



**T.C.
DÜZCE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

ELEKTRİK-ELEKTRONİK VE BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**YERÇEKİMİ ARAMA ALGORİTMASI İÇİN YENİ OPERATÖRLERİN
GELİŞTİRİLMESİ**

DOKTORA TEZİ

FERZAN KATIRCIOĞLU

MAYIS 2016

DÜZCE

KABUL VE ONAY BELGESİ

Ferzan KATIRCIOĞLU tarafından hazırlanan "Yerçekimi Arama Algoritması İçin Yeni Operatörlerin Geliştirilmesi" isimli lisansüstü tez çalışması, Düzce Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 02.05.2016 tarih ve 2016/431 sayılı kararı ile oluşturulan jüri tarafından Elektrik-Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı'nda Doktora Tezi olarak kabul edilmiştir.

Üye
(Tez Danışmanı)
Doç. Dr. Uğur GÜVENÇ
Düzce Üniversitesi

Üye
Doç. Dr. Oğuz ÜSTÜN
Abant İzzet Baysal Üniversitesi

Üye
Doç. Dr. Resul KARA
Düzce Üniversitesi

Üye
Doç. Dr. Pakize ERDOĞMUŞ
Düzce Üniversitesi

Üye
Yrd. Doç. Dr. Nihat DALDAL
Abant İzzet Baysal Üniversitesi

Tezin Savunulduğu Tarih : 20 Mayıs 2016

ONAY

Bu tez ile Düzce Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu Ferzan KATIRCIOĞLU'nun Elektrik-Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı'nda Doktora derecesini almasını onamıştır.

Doç. Dr. Resul KARA
Fen Bilimleri Enstitüsü Müdürü

BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını beyan ederim.

20 Mayıs 2016

Ferzan KATIRCIOĞLU



Sevgili kızım Elif Berra'ya ithaf olunur.

TEŐEKKÜR

Doktora öğrenimim ve bu tezin hazırlanmasında süresince gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocam Doç. Dr. Uğur GÜVENÇ'e en içten dileklerle teşekkür ederim.

Tez çalışmam boyunca değerli katkılarını esirgemeyen Doç. Dr. Resul KARA'ya şükranlarımı sunarım. Ayrıca uygulama ve teori çalışmalarında bana yardım eden Yrd. Doç. Dr. Serhat KIRIK ve Öğr. Gör. Zafer CİNGİZ'e çok teşekkür ederim.

Bu çalışmada ve hayatımın birçok evrelerinde maddi ve manevi desteklerini esirgemeyen tüm aile üyelerine sonsuz teşekkürlerimi sunarım.

20 Mayıs 2016

Ferzan KATIRCIOĞLU

TEŞEKKÜR.....	i
İÇİNDEKİLER.....	ii
ŞEKİL LİSTESİ	iv
ÇİZELGE LİSTESİ	vi
SİMGELER VE KISALTMALAR.....	ix
ÖZET.....	1
ABSTRACT.....	2
EXTENDED ABSTRACT	3
1. GİRİŞ	7
1.1. AMAÇ VE KAPSAM	8
1.2. GENEL BİLGİLER.....	11
1.2.1. Optimizasyon Problemlerinin Sınıflandırılması	14
1.2.2. Optimizasyon Metotları	14
1.2.3. Sezgisel Optimizasyon Teknikleri	15
1.3. YERÇEKİMİ ARAMA ALGORİTMASI LİTERATÜR ÖZETİ	17
1.3.1. YAA Geliştirme Çalışmaları.....	22
1.3.2. YAA Yerçekimi Parametresi Üzerine Yapılan Çalışmalar	24
2. MATERYAL VE YÖNTEM.....	28
2.1. YERÇEKİMİ ARAMA ALGORİTMASI.....	28
2.2. YAA PARAMETRELERİNİN İNCELENMESİ.....	388
2.2.1. Yerçekimi Başlangıç Sabiti (G_0)'ın YAA Üzerindeki Etkisi	38
2.2.2. YAA α Parametresinin $G(t)$ Üzerindeki Etkisinin İncelenmesi.....	40
2.2.3. Ajan Sayısının YAA Üzerindeki Etkisinin İncelenmesi	43
2.2.4. Döngü Sayısının YAA Üzerindeki Etkisinin İncelenmesi	44
2.2.5. Kbest Ajanlarının YAA Üzerindeki Etkisinin İncelenmesi	45
2.2.6. YAA son_ajan Değerinin YAA Üzerindeki Etkisinin İncelenmesi	47
2.2.7. Kbest Farklı Azalma Eğrileri Altında, YAA'ya Etkisinin İncelenmesi	50
2.3. YAA İÇİN GELİŞTİRİLEN OPERATÖRLER.....	53

2.3.1. Kaotik Sarsıntılı Operatör	53
2.3.2. Kurtulma Hız Operatörü	62
2.3.3. En İyi Ajan Operatörü	68
3. BULGULAR VE TARTIŞMA.....	73
3.1. KSYAA UYGULAMA SONUÇLARI	76
3.2. KHYAA UYGULAMA SONUÇLARI.....	90
3.3. EİAYAA UYGULAMA SONUÇLARI.....	100
4. SONUÇLAR VE ÖNERİLER.....	110
5. KAYNAKLAR.....	115
6. EKLER	125
EK-1. TEK MODLU KARŞILAŞTIRMA TEST FONKSİYONLARI.....	125
EK-2. ÇOK MODLU KARŞILAŞTIRMA TEST FONKSİYONLARI	126
EK-3. ÇOK MODLU DÜŞÜK BOYUTLU TEST FONKSİYONLARI.....	127
ÖZGEÇMİŞ	128

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 2.1. Newton yerçekimi kanununun şekilsel gösterimi [4]	28
Şekil 2.2. Ajana etkiyen toplam kuvvetin şekilsel gösterimi	33
Şekil 2.3. Yerçekimi arama algoritmasının akış şeması [4]	37
Şekil 2.4. YAA α parametresinin yerçekimi parametresi üzerine etkisi	42
Şekil 2.5. Her döngüde kuvvete tesir eden ajanların gösterimi	46
Şekil 2.6. Her son döngüde kullanılan <i>son_ajan</i> grafikleri	49
Şekil 2.7. Farklı <i>Kbest</i> fonksiyonlarının azalma grafikleri	52
Şekil 2.8. Yerel minimuma yaklaşma ve takılma durumu	54
Şekil 2.9. F11 için küresel en iyi değerin değişmeme durumunun gözlemlenmesi.	55
Şekil 2.10. F6 için küresel en iyi değerin değişmeme durumunun gözlemlenmesi	57
Şekil 2.11. Yerel minimuma takılma durumunun tespiti	58
Şekil 2.12. Kaotik sarsıntı oluşturma sözel kodu	58
Şekil 2.13. Kaotik sarsıntı sonucu yerel minimumdan kurtulma durumu	59
Şekil 2.14. Kaotik haritaların görünümleri	60
Şekil 2.15. Kaotik sarsıntı operatörü akış şeması	61
Şekil 2.16. Ajanların sembolik hareketlerinin gösterimi	62
Şekil 2.17. Tek merkezli basit çekim alanındaki kurtulma hızı	63
Şekil 2.18. Grupsal hareketinin dışındaki ajanlara kurtulma hızı uygulanmış gösterimi	64
Şekil 2.19. Dışta kalmış ajanlara negatif kurtulma hızı uygulanmış gösterimi	65
Şekil 2.20. KHYAA sözel kodu	66
Şekil 2.21. KHYAA akış şeması	68
Şekil 2.22. <i>Kbest</i> azalma oranı	69
Şekil 2.23. EİAYAA sözel kodu	70
Şekil 2.24. En iyi ajan operatörü akış şeması	71
Şekil 3.1. YAA ile KSYAA algoritmalarının tek modlu fonksiyonlarda yakınsama özelliklerinin karşılaştırılması	77

Şekil 3.2.	KSYAA tek modlu fonksiyonlarda yerçekimi parametresinin değişim eğrisi	78
Şekil 3.3.	YAA ile KSYAA çok modlu fonksiyonlarda yakınsama özelliklerinin karşılaştırılması	81
Şekil 3.4.	KSYAA çok modlu fonksiyonlarda yerçekimi parametresinin değişim eğrisi	82
Şekil 3.5.	YAA ile KSYAA yakınsama özelliklerinin karşılaştırılması	85
Şekil 3.6.	KSYAA uygulama sonrasında yerçekimi parametresinin değişim eğrisi	86
Şekil 3.7.	KSYAA kaotik fonksiyonlarının sarsıntı eğrileri	89
Şekil 3.8.	YAA ile KHYAA yakınsama özelliklerinin karşılaştırılması	91
Şekil 3.9.	YAA ile KHYAA çok modlu yakınsama özelliklerinin karşılaştırılması	94
Şekil 3.10.	YAA ile KHYAA algoritmasının çok modlu düşük boyutlu fonksiyonlarda yakınsama özelliklerinin karşılaştırılması	97
Şekil 3.11.	YAA ile EİAYAA tek modlu fonksiyonlarda yakınsama özelliklerinin karşılaştırılması	101
Şekil 3.12.	YAA ile EİAYAA çok modlu fonksiyonlarda yakınsama özelliklerinin karşılaştırılması	104
Şekil 3.13.	YAA ve EİAYAA arasında çok modlu düşük boyutlu fonksiyonlarda yakınsama özelliklerinin karşılaştırılması	107

ÇİZELGE LİSTESİ

	<u>Sayfa No</u>	
Çizelge 2.1.	F1 için G_0 başlangıç değerinin YAA'ya etkisi	39
Çizelge 2.2.	F12 için G_0 başlangıç değerinin YAA'ya etkisi	39
Çizelge 2.3.	F23 için G_0 başlangıç değerinin YAA'ya etkisi	40
Çizelge 2.4.	F2 için α parametresi etkisinin gözlemlenmesi	41
Çizelge 2.5.	F3 için ajan sayısının YAA ile ilişkisi	43
Çizelge 2.6.	F15 için ajan sayısının YAA ile ilişkisi	44
Çizelge 2.7.	F4 için döngü sayısının YAA ile ilişkisi	45
Çizelge 2.8.	F21 için döngü sayısının YAA ile ilişkisi	45
Çizelge 2.9.	F1 için K_{best} ajan sayısının YAA ile ilişkisi	47
Çizelge 2.10.	F2 için K_{best} ajan sayısının YAA ile ilişkisi	47
Çizelge 2.11.	F3 için son_ajan ile YAA ile ilişkisi	48
Çizelge 2.12.	F10 için K_{best} farklı azalma fonksiyonlarındaki davranışının incelenmesi	51
Çizelge 2.13.	F13 için K_{best} farklı azalma fonksiyonlarındaki davranışının incelenmesi	51
Çizelge 2.14.	Algoritmada kullanılan kaotik fonksiyonlar	60
Çizelge 3.1.	Z değeri istatistik değerler tablosu	75
Çizelge 3.2.	KSYAA tek modlu fonksiyonlarının minimize sonuçları	76
Çizelge 3.3.	KSYAA F1 için Wilcoxon işaretli sıra testi sonucu	79
Çizelge 3.4.	KSYAA tek modlu için Wilcoxon işaretli sıra testi sonuçları	79
Çizelge 3.5.	YAA ve KSYAA arasında tek modlu fonksiyonlar için standart sapma ve işlem süresi karşılaştırması.	80
Çizelge 3.6.	KSYAA yüksek modlu çok boyutlu fonksiyonlarının minimize sonuçları	80
Çizelge 3.7.	KSYAA F8 için Wilcoxon işaretli sıra testi sonucu	83
Çizelge 3.8.	KSYAA çok modlu için Wilcoxon işaretli sıra testi sonuçları	83
Çizelge 3.9.	YAA ve KSYAA arasında standart sapma ve işlem süresi karşılaştırması	84

Çizelge 3.10.	KSYAA yüksek modlu düşük boyutlu minimize sonuçları	84
Çizelge 3.11.	F16 fonksiyonu için Wilcoxon işaretli sıra testi sonucu	87
Çizelge 3.12.	F17 fonksiyonu için Wilcoxon işaretli sıra testi sonucu	87
Çizelge 3.13.	KSYAA çok modlu düşük boyutlu fonksiyonları için Wilcoxon işaretli sıra testi sonuçları	88
Çizelge 3.14.	YAA ve KSYAA arasında çok modlu düşük boyutlu fonksiyonları için standart sapma ve süre karşılaştırması.	88
Çizelge 3.15.	Kaotik fonksiyonların performans sonuçları	88
Çizelge 3.16.	KHYAA tek modlu fonksiyonların minimize sonuçları	90
Çizelge 3.17.	KHYAA F5 için Wilcoxon işaretli sıra testi sonucu	92
Çizelge 3.18.	YAA ve KHYAA tek modlu Wilcoxon işaretli sıra testi sonuçları	92
Çizelge 3.19.	YAA ve KHYAA arasında tek modlu fonksiyonlar için standart sapma ve işlem süresi karşılaştırması	93
Çizelge 3.20.	KHYAA yüksek modlu çok boyutlu fonksiyonların minimize sonuçları	93
Çizelge 3.21.	F10 fonksiyonu için Wilcoxon işaretli sıra testi sonucu	95
Çizelge 3.22.	KHYAA çok modlu fonksiyonlar için Wilcoxon işaretli sıra testi sonuçları	95
Çizelge 3.23.	YAA ve KHYAA arasında çok modlu fonksiyonlar için standart sapma ve işlem süresi karşılaştırması	95
Çizelge 3.24.	KHYAA yüksek modlu düşük boyutlu fonksiyonların minimize sonuçları	96
Çizelge 3.25.	KHYAA F17 için Wilcoxon işaretli sıra testi sonucu	98
Çizelge 3.26.	KHYAA F15 için Wilcoxon işaretli sıra testi sonucu	98
Çizelge 3.27.	KHYAA çok modlu düşük boyutlu fonksiyonlar için Wilcoxon işaretli sıra testi sonuçları	99
Çizelge 3.28.	YAA ve KHYAA arasında standart sapma ve işlem süresi karşılaştırması	99
Çizelge 3.29.	EİAYAA tek modlu fonksiyonların minimize sonuçları	100
Çizelge 3.30.	EİAYAA F5 için Wilcoxon işaretli sıra testi sonucu	102
Çizelge 3.31.	EİAYAA tek modlu Wilcoxon işaretli sıra testi sonuçları.	102
Çizelge 3.32.	YAA ve EİAYAA arasında tek modlu fonksiyonlar için	103

	standart sapma ve işlem süresi karşılaştırması	
Çizelge 3.33.	EİAYAA yüksek modlu çok boyutlu minimize sonuçları	96
Çizelge 3.34.	EİAYAA F10 için Wilcoxon işaretli sıra testi sonucu	105
Çizelge 3.35.	EİAYAA çok modlu Wilcoxon işaretli sıra testi sonuçları	105
Çizelge 3.36.	YAA ve EİAYAA arasında çok modlu fonksiyonlar için standart sapma ve işlem süresi karşılaştırması	106
Çizelge 3.37.	EİAYAA düşük boyutlu fonksiyonların minimize sonuçları	106
Çizelge 3.38.	EİAYAA F18 için Wilcoxon işaretli sıra testi sonucu	108
Çizelge 3.39.	EİAYAA F15 için Wilcoxon işaretli sıra testi sonucu	108
Çizelge 3.40.	EİAYAA çok modlu düşük boyutlu fonksiyonlar için Wilcoxon işaretli sıra testi sonuçları	109
Çizelge 3.41.	YAA ve EİAYAA arasında standart sapma ve işlem süresi karşılaştırması	109

SİMGELER VE KISALTMALAR

BT	Benzetimli Tavlama
BUYAA	Bulanık Uyarlamalı Yerçekimi Arama Algoritması
ÇDKY	Çapraz Doğrulama Komşu Yöntemi
DED	Dinamik Ekonomik Dağıtım
EİAYAA	En İyi Ajan Yerçekimi Arama Algoritması
GA	Genetik Algoritma
GDYAA	Gerçek Değerli Yerçekimi Arama Algoritması
İYAA	İkili Yerçekimi Arama Algoritması
KEYAA	Kuantum Esinlemeli Yerçekimi Arama Algoritması
KHO	K Harmonik Ortalama
KHYAA	Kurtulma Hızlı Yerçekimi Arama Algoritması
K_i	İntegral Kazancı
K_o	Oransal Kazanç
KSYYAA	Kaotik Sarsıntılı Yerçekimi Arama Algoritması
K_t	Türevsel Kazanç
KYA	Kaotik Yerel Arama
KYAA	Kaotik Yerçekimi Arama Algoritması
K-YK	K-En Yakın Komşu Yöntemi
MDP	Montaj Dizi Planlama
MKP	Motif Keşfetme Problemi
MÖ	Milattan Önce
OITK	Oransal İntegral Türevsel Kontrol
PDHB	Pompa Depolama Hidro Birimi
PSO	Parçacık Sürü Optimizasyonu
PSOYAA	Parçacık Sürü Optimizasyonlu Yerçekimi Arama Algoritması
SED	Statik Ekonomik Dağıtım
TAA	Tabu Araştırma Algoritması
TPM	Tamsayı Programlama Modeli
YAA	Yerçekimi Arama Algoritması
YAA-K	Kümeleme Yerçekimi Arama Algoritması
YAA-KO	K Ortalama Yerçekimi Arama Algoritması

YZ	Yapay Zeka
ZEKUYAA	Zıt Esaslı Kendinden Uyarlamalı Yerçekimi Arama Algoritması
a	Ajanın ivmesi
<i>best</i>	En iyi uygunluk değeri
n	Fonksiyonların boyut sayısı
Fbest	En iyi uygunluk değeri
F	Ajana tesir eden kuvvet
<i>son_don</i>	Kurtulmaya tabii olacak son döngüdeki ajan sayısı
<i>son_ajan</i>	Son döngüde kuvvet hesabında yer alacak ajana sayısı
<i>fit_i</i>	İlgili ajanın uygunluk değeri
G	Yerçekimi parametresi
G_0	Yerçekimi sabiti başlangıç değeri
<i>hbest</i>	Kurtulma hızına tabii olacak ajan sayısı değeri
<i>Kbest</i>	En iyi uygunluk sayısı
<i>Kworst</i>	En kötü uygunluk sayısı
M	Ajanın kütlesi
M_b	En büyük kütle
M_{En}	En iyi ajana diğer ajanların mesafe vektörü
R	Ajanlar arasındaki öklid mesafesi
<i>rand_i</i>	[0,1] Aralığında rastgele bir değer
t_0	Kozmik kuantum zamanı
t	Arama anındaki döngü değeri
T	Maksimum döngü sayısı
<i>Takılma_say</i>	En iyi değer değişmeme döngü sayısı
X	Ajanın arama uzayı içerisindeki pozisyonu
V	Ajanın hızı
V_k	KHYAA kurtulma hızı
<i>worst</i>	En kötü uygunluk değeri
α	Yerçekimi parametresi azalma oranı
ϵ	Kuvvet hesabında küçük sabit bir sayı
$\beta_{1,2}$	<i>Kbest</i> üstel fonksiyonların azalma sabiti
ω	Evrenin gerçek yaş belirleme katsayısı
δ	Gerçek dünya yerçekimi kuvveti azalma oranı

ÖZET

YERÇEKİMİ ARAMA ALGORİTMASI İÇİN YENİ OPERATÖRLERİN GELİŞTİRİLMESİ

Ferzan KATIRCIOĞLU

Düzce Üniversitesi

Fen Bilimleri Enstitüsü, Elektrik-Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı

Doktora Tezi

Danışman: Doç. Dr. Uğur GÜVENÇ

Mayıs 2016, 129 sayfa

Sezgisel optimizasyon teknikleri doğadan esinlenmiş olup, çözüm uzayı geniş ve büyük olan problemlerde, tüm çözüm uzayını taramadan sezgisel olarak çok kısa sürede en uygun değere veya bu değere yakın sonuç elde edebilen algoritmalarlardır. Sezgisel optimizasyon tekniklerinden biri olan Yerçekimi Arama Algoritması (YAA), son yıllarda mühendislik problemlerinin çözümünde oldukça ilgi görmektedir. Kütleler olarak adlandırılan ajanlar, Newton yerçekimi ve hareket kanunlarının simülasyonu ile en uygun çözümü bulmak üzere tanımlanmıştır. YAA üzerinde inceleme ve analiz çalışmalarından sonra, yerel minimuma takılma, grupsal davranıştan kopma ve hassas arama yapmama gibi sakıncaların olduğu tespit edilmiştir. Bu tez çalışmasında, her bir sakıncayı ortadan kaldıran veya azaltan üç yeni operatör geliştirilmiştir. Önerilen birinci operatörde, yerel minimuma takılma ve küresel uygun değerden uzak arama durumlarında, yerçekimi parametresi üzerinde kaotik sarsıntı oluşturularak ajanların hız ve dolayısı ile pozisyonlarında değişiklikler yapılmıştır. Geliştirilen ikinci operatör de, grupsal davranışın dışında veya uzakta kalmış ajanların hızlarını artırmak ve grup içerisine düşürmek için kurtulma hızının negatif yönde eklenmesi ile sağlanmıştır. Böylece, arama içerisinde sürü ve grupsal yaklaşımı mükemmelleştirme çalışması gerçekleştirilmiştir. Son olarak geliştirilen operatör de amaç, en iyi sonuç değerine sahip ajanın bir sonraki döngüde toplam kuvvetini bulurken en kötü kütleye sahip ajanları devreye sokarak toplam kuvveti, dolayısıyla hızının düşük olmasını sağlamaktır. Geliştirilen üç yeni operatör, 23 tek ve çok modlu karşılaştırma test fonksiyonlarına uygulanmıştır. Her bir fonksiyon 30'ar kez çalıştırılarak en iyi, ortalama ve ortanca değerleri alınmıştır. Elde edilen sonuçlar, Lipschitzian matematiksel yöntemi ve sezgisel optimizasyon algoritmalarından olan Genetik Algoritma (GA), Parçacık Sürü Optimizasyonu (PSO) ve YAA ile karşılaştırılmıştır. Her üç operatörün standart YAA ile her fonksiyon için yakınsama hızı, standart sapması ve grafiksel karşılaştırmalarına da yer verilmiştir. Ayrıca Wilcoxon işaretli sıra testi kullanılarak, standart YAA ve yeni operatörlerin iki bağımlı gruplar halindeki verilerin ortalamalarının karşılaştırılması yapılmıştır. Elde edilen sonuçların diğer yöntemlerden daha iyi sonuçlar verdiği görülmüştür.

