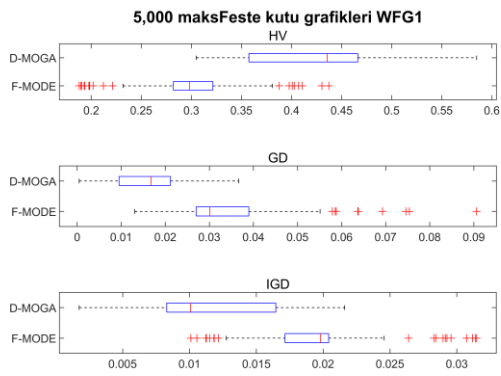
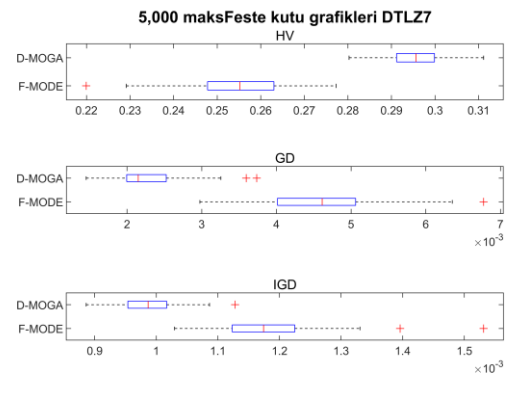
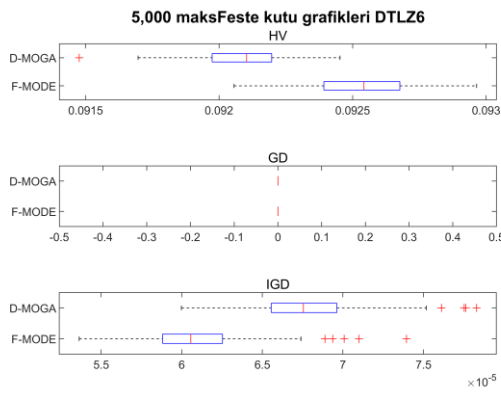
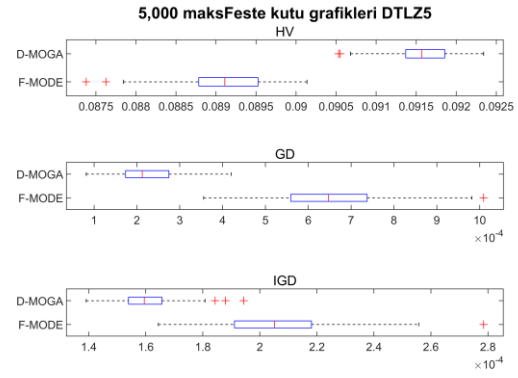
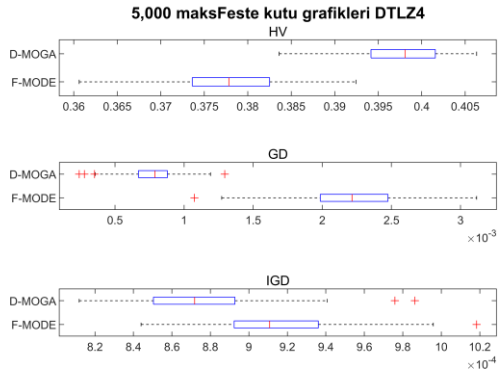


