

T.C.
BİLECİK ŐEYH EDEBALI ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĐİ ANABİLİM DALI

**GELİŐTİRİLMİŐ ŐAHİN ALGORİTMASININ
ÇOK AMAÇLI OPTİMİZASYON PROBLEMLERİNE UYARLANMASI**

YÜKSEK LİSANS TEZİ
MERYEM KUŐOĐLU

TEZ DANIŐMANI
PROF. DR. UĐUR YÜZGEÇ

BİLECİK, 2021
10421026

T.C.
BİLECİK ŐEYH EDEBALI ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĐİ ANABİLİM DALI

**GELİŐTİRİLMİŐ ŐAHİN ALGORİTMASININ
ÇOK AMAÇLI OPTİMİZASYON PROBLEMLERİNE UYARLANMASI**

YÜKSEK LİSANS TEZİ
MERYEM KUŐOĐLU

TEZ DANIŐMANI
PROF. DR. UĐUR YÜZGEÇ

BİLECİK, 2021
10421026

BEYAN

Geliştirilmiş Şahin Algoritmasının Çok Amaçlı Optimizasyon Problemlerine Uyarlanması adlı yüksek lisans/doktora/sanatta yeterlik tezi/dönem projesinin hazırlık ve yazımı sırasında bilimsel araştırma ve etik kurallarına uyduğumu, başkalarının eserlerinden yararlandığım bölümlerde bilimsel kurallara uygun olarak atıfta bulunduğumu, kullandığım verilerde herhangi bir tahrifat yapmadığımı, tezin herhangi bir kısmının Bilecik Şeyh Edebali Üniversitesi veya başka bir üniversitede başka bir tez çalışması olarak sunulmadığını, aksinin tespit edileceği muhtemel durumlarda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Bu çalışmanın, Bilimsel Araştırma Projeleri (BAP), TÜBİTAK veya benzeri kuruluşlarca desteklenmesi durumunda; projenin ve destekleyen kurumun adı proje numarası ile birlikte, ETİK KURUL onayı alınması durumunda ise ETİK KURUL tarih karar ve sayı bilgilerinin beyan edilmesi gerekmektedir.			
DESTEK ALINMIŞTIR	<input type="checkbox"/>	DESTEK ALINMAMIŞTIR	<input checked="" type="checkbox"/>
Destek alındı ise;			
Destekleyen kurum;			
Desteğin Türü		Proje Numarası	
1- BAP (Bilimsel Araştırma Projesi)			
2- TÜBİTAK			
Diğer;.....			
ETİK KURUL onayı var ise;			
ETİK KURUL karar tarih/sayı:	/.....	

Meryem Kuşoğlu

Tarih

.....

İmza

.....

ÖN SÖZ

Bu tez çalışmasının yazılmasında, çalışmamı sahiplenerek takip eden danışmanım Sayın Prof. Dr. Uğur YÜZGEÇ' e değerli katkı ve emekleri için teşekkürlerimi ve saygılarımı sunarım.

Son olarak bu günlere ulaşmamdaki emekleri adına değerli aileme teşekkür ederim.

Meryem Kuşoğlu

2021



ÖZET

GELİŞTİRİLMİŞ ŞAHİN ALGORİTMASININ

ÇOK AMAÇLI OPTİMİZASYON PROBLEMLERİNE UYARLANMASI

Bu çalışmada şahinlerden esinlenilmiş Harris Şahin Optimizasyonu (Harris Hawk Optimizer-HHO) algoritmasından ilham alınarak Karşıtlık Temelli Çok Amaçlı Harris Şahin Optimizasyon algoritması (Opposition learning-based Multi-Objective Harris Hawks Optimizer- OppMOHHO) geliştirilmiştir. İlk aşamada; HHO algoritmasının yapısı korunarak üzerine Pareto optimal sonuçları kayıt etmek ve yeniden kullanmak için bir arşiv yapısı eklenmiştir. Bu arşiv yapısı şahinlerin pozisyonlarını ve çözümleri kayıt etmek için kullanılmaktadır. Çözümlerin seçimi için rulet çarkı mekanizması kullanılmıştır. Seçim hedefin (avın) en iyi sonucu olarak güncellenmektedir. Böylece Çok Amaçlı Harris Şahin Optimizasyonu algoritması elde edilmiştir. İkinci aşamada; MOHHO algoritması üzerine karşıtlık öğrenme mekanizması eklenerek OppMOHHO algoritması geliştirilmiştir. Bu geliştirme ile algoritmanın başarımını yükseltmek hedeflenmiştir.

OppMOHHO algoritmasının başarımını değerlendirmek için literatürde bulunan kısıtlamasız ZDT ve DTLZ test fonksiyon serileri kullanılmıştır. Kullanılan on üç test fonksiyonu için OppMOHHO algoritması, geliştirilen MOHHO algoritması ve literatürde bulunan üç adet meta sezgisel optimizasyon algoritması karşılaştırılmıştır. Bu algoritmaların istatistiksel karşılaştırılması için Ters Nesil Mesafesi, Nesil Mesafesi, Aralık, Yayılma ve Maksimum Yayılma metriği kullanılmıştır.

Sonuçlar incelendiğinde OppMOHHO algoritması ZDT test fonksiyonu serisi için; ters nesil mesafe metriğinde OppMOHHO, nesil mesafesi metriğinde ilk üç içerisinde, aralık metriğinde ilk iki içerisinde, yayılma metriğinde ilk üç içerisinde ve maksimum yayılma metriğinde ilk iki içerisinde yüksek yakınsama davranışına sahip olduğunu nicel olarak göstermiştir. DTLZ test fonksiyonu için; ters nesil mesafe metriği için OppMOHHO, nesil mesafe metriği için ilk dört içerisinde, aralık metriği için ilk üç içerisinde, yayılma metriği için OppMOHHO ve maksimum yayılma metriği için de OppMOHHO algoritması yüksek yakınsama davranışına sahip olduğunu nicel olarak göstermiştir.

Anahtar Kelimeler: Çok Amaçlı Optimizasyon, Karşıtlık öğrenme, Harris Şahin Optimizasyon Algoritması, Ters Nesil Mesafe.

ABSTRACT

ADAPTATION OF IMPROVED HAWK ALGORITHM TO MULTI OBJECTIVE OPTIMIZATION PROBLEMS

In this study, Harris Hawks Optimization, which was developed with inspiration from hawks, have taken as an example. First stage; Preserving the structure of the HHO algorithm, an archive structure has been added to save and reuse Pareto optimal results. This archive is used to simulate hawk positions and solutions. The roulette wheel mechanism is used for the selection of solutions. The selection result is updated as the best result of the target. Thus, the Multi-Objective Harris Hawk Optimization algorithm has obtained. In the second stage; Multi Objective Harris Hawks Optimization algorithm based on Opposition Learning--based has been developed by adding the opposition learning mechanism on the MOHHO algorithm. It is aimed to increase the performance of the algorithm with the OppMOHHO algorithm.

In order to better see the performance of the OppMOHHO algorithm, the unrestricted ZDT and DTLZ test function series in the literature have used. The thirteen test functions used have compared with OppMOHHO and four metaheuristic optimization algorithms found in the literature. For the statistical comparison of these algorithms, the Inverted Generational Distance, Generational Distance, Spacing, Spread and Maximum Spread metric have calculated. The results are given with tables and graphs.

When the results are examined, for the OppMOHHO algorithm ZDT test function series; In the Inverted Generation Distance metric, OppMOHHO has quantitatively shown that it has high convergence behavior in the top three in the generation distance metric, in the top two in the interval metric, in the top three in the spread metric, and in the top two in the maximum spread metric. For DTLZ test function; The OppMOHHO algorithm for the inverted generation distance metric, the top four for the generation distance metric, the top three for the interval metric, the OppMOHHO for the spread metric and the OppMOHHO for the maximum spread metric have quantitatively shown that it has high convergence behavior.

Keywords: Multi Objective Optimizer, Opposition Learning, Harris Hawks Optimizer, Inverted Generational Distance.

İÇİNDEKİLER

	Sayfa
ÖZET	ii
İÇİNDEKİLER.....	iv
TABLolar LİSTESİ.....	vi
ŞEKİLLER LİSTESİ.....	vii
KISALTMALAR VE SİMGELER LİSTESİ	viii
1. GİRİŞ.....	1
2. MATERYAL VE YÖNTEM	5
2.1. Çok Amaçlı Optimizasyon Problemlerinin Tanımlanması.....	5
2.2. Pareto Tabanlı Yaklaşım.....	5
2.3. Harris Şahin Optimizasyonu	6
2.3.1. Keşif Aşaması	7
2.3.2. Keşiften Sömürüye Geçiş.....	7
2.3.3. Sömürü Aşaması	7
2.4. Çok Amaçlı Harris Şahin Optimizasyonu	11
2.5. Karşılıklı Öğrenme Tabanlı Çok Amaçlı Harris Şahin Optimizasyonu	13
3. DENEYSEL ÇALIŞMALAR	15
3.1. Başarım Metrikleri	15
3.1.1. Nesil Mesafe Metriği	15
3.1.2. Ters Nesil Mesafesi Metriği.....	16
3.1.3. Aralık Metriği.....	16
3.1.4. Yayılma Metriği	17
3.1.5. Maksimum Yayılma Metriği.....	17
3.2. Ayarlar	17
3.2.1. Çok Amaçlı Optimizasyon Algoritmaları	18

3.2.2. ZDT Test Fonksiyonu Serisi.....	18
3.2.3. DTLZ Test Fonksiyonu Serisi.....	19
3.3. ZDT Test Fonksiyonu Serisi İçin Karşılaştırmalı Sonuçları	21
3.4. DTLZ Test Fonksiyonu Serisi İçin Karşılaştırmalı Sonuçları.....	33
4. SONUÇ VE TARTIŞMA	48
KAYNAKÇA	49



TABLULAR LİSTESİ

Sayfa

Tablo 3.1. ZDT Test Fonksiyonu Serisi için Çok Amaçlı Optimizasyon Test Problemleri ...	18
Tablo 3.2. DTLZ Test Fonksiyonu Serisi için Çok Amaçlı Optimizasyon Test Problemleri	19
Tablo 3.3. OppMOHHO ve diğer algoritmaların ZDT fonksiyonu serisi için IGD metrik sonuçları	21
Tablo 3.4. OppMOHHO ve diğer algoritmaların ZDT fonksiyonu serisi için GD metrik sonuçları	29
Tablo 3.5. OppMOHHO ve diğer algoritmaların ZDT fonksiyonu serisi için S metrik sonuçları	30
Tablo 3.6. OppMOHHO ve diğer algoritmaların ZDT fonksiyonu serisi için Sp metrik sonuçları	31
Tablo 3.7. OppMOHHO ve diğer algoritmaların ZDT fonksiyonu serisi için MS metrik sonuçları	32
Tablo 3.8. OppMOHHO ve diğer algoritmaların DTLZ fonksiyonu serisi için IGD metrik sonuçları	33
Tablo 3.9. OppMOHHO ve diğer algoritmaların DTLZ fonksiyonu serisi için GD metrik sonuçları	43
Tablo 3.10. OppMOHHO ve diğer algoritmaların DTLZ fonksiyonu serisi için S metrik sonuçları	44
Tablo 3.11. OppMOHHO ve diğer algoritmaların DTLZ fonksiyonu serisi için Sp metrik sonuçları	45
Tablo 3.12. OppMOHHO ve diğer algoritmaların DTLZ fonksiyonu serisi için MS metrik sonuçları	46

ŞEKİLLER LİSTESİ

	Sayfa
Şekil 2.1. Baskın ve baskılanmış şekil gösterimi.....	6
Şekil 2.2. Harris şahin optimizasyonunun aşamaları	6
Şekil 2.3. HHO'nun Sert kuşatma durumunda genel vektörlerin örneği	8
Şekil 2.4. Aşamalı Hızlı Dalışlarla Yumuşak Kuşatma genel vektörlerin örneği.....	9
Şekil 2.5. 2D ve 3D alanda aşamalı hızlı dalışlar ile sert kuşatma durumunda genel vektör örneği	10
Şekil 3.1. ZDT1 fonksiyonu için pareto optimal grafikleri.....	23
Şekil 3.2. ZDT2 fonksiyonu için pareto optimal grafikleri.....	24
Şekil 3.3. ZDT3 fonksiyonu için pareto optimal grafikleri.....	25
Şekil 3.4. ZDT4 fonksiyonu için pareto optimal grafikleri.....	26
Şekil 3.5. ZDT5 fonksiyonu için pareto optimal grafikleri.....	27
Şekil 3.6. ZDT6 fonksiyonu için pareto optimal grafikleri.....	28
Şekil 3.7. ZDT test fonksiyon serisi için IGD metriği istatistiksel sonuçları	29
Şekil 3.8. DTLZ1 fonksiyonu için pareto optimal grafikleri	35
Şekil 3.9. DTLZ2 fonksiyonu için pareto optimal grafikleri	36
Şekil 3.10. DTLZ3 fonksiyonu için pareto optimal grafikleri	37
Şekil 3.11. DTLZ4 fonksiyonu için pareto optimal grafikleri	38
Şekil 3.12. DTLZ5 fonksiyonu için pareto optimal grafikleri	39
Şekil 3.13. DTLZ6 fonksiyonu için pareto optimal grafikleri	40
Şekil 3.14. DTLZ7 fonksiyonu için pareto optimal grafikleri	41
Şekil 3.15. DTLZ test fonksiyon serisi için IGD metriği istatistiksel sonuçları.....	43

KISALTMALAR VE SİMGELER LİSTESİ

- HHO** : Harris Şahin Optimizasyonu (Harris Hawks Optimizer)
- MOHHO** : Çok Amaçlı Harris Şahin Optimizasyonu (Multi Objective Harris Hawks Optimizer)
- OppMOHHO** : Karşıtlık Öğrenmesine Dayanan Çok Amaçlı Harris Şahin Optimizasyon (Opposition learning-based Multi-Objective Harris Hawks Optimizer)
- MOALO** : Çok Amaçlı Karınca Aslanı Optimizasyonu (Multi Objective Ant Lion Optimizer)
- MODA** : Çok Amaçlı Yusufçuk Optimizasyonu (Multi Objective Dragonfly Optimizer)
- MOMVO** : Çok Amaçlı Çok Hedefli Optimizasyon (Multi Objective Multi Verse Optimizer)
- MOPSO** : Çok Amaçlı Parçacık Sürüsü Optimizasyon (Multi Objective Particle Swarm Optimizer)
- IGD** : Ters Nesil Mesafesi (Inverted Generational Distance)
- GD** : Nesil Mesafesi (Generational Distance)
- S** : Aralık (Spacing)
- Sp** : Yayılma (Spread)
- MS** : Maksimum Yayılma (Maximum Spread)

1. GİRİŞ

Optimizasyon; bir problemin karşısında belirli kurallar çerçevesinde en iyi sonucu veya çözümü elde etmektir. Bir başka deyişle istenilen çıktıyı elde edebilmek için girdilerin ne olacağına belirlenme sürecidir. Bu süreçte değeri belirlenecek olan öge karar değişkeni olarak adlandırılmaktadır. Karar değişkeninin değerinin belirlenmesi için bazı kısıtların sağlanması gerekmektedir. Bu kısıtları sağlayan çözüm Uygun Çözüm olarak isimlendirilmektedir. Uygun çözümler arasında belirlenen amaca göre en iyi olan çözüm Optimum Çözüm olarak adlandırılmaktadır. Problemlerin çözümü aşamasında farklı teknikler kullanılmaktadır. Optimizasyon işlemlerinde kullanılan analitik yöntemler tekniği, toleranslar dahilinde optimum sonucu vermektedir. Dezavantajı olarak boyutu büyük olan problemlerde, sonuç almak çok uzun sürmekte ya da alınamamaktadır. Sonuç alabilmek için farklı yöntemler kullanılmaktadır.

Sezgisel yöntemler tekniği, probleme özgüdür ve belirli bir algoritma sürecini takip etmektedir. Analitik yöntemlere göre hızlı olmasının yanında çözüm garantisi yoktur. Meta sezgisel yöntemler ise belirli algoritmanın problemin yapısına uyarlanması ile elde edilmektedir. Meta sezgisel algoritmalar doğadan esinlenmekte, popülasyon tabanlı, dinamik, tek ya da çok amaçlı yapılara uyum sağlama ve hafıza kullanma yapılarına sahiptir (Shadkam ve Bijari, 2014 : 9). Geliştirilen optimizasyon sistemleri tek ya da çok amaçlı olarak sınıflandırılmaktadır. Problemler sınırlı, sınırsız, sürekli ve ayrık olarak sınıflandırılmaktadır. Matematiksel yöntemler (Heidari ve diğerleri, 2019 : 23) ile elde edilen sınıflandırma yetersiz kaldığı için iteratif ve popülasyon tabanlı olarak iki farklı (Talbi, 2009) metodu olan meta sezgisel algoritmalar bu sınıflandırmaların bir kısmını karşılamaktadır (Dréo, Pétrowski, Siarry, ve Taillard, 2006). Popülasyon temelli algoritmalar doğadaki canlıları taklit ederek geliştirilmiştir (Mirjalili, 2018 : 12; Karaboga ve Basturk, 2008 : 10; Mirjalili, 2015 : 21).

Gerçek hayatta karşılaşılan problemler tek hedef ya da birden fazla hedef içermektedir. Probleme göre bir hedef (amaç) en üst seviyeye çıkarmak istenirken başka bir hedef en aza indirilmek istenebilmektedir (MathWorks, 2019). Bu amaçlar birbirleri ile çelişmekte ve zorluk seviyesi artmaktadır (Kaya ve Fıglalı, 2016 : 9). Bu yüzden çözümü için çok amaçlı optimizasyon sistemleri kullanılmaktadır. Çok amaçlı optimizasyon problemlerinde her bir amaç için maliyet fonksiyonu hesaplanmaktadır. Bu hesap sonucunda baskın olmayan çözümler belirlenmektedir. 1990'lı yıllardan bu yana kullanılan baskınlık kavramı sayesinde çözümler değerlendirilmekte ve kıyaslanmaktadır (Akbel ve Kahraman, 2020 : 14; Ke, Zhang, ve Battiti, 2014 : 12; Ishibuchi ve Murata, 1996 : 5; Zhang, Tian , Cheng, ve Jin, 2014 : 12; Serafini, 1994).

Heidari ve arkadaşları (Heidari ve diğerleri, 2019 : 23) Harris Şahin Optimizasyon (HHO) algoritmasını şahinlerin avlanma stratejilerini gözlemleyerek geliştirmişlerdir. Bu algoritma popülasyon temelli olmasının yanı sıra türevsel işlemler içermemektedir. HHO algoritmasını üç ana aşama ile geliştirmişlerdir. Bunlar, avı bulma, sürpriz saldırı ve saldırı şeklinin belirlenmesidir.

Literatürde Milad (Eshkevari, 2020) tarafından geliştirilen MOHHO algoritması bulunmaktadır. Milad geliştirdiği algoritmayı çok amaçlı parçacık sürüsü optimizasyon algoritmasından (Multi Objective Particle Swarm Optimizer, MOPSO) esinlenmiştir. Heidari'nin algoritmasına ek olarak MOPSO algoritmasının yapısında bulunan lider seçme işlemi gerçekleştirmiş ve geliştirdiği algoritmada lider şahin belirlemiştir. Çalışmasının devamında optimizasyonu bu çerçevede geliştirmiştir.

Fu ve Lu tarafından (Fu ve Lu, 2020 : 17) yapılan Güçlendirilmiş Çok Amaçlı Harris Şahin Optimizasyonu ile Hibrit Stratejilere Dayalı Hidrolik Türbin Yönetim Sistemleri için FOPID Kontrolörünün Çok Amaçlı Optimum Kontrolü adlı çalışmada farklı koşullarda HTGS'nin parametrelerini ayarlamayı geliştirebilmek için, FOPID kontrolörünü çok amaçlı optimizasyon problemi, yüksüz bozulma ve yükte bozulma çalışma koşulları göz önüne alınarak, her iki çalışma koşulundaki ITAE başarımlarını göstergelerinin amaç fonksiyonları olarak kullanıldığı durumlarda oluşturmuşlardır. Çalışmalarında yeni bir MOHHO algoritması versiyonu geliştirmişlerdir. Ardından MOHHO'nun küresel arama kapasitesi, MOHHO 'yu Latin hiperküp örnekleme başlatma, değiştirilmiş diferansiyel evrim operatörü ve mutasyon operatörünü içeren melez stratejilerle (HMOHHO) birleştirerek ve tavşan enerjisinin doğrusal modelini doğrusal olmayan bir modelle değiştirerek büyük ölçüde geliştirmişlerdir. Geliştirme sonrasında önerdikleri HMOHHO algoritmasının etkinliğini doğrulayan NSGAIII, MOGWO, MOPSO ve MOHHO ile karşılaştırmışlardır. Pratik uygulama ve karşılaştırmalı analiz, FOPID kontrolörünün yapılandırılmış çok amaçlı probleminin, değişken çalışma koşulları altında HTGS için uygun olduğunu ve önerilen HMOHHO algoritmasının tüm algoritmalar arasında olağanüstü üstünlüğe sahip olduğunu göstermişlerdir.

Hossain ve diğerleri (Hossain, ve diğerleri, 2021 : 12) yaptıkları çalışmada bilişsel radyo tabanlı araç ad hoc ağı (CR-VANET) için bir yönlendirme protokolü önermişlerdir. Spektrum algılamasından sonra araç boş spektrumu bulacaktır. Yol kenarı birimi (RSU) daha sonra bu boş kanalı araçlara atayacaktır. Veri iletimi için biyo-ilhamlı 2-Hop Çok Amaçlı Harris Şahin Optimizasyonu (2HMO-HHO) algoritması tarafından seçilen atanan kanaldaki en uygun iki sekme komşuları aracılığıyla gerçekleştirmişlerdir. Önerdikleri modeli benzetim yolu ile test

etmişler ve önceki çalışmalarına göre daha iyi sonuç almışlardır. Daha iyi verim, daha yüksek paket teslim oranı, daha az paket kayıp oranı, daha düşük gecikme ve daha düşük ağ yükü elde etmişlerdir.

Ewees ve Elaziz (Ewees ve Elaziz, 2020 : 16) yaptıkları çalışmada önerdikleri algoritmada (CMVHHO) temel MVO'nun tüm rastgele parametreleri üzerinde en iyi haritayı ve uygun parametreyi seçmek için on farklı KAOS haritası ile test etmişlerdir. Önerdikleri algoritmayı global optimizasyon problemlerinin minimum değerlerini bulmada değerlendirmişlerdir. Bunun yanı sıra dört mühendislik problemini çözmeye test etmişlerdir. Sonuç olarak önerdikleri algoritmanın global optimal çözümleri aramada en iyi sonuçları gösterdiğini, yakınsama ve istatistiksel analiz açısından yüksek başarımlar gösterdiğini özetlemişlerdir. Ayrıca önerdikleri CMVHHO dört mühendislik problemini çözmek için başarıyla uygulamışlar ve en son yöntemlere kıyasla en iyi sonuçları gözlemlemişlerdir. Deneysel sonuçlar, CMVHHO'nun başarımlarını iyileştirdiği için Kaotik Daire haritasının tüm haritalar arasında en iyi harita olduğu sonucuna varmışlar, ayrıca HHO önerilen algoritma davranışında olumlu etkilendiğini gözlemlemişlerdir. Önerdikleri CMVHHO'nun sınırlaması, temel MVO'dan daha az CPU zamanı almasına rağmen, PSO ve GA algoritmalarından daha fazla CPU zamanı tüketmesidir.