Anahtar sözcükler: Optimizasyon, Yerçekimi arama algoritması, Kaotik, Kurtulma hızı

ABSTRACT

IMPROVING OF NEW OPERATORS FOR GRAVITATIONAL SEARCH

ALGORITHM

Ferzan KATIRCIOĞLU

Duzce University

Graduate School of Natural and Applied Sciences, Department of Electrical-Electronic
and Computer Engineer

Doctoral Thesis

Supervisor: Doç. Dr. Uğur GÜVENÇ

May 2016, 129 pages

Heuristic optimization techniques are inspired from nature and are algorithms that can obtain results at the most suitable value or close to this value in a very short time heuristically in problems with broad and large solution space without scanning the whole solution space. The Gravitational Search Algorithm (GSA) among one of the heuristic optimization techniques attracts great attention in the solution of engineering problems in recent years. The agents called masses were defined in order to find the most appropriate solution with the simulation of Newton's laws of gravity and motion. After the examination and analysis studies on the GSA algorithm, it was determined that there are drawbacks such as being stuck in a local minimum, break from group behaviour and not performing a sensitive search. In this thesis study, three new operators that eliminate or reduce each drawback were developed. In the first operator suggested, changes were made in the velocity and consequently positions of the agents by creating chaotic shake on the gravitational constant in cases of being stuck in a local minimum and searching away from the appropriate global value. That the escape velocity was added in the negative direction was ensured in order to increase the velocity of the agents that remain outside or far from the group behaviour and reduce it within the group in the second operator developed. Hence, the study on idealizing the flock and group approach was performed in the search. The aim of the last developed operator is to ensure that the total force and consequently velocity are low by activating the agents with the worst mass when finding the total force of an agent with the best result value in the next iteration. Three new operators developed were applied to 23 single and multiple modal Benchmark test functions. Each function was activated 30 times, and the best, average and median values were taken. The results obtained were compared with Lipschitzian mathematical method and GA, PSO and GSA, among heuristic optimization algorithms. The standard GSA of all three operators and convergence speed, standard deviation, and graphical comparisons were included for each function. Furthermore, the comparison of the averages of the data in two dependent groups of the standard GSA and new operators was performed using the Wilcoxon signed-rank test. It was seen that the results obtained yield better results than other methods.

Keywords: Optimization, Gravitational search algorithm, Chaotic, Escape velocity

EXTENDED ABSTRACT

IMPROVING OF NEW OPERATORS FOR GRAVITATIONAL SEARCH ALGORITHM

Ferzan KATIRCIOĞLU
Duzce University

Graduate School of Natural and Applied Sciences, Department of Electrical-Electronic
and Computer Engineer

Doctoral Thesis

Supervisor: Doç. Dr. Uğur GÜVENÇ

May 2016, 129 pages

1. INTRODUCTION:

Optimization means finding the best design among possible designs of a system. Mathematical and heuristic optimization techniques are used in the solution of optimization problems. Heuristic optimization techniques were developed inspiring from nature. Heuristic algorithms are algorithms that can obtain results with the most suitable value or close to this value in very short periods without scanning the whole solution space in problems with broad and large solution space. GSA, which is among heuristic optimization techniques, is preferred in the solution of complex engineering problems in recent years. GSA is based on Newton's laws of gravity and motion among the laws of physics. In Newton's gravity law, each part pulls another part with particular force. This gravitational force is directly proportional to the mass of the particles and inversely proportional to the distance between them. In the second rule of Newton's laws of motion, it is said that when a force is applied to a particle, its acceleration is subject to the force applied and mass of the particle. A series of agents called masses in GSA are defined in order to find the most appropriate solution with the simulation of Newton's laws of gravity and motion.

In this thesis study within the study method and plan, first the positive or negative behaviours shown by GSA parameters against test functions for different values were examined, and the disadvantages were revealed. After the analyses and results obtained had been interpreted, three new operators were suggested. The operators developed were applied to 23 single and multiple modal Benchmark test functions. The results were obtained compared with GA, PSO, and standard GSA, which are among

Lipschitzian mathematical method and heuristic optimization algorithms, the methods known in the literature. For each Benchmark function of the operators developed, the examination of convergence velocity, standard deviation, processing time and graphical analysis were performed comparatively with standard GSA. Furthermore, the Wilcoxon signed-rank test was also used in order to prove the strength of the development brought by the operators. The most important property of this test is that it considers the amount of the difference without regarding the sign between the differences.

2. MATERIAL AND METHODS:

The first operator is suggested a study based on the gravitational constant in the standard GSA. The gravitational constant is an important variable used in changing the places of the agents within the search space and determining the velocity of the agents. The value of the gravitational constant gradually decreases, the velocity of the agents also decreases and clustering occurs around heavy agents with slow motions due to the gravitational force of the heavy agents. This is called precise search condition, and it is ensured to occur in the last iterations. How to determine the situation of being stuck in a local minimum in the search process of the GSA algorithm is as important as the solution of the problem. For this, inferences can be made about the search process by comparing the best result value found for each iteration with the previous one. If the best result value does not change as the number of iterations increases especially at the initial and middle stages of the search, it is concluded that it gets stuck in the local lowest value or the search occurs in bad regions. After determining the problem, the velocities and positions of the agents are changed by creating a small chaotic shake on the gravitational constant in order to avoid the local minimum and make the search algorithm dynamic. If it is outside these conditions, the existing condition is maintained if the normal search is being realized. In the case of being stuck in the local minimum and when a bad search is performed, the velocity of the agents needed to get rid of this situation is ensured by creating a chaotic shake.

In the second operator of the thesis study, it was suggested that it could be ensured by negatively adding the escape speed above in order to increase the velocities of the agents that remained outside or far from the group behaviour or drop them in the group. When the total forces affecting the far agents were calculated, it was determined that their total forces were low due to the distance, and thus, their velocity would also be

low. In this case, they maintain their search in bad regions as their velocity is low, and their contribution to the optimization result decreases. Group behaviour motion, one of the main properties of heuristic optimization techniques. However, if the distance between the agents is large and the mass is small, the tendency to remain outside the group behaviour increases. The escape velocity in physics means the minimum velocity needed for an object in a three-dimensional space to escape from the mass gravitational field affecting itself. An object with escape velocity does not fall back to the object that is trying to pull it or does not move on an orbit around that object. In the case of escaping from a single-centred, simple gravitational field, escape velocity is the value when the kinetic energy of an object is equal to the mass gravitational potential energy. One of the most important differences that differentiate from classical GSA is that the velocity and position updates are performed in different process divisions. The relevant agent's being far is questioned before velocity updates. If it is not included in the group of far agents, velocity update is performed as classical GSA. If it is included in the far agent group, the escape velocity is added to the classical velocity. The main flow is maintained by entering the agent group divided into two groups in the position update together. Thus, the study on perfecting the flock and group approach is ensured in the search.

In the last operator suggested, it was ensured that the total force, and thus, the velocity were low while finding the total force of the agent with the best result value in the next iteration by activating the agents with the worst mass affecting this agent. The standard GSA optimization algorithm is based on the gravitational force between the masses and the object with the highest mass moves slower. Furthermore, the direction of the gravitational force is towards the big object. The objects with mass in Newton's law of gravitation were turned into agents doing a search in the GSA algorithm. While all of the agents are included in the force calculation in the first iteration of the search, the agents chosen among agents with big masses are included in the force calculation in the next iterations. This means that the agent with a big mass accelerates as others, has a great velocity and changes position to a great extent. Nevertheless, it was supposed to have great mass and move very slowly as a result of having the best result value. With the help of this operator, the agent with the best value will increase the convergence property to the best result value property by changing position at very low rates. The velocity calculation peculiar to the best agent was performed by means of reducing its

velocity by activating agents with low mass when finding the total force. The part of the thesis so far was performed in Matlab engineering software.

3. RESULTS AND DISCUSSIONS:

At the stage of examining standard GSA parameters, the most appropriate parameter intervals for G_0 value were determined to be between 100 and 1000, for α parameter between 10 and 20, in the agent number assessment between 30 and 100, the number of iterations was determined to be around 500 if it was desired to achieve the result in a short time, and that the number of iterations should be 1000 in applications with no significance for the user. Furthermore, good results and high convergence velocity were obtained in the standard GSA algorithm in results such as giving the total number of agents of *son_ajan* percentage values small values such as 10% using the Kbest approach.

The first operator suggested was named the Gravitational Search Algorithm with Chaotic Shake. The positions of the agents were changed to a greater extent by creating a chaotic shake on the gravity constant in the cases of being stuck in the local minimum and being in the bad search region. Circke, Logistic, Singer and Sinusoidal chaotic functions used in the operator allowed for creating a shake between the iterations 50 and 250, and it was not allowed in the last 60% portion. Among chaotic functions, the results of Singer chaotic function are striking, and thus, this function was preferred in all experimental studies. Good results and high convergence velocity were obtained in the comparison tables and graphical demonstrations for each of single and multiple modal test functions of operator. While the null hypothesis was denied in a total of 11 functions according to the Wilcoxon signed-rank test results of the new operator, the null hypothesis was realized in other functions. That the standard deviation value for test functions in the operator was found low showed that this algorithm was good in the data among them. Chaotic shake operator has high performance in the sense of not being stuck in the local minimum and being found easily when compared to the GSA algorithm.

The second operator suggested was defined as the Escape Velocity Gravitational Search Algorithm . The operator was first applied to 23 Benchmark test functions and compared to heuristic and mathematical algorithms that are widely used in the literature.

It yielded good results with great differences according to this comparison. In the graphical examination, it was visually proved that the convergence velocity in functions is high. The standard deviation value in test functions was found low when compared to the standard GSA values; it shows that the rate of approaching the best results is high in each operation. Thus, the study on perfecting the flock and group approach was ensured in the search, and it was seen that this reflects positively on the search.

The last operator performed in this thesis study was named the Best Exceeding Gravitational Search Algorithm. Upon examining the data, it was determined that the improvement was at great rates. According to the graphical examination, it was shown that the operator went to the result quite early. The operator results of the Wilcoxon signed-rank test that performs the one-to-one and ranked comparison of the data received rejected the null hypothesis for 11 functions. Its standard deviation is smaller, and the results are closer to each other. Furthermore, the rate of converging to the global minimum value is better in both operations compared to GSA. The operation time for a single iteration is great when compared to GSA, but it is shorter when compared to Chaotic shake operator. Consequently, the sensitive search was performed by slowing down the velocity in these operator and its performance is strong when compared to the standard GSA.

4. CONCLUSION AND OUTLOOK:

The basic structure that must be used in the optimization-based applications of engineering is the property of velocity and convergence to the actual value. Three new operators realized for GSA will make all engineering applications much more flexible and practical. With this flexible and practical optimization algorithm structure that will be found as a result of this thesis, its preferring and development in the engineering applications in our country will be ensured. It gained three important properties as the GSA with a chaotic shake that does not get stuck in the local minimum, the GSA with the escape velocity that ensures the group behaviour motion very well, and the GSA, the sensitive search property of which is increased. They can represent real life problems more accurately due to these properties. The results obtained in this thesis will be gradually published in the magazines within the scope of SCI on the subject and contribute significantly to SCI index scientific publications produced by our country.

1. GİRİŞ

1.1. AMAÇ VE KAPSAM

"Optimum" Yunanca bir kelime olup, elde bulunan olanaklar ölçüsünde nihai sonuç manasına gelir. Bu kelimedenden türetilerek benzer şekilde optimizasyon ise çeşitli problemleri çözmek amacıyla karar değişkenleri ve kısıtlar altında, en uygun çözümü arama yöntemlerine denir [1]. Bu problemlerde en uygun çözümün bulunması problemin en büyük veya en küçük değerinin elde edilmesi anlamına gelir. Literatürde kullanılan optimizasyon metotlarını genel olarak iki gruba ayırılır. Bunlardan birincisi kesin çözümü üretebilen deterministik teknikler, ikincisi optimal çözümü üreten sezgisel tekniklerdir. "Heruistic (Sezgisel, Buluş, Keşif)" kelimesi de Yunanca bir kelime olup, "Bilmek için", "Bulmak için", "Keşfetmek için" veya "Bir araştırmaya rehberlik etmek için" anlamlarında kullanılır. Optimizasyon teknikleri içerisinde yer alan sezgisel algoritmalar, herhangi bir amacı gerçekleştirmek veya hedefe varmak için doğal yaşamdaki davranışlardan esinlenen algoritmalar ve anlaşılabilirlik açısından çok basit yapıya sahiptirler [2]. Gerçek yaşam problemlerinin çoğunda problemin çözüm uzayı tüm çözümlerin değerlendirilemeyeceği kadar büyük olur. Bunun içinde kabul edilebilir çözümlerin değerlendirilerek iyi bir çözümün bulunması gerekmektedir. Bazı çözümlerin neye göre ve nasıl seçileceği kullanılan sezgisel tekniğe göre değişir. Maalesef değerlendirmeye dahil olan çözümlerin içerisinde optimal çözümün yer alması garanti edilememektedir. Bu sebeple de sezgisel tekniklerin bir optimizasyon problemine önerdiği çözüm, optimal değil yakın çözüm olarak algılanmalıdır [3].

Bu tez çalışmasında, sezgisel bir teknik olan ve Newton yerçekimi ve hareket yasaları esas alan YAA üzerinde çalışılmıştır. Yerçekimi Kanunu göre, her ajan diğer ajanları çeker ve iki ajan arasındaki yerçekimi kuvveti ajanların kütlelerinin doğrudan çarpımı ile doğru, ikisi arasındaki mesafe ile ters orantılıdır. Hareket Kanununun bu algorithmada güncellenmesi ile, herhangi bir ajanın geçerli hızı, hızda sapma ve onun önceki hız değerinin toplam miktarına eşittir [4]. YAA genel özellikleri arasında, bir ajanın kuvveti komşularını da etkilenmesinden dolayı, çevresiyle etkileşim halindedir ve ağır bir ajan etkili büyük bir çekim yarıçapına ve çekim yoğunluğuna sahiptir. Algoritma içerisinde kullanılan yerçekimi parametresi, araştırmanın doğruluğunu düzenler, zamanla birlikte azaltılma özelliğine sahiptir. YAA bellek-azaltıcı bir algoritmadır, ancak bellek

algoritmalarından olan Tabu Arama ve Karınca Koloni Algoritmaları gibi etkili çalışma davranışı gösterir.

Ajanların YAA içerisinde, çözüm uzayı içerisinde geniş bölgeyi arama özelliği olan küresel keşif ve yerel minimuma takılma sakıncasının uygun kontrolü ile etkili uygun çözüm bulmak çok önemlidir. Bu uygun kontrolü sağlanması ve YAA'nın performansının artırılması YAA parametrelerinin doğru kullanımına doğrudan bağlıdır. Yerçekimi bu parametrelerden biri olup, yer çekim kuvvetinin etkisini kontrol eden bir değişkendir. Ajanların hızlarına ve pozisyonlarına etki eder, yani arama kabiliyeti büyük oranda yerçekimi parametresine bağlıdır. YAA bir çok optimizasyon problemi için hızlı yakınsamasına rağmen, küresel uygun sonuca ulaşma da zorluklar yaşadığı gözlemlenmiştir. Başka bir ifade ile YAA uygulamalarında arama esnasında yerel minimuma takılma durumunun çok karşılaşılmıştır. Yapılan eleştirilerden ve arama kabiliyetindeki eksikliğinden dolayı, klasik YAA içerisindeki yerçekimi parametresi yerel minimuma takılma sakıncasını beraberinde getirmektedir. Yerçekimi parametresinin tekrar gözden geçirilmesi ve yerel minimum sakıncasını kaldırabilecek değişimlerin yapılması zorunluluk haline gelmiştir.

YAA, kütleyle sahip nesnelere birbirine doğru hızlanma eğilimini esas alır. Ajanların hız hesaplanmasında, diğer ajanların kendisine etkisi sonucu oluşan toplam kuvvet önemli bir değişkendir. Uzakta yer alan ajanların mesafeden dolayı toplam kuvvetlerinin düşük olduğu ve dolayısıyla hızlarının da düşük olacağı tespit edilmiştir. Bu durumda hızlarının yavaş olmasından dolayı ajanların birbirine doğru hızlanma eğilim özelliği bozulacak ve kötü bölgelerdeki aramalarına devam edeceklerdir. Bu durum, grupsal davranışın uzağında kalan ajanların, optimizasyon sonucuna katkılarının azalması anlamına gelir. Sezgisel tekniklerin ortak özelliklerinden biri de grupsal davranış özelliğini geliştirerek yakınsama hızını artırmaktır. Belirtilen sakıncayı ortadan kaldırarak grupsal davranış özelliğini ve bunun sonucu olarak yakınsama hızını artırmak üzerinde çalışılması gereken başka bir çalışma konusudur.

Yerçekimi Kanununu esas alan optimizasyon algoritmasında ajanlar bir nesne gibi düşünülmüş ve onların performansları kütleleri ile ölçeklendirilmiştir. Bütün bu nesnelere yerçekimi kuvveti ile birbirlerini çekerler ve bu kuvvet daha ağır kütleli nesnelere doğru bütün nesnelere hareketine sebep olur. Bu yüzden iyi çözümlere karşılık gelen ağır ajanlar hafif olanlara göre daha yavaş hareket etmesi gerekir. Klasik

YAA yaklaşımına göre ise, kendisine etki eden ajanlar en iyi kütleyle sahip ajanlar arasından seçilmektedir. Buda bu çok büyük kütleli ajanın diğerleri gibi ivmesi, hızının büyük olması ve büyük oranda pozisyon değiştirmesi anlamına gelir. Bunun sonucu olarak da ağır ajanların etkisinden dolayı, klasik algoritmanın hassas arama kabiliyeti düşüktür ve hassas arama özelliğinin artırılması gerekir.

Bugüne kadar yapılan YAA uygulamalarında yerel minimuma takılma, grupsal davranış bozukluğu ve hassas arama kabiliyet yetersizliğinden dolayı mühendislik bakış açısı ile bu algoritmalara uyumluluk ve sağlamlık eklenmesi arzu edilir hale gelmiştir. Özel problem içeren pratik uygulamalarda bu tür düzenleme gereksinimleri önemlidir ve optimizasyon algoritmasına dinamik bir davranış katacaktır. Optimizasyon algoritmasında düzenleme ve gelişme eklemek için iki önemli bağıntıyı anlamış olmak gerekir. Birincisi algoritmanın davranışı ve parametreleri arasındaki nitel ve nicel ilişkinin analizinin iyi yapılması gerekir. Diğeri ise arama ile algoritmanın davranışı arasındaki nitel ve nicel ilişkinin analizidir. Algoritmanın uyarlanması için bu analizlerin sonuçlarının çok dikkatli tartılması gerekir ve böylece ideal algoritma davranışı ortaya konulabilir.

Bu bağlamda, yapılan çalışma yöntemi ve planı içerisinde, ilk olarak YAA'nın farklı parametreler için test fonksiyonlarına karşı gösterdiği olumlu veya olumsuz davranışlar incelenmiş, dezavantajları ortaya konmuştur. Bu parametrelerin etkileri gözlemlenerek ve aralarındaki ilişkiler ortaya konmuştur. Buradan elde edilen sonuçlar yukarıda bahsedilen yerel minimuma takılma, grupsal davranış uzaklaşarak yakınsama hızının azalması ve hassas arama özelliğinin kötü olması sakıncalarını ortadan kaldırmak için önemli fikirler verecektir.

Bir sonraki adım, YAA optimizasyon algoritması üzerinde yapılan çalışmalar incelenmiş, YAA'ya getirmiş oldukları değişimler uygulanarak analiz edilmiştir. Bu durum, oluşturacağımız önerilerin özgünlüğü ve yeni fikirler verme açısından temel bir yaklaşım oluşturmuştur.

Daha sonraki çalışma adımında, sakıncalar ortadan kaldırılmış, dinamik ve performansı güçlü operatörler tasarlama aşaması gerçekleştirilmiştir. Özellikle yerel minimumdan kurtulma, grupsal davranış sağlama ve hassas arama üzerinde ayrı operatörler önerilerek, YAA'nın farklı problemlere karşı farklı çözümler sunmak, kullanıcıya her

durumda güçlü performanslı optimizasyon algoritması önermek, amaçlanmıştır. Oluşturulan üç yeni YAA operatörü 23 tek ve çok modlu test fonksiyonlarına ayrı uygulanacak ve ayrıca sonuçlar bilinen Matematiksel Yöntem, GA, PSO ve standart YAA ile karşılaştırılması yapılmıştır. Beklenen sonuca göre, yeni YAA çalışması optimizasyon sürecine dinamik bir davranış katarak, en uygun değeri bulma amacını gerçekleştirmiştir.

Bu çalışma dört bölümden oluşmaktadır. Çalışmanın birinci kısmı olan “Giriş” bölümünde, Optimizasyon, Sezgisel Optimizasyon Teknikleri ve YAA'nın mühendislik problemlerinin çözümünde yeri ve önemi ile çalışmanın amaçları ortaya konularak, çalışma alanı içerisindeki temel kavramlar açıklanmıştır. Ayrıca, çalışma ile ilgili ulusal ve uluslar arası ölçekte literatür incelemesine yer verilmiştir. Çalışmanın ikinci kısmını “Materyal ve Yöntem” bölümü oluşturmaktadır. Bu bölümde, öncelikle YAA tanıtımı, daha sonra bu algoritmanın performansını etkileyen parametreler detaylı bir şekilde incelendikten sonra algoritmaya etkileri üzerine tespitler yapılmıştır. Değerlendirme sonucu ortaya çıkan sonuçlar ışığında Kaotik Sarsıntılı Yerçekimi Arama Algoritması (KSYAA), Kurtulma Hızlı Yerçekimi Arama Algoritması (KHYAA) ve En İyi Ajan Yerçekimi Arama Algoritması (EİAYAA) olmak üzere üç yeni operatör önerisi yapılmıştır. Üçüncü kısım olan “Bulgular ve Tartışma” bölümünde ise, önerilen üç yeni operatörün performansı konu ile ilgili daha önce yapılan çalışmalarla karşılaştırılmıştır. Ayrıca bu üç yeni operatör üzerinde yapılan ölçümler ve analizler neticesinde elde edilen veriler değerlendirildi. Çalışmanın son kısmını ise “Sonuç ve Öneriler” bölümü oluşturmaktadır. Bu bölümde, çalışmadan elde edilen önemli sonuçlar ve önerilen operatörlerin performans özelliklerine yer verilmiştir. Bu bilgiler ışığında bazı öneriler sunulmuştur.