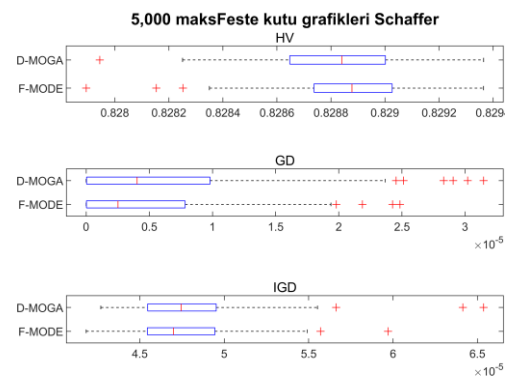
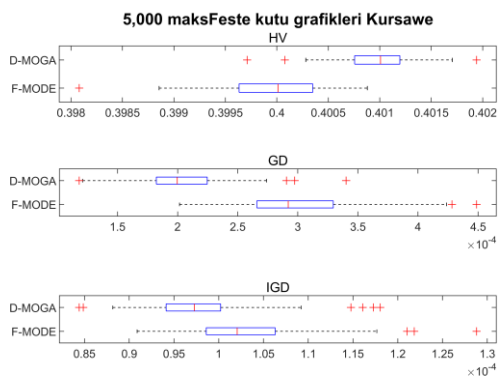
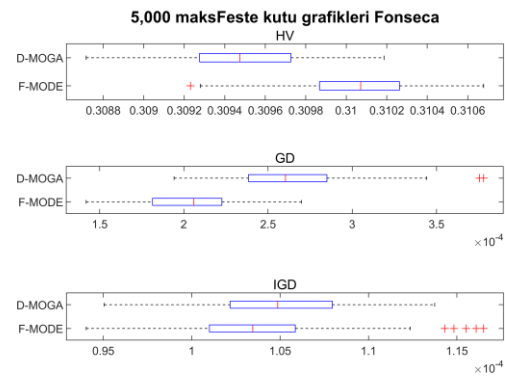
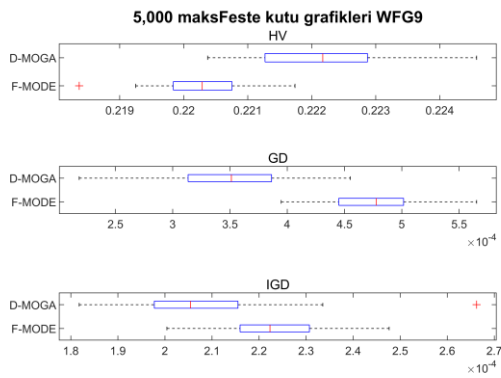
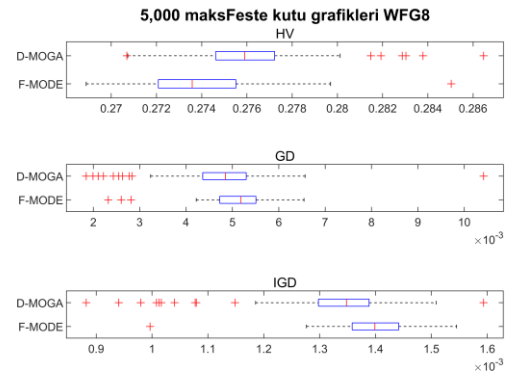
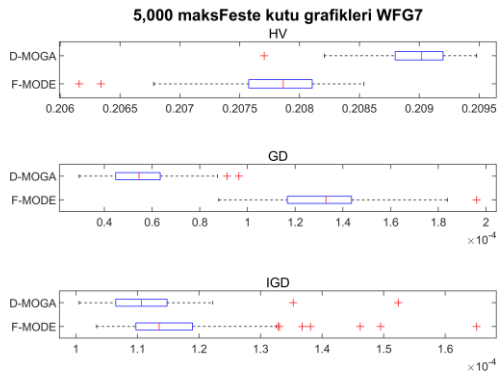
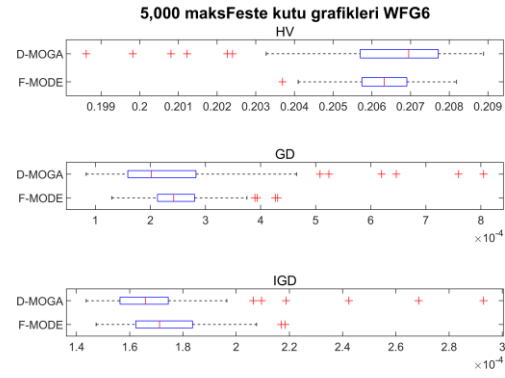
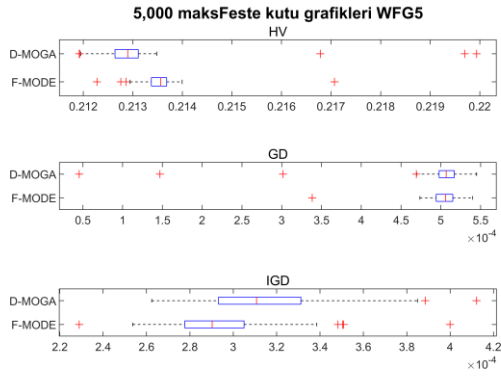
5,000 maksFeste Pareto Yüzeyler Kursawe