Dieu Tien Bui (Bui , ve diğerleri, 2019 : 22), çalışmasında heyelan duyarlılık haritasının mekânsal modellemesinde hesaplama eksikliğinin üstesinden gelebilmek için HHO algoritmasını yapay sinir ağı ile sentezleyerek kullanmıştır. HHO-ANN algoritması tarafından üretilen heyelan duyarlılık haritası, görünmeyen heyelan olaylarını tahmin etme açısından yapay sinir ağı haritasından daha başarılı olduğunu göstermiştir.

Hao Chen (Chen, 2020 : 23), çalışmasında başarımlarını arttırmak için HHO algoritmasına kaos stratejisi, çok nüfus mekanizmasını ve diferansiyel evrim (DE) stratejilerini eklemiştir. Böylece HHO popülasyonunun çeşitliliğini artırmıştır (CMDHHO). Genel olarak CMDHHO, HHO'nun temel küresel ve yerel arama kapasitelerini önemli ölçüde geliştirmiştir.

N. A. Golilarz (Golilarz N. A., 2019 : 7), CCP (dokuz kontrol grafiği modeli) modellerinin tanınması için önerilen ConvNet (evrişimli bir sinir ağı) ve optimizasyon algoritmasına dayalı yeni bir otomatik yöntem önermiştir. HHO-ConvNet kalite ve başarımlar analizini görmek için birkaç deney ile gerçekleştirilen diğer yöntemlerle karşılaştırmıştır. Önerdiği CCP sistemi 99.80 doğru sınıflandırma kaydetmiştir.

N. Amiri Golilarz (Golilarz N. A., 2019 : 9), çalışmasında görüntü üzerindeki parazit ve gürültüleri gidermek için HHO algoritmasını kullanmıştır. Elde ettiği sonuçları alternatif sonuçlar ile karşılaştırmıştır. HHO algoritmasının görsel ve kantitatif olarak JADE algoritmasından çok daha iyi başarımlar gösterdiğini kaydetmiştir.

Vikram Kumar Kamboj (Kamboj, 2020 : 15), mevcut HHO algoritmasına Sinüs-Kosinüs algoritması kullanarak başarılı bir şekilde yükseltmiştir. Geliştirdiği hibrit hHHO-SCA algoritmasını sürekli, ayrık, kısıtlı, doğrusal olmayan ve dış bükey mühendislik tasarımı üzerinde test etmiştir. Karşılaştırdığı diğer algoritmalar karşısında başarılı olduğunu gözlemlemiştir.

X. Bao (Bao, 2019 : 17), renkli görüntü segmentasyonu için HHO ve DE algoritmalarının hibridizasyonundan esinlenip alternatif bir yöntem önermiştir. Otsu yöntemi, Kapur entropisini ve evrensellik değerlerini kullanmıştır. Çeşitli görüntüler üzerinde yaptığı çalışmanın iyi sonuçlar elde etmiştir.

Akdağ O (Akdağ, 2020 : 9), aktif güç kayıplarının minimizasyonu için yaptığı çalışmada HHO algoritmasını Optimum Yük Akış problemine uygulamıştır. Çalışmasında aktif güç kaybını 22.68MW'a düşürmüştür.

Islam, M. Z. (Islam, 2020 : 26), HHO tabanlı tek ve çok amaçlı çevresel emisyonu dikkate alan optimal güç akışı üzerine yaptığı çalışmasında toplam yakıt maliyeti, aktif güç kaybı ve çevresel emisyon maliyetini telafi etmeye çalışmıştır. Elde ettiği sonuçları Balina Optimizasyon Algoritması, Salp Sürü Algoritması, Güve Alevi ve Ateş Solucanı Optimizasyon algoritmaları ile kıyasladığında HHO algoritmasının daha başarılı olduğunu kaydetmiştir.

Bu çalışma kapsamında HHO algoritmasının ilk olarak MOMVO algoritmasından esinlenerek yeni bir MOHHO algoritması versiyonu geliştirilmiştir. İlk olarak boyutunu önceden belirlediği, mevcut iterasyondaki seçilen baskın olmayan pareto optimal çözümleri tutmak için bir arşiv yapısı tasarlanıp algoritmaya eklenmiştir. Her bir iterasyonda baskın olmayan çözümler ile arşivi karşılaştırılmış ve arşivde güncelleme yapılmıştır. Ayrıca arşiv dolduğunda yeni çözümleri saklayabilmek için en kalabalık nüfusun silinme olasılığının ortadan kaldırıldığını kabul edilmiştir. Çözümlerin seçimi için rulet çarkı yöntemini kullanılmıştır. Diğer parametrelerde de gerekli düzenlemeler yapılmıştır. Çalışmanın devamında popülasyondaki çözümlerin kalitesini arttırmak için Karşıtlık Öğrenme Mekanizması kullanılmıştır. Sonuç olarak Karşıtlık Öğrenmesine Dayanan Çok Amaçlı Harris Şahin Optimizasyonu (OppMOHHO) geliştirmiştir.

2. MATERYAL VE YÖNTEM

2.1. Çok Amaçlı Optimizasyon Problemlerinin Tanımlanması

Çok amaçlı optimizasyonlarda amaç (Deb K. , 2011), çeşitliliği en fazla olan Pareto optimal çözümlerin doğru bir şekilde elde etmektir (Deb K. P., 2002 : 15).Çok amaçlı problemler üzerinde çalışırken tüm sonuçlar arasında en iyi olanı bulmayı hedeflemektedir. En iyi sonucu bulma aşamasında maksimizasyon ya da minimizasyon yapılmaktadır. Matematiksel formülasyonu aşağıdaki gibidir (Abdel-Basset, 2021 : 28).

$$\text{Minimum } F(x) = \{ f_1(x), f_2(x), f_3(x), \dots, f_m(x) \}, m \geq 2 \quad (2.1)$$

$$nq_i(x) \geq 0, i = 1, 2, \dots, z \quad (2.2)$$

$$q_i(x) = 0, i = 1, 2, 3, \dots, k \quad (2.3)$$

$$L_i \leq x_{i1} \leq U_i, i = 1, 2, 3, \dots, p \quad (2.44)$$

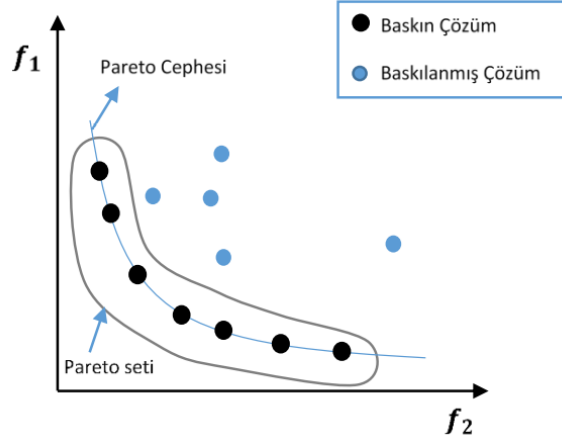
Yukarıdaki denklemlerde; m , hedef sayısını, q_i eşitsizlik kısıtlaması, k eşitsizlik kısıtlamasının sayısını, p değişken sayısını, L alt sınırı ve U üst sınırı göstermektedir.

2.2. Pareto Tabanlı Yaklaşım

Vilfredo Frederico Damaso tarafından 1900'lü yılların başında bulunduğu bir terimdir. Bu terim sayesinde tek amaçlı optimizasyon problemlerinde arama uzayı tek boyutlu olduğundan ortaya çıkan çözümlerin sayısal değerlerini karşılaştırarak en iyi çözüm sunulabilmektedir. Bu yaklaşım çok amaçlı optimizasyon problemlerinde ise pareto çözüm olabilmesi için amaçların herhangi biri için en kötü olmayan ve en azından bir amaç için diğerlerinden daha iyi olması gerekmektedir. Başka bir deyişle baskın olmayan çözümler en iyi olarak adlandırılmaktadır. Baskınlık kavramını detaylandırmak gerekirse arama uzayında x_1 birinci çözüm adayı, x_2 ikinci çözüm adayı olduğu varsayılır ise ve koşullar aşağıdaki gibi ise;

- x_2 çözüm adayının tüm sonuçları x_1 'den kötü
- x_1 çözüm adayının en az bir amaç fonksiyonunda x_2 'ye göre daha iyi değerlere sahip olması durumunda x_1 çözümü x_2 'yi baskılıyor şeklinde değerlendirilmektedir (Akbel ve Kahraman, 2020 : 14).

Baskın ve baskılanmış grafik gösterimine örnek Şekil 2.1.'de verilmiştir.

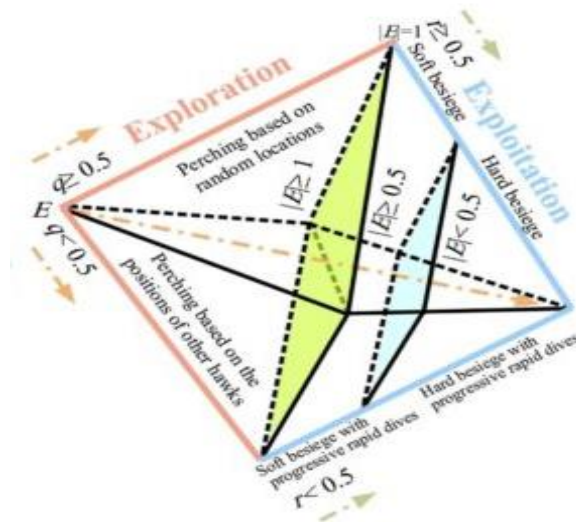


Şekil 2.1. Baskın ve baskılanmış şekil gösterimi

Kaynak: Akbel ve Kahraman, 2020 : 14

2.3. Harris Şahin Optimizasyonu

HHO algoritması, şahinlerin doğadaki davranışlarından etkilenmiş yani popülasyon tabanlı ve gradyen içermemektedir (Heidari, 2019). Herhangi bir problem karşısında HHO algoritması keşif, keşiften sömürüye geçiş ve sömürü aşamalarından oluşmaktadır (Ismael, 2020 : 14). HHO algoritmasının ana mantığı, Harris'in şahinlerinin doğada "sürpriz sıçrama" olarak adlandırılan işbirlikçi davranış ve kovalama tarzlarından esinlenilmiştir. Şekil 2.2'de HHO algoritmasının tüm aşamaları gösterilmektedir.



Şekil 1.2. Harris şahin optimizasyonunun aşamaları

Kaynak: Heidari, ve diğerleri, 2019 : 23

2.3.1. Keşif Aşaması

Şahinlerin avlarını tespit ettiği ve gözlemlediği aşamadır. Doğada şahinlerin avlarını bulmaları zaman alabilmektedir. Böyle durumda şahinlerin bir yere konumlanıp beklemeleri gerekmektedir. Bu işlemlerin modellenmiş hali aşağıda verilmiştir (Mirjalili S. , 2020; Jia, 2019 : 35).

$$X(t + 1) = \begin{cases} X_{rand}(t) - r_1 | X_{rand}(t) - 2r_2X(t) | & q \geq 0.5 \\ (X_{rabbit}(t) - X(t)) - r_3(LB + r_4(UB - LB)) & q < 0.5 \end{cases} \quad (2.5)$$

$$X_m(t) = \frac{1}{N} \sum_{i=1}^N X_i(t) \quad (2.6)$$

Denklem 2.5.'de bulunan; $X_{rand}(t)$ mevcut popülasyondan rastgele seçilmiş bir şahin, $X(t + 1)$, bir sonraki iterasyonda t şahinlerinin pozisyon vektörü, $X(t)$ şahinlerin mevcut konumu, $X_{rabbit}(t)$, tavşanın yani avın konumunu göstermektedir. r_1, r_2, r_3, r_4 ve q , $[0,1]$ arasında rastgele sayılardır. LB ve UB belirlenen optimizasyon probleminin üst ve alt sınırıdır. x_m popülasyonun ortalama konumunu temsil etmektedir. $X_{rabbit}(t)$ seçilen rastgele bir şahindir. Denklem 2.6.'daki gibi hesaplanmaktadır. Formülasyondaki N popülasyon boyutu, $x_m(t)$ iterasyondaki her bir bireyin konumudur.

2.3.2. Keşiften Sömürüye Geçiş

Meta sezgisel algoritmaların başarımını önemli ölçüde etkileyen bu aşamada E değişkeni bulunmaktadır. E değişkeni hedefin (tavşanın) kaçan enerjisini göstermektedir. Keşiften sömürü aşamasına geçilirken kullanılmakta ve her bir iterasyondan sonra azalmaktadır. Denklemi;

$$E = 2E_0 * \left(1 - \frac{t}{T}\right) \quad (2.7)$$

E_0 , $[-1,1]$ aralığında rastgele bir sayıdır. t mevcut iterasyonu ve T maksimum iterasyonu göstermektedir. $|E| \geq 1$ keşif aşaması avı aramak için kullanılır; $|E| < 1$ ise, gelecek vaat eden alandan yararlanmak için kullanılmaktadır.

2.3.3. Sömürü Aşaması

Şahinlerin saldırıya geçtiği aşamanın modellenmesidir. Bu aşamada dört farklı yöntem bulunmaktadır. Avın kaçıp kaçamayacağını belirlemek için rastgele bir sayı (r) kullanılır. $r < 0.5$ durumu avın kaçtığını, $r \geq 0.5$ avın kaçmadığını gösterir. E (avın kaçan enerjisi) şahinin

davranışını etkiler. Eğer $|E| \geq 0.5$ ise, yumuşak kuşatma; $|E| < 0.5$ ise sert kuşatma gerçekleştirirler.

Yöntemlerden biri olan yumuşak kuşatma; avın kaçmak için enerjisinin olduğu varsayılan bu aşamada şahinlerin avın enerjisini kaçırmak için yanıtacak saldırılar yapmaktadır. Matematiksel modeli;

$$X(t + 1) = \Delta X(t) - E |JX_{rabbit}(t) - X(t)| \quad (2.8)$$

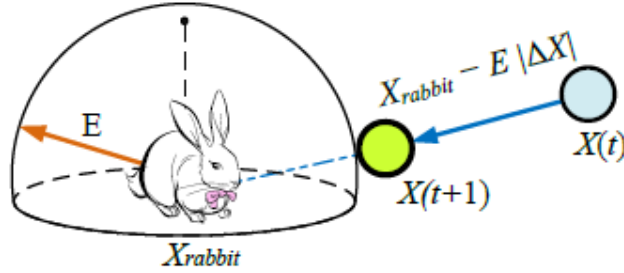
$$\Delta X(t) = X_{rabbit}(t) - X(t) \quad (2.9)$$

Formülasyondaki; $\Delta X(t)$, X tavşanın konum vektörü ile t yinelemesindeki geçerli konum arasındaki farkı, $X(t)$ tavşanın konum vektörünü, t iterasyonu göstermektedir. J, tavşanın doğadaki hareketini simüle etmek içindir ve her iterasyonda rastgele hesaplanır.

Bir başka yöntem sert kuşatma; avının enerjisini iyice azaltan şahinlerin, avı yakalamak için sürpriz pençe hareketi için hiç kuşatma yapmasına gerek kalmadığı aşamadır. Denklem 2.9.'daki gibidir.

$$X(t + 1) = X_{rabbit}(t) - E |\Delta x(t)| \quad (2.10)$$

Bu adımın bir Şahinli basit bir örneği Şekil 2.3'de gösterilmektedir.



Şekil 2.3. HHO'nun Sert kuşatma durumunda genel vektörlerin örneği

Kaynak: Heidari, ve diğerleri, 2019 : 23)

Diğer bir yöntem de Aşamalı Hızlı Dalışlarla Yumuşak Kuşatma; Bu aşamada avın hala enerjisi olduğu için şahinlerin ilk olarak yumuşak kuşatma yapması gerekmektedir. Denklem 2.11.'de bulunan LF tabanlı modele göre dalış yapılıp yapılmayacağı tahmin edilmektedir. Problemin boyutu D olarak gösterilir. S, 1xD boyutuna göre rasgele bir vektördür. LF, levy fight fonksiyonu kullanılarak hesaplanmaktadır.

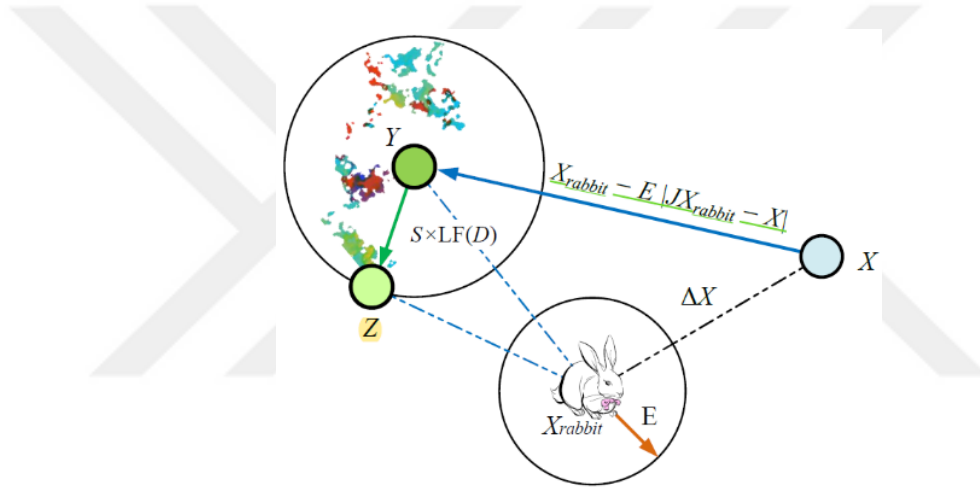
$$Y = X_{rabbit}(t) - E |JX_{rabbit}(t) - X(t)| \quad (2.11)$$

$$Z = Y + SxLF(D) \quad (2.12)$$

$$LF(x) = 0.01x \frac{ux\sigma}{|v|^{\frac{1}{\beta}}}, \quad \sigma = \left(\frac{\Gamma(1 + \beta) + \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1 + \beta}{2}\right) x\beta x 2^{\frac{\beta-1}{2}}}\right) \quad (2.13)$$

$$X(t + 1) = \begin{cases} Y & \text{if } F(Y) < F(X(t)) \\ Z & \text{if } F(Z) < F(X(t)) \end{cases} \quad (2.14)$$

β , 1.5 değerindedir. u ve v (0,1) aralığında rasgele bir sayıdır. Yumuşak kuşatma aşamasında şahinlerin yerini güncellemek için denklem 2.10. kullanılır. Buradaki Y ve Z eşitliklerini hesaplanırken 2.10. ve 2.11. denklemler kullanılır. Şahin için bu adımın basit bir görsele dönüştürülmüş hali Şekil 2.4'te verilmiştir.



Şekil 2.4. Aşamalı Hızlı Dalışlarla Yumuşak Kuşatma genel vektörlerin örneği
Kaynak: Heidari, ve diğerleri, 2019 : 23

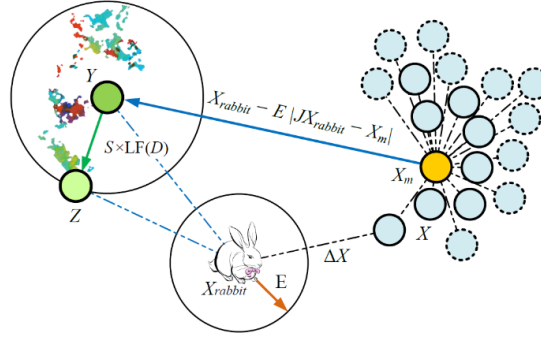
Son olarak aşamalı hızlı dalışlarla sert kuşatma yöntemi; bu aşamada avın enerjisi bitmiş ya da yoktur. Şahinin avını yakalamak için sert kuşatma yaptıktan sonra sürpriz pençe hareketi ile avını yakalamaktadır. Formülasyonu;

$$X(t + 1) = \begin{cases} Y & \text{if } F(Y) < F(X(t)) \\ Z & \text{if } F(Z) < F(X(t)) \end{cases} \quad (2.15)$$

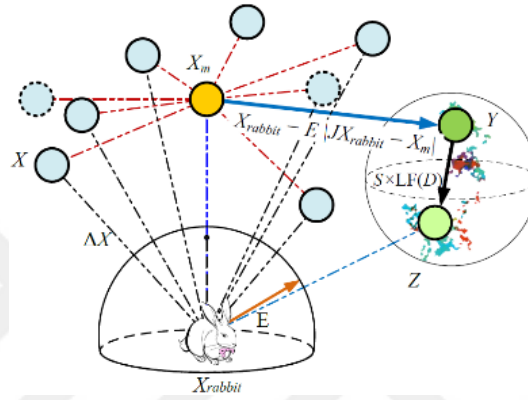
$$Y = X_{rabbit}(t) - E |X_{rabbit}(t) - X_m(t)| \quad (2.16)$$

$$Z = Y + SxLF(D) \quad (2.17)$$

Buradaki $x_m(t)$ değeri 2.5. denklem kullanılarak elde edilir. Bu aşamada bahsedilen işlemlerin basit bir örneği Şekil 2.5.'de gösterilmektedir.



(a) 2D uzayda



(b) 3D uzayda

Şekil 2.5. 2D ve 3D alanda aşamalı hızlı dalışlar ile sert kuşatma durumunda genel vektör örneği

Kaynak: Heidari, ve diğerleri, 2019 : 23

HHO algoritmasının sözde kodu aşağıda bulunmaktadır (Heidari, ve diğerleri, 2019 : 23) .

Giriş: Popülasyon boyutu (N), maksimum iterasyon (T) ve maksimum boyut belirlenmesi

Çıkış: Hedefin pozisyonu ve en iyi sonuç

Başlangıç popülasyonu X_i ($i = 1, . N$)

While (Durdurma koşuluna kadar) do

 Şahinlerin uygunluk değerleri hesaplanır

 Hedeflerin en iyi konumunu hesaplanır (X_{rabbit})

for (Her şahin için (X_i)) do

 /*Hedefin enerjisini güncellenir (E_0) ve Atlama aralığının hesaplanır (J)*/

 E metriğinin hesaplanır (Denklem (2.7.))

```

if |E|≥1
    /* konum vektörünü güncellenir (Denklem(2.5.)) */
else if |E|<1
    if r≥0.5 & |E|≥0.5
        /* konum vektörünü güncellenir (Denklem (2.8.)) */
    son
    if r≥0.5 & |E|<0.5
        /* konum vektörünü güncellenir (Denklem (2.9.)) */
    son
    if r<0.5 & |E|≥0.5
        /* konum vektörünü güncellenir (Denklem.( 2.14.)) */
    son
    if r<0.5 & |E|<0.5
        /* konum vektörünü güncellenir (Denklem(2.15.)) */
Son
Son

```

2.4. Çok Amaçlı Harris Şahin Optimizasyonu

Geliştirilen MOHHO algoritmasında, HHO algoritmasının yapısı korunarak çok amaçlı işlev kazandırmak için bir arşiv ve rulet çarkı seçim yöntemi eklenmiştir. Algoritmada popülasyon boyutunu (N), maksimum iterasyon (T) ve arşiv boyutunu (S) ayarladıktan sonra değişkenler rastgele aralıkta başlatılmıştır. Her bir şahin için $f_1(x)$ ve $f_2(x)$ değerlerini hesaplamıştır. Mevcut iterasyonda baskın olmayan çözümleri seçmiş ve böylece tavşanın pozisyonu arşivin daha az nüfuslu bir alandan rulet çarkı yöntemi ile çözüm seçmek için kalabalık mesafeden yararlanan lider seçim mekanizması tarafından elde etmiştir. Elde edilen çözüme yakın çözümlerin hesaplanması için aşağıdaki formülasyonu kullanmıştır.

$$P_i = \frac{N_i}{c}, c > 1 \quad (2.18)$$

Yukarıdaki formülasyon da c bir sabiti göstermektedir. N_i , i . çözümün yakınındaki çözümlerin sayısıdır. Sonrasında her bir şahin için yeni değerler hesaplayıp baskın olmayan çözümleri bulup arşivlemiştir. Sonuç olarak da arşivin çıktısını almıştır.