1.2. GENEL BİLGİLER

Çeşitli problemlerin karar değişkenlerine bağlı olarak, en uygun çözümünü arama yöntemlerine optimizasyon denir [1]. Bu problemlerde en uygun çözümün bulunması demek, aslında kastedilen herhangi bir kıstasın maksimize veya minimize edilmesi anlamına gelir. Doğal olarak bahsi geçen kıstas bir fonksiyonla temsil edilir ve amaç fonksiyonu denilir. Minimize veya maksimize edilen niceliği gösterir. Amaç fonksiyonu tüm kısıtları sağlayacak şekilde bir çözüm bulmalıdır.

Değişkenler veya bilinmeyenler kümesindeki elemanlar amaç fonksiyonu etkileyen değerlerden oluşur. Denklem (1.1) karar değişkenleri x vektörü ile temsil edilecek olursa, amaç fonksiyonu Denklem (1.2) deki gibi ifade edilir [2].

$$x = (x_1, x_2, \dots, x_i, \dots, x_n) \quad (1.1)$$

$$f(x) = f(x_1, x_2, \dots, x_n) \quad (1.2)$$

x bilinmeyenler kümesinin alabileceği değerleri kısıtlayan kümeye kısıtlar kümesi denir. Problemlerin çoğu her değişken için belirli bir sınır kısıtlayıcısına sahiptir. Denklem (1.3) deki gibi tanımlanan p tane eşitlik sınırlaması ve yine Denklem (1.4)'deki gibi tanımlanan m tane eşitsizlik sınırlaması olduğu varsayılırsa,

$$h_j(x) = h_j(x_1, x_2, \dots, x_n) = 0 \quad 1 \leq j \leq p \quad (1.3)$$

$$g_i(x) = g_i(x_1, x_2, \dots, x_n) \leq 0 \quad 1 \leq i \leq m \quad (1.4)$$

buna göre minimumu araştırılan bir optimizasyon problemi Denklem (1.5) deki gibi ifade edilir [5].

$$\min f(x) \leq f(x) \quad (1.5)$$

İnsanoğlunun yapmak istediği bir işin en uygun yolun seçilmesi ile gerçekleştirilme fikri insanlık tarihi kadar eskidir. Antik Mısır döneminde, Mısırlılar Nil Nehri'nin bahar dönemlerindeki taşmalarından dolayı nehrin kıyısından uzaklaşıp, sular çekildiğinde geri dönüyorlardı. Çekilme işlemi çok kısa sürede ve zamanında yapılamamasından dolayı, Mısırlılar en iyi çekilme zamanını hesaplamaya çalışmışlardır. Bu çabalar, ölçme ve karar verme aracı olarak düzlem geometrisinin temellerinin oluşmasına sağlamıştır [6]. Yunan matematikçi ve adı geometriyle en çok özdeşleşen kişi Öklid geometrinin başlangıcından kendi zamanına kadar bilinenleri 'Öğeler' adını verdiği kitaplarında toplamıştır. M.Ö. 300 yıllarında Öklid iki nokta arasındaki minimum mesafeyi kenar uzunlukları verilen dikdörtgenler arasında en büyük alana sahip kareyi bularak kanıtladı. Buna göre iki boyutlu düzlemde iki nokta arasındaki mesafe basitçe

iki noktanın x ve y koordinatlarının ayrı farklarının hipotenüs'üne eşittir [7]. Yine Öklid ile yakın dönemlerde yaşamış Heron ise Catoprica içinde ışığın aynadan yansması ile iki nokta arasındaki en kısa yolun bulunduğunu kanıtladı [8].

14. yy. da başlayan Rönesans kültürel hareketi, kentlerin birleşip ulus devletleri oluşturması, coğrafi keşifler gibi nedenlerden dolayı sanatın ve bilimin konusu insan odaklı olmuştur. Bu gelişmelerden biri olarak 1615 yılında J. Kepler bir şarap varilinin uygun boyutlarını çözdü. Ayrıca Kepler, yeni bir eş aramaya başladığı zaman, dinamik programlamanın klasik uygulaması olan sekreter problemini en erken versiyonunu formüleştiren çalışmasını gerçekleştirdi [9]. 1660 yılında I. Newton ve 1670 yılında G.W. von Leibniz CoV olarak adlandırılan varyasyon hesaplamalarının esaslarını oluşturarak matematiksel analizin temelini atmış oldular. Kalkülüs'ün (Calculus) 17. yy. da geliştirilmesi optimizasyon teorisinin gelişimi açısından önemli bir dönüm noktasıdır. Böylece matematiksel bir fonksiyonun ve bağımsız değişkenlerinin maksimum veya minimum cinsinden optimal koşullarının elde edilmesi sağlanmış oldu [10]. Konveks fonksiyonlar Hadamard (1883), Hölder (1889) ve Stolz (1893)'ün çalışmaları içerisinde fikir olarak görülse de 1905 yılında Jensen konveks fonksiyonların tanıtımını yaparak ilk kez bu ad ile anılmaya başlanıldığı söylenebilir. Takip eden yıllarda 1917 de Hancock tarafından "Minimum ve Maksimum Teorisi" adı altında ilk optimizasyon metin kitabını yayınlandı [11].

İkinci dünya savaşından sonra optimizasyon yöneylem araştırması gelişti. 1944 de J. von Neuman ve O. Morgenstern dinamik programlama fikrini kullanarak ardışık karar problemlerini çözdüler [12]. Lineer programların çözümü için Simplex yöntem 1947'de G.B. Dantzig tarafından geliştirildi [13]. Bu, optimizasyon dünyasında önemli bir gelişme olup, devrim niteliğindedir. 1951 yılında H. Markowitz kuadratik optimizasyon esaslı portföy teorisini sundu ve ekonomi alanında Nobel anma ödülü aldı [14]. 1957 yılında R. Bellman en iyilik prensibini sundu [15].

1992'de J.H. Holland tarafından bir sezgisel optimizasyon tekniği olarak kabul edilen genetik algoritma geliştirildi [16]. Bu tarihten itibaren sezgisel teknikler çağdaş optimizasyon dünyasında ve her türlü karmaşık problemlerde her geçen gün artan bir ivmeyle önemli katkılar sunmaya devam etmektedir.

1.2.1. Optimizasyon Problemlerinin Sınıflandırılması

Optimizasyon problemleri değişkenlerin sayısı, değişkenlerin türü, sınırlar ve doğrusallık gibi bir kaç karakteristik özelliğe göre sınıflandırılabilir. Amaç fonksiyonunu etkileyen değişkenlerin sayısı bir ise tek parametreliden fazla ise çok parametrelili optimizasyon problemi adını alır. Bazı problemler içerisinde, birden fazla en küçük yapılacak fonksiyon bulunabilir. Bu durumda bu problemler birden fazla amaç fonksiyonuna sahip olurlar ve çok amaçlı optimizasyon problemi olarak adlandırılırlar.

Doğrusal problemlerin amaç fonksiyonları doğrusal değişkenlere sahiptir. Kuadratik problemler kuadratik fonksiyonlar kullanır. Diğer doğrusal olmayan amaç fonksiyonları kullanıldığında problem doğrusal olmayan optimizasyon problemi halini alır.

Amaç fonksiyonunun, x ile ilgili herhangi bir sınırlama olmaksızın minimize veya maksimize edilmesi gerekiyor ise sınırlamasız optimizasyon, x ile ilgili sınırlamanın veya sınırlamaların bulunduğu optimizasyon problemi ise sınırlamalı optimizasyon problemi olarak isimlendirilir.

Diğer bir gruplandırma ise sürekli ve ayrık optimizasyon şeklindedir. Ayrık niceliklerin optimal olarak düzenlenmesi, gruplanması ve sıraya konulması veya seçilmesi problemi ayrık optimizasyon problemi olarak adlandırılır. Tasarım değişkenlerinin veya parametrelerinin alacağı değerler sürekli değerler ise bu tür problemlere sürekli optimizasyon problemi denir [5].

1.2.2. Optimizasyon Metotları

Herhangi bir optimizasyon probleminin kesin veya optimal çözümünü bulmaya çalışan yöntemlere optimizasyon yöntemleri denir. Optimizasyon metotlarını genel olarak iki gruba ayırmak yerinde olur. Bunlardan birincisi kesin çözümü üretebilen determinist teknikler, ikincisi optimal çözümü üreten sezgisel tekniklerdir [2]. İki grubunda birbirlerine karşı avantaj ve dezavantajları bulunmaktadır. Örneğin determinist teknikler belirli bir probleme çözüm sunabilirken, sezgisel teknikler uyarılma ile birçok optimizasyon problemine uygulanabilmektedir. Ayrıca klasik teknikler için en iyi değer bulunmaya çalışıldığı amaç fonksiyonunun matematiksel olarak modellenmesi gerekmektedir. Halbuki gerçek dünya problemlerinin çoğunda bu mümkün değildir ve bu tür problemlere sezgisel teknikler cevap vermektedir. Bu yüzden zeki ve etkili sezgisel yaklaşımların, zor problemlerin çözümünde kullanılması üzerine çeşitli

yaklaşımlar geliştirilmiştir [17].

1.2.3. Sezgisel Optimizasyon Teknikleri

Bir optimizasyon probleminin zorluk derecesine göre çözüm uzayı ve çözümün bulunma süresi büyüyebilmektedir. Çözüm uzayındaki tüm elemanların denenmesi ile en iyiyi bulma çalışmasına sayım tekniği denir. Kullanılan bu teknikle mutlaka en iyi çözüme ulaşılır, fakat her elemanın denenmesi çözüm uzayı çok büyük olduğundan süre uzar. Bu dezavantajı ortadan kaldırmak için ortaya atılmış olan sezgisel algoritmalar, belirli bir çözüm uzayında rastgele seçimlerle en iyiyi bulmaya çalışırlar. Optimizasyon teknikleri içerisinde yer alan sezgisel algoritmalar, herhangi bir amacı gerçekleştirmek veya hedefe varmak için doğal yaşamdaki davranışlardan esinlenen algoritmalarlardır [18].

Sezgisel algoritmalara gerek duyulmasının sebepleri maddeler halinde listelenmiştir.

- ✓ Optimizasyon problemi, kesin çözümü bulma işleminin tanımlanamadığı bir yapıya sahiptir ve olumlu sonuçlar verir.
- ✓ Anlaşılabilirlik açısından çok basit yapıya sahiptir.
- ✓ Öğrenme amaçlı ve kesin çözümü bulma işleminin bir parçası olarak kullanılabilir.
- ✓ Gerçek dünya problemlerinin zor tarafları matematik formülleri ile yapılan tanımlamalarda genellikle ihmal edilir. Sezgisel algoritmalarda gerçek dünya problemlerinin model parametreleri her yönü ile belirlenebilir [5].

Gerçek yaşam problemlerinin çoğunda problemin çözüm uzayı sonsuz ve tüm çözümlerin değerlendirilemeyeceği kadar büyük olur. Bunun içinde kabul edilebilir çözümlerin değerlendirilerek iyi bir çözümün bulunması gerekmektedir. Bazı çözümlerin neye göre ve nasıl seçileceği kullanılan sezgisel tekniğe göre değişir. Maalesef değerlendirmeye dahil olan çözümlerin içerisinde optimal çözümün yer alması garanti edilememektedir. Bu sebeple de sezgisel tekniklerin bir optimizasyon problemine önerdiği çözüm, optimal değil iyi çözüm olarak algılanmalıdır [2].

Çok sık kullanılan ve bilinen sezgisel optimizasyon tekniklerinden olan Tabu araştırma algoritması (TAA) kavramını Glover ortaya atmış olup, oldukça yeni ve zor problemlerin çözümünde kullanılan yönlendirilmiş bir optimizasyon algoritmasıdır [19]. Çok modlu problemler için küresel optimum çözümleri bulma yeteneğine sahiptir.

Temel yaklaşım, bir sonraki döngüde tekrarın yasaklanması ve cezalandırılmasıdır. Tabu araştırma algoritmasında daha önceden denenmiş çözümler, tekrar denenmeyerek, araştırma yönünün, denenmeyen alanlara yönlendirilmesi amaçlanmıştır. Eğer bu komşulardan daha önce denenmişler varsa, bunların yerine başka komşular üretilerek araştırmanın hızlandırılması sağlanır ve yerel optimumda kısır döngüye girilmesi engellenir [20].

Benzetimli Tavlama (BT) algoritması, Kirkpatrick tarafından eniyileme problemlerinin çözümü için geliştirilmiş bir yerel arama algoritmasıdır. Bir metalin doğasında olan özellikler kullanılmış olup, metalin soğuyarak veya donarak minimum enerjili kristal yapısına dönüşmesi minimumun araştırılması arasındaki benzerlikten faydalanılmıştır. Yerel minimumdan kurtulabilme yeteneği özelliği ile diğer sezgisel teknikler arasında dikkat çekicidir [21].

Genetik optimizasyon algoritmaları daha iyi ve daha kuvvetli olan bireylere kazanma şansının verildiği, biyolojik olaylara benzetilerek geliştirilmiş, paralel ve küresel bir arama tekniğidir. Çözüm için "iyi"nin ne olduğunu belirleyen bir uygunluk fonksiyonu ve yeni çözümler üretmek için yeniden kopyalama, değiştirme gibi operatörleri kullanır. Genetik algoritma bir grup çözümle uğraşır ve bu sayede çok sayıda çözümün içinden iyileri seçilip kötülerini elenebilir [22].

PSO tekniği ilk olarak kuş ve balık sürülerinin hareketlerinden esinlenerek doğrusal olmayan nümerik problemlere optimal sonuçlar bulmak için 1995–1996 yıllarında sosyolog-psikolog James Kennedy ve elektrik mühendisi Russel Eberhart tarafından ortaya atılmıştır. PSO popülasyon tabanlı olasılıksak bir optimizasyon yöntemi olup çok parametrelili ve çok değişkenli optimizasyon problemlerine çözümler üretmek için kullanılmaktadır [23].

Karınca koloni optimizasyonu Dorigo ve arkadaşları tarafından yapılmıştır. Yaptıkları çalışmada kendi sistemlerine "karınca sistemi", elde ettikleri algoritmaya ise "karınca algoritması" adını vermişlerdir. Karınca çevre şartlarına göre besin kaynağı ile evi arasında gidebileceği yolları belirlemektedir. Belirlenen yollardan birinden ilk geçen keşif ve araştırma yapan karınca yola feromon adında bir koku bırakmaktadır. Eğer yol kısa ise bu koku daha yoğun olmaktadır [24].

Son dönemlerde önerilen ve kullanımı yeni algoritmalarından biri olan Ateş böceği sürü

optimizasyonu, K. N. Krishnanand ve D. Ghose tarafından 2005 yılında geliştirilmiştir. Çok modlu fonksiyonları optimize etmek için önerilen sürü zekâsı tabanlı bir algoritmadır. Bu yöntemi kullanmanın temel amacı tüm yerel maksimumları yakalamayı sağlamaktır [25]. Kedi Sürüsü Optimizasyonu ise kedigillerin hareketlerinin incelenmesiyle ortaya çıkarılmıştır. Bu optimizasyondaki matematiksel modeller kedilerin hareketlerinin çözümlenmesiyle meydana gelmiştir [26]. Bir başka çalışma olan Bakteriyel Besin Arama Algoritması, E. coli bakterisinin beslenme davranışından esinlenerek karmaşık mühendislik problemlerini çözmek için geliştirilmiş bir hesaplama tekniğidir. Bakteriler sınırlı algı ve hareket kabiliyetlerini kullanarak optimum düzeyde enerji harcıyıp beslenme faaliyetlerini gerçekleştirmeleri gerekmektedir [27]. Yapay balık sürüsü optimizasyonunda diğerlerine benzer şekilde, yiyecek aramada balık sürüsünün sosyal davranışlarının benzetimi esas alınmıştır. Küresel optimuma ulaşmak için balık bireyinin yerel aramasıyla toplanma ve izleme gibi balık davranışlarını taklit etmeyi amaçlar. Bir yapay balık kendi faaliyetleri ve arkadaşlarının faaliyetleri yolu ile çevresini etkileyecektir [28]. Ateş böceği algoritması, Dr. Xin-She Yang tarafından geliştirilmiş olup, ateşböceklerinin sosyal davranışlarını baz alır. Ateşböceği algoritmasının en önemli özelliği, verilen bir optimizasyon probleminin amaç fonksiyonu, ateşböceği sürüsüne parlak ve daha çekici yerlere gitmede yardım eden yanıp sönen ışık ya da ışık şiddeti ile ilişkilendirilmesidir [29]. Yeni önerilen algoritmalarından olan Kurt kolonisi algoritmasında kurtların sıkı bir organize sisteme sahip olmasından esinlenilerek geliştirilmiştir. Kurtlar görevleri diğerleriyle bölüşmektedirler ve avlandıkları zaman tutarlı adımlar atmaktadırlar. Az miktarda yapay kurt aktif olduğu av aralığında aramaya atanmaktadır. Arama kurtları avı keşfettiği zaman, avın konumunu diğer kurtlara ulumayla bildirmektedir. Kurt kolonisinin atanma kuralı, yiyeceğin ilk olarak güçlü kurda atanması ve daha sonra zayıf olana atanmasıdır [30].

1.3. YERÇEKİMİ ARAMA ALGORİTMASI LİTERATÜR ÖZETİ

Optimizasyon temelli mühendislik uygulamalarında, optimizasyon tekniğinden beklenen en önemli özellikler basitlik, kolay uygulanabilirlik, hız ve gerçek değere yakınlık olarak belirtilebilir. YAA optimizasyon tekniği 2009 yılında yayınlanmasından sonra gerçek yaşam problemlerinin çözümlerinde basit ve kolay uygulanabilir özelliğinden dolayı çok tercih edilmiştir. Bundan dolayı da geniş bir uygulama alanları

içerisinde yer almıştır.

Mühendislik uygulama alanlarından biri olan şev stabilize modeli, gerilme ve su seviyesi gibi bir dizi başlangıç koşuluna maruz kalan kaya kütesinin mekanik tepkisini sınır koşullarını ve şev kazısı gibi sonradan oluşan değişiklikleri değerlendirme amaçlı bilgisayar programıdır [31]. Küresel araştırma yeteneğini kontrol etmek için maksimum hız kısıtı uyarlamalı olarak kullanılmıştır. Elde edilen uyarlanabilir Modifiye Yerçekimi Arama Algoritmasında olasılık ve determinist şev stabilize analizleri içinde minimum güvenilirlik endeksi ve güvenlik minimum faktörünün araştırılması işleminde kullanılmıştır [32].

Bilgisayar dünyasında geliştirilen teknikler sayesinde moleküler biyoloji bilimi kapsamlı gelişmelere tanık olmaktadır. Verilerin saklanması, düzenlenmesi, birleştirilmesi, kataloglanması ve kolayca erişilmesi ile ilgilenen bilim dalına biyoinformatik denir [33]. Biyoinformatik alanında birçok organizmayı barındıran küçük moleküllerden karmaşık sistemlere kadar inceleme yapan problemlerin çoğu hesaplama teknikleri kullanılarak ele alınmaktadır. Deoksiribonükleik asit dizilerinin kümesi içinde yer alan motifler önemli dizi analizlerinden biri olup Motif Keşfetme Problemi (MKP) olarak bilinmektedir. Bu çalışma içerisinde MKP çözmek için hesaplanabilir sürü zekası önerilmiştir. Bireyler ile ilişkilendirilmiş yol vektörlerine uyan hücreler pozisyon güncellemede kullanılmıştır [34].

Sınıflandırma birçok bilim dalında kullanılan bir karar verme ve ayrıştırma işlemidir. Denetimli sınıflandırma teknikleri Bayesian sınıflandırıcısı, k yakın komşu, sinir ağı ve kural bazlı sınıflandırıcılar gibi çok çeşitlere ayırmak mümkündür. Bu sınıflandırıcılar bilinmeyen desenlerin özellik uzayından ayrılmış bölgelere ayırmak için zorunlu büyük düzlemler kararını tahmin ederler. Bu çalışma içerisinde, bir önceki bilgiye ihtiyaç duymaksızın fonksiyon tahmin kararı için yapay zeka tekniği YAA kullanılmıştır. Zeki YAA esaslı sınıflandırma tekniği olarak tanıtılan ZYAA-sınıf da, YAA'nın parametrelerini güncellemek için bulanık sistem tasarımı kullanılmış, özellik uzayından tahmini algoritmalarına karar veren fonksiyonu inşa etme işlemine uygulanmıştır [35].

Bir başka uygulama alanı içerisinde görüntü, konum ve renk özelliklerine sahip özellik uzayı içerisinde haritalanmıştır. YAA gezinme stratejisi kullanılarak kütleler benzer bölgeleri bulmak için özellik uzayı içerisinde gezinirler ve birleşme çalışması

kullanılarak bölge büyüme tarafından görüntü bölgelere ayrılır. Başlangıç da görüntü içerisindeki her bir piksel birim kütleyle sahip bir parçacık şeklinde atanır. Seyahat etme operatörü ile nesnelere diğer nesnelere yerçekimi kuvveti altında hareket eder. Ajanların sayısı birleşme çalışması ile her döngü de azaltılır. Artık ajanlar bölgeleri temsil ederken ajanların renkleri bölgelerin renklerini oluşturur [36].

GA çaprazlama ve mutasyon operatörleri ve YAA hız deplasman formülü ile birleştirilerek YAA'nın hızlı yakınsaması özelliği elde edilmiştir ve GA'nın küresel optimizasyonuna entegre edilmiştir. Çok seviyeli eşitleme kullanılarak görüntü bölümlenme çalışmalarında kullanılmıştır [37].

Tedarik zinciri içerisinde tedarikçiler, imalatçılar, dağıtıcılar, toptancılar, perakendeciler gibi çeşitli kısımlardan oluşur ve ürünün temininden son tüketiciye dağıtım ve pazarlanmasına kadarki tüm iş süreçlerinin birlikte uyum içinde hareketini sağlar, mal ve bilgi akışını yöneten bütünleşik bir sistemdir. Bu çalışma da, çeşitli coğrafi bölgeler içinde dağıtılan çoklu üreticilerin seri istifleme makineleri üzerinde işleme ihtiyaç duyan, ürün ve araçların planlama problemini incelemiştir. İki durumlu kaynak zinciri ortamlarında, araç planlama ve üretim problemlerini çözmek için kullanılmıştır. İşin süresi ve büyüklüğü tiplerin farklılığından dolayı değişkendir ve her bir seri işleme başlamadan önce kurulum zamanı alır. Tamamlama zamanını minimize eden problem tamsayı programlama modeli ile matematiksel formül haline getirilerek TPM modeli kullanılarak kanıtlanmıştır [38].