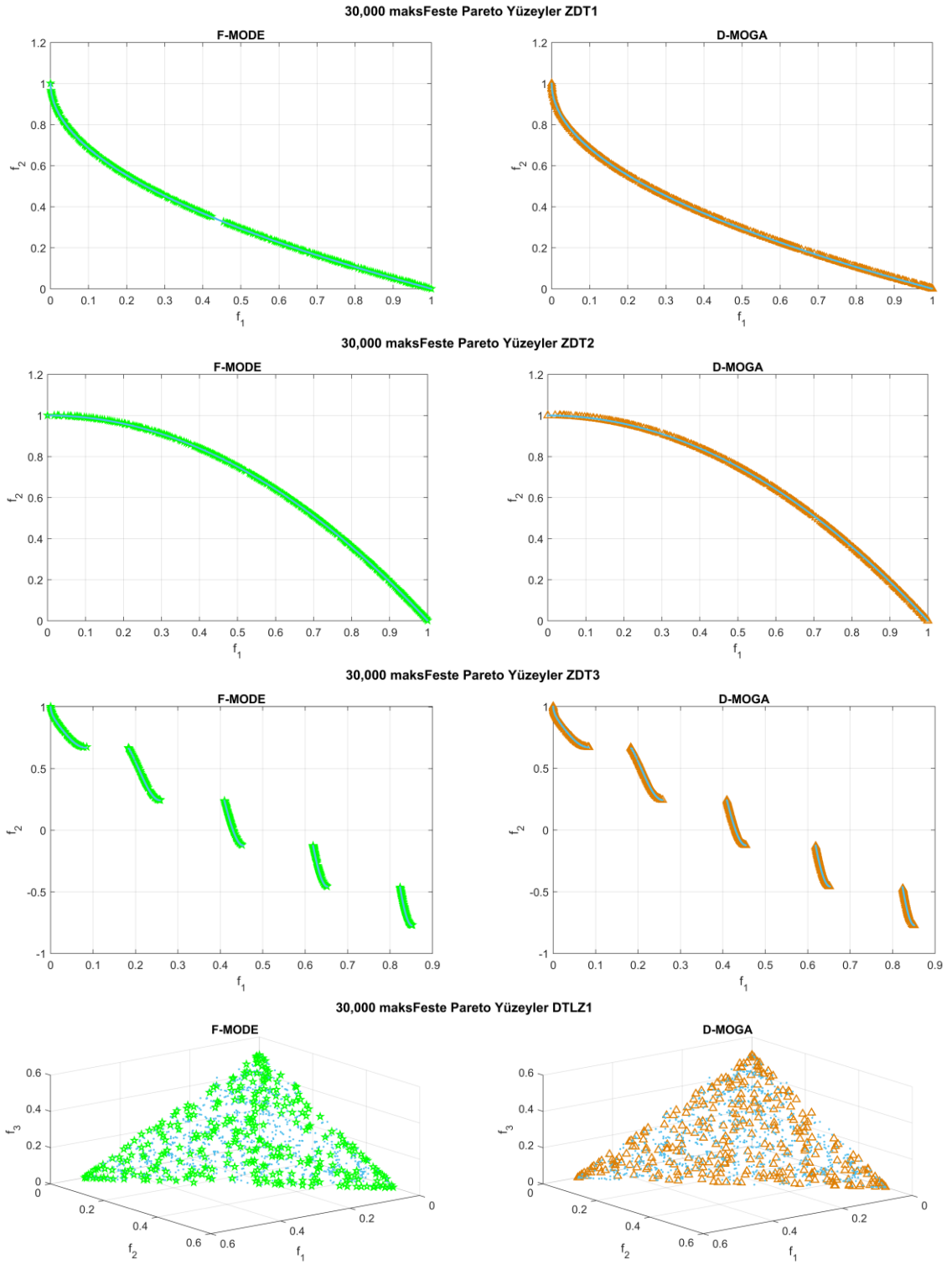
**EK-1.2. 5,000 maksFes F-MODE ve D-MOGA Kutu grafikler**





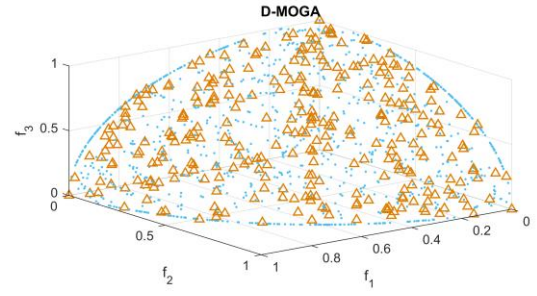
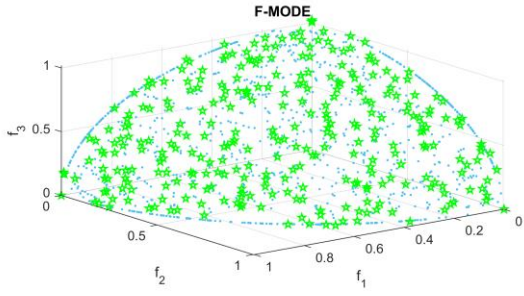