Geliştirilmiş olan MOHHO algoritmasının sözde kodu aşağıda bulunmaktadır.

Giriş: N, T, S başlangıç değer verme

Çıkış: Hedefin pozisyonu ve en iyi sonuç

Başlangıç popülasyonu X_i ($i = 1, . N$)

For $i:1 \leq T$

 for $i:1 \leq N$

 Başlangıç değerleri hesaplanır

 son

 /* Arama alanı dışında herhangi bir sonuç varsa değiştirilir */

 /* İlgili uygunluk değerlerini hesaplanır */

 /* Baskın olmayan çözümler bulunur. */

 /* Sahip olunan hâkim olmayan çözümlere göre arşiv güncellenir */

Eğer arşiv dolu ise

/* Yenilerini güncellemek için arşivdeki çeşitli çözümler kaldırılır. */

 /* Bir rulet çarkı ve Denklem (2.18.) uygulanır. */

son

 Eğer arşiv dolu ise

 /* Sınırlar yeni sonuçlar ile güncellenir*/

 Son

 for $i:1 \leq N$

 /*Hedefin kaçan Enerjisi güncellenir */

 /*arşivden rasgele şahin seçilir*/

 /*Rulet çarkı yöntemi ile en iyi çözüm seçilir */

 if $|E| \geq 1$

 /* konum vektörünü güncellenir (Denklem(2.5.)) */

 else if $|E| < 1$

 if $r \geq 0.5$ & $|E| \geq 0.5$

 /* konum vektörünü güncellenir (Denklem (2.8.)) */

 son

 if $r \geq 0.5$ & $|E| < 0.5$

 /* konum vektörünü güncellenir (Denklem (2.9.)) */

 son

 if $r < 0.5$ & $|E| \geq 0.5$

 /* konum vektörünü güncellenir (Denklem.(2.14.)) */

 son

 if $r < 0.5$ & $|E| < 0.5$

 /* konum vektörünü güncellenir (Denklem(2.15.)) */

Son
Son
Son for
Son for

2.5. Karşıtlık Öğrenme Tabanlı Çok Amaçlı Harris Şahin Optimizasyonu

Geliştirilen OppMOHHO algoritmasında karşıtlık öğrenme yapısını, başlangıçta popülasyonun çözümlerinin kalitesini arttırmak için kullanılmıştır. Bu mekanizma orijinal çözüm ile karşıt çözüm arasında iki yönlü arama yapmaktadır. Karşıtlık öğrenme mekanizması tüm çözümlerin arasından en uygununu almaktadır (Tubishat, 2020 : 10; Rahnamayan, 2006 : 7). $x \in [a, b]$ tanımlı x reel sayısı için karşıt noktaları denklem 2.19.'daki gibi tanımlanmaktadır.

$$x^* = a + b - x \quad (2.19)$$

$$x_j^* = a_j + b - x_j \quad (2.20)$$

Benzer şekilde, tanım aşağıdaki gibi daha yüksek boyutlar için genelleştirilebilir. $x=(x_1, x_2, \dots, x_D)$, D boyutlu uzayda bir nokta olsun, burada $x_1, x_2, \dots, x_D \in \mathbb{R}$ ve $x_j \in [a_j, b_j]$, $j \in 1, 2, \dots, D$. Karşıt nokta $x^* = x_1^*, x_2^*, \dots, x_D^*$ denklem 2.21.'deki gibi tanımlanır.

Karşıtlık öğrenme yapısı, mevcut çözüm aranırken arama ajanı çözüme uzak ise arama karşıt nokta çözüme daha yakın olduğu varsayılarak karşıt noktada çözüm aranmaktadır. Karşıtlık öğrenme mekanizmasını kullanıldığı bir diğer alanda karşıt temelli jenerasyon atlama ile atlama hız katsayısına göre arama temsilcisinin karşıt noktasını optimizasyon boyunca hesaplamaktadır. Bulduğu arama temsilcisi ile mevcut arama temsilcisi karşılaştırıp maliyet fonksiyonunun sonucuna göre bir sonraki adım için saklanmaktadır (Ewees ve Elaziz, 2020 : 16). Karşıt temelli jenerasyon atlama işlemi aşağıdaki formülasyon ile gerçekleştirilmektedir.

$$X_{i,g+1} = \begin{cases} x_{i,g} = a + b - \check{x}_{i,g} & \text{eğer}(r_1 < j_r) \\ x_{i,g}, & \text{diğer} \end{cases} \quad (2.21)$$

Bu denklemde, j_r atlama hızı değerini, $x_{i,g+1}$ bir sonraki iterasyondaki arama ajanını, $\check{x}_{i,g}$ karşıt arama ajanını, r_1 rastgele sayıyı $[0,1]$ göstermektedir.

OppMOHHO algoritmasının çalışma aşamaları kısaca aşağıda bahsedildiği gibidir:

Adım 1. Popülasyon boyutu (N), maksimum iterasyon (T), arşiv boyutu (S) değerleri ve atlama hızı değeri verilir. Değişken aralıkları içinde rastgele başlatılır.

Adım 2. Başlangıçta karşıt popülasyon hesaplanır.

Adım 3. Her bir birey (şahin) ve karşıt birey için $f_1(x)$ ve $f_2(x)$ maliyet fonksiyonu değerleri hesaplanır. Maliyet değerlerine göre en iyi N değer başlangıç çözümü olarak alınır.

Adım 4. Mevcut iterasyonda popülasyondaki baskın olmayan pareto optimal çözümler seçilir; Böylece $X_{rabbit}(t)$, tavşanın pozisyonu arşivin daha az nüfuslu bir alanından rulet çarkı yöntemi ile seçilir.

Adım 5. Her bir şahin pozisyonu, HHO algoritmasındaki denklemlere göre güncellenir.

Adım 6. Her şahin için yeni amaç fonksiyon değeri hesaplanır sonrasında baskın olmayan çözümler güncellenir; böylece daha iyi çözümler arşivlenir.

Adım 7. Karşıtlık tabanlı jenerasyon atlama işlemi ile mevcut aday çözümlerin karşıt noktaları hesaplanır, mevcut ve karşıt noktalar arasından maliyet değerlerine göre en iyi N aday çözüm bir sonraki iterasyon için seçilir.

Adım 8. Arşiv dolduğunda, arşivde alan oluşturmak için arşivdeki veriler rulet çarkı yöntemi ile silinir. Adım 4'e gidilerek aynı işlemler maksimum iterasyon sayısına ulaşana kadar devam edilir.

Adım 9. Maksimum iterasyon sayısına ulaşıldığında, arşivdeki en iyi çözümün çıktısı alınır.

3. DENEYSEL ÇALIŞMALAR

3.1. Başarım Metrikleri

Algoritmaların başarımlarını ölçmek amacıyla beş başarım metriği kullanılmıştır. Bunlar GD, IGD, S, Sp ve MS metrikleridir.

3.1.1. Nesil Mesafe Metriği

Nesil mesafe (GD) olarak geliştirilmiş başarım göstergesi Pareto cephelerinin sürekli ve süreksiz yaklaşımlarını karşılaştırmak için kullanılmaktadır. GD, önceden belirlenmiş bir referans noktası seti ile karşılaştırıldığında elde edilen çözümün kalitesini değerlendirmektedir. Bu gösterge çözüm ile bir referans noktası arasındaki ortalama mesafeyi değerlendirir. GD değerinin düşük olduğu algoritma daha iyi olarak nitelendirilir.

P^* : Pareto - optimal cephesi (istenen)

Q : Elde edilen Pareto – cephesi

$$GD = \frac{1}{N_Q} \left[\sum_{i=1}^{N_Q} d_i^P \right]^{1/P} \quad (3.1)$$

N_Q : Elde edilen Pareto - çözümlerinin sayısı

P : Amaç sayısı

d_i : P^* nin en yakın üyesi ve i . çözüm arasındaki mesafe

$P=2$ için d_i öklid mesafesidir ve aşağıdaki gibidir.

$$d_i = \min_{k \in P^*} \sqrt{\sum_{m=1}^M (f_m^{(i)} - f_m^{*(k)})^2} \quad (3.2)$$

$f_m^{(i)}$: i . elde edilen çözümün m . amaç fonksiyonu

$f_m^{*(k)}$: P^* nin k . üyesinin m . amaç fonksiyonu

M : Amaç fonksiyonu sayısı

Bu metrik, P^* 'dan Q ' nun çözümlerinin bir ortalama mesafesini değerlendirir. GD değerinin düşük olduğu algoritma daha iyidir.

3.1.2. Ters Nesil Mesafesi Metriği

Ters nesil mesafe (IGD) olarak geliştirilmiş başarımların göstergesi Pareto cephelerinin sürekli ve süreksiz yaklaşımlarını karşılaştırmak için kullanılmaktadır. IGD, önceden belirlenmiş bir referans noktası seti ile karşılaştırıldığında elde edilen çözümün kalitesini değerlendirir. Bu gösterge çözüm ile bir referans noktası arasındaki ortalama mesafeye değerlendirir. IGD değerinin düşük olduğu algoritma daha iyidir.

$$IGD = \frac{1}{N_P} \left[\sum_{j=1}^{N_P} \hat{d}_j^P \right]^{1/P} \quad (3.3)$$

N_P : Gerçek Pareto - çözümlerinin sayısı

P : Amaç sayısı

\hat{d}_j : j. Gerçek çözüm ve en yakın Pareto - optimal çözüm arasındaki mesafe

P=2 için

$$\hat{d}_j = \min_{k \in Q} \sqrt{\sum_{m=1}^M (f_m^{*(j)} - f_m^{(k)})^2} \quad (3.4)$$

$f_m^{*(j)}$: P* nin j. üyesinin m. Amaç fonksiyonu

$f_m^{(k)}$: k. elde edilen çözümün m. Amaç fonksiyonu

M : Amaç fonksiyonu sayısı

3.1.3. Aralık Metriği

Aralık (S) metriği, elde edilen baskın olmayan ardışık çözümler arasındaki bağıl mesafeyi değerlendirmektedir. Mesafe ölçümü, elde edilen baskın olmayan çözüm setindeki i. çözüm ile diğerleri arasındaki amaç fonksiyonu değerlerindeki mutlak farkın toplamının minimum değeridir. Aralık (S) metriği farklı d_i değerlerinin standart sapmasını ölçmektedir. Küçük aralığa sahip bir çözüm seti bulan bir algoritma iyi olarak nitelendirilir.

$$S = \sqrt{\frac{1}{N_Q} \sum_{i=1}^{N_Q} (d'_i - \bar{d})^2} \quad (3.5)$$

d'_i : Elde edilen Pareto_ çözümlerinin i. elde edilen çözüme uzaklığı

$$d_i' = \min_{k \in Q, k \neq i} \left\{ \sum_{m=1}^M (f_m^{(i)} - f_m^{(k)}) \right\} \quad (3.6)$$

$f_m^{(k)}$: Q' nun k. Üyesinin m. Amaç fonksiyonu

$$\bar{d} = \frac{1}{N_Q} \sum_{i=1}^{N_Q} d_i \quad (\text{ortalama mesafe}) \quad (3.7)$$

3.1.4. Yayılma Metriği

Yayılma (Spread (Sp)) metriği, elde edilen çözümler arasındaki yayılmanın boyutunu ölçmektedir. Sp'nin küçük olması algoritmanın daha iyi bir çözüm seti bulabildiğini gösterir.

$$\Delta = \frac{\sum_{m=1}^M d_m^e + \sum_{i=1}^{N_Q} |d_i'' - \bar{d}|}{\sum_{m=1}^M d_m^e + N_Q \bar{d}} \quad (3.8)$$

d_m^e : m. Amaç fonksiyonuna uygun P* ve Q' nun sınır çözümleri arasındaki mesafe

d_i'' : Komşu çözümler arasındaki Öklid mesafesi

3.1.5. Maksimum Yayılma Metriği

Bu metrik (MS) gerçek Pareto çözümleri ve elde edilen Pareto çözümlerinin sınır değerleri arasındaki mesafeyi ölçer. MS değerlerinin yüksek olması, algoritmanın çözümlerinin daha iyi yayılım gösterdiği anlamına gelmektedir.

$$MS = \sqrt{\frac{1}{M} \sum_{i=1}^M \left[\frac{\max(f_i^{(min)}, f_i^{*(min)}) - \min(f_i^{(max)}, f_i^{*(max)})}{f_i^{*(max)} - f_i^{*(min)}} \right]^2} \quad (3.9)$$

3.2. Ayarlar

DeneySEL çalışmalarda, algoritmalar arası karşılaştırmaları eşit şartlar altında ve standartlara uygun bir şekilde gerçekleştirmek için ZDT ve DTLZ test fonksiyon serileri kullanılmıştır. Buna göre her bir karşılaştırma problemi için birbirinden bağımsız tekrar sayısı 10'dur. Maksimum iterasyon 500, arşiv boyutunu 100, problem boyutu 5 alınarak test edilmiştir.

3.2.1. Çok Amaçlı Optimizasyon Algoritmaları

Deneylerde geliştirilen OppMOHHO algoritması, MOHHO algoritması ve meta-sezgisel arama tabanlı çok amaçlı optimizasyon algoritmaları arasından üç algoritma seçilmiştir. Bunlar; MOMVO (Çok Amaçlı Çok Hedefli Optimizasyon-Multi Objective Multi Verse Optimizer) (Mirjalili, Jangir, Mirjalili, Saremi, ve Trivedi, 2017 : 22), MOALO (Çok Amaçlı Karınca Aslanı Optimizasyonu-Multi Objective Ant Lion Optimizer) (Mirjalili, Jangir, ve Saremi, 2017 : 16) ve MODA (Çok Amaçlı Yusufçuk Optimizasyonu-Multi Objective Dragonfly Optimizer) (Mirjalili S. , 2016 : 20) algoritmalarıdır.

3.2.2. ZDT Test Fonksiyonu Serisi

ZDT test fonksiyonu serisi iki amaçlı problemler için tasarlanmıştır. Adını yazarları Zitzler, Deb ve Thiele (Zitzler, Deb, ve Thiele., 2000 : 22)'den alıyor. Yazarlar makalede, hepsi iyi düşünülmüş bir fonksiyon kombinasyonundan kaynaklanan 6 farklı ölçeklenebilir problem seti önermektedir ve yapım yoluyla, ilginç problemler yaratırken herhangi bir noktanın Pareto cephesine olan mesafesini ölçmeye izin vermektedir. Tablo 3.1.'de ZDT test fonksiyonlarının matematiksel denklemleri bulunmaktadır.

Tablo 3.1. ZDT Test Fonksiyonu Serisi için Çok Amaçlı Optimizasyon Test Problemleri

ZDT1	$f_1(x) = x_1$ $f_2(x) = g(x)(1 - \sqrt{x_1/g(x)})$ $g(x) = \left(\frac{9}{(N-1)}\right) \sum_{i=2}^N x_i$
------	--

ZDT2	$f_1(x) = x_1$ $f_2(x) = g(x)(1 - (x_1/g(x))^2)$ $g(x) = \left(\frac{9}{(N-1)}\right) \sum_{i=2}^N x_i$
------	---

ZDT3	$f_1(x) = x_1$ $f_2(x) = g(x)(1 - \sqrt{(x_1/g(x))} - (x_1/g(x)) \sin(10\pi x_1))$
------	--

$$g(x) = \left(\frac{9}{(N-1)}\right) \sum_{i=2}^N x_i$$

ZDT4

$$f_1(x) = x_1$$

$$f_2(x) = g(x) \left(1 - \sqrt{x_1/g(x)}\right)$$

$$g(x) = 1 + 10(N-1) \sum_{i=2}^N (x_i^2 - 10 \sin(4\pi x_i))$$

ZDT5

$$f_1(x) = 1 + u_1$$

$$f_2(x) = g(x) \left(1/(1 + u_i)\right)$$

$$g(x) = \sum_{i=2}^N v_i$$

$$\text{Eğer } \begin{cases} x_i = 0, & u_i = 0 \\ \text{değil,} & u_i = 0 \end{cases}$$

$$\text{Eğer } \begin{cases} u_i < 5, & v_i = 2 + u_i \\ u_i = 5, & v_i = 1 \end{cases}$$

ZDT6

$$f_1(x) = 1 - \exp(-4x_1) \sin(6\pi x_1)^6$$

$$f_2(x) = g(x) \left(1 - (x_1/g(x))^2\right)$$

$$g(x) = 1 + 9 \left(\frac{(\sum_{i=2}^N x_i)}{(N+1)^{0.25}} \right)$$

3.2.3. DTLZ Test Fonksiyonu Serisi

DTLZ test fonksiyonu serisi, ölçeklenebilir uygunluk boyutlarına sahip çok amaçlı problemler için tasarlanmıştır. Fonksiyon seti adını yazarları Deb, Thiele, Laumanns ve Zitzler (Deb, Thiele, Laumanns, ve Zitzler, 2005 : 40)'den alıyor. Bu test takımındaki tüm problemler, uygunluk boyutunda ölçeklenebilir, kutu kısıtlamalı sürekli n-boyutlu çok amaçlı problemlerdir. Karar uzayının boyutu, $k + \text{fdim} - 1$, $k + \text{fdim} - 1$, fdim ise hedef sayısı ve k parametresidir. Tablo 3.2.'de DTLZ test fonksiyonlarının matematiksel denklemleri bulunmaktadır.

Tablo 3.1. DTLZ Test Fonksiyonu Serisi için Çok Amaçlı Optimizasyon Test Problemleri

DTLZ1	$f_1(x) = \frac{1}{2} x_1(1 + g(x))$ $f_2(x) = \frac{1}{2} (1 - x_1)(1 + g(x))$ $g(x) = 100[x + \sum_{x_i \in x} (x_i - 0.5)^2 - \cos(20 \cdot \pi(x_i - 0.5))]$
DTLZ2	$f_1(x) = (1 + g(x)) \cos\left(x_1 \frac{\pi}{2}\right)$ $f_2(x) = (1 + g(x)) \sin\left(x_1 \frac{\pi}{2}\right)$ $g(x) = \sum_{x_i \in x} (x_i - 0.5)^2$
DTLZ3	$f_1(x) = (1 + g(x)) \cos\left(x_1 \frac{\pi}{2}\right)$ $f_2(x) = (1 + g(x)) \sin\left(x_1 \frac{\pi}{2}\right)$ $g(x) = 100 \cdot \left[x + \sum_{x_i \in x} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)) \right]$
DTLZ4	$f_1(x) = (1 + g(x)) \cos\left(x_1^\alpha \frac{\pi}{2}\right)$ $f_2(x) = (1 + g(x)) \sin\left(x_1^\alpha \frac{\pi}{2}\right)$ $g(x) = \sum_{x_i \in x} (x_i - 0.5)^2, \alpha = 100$
DTLZ5	$f_1(x) = (1 + g(x)) \cos\left(x_1 \frac{\pi}{2}\right)$ $f_2(x) = (1 + g(x)) \sin\left(x_1 \frac{\pi}{2}\right)$ $g(x) = \sum_{x_i \in x} (x_i - 0.5)^2,$ $x_i = \frac{\pi}{4(1 + g(x))} (1 + 2g(x)x_i)$
DTLZ6	$f_1(x) = (1 + g(x)) \cos\left(x_1 \frac{\pi}{2}\right)$ $f_2(x) = (1 + g(x)) \sin\left(x_1 \frac{\pi}{2}\right)$ $g(x) = \sum_{x_i \in x} x_i^{0.1}, x_i = \frac{\pi}{4(1 + g(x))} (1 + 2g(x)x_i)$

$$f_1(x) = x_1$$

$$f_2(x) = (1 + g(x)) \left(M - \frac{x_1}{1 + g(x)} (1 + \sin(3\pi x_1)) \right), M = 2$$

$$g(x) = 1 + \frac{9}{|x|} \sum_{x_i \in x} x_i$$

ZDT ve DTLZ test fonksiyon serileri için çok amaçlı problem setindeki her bir problem için tüm algoritmalar 10 kez çalıştırılmış ve her bir çalıştırma sonucunda hesaplanan GD, IGD, S, Sp ve MS metrikleri hesaplanmıştır. Metrik değerleri aşağıda verilmiştir.

3.3. ZDT Test Fonksiyonu Serisi İçin Karşılaştırmalı Sonuçları

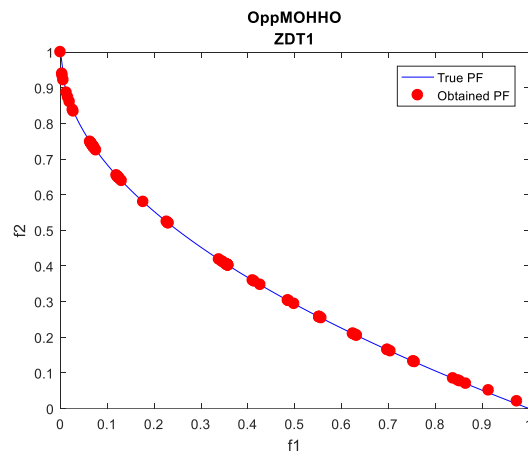
GD, IGD, S, Sp ve MS metrikleri için tüm algoritmaların istatistiksel sonuçları (Minimum, Maksimum, Ortalama ve Standart Sapma) bulunmaktadır. Veriler GD, IGD, S, Sp ve MS metriklerinin istatistiksel ortalama değerine göre sıralanmıştır. Sonuçlar Tablo 3.3. – Tablo 3.7. arasında verilmiştir.