Bilgisayar haberleşme sistemlerinde kablosuz örgü ağlar ağ performansını ve diğer ağ hizmetlerini geliştirmek için ortaya çıkmıştır. Çoklu radyo örgü ağları içerisinde iyi kanal atama işlemi eş kanal girişim sayısını azaltır ve ağ verimini geliştirir. Yapılan bu çalışmada, geliştirilmiş YAA kanal atama problemlerini çözmek için önerilmiştir. Ayrık yerel arama operatörü YAA ile birleştirilmiştir. Ağ bağlantısı sağlandığı durumda ağ hacmini artırmak ve parazitleri minimize etmek için kullanılmıştır [39].

Ekonomik yük dağıtımını mühendislik uygulama alanlarında YAA ile çözüm arayışı en çok tercih edilen uygulama alanlarından biridir. Klasik ekonomik güç dağıtımında, yakıt maliyeti fonksiyonu ikinci dereceden bir fonksiyonla gösterilmektedir ve bu durum uygun çözümün fiziksel ihmallere dolaylı hatalı olmasına yol açmaktadır. Üretim biriminin yakıt maliyetine valf nokta etkisinin dahil edilmesi, yakıt maliyetinin

gösterimini daha uygun hale getirmekte ve uygun çözüme yaklaştırılması sağlanmaktadır. Valf nokta etkisi katılmasıyla yakıt maliyet fonksiyonu konveks olmayan karakteristiğe sahip olur ve sinüzoidal dalgalanmalar şeklindedir [40]. Yerçekimi arama algoritmasının valf nokta etkili ekonomik yük dağıtım problemlerine uygulanma çalışmaları da yapılmış olup, kısa yürütme zamanı ve yüksek uygun sonuçlar elde edilmiştir [41]. Ayrıca melez parçacık sürü optimizasyonu ve yerçekimi arama algoritması olan PSOYAA'nın küresel uygun değeri arama kabiliyetini artırmak için bulanık mantık kullanılmıştır. Bu algoritma çalışma içerisinde valf noktalı ekonomik yük dağıtımına uygulama örneklerinden biri olarak gösterilebilir [42].

Ekonomik güç dağıtımında fosil yakıtlı santraller tarafından yaratılan çevre kirliliği önemli bir konudur ve karbondioksit(CO₂), kükürt dioksit(SO₂), ve azot oksit(NO_x) içeren gaz ve kül parçacıkları üretmektedirler. Emisyon miktarının çok yüksek olması durumunda, öldürücü ekolojik etkilere sebep olabilmekte ve insanların yanı sıra diğer canlılara ve bitkilere zarar verebilmektedir. Aynı zamanda küresel ısınmaya da neden olduğu bilinmektedir [43]. Çevresel ekonomik güç dağıtımında, çevreye olan etkiyi azalmak için toplam emisyonu düşürmek amaçlanır ve daha az emisyon üreten birimleri daha fazla kullanmak üzerine çalışır [44,45]. Shaw ve arkadaşları YAA yakınsama hızını geliştirmek için zıt sayılar kullanılan yeni bir muhalif tabanlı algoritma elde etmişlerdir ve bu yeni algoritmayı birleştirilmiş ekonomik ve emisyon yük dağıtım problemlerine uygulamışlardır [46]. Bir diğer birleştirilmiş ekonomik ve emisyon dağıtım probleminde, YAA ile uygun sonuç bulma işlemi Güvenç ve arkadaşları tarafından önerilmiştir [47]. Termal jeneratörler ve rüzgar türbinlerindeki yakıt maliyeti ve NO_x yayılımını minimize eden ekonomik yük dağıtım problemi YAA kullanılarak gerçekleştirilmiştir. Ayrıca, rüzgar gücünün etkinliği ve önemi NO_x yayılımı açısından bu çalışma da incelenmiştir [48]. Jiang ve arkadaşları PSO ile YAA hız noktasında melez bir işlem yaparak, çeşitli kısıtlar altında ekonomik emisyon ve yük dağıtım problemlerine uygulamışlardır [49].

Dağıtım periyodu sadece bir zaman aralığı için ise elektrik güç sistemlerinde ekonomik dağıtım Statik Ekonomik Dağıtım (SED) denirken, çeşitli zaman aralıkları içeren bir süreç olduğu zamanda ise Dinamik Ekonomik Dağıtım (DED) denir. DED sürekli çalışma aralığında ünitelerin ve elektrik şebekesinin çalışmasının içsel kavrama özelliğine ihtiyaç duyduğu için daha çok kısıtlar ve yüksek değişkenli boyutlar içerir.

Bundan dolayı, SED' den daha uygun çözümlerin bulunması zordur. Ancak, DED güç sisteminin daha doğru çalışmasını ortaya çıkarabilir, verileri daha araştırılmış ve uygulanabilir özelliğe sahiptir [50]. Swain ve arkadaşlarının gerçekleştirdiği çalışmada dinamik ekonomik yük dağıtım problemi YAA kullanılarak çözümlenmeye çalışılmıştır. Bu çözümlenme de, ardışık kuadratik programlama metodu kullanılmadan önce ve sonra ki sonuçlar karşılaştırılmıştır [51]. Bir diğer çalışmada, ajanların hızını güncellemek için PSO'dan faydalandıktan sonra, fizibilite-tabanlı seçim karşılaştırma teknikleri rastgele seçim stratejisinin yerine konarak yeni bir YAA dinamik ekonomik yük dağıtım problemlerine uygulanmıştır [52].

Düşük gürültülü kuvvetlendirici, bir haberleşme sisteminin alıcı bloğunda genelde antenden sonra gelir ve diğer kuvvetlendiricilerden en önemli farkı, işarete eklediği gürültünün diğer kuvvetlendiricilerden çok daha az olmasıdır. Bahsi geçen özelliğe ulaşmak için düşük gürültülü kuvvetlendiricinin içeriğindeki parametrelerin uygun seçiminde YAA-K olarak adlandırılan yeni bir geliştirme algoritması kullanılmıştır. Standart YAA algoritmasının karmaşıklığı ve hesaplama zamanını azaltmayı amaçlamıştır [53].

Rüzgar ve güneş çiftliklerinin artışı ile birlikte yenilenebilir enerji entegrasyon sistemleri günden güne şekillenmekte ve önem kazanmaktadır. Bununla birlikte bu sistemlerin uygun kontrolünün yapılması, zorlukları beraberinde getirmektedir. Özellikle pompalı depolama hidro birimini (PHB) geri dönüşüm özellikleri olduğu için 'S' karakteristik alan içinde başarısız olma eğilimindedir. 'S' karakteristik alanı orta-düşük su başında boş çalışmada birim dönme hız salınımını geniş bir yelpazede kontrol eder. Bu kararsız çalışma problemlerini etkin bir şekilde çözmek için, uyarlanabilir hızlı bulanık kademeli sıra Oransal İntegral Türevsel (OİT) önerilmiştir. Bu kontrollerin en uygun parametre seçimi için ise Bakteriyel Yiyecek Arama Algoritmasının kemotaksi operatörü ile belirli olasılıklar dahilinde yerel minimumdan kaçınan YAA ile birleştirilmiştir [54].

Kontrol sistemlerinde optimizasyon tekniği olarak YAA'nın tercih edilmesini görmek mümkündür. OİT Kontrol endüstriyel kontrol sistemlerinde yaygın olarak kullanılan geri bildirim mekanizmasıdır. OİT kontrol de üzerinde çalışılan fiziksel nicelik ile istenilen ayar noktası arasındaki hata değerini hesaplar ve hatayı en aza indirerek istenilen değere ulaşmaya çalışır. Çalışma mantığında, oransal O, İntegral I ve türev

değeri T olmak üzere üç ayrı parametre içerir. OİT parametrelerinin ayarlanması için Ziegler-Nichols metodu, frekans domeni metodu ve zaman domeni metodu gibi yöntemler önerilmiştir. Yapay zekanın (YZ) gelişimi ile birlikte birçok dşşünel algoritmalar OİT parametrelerinin ayarlanmasında kullanılmıştır. Atık su arıtma tesisinde K_o , K_i ve K_t OİT parametrelerini ilgili proses de en uygun değeri YAA ile elde edilmiştir [55]. Elsis ve arkadaşlarının yaptığı başka bir çalışmada, iki alanlı güç sistemlerinde osilasyonların sönümlemesini hızlandırmak için YAA ile çalışan çift Oransal-İntegral kontrol gerçekleştirilmiştir [57]. Hata kriterinin integralini minimize etme, set noktasını takibini yapma ve bozucuları göz önünde bulundurma amaçlarını gerçekleştirmek için yeni bir kontrol tasarlanmıştır. Belirtilen tasarım kriterlerini gerçekleştirmek için OİT geri besleme döngüsünün dışına ileri ön filtre beslemesi ile birlikte iki dereceli serbest kontrol düzenlemesi yapılmıştır [54].

1.3.1. YAA Geliştirme Çalışmaları

Standart YAA'nın araştırma ve işletme yeteneklerini geliştirmek için birçok öneri getirilmiştir. Bunlardan ilkinde, Sarafrazi ve arkadaşları tarafından gerçekleştirilen çalışmada bozulma olarak adlandırılan özünü astrofizikten alan bir operatör kullanılmıştır. Bozulma operatörü doğal yaşamdan esinlenilmiştir. Yerçekimsel kuvvete bağlı kütlesi küçük parçacıkların büyük bir nesneye yaklaştığında paramparça olma eğilimindedirler. Bundan dolayı, ajanların kütleleri özel eşik değerinden daha küçük değerlere kesilerek YAA'nın keşif ve sömürü yetenekleri geliştirilmiştir [58].

Kaotik sistemde parametre tanımlama probleminin çözümü kaotik yerçekimi arama algoritması (KYAA) önerilerek gerçekleştirilmiştir. İlk olarak sonsuz çökme içeren iteratif kaotik haritalar tanıtılmış ve kaotik yerel arama (KYA) tasarlanmıştır. Sonra bir prosedür çerçevesi içerisinde KYA standart algoritma ile birleştirilmiştir. KYAA kaba standart yerçekimi arama algoritması ve ince kaotik yerel aramadan oluşmaktadır. KYAA'nın etkinliğini göstermek için Lorenz sisteminin çevrimiçi ve çevrimdışı parametre tanımlaması karşılaştırmalı deneyleri yapılmıştır [59].

Son yıllarda güvenlik iletişimde kaotik sistemlerin gelişimi görülmektedir. Ancak birçok kaotik sistem kanal gürültülerini ortadan kaldırma da başarısızdır. Doğrusal olmayan filtre modellemede kanal gürültülerini azaltmak için parametre tahmininde kullanılan bu çalışmada, standart YAA'nın içerisine kaotik operatör yerleştirilmiştir. Düzenlenmiş Yerçekimi Arama Algoritması olarak adlandırılan kaotik operatör

YAA'nın erken yakınsama dezavantajını minimize etmektedir [60].

Çok modlu problemlerin içinde çoklu çözümler bulunması için geliştirilmiş algoritma da ana sürü kütleleri küçük alt sürülere bölünür. K- yakın komşuluk ilişkisi olan eleme stratejisi kullanılmış ve alt sürüleri korumak için aktif yerçekimi kütle formül işlemi tekrar düzenlenmiştir [61].

Birçok araştırmacı, çok sık kullanılan ve bilinen sezgisel algoritmaların temel referans noktaları ve özelliklerini ortaya koymuş, güçlü yanları ve sakıncalarına göre algoritmaları birleştirme yoluna gitmişlerdir. YAA ile yapılan birleştirme çalışmalarını tarihsel akış içerisinde bazıları verilmiştir. Bunlardan ilkinde, PSO içerisindeki sosyal düşünce (gbest) yeteneği ile YAA'nın yerel arama kabiliyeti birleştirilmiştir [62]. K-harmonic ortalama (KHO) içerisine YAA'nın gelişmiş versiyonu eklenmiştir. Böylece kümeleme için melez bir algoritma oluşturulmuştur [63]. YAA'nın taşıma metotları ile Yapay Arı Kolonisi'nin araştırma mekanizması olan üç adım metodu birleştirilmiştir [64]. K-ortalama Yerçekimi Arama Algoritmasında, YAA için ilk popülasyonun üretilmesi için K-ortalama algoritması eklenmiştir. K-ortalama algoritması yerel optimumdan kaçınmaya ve YAA'nın yakınsama hızını artırmaya yardım etmiştir [65]. Kuantumdan ilham alınmış ikili (KEYAA) ile çapraz doğrulama (ÇDKY) ile K-en yakın komşu yöntemi (K-YK) birleştirilmiştir. İkili problemler içerisinde uygun özellik alt kümesi ile sınıflandırma doğruluğunu geliştirmiştir [66]. YAA içerisine bir önceki yerel optimumu ve küresel optimum çözümleri saklayan PSO özelliği uygulanmıştır. Böylece güncelleme süreci içerisinde YAA'nın içerisindeki parçacık bellek yeteneği küresel optimum ve sahip olduğu yerel optimum çözümleri hatırlamak için modifiye edilmiştir [67]. Bağımsız Araştırma Diferansiyel Gelişimi ile Gelişmiş Yerçekimi Arama Algoritmasında birleştirilmiştir. Her bir döngü boyunca YAA'ya tanıtılmış ve yeni bir parametre olan pro yerleştirilmiştir. Böylece YAA'nın istismar yeteneğinden ve yerel optimum içerisinde kolay tuzağa düşmenin üstesinden gelinmiştir [68]. Gerçek değerli (GDYAA) ile ikili (İYAA) birleştirilmiştir. GDYAA burada SVM parametrelerinin uygun değerini optimize etmek için ve İYAA ise giriş özelliklerinin alt küme seçim işleminin optimize etmek için tercih edilmiştir. Önerilen algoritma çalışması ikili sınıflandırma da kullanılmıştır [69]. Yerçekimsel Parçacık Sürü Optimizasyonunda PSO'nun hızı ve YAA'nın yakınsaması olan yerçekimi ile pozisyon güncellemesi yapılmıştır [70]. YAA'nın ve PSO'nun hızını esas alarak, parçacıkların

pozisyonlarını güncelleyen melez bir çalışmadır. Ekonomik emisyon yük sevk problemlerinde uygulanmıştır [71].

YAA çeşitliliğin çabuk kaybolmasından dolayı erken yakınsama problemi yaşamaktadır. Bu sakıncayı ortadan kaldırmak ve standart algoritmanın performansını artırmak için Parçalı Doğrusal ve Sıralı Kuadratik Programlama Yerçekimi Arama Algoritması önerilmiştir. Bu operatör çeşitlilik düzenleme mekanizmasını kullanır ve yerel arama stratejisini standart uygulamaya göre hızlandırır. Sürekli çok modlu ve çok değişkenli doğrusal olmayan optimizasyon problemlerini çözmek için B-kama eğrili veri montaj işlemine uygulanmıştır [72].

İbrahim ve arkadaşlarının önerdiği Kural Esaslı Çok Durumlu Yerçekimi Arama Algoritması çalışmasında, her ajanın hızı ve pozisyonu güncellenmesini esas alır. Her bir ajanın şu andaki durumu bir merkez gibi tanımlandıktan sonra güncellenecek hız bir yarıçap gibi daire oluşturarak gerçekleştirilir. Daire içerisinde pozisyonlandırılmış herhangi bir durum iç durum grup üyesi gibi, dairenin dışındaki ise dış grup üyesi gibi tanımlanmıştır. İç grupları I, dış grupları Q ve alabilecek durumları S ile gösterildiğinde bir sonraki döngü içerisinde alabilecek pozisyon bir denklem tercihi ile gerçekleştirilmiştir. Elde edilen algoritma, endüstriyel üretim ve ürün tasarım birimlerinde büyük değişiklikler yapan montaj dizi planlama (MDP) problemlerine uygulanmıştır. Diğer metotlara göre güçlü performans gösterdiği tespit edilmiştir [73].

1.3.2. YAA Yerçekimi Parametresi Üzerine Yapılan Çalışmalar

YAA içerisindeki yerçekimi parametresi, arama uzayı içerisindeki algoritmanın arama kabiliyetini etkileyen önemli bir değişkendir. Algoritma içerisindeki güçlü özelliğinden dolayı bu parametre üzerinde yapılan değişiklikler ve iyileştirmeler ayrı bir başlık altında incelenmiştir. Bu alandaki çalışmalarının ilkinde, YAA karakteristik yakınsamasını geliştirmek için kendinden uyarlamalı mutasyon tekniği sunulmuştur. Mutasyon tekniği içerisinde çözümleri düzenlemek için iki metot teklif edilmiştir. Olasılık modeline göre bu metotlardan biri seçilir. Olasılık modelinde , daha iyi uygun sonuç sağlamak için her bir metodun yeteneğini esas alır.

Metot 1: YAA da az bellek gerektiren algoritma: Parçacıklar bir önceki döngüde bulunan uygun bilgileri kullanmaz. Bu mutasyon tekniğinde o ana kadar algoritma tarafından bulunmuş en iyi sonuçları kullanımı için tasarlanmıştır.

Metot 2: Bu mutasyon tekniğinde durgunluğu azaltmak ve yerel optimumdan kaçınmak gibi problemleri çözmek için önerilmiştir [74].

Sombra ve arkadaşları bulanık mantık kullanarak her bir ajanın hızını ve yerçekimi parametresini değiştiren α parametresi üzerinde çalışmışlardır. Döngü boyunca α değerini değiştirildiğinde, farklı bir yerçekimi parametresi değeri elde edilmiş olur. Farklı yerçekimi parametresi değeri oluşturarak her bir ajanın yerçekimi kuvveti etkilenmiş ve arama uzayı içerisinde ajanların ivmesi değişmiştir. Böylece nihai sonuç iyileşir. Bu yaklaşımda iki uygulama elde edilmiştir. Birincisinde, verilen farklı yerçekimi ve ivmeleri denemek için α değeri rastgele değişim şeklinde kullanılmıştır. Diğer uygulamada ise α parametrelerini değiştirmek için bulanık mantık kullanılmış, döngü sürecinde α değeri artan veya azalan olabilecektir.

Her bir döngü içerisinde α çok küçük vermek zorunda olduğu tecrübe ile tespit edilmiştir. Çünkü, büyük miktarda hızlanma ve kuvvet uygulayabilmek için yerçekimi parametre değerinin yüksek olmasına ihtiyaç vardır. Son döngüler içerisinde ise α değerini yüksek verme zorunluluğu vardır. Çünkü bu durum, algoritmanın sömürü amacına yardım edecek yerçekimi parametre değerinin çok küçük seçilmesiyle hassas arama işlemi gerçekleştirilmiş olacaktır [75].

Niknam ve arkadaşları YAA algoritmasının performansını büyük oranda etkileyen parametrelerden olan yerçekimi parametresi ve $Kbest$ (en iyi uygunluk sayısı) üzerinde çalışmışlardır. En iyi sonuçlara ulaşmak için standart çalışma boyunca bu parametrelerin nasıl değişeceği pek açık değildir. Bu çalışma içerisinde YAA algoritmasının giriş değerleri yerçekimi parametresi ve $Kbest$ belirlenen verilen sınırlar içerisinde üretilmiştir. Ayrıca her bir döngüde YAA ile birlikte güncellenirler yani $Kbest$ ve yerçekimi parametresi rastlantısal değışkendirler.

Kendinden uyarlamalı olarak adlandırılan bu yapı yerçekimi parametresi ve $Kbest$ sabit veya doğrusal değışen şeklinde olduğu zaman, daha iyi sonuçları amaçlamaya yardım eder. Zıt Esaslı Kendinden Uyarlamalı Yerçekimi Arama Algoritması (ZEKUYAA) metodolojinin arkasındaki temel fikir, umut verici üretilen çözümlerdeki tecrübeleri esas alan birkaç etkili mutasyon metodunu uyarlamalı olarak seçme işlemidir. Bu nedenle optimizasyon prosedürünün farklı aşamalarında, çoklu strateji yöntemi, önceki

nesillerde belli sayılar içerisinde gelişmiş çözümler üreten başarı oranını temel alan farklı olasılıkları atayabilir ve ZEKUYAA içerisinde iki mutasyon stratejisi kullanılmıştır [76].

Kumar ve arkadaşlarının 2013 yılında yaptıkları Bulanık Uyarlamalı Yerçekimi Arama Algoritması (BUYAA) göre, yerçekimi parametresi aramanın başarı ve başarısızlığında önemli rol alır. Arama başarısızlığının ana sebepleri aşağıda verilmiştir.

- ✓ Ajanların hızlarını hızlıca artırmak ve ajanları arama uzayının dışına itmek.
- ✓ Ajanların hızlarını hızlıca yavaşlatmak ve ajanları hareketsiz yapmak.
- ✓ Ajanlar yerel optimum sonuçtan kaçırılmamak

Arzulanan bu durumlardan kaçınmak için sapma ile ilgili ve ajanların yakınsaması ile ilgili ajanların davranışları ile parametreler arasındaki ilişkiyi düzgün analiz etmek önemlidir. Bu yüzden bulanık IF/THEN kuralları kullanılarak bulanık uyarlamalı dinamik yerçekimi parametresi tasarlanmıştır. BUYAA konsepti içerisinde pozisyon ve hız güncellemeleri klasik YAA ile aynıdır. Fakat, yerçekimi parametresi bulanık IF/THEN kuralları kullanılarak döngü büyüdükçe dinamik bir şekilde ayarlanmıştır [77].

Yerçekimi parametresi, yerçekimi kuvvetinin etkisini kontrol eden modüldür. Geçerli arama alanı içerisinde algoritmanın ince ayara doğru hareket etmesini küçük yerçekimi parametresi yaparken, daha büyük yerçekimi parametresi algoritmayı küresel aramaya doğru hareket ettirdiği gerçektir.

Yerçekimin azalma etkisinden dolayı, yerçekimi parametresinin gerçek değeri evrenin yaşına bağlıdır. Geleneksel YAA da yerçekimi parametresinin değişimini göstermek için üstel fonksiyon kullanılmıştır.

YAA'nın arama yeteneğini geliştirmek için, bu çalışma içerisinde bir tür parçalı fonksiyon tipi olan yeni yerçekimi parametre fonksiyonu tasarlanmıştır. Yerçekimi parametresinin azalma oranını kontrol etmek için üstel fonksiyon ile üç parçaya bölünmüş parçalı fonksiyon karşılaştırıldığında, parçalı fonksiyonun daha esnek olduğu görülmüştür. Bölümlenmiş üç parça sırasıyla, kaba arama durumu, orta evre arama ve ince arama durumu olarak adlandırılır. İlk kaba arama durumunda, yerçekimi parametresi büyük oranlarda azaltılarak, arama hızı azaltılması amaçlanmıştır. İkinci aşama

içerisinde yerçekimi parametresinin azalma oranı yavaşlatılarak, küresel optimum değere yaklaşma amaçlanmıştır. Son aşama olan ince arama durumunda, yerçekimi parametresinin azalma oranı çok küçük tutularak, titiz bir şekilde küresel optimum değerine yaklaşma ve yakalama gerçekleştirilmiştir [78].