## EK-2.1. 30,000 maksFes F-MODE ve D-MOGA Pareto-optimum yüzeyler

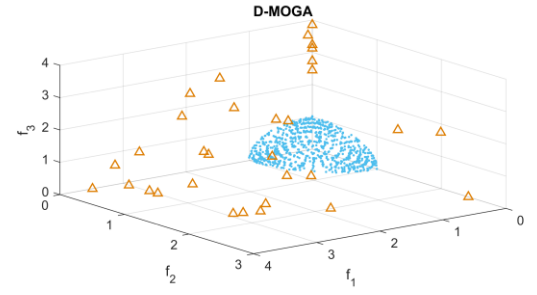
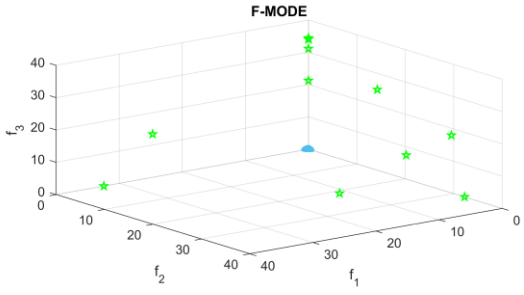




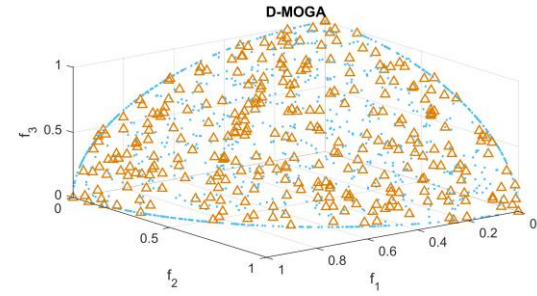
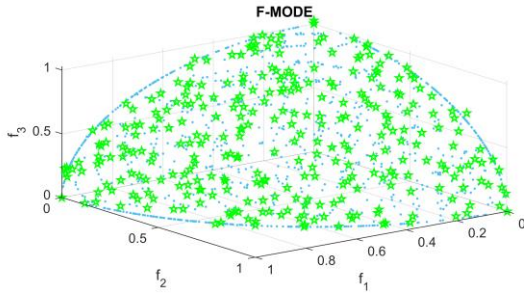
30,000 maksFeste Pareto Yüzeyler DTLZ2



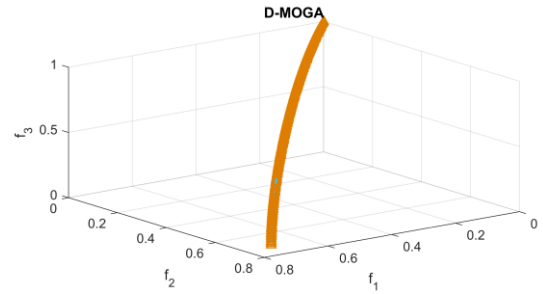
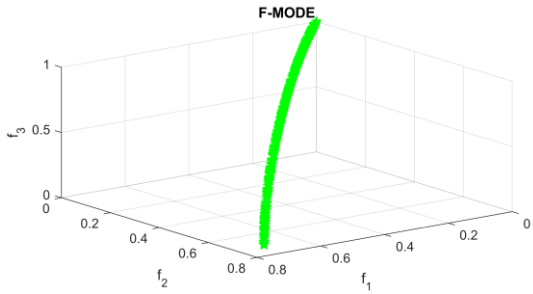
30,000 maksFeste Pareto Yüzeyler DTLZ3



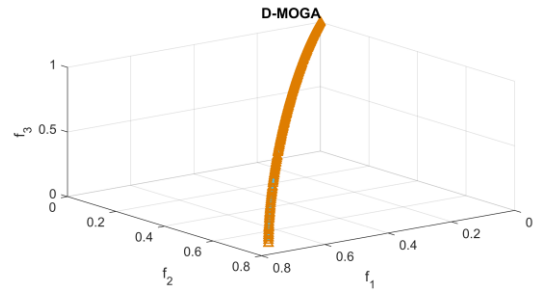
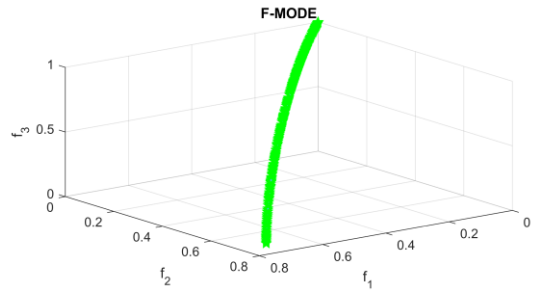
30,000 maksFeste Pareto Yüzeyler DTLZ4