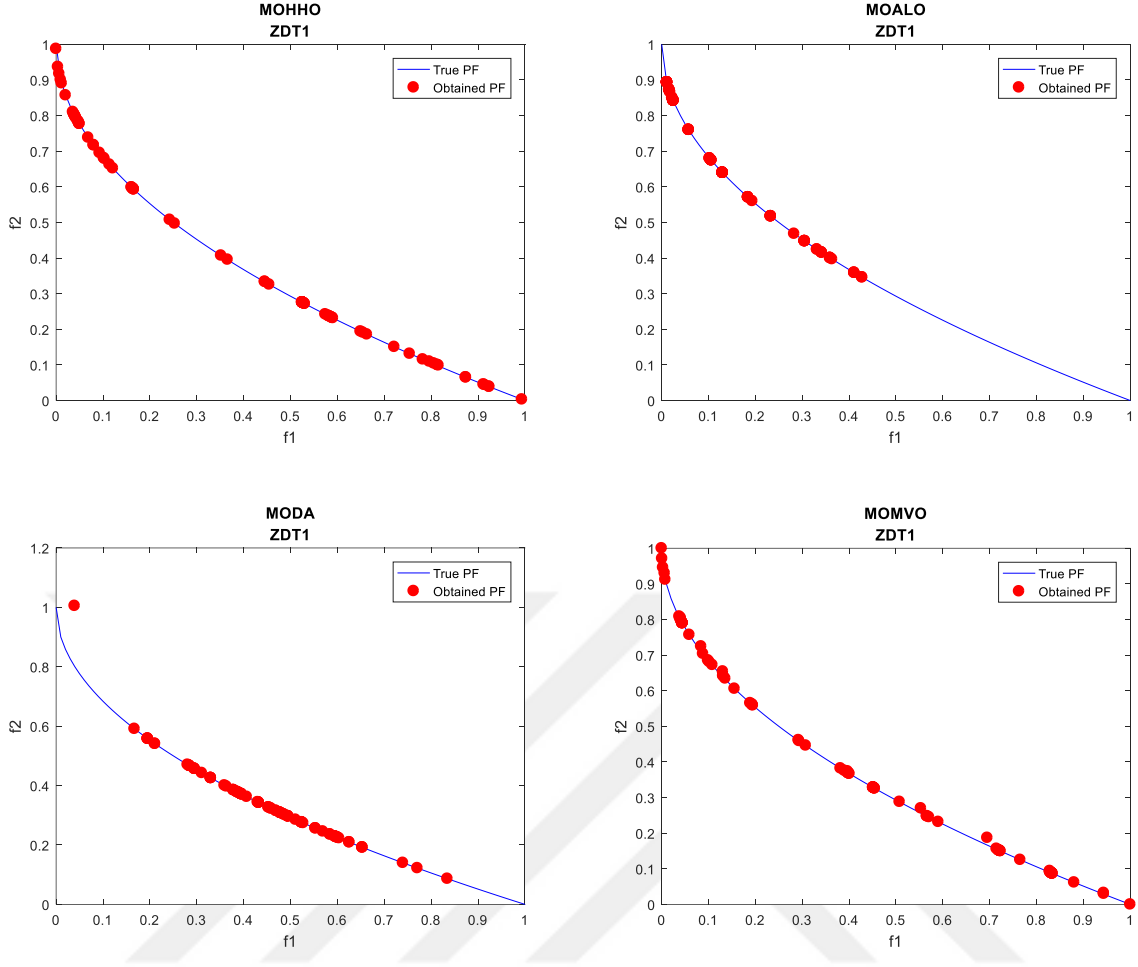
Tablo 3.3. OppMOHHO ve diğer algoritmaların ZDT fonksiyonu serisi için IGD metrik sonuçları

IGD	OppMOHHO	MOHHO	MOALO	MODA	MOMVO
Min	2.0968e-03	2.1558e-03	7.5053e-03	2.6238e-03	2.2693e-03
Maks	3.1284e-03	3.1261e-03	2.8622e-02	3.5280e-02	3.4617e-03
ZDT1 Ort.	2.6128e-03	2.6590e-03	1.6085e-02	1.0384e-02	2.9402e-03
SD	3.5205e-04	3.1904e-04	6.5678e-03	9.2860e-03	4.1275e-04
Sıra	1	2	5	4	3
Min	1.9331e-03	2.0533e-03	5.8187e-03	3.8275e-03	2.0374e-03
Maks	3.3973e-03	4.8615e-03	3.7475e-02	1.7363e-02	3.4647e-03
ZDT2 Ort.	2.3588e-03	2.7954e-03	2.0062e-02	8.1583e-03	2.6620e-03
SD	4.9234e-04	7.8483e-04	1.0799e-02	3.9199e-03	4.5609e-04
Sıra	1	3	5	4	2
Min	2.3860e-02	2.3504e-02	2.4158e-02	2.5003e-02	2.3822e-02
Maks	2.4885e-02	2.5167e-02	3.2551e-02	3.2223e-02	2.5356e-02
ZDT3 Ort.	2.4463e-02	2.4399e-02	2.7230e-02	2.6289e-02	2.4433e-02
SD	3.0698e-04	4.2077e-04	2.8361e-03	2.1390e-03	5.3375e-04

Sıra	3	1	5	4	2
Min	2.4242e-03	2.5782e-03	9.2187e-03	2.8312e-03	2.3507e-03
Maks	5.2750e-03	8.8654e-02	1.0964e-01	1.4854e-02	1.1429e-01
ZDT4 Ort.	3.2928e-03	2.0324e-02	3.2029e-02	7.0687e-03	3.0984e-02
SD	8.4513e-04	3.5679e-02	2.9537e-02	4.2805e-03	3.6721e-02
Sıra	1	3	5	2	4
Min	1.5698e-03	1.4355e-03	5.8342e-02	3.8521e-02	1.6020e-03
Maks	5.5539e-02	5.5223e-02	6.6722e-02	5.9623e-02	5.7090e-02
ZDT5 Ort.	3.1860e-02	3.9140e-02	6.1894e-02	5.4596e-02	3.5679e-02
SD	2.3616e-02	2.2496e-02	2.7238e-03	5.8588e-03	2.5463e-02
Sıra	1	3	5	4	2
Min	1.2727e-03	1.6015e-03	3.9523e-03	1.5319e-03	8.5273e-04
Maks	6.5324e-03	3.6562e-03	3.8021e-02	3.3530e-03	1.9307e-03
ZDT6 Ort.	2.4627e-03	2.3745e-03	1.2621e-02	2.4403e-03	1.2444e-03
SD	1.4904e-03	6.5828e-04	1.0242e-02	6.4006e-04	3.5523e-04
Sıra	4	2	5	3	1
Ort. Sıra	1.8334	2.3334	5.0000	3.5000	2.3334
Genel Sıra	1	2	5	3	2

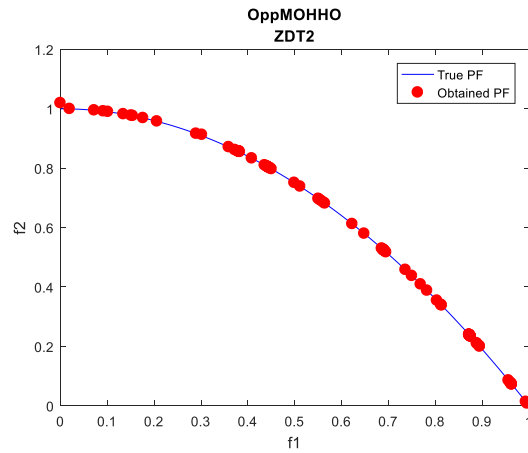
IGD metriği karşılaştırması sonrasında OppMOHHO algoritması MOHHO ve diğer algoritmalara üstün gelmiştir. IGD metriği için ZDT1-6 test fonksiyonları için grafikler aşağıda verilmiştir.

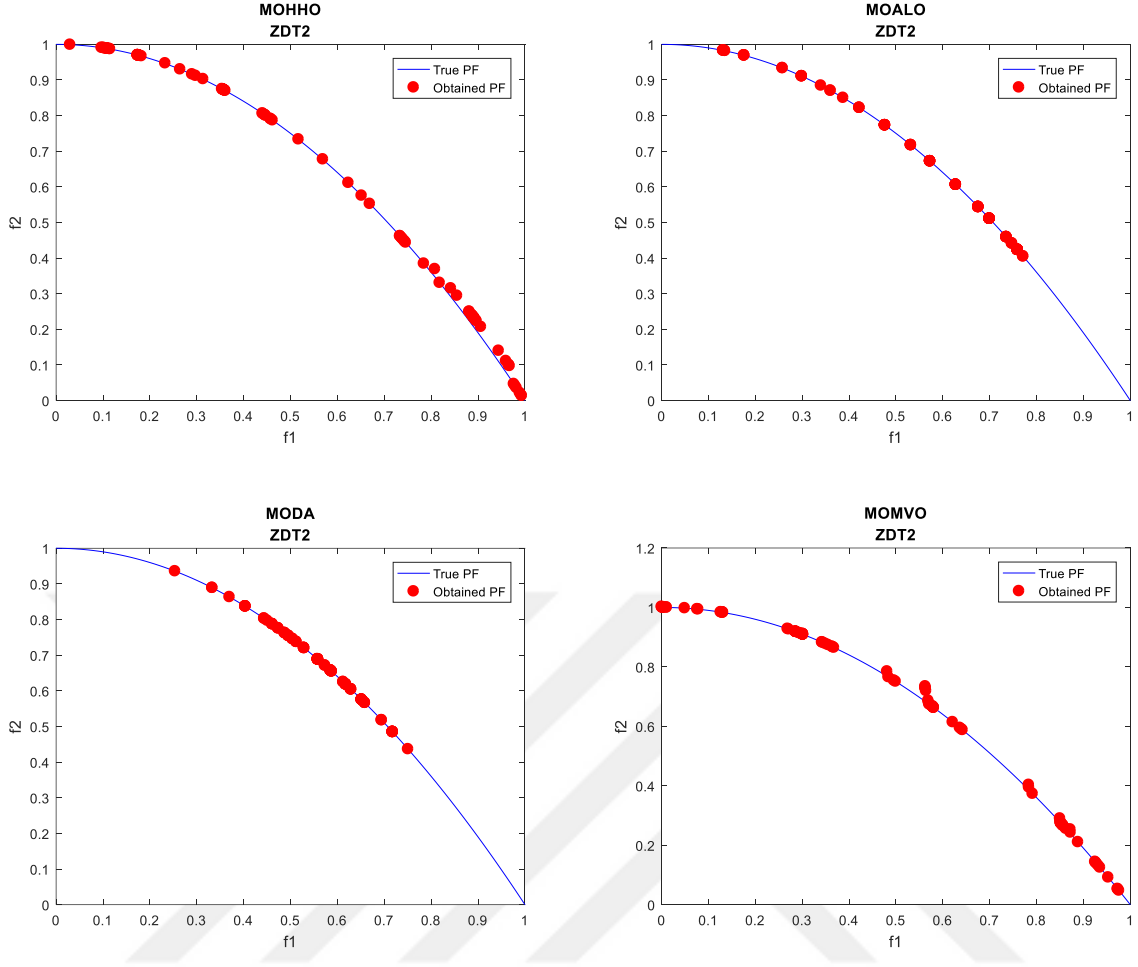




Şekil 3.1. ZDT1 fonksiyonu için pareto optimal grafikleri

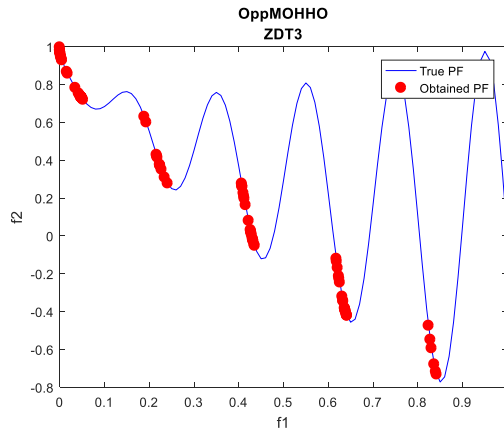
Şekil 3.1.' de görüldüğü üzere ZDT1 test fonksiyonunun grafiğinde en iyi dağılımı OppMOHHO algoritması vermiştir. MODA ve MOMVO algoritmalarında sapma olurken MOALO algoritması çözümlerde yetersiz kalmıştır.

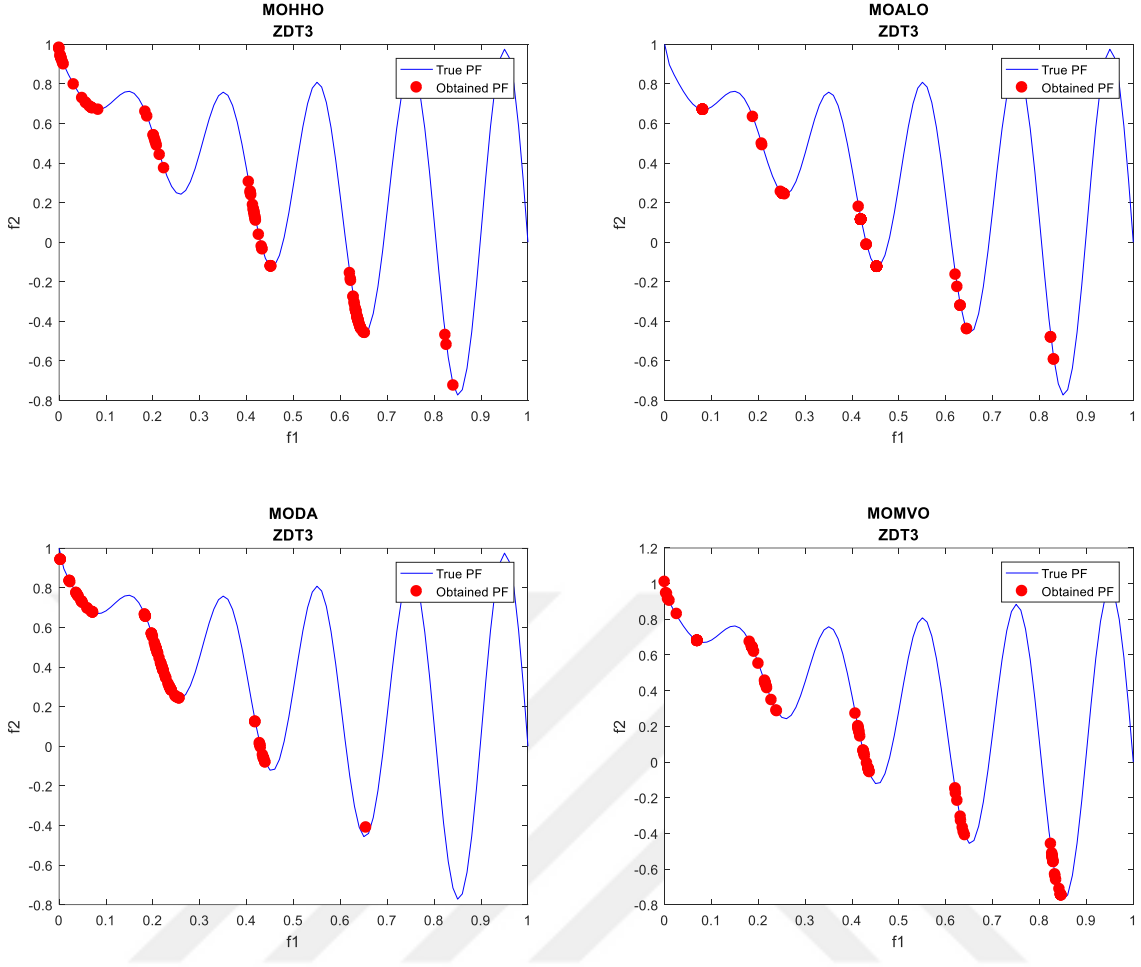




Şekil 3.2. ZDT2 fonksiyonu için pareto optimal grafikleri

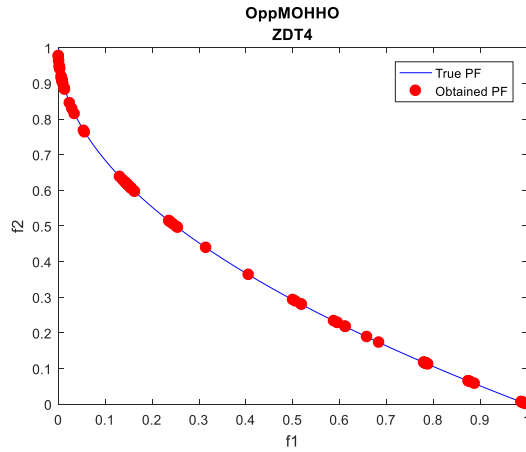
ZDT2 test fonksiyonunun sonuçları Şekil 3.2.'de görülmektedir. Grafikler incelendiğinde OppMOHHO ve MOHHO algoritmaların test fonksiyonunun grafiğini takip konusunda daha iyi olduğu gözlemlenmektedir. Fakat MOHHO algoritması grafiğinin devamında küçük sapmalar bulunmakta ve bu durumda OppMOHHO algoritmasının daha başarılı olduğunu göstermektedir.

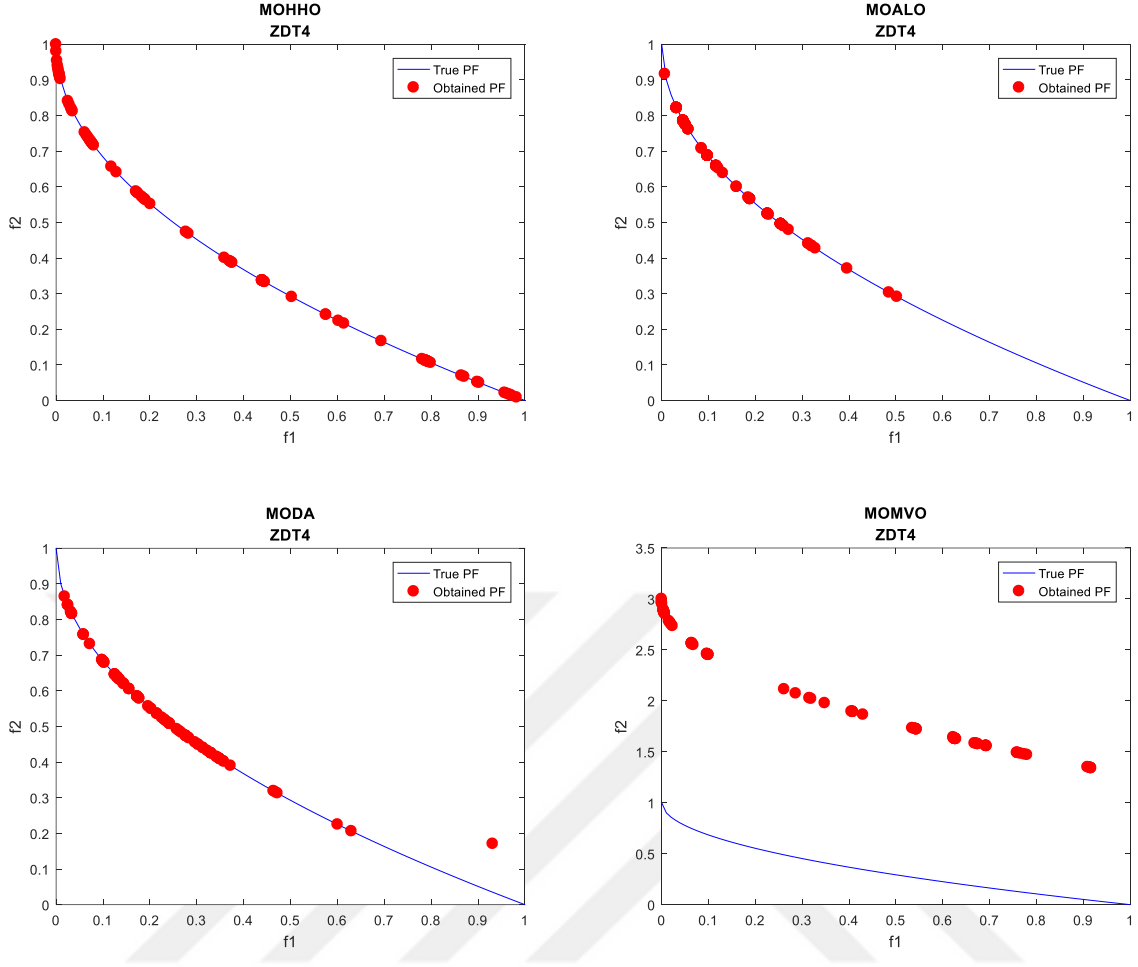




Şekil 3.3. ZDT3 fonksiyonu için pareto optimal grafikleri

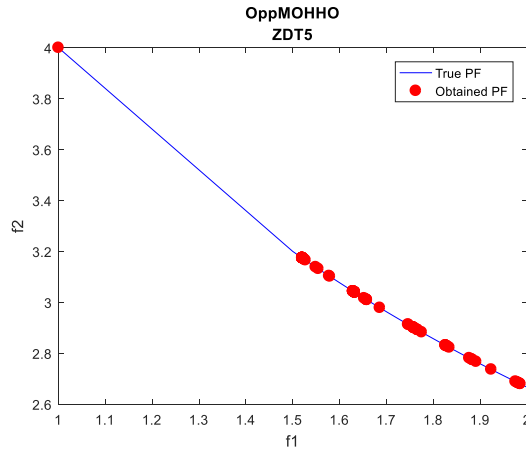
IGD metriği için Şekil 3.3.'de ZDT3 test fonksiyonunun grafikleri verilmiştir. Grafikler incelendiğinde test fonksiyonunun grafiği dalgalı görünmektedir. MOALO, MODA ve MOMVO algoritmalarının sonuçlarının kötü olduğu gözlemlenmektedir. Bu test fonksiyonu için en iyi sonucu MOHHO algoritması vermektedir.

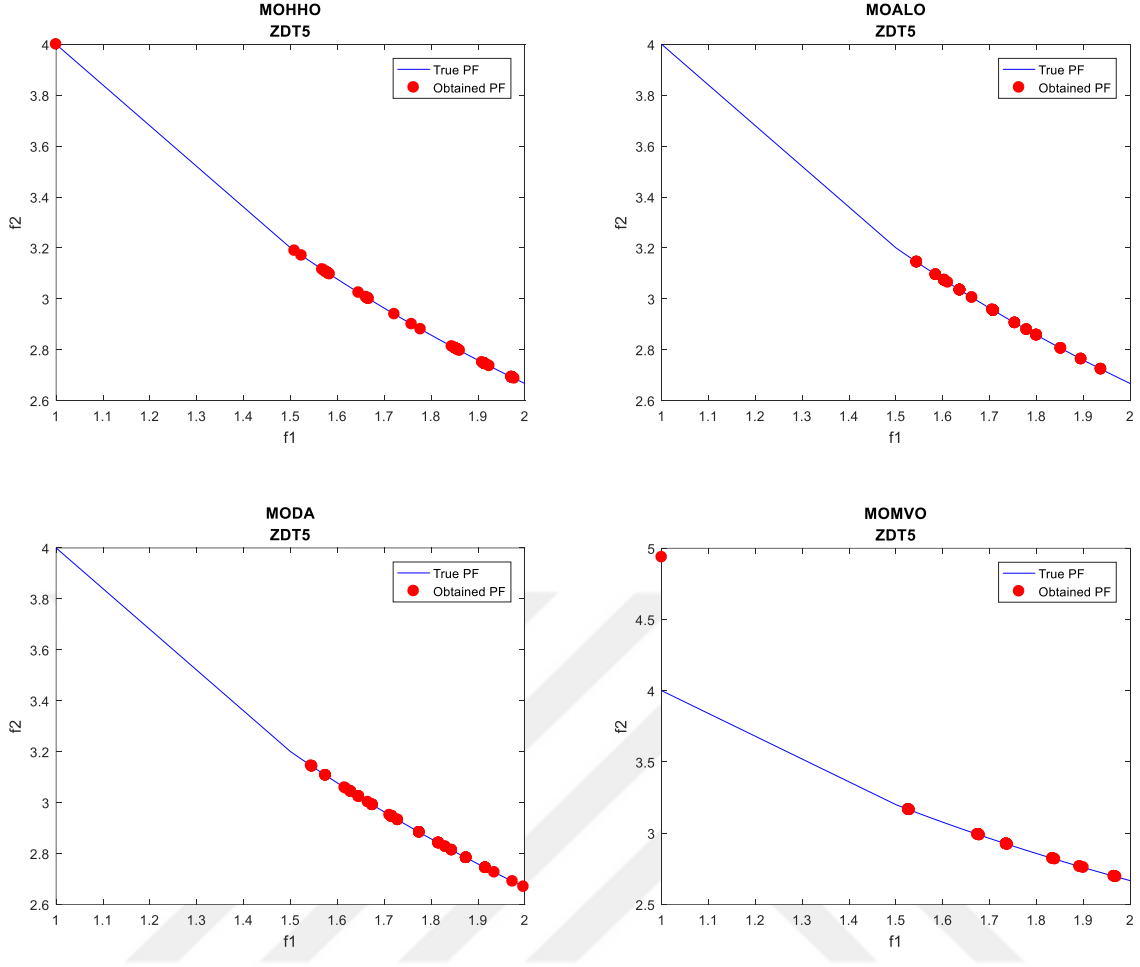




Şekil 3.4. ZDT4 fonksiyonu için pareto optimal grafikleri

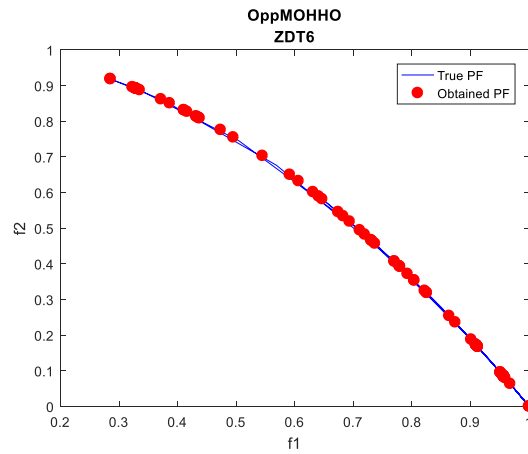
Şekil 3.4. 'de ZDT4 test fonksiyonunun sonuçları verilmiştir. MODA ve MOMVO algoritmalarının sonuçtan saptığı ve MOALO algoritmasının yetersiz kaldığı görülmektedir. OppMOHHO algoritması ve MOHHO algoritmasına bakıldığında OppMOHHO algoritmasının daha başarılı olduğu gözlemlenmektedir.