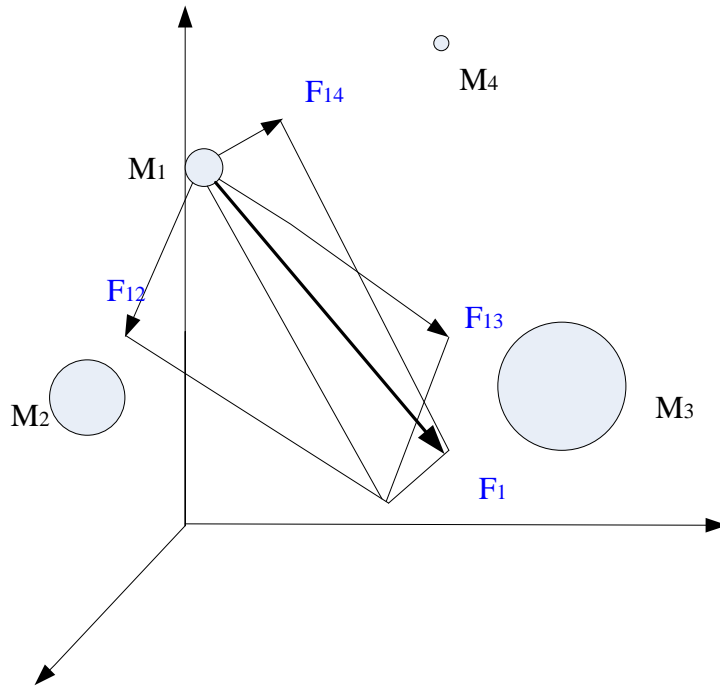


2. MATERYAL VE YÖNTEM

Bu bölümde, öncelikle Yerçekimi Arama Algoritmasına esin kaynağı oluşturan yasalar ve klasik algoritmanın tanıtımı yapılmıştır. Daha sonra bu algoritmanın performansını etkileyen parametreler detaylı bir şekilde incelendikten sonra algoritmaya etkileri üzerine tespitlerde bulunulmuştur. Tespitlerin ışığında klasik algoritmanın performansını artıracak Kaotik Sarsıntılı Yerçekimi Arama Algoritması, Kurtulma Hızlı Yerçekimi Arama Algoritması ve En İyi Ajan Yerçekimi Arama Algoritması olmak üzere üç yeni operatörün tanıtımına yer verilmiştir.

2.1. YERÇEKİMİ ARAMA ALGORİTMASI

Doğada, yerçekimi kuvveti, elektromanyetik kuvvet, zayıf nükleer kuvvet ve güçlü nükleer kuvvet olmak üzere dört çekim kuvveti mevcuttur. Yerçekimi her yerdedir ve bu kaçınılmaz özelliği ile diğer kuvvetlerden farklıdır.



Şekil 2.1. Newton yerçekimi kanununun şekilsel gösterimi [4].

Newton'un yerçekimi kanununa göre Şekil 2.1'de sembolik olarak gösterildiği gibi, her parçacık diğer parçacıkları yerçekimi kuvveti ile birlikte çeker. İki parçacık arasındaki

yerçekimi kuvveti onların kütleleri ile doğru orantılı aralarındaki mesafenin karesi ile ters orantılıdır.

$$F = G \frac{M_1 \cdot M_2}{R^2} \quad (2.1)$$

SI birimlerine göre, Denklem (2.1)'deki F yerçekimi kuvvetinin büyüklüğü ve birimi Newton (N), G yerçekimi sabiti ve yaklaşık olarak $6.67 \cdot 10^{-11} \text{ Nm}^2/\text{kg}^2$ 'a eşittir. M_1 ve M_2 birimleri kilogram (kg) olup, sırasıyla birinci ve ikinci parçacıkların kütleleri ve R , birimi Metre (m) iki parçacık arasındaki mesafedir. Yerçekimi kuvvetini, 1687 yılında Isaac Newton Principia kitabında yayınlamıştır. Klasik mekaniğin temelleri oluşturulan bu kitap tarihte en önemli bilimsel kitaplardan biri olarak yer almıştır. Ayrıca bu çalışmasında hareketin üç kanununu da ortaya koymuştur [79].

Newton hareket yasalarından ikinci kuralında, bir parçacığa F kuvveti uygulandığı zaman, a ivmelenmesi uygulanan kuvvete ve parçacığın kütlesine bağlı olduğunu söyler.

$$a = \frac{F}{M} \quad (2.2)$$

Denklem (2.1) ve Denklem (2.2)'ye göre, evrendeki bütün parçacıklar arasında büyük ve birbirlerine yakın olanların yerçekimi kuvveti etkisi daha yüksektir. Dirac ve Jordan teorisine göre yerçekimi zamanla azalacak ve oran ω parametresi ile evrenin gerçek yaşına bağlıdır. Denklem (2.3) yerçekimi sabitinin yaş ile birlikte azalmasını verir.

$$G(t) = G_0 \cdot \left(\frac{t_0}{t}\right)^{-\delta} \quad (2.3)$$

Denklem (2.3)'deki $G(t)$, t zamanındaki evren yaşının yerçekimi sabitinin değeri, G_0 ise t_0 başlangıç zamanındaki ilk kozmik kuantum yerçekimi sabitinin değeridir. Eşitlikteki δ azalma oranı $\delta = 2/(4 + 3\omega)$ şeklinde ω ile ilişkilendirilmiştir. Dirac teorisine göre şu andaki yerçekimi kuvveti kozmik zamandakinden yirmi kat daha azdır ve bu durumda $\omega = 6$ alınır [80].

Temel fizik bilimi içerisinde aktif yerçekimi kütlesi, pasif yerçekimi kütlesi ve

eylemsizlik kütlesi çeşitlerini Newton'un yasaları ile ilişkilendirmek mümkündür. Aktif yerçekimi kütlesi, M_a , maddenin kütle çekimsel alanının bir ölçüsüdür, bir nesnenin serbest düşmesine izin verip ivmesini ölçerek hesaplanabilir. Örneğin, Ay'ın yüzeyinde serbest düşme yapan bir cismin ivmesi, Dünya'nın üzerinde serbest düşme yapan bir cismin ivmesinden küçüktür. Pasif yerçekimsel kütle M_p , maddenin kütle çekim alanıyla etkileşime giren ve ivmelenmeye maruz kalan büyüklüktür. Eylemsizlik kütlesi M_i , bir cismin ivmelenmeye karşı gösterdiği dirençtir ve kuvveti ve ivmesi biliniyor ise eylemsizlik kuvvetini hesaplanabilir. Aynı kuvvet uygulandığında küçük kütleli cisimler büyük kütleli cisimlere göre daha büyük bir ivmelenmede bulunacaklardır yani büyük kütleli cisimler büyük eylemsizliğe sahiplerdir [4,81].

Zayıf eşitlik prensibine göre, cismin eylemsizlik kütlesi pasif yerçekimsel kütlesine eşittir. Ayrıca izafiyet teorisi ve momentumun korunumu ile ilgili diğer teoriler içerisinde pasif yerçekimsel kütle aktif yerçekimsel kütleyle eşit. Sıvının enerji momentum ifadesi, Einstein alan eşitliği ve kütle enerji yoğunluğu ve aktif yerçekimsel kütle yoğunluk hareketlerini veren diğer kaynaklar içerisinde aktif, pasif ve eylemsizlik kütlelerinin eşit alınabileceği belirtilmektedir [82].

Bir cismin geçerli hızı, cismin ivmesi ve onun önceki hızının toplam miktarına eşittir. Bu durumda, cismin yeni alacağı pozisyonda bir önceki pozisyonuna yeni oluşan hızın eklenmesi ile belirlenir [83].

Sezgisel optimizasyon tekniklerinde en önemli kavramlardan bir tanesi de uygunluk fonksiyonu veya İngilizce kelime kullanımı ile fitness fonksiyonudur. Günlük yaşamda karşılaşılan karmaşık problemlerin bazılarını sayısal yöntemler ile çözümünü bulmak oldukça zordur. Fonksiyonun alabileceği değerle uzayı büyük olduğundan bu problem daha da zorlaşır. Bu tip problemler için iki şeyin kesin belirlenmiş olması gerekir. Problem çözümlerinin kodlanması ve çözümlerinin aranan çözüme yakınlığının ölçüsü olan uygunluk fonksiyonunun tespitidir. Uygunluk fonksiyonu bir elemanın (olası bir çözümün) aranan elemana (gerçek aranan çözüme) yakınlığının ölçüsünü verecek şekilde seçilmiş fonksiyondur. Optimizasyon problemlerinde bu fonksiyon, optimumu aranan fonksiyon olarak seçilebilir. Çözüm aralığında rastgele seçilmiş M tane eleman $\vec{\chi}_1, \vec{\chi}_2, \dots, \vec{\chi}_M \equiv \{\vec{\chi}_k\}$ $k = 1 \dots M$ olacak şekilde kabul edilir.

$$f_k \equiv f(\vec{\chi}_k) \quad (2.4)$$

$$\bar{f} = \sum_{k=1}^M f_k \quad (2.5)$$

Denklem (2.4) k . elemanın uygunluğu, Denklem (2.5) bütün elemanların değerlerinin toplamı olması durumunda Denklem (2.6) k . elemanın uygunluk fonksiyonunu vermektedir [84].

$$U_k = \frac{f_k}{f} \quad (2.6)$$

Yerçekimi kanununu esas alan YAA, aramayı gerçekleştirecek olan ajanlar bir nesne gibi düşünülmüş ve onların performansları kütleleri ile ölçeklendirilmiştir. Bütün bu nesnelere yerçekimi kuvveti ile birbirlerini çekerler ve bu kuvvet daha ağır kütleli nesnelere doğru bütün nesnelere hareketine sebep olur. YAA da, her bir ajan, konum, pasif yerçekimi kütlesi ve aktif yerçekimi kütlesi, atalet kütlesi olmak üzere dört özelliğe sahip olur. Ajanın konumu, problemin bir çözümüne karşılık gelirken, diğer yerçekimi ve atalet kütleleri uygunluk fonksiyonunun kullanımını belirler.

N ajana sahip bir sistem düşünelim ve i . ajanın konumunu aşağıdaki Denklem (2.7)'deki gibi tanımlanır.

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n) \quad i = 1, 2, \dots, N \quad (2.7)$$

Denklemdaki n , üzerinde çalışılan problemin boyut sayısı ve x_i^d ise i adlı ajanın d . boyut içerisindeki pozisyonu şeklinde tanımlanır. Her ajanın bütün boyutlar içerisinde bir pozisyonu vardır ve i ajanın bütün boyutlardaki pozisyonlarını X_i vektörü ile gösterilmiştir.

YAA algoritması aramaya başlamadan önce ilk olarak ajanlar rastgele olarak dağıtılırlar ve böylece ilk döngü için ajanlar pozisyonlarını almış olur. Bu pozisyon bilgilerine göre uygunluk değerleri, kütleleri, kuvvetleri, hız ve ivme gibi büyüklükleri hesaplanır. Belirli bir t zamanında j ajanını i ajanı üzerindeki kuvvet etkisi aşağıdaki gibi tanımlanır [4].

$$F_{ij}^d(t) = G(t) \frac{M_{pi}(t).M_{aj}(t)}{R_{ij}(t)+\varepsilon} (x_j^d(t) - x_i^d(t)) \quad (2.8)$$

Denklem (2.8)'de ki M_{aj} , j ajanına bağlı aktif yerçekimi kütlelerini, M_{pi} , i ajanına bağlı pasif yerçekimi kütlelerini, $G(t)$ t zamanının da ki yerçekimi sabitini, ε iki ajanın arasındaki öklid mesafenin sıfır olması durumunda oluşacak olan problemi ortadan kaldırmak için etkisi çok az olması için küçük bir sabit sayı, $R_{ij}(t)$ i ve j ajanları arasındaki öklid mesafesini temsil etmektedir. Standart YAA uygulamasında, Denklem (2.1) yer alan yerçekimi kanunundaki formüle göre öklid mesafesinin karesi daha iyi sonuç elde etmesinden dolayı alınmamıştır. Ayrıca Denklem (2.8)'de $(x_j^d(t) - x_i^d(t))$ ifadesi ilgili boyuttaki pozisyon vektörünü göstermektedir ve standart YAA da eklendiği görülmektedir. Bunun sebebi olarak da her boyuttaki ajanların birbirlerine yapmış oldukları kuvveti boyut içerisindeki mesafelerini de katarak boyut içerisinde kuvvetin tesirini daha iyi ortaya koymaktır. $R_{ij}(t)$ i ve j ajanları arasındaki öklid mesafesi Denklem (2.9) genel gösterimi ile, Denklem (2.10)'da daha açık bir ifade ile gösterilmiştir.

$$R_{ij}(t) = \|X_i(t), X_j(t)\|_2 \quad (2.9)$$

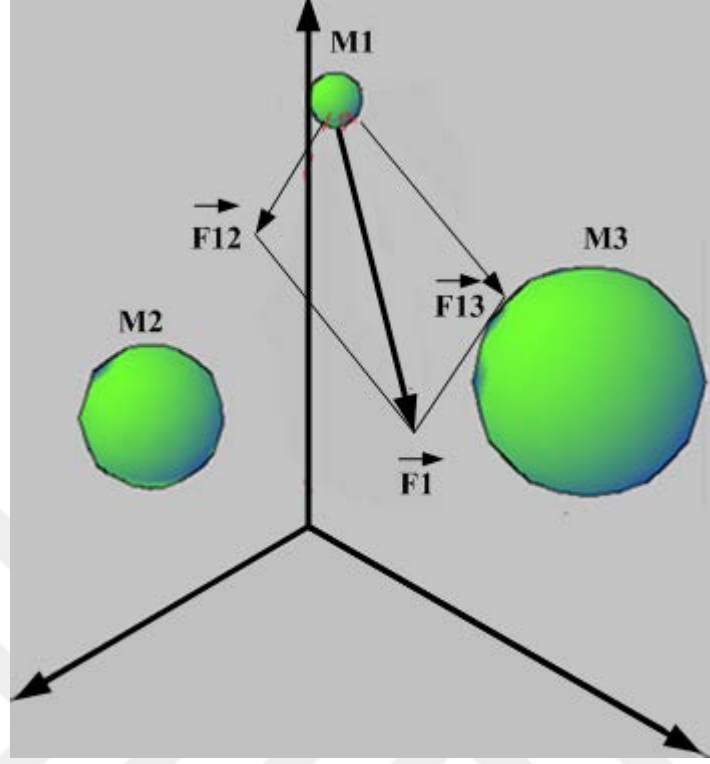
$$R_{ij}(t) = \sqrt{\sum_{d=1}^n (x_i^d(t) - x_j^d(t))^2} \quad (2.10)$$

Algoritmaya rastgelelik özelliği vermek için d boyutu içerisindeki i ajanının üzerindeki toplam kuvvet, diğer ajanların d . boyuttaki i ajanı üzerinde rastgele ağırlaştırılmış toplamları şeklinde ifade edilir. Böylece her bir ajan için her boyut da ona tesir eden toplam kuvvet hesaplanmış olur. Denklem (2.11)'deki $rand_j$ ifadesi $[0,1]$ aralığında olabilir ve rastgele belirlenir.

$$F_i^d(t) = \sum_{j=1, j \neq i}^N rand_j F_{ij}^d(t) \quad (2.11)$$

Yukarıdaki Denklem (2.11)'deki $F_i^d(t)$ i ajanının toplam kuvvet vektörünü, j ise arama içerisinde i ajanına tesirde bulunan bütün ajanları temsil etmektedir. Algoritma içerisinde ajanın kendisine bir kuvvet uygulayamamağından $i=j$ durumu kabul

edilmemiştir.



Şekil 2.2. Ajana etkiyen toplam kuvvetin şekilsel gösterimi.

Şekil 2.2'de M1 ajanına etkiyen iki ajanın yerçekimsel kuvveti sembolize edilmiştir. Bu sembolik gösterimin amacı Denklem (2.11) daha iyi anlam kazandırmak hedeflenmiştir. Şekil 2.2'ye göre M1 ajanının etkiyen toplam kuvvet Denklem (2.12) de verilmiştir.

$$\vec{F}_1 = \vec{F}_{12} + \vec{F}_{13} \quad (2.12)$$

Denklem (2.11) ilgili boyut için olmasının yanında, YAA algoritmasına rastlantısal özellik vermek için (0,1) aralığında bir çarpan katılmıştır. Denklem (2.9) da ise matematiksel vektör işlemini mantıksal manada sunulmuştur. Bu aşamadan sonra, hareket kanunu kullanarak i ajanının ivmesi hesaplanır.

$$a_i^d(t+1) = \frac{F_i^d(t)}{M_{ii}(t)} \quad (2.13)$$

Denklem (2.13) içerisindeki M_{ii} ifadesi, i ajanının eylemsizlik kütesini temsil

etmektedir. Ayrıca ajanın gelecek hızı şu anki hızına ivmesi eklenerek, ajanın yeni pozisyonu ve hızı aşağıdaki gibi hesaplanabilir.

$$v_i^d(t + 1) = rand_i * v_i^d(t) + a_i^d(t + 1) \quad (2.14)$$

$$x_i^d(t + 1) = x_i^d(t) + v_i^d(t + 1) \quad (2.15)$$

Yine yukarıdaki Denklem (2.14)'deki $rand_i$ ifadesi $[0,1]$ aralığında rastgele bir değer olup, araştırmaya rastgele bir karakteristik vermek için kullanılmıştır.

Yerçekimi parametresi olan $G(t)$, arama doğruluğunu kontrol etmek için azaltılacak ve başlangıçta G_0 denilen sabit bir değer ile başlatılmıştır. Denklem (2.16) genel gösterimi verilirken uygulama da Denklem (2.17)'deki zamanın fonksiyonu olarak azalan üstel bir fonksiyon tercih edilmiştir.

$$G(t) = G(G_0, t) \quad (2.16)$$

$$G(t) = G_0 \cdot \exp(-\alpha \cdot \frac{t}{T}) \quad (2.17)$$

Denklem (2.17)'deki α küçük üstel bir sabit olup, $G(t)$ 'nin azalma oranını ve azalma şeklini etkiler. G_0 sabit ve yerçekimi parametresinin başlangıç değeri olarak kabul edilir. Denklem (2.17) de $G(t)$ 'nin sonuçlarına etki yapan T optimizasyon süreci içerisinde maksimum döngüyü, t ise arama esnasındaki içinde bulunulan döngü sayısını temsil etmektedir. Döngü ilerledikçe $-\alpha \cdot t$ değeri büyümesinden dolayı $G(t)$ azalacaktır. Aramanın başlangıç evrelerindeki döngüler içerisinde $-\alpha \cdot t$ 'yi çok küçük vermek zorunda olduğu tecrübe ile tespit edilmiştir. Çünkü, büyük miktarda hızlanma ve kuvvet uygulayabilmek için $G(t)$ değerinin yüksek olmasına ihtiyaç vardır. Böylece algoritmanın ortaya çıkartılmasına yardımcı olur. Son döngü içinde ise $-\alpha \cdot t$ değerini yüksek verme zorunluluğu vardır. Çünkü bu durum, algoritmanın sömürü amacına yardım edecek $G(t)$ değerinin çok küçük seçilmeye ihtiyacı vardır.

Yerçekimi ve eylemsizlik kütleleri uygunluk değerlendirilmesi ile basitçe hesaplanabilir. Ağır bir kütleli anlamı çok etkili bir ajan şeklinde düşünülür. Bunun

anlamı büyük ajanların daha büyük çekimlere sahip olduğu ve daha yavaş hareket ettikleri şeklindedir. Yerçekimi ve eylemsizlik kütlelerinin eşit olduklarını varsayarsak, aşağıdaki eşitlik ile yerçekimi ve eylemsizlik kütleleri güncellenebilir.

$$M_{ai} = M_{pi} = M_{ii} = M_i \quad i = 1, 2, \dots, N \quad (2.18)$$

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad (2.19)$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)} \quad (2.20)$$

Denklem (2.19)'da yer alan $fit_i(t)$ t zamanında i ajanının uygunluk değerini temsil eder. Denklem (2.20) matematiksel ifade ilgili ajanın uygunluk değeri hesaplandıktan sonra bütün ajanların uygunluk değerlerinin toplamına oranını yani o ajanın kütesini gösterir. Ayrıca Denklem (2.19) yer alan $worst(t)$ ve $best(t)$ değerleri minimize problemleri için aşağıdaki şekilde gibi tanımlanır.

$$best(t) = \min_{j \in \{1, \dots, N\}} fit_j(t) \quad (2.21)$$

$$worst(t) = \max_{j \in \{1, \dots, N\}} fit_j(t) \quad (2.22)$$

Maksimizasyon problemi için yukarıdaki eşitlikler sırasıyla yer değiştirerek aşağıdaki eşitlikler kullanılabilir.

$$best(t) = \max_{j \in \{1, \dots, N\}} fit_j(t) \quad (2.23)$$

$$worst(t) = \min_{j \in \{1, \dots, N\}} fit_j(t) \quad (2.24)$$

YAA algoritmasında rastgelelik özelliğini artırmak için Denklem (2.11)'de d boyutu içerisindeki i ajanının üzerindeki toplam kuvvet hesaplanırken bütün ajanların i ajanına etkiyen kuvvetlerinin toplamları şeklinde ifade edilmiştir. Ancak araştırma yeteneğini artırma ve kötü arama durumu arasında iyi bir uzlaşma yolu sağlamak ve hesaplama

fazlalığını azaltmak için ajan sayısını azaltma gerçekleştirilmiştir. Bundan dolayı kuvvet hesabında ilgili ajana büyük kuvvetler uygulayan ajanların kümesi önerilmiştir. Ancak kötüye kullanım yeteneğini artırma ve araştırma gücünü azaltabileceği için dikkatli olunması gerekir. $Kbest$ başlangıç değeri ile birlikte ve zamanla azalan zamanın bir fonksiyonudur. Böyle bir yol içerisinde başlangıçta, bütün ajanlar kuvvet uygular ve zaman geçtikçe $Kbest$ doğrusal olarak azalır. Ve sonunda, diğerlerine kuvvet uygulayan tek bir ajan olacaktır. Bundan dolayı, Denklem (2.11) tekrar aşağıdaki gibi düzenlenmelidir.

$$F_i^d(t) = \sum_{j \in Kbest, j \neq i}^{N} rand_j F_{ij}^d(t) \quad (2.25)$$

Eşitlikteki $Kbest$ en iyi uygunluk değeri ve en büyük kütle sahip ajanlar kümesidir ve döngüler ilerledikçe azalacaktır.