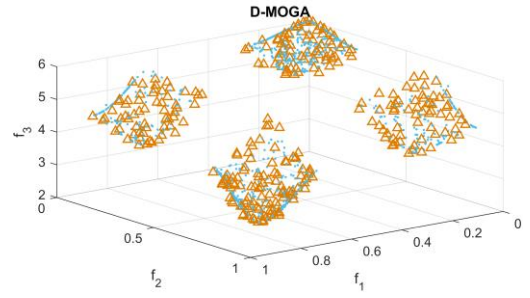
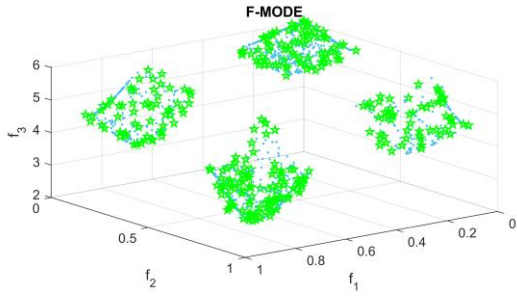
30,000 maksFeste Pareto Yüzeyler DTLZ5



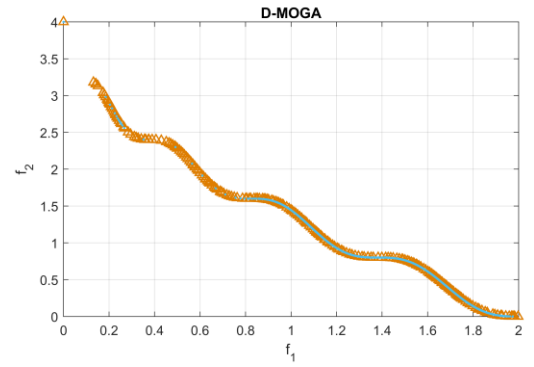
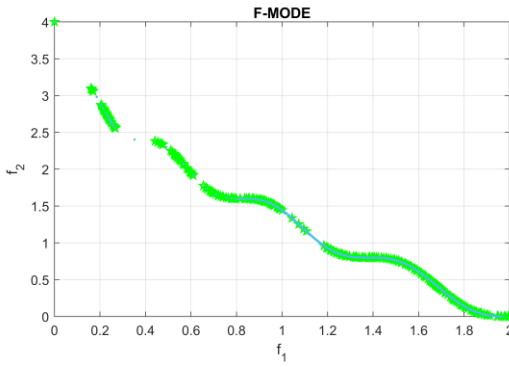
30,000 maksFeste Pareto Yüzeyler DTLZ6



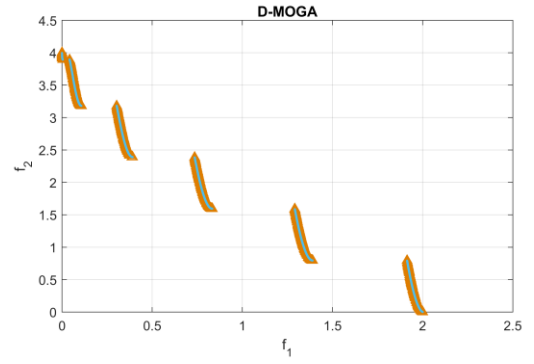
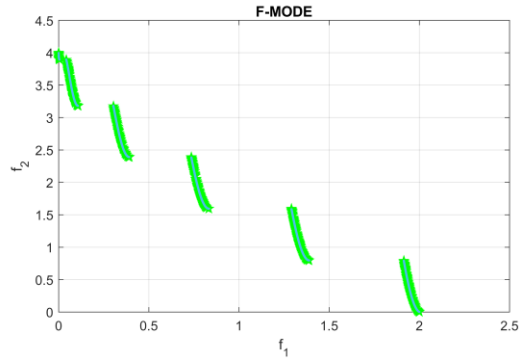
30,000 maksFeste Pareto Yüzeyler DTLZ7