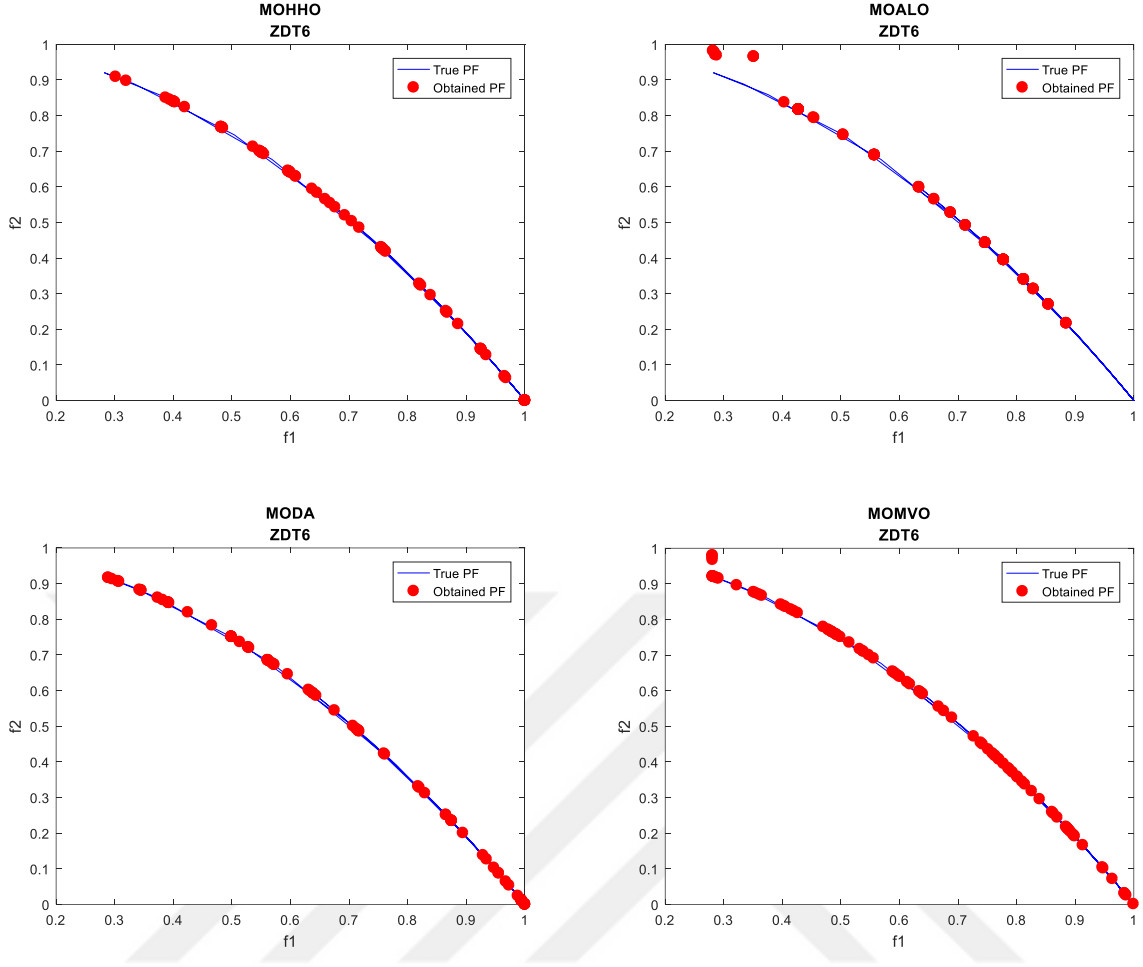




Şekil 3.5. ZDT5 fonksiyonu için pareto optimal grafikleri

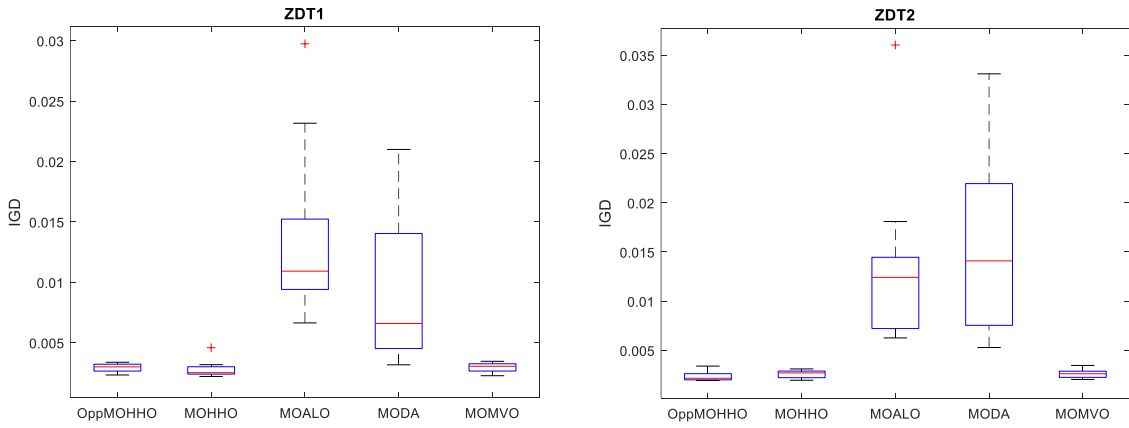
Şekil 3.5.'te ZDT5 için sonuçlar bulunmaktadır. Sonuçlara bakıldığında grafiği en iyi ve sık notalar ile takip edenin OppMOHHO algoritması olduğu görülmektedir.

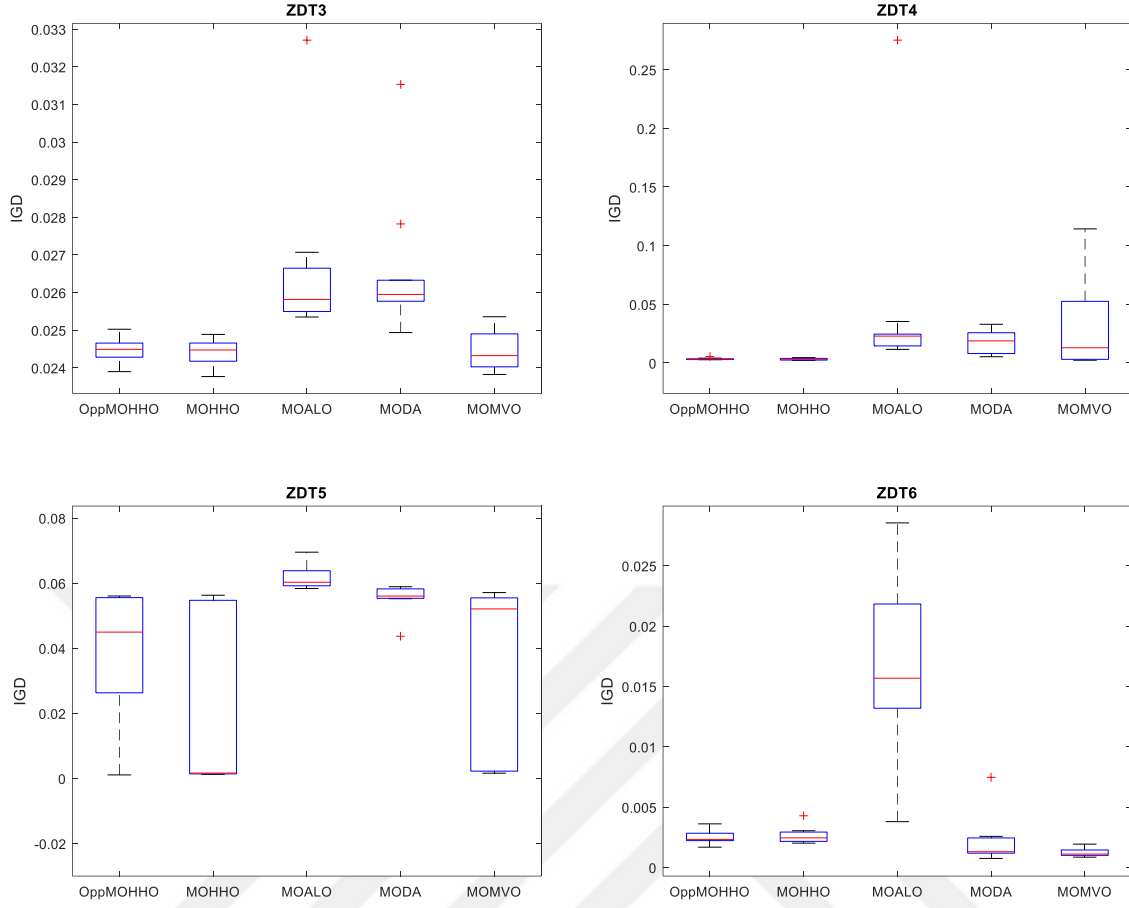




Şekil 3.6 ZDT6 fonksiyonu için pareto optimal grafikleri

ZDT6 test fonksiyonu için Şekil 3.6.'ya bakıldığında OppMOHHO algoritmasının başarılı olduğu görülmektedir. Diğer algoritmalarda sapma ya da seyreklikler görülmektedir.





Şekil 3.7. ZDT test fonksiyonu serisi için IGD metriği istatistiksel sonuçları

Şekil 3.7.' de ZDT test fonksiyon serisi için IGD metriğinin istatistiksel sonuçlarının kutu grafiği verilmiştir. ZDT1'in kutu grafiği incelendiğinde en dar alana sahip algoritmanın OppMOHHO algoritması olduğu görülmektedir. ZDT2 için bakıldığında en dar alana sahip OppMOHHO algoritması iken en geniş alana sahip algoritma ise MODA'dır. ZDT3 test fonksiyonunda ise MOHHO algoritmasının daha iyi olduğu görülmektedir. ZDT4 için bakıldığında OppMOHHO algoritması alanının diğer algoritmalara bakıldığında en dar alana sahip olduğu dikkat çekmektedir. ZDT5 ve ZDT6 test fonksiyonlarında yine OppMOHHO algoritmasının daha başarılı olduğu görülmektedir.

Tablo 3.4. OppMOHHO ve diğer algoritmaların ZDT fonksiyonu serisi için GD metrik sonuçları

GD	OppMOHHO	MOHHO	MOMVO	MOALO	MODA
Min	5.6775e-04	6.2339e-04	5.8544e-04	3.7880e-04	3.7048e-04
ZDT1 Maks	1.5398e-03	2.2644e-03	1.2063e-03	1.0059e-03	7.8808e-04
Ort.	9.6227e-04	1.1282e-03	8.7533e-04	5.8532e-04	5.1558e-04

	SD	3.0100e-04	4.5534e-04	1.8989e-04	1.8812e-04	1.3102e-04
	Sıra	4	5	3	2	1
ZDT2	Min	4.2402e-04	4.5304e-04	4.4909e-04	2.5972e-04	3.6850e-04
	Maks	6.4145e-04	4.7024e-03	5.0408e-02	4.7458e-04	6.8909e-04
	Ort.	5.1109e-04	1.0795e-03	9.9006e-03	3.7285e-04	5.0448e-04
	SD	6.3377e-05	1.3244e-03	1.9736e-02	7.3826e-05	8.5651e-05
	Sıra	3	4	5	1	2
ZDT3	Min	1.9213e-03	2.2510e-03	2.2872e-03	2.1665e-03	1.8340e-03
	Maks	3.6720e-03	3.5679e-03	2.3741e-01	3.1762e-03	3.6256e-03
	Ort.	2.6378e-03	2.7162e-03	2.8622e-02	2.6156e-03	2.6002e-03
	SD	5.9998e-04	4.3506e-04	7.3752e-02	3.4186e-04	5.9012e-04
	Sıra	3	4	5	2	1
ZDT4	Min	5.4722e-04	4.9490e-04	1.1610e-03	3.1636e-04	3.4461e-04
	Maks	1.3356e-03	1.4674e-03	2.9887e-01	6.3570e-04	9.4523e-04
	Ort.	8.8999e-04	9.6306e-04	7.6138e-02	4.4328e-04	5.4208e-04
	SD	2.3734e-04	3.1108e-04	9.3630e-02	9.8370e-05	1.7014e-04
	Sıra	3	4	5	1	2
ZDT5	Min	3.4459e-04	3.4249e-04	3.1480e-04	2.9954e-04	3.0406e-04
	Maks	7.0919e-03	5.0170e-03	1.0338e-02	4.5870e-04	9.2812e-03
	Ort.	2.4673e-03	8.8375e-04	2.5613e-03	4.1009e-04	1.6728e-03
	SD	2.7157e-03	1.4535e-03	3.3318e-03	5.4040e-05	2.9018e-03
	Sıra	4	2	5	1	3
ZDT6	Min	1.2799e-03	1.3563e-03	2.0381e-03	2.7190e-03	1.2574e-03
	Maks	2.7576e-03	4.1320e-03	3.4881e-03	5.6617e-03	2.7831e-03
	Ort.	2.1119e-03	2.5497e-03	2.5045e-03	3.6716e-03	2.1546e-03
	SD	4.5092e-04	8.1141e-04	4.9712e-04	8.9233e-04	4.7764e-04
	Sıra	1	4	3	5	2
Ort. Sıra	3.0000	3.8334	4.3334	2.0000	1.8334	
Genel Sıra	3	4	5	2	1	

GD metriği kaşısm da OppMOHHO algoritması MOHHO algoritmasından daha iyi sonuç vermiştir fakat en iyi sonucu MODA algoritması vermiştir.

Tablo 3.5. OppMOHHO ve diğer algoritmaların ZDT fonksiyonu serisi için S metrik sonuçları

S	OppMOHHO	MOHHO	MOMVO	MOALO	MODA	
ZDT1	Min	1.0636e-02	1.4430e-02	6.8648e-03	1.9292e-03	1.3061e-02
	Maks	1.6645e-02	2.8150e-02	1.2991e-02	2.2668e-02	4.8674e-02
	Ort.	1.3328e-02	1.9484e-02	9.3604e-03	1.1716e-02	2.5621e-02
	SD	2.2846e-03	4.8434e-03	2.2797e-03	6.6144e-03	1.1196e-02
	Sıra	3	4	1	2	5
ZDT2	Min	1.1778e-02	8.2053e-03	6.2191e-03	4.8304e-03	8.1943e-03
	Maks	1.8556e-02	5.0061e-02	5.0654e-01	2.9846e-02	5.7376e-02
	Ort.	1.5283e-02	2.1415e-02	1.0356e-01	1.2700e-02	2.7693e-02
	SD	2.3985e-03	1.1481e-02	1.9559e-01	7.7356e-03	1.4149e-02
	Sıra	2	3	5	1	4

ZDT3	Min	1.2782e-02	1.4881e-02	1.1936e-02	5.4279e-03	1.1329e-02
	Maks	2.1907e-02	2.3796e-02	2.6962e-01	3.7470e-02	4.5788e-02
	Ort.	1.7280e-02	1.9904e-02	4.3704e-02	1.5447e-02	2.7271e-02
	SD	2.9937e-03	2.8291e-03	7.9717e-02	9.4872e-03	1.1625e-02
	Sıra	2	3	5	1	4
ZDT4	Min	9.9790e-03	1.1160e-02	1.0475e-02	3.2929e-03	1.2986e-02
	Maks	2.0822e-02	2.4099e-02	2.9608e+00	1.5647e-02	6.3151e-02
	Ort.	1.6399e-02	1.6147e-02	3.1102e-01	1.0429e-02	3.2890e-02
	SD	3.5290e-03	4.1887e-03	9.3105e-01	4.8017e-03	1.3975e-02
	Sıra	3	2	5	1	4
ZDT5	Min	2.4443e-03	3.3265e-03	8.2531e-04	1.5260e-03	2.1622e-03
	Maks	1.8051e-01	1.8322e-01	2.5178e-01	1.2712e-02	4.9681e-02
	Ort.	6.1101e-02	7.3653e-02	1.1509e-01	6.2454e-03	1.3935e-02
	SD	7.6043e-02	7.2744e-02	8.4892e-02	3.3803e-03	1.3363e-02
	Sıra	3	4	5	1	2
ZDT6	Min	8.0905e-03	8.6873e-03	8.3628e-03	3.2758e-03	9.6007e-03
	Maks	1.8433e-02	1.5153e-02	1.2146e-02	3.0896e-02	2.0584e-02
	Ort.	1.1829e-02	1.2854e-02	9.8041e-03	1.3083e-02	1.3506e-02
	SD	2.9042e-03	1.9837e-03	1.2014e-03	7.7259e-03	3.5918e-03
	Sıra	2	3	1	4	5
Ort. Sıra	2.5000	3.1667	3.6667	1.6667	4.0000	
Genel Sıra	2	3	4	1	5	

S metriği sonucun da OppMOHHO algoritması MOHHO karısında başarılı olmuştur fakat MOALO optimizasyonu daha başarılı olmuştur.

Tablo 3.6. OppMOHHO ve diğ er algoritmaların ZDT fonksiyonu serisi için Sp metrik sonuçları

Sp	OppMOHHO	MOHHO	MOMVO	MOALO	MODA	
ZDT1	Min	1.2921e+00	1.2472e+00	1.2375e+00	1.2566e+00	1.3474e+00
	Maks	1.4408e+00	1.4084e+00	1.3899e+00	1.5088e+00	1.7176e+00
	Ort.	1.3467e+00	1.3331e+00	1.3200e+00	1.3760e+00	1.4935e+00
	SD	4.3439e-02	6.2206e-02	5.5438e-02	9.0834e-02	1.1328e-01
	Sıra	2	3	1	4	5
ZDT2	Min	1.1912e+00	1.2511e+00	1.2214e+00	1.1669e+00	1.0811e+00
	Maks	1.4235e+00	1.4471e+00	1.3843e+00	1.5651e+00	1.6822e+00
	Ort.	1.2979e+00	1.3411e+00	1.2947e+00	1.3399e+00	1.4100e+00
	SD	6.4076e-02	5.8950e-02	4.7585e-02	1.3084e-01	1.7469e-01
	Sıra	2	4	1	3	5
ZDT3	Min	1.1153e+00	1.1165e+00	1.2450e+00	1.3714e+00	1.2970e+00
	Maks	1.2799e+00	1.2358e+00	1.3873e+00	1.4886e+00	1.5210e+00
	Ort.	1.1984e+00	1.1719e+00	1.2895e+00	1.4163e+00	1.4527e+00
	SD	6.2262e-02	4.2015e-02	4.7650e-02	3.7367e-02	6.3678e-02
	Sıra	2	1	3	4	5
ZDT4	Min	1.3357e+00	1.3488e+00	1.1135e+00	1.3084e+00	1.3703e+00
	Maks	1.4989e+00	1.4242e+00	1.4582e+00	1.5276e+00	1.6686e+00

Ort.	1.4087e+00	1.3803e+00	1.2367e+00	1.3779e+00	1.4664e+00
SD	5.5870e-02	2.2395e-02	1.2381e-01	7.1602e-02	8.9887e-02
Sıra	4	3	1	2	5
Min	1.2054e+00	1.1984e+00	1.2819e+00	1.1343e+00	1.1378e+00
Maks	1.6135e+00	1.7116e+00	1.7994e+00	1.2688e+00	1.4679e+00
ZDT5 Ort.	1.3671e+00	1.4410e+00	1.5854e+00	1.2089e+00	1.2672e+00
SD	1.6582e-01	2.3764e-01	2.0891e-01	3.7023e-02	9.1325e-02
Sıra	3	4	5	1	2
Min	1.2447e+00	1.2229e+00	7.5069e-01	1.4171e+00	8.7126e-01
Maks	1.4896e+00	1.4903e+00	9.7161e-01	1.5730e+00	1.6731e+00
ZDT6 Ort.	1.3834e+00	1.3917e+00	8.7577e-01	1.5056e+00	1.4329e+00
SD	8.1676e-02	8.3653e-02	6.9436e-02	5.0041e-02	2.7106e-01
Sıra	2	3	1	5	4
Ort. Sıra	2.5000	3.0000	2.0000	3.1667	4.3334
Genel Sıra	2	3	1	4	5

Sp metriği için OppMOHHO algoritması MOHHO algoritması karşında iyi sonuç vermiş vermiştir. En iyi sonucu MOMVO algoritması vermiştir.

Tablo 3.7. OppMOHHO ve diğer algoritmaların ZDT fonksiyonu serisi için MS metrik sonuçları

MS	OppMOHHO	MOHHO	MOMVO	MOALO	MODA
Min	9.2431e-01	9.7340e-01	9.4850e-01	4.3140e-01	6.0733e-01
Maks	1.0000e+00	1.0720e+00	1.0013e+00	8.2018e-01	9.7576e-01
ZDT1 Ort.	9.7737e-01	9.9892e-01	9.8382e-01	6.6041e-01	8.0023e-01
SD	2.0572e-02	2.8935e-02	1.6582e-02	1.3850e-01	1.3903e-01
Sıra	2	1	3	5	4
Min	9.6299e-01	9.1500e-01	9.7957e-01	3.4088e-01	5.6940e-01
Maks	1.0038e+00	1.2478e+00	4.3172e+00	8.1576e-01	9.9465e-01
ZDT2 Ort.	9.8841e-01	1.0165e+00	1.6113e+00	5.6161e-01	8.4295e-01
SD	1.1510e-02	9.3338e-02	1.3079e+00	1.7401e-01	1.3788e-01
Sıra	5	2	1	4	3
Min	8.8232e-01	8.5446e-01	8.9976e-01	6.0701e-01	8.5755e-01
Maks	9.2935e-01	9.3716e-01	2.4132e+00	8.8012e-01	9.2618e-01
ZDT3 Ort.	9.1459e-01	9.0890e-01	1.1601e+00	7.4870e-01	8.9171e-01
SD	1.4455e-02	2.3588e-02	5.3127e-01	1.1055e-01	2.7095e-02
Sıra	2	3	1	5	4
Min	9.6200e-01	9.2302e-01	9.8454e-01	4.8955e-01	6.0041e-01
Maks	1.0000e+00	9.9181e-01	2.1724e+01	8.4830e-01	9.3356e-01
ZDT4 Ort.	9.8543e-01	9.6203e-01	3.2212e+00	6.1648e-01	7.8847e-01
SD	1.0560e-02	2.4762e-02	6.5031e+00	1.2793e-01	1.0045e-01
Sıra	2	3	1	5	4
Min	4.2634e-01	4.0872e-01	3.8480e-01	1.9209e-01	3.2679e-01
Maks	1.2022e+00	1.1775e+00	1.4079e+00	3.8745e-01	6.0694e-01
ZDT5 Ort.	7.1374e-01	7.2932e-01	8.9188e-01	3.0120e-01	4.2547e-01

	SD	3.2316e-01	3.1314e-01	3.5528e-01	5.6992e-02	8.5015e-02
	Sıra	3	2	1	5	4
ZDT6	Min	9.2493e-01	9.5144e-01	9.9683e-01	7.9482e-01	9.5871e-01
	Maks	1.0389e+00	1.0339e+00	1.0375e+00	9.8042e-01	1.0447e+00
	Ort.	9.8742e-01	9.9029e-01	1.0161e+00	9.1248e-01	9.9750e-01
	SD	3.0998e-02	2.9278e-02	1.6916e-02	6.5535e-02	2.0745e-02
	Sıra	4	3	1	5	2
	Ort. Sıra	3.0000	2.8334	1.3334	4.8334	3.5000
	Genel Sıra	3	2	1	5	4

MS metriği karşısında OppMOHHO, MOHHO algoritmasından geride kalmış ve en iyi sonucu MOMVO algoritması vermiştir.