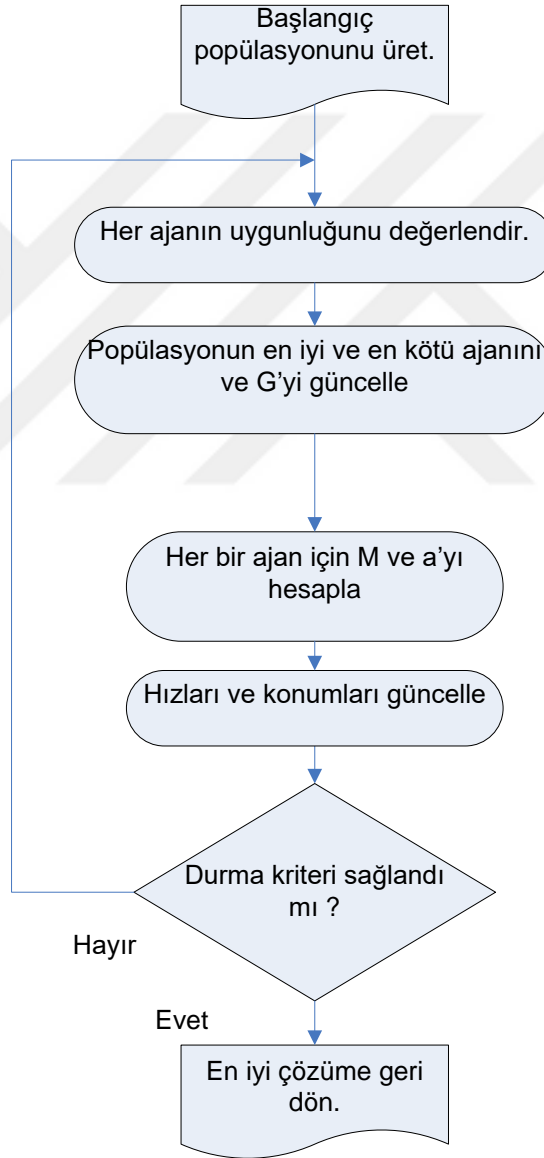
$$Kbest = son_ajan + \left(1 - \frac{t}{T}\right) \cdot (100 - son_ajan) \quad (2.26)$$

Denklem (2.26)'da ki son_ajan katsayısı en son döngüde kuvvet hesabında yer alacak ajana sayısını göstermektedir. Ayrıca denklemdeki t arama esnasına içinde bulunulan döngüyü, T aramanın maksimum döngü sayısını temsil etmektedir. Standart YAA adımları aşağıda verilmiştir [4].

- a) Arama alanını tanımlama
- b) Rastgele başlatma
- c) Ajanların uygunluk değerlerinin tespiti
- d) $i = 1, 2, \dots, N$ için $G(t)$, $best(t)$, $worst(t)$ ve $M_i(t)$ değerlerini tekrar güncelleme
- e) Farklı yönler içerisindeki toplam kuvveti hesaplama
- f) İvme ve hızı hesaplama
- g) Ajanların pozisyonlarını güncelleme
- h) c'den g'ye kadar adımları verilen kritere ulaşana kadar tekrarlama
- i) Sonlandırma

YAA'nın prensipleri ve akışı Şekil 2.3 de gösterilmiştir. Aşağıda algoritmanın nasıl çalıştığını görmek için etkili uyarılara ve YAA özelliklerine yer verilmiştir.

- ❖ Her bir ajan diğerlerinin performanslarından etkilenir, bu yüzden yerçekimi kuvveti bilgi aktarma ve ajanlar arası iletişim aracıdır.
- ❖ Bir ajanın kuvveti onun komşularından etkilenir, bundan dolayı pozisyon alma yönünden onu çevresindeki alanda görebilirsiniz.
- ❖ Ağır bir ajan etkili büyük bir çekim yarıçapına ve çekim yoğunluğuna sahiptir. Bu yüzden yüksek performanslı ajanlar büyük bir yerçekimi kuvvetine sahiptirler. Sonuç olarak, ajanlar en iyi ajana doğru hareket etme eğilimindedirler.



Şekil 2.3. Yerçekimi arama algoritmasının akış şeması [4].

- ❖ Eylemsizlik kütlesi harekete karşıdır ve ajanın hareketini yavaşlatır. Bu yüzden ağır atalet yığınları ajanlar yavaş hareket eder ve bu yüzden daha yerel arama alanı araştırırlar.
- ❖ Yerçekimi parametresi araştırmanın doğruluğunu düzenler, zamanla birlikte azaltılır.
- ❖ YAA bellek-azaltıcı algoritmadır. Ancak bellek algoritmaları gibi etkili çalışır.
- ❖ YAA da yerçekimi ve eylemsizlik kütleleri aynı varsayılmıştır. Araştırma uzayı içerisinde büyük bir eylemsizlik kütlesi ajanın hareketini ağırlaştırır ve bu yüzden daha hassas bir arama gerçekleşir. Bu söylenenin tam aksine büyük bir yerçekimi kütlesi ajanın daha hızlı hareketine sebep olur. Bu durum da hızlı bir yakınsama özelliğinin kazanılmasını sağlar [4].

2.2. YAA PARAMETRELERİNİN İNCELENMESİ

YAA optimizasyon algoritması içinde birbirlerine bağlı birçok sabit ve değişken vardır. Örneğin yerçekimi parametresi her döngüdeki değeri ajanların hızlarını ve pozisyonlarını etkilemektedir, yani YAA'nın arama kabiliyeti büyük ölçüde bu parametreye bağlıdır. Yine bir başka örnek olarak, algoritmanın uygun döngü ve ajan sayısı en iyi sonucu belirleme işleminde yapılması gereken en önemli temel başlangıç adımıdır. Bundan dolayı algoritmanın davranışı ve parametreleri arasındaki nitel ve nicel ilişkinin iyi bir analizi gereklidir. Algoritmanın bu yönde modifikasyonu için bu analizlerin sonuçlarının çok dikkatli incelenmesi gerekir [85]. Bundan dolayı, arama koşulları ve problemin durumuna göre, iyi sonuçlar veren çözüm önerileri getirilebilecektir. Bu bölümde parametreler arama sonuçlarına (en uygun değere) ve arama sürelerine nasıl etki ediyor, parametrelerin en uygun değerleri için ne söylenebilir gibi sorulara yanıtlar aranmıştır.

2.2.1. Yerçekimi Başlangıç Sabiti (G_0)'ın YAA Üzerindeki Etkisi

Denklem (2.17)'de yer verilen yerçekimi parametresi $G(t)$, aramayı gerçekleştiren ajanların kuvvet hesabında yer alan hız ve pozisyonlarına etki eden ve zamanla azalan en önemli parametrelerden biridir. Başlangıçta G_0 denilen sabit bir değer ile başlatılmıştır. YAA $\alpha=20$, $N=30$, boyut=30, döngü=1000, değişkenleri en uygun değerler olarak seçilmiş olup, deney süresince değiştirilmemiştir. Ek-1'de yer alan tek boyutlu karşılaştırma fonksiyonlarından F1, Ek-2'deki çok boyutlu karşılaştırma

fonksiyonlarından F12 ve Ek-3'deki çok modlu düşük boyutlu fonksiyonlardan F23 test fonksiyonları amaç fonksiyonu olarak kullanılmıştır. YAA bütün bu parametreleri sabit tutulmuş, sadece yerçekimi parametresi başlangıç sabiti G_0 1'den başlanılarak 10, 100, 1000 ve 10000' e kadar beş farklı değer üzerinden değiştirilerek sonuç değerleri ve optimizasyon bitiminde geçen süre veri olarak kaydedilmiştir.

Çizelge 2.1 G_0 değerinin 1 ve 10 aralığında sonuç değerinin beş basamaklı büyük değerler oldukları, görülmektedir. 100 değerinde sonuç değerlerinin dört basamaklı sonuçlara düştüğü, 1000 ve 10000 sütunlarında sonuçların üç basamaklı olduğunu ve iki sütun arasında değişikliğin fazla olmadığı tespit edilmiştir.

Çizelge 2.1. F1 için G_0 başlangıç değerinin YAA'ya etkisi.

G_0 yerçekimi sabitinin YAA'ya etkisinin incelenmesi										
Kul.Fonk.	$G_0=1$		$G_0=10$		$G_0=100$			$G_0=1000$		$G_0=10000$
	En İyi	G. Zaman	En İyi	G. Zaman	En İyi	G. Zaman	En İyi	G. Zaman	En İyi	G. Zaman
F1	53236	0,9483	61691	0,9541	2453	0,9553	453	0,9442	538	0,9642
	80134	0,9508	63955	0,9538	2677	0,9511	210	0,9497	114	0,9457
	72668	0,9356	56435	0,9429	4524	0,9500	511	0,9561	674	0,9509
	54302	0,9491	58239	0,9564	3688	0,9649	313	0,9544	455	0,9485
	66319	0,9498	56822	0,9558	2882	0,9433	593	0,9624	552	0,9622
	67253	0,9359	61178	0,9463	1720	0,9532	516	0,9423	364	0,9481
	54383	0,9464	57082	0,9426	2768	0,9411	349	0,9546	237	0,9292
	55450	0,9401	57463	0,9317	2202	0,9473	614	0,9524	301	0,9547
	76719	0,9262	68304	0,9441	2464	0,9520	839	0,9372	771	0,9347
	68532	1,0192	52585	0,9520	2111	0,9533	280	0,9447	232	0,9504
ORT	64899	0,9501	59375	0,9479	2748	0,9511	467	0,9498	423	0,9488

Çizelge 2.2. F12 için G_0 başlangıç değerinin YAA'ya etkisi.

G_0 yerçekimi sabitinin YAA'ya etkisinin incelenmesi											
K.F.	$G_0=1$		$G_0=10$		$G_0=100$		$G_0=1000$		$G_0=10000$		
	En İyi	G. Zaman	En İyi	G. Zaman	En İyi	G. Zaman	En İyi	G. Zaman	En İyi	G. Zaman	
F12	532090000	0,9123	183270000	0,9541	16,4474	0,9233	8,6126	0,9642	26,75	0,9872	
	633580000	0,9508	242700000	0,9488	15,3043	0,9651	14,6094	0,9567	9,393	0,9637	
	425440000	0,9456	243880000	0,9779	18,3750	0,9450	13,3926	0,9251	14,73	0,9259	
	474160000	0,9491	193420000	0,9454	22,4189	0,9239	7,9620	0,9854	17,20	0,9585	
	413180000	0,9698	276840000	0,9338	20,1159	0,9443	8,3077	0,9694	6,805	0,9412	
	501870000	0,9229	240080000	0,9243	27,8072	0,9532	7,5147	0,9363	7,089	0,9741	
	606830000	0,9464	147940000	0,9456	11,4553	0,9411	7,5576	0,9256	8,293	0,9962	
	482570000	0,9831	209110000	0,9787	12,9001	0,9333	6,2697	0,9744	9,860	0,9637	
	630300000	0,9262	238610000	0,9951	18,0967	0,9520	15,7332	0,9412	6,648	0,9257	
	330630000	1,0112	180270000	0,965	15,7088	0,9123	12,7033	0,936	7,553	0,9144	
	ORT	503065000	0,9517	21561200	0,9568	17,8629	0,9393	10,2662	0,9515	11,432	0,95506

Yine aynı şekilde, Çizelge 2.2 G_0 değerinin 1 ve 10 aralığında sonuç değerinin dokuz basamaklı büyük değerler oldukları, görülmektedir. 100 değerinde sonuç değerlerinin iki basamaklı sonuçlara düştüğü, 10000 sütunlarında sonuçların bir önceki sütuna göre değişikliğin fazla olmadığı tespit edilmiştir. Bu iki tablodan yola çıkılarak G_0 değerinin 100 ile 1000 aralığında en iyi sonuçlar verdiği görülmüştür.

Çizelge 2.3 G_0 değerinin 1 ve 10 aralığında sonuç değerinin tek basamaklı negatif değerle de gezindiği, görülmektedir. 100 ve 1000 değerlerinde sonuç değerlerinin iki basamaklı negatif değerler olarak bu sütunlarda daha küçük değeri bulmuştur. G_0 değerinin 100 ile 1000 aralığında en iyi sonuçlar verdiği ve bütün değerler için optimizasyon süresine herhangi bir etkisinin olmadığı tespit edilmiştir.

Çizelge 2.3. F23 için G_0 başlangıç değerinin YAA'ya etkisi.

G_0 yerçekimi sabitinin YAA'ya etkisinin incelenmesi										
K.F.	$G_0=1$		$G_0=10$		$G_0=100$		$G_0=1000$		$G_0=10000$	
	En İyi	G.	En İyi	G.	En İyi	G.	En İyi	G.	En İyi	G.
	Zaman		Zaman		Zaman		Zaman		Zaman	
F23	-1,234	0,903	-10,536	0,890	-10,536	0,892	-10,536	0,907	-10,536	0,922
	-1,488	0,896	-2,421	0,893	-2,427	0,898	-10,536	0,988	-10,536	0,892
	-1,517	0,898	-10,536	0,882	-10,536	0,914	-10,536	0,908	-10,536	0,898
	-1,677	0,901	-2,871	0,892	-10,536	0,918	-2,427	0,895	-10,536	0,908
	-1,429	0,886	-10,536	0,886	-10,536	0,897	-10,536	0,925	-2,806	0,893
	-1,869	0,903	-10,536	0,890	-10,536	0,912	-10,536	0,902	-10,536	0,885
	-0,958	0,894	-10,536	0,883	-2,421	0,891	-10,536	0,894	-10,536	0,891
	-2,347	0,889	-10,536	0,894	-10,536	0,936	-10,536	0,896	-10,536	0,893
	-1,326	0,891	-2,427	0,893	-10,536	0,891	-2,871	0,887	-10,536	0,896
	-1,077	0,896	-2,806	0,891	-10,536	0,905	-10,536	0,902	-5,128	0,884
ORT	-1,492	0,8957	-7,3741	0,8894	-8,9136	0,9054	-8,9586	0,9104	-9,2222	0,8962

Standart YAA yerçekimi parametresinin başlangıç değeri olan G_0 parametresi özellikle aramanın başlangıç evrelerinde yüksek verilmesi aramanın performansını olumlu yönde etkilemiştir, 100-1000 bandı bu konuda önerilen en iyi aralık olarak tespit edilmiştir.

2.2.2. YAA α Parametresinin $G(t)$ Üzerindeki Etkisinin İncelenmesi

Arama kabiliyetini önemli ölçüde etkileyen yerçekimi parametresi içerisinde, G_0 yerçekimi başlangıç değeri, döngü sayısı ve α azalma oranı mevcuttur. Denklem (2.17)'deki α azalma oranı başlangıç döngülerinde çok küçük negatif üstel etkidir. Ancak döngü sayısı artıkça bu negatif etki döngü sayısına bağlı olarak büyür ve tesiri artar. Bunun sonucu olarak $G(t)$ döngüler ilerledikçe azalma gösterecektir. Başka bir

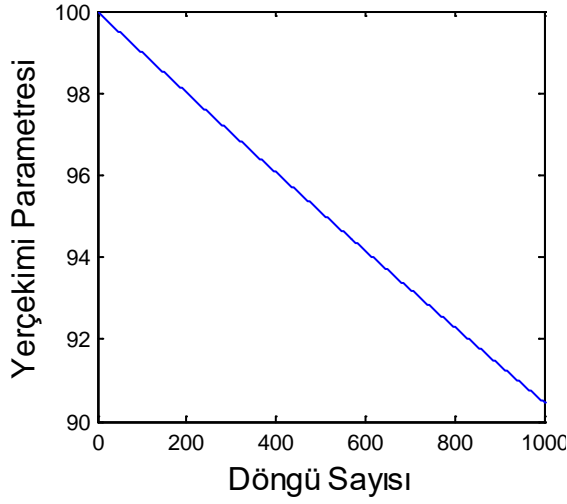
deyişle α alfa deęeri $G(t)$ azalma oranını ve řeklini belirlemektedir [85].

Bu uygulama evresinde ama fonksiyonu olarak Ek-1'de yer alan F2 fonksiyonu üzerinde alıřılmıştır. YAA ierisindeki $G_0=100$, $N=30$, boyut=30, dngü=1000, deęiřkenleri en uygun deęerler olarak seilmiş olup, deney sresince deęiřtirilmemiřtir. Btn bu parametreler sabit tutulmuř, sadece α sabiti 0,1, 1, 10, 20 ve 100 beř farklı deęer zerinden deęiřtirilerek sonu deęerleri veri olarak kaydedilmiřtir.

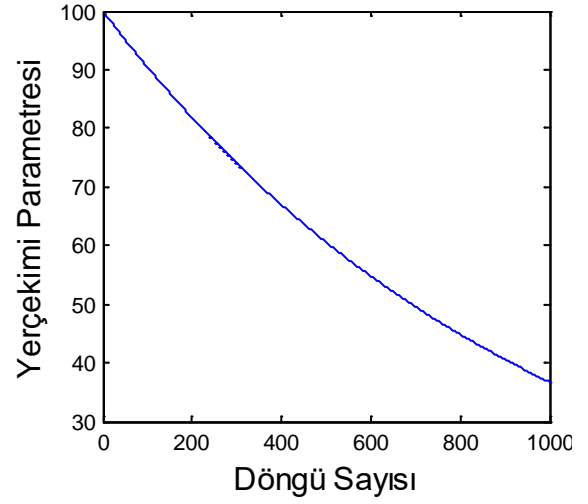
izelge 2.4. F2 iin α parametresi etkisinin gzlemlenmesi.

α sabitinin YAA'ya etkisinin incelenmesi					
	$\alpha=0.1$	$\alpha=1$	$\alpha=10$	$\alpha=20$	$\alpha=100$
Kul.Fonk.	En İyi	En İyi	En İyi	En İyi	En İyi
F2	10,8687	6,6946	1,0361	7,8784	29,9182
	12,1940	5,3318	0,0072	8,1611	21,5727
	11,2834	6,4826	0,4127	6,4330	26,3746
	17,5827	6,4696	0,2189	6,0756	30,3449
	12,0754	7,2976	0,5373	4,1290	29,4277
	14,5740	7,6511	0,2787	7,4551	20,8522
	15,5809	6,3697	0,0873	7,1319	26,4088
	13,4361	7,2839	0,8654	6,9822	27,9537
	11,3460	7,1696	0,8355	9,3590	28,0720
	13,6863	5,9329	0,1353	9,8424	29,6315
ORT	13,26275	6,66834	0,44144	7,34477	27,05563

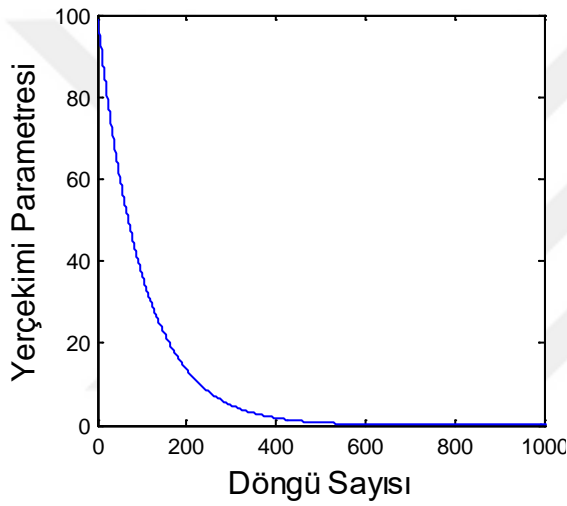
izelge 2.4'de alıřma sreleri yaklařık benzer olduęu iin optimizasyon iřlem sresine yer verilmemiřtir. α parametresinin 10 olduęu stunda en iyi sonular elde edildięi grlmektedir. Azalma oranını daha fazla artırıldıęı 100 deęerlerinde sonuların bozulduęu son stn da verilmiřtir. Sonu olarak α azalma oranını parametresini byk vermek sakınılması gereken bir durumdur. řekil 2.4 (a) ve (b)'de α parametresinin ok kk verilmesi yerekimi parametresinin her dngde azalma oranının yavař olması ile sonulandıęını ve bununla optimizasyon algoritmasının en iyi sonuları bulamadıęı gzlemlenmiřtir. 10 ve 20 tercihlerinde yerekimi parametresinin 100 deęerinden sıfıra doęru yumuřak ve emberimsi azaldıęı řekil 2.4 (c) ve řekil 2.4 (d) belirlenmiřtir ve bu azalma eęrileri altında en iyi sonu elde edilmiřtir. Yine farklı olarak α parametresini 100 ve daha byk deęerler verildięinde yerekimi parametresinin her dngde azalma oranının ok hızlı dřtę řekil 2.4 (e)'de grlmektedir. Ancak bu durum optimizasyon sonularına iyi bir řekilde yansımamıřtır. Yani $G(t)$ deęerinin her dngdeki azalma oranını ok artırılması ve ok erken dnglerde hızla azalması ve aramanın kt sonulandırılmasına neden olmaktadır.



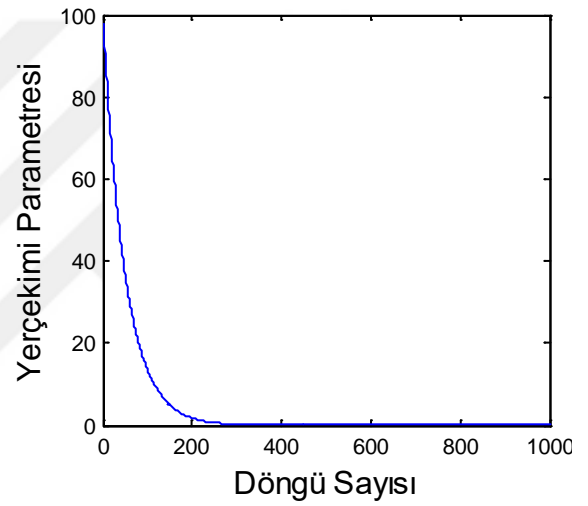
(a)



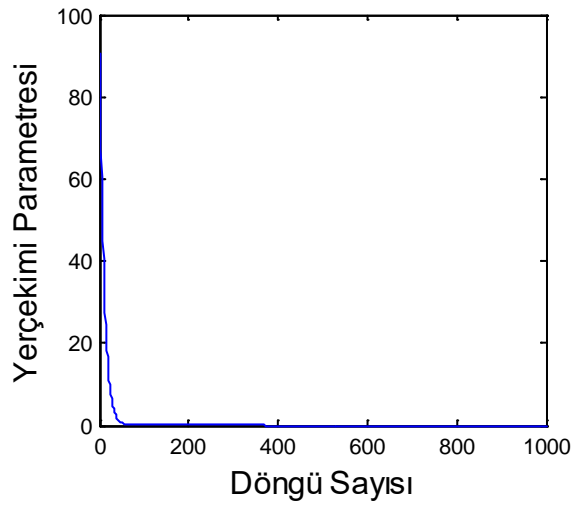
(b)



(c)



(d)



(e)

Şekil 2.4. YAA α parametresinin yerçekimi parametresi üzerine etkisi.