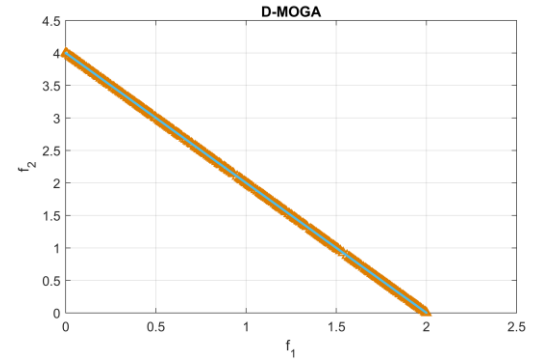
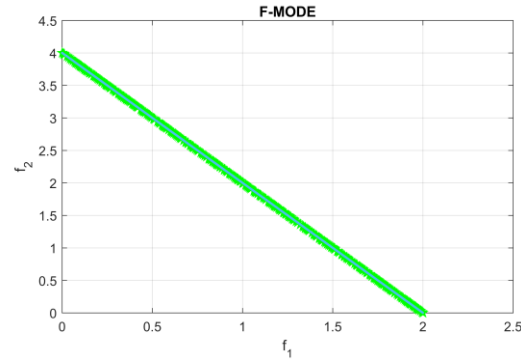
30,000 maksFeste Pareto Yüzeyler WFG1



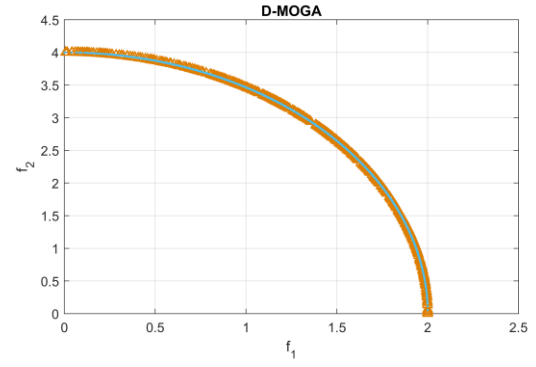
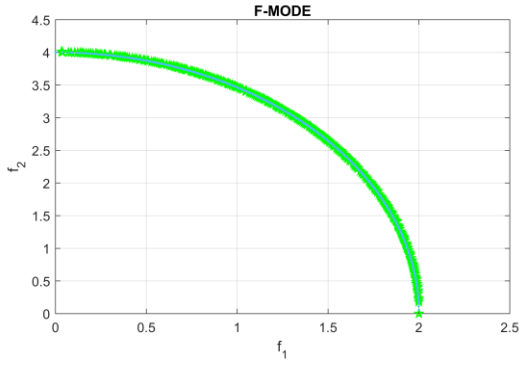
30,000 maksFeste Pareto Yüzeyler WFG2



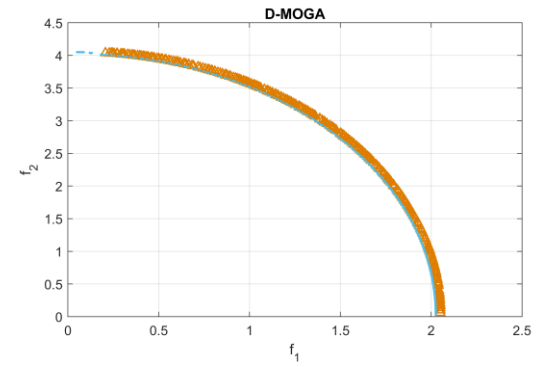
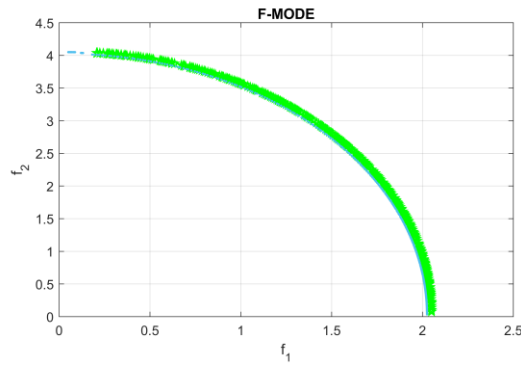
30,000 maksFeste Pareto Yüzeyler WFG3