3.4. DTLZ Test Fonksiyonu Serisi İçin Karşılaştırmalı Sonuçları

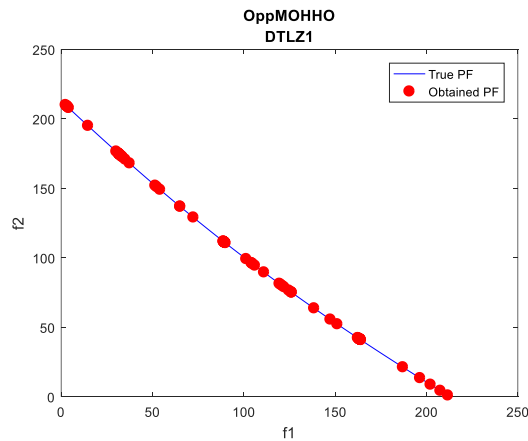
GD, IGD, S, Sp ve MS metrikleri için tüm algoritmaların istatistiksel sonuçları (Minimum, Maksimum, Ortalama ve Standart Sapma) hesaplanmaktadır. Veriler GD, IGD, S, Sp ve MS metriklerinin istatistiksel ortalama değerine göre sıralanmıştır. Sonuçlar Tablo 3.8. – Tablo 3.12. arasında verilmiştir.

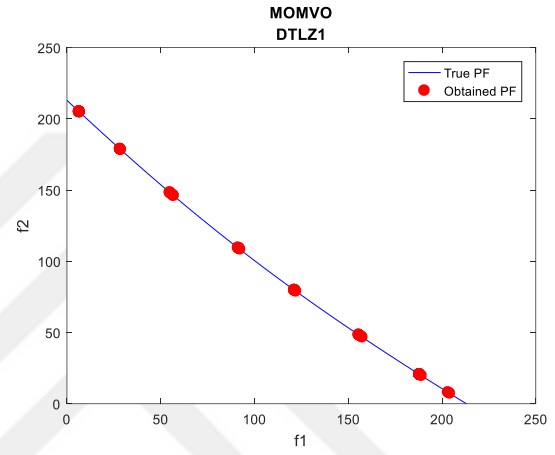
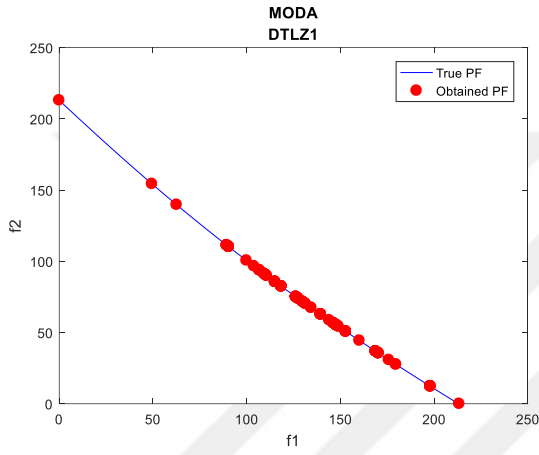
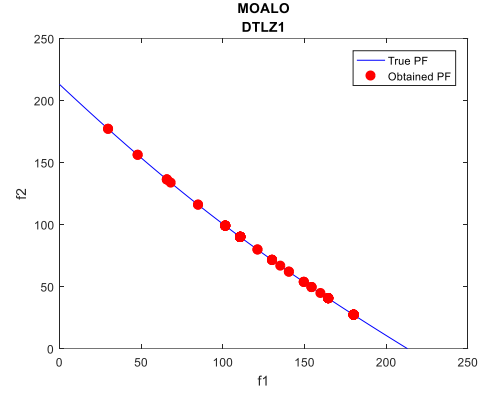
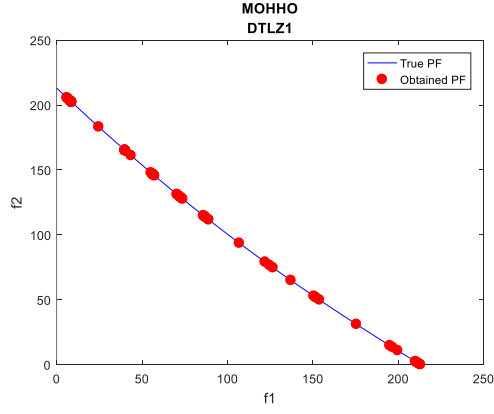
Tablo 3.8. OppMOHHO ve diğer algoritmaların DTLZ fonksiyonu serisi için IGD metrik sonuçları

	IGD	OppMOHHO	MOHHO	MOALO	MODA	MOMVO
DTLZ1	Min	2.3713e-03	2.7842e-03	5.0948e-03	4.0570e-03	4.0955e-03
	Maks	4.0476e-03	4.3296e-03	2.0949e-02	8.9709e-03	6.1863e-03
	Ort.	3.1928e-03	3.3359e-03	1.1790e-02	1.5654e-02	5.2697e-03
	SD	5.7775e-04	5.1613e-04	5.3913e-03	4.0490e-03	6.3652e-04
	Sıra	1	2	4	5	3
DTLZ2	Min	2.4414e-03	2.3326e-03	4.6225e-03	2.6737e-03	3.0198e-03
	Maks	3.6070e-03	2.8607e-03	2.2476e-02	8.0188e-03	4.8298e-03
	Ort.	3.0624e-03	4.1200e-03	1.0155e-02	1.6651e-02	4.0925e-03
	SD	3.2730e-04	5.3552e-04	5.4869e-03	4.3550e-03	6.4086e-04
	Sıra	1	3	4	5	2
DTLZ3	Min	2.1742e-03	2.3790e-03	5.9870e-03	2.0867e-03	3.5644e-03
	Maks	3.7952e-03	4.6083e-03	2.3279e-02	1.0093e-02	6.4628e-03
	Ort.	2.8705e-03	3.1324e-03	1.0955e-02	2.1359e-02	5.0731e-03
	SD	5.1723e-04	7.6740e-04	5.7693e-03	5.7098e-03	9.4826e-04
	Sıra	1	2	4	5	3

	Min	3.1531e-04	3.7675e-04	3.7206e-04	3.5190e-04	1.9782e-04
	Maks	6.0985e-03	3.4712e-03	5.0938e-02	1.7640e-03	1.0639e-02
DTLZ4	Ort.	1.7793e-03	8.6947e-04	1.1329e-02	4.4739e-03	4.4975e-03
	SD	1.8311e-03	9.3252e-04	1.6088e-02	1.1619e-03	4.5548e-03
	Sıra	2	1	5	3	4
	Min	2.2581e-03	2.3903e-03	5.2444e-03	3.7488e-03	3.4899e-03
	Maks	5.4453e-03	2.9920e-03	1.4822e-02	7.9241e-03	6.1499e-03
DTLZ5	Ort.	3.3152e-03	3.8196e-03	1.0094e-02	1.5385e-02	4.5959e-03
	SD	9.7291e-04	5.2784e-04	3.1573e-03	4.1468e-03	7.8736e-04
	Sıra	1	2	4	5	3
	Min	2.7638e-03	2.5795e-03	7.2791e-03	4.9866e-03	1.9333e-03
	Maks	5.0002e-03	3.7874e-03	3.0881e-02	1.4268e-02	3.2449e-03
DTLZ6	Ort.	3.7824e-03	4.8684e-03	2.0531e-02	3.1381e-02	2.6784e-03
	SD	7.0167e-04	7.9539e-04	7.8245e-03	8.4833e-03	4.3023e-04
	Sıra	2	3	4	5	1
	Min	1.3514e-02	1.3302e-02	1.6636e-02	1.3617e-02	1.3670e-02
	Maks	1.6369e-02	1.7713e-02	5.7761e-02	1.6248e-02	5.1461e-02
DTLZ7	Ort.	1.4184e-02	5.1861e-02	4.4513e-02	3.0006e-02	1.7899e-02
	SD	8.3673e-04	1.2011e-02	1.8450e-02	4.9181e-03	1.1801e-02
	Sıra	1	5	4	3	2
	Avg. Sıra	1.2857	2.5714	4.1428	4.4285	2.5714
	Genel Sıra	1	2	4	5	2

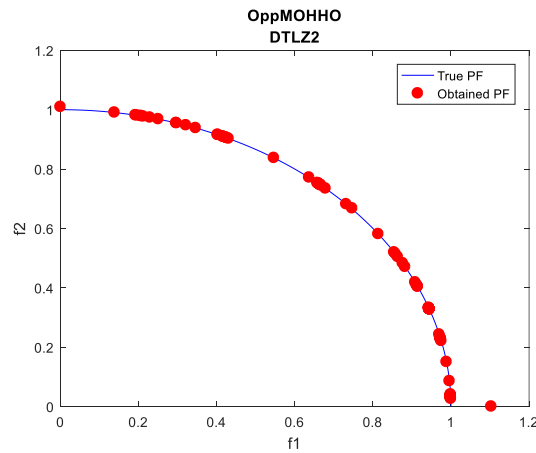
DTLZ test fonksiyonu serisi için IGD metriği sonuçlar incelendiğinde en iyi sonucu OppMOHHO algoritması vermiştir. Aşağıda sırası ile tüm DTLZ test fonksiyonları için IGD metriğinin grafikleri verilmiştir.

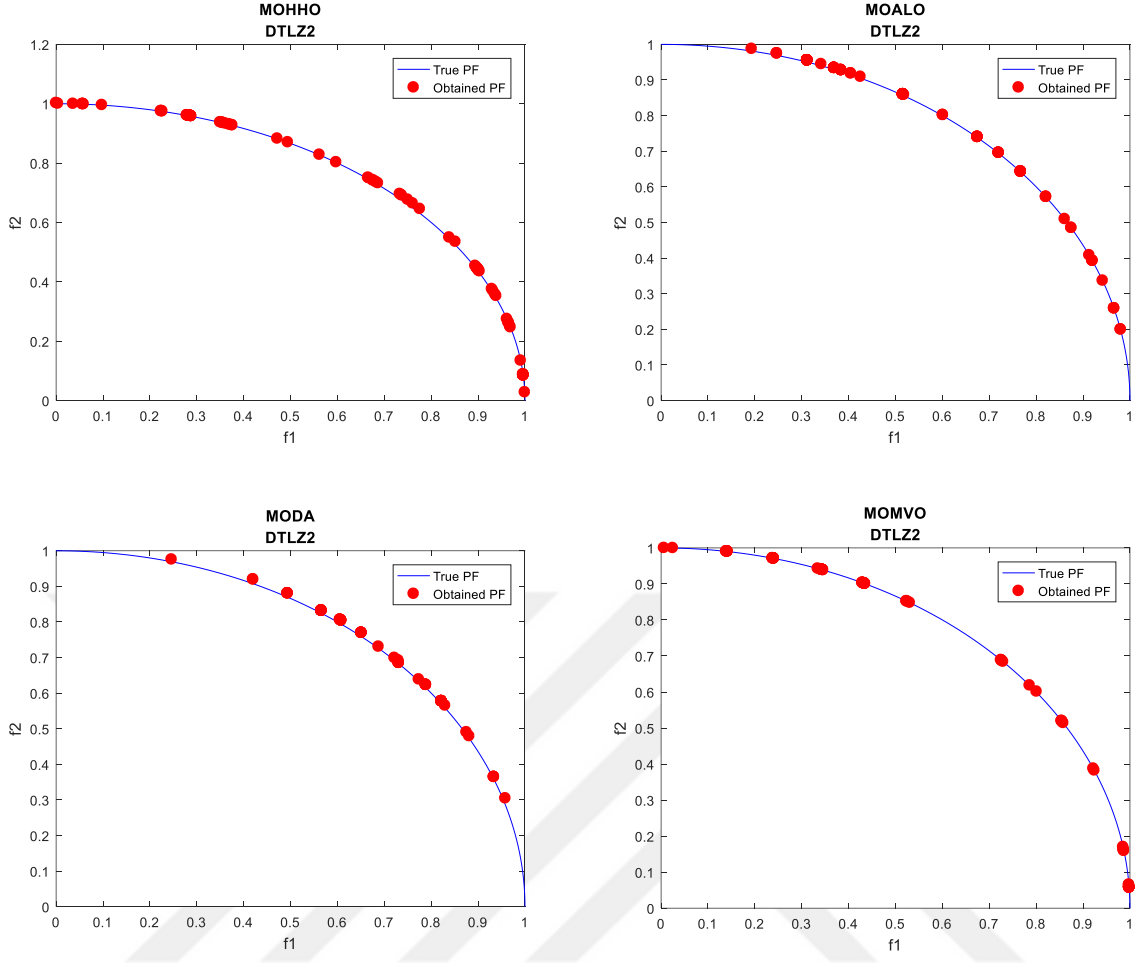




Şekil 3.8. DTLZ1 fonksiyonu için pareto optimal grafikleri

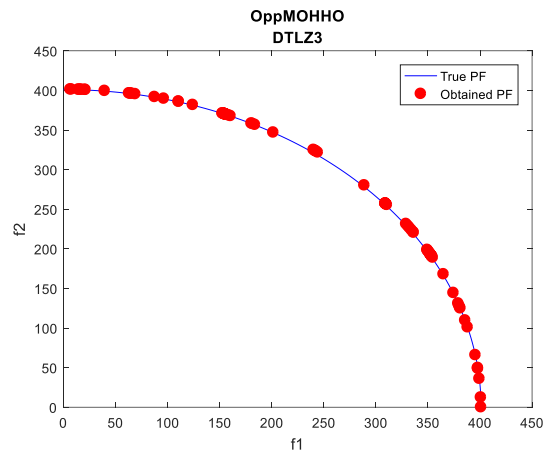
IGD metriği için Şekil 3.8.'de DTLZ1'in grafiği verilmiştir. Sonuçlar incelendiğinde OppMOHHO algoritmasının sonucunun daha iyi olduğu görülmektedir.

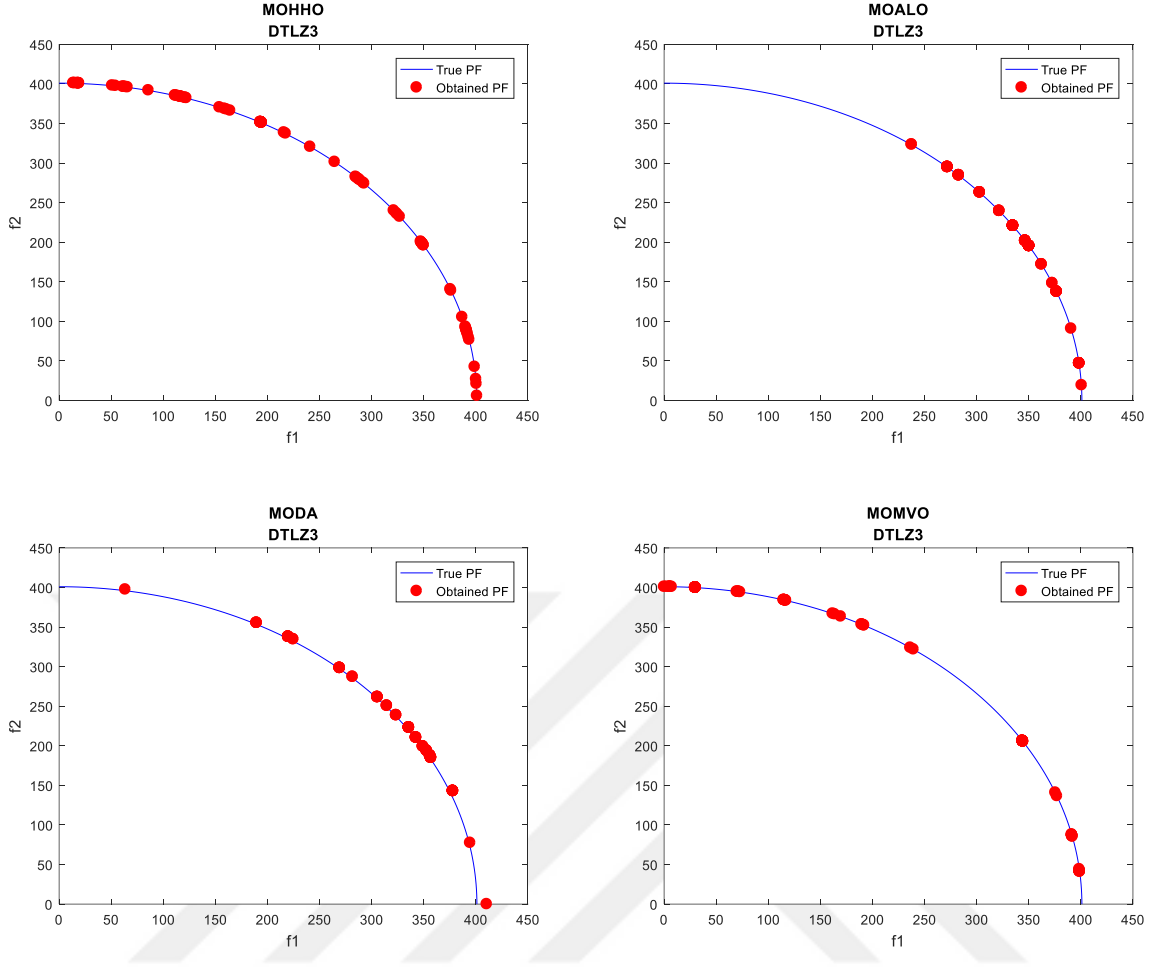




Şekil 3.9. DTLZ2 fonksiyonu için pareto optimal grafikleri

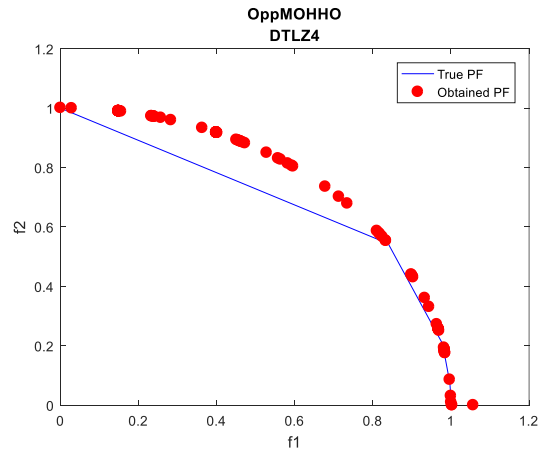
DTZL2 fonksiyonun grafiği için Şekil 3.9.'a bakıldığında OppMOHHO algoritmasının diğer algoritmalarla göre daha iyi sonuçlar aldığı görülmektedir.

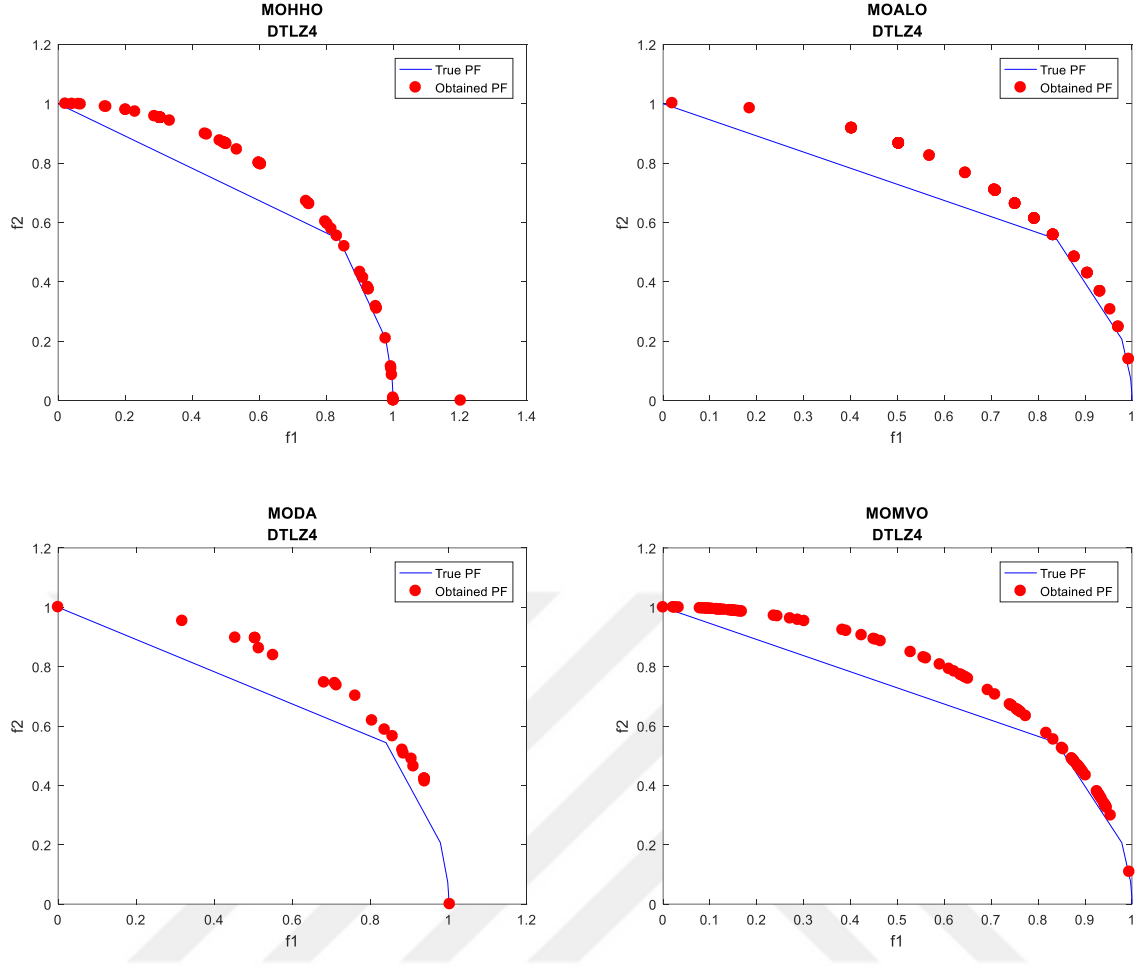




Şekil 3.10. DTLZ3 fonksiyonu için pareto optimal grafikleri

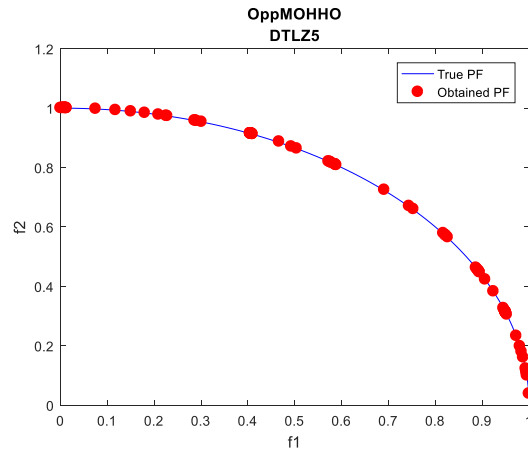
Şekil 3.10.'a bakıldığında DTLZ3 fonksiyonu için MOALO, MODA ve MOMVO algoritmalarının seyrek olduğu görülmektedir. MOHHO ve OppMOHHO algoritmalarına bakıldığında ise OppMOHHO algoritmasının daha başarılı olduğu görülmektedir.