(a) $\alpha=0,1$ (b) $\alpha=1$ (c) $\alpha=10$ (d) $\alpha=20$ (e) $\alpha=100$

Aramanın başlangıç evrelerindeki döngüler içerisinde α 'yı çok küçük vermek zorunda olduğu tecrübe ile tespit edilmiştir. Çünkü, büyük miktarda hızlanma ve kuvvet uygulayabilmek için $G(t)$ değerinin yüksek olmasına ihtiyaç vardır. Böylece algoritmanın ortaya çıkartılmasına yardımcı olur. Son döngü içinde ise α değerini yüksek verme zorunluluğu vardır. Çünkü bu durum, $G(t)$ değerinin çok küçük seçilmesi anlamına gelip, algoritmada hassas arama özelliğini artıracaktır. α değerinin 10-20 bandında tutulması sonucu çıkarılmıştır.

2.2.3. Ajan Sayısının YAA Üzerindeki Etkisinin İncelenmesi

Bu gözlem aşamasında, daha önce tespit edilen en iyi G_0 performansı olan 100 ile en iyi α sabitinin 10 değerleri altında Ek-1'deki F3 ve Ek-3'deki F15 test fonksiyonları sırasıyla amaç fonksiyonu olarak kullanılmıştır. Ajan sayısı 1, 10, 30, 100 ve 500 değerleri kullanılarak her birinden 10'ar ölçüm alınmıştır.

Çizelge 2.5. F3 için ajan sayısının YAA ile ilişkisi.

Ajan sayısı ile YAA ilişkisinin incelenmesi										
K.F.	Ajan=1		Ajan=10		Ajan=30		Ajan=100		Ajan=500	
	En İyi	G. Zaman	En İyi	G. Zaman	En İyi	G. Zaman	En İyi	G. Zaman	En İyi	G. Zaman
F3	802540	0,2056	21253	0,2818	3449	0,7307	546	4,8894	44	102,1659
	404680	0,1784	25782	0,2828	5028	0,7450	636	4,8516	49	102,1873
	1871200	0,1740	85953	0,2810	2781	0,7455	766	4,8061	105	102,7401
	1751200	0,1767	106790	0,2800	1203	0,7432	899	4,7934	72	103,0766
	1253100	0,1755	127520	0,2781	2884	0,7453	894	4,7697	65	102,5409
	1574000	0,1951	180670	0,2790	2204	0,7412	455	4,8234	94	102,9326
	3626000	0,1748	248060	0,2794	3982	0,7422	347	4,8551	58	102,3102
	4340000	0,1755	143910	0,2801	2807	0,7308	499	4,8420	61	103,3496
	2971800	0,1735	143910	0,2776	3613	0,7532	310	4,7705	53	102,3004
	9540900	0,1743	143910	0,2844	8982	0,7420	323	4,8349	46	102,6885
ORT	2813542	0,1803	122775,8	0,2804	3693,3	0,7419	567,5	4,8236	64,7	102,6292

Çizelge 2.5 ve Çizelge 2.6'da görüldüğü gibi, küçük ajan sayılarında aramayı düzgün yapmadığı en iyi sonuçların çok büyük değerler olduğu tespit edilmiştir. Bu negatif özelliğe karşın optimizasyon süresinin çok kısa olduğu dikkat çekmiştir. Bu duruma aramanın yerel en küçük değerde takıldığı ifade edilir. Çizelge 2.6'da ajan sayısının 30 dan sonra en iyi sonuçları üzerinde bir değişiklik yapmadığı ve yüksek ajan sayısı vermenin bir anlamı olmadığı tespit edilmiştir. Ayrıca çok yüksek ajan sayısı vermek süreyi de büyük oranda artırmıştır.

Çizelge 2.6. F15 için ajan sayısının YAA ile ilişkisi.

Ajan sayısı ile YAA ilişkisinin incelenmesi										
K.F.	Ajan=1		Ajan=10		Ajan=30		Ajan=100		Ajan=500	
	En İyi	G. Zaman	En İyi	G. Zaman	En İyi	G. Zaman	En İyi	G. Zaman	En İyi	G. Zaman
F15	222	0,1672	0,0015	0,2129	0,0011	0,4852	0,0048	3,1105	0,0013	68,6254
	337	0,1668	0,0015	0,2191	0,0011	0,4877	0,0012	3,1200	0,0014	68,1531
	16	0,1688	0,0017	0,2124	0,0014	0,4721	0,0030	3,1180	0,0031	68,3778
	15	0,1679	0,0015	0,2121	0,0016	0,4838	0,0033	3,0998	0,0041	69,3491
	563	0,1683	0,0019	0,2128	0,0005	0,4784	0,0010	3,0646	0,0013	61,4685
	1607	0,1686	0,0009	0,2150	0,0008	0,4738	0,0038	3,0951	0,0010	67,3567
	177	0,1712	0,0006	0,2143	0,0014	0,4732	0,0022	3,0998	0,0017	68,2909
	521	0,1735	0,0008	0,2133	0,0012	0,4691	0,0007	3,1027	0,0014	69,7412
	24	0,1698	0,0013	0,2149	0,0012	0,4701	0,0022	3,0918	0,0011	68,2178
	198	0,1732	0,0007	0,2165	0,0013	0,4742	0,0005	3,0693	0,0015	67,9813
ORT	368	0,1695	0,0012	0,21433	0,0011	0,47676	0,0022	3,0971	0,0017	67,7561

Orta değerli aşan sayılarında, 30 ve 100 gibi kayda değer en iyi sonuçlar elde edilmesinin yanında geçen süreninde normal olduğu söylenebilir. Ajan sayısının 500 gibi büyük değerler verilmesi durumunda ise geçen sürenin çok büyük olduğu görülmektedir. Sonuç olarak ajan sayısının çok az verilmesi yerele takılma, çok büyük değerler verilmesi de optimizasyon süresinin çok uzun olması ile sonuçlanmaktadır. 30 ile 100 aralığında değerler vermek, en iyi sonuç ve zaman dengesini sağladığı düşünülmektedir.

2.2.4. Döngü Sayısının YAA Üzerindeki Etkisinin İncelenmesi

Bu bölümdeki uygulama safhasında, daha önce tespit edilen en iyi G_0 performansı 100, en iyi α sabitinin 10 ve en iyi ajan sayısı 30 değerleri sabit tutularak döngü sayısının YAA üzerindeki etkisi incelenmiştir. Bunun için Ek-1'deki F4 ve Ek-3'deki F21 test fonksiyonları kullanılmıştır. Çizelge 2.7 ve Çizelge 2.8'de elde edilen uygulama sonuçlarına göre en iyi sonuçların döngü 500 de elde edildiği, geçen sürenin de bir önceki döngü sayısına göre dört kat artışı tespit edilmiştir.

İki tabloda dikkatli incelendiğinde döngü sayısının 500'ün üzerinde verilmesinin uygun olacağı kanısına varılmıştır. Kullanıcının geçen zaman parametresini düşünerek uygulamalarda kısa zamanda sonuca ulaşmak isterse 500, geçen sürenin kullanıcı tarafından önemi olmayan uygulamalarda ise 500'den daha yüksek değerleri seçmesinin doğru olacağı sonucuna varılmıştır.

Çizelge 2.7. F4 için döngü sayısının YAA ile ilişkisi.

Döngü sayısı ile YAA algoritmasının ilişkisinin incelenmesi										
Kul.Fonk.	Döngü=1		Döngü=10		Döngü=50		Döngü=100		Döngü=500	
	En İyi	G. Zaman	En İyi	G. Zaman	En İyi	G. Zaman	En İyi	G. Zaman	En İyi	G. Zaman
F4	90	0,1711	76	0,2035	26	0,3704	17	0,5669	2,000	2,1008
	81	0,1621	81	0,2003	24	0,3689	17	0,5605	0,758	2,1366
	83	0,1639	77	0,2038	22	0,3586	18	0,5591	0,538	2,1267
	91	0,1521	77	0,2042	20	0,3565	11	0,5598	0,002	2,1874
	82	0,1632	73	0,2001	19	0,3603	19	0,5698	2,008	2,1454
	90	0,1705	76	0,1996	24	0,4001	11	0,5566	2,730	2,1329
	86	0,1787	77	0,1991	18	0,3735	15	0,5603	2,104	2,1365
	89	0,1661	70	0,2030	20	0,3598	11	0,5591	3,517	2,1456
	87	0,1621	70	0,1983	21	0,3696	11	0,5583	2,585	2,1607
	94	0,1631	78	0,2016	19	0,3605	17	0,5677	0,952	2,1441
ORT	87,3	0,16529	75,5	0,20135	21,3	0,36782	14,7	0,56181	1,7194	2,14167

Çizelge 2.8. F21 için döngü sayısının YAA ile ilişkisi.

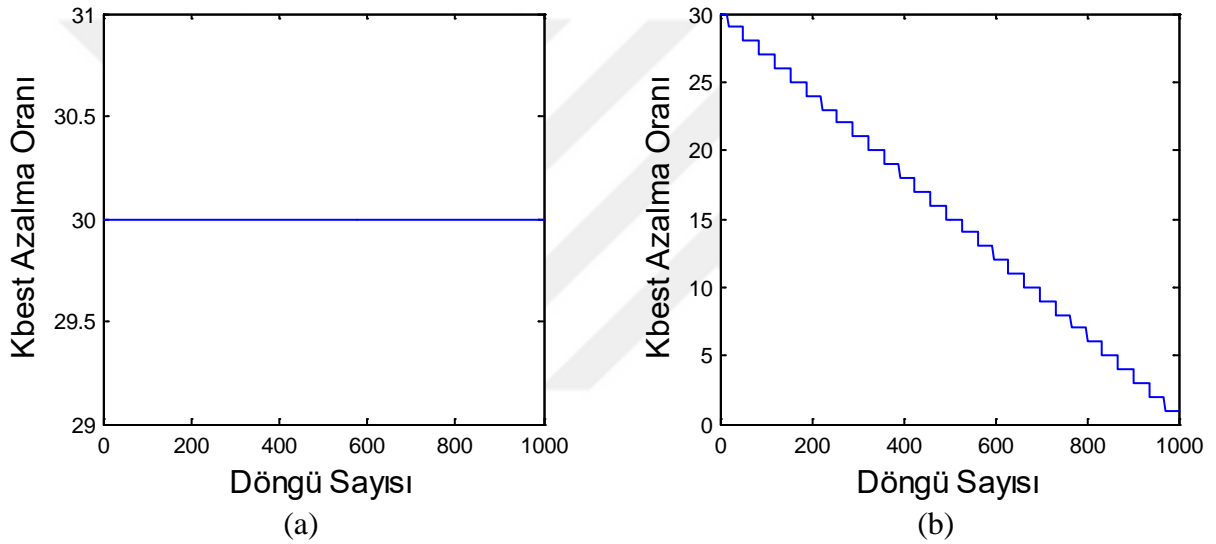
Döngü sayısı ile YAA algoritmasının ilişkisinin incelenmesi											
K.F.	Döngü=1		Döngü=10		Döngü=50		Döngü=100		Döngü=500		
	En İyi	G. Zaman	En İyi	G. Zaman	En İyi	G. Zaman	En İyi	G. Zaman	En İyi	G. Zaman	
F21	-0,867	0,4856	-1,680	0,1915	-2,628	0,3224	-10,153	0,4751	-10,153	1,7490	
	-0,337	0,1849	-0,659	0,1866	-10,153	0,3154	-10,153	0,4739	-10,153	1,7436	
	-0,082	0,1716	-1,302	0,1857	-10,153	0,3156	-10,153	0,4756	-5,055	1,7518	
	-0,114	0,1719	-2,515	0,1891	-10,153	0,3143	-10,153	0,4744	-10,153	1,7497	
	-0,622	0,1824	-2,261	0,1882	-5,055	0,3153	-10,153	0,4787	-10,153	1,7530	
	-0,428	0,1545	-2,005	0,1861	-10,153	0,3179	-2,630	0,4838	-10,153	1,7542	
	-0,359	0,1590	-1,107	0,1891	-10,153	0,3358	-2,682	0,4812	-10,153	1,7654	
	-0,571	0,1569	-2,427	0,1888	-5,055	0,3218	-10,153	0,4794	-10,153	1,7423	
	-0,362	0,1605	-1,270	0,1878	-2,301	0,3258	-10,153	0,4741	-2,682	1,7651	
	-0,245	0,1596	-2,727	0,1927	-2,630	0,3178	-2,630	0,4777	-5,108	1,7620	
	ORT	-0,398	0,1986	-1,795	0,1885	-6,843	0,3202	-7,901	0,4773	-8,391	1,7536

2.2.5. Kbest Ajanlarının YAA Üzerindeki Etkisinin İncelenmesi

YAA algoritmasında ajanlara etkiyen toplam kuvvet hesabı iki şekilde gerçekleştirilir. Birincisinde Rashedi Denklem (2.12)'deki gibi bütün ajanların i ajanına etkiyen kuvvetlerinin toplamları şeklinde ifade etmiştir. İkincisinde, Denklem (2.25)'de belirtildiği gibi kuvvet hesabına girecek ajan sayısı her döngüde azaltılmıştır. Böylece arama ve sömürüyü kontrol ederek, YAA performansı artırılmak istenmiştir. Ajanların azaltma oranını belirleyen $Kbest$ değişkeni zamanın bir fonksiyonudur. Başlangıçta

bütün ajanlar ilgili ajana kuvvet uygulayacak ve zaman geçtikçe her döngüde K_{best} doğrusal olarak azalacaktır. En son döngüde programda belirtilecek son_ajan kadar ajan, ilgili ajana etkiyecektir.

YAA da $G_0=100$, $\alpha=10$, Döngü=100, $N=30$, boyut=30 değişkenleri kendileri için en uygun değerler olarak seçilmiş olup, deney süresince değiştirilmemiştir. Uygulama çalışması Ek-1'de yer verilen F1 ve F2 fonksiyonlarına uygulanmıştır. Öncelikle her döngüde ilgili ajana etkiyen kuvvetler hesaplanırken bütün ajanlar dikkate alınmıştır. İkinci aşamada K_{best} ajan sayısından başlatılmış, her döngüde azaltılarak son döngüde $son_ajan = 2$ olana kadar düşürülmüştür.



Şekil 2.5. Her döngüde kuvvete tesir eden ajanların gösterimi.

(a) Bütün ajanlar (b) K_{best} ajanlar

Şekil 2.5. (a) Denklem (2.11)'in YAA da uygulanması üzerine oluşan eğridir ve ilgili ajana bütün ajanların her döngüde etkilediği görülmektedir. Şekil 2.5. (b) ise Denklem(2.25'in algoritmada uygulanması sonucu oluşan K_{best} eğrisidir ve her döngü içerisinde ajan sayısının azaldığını görülmektedir. Son döngüde ajan sayısını küçük bir değerde kaldığı görülmektedir.

Çizelge 2.9 ve Çizelge 2.10 incelendiğinde K_{best} kullanılması durumunda optimizasyon süresi yaklaşık olarak yarıya düştüğü görülmektedir. Bu önemli katkının yanında en iyi sonuçların K_{best} kullanılarak elde edildiği özellikle F2 fonksiyonunda tespit edilmiştir. Ancak yinede yerel minimumdan kaçınmadığı durumlarda gerçekleşmiştir. Sonuç

olarak kütlesi büyük ajanların seçimini sağlayan *Kbest* kullanımını süre ve en iyi sonuç bakımından tercih edilmelidir.

Çizelge 2.9. F1 için *Kbest* ajan sayısının YAA ile ilişkisi.

<i>Kbest</i> YAA ilişkisinin incelenmesi				
Kul.Fonk.	Her	Ajan	Her döngüde <i>Kbest</i>	
	döngüde Sayısı En İyi	G. Zaman	En İyi	G. Zaman
F1	1361	0,8811	489	0,5593
	523	0,8936	1023	0,5629
	840	0,8883	1001	0,5589
	849	0,9047	1218	0,5683
	719	0,8927	268	0,5678
	1245	0,8855	1133	0,5809
	1966	0,8867	1202	0,5602
	1264	0,8837	751	0,5697
	863	0,8935	1093	0,5708
	969	0,8887	360	0,5629
ORT	1059,9	0,8898	853,8	0,5661

Çizelge 2.10. F2 için *Kbest* ajan sayısının YAA ile ilişkisi.

<i>Kbest</i> YAA ilişkisinin incelenmesi				
Kul.Fonk.	Her	Ajan	Her döngüde <i>Kbest</i>	
	döngüde Sayısı En İyi	G. Zaman	En İyi	G. Zaman
F2	0,914	0,9067	2,256	0,5761
	2,223	0,8975	1,143	0,5687
	2,417	0,9033	0,022	0,5760
	2,340	0,8892	0,758	0,5726
	1,125	0,8979	0,005	0,5729
	3,185	0,8983	2,039	0,5665
	0,333	0,8966	0,440	0,5644
	1,074	0,8903	0,064	0,5670
	0,822	0,8856	0,714	0,5764
	2,951	0,8923	1,657	0,5737
ORT	1,7384	0,89577	0,9098	0,57143

2.2.6. YAA *son_ajan* Değerinin YAA Üzerindeki Etkisinin İncelenmesi

İlgili ajana bütün ajanların etkidiği kuvveti katmak yerine, her döngüde ajan sayısını düşürerek kütlesi fazla olan ajanlar etkime işlemini gerçekleştiren *Kbest* içerisinde *son_ajan* adında bir parametre mevcuttur. Bu parametre aramanın son döngüsünde ilgili ajanın kuvvet hesabı için katılması gereken ajan sayısı olarak tanımlanır. Bu uygulama

safhasında *son_ajan* parametresinin YAA algoritmasına etkisi olup olmadığı üzerine inceleme yapılmıştır.

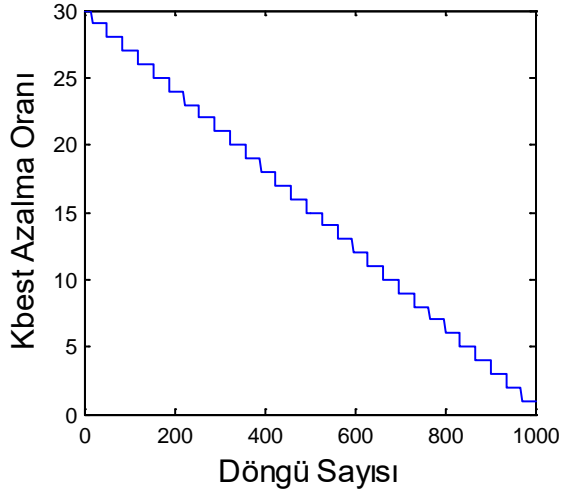
Çizelge 2.11. F3 için *son_ajan* ile YAA ile ilişkisi.

<i>son_ajan</i> değerinin YAA ile ilişkisinin incelenmesi					
	<i>son_ajan</i> =2	<i>son_ajan</i> =10	<i>son_ajan</i> =30	<i>son_ajan</i> =50	<i>son_ajan</i> =80
Kul.Fonk.	En İyi	En İyi	En İyi	En İyi	En İyi
F3	6531	3298	1571	5330	2644
	4210	2506	2932	2891	2526
	2003	2852	3806	2276	2940
	2287	2056	1777	2495	1771
	1860	1941	2181	2373	3180
	2469	5364	3313	1018	2732
	2035	2909	1894	2454	2082
	3731	3301	2199	3088	3643
	2091	2150	3613	1996	2939
	2525	2059	2233	4468	1870
ORT	2974,2	2843,6	2551,9	2838,9	2632,7

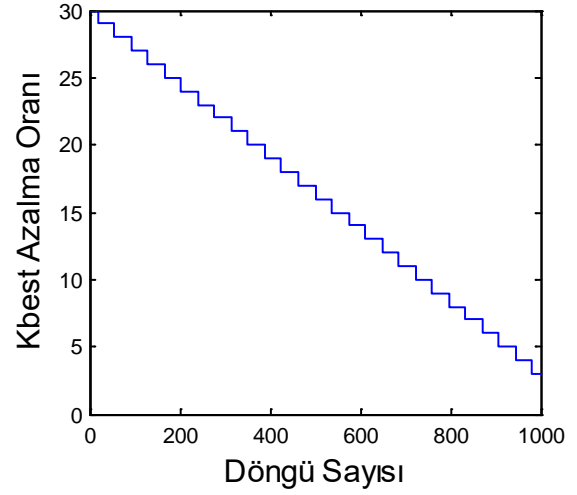
Uygulama da Ek-1'deki F3 fonksiyonu amaç fonksiyonu olarak seçilmiş ve *son_ajan* sabiti %2, 10, 30, 50 ve 80 vererek en iyi sonuçlara ve döngülerin en iyilerine etkisine bakılmıştır. YAA Algoritmasında $G_0=100$, $\alpha=10$, döngü=100, $N=30$, boyut=30 değişkenleri kendileri için en uygun değerler olarak seçilmiş olup, uygulama süresince değiştirilmemiştir.

Çizelge 2.11'den çıkarılan sonuç, çeşitli *son_ajan* katsayıları YAA içerisinde en iyi sonuçları etkilemediği görülmüştür. Beş sütun da benzer sonuçlar elde edilmiştir. Ancak algoritma son döngülerde sadece kütlesi çok büyük olan ajanların çok sayıda devreye sokulması, ajanların hızını etkileyeceğinden hassas arama özelliğini bozar. Ayrıca ajanların kuvvet hesabında hesaplama artacağından işlem süresini de artırır. Bundan dolayı, son döngülerde hassas ince arama yapmak ve işlem süresini düşürmek için *son_ajan* yüzde değerlerini küçük vermek gerekir.