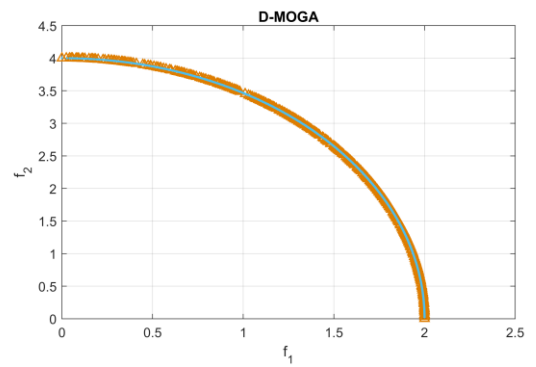
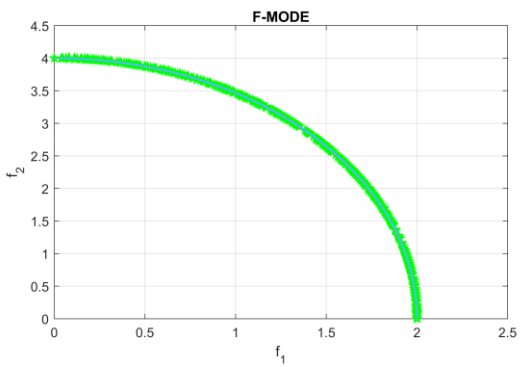
30,000 maksFeste Pareto Yüzeyler WFG4



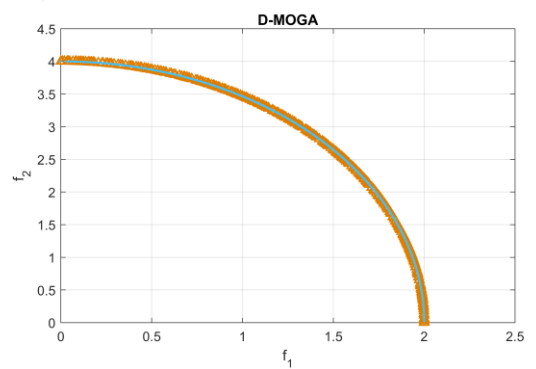
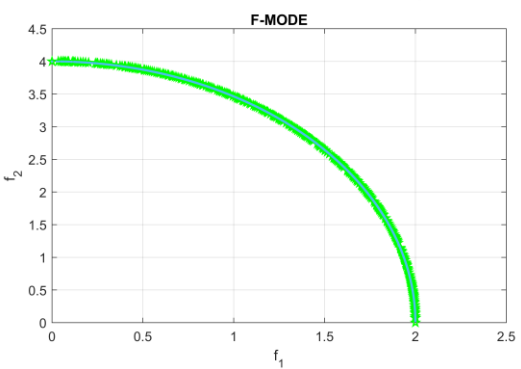
30,000 maksFeste Pareto Yüzeyler WFG5



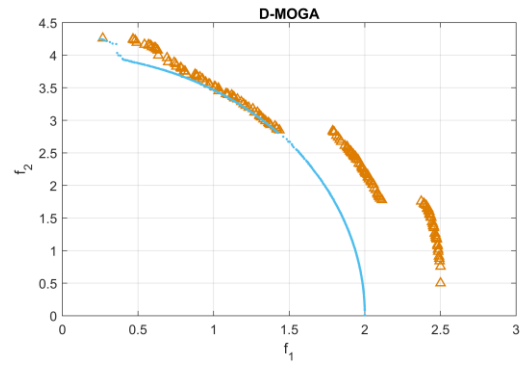
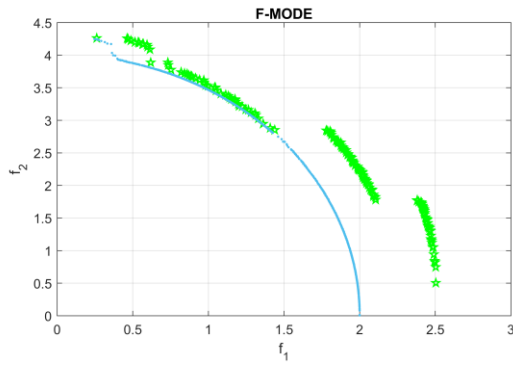
30,000 maksFeste Pareto Yüzeyler WFG6



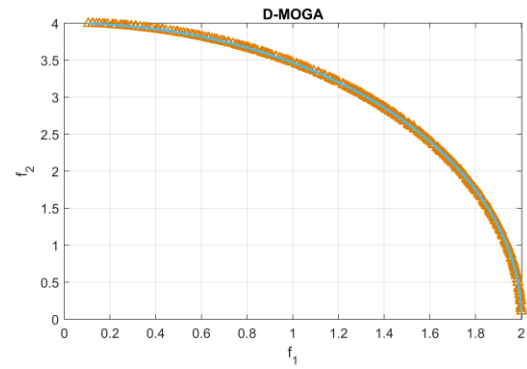
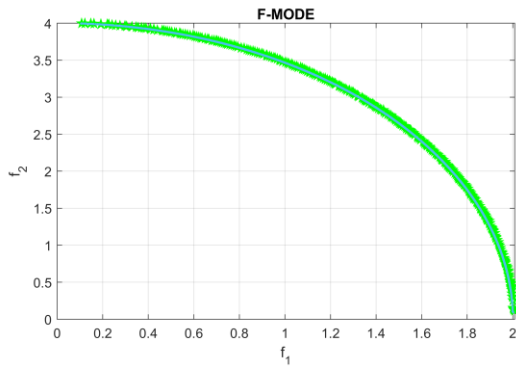
30,000 maksFeste Pareto Yüzeyler WFG7