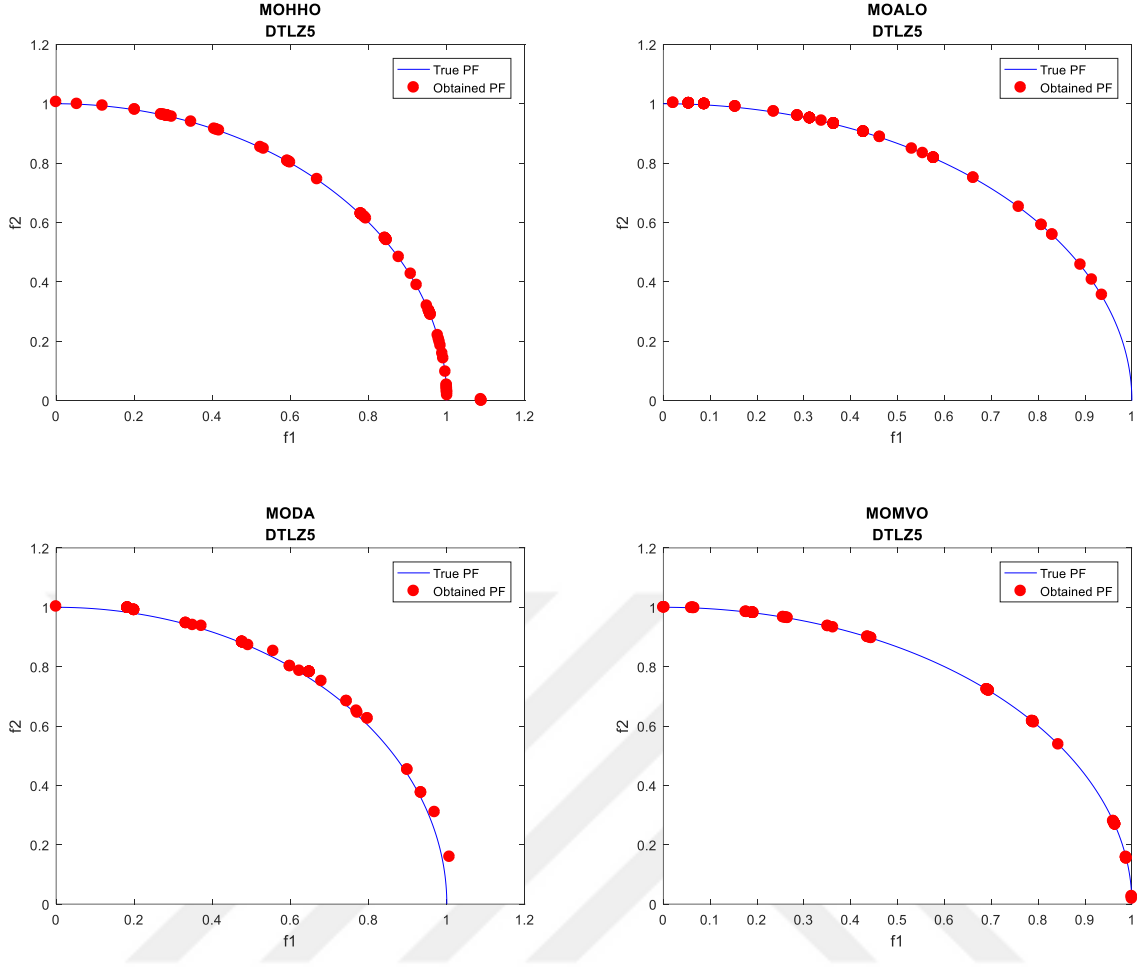




Şekil 3.11. DTLZ4 fonksiyonu için pareto optimal grafikleri

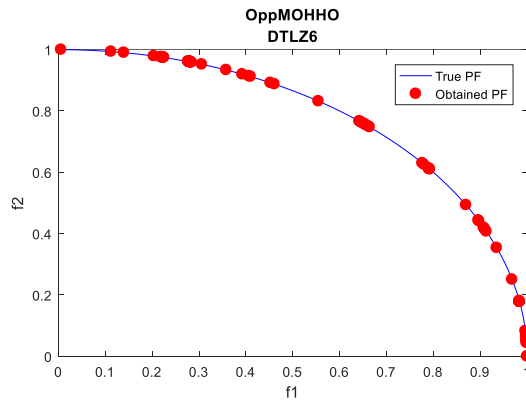
DTLZ4 fonksiyonun pareto optimal sonuçlarının grafikleri Şekil 3.11. incelendiğinde test fonksiyonunun sonucunu izleyen bir algoritmanın olmadığı görülmektedir. Aralarında en iyi çözümü MOHHO algoritması vermiştir.

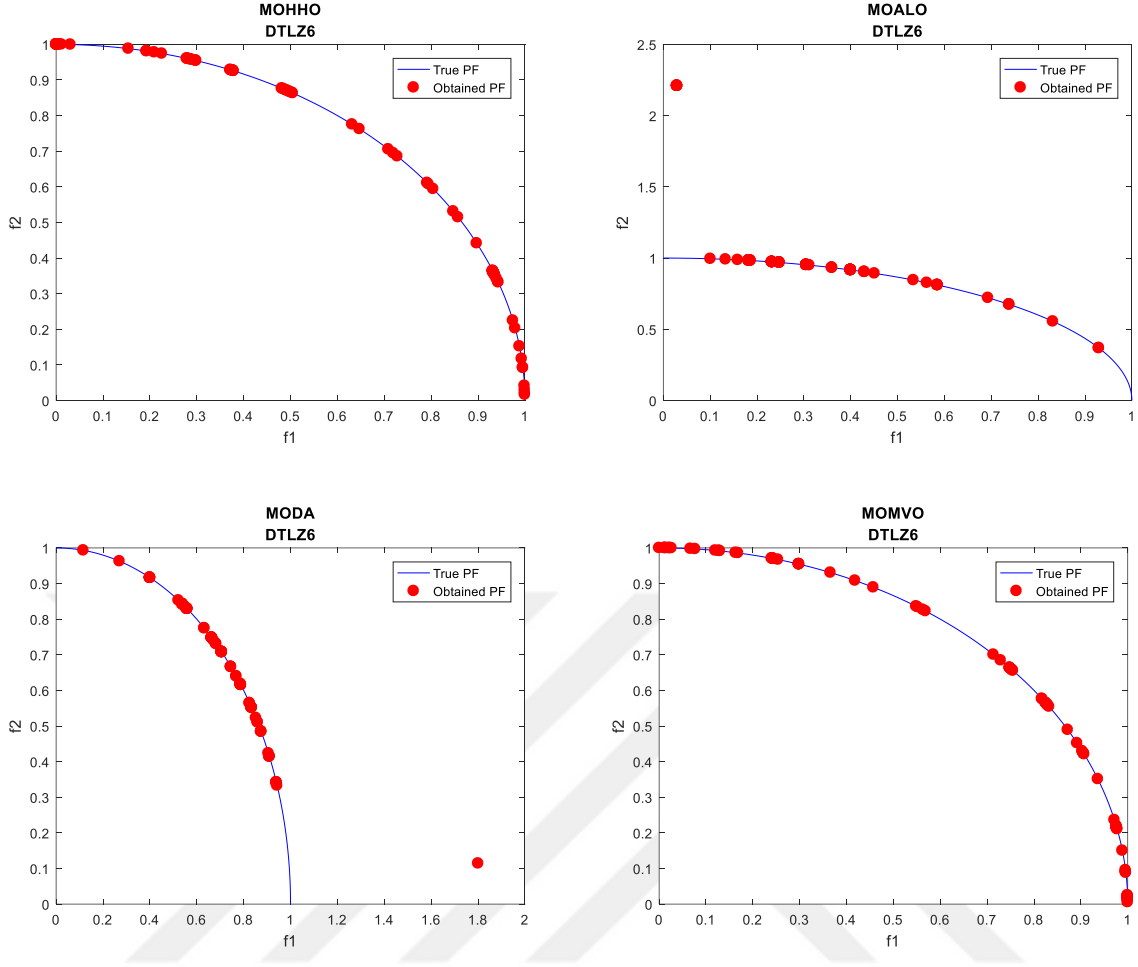




Şekil 3.12. DTLZ5 fonksiyonu için pareto optimal grafikleri

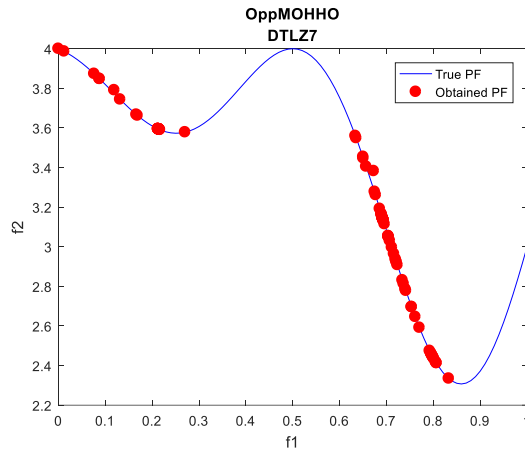
Şekil 3.12.'deki grafiklere bakıldığında DTLZ5 fonksiyonu için en iyi sonucu OppMOHHO algoritmasının verdiği görülmektedir. Diğer algoritmalarda sapma ve seyreklikler gözlemlenmektedir.

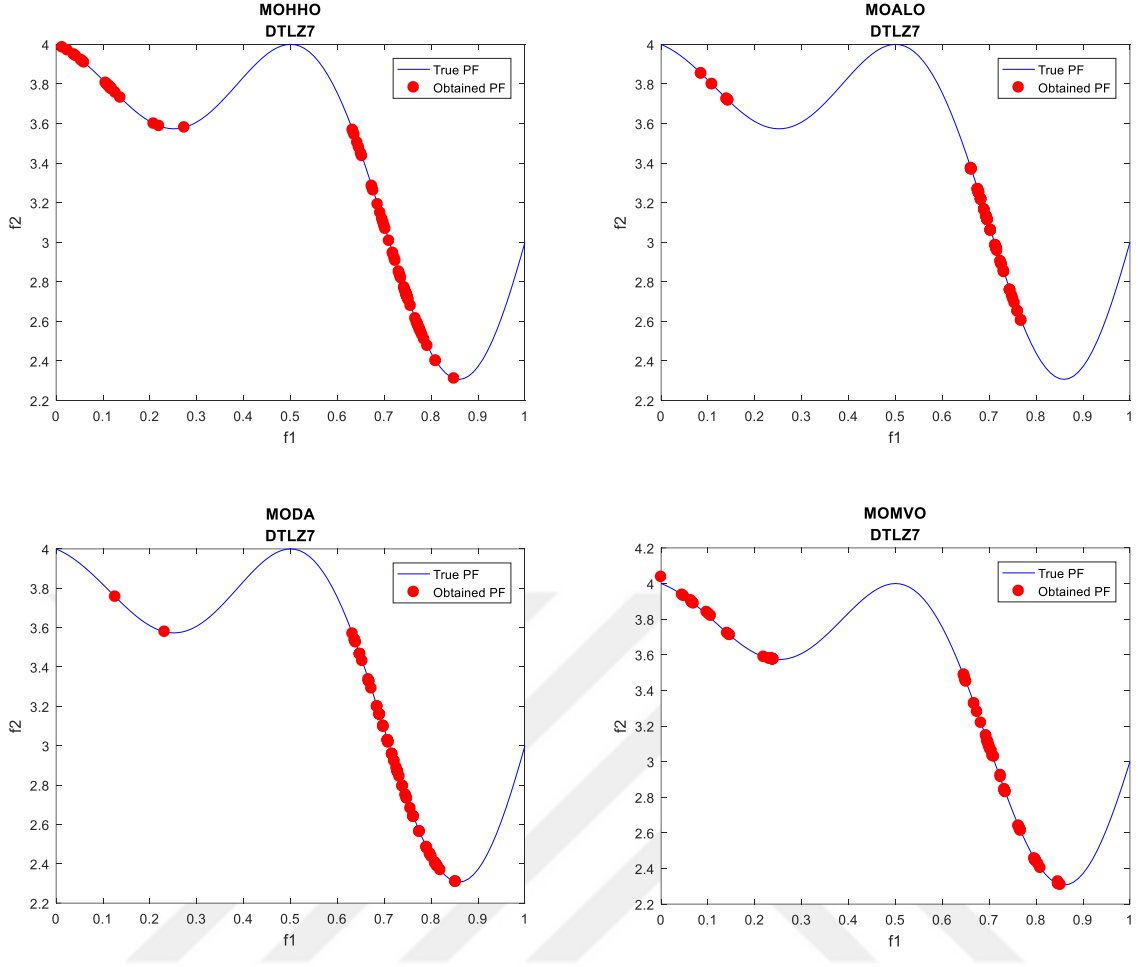




Şekil 3.13. DTLZ6 fonksiyonu için pareto optimal grafikleri

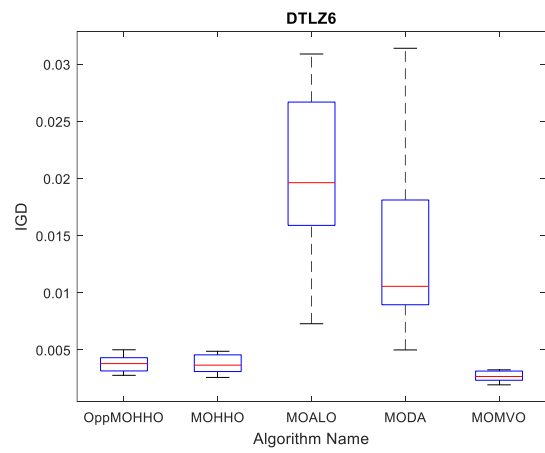
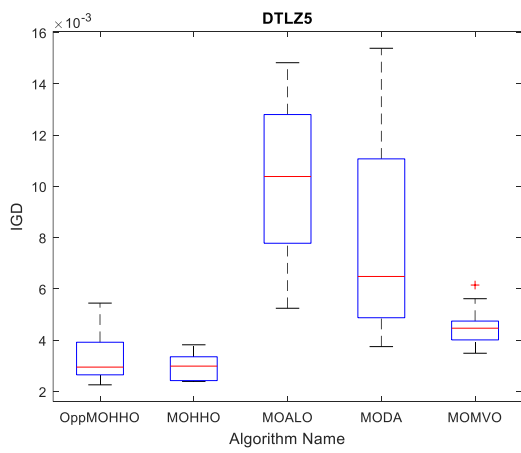
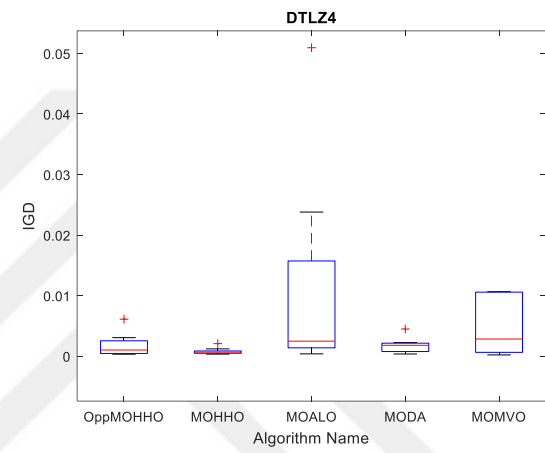
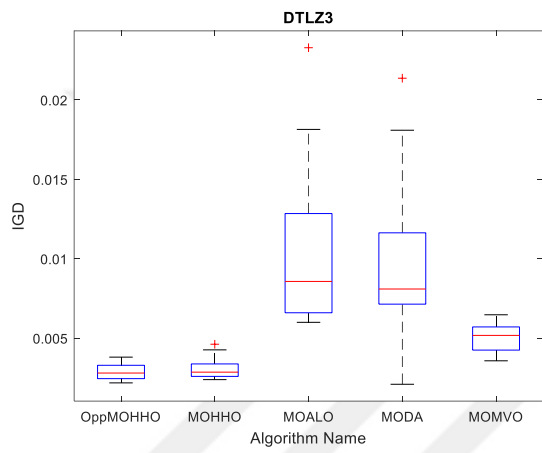
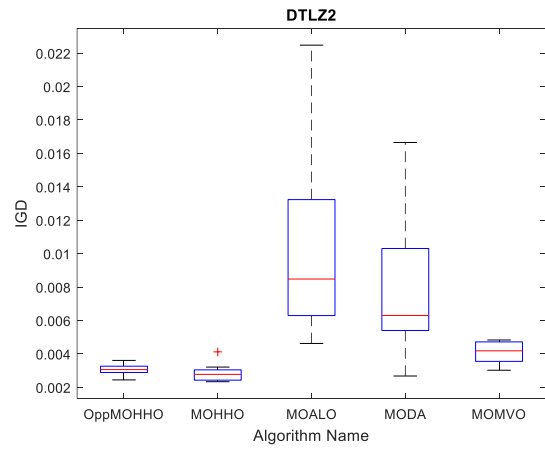
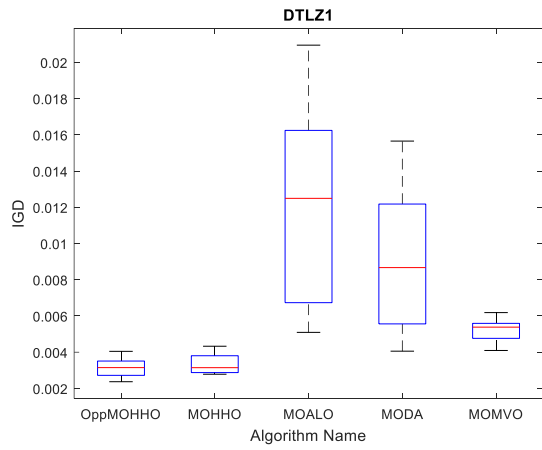
DTLZ6 için Şekil 3.13.'e bakıldığında OppMOHHO ve MOHHO algoritmalarında seyreklik, MOALO ve MODA algoritmalarında sapma görülmektedir. DTLZ6 test fonksiyonu için en iyi sonucu MOMVO algoritması vermektedir.

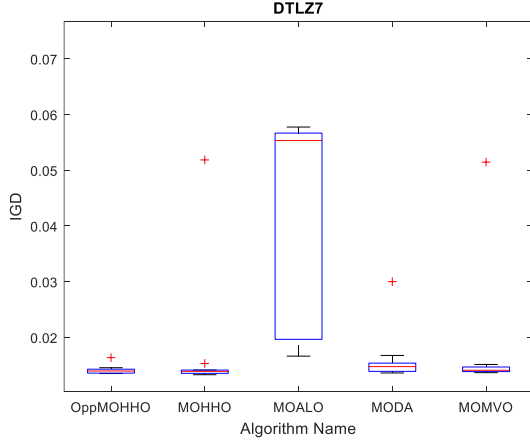




Şekil 3.14. DTLZ7 fonksiyonu için pareto optimal grafikleri

DTLZ test fonksiyonu sonuçları Şekil 3.14.'te bulunmaktadır. Grafikler incelendiğinde OppMOHHO algoritmasının daha başarılı olduğu görülmektedir. Diğer algoritmalar daha seyrek bir grafik çıktısı vermişlerdir. MOHHO algoritmasının grafiği iyi olsa da OppMOHHO algoritması daha iyidir. Sonuçlar ayrıca kutu grafiği olarak Şekil 3.15.'te bulunmaktadır.





Şekil 3.15. DTLZ test fonksiyon serisi için IGD metriği istatistiksel sonuçları

Şekil 3.15.'te DTLZ test fonksiyonlarının kutu grafiği verilmiştir. Grafiğe bakıldığında; DTLZ1 ve DTLZ2 test fonksiyonları için en iyi sonucu OppMOHHO algoritması verirken en kötü sonucu MOALO algoritması vermiştir. DTLZ3 fonksiyon sonucunda OppMOHHO algoritması yine en iyi sonucu vermiştir. En kötü sonucu MODA algoritması vermiştir. DTLZ4 test fonksiyonuna bakıldığında OppMOHHO ve MOHHO algoritmasının yakın olduğu görülmektedir. Fakat bu test fonksiyonunda yakın olmasına rağmen en iyi sonucu MOHHO algoritması vermiştir. DTLZ5 fonksiyonu için en iyi sonucu OppMOHHO algoritması vermiştir. DTLZ6 test fonksiyonuna bakıldığında en iyi sonucu MOMVO algoritması vermiştir. DTLZ7 fonksiyonunda ise OppMOHHO algoritması en iyi sonucu vermiştir.

Tablo 3.9. OppMOHHO ve diğer algoritmaların DTLZ fonksiyonu serisi için GD metrik sonuçları

GD	OppMOHHO	MOHHO	MOMVO	MOALO	MODA	
DTLZ1	Min	7.5135e-02	7.2157e-02	6.4571e-02	4.1416e-02	5.2711e-02
	Maks	1.1983e-01	1.0589e-01	9.9877e-02	1.0240e-01	1.0787e-01
	Ort.	9.0560e-02	8.8026e-02	8.5509e-02	7.8488e-02	8.1612e-02
	SD	1.2383e-02	9.9737e-03	1.1764e-02	1.9264e-02	1.7709e-02
	Sıra	5	4	3	1	2
DTLZ2	Min	4.2333e-04	3.8316e-04	3.2886e-04	3.2790e-04	1.2010e-03
	Maks	2.0734e-03	4.4404e-03	4.8696e-04	6.1254e-04	1.2211e-02
	Ort.	1.0267e-03	1.2670e-03	4.2009e-04	4.6018e-04	3.9809e-03
	SD	5.4360e-04	1.3646e-03	5.3665e-05	1.0220e-04	3.6227e-03
	Sıra	3	4	1	2	5
DTLZ3	Min	1.6544e-01	1.5755e-01	1.3508e-01	1.3664e-01	2.7637e-01
	Maks	3.4001e-01	3.5901e-01	2.2284e-01	2.5419e-01	5.9629e-01
	Ort.	2.4336e-01	2.2643e-01	1.7781e-01	1.9016e-01	3.9071e-01
	SD	5.9075e-02	5.7806e-02	3.2019e-02	3.3568e-02	8.9065e-02

	Sıra	4	3	1	2	5
DTLZ4	Min	1.5743e-02	1.7986e-02	1.8531e-02	1.3069e-02	1.1147e-02
	Maks	2.5912e-02	2.6519e-02	2.9730e-02	3.3158e-02	6.7945e-02
	Ort.	2.0231e-02	2.1897e-02	2.5101e-02	2.1793e-02	3.1231e-02
	SD	3.8755e-03	3.2670e-03	4.0839e-03	5.7713e-03	1.4893e-02
	Sıra	1	3	4	2	5
DTLZ5	Min	3.8682e-04	4.4981e-04	3.9164e-04	2.9464e-04	8.6642e-04
	Maks	2.3313e-03	3.8556e-03	5.3868e-04	5.6611e-04	4.0557e-03
	Ort.	8.6628e-04	1.3435e-03	4.9213e-04	4.6135e-04	2.0523e-03
	SD	5.7427e-04	9.6008e-04	4.7856e-05	8.2604e-05	9.8652e-04
	Sıra	3	4	2	1	5
DTLZ6	Min	3.7005e-04	4.0562e-04	3.8838e-04	3.8265e-04	3.7954e-04
	Maks	2.5967e-02	2.3641e-02	4.5769e-04	1.3520e-01	5.7559e-04
	Ort.	1.0243e-02	6.8968e-03	4.3451e-04	1.8712e-02	4.6978e-04
	SD	7.6554e-03	7.3096e-03	1.9697e-05	4.2173e-02	6.0562e-05
	Sıra	5	3	1	4	2
DTLZ7	Min	1.2831e-03	1.2075e-03	6.2871e-04	5.1911e-04	1.4907e-03
	Maks	2.6253e-03	2.7570e-03	9.2236e-02	1.9669e-03	2.1768e-03
	Ort.	1.7157e-03	1.6641e-03	1.9258e-02	8.7181e-04	1.7713e-03
	SD	4.0914e-04	4.5233e-04	3.7243e-02	4.4360e-04	2.3665e-04
	Sıra	3	2	5	1	4
Ort. Sıra	3.4285	3.2857	2.4285	1.8571	4.0000	
Genel Sıra	4	3	2	1	5	

GD metriği için DTLZ test fonksiyonu sonuçları incelendiğinde; DTLZ1 için MOLAO, DTLZ2 için MOMVO, DTLZ3 için MOMVO, DTLZ4 için OppMOHHO, DTLZ5 için MOALO, DTLZ6 için MOMVO ve DTLZ7 için de MOALO algoritmaları en iyi sonucu vermiştir.