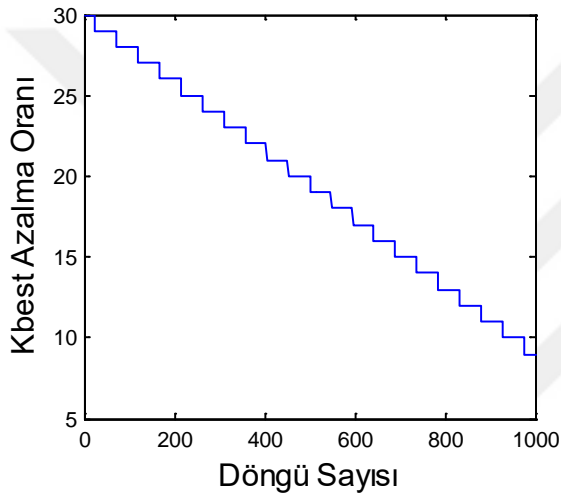
Şekil 2.6'da *Kbest* yaklaşımında son döngüdeki *son_ajan* yüzdesine göre ajan kalarak aramayı bitiren grafikler verilmiştir. Şekil 2.6 (a) ve (b) bu uygulama safhası için önerilen aralıklardır. Şekil 2.6 (d) ve (e) ise *Kbest* kullanım etkisini azaltan ve hesaplama güçlüğü problemini sağlayan değerlerin grafikleridir.



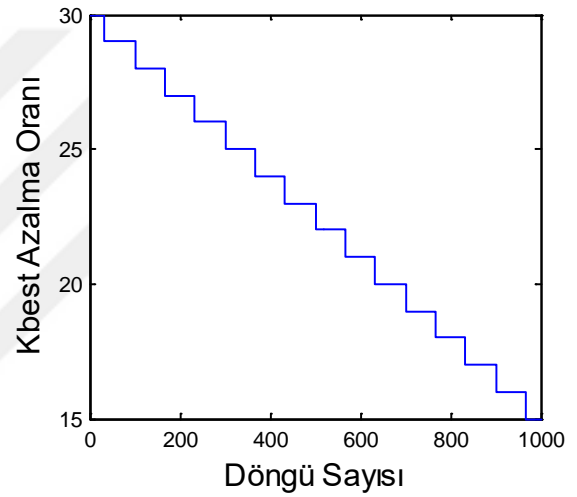
(a)



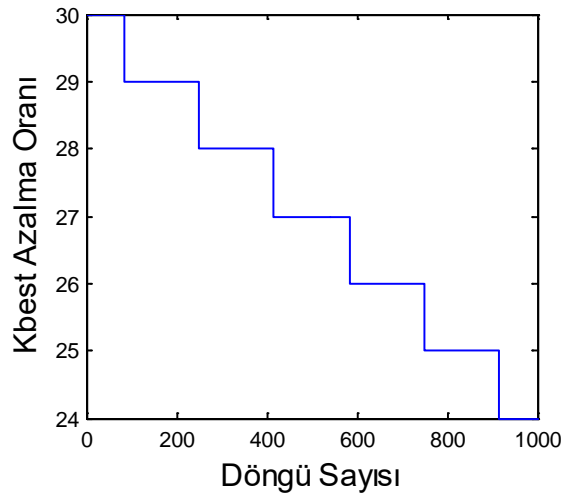
(b)



(c)



(d)



(e)

Şekil 2.6. Her son döngüde kullanıla *son_ajan* grafikleri.

(a) $son_ajan = \%2$ (b) $son_ajan = \%10$ (c) $son_ajan = \%30$ (d) $son_ajan = \%50$

(e) $son_ajan = \%80$

2.2.7. Kbest Farklı Azalma Eğrileri Altında, YAA'ya Etkisinin İncelenmesi

Her döngüde ilgili ajana etkiyen K_{best} ajan sayısı referans makalede doğrusal olarak azaltılmıştır [4]. Bununla beraber iki farklı azalma yaparak YAA üzerindeki değişimler incelenmiştir. İlk fonksiyonda, azalma oranını yerel arama bölümlerinde yavaş tutup, ajan sayısı diğer iki eğriye göre daha fazladır. Son döngüye yaklaşırken küresel aramalarda azalma oranı hızlandırılmış ajan sayısı düşürülmüştür. Bu davranışı sağlayabilmek için K_{best} Denklem (2.27) numaralı fonksiyondan faydalanılmıştır.

$$K_{best} = a + (K_1 - son_ajan) \cdot \frac{1}{b + e^{(-\beta_1 \frac{t}{T})^2}} \quad (2.27)$$

Eşitlikteki a, b ve K_1 sabit olup azalmanın oranının önce yavaş son döngülerde hızlı olması için verilen sabitler olup, sırasıyla 4, 9 ve 1000 değerlerine sahiptirler. Son döngüdeki ilgili ajana etkiyen ajan sayısı son_ajan ve değeri 2 alınmıştır. Ayrıca, β_1 değeri üstel fonksiyonun azalma katsayısıdır ve 3 dür [85].

İkinci fonksiyon olarak bir öncekinin tam tersine yerel aramalarda ajan sayısını hızla düşürerek, en iyi sonuca yaklaşırken azalma oranı da yavaşlatılmıştır. Son döngüye yaklaşırken ilgili ajana etkiyen ajan sayılarının diğer fonksiyonlara göre burada daha fazla olduğu söylenebilir. Bu davranışı sağlayabilmek için K_{best} Denklem (2.28)'deki fonksiyondan faydalanılmıştır [85].

$$K_{best} = (K_2 - son_ajan) \cdot e^{(-\beta_2 \frac{t}{T})} \quad (2.28)$$

Eşitlikte K_2 sabiti uygulamalarda 100 olarak alınmıştır. β_2 değeri üstel fonksiyonun azalma eğrisini oluşturmaktadır ve uygulama safhasında 3, son_ajan değeri olarak da 2 verilmiştir.

Denklem (2.27), Denklem (2.28) ve standart uygulama olan Denklem (2.26)'da elde edilen sayısal ondalıklı değerler matlab ortamında en yakın tamsayı ajan sayısına dönüştürmek için Denklem (2.29)'da ki eşitlik kullanılmıştır.

$$K_{best} = round(N \cdot \frac{K_{best}}{K_3}) \quad (2.29)$$

Denklem (2.29)'da ki K3 tamsayıya dönüştürme sabitidir ve 100 alınmıştır. N ise arama içerisinde kullanılan ajan sayısıdır.

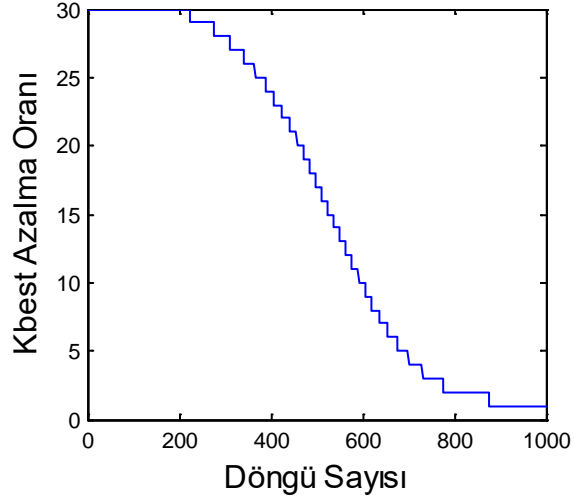
Ek-2'deki F10 ve F13 fonksiyonları uygulama için amaç fonksiyonudur. YAA içerisinde $G_0=100$, $\alpha=10$, döngü=100, N=30, boyut=30 değişkenleri kendileri için en uygun değerler olarak seçilmiş olup, uygulama süresince değiştirilmemiştir. Burada ilgili ajan kuvvet hesabına girecek ajan sayısını belirleyen Denklem (2.27), (2.28) ve (2.26) sırasıyla uygulanmıştır.

Çizelge 2.12. F10 için K_{best} farklı azalma fonksiyonlarındaki davranışının incelenmesi.

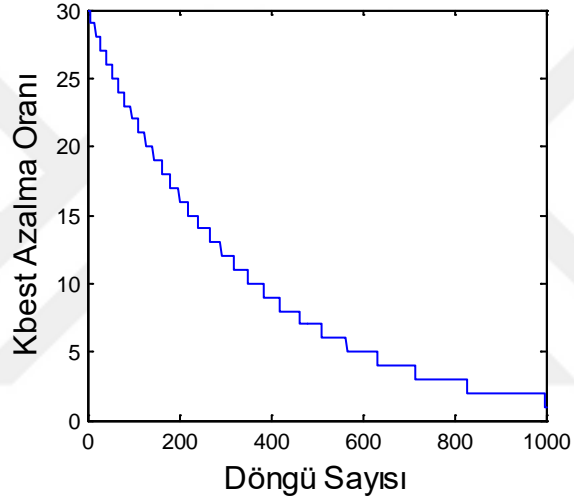
<i>K_{best} farklı azalma fonksiyonlarındaki davranışının incelenmesi</i>			
Kul.Fonk.	Önce azalma oranı yavaş daha sonra hızlı	Önce azalma oranı hızlı daha sonra yavaş	Doğrusal azalma oranı her döngüde eşit
	En İyi	En İyi	En İyi
F10	2,744	2,579	1,899
	2,413	1,155	1,501
	2,070	1,501	2,738
	1,901	2,221	2,220
	2,238	0,064	1,501
	2,922	0,680	0,931
	0,948	1,899	1,906
	3,258	1,501	1,646
	3,361	1,155	1,899
	3,409	2,408	0,934
ORT	2,526	1,516	1,718

Çizelge 2.13. F13 için K_{best} farklı azalma fonksiyonlarındaki davranışının incelenmesi.

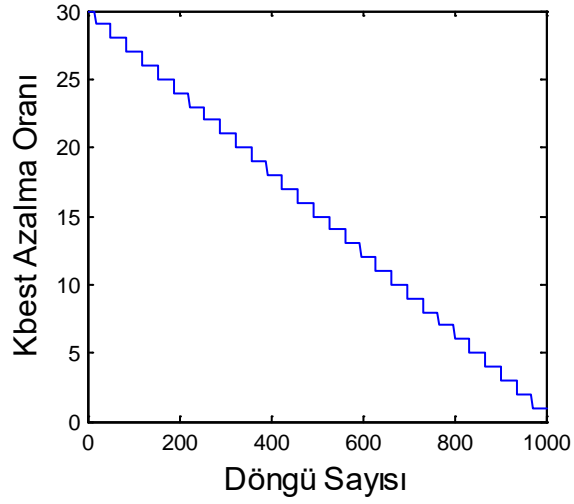
<i>K_{best} farklı azalma fonksiyonlarındaki davranışının incelenmesi</i>			
Kul.Fonk.	Önce azalma oranı yavaş daha sonra hızlı	Önce azalma oranı hızlı daha sonra yavaş	Doğrusal azalma oranı her döngüde eşit
	En İyi	En İyi	En İyi
F13	38,322	15,179	41,306
	44,385	34,277	37,731
	42,711	29,533	48,766
	39,881	37,299	60,684
	35,676	34,989	51,994
	40,936	41,727	24,855
	41,709	22,774	40,861
	39,789	18,488	33,596
	103,306	31,306	23,482
	40,286	22,102	40,011
ORT	46,700	28,767	40,329



(a)



(b)



(c)

Şekil 2.7. Farklı *Kbest* fonksiyonlarının azalma grafikleri.

(a) Azalma oranı önce yavaş sonra hızlı (b) Azalma oranı önce hızlı sonra yavaş

(c) Standart uygulama azalma oranı doğrusal

Yapılan uygulama sonucunda Çizelge 2.12 ve Çizelge 2.13 sırasıyla F10 ve F13 fonksiyonları için sonuçlar alınmıştır. İki test fonksiyonunda da önce azalma oranı hızlı daha sonra yavaş olan Denklem (2.28)'deki fonksiyon da daha iyi sonuçlar elde edilmiştir. Özellikle F13 fonksiyonunda standart YAA uygulamasına göre büyük fark oluşturmuştur. Bir sonraki sıralamayı doğrusal azalma oranı her döngüde eşit olan standart YAA almıştır.

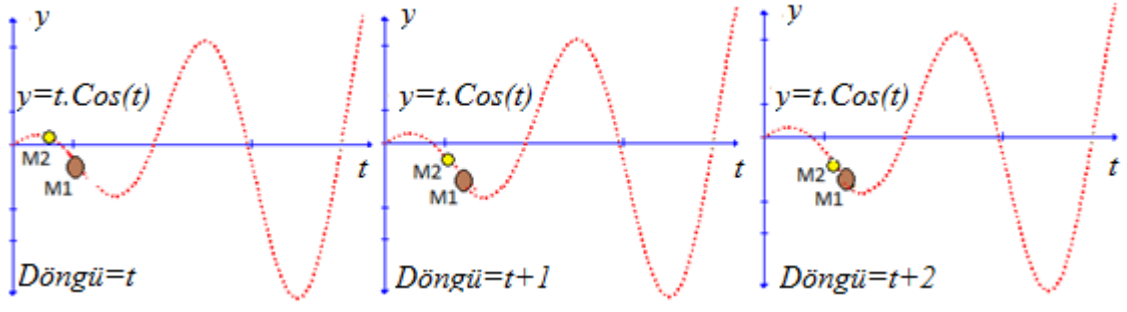
Şekil 2.7'de görüldüğü gibi ilgili ajana etkiyen 30 ajan sayısının her döngüde azaldığı ve azalma oranlarının döngünün başlangıcında ve sonlarında farklılık gösterdiği görülmektedir. Denklem (2.27)'deki fonksiyon Şekil 2.7 (a)'da, Denklem (2.28)'deki fonksiyon grafiği Şekil 2.7 (b)'de, Denklem (2.26) çıkışı olan standart YAA uygulaması Şekil 2.7 (c) de verilmiştir. Şekil 2.7 (b)'nin diğerlerine göre önemli ölçüde YAA'yı olumlu ölçüde etkilemiştir. Şekil 2.7 (c) en yakın tamsayı değerleri ve her döngüde farklı olduğu için grafikler testere formunu almıştır.

Küresel aramada ve son döngülerde daha hassas sonuçlar yakalayabilmek için ilgili ajana etkiyen ajan sayısını yavaş değiştirmek gerekliliği sonuç olarak çıkmıştır. Ayrıca ilk döngülerde ajan sayısı diğer ikisine göre daha az olduğu için süre bakımında ve kaba arama özelliği bakımından dinamik bir davranış kazanmıştır. Bu inceleme ve sonuç geliştirme aşamasında dikkate alınması gereken bir özelliktir.

2.3. YAA İÇİN GELİŞTİRİLEN OPERATÖRLER

2.3.1. Kaotik Sarsıntılı Operatör

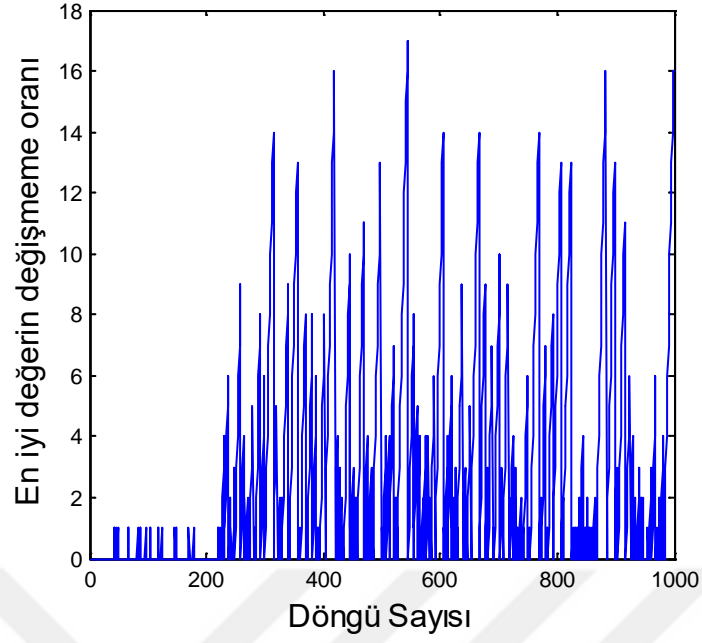
Bu tez çalışmasında geliştirilen kaotik sarsıntılı operatörün YAA uygulanması sonucu elde edilen yeni yöntem Kaotik Sarsıntılı Yerçekimi Arama Algoritması (KSYAA) olarak adlandırılmıştır. YAA içerisinde yerçekimi parametresi $G(t)$, arama uzayı içerisindeki ajanların yerlerini değiştirmede kullanılan, ajanların hızını belirleyen önemli bir değişkendir. Denklem (2.8)'de ajana etkiyen kuvvet formülünde yer alır, ancak etkisi aramada önemli yer tutan ajan hızlarına kadar tesir eder. Algoritmanın başlangıç döngülerinde, yerçekimi yüksek değerlerde olup, yerçekimi kuvvetlerinin de şiddetlerinin yüksek olması ile sonuçlanır. Bu da başlangıç döngülerde ajanların hızlı hareketi ile sonuçlanır. Aramanın ilk evreleri için ajanların hızlı hareketi istenen bir durumdur.



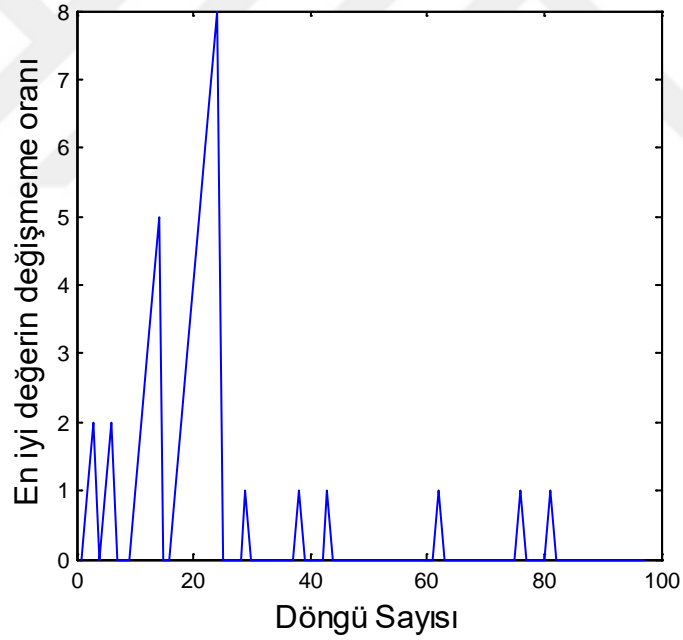
Şekil 2.8. Yerel minimuma yaklaşma ve takılma durumu.

Denklem (2.17)'deki azalan fonksiyona göre $G(t)$ değeri giderek azalır, ajanların hızı da yavaşlar ve ağır ajanların çekim kuvveti nedeniyle ağır ajanların çevresinde yavaş hareketler ile kümeleşme durumu gerçekleşir. Bunun için de hassas arama durumu denir ve son döngülerde oluşması sağlanır. Aramanın başlamasından belli bir döngü sonra yerel minimuma takılma durumu gerçekleşmiş ise son döngülere yaklaşırken ajanlar bu bölgeden çıkamaz ve arama kötü değerler ile sonuçlanır. YAA için geçmiş yıllarda yapılan çalışmaların çoğunda erken döngülerde yerel minimuma takılma ve sonra son döngülere doğru ajan hızlarının yavaşlamasında dolayı yerel minimum noktasından çıkamama davranışları bu algoritmanın dezavantajlarından biri olarak dile getirilmiştir [86-89].

Şekil 2.8'de YAA algoritmasının yerel minimuma niçin çok takılma durumu ile karşılaştığını basit olarak $y = t \cdot \text{Cos}(t)$ bir boyutlu amaç fonksiyonu üzerinde anlatılmağa çalışılmıştır. Sistemde iki ajanın arama yaptığını ve ilk döngülerdeki yüksek $G(t)$ değerlerinin bittiğini aramanın orta evrelerine geldiğini varsayalım. Kütleleri küçük olan ajanlar az da olsa ağır ajanları kendisine doğru çektiği için ağır ajanın yerel en küçük değere yaklaşmasını önler. M1 ajanın kütlesi yüksek olup M2 ajanını kendisine doğru çekmektedir ve ağırlığından dolayı yavaş hareket etmektedir. M2 de M1 az da olsa çektiğinden yerel en küçük değere yaklaşmasını önlemektedir. $G(t)$ değerinin azalmasından dolayı ajanların hızları ve pozisyonlarının değişim miktarları azalmıştır. Bundan dolayı bu iki ajan bu fonksiyonun sağ tarafında yer alan küresel minimum değerini yakalaması için gerekli hız oluşmaz ve yerel minimum noktalarını çevresinde arama yapar. Bu dezavantajından dolayı son döngülerde yerel minimumdan çıkamadığı için küresel en iyi değeri yakalayamaz.



(a)



(b)

Şekil 2.9. F11 için küresel en iyi değerin değışmeme durumunun gözlemlenmesi.

(a) Döngü=1000 de değışmeme durumu (b) Döngü=100 değışmeme durumu

YAA'nın arama süreci içerisinde yerel minimuma takılma durumunun nasıl tespit edileceği de problemin çözümü kadar önemlidir. Bunun için özellikle her döngü için bulunan en iyi sonuç değerinin bir önceki ile karşılaştırılması yapılarak arama süreci

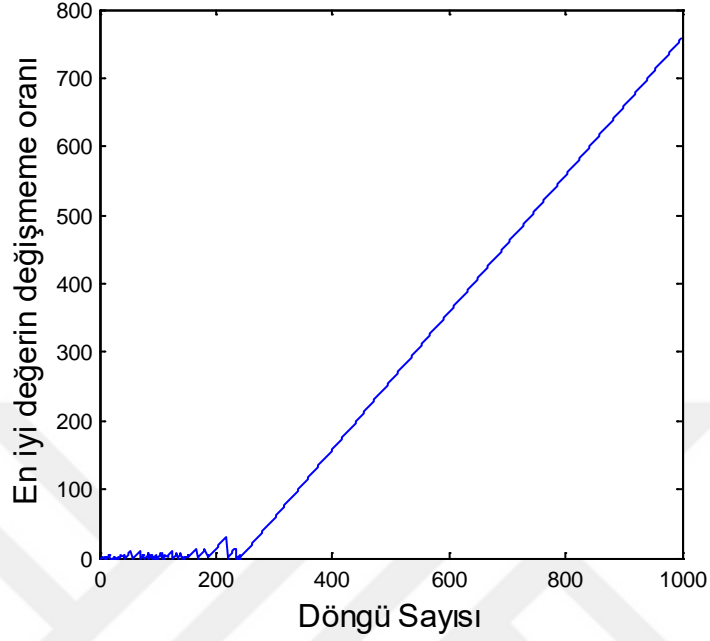
hakkında çıkarım yapılabilir. Özellikle aramanın başlangıç ve orta evrelerinde en iyi sonuç değeri döngü sayısı arttıkça değişmiyor ise yerel en küçük değere takılma durumu ortaya çıkmış veya arama kötü bölgelerde gerçekleşiyor sonucu çıkartılır. Daha sonra ajanların yerçekimi parametresinin azalmasından dolayı hızları