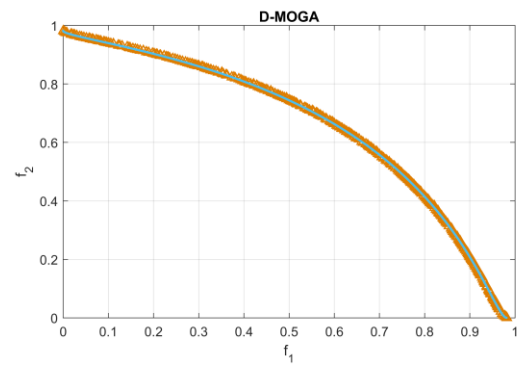
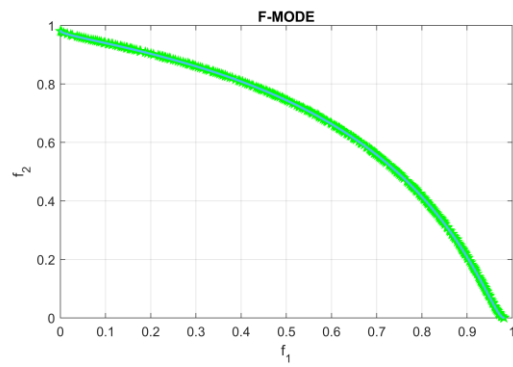
30,000 maksFeste Pareto Yüzeyler WFG8



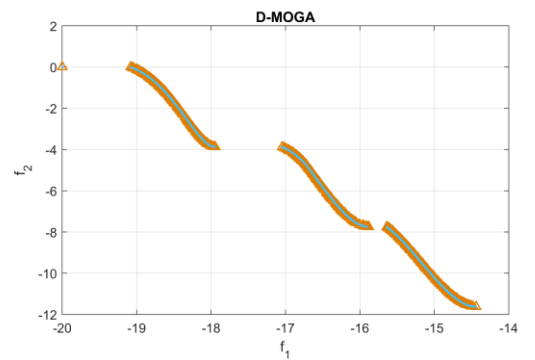
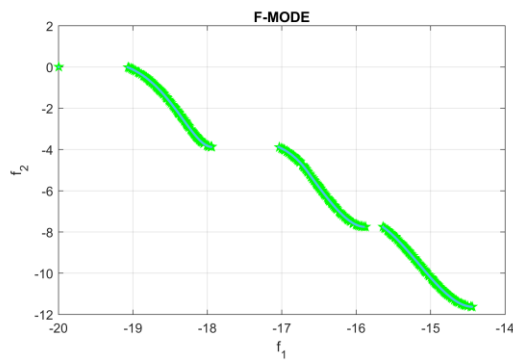
30,000 maksFeste Pareto Yüzeyler WFG9

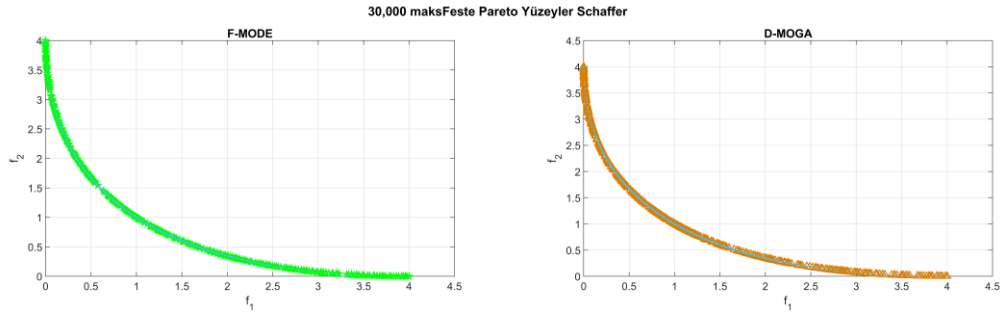


30,000 maksFeste Pareto Yüzeyler Fonseca



30,000 maksFeste Pareto Yüzeyler Kursawe





**EK-2.2. 30,000 maksFes F-MODE ve D-MOGA Kutu grafikler**