Tablo 3.10 OppMOHHO ve diğer algoritmaların DTLZ fonksiyonu serisi için S metrik sonuçları

S	OppMOHHO	MOHHO	MOMVO	MOALO	MODA	
DTLZ1	Min	2.5054e+00	1.7121e+00	2.3983e-01	5.0623e-01	9.2383e-01
	Maks	5.7698e+00	6.5522e+00	3.2348e+00	2.9608e+00	1.3208e+01
	Ort.	3.8561e+00	3.5923e+00	1.1291e+00	1.8608e+00	7.7245e+00
	SD	1.1054e+00	1.2821e+00	1.0299e+00	9.3996e-01	4.2388e+00
	Sıra	4	3	1	2	5
DTLZ2	Min	1.1296e-02	6.8612e-03	1.1484e-03	4.4376e-03	9.8457e-03
	Maks	2.8995e-02	4.9066e-02	2.8864e-02	1.5238e-02	1.2277e-01
	Ort.	2.0623e-02	2.3384e-02	1.2564e-02	9.5996e-03	5.4778e-02
	SD	5.7117e-03	1.1313e-02	9.4051e-03	3.5410e-03	3.6828e-02
	Sıra	3	4	2	1	5
DTLZ3	Min	5.8432e+00	3.4643e+00	1.1535e+00	1.3369e+00	5.8568e+00

	Maks	8.8518e+00	9.7105e+00	6.6982e+00	8.2999e+00	2.2107e+01
	Ort.	7.2593e+00	7.0562e+00	3.0637e+00	5.1930e+00	1.2446e+01
	SD	1.1437e+00	1.7681e+00	1.6444e+00	2.3062e+00	4.8899e+00
	Sıra	4	3	1	2	5
DTLZ4	Min	1.1907e-02	9.7285e-03	8.2641e-03	8.8427e-03	1.4359e-02
	Maks	4.4097e-02	5.9204e-02	1.8047e-02	2.3196e-02	2.3955e-01
	Ort.	2.0404e-02	2.4256e-02	1.3292e-02	1.4903e-02	6.3000e-02
	SD	1.0786e-02	1.3707e-02	2.7892e-03	4.7133e-03	7.1915e-02
	Sıra	3	4	1	2	5
DTLZ5	Min	1.5184e-02	8.3132e-03	1.5492e-03	6.4939e-03	1.4536e-02
	Maks	2.1618e-02	5.1869e-02	1.7762e-02	1.8408e-02	5.1686e-02
	Ort.	1.7630e-02	2.0528e-02	1.0657e-02	1.0226e-02	3.1017e-02
	SD	2.3306e-03	1.2212e-02	5.4558e-03	3.6278e-03	1.1265e-02
	Sıra	3	4	2	1	5
DTLZ6	Min	1.1978e-02	1.3172e-02	7.8492e-03	5.7266e-03	1.0614e-02
	Maks	1.5709e-01	2.4938e-01	1.3239e-02	3.4521e-02	6.0323e-02
	Ort.	7.1344e-02	7.6966e-02	1.1046e-02	1.5980e-02	2.7117e-02
	SD	5.6579e-02	7.2492e-02	1.7588e-03	7.5819e-03	1.4412e-02
	Sıra	4	5	1	2	3
DTLZ7	Min	1.6885e-02	1.1238e-02	4.1199e-03	3.3999e-03	8.6764e-03
	Maks	2.7472e-02	3.5586e-02	9.2349e-01	1.1484e-02	7.8849e-02
	Ort.	2.0892e-02	1.8322e-02	1.9070e-01	6.4157e-03	3.3641e-02
	SD	4.1219e-03	7.0166e-03	3.7482e-01	2.9193e-03	2.1651e-02
	Sıra	3	2	5	1	4
	Ort. Sıra	3.4285	3.5714	1.8571	1.5714	4.5714
	Genel Sıra	3	4	2	1	5

S metriği için; DTLZ1 fonksiyonunda MOMVO, DTLZ2 için MOALO, DTLZ3 için MOMVO, DTLZ4 için MOMVO, DTLZ5 fonksiyonu için MOALO, DTLZ6 için MOMVO ve DTLZ7 içinde MOALO en iyi sonuçları vermiştir.

Tablo 3.11. OppMOHHO ve diğer algoritmaların DTLZ fonksiyonu serisi için Sp metrik sonuçları

	Sp	OppMOHHO	MOHHO	MOMVO	MOALO	MODA
DTLZ1	Min	1.3247e+00	1.4058e+00	1.5992e+00	1.3086e+00	1.0043e+00
	Maks	1.5264e+00	1.5916e+00	1.7702e+00	1.5663e+00	1.6467e+00
	Ort.	1.4519e+00	1.4733e+00	1.6965e+00	1.4063e+00	1.4433e+00
	SD	6.2408e-02	5.3387e-02	5.1638e-02	7.5267e-02	1.7353e-01
	Sıra	3	4	5	1	2
DTLZ2	Min	1.2890e+00	1.2962e+00	1.5699e+00	1.3438e+00	1.2614e+00
	Maks	1.3971e+00	1.4141e+00	1.6780e+00	1.4535e+00	1.6407e+00
	Ort.	1.3499e+00	1.3504e+00	1.6091e+00	1.4028e+00	1.4247e+00
	SD	3.5985e-02	3.5339e-02	3.6898e-02	4.0943e-02	1.1877e-01

	Sıra	1	2	5	3	4
DTLZ3	Min	1.2479e+00	1.2716e+00	1.5665e+00	1.3063e+00	1.3033e+00
	Maks	1.4282e+00	1.4909e+00	1.6871e+00	1.5679e+00	1.4448e+00
	Ort.	1.3336e+00	1.3685e+00	1.6242e+00	1.4428e+00	1.3557e+00
	SD	5.3584e-02	6.0933e-02	3.6830e-02	7.5579e-02	4.9034e-02
	Sıra	2	3	5	4	1
DTLZ4	Min	1.1335e+00	1.0562e+00	8.9916e-01	7.9578e-01	8.4148e-01
	Maks	1.3628e+00	1.3733e+00	1.1341e+00	1.6304e+00	1.5372e+00
	Ort.	1.3000e+00	1.2567e+00	1.0031e+00	1.4251e+00	1.1850e+00
	SD	6.4818e-02	1.0211e-01	6.9810e-02	2.5388e-01	2.0257e-01
	Sıra	4	3	1	5	2
DTLZ5	Min	1.1878e+00	1.2949e+00	1.4848e+00	1.3717e+00	1.2974e+00
	Maks	1.5036e+00	1.4033e+00	1.6821e+00	1.4903e+00	1.4739e+00
	Ort.	1.3376e+00	1.3357e+00	1.5948e+00	1.4126e+00	1.3764e+00
	SD	8.6611e-02	3.7791e-02	5.9922e-02	3.6283e-02	6.5544e-02
	Sıra	2	1	5	4	3
DTLZ6	Min	1.3296e+00	1.3408e+00	1.2337e+00	1.3402e+00	1.3084e+00
	Maks	1.5121e+00	1.5292e+00	1.4537e+00	1.5162e+00	1.5276e+00
	Ort.	1.3881e+00	1.3936e+00	1.3394e+00	1.4058e+00	1.4346e+00
	SD	5.8894e-02	5.4993e-02	6.3322e-02	4.8954e-02	8.1023e-02
	Sıra	1	3	2	4	5
DTLZ7	Min	1.2082e+00	1.1586e+00	1.0193e+00	1.1176e+00	1.2872e+00
	Maks	1.3164e+00	1.3331e+00	1.3936e+00	1.4456e+00	1.5274e+00
	Ort.	1.2574e+00	1.2610e+00	1.2610e+00	1.1914e+00	1.4316e+00
	SD	3.6824e-02	5.9214e-02	1.0299e-01	1.3263e-01	6.8611e-02
	Sıra	2	3	3	1	4
Ort. Sıra		2.1428	2.7142	3.7142	3.1428	3.0000
Genel Sıra		1	2	5	4	3

Sp metriği için; DTLZ1 için MOALO, DTLZ2 için OppMOHHO, DTLZ3 için MODA, DTLZ4 için MOMVO, DTLZ5 fonksiyonu için MOHHO, DTLZ6 için OppMOHHO ve DTLZ7 içinde MOALO en iyi sonuçları vermiştir. Genel duruma bakıldığında OppMOHHO algoritması daha başarılı bulunmuştur.

Tablo 3.12. OppMOHHO ve diğer algoritmaların DTLZ fonksiyonu serisi için MS metrik sonuçları

MS	OppMOHHO	MOHHO	MOMVO	MOALO	MODA	
DTLZ1	Min	9.4462e-01	9.1502e-01	8.3405e-01	3.8736e-01	6.2403e-01
	Maks	9.9442e-01	1.0020e+00	9.9860e-01	7.4824e-01	9.6347e-01
	Ort.	9.7711e-01	9.6807e-01	9.4604e-01	5.6273e-01	7.9085e-01
	SD	1.6474e-02	2.8942e-02	4.9038e-02	1.2592e-01	1.2509e-01
	Sıra	1	2	3	5	4
DTLZ2	Min	9.6036e-01	9.8518e-01	8.6054e-01	5.6436e-01	5.5524e-01

	Maks	1.1330e+00	1.3120e+00	9.9931e-01	7.7133e-01	1.3441e+00
	Ort.	1.0472e+00	1.0537e+00	9.6313e-01	6.7543e-01	9.4102e-01
	SD	4.7024e-02	1.0693e-01	4.0217e-02	7.7162e-02	2.2715e-01
	Sıra	1	2	3	5	4
DTLZ3	Min	9.8084e-01	9.8638e-01	9.1358e-01	4.9085e-01	5.0094e-01
	Maks	1.0465e+00	1.0440e+00	9.9051e-01	8.6321e-01	1.0075e+00
	Ort.	1.0112e+00	1.0094e+00	9.6084e-01	7.0066e-01	7.4562e-01
	SD	1.9861e-02	1.4980e-02	2.2460e-02	9.9277e-02	1.6652e-01
	Sıra	2	1	3	5	4
DTLZ4	Min	1.0067e+00	9.8546e-01	9.1102e-01	7.0352e-01	7.6639e-01
	Maks	1.1858e+00	1.3236e+00	9.9150e-01	1.0138e+00	1.0409e+00
	Ort.	1.0603e+00	1.0732e+00	9.6753e-01	9.0797e-01	9.1579e-01
	SD	6.3553e-02	1.0413e-01	2.4248e-02	1.1826e-01	9.8926e-02
	Sıra	1	2	2	4	3
DTLZ5	Min	9.8082e-01	9.7500e-01	9.1907e-01	6.0897e-01	5.1505e-01
	Maks	1.0759e+00	1.2222e+00	9.9091e-01	8.0481e-01	9.5414e-01
	Ort.	1.0256e+00	1.0625e+00	9.7160e-01	6.7446e-01	7.3437e-01
	SD	3.0054e-02	6.6901e-02	2.1030e-02	5.9945e-02	1.5698e-01
	Sıra	1	2	3	5	4
DTLZ6	Min	9.9721e-01	9.7282e-01	9.4675e-01	6.2605e-01	6.5833e-01
	Maks	1.9411e+00	2.6713e+00	9.9488e-01	2.7816e+00	9.7982e-01
	Ort.	1.4458e+00	1.4327e+00	9.8295e-01	1.0273e+00	8.1381e-01
	SD	3.1712e-01	5.3485e-01	1.3892e-02	6.6466e-01	1.1711e-01
	Sıra	2	1	4	3	5
DTLZ7	Min	8.8364e-01	8.7854e-01	2.5161e-01	1.5459e-01	6.5424e-01
	Maks	9.8362e-01	9.6333e-01	4.5962e+00	8.5107e-01	9.1990e-01
	Ort.	9.1865e-01	9.1515e-01	1.5687e+00	2.9921e-01	8.0016e-01
	SD	2.8242e-02	2.5219e-02	1.5586e+00	2.6605e-01	1.0328e-01
	Sıra	3	2	1	5	4
	Ort. Sıra	1.5714	1.7142	2.7142	4.5714	4.0000
	Genel Sıra	1	2	3	4	4

MS metriği için; DTLZ1 fonksiyonu için OppMOHHO, DTLZ2 için OppMOHHO, DTLZ3 için MOHHO, DTLZ4 için OppMOHHO, DTLZ5 fonksiyonu için OppMOHHO, DTLZ6 için MOHHO ve DTLZ7 içinde MOMVO en iyi sonuçları vermiştir. Genel duruma bakıldığında OppMOHHO algoritması daha başarılı bulunmuştur.

4. SONUÇ VE TARTIŞMA

Çalışmada ilk olarak HHO algoritmasının yapısı korunarak MOHHO algoritması geliştirilmiştir. Geliştirme sırasında HHO algoritmasına arşiv yapısı ve rulet çarkı seçme yöntemi eklenmiştir. Parametrelerde gerekli düzenlemeler yapıldıktan sonra MOHHO algoritması tamamlanmıştır. Daha sonra MOHHO algoritmasının sonuçlarını iyileştirmek için karşıtlık öğrenme temeli eklenmiştir. Gerekli parametre düzenlemeleri yapıldıktan sonra geliştirme tamamlanmıştır.

Önerdiği OppMOHHO algoritması literatürden alınan ZDT ve DTLZ test fonksiyonu serileri için birbirinden bağımsız 10 tekrarlı olarak orijinal MOHHO, MOALO, MODA ve MOMVO algoritmaları ile birlikte test edilmiştir. IGD, GD, S, Sp ve MS başarımları metriği sonuçları bakıldığında, ZDT test fonksiyonu için; IGD metriğinde OppMOHHO, GD metriğinde MODA, S metriğinde MOALO, Sp metriğinde MOMVO ve MS metriğinde MOMVO'nun yüksek yakınsama davranışına sahip olduğu nicel olarak göstermiştir. DTLZ test fonksiyonu için; IGD metriği için OppMOHHO, GD için MOALO, S metriği için MOALO, Sp için OppMOHHO ve MS metriği için de OppMOHHO algoritması yüksek yakınsama davranışına sahip olduğunu nicel olarak göstermiştir. Sonuç olarak, OppMOHHO algoritmasının ZDT ve DTLZ test fonksiyonlarında farklı metriklerde başarımları ve başarısızlıkları bulunmaktadır. Genel olarak bakıldığında orijinal MOHHO algoritması karşısında daha başarılı olduğu görülmektedir.

Gelecek çalışmalarda literatürde bulunan diğer test fonksiyon serileri ile ve başka popüler algoritmalar ile karşılaştırılıp başarısı gözlemlenebilir. Duruma göre başarımlarını arttırmak için OppMOHHO algoritmasına farklı yapılar ile geliştirilebilir.

KAYNAKÇA

- Abdel-Basset, M. M.** (2021). A novel whale optimization algorithm integrated with Nelder–Mead simplex for multi-objective optimization problems. *Knowledge-Based Systems*, 1-28.
- Akbel, M., & Kahraman, H.** (2020). Çok Amaçlı Meta-Sezgisel Optimizasyon Algoritmalarının Performanslarının Karşılaştırılması. *Mühendislik Bilimleri ve Tasarım Dergisi*, 185-199.
- Akdağ, O. A.** (2020). Harris Şahini Optimizasyon Algoritması ile Aktif Güç Kayıplarının Minimizasyonu. *Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi*, 481-490.
- Bao, X. J.** (2019). A novel hybrid harris hawks optimization for color image multilevel thresholding segmentation. *Ieee Access*, 76529-76546.
- Bui, D. T., Moayedi, H., Kalantar, B., Osouli, A., Pradhan, B., Nguyen, H., & Rashid, A. S.** (2019). A Novel Swarm Intelligence—Harris Hawks Optimization for Spatial Assessment of Landslide Susceptibility. *Sensors*, 1-22.
- Chen, H. H.** (2020). Multi-population differential evolution-assisted Harris hawks optimization: Framework and case studies. *Future Generation Computer Systems*, 175-198.
- Deb, K.** (2011). *Multi-objective optimisation using evolutionary algorithms: an introduction*. In *Multi-objective evolutionary optimisation for product design and manufacturing*. Springer, London.
- Deb, K. P.** (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 182-197.
- Deb, K., Thiele, L., Laumanns, M., & Zitzler, E.** (2005). Scalable test problems for evolutionary multiobjective optimization. *Evolutionary multiobjective optimization*, 105-145.
- Dréo, J., Pétrowski, A., Siarry, P., & Taillard, E.** (2006). *Metaheuristics for hard optimization: methods and case studies*. Springer Science & Business Media.
- Eshkevari, M.** (2020, 1 6). *Multi-objective-Harris-hawks-optimization-MOHHO*. Github: <https://github.com/miladesh1212/Multi-objective-Harris-hawks-optimization-MOHHO>
adresinden alındı

- Ewees , A. A., & Elaziz, M. A.** (2020). Performance analysis of Chaotic Multi-Verse Harris Hawks Optimization: A case study on solving engineering problems. *Engineering Applications of Artificial Intelligence*, 1-16.
- Fu , W., & Lu, Q.** (2020). Multiobjective Optimal Control of FOPID Controller for Hydraulic Turbine Governing Systems Based on Reinforced Multiobjective Harris Hawks Optimization Coupling with Hybrid Strategies. *Complexity*, 1-17.
- Golilarz, N. A.** (2019). A new automatic method for control chart patterns recognition based on ConvNet and harris hawks meta heuristic optimization algorithm. *IEEE Access*, 149398-149405.
- Golilarz, N. A.** (2019). Satellite image de-noising with Harris hawks meta heuristic optimization algorithm and improved adaptive generalized Gaussian distribution threshold function. *IEEE Access*, 57459-57468.
- Heidari, A. A.** (2019, 3 2). *Harris Hawks Optimization (HHO)*. EVOML RESEARCH GROUP: <http://evo-ml.com/2019/03/02/hho/> adresinden alındı
- Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H.** (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, 849-872.
- Hossain, M. A., Noor , R. M., Yau, K.-L. A., Azzuhri, S. R., Z'Abbar, M. R., Ahmedy, I., & Jabbarpour, M. R.** (2021). Multi-Objective Harris Hawks Optimization Algorithm Based 2-Hop Routing Algorithm for CR-VANET. *IEEE*, 58230-58242.
- Ishibuchi, H., & Murata, T.** (1996). Multi-objective genetic local search algorithm. *IEEE*, 119-124.
- Islam, M. Z.** (2020). A Harris Hawks optimization based single-and multi-objective optimal power flow considering environmental emission. *Sustainability*, 1-26.
- Ismael, O. M.** (2020). Improving Harris hawks optimization algorithm for hyperparameters estimation and feature selection in v-support vector regression based on opposition-based learning. *Journal of Chemometrics*, 1-14.
- Jia, H. L.** (2019). Dynamic harris hawks optimization with mutation mechanism for satellite image segmentation. *Remote sensing*, 1-35.

- Kamboj, V. K.** (2020). An intensify Harris Hawks optimizer for numerical and engineering optimization problems. *Applied Soft Computing*, 1-15.
- Karaboga, D., & Basturk, B.** (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*, 687-697.
- Kaya, S., & Fıglalı, N.** (2016). Çok Amaçlı Optimizasyon Problemlerinde Pareto Optimal Kullanımı. *Sosyal Bilimler Araştırma Dergisi*, 9-18.
- Ke, L., Zhang, Q., & Battiti, R.** (2014). Hybridization of Decomposition and Local Search for Multiobjective Optimization. *IEEE transactions on cybernetics*, 1808-1820.
- MathWorks.** (2019, 03 21). *Minimize multiple objective functions subject to constraints*. MathWorks: <https://www.mathworks.com/discovery/multiobjective-optimization.html> adresinden alındı
- Mirjalili, S.** (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, 228-249.
- Mirjalili, S.** (2016). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. . *Neural Computing and Applications*, 1053-1073.
- Mirjalili, S.** (2018). Genetic Algorithm. *Evolutionary Algorithms and Neural Networks*, 43-55.
- Mirjalili, S.** (2020). *Harris Hawks Optimization (HHO)*. <http://mdm.wzu.edu.cn/HHO.html> adresinden alındı
- Mirjalili, S., & Dong, J. S.** (2020). Multi-objective Particle Swarm Optimization. *Multi-Objective Optimization using Artificial Intelligence Techniques*, 21-36.
- Mirjalili, S., Jangir, P., & Saremi, S.** (2017). Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems. *Applied Intelligence*, 79-95.
- Mirjalili, S., Jangir, P., Mirjalili, S., Saremi, S., & Trivedi, I.** (2017). Optimization of problems with multiple objectives using the multi-verse optimization algorithm. *Knowledge-Based Systems*, 1-22.
- Rahnamayan, S. T.** (2006). Opposition-based differential evolution algorithms. *IEEE International Conference on Evolutionary Computation* (s. 2010-2017). IEEE.

Serafini, P. (1994). *Simulated Annealing for Multi Objective Optimization Problems*. Springer, New York: NY.

Shadkam, E., & Bijari, M. (2014). Evaluation the efficiency of cuckoo optimization algorithm. *International Journal on Computational Sciences & Applications (IJCSA)*, 39-47.

Talbi, E.-G. (2009). *Metaheuristics: From Design to Implementation*. John Wiley & Sons.

Tubishat, M. I. (2020). Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection. *Expert Systems with Applications*, 1-10.

Zhang, X., Tian , Y., Cheng, R., & Jin, Y. (2014). An Efficient Approach to Nondominated Sorting for Evolutionary Multiobjective Optimization. *IEEE*, 201-213.

Zitzler, E., Deb, K., & Thiele., L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation* , 173-195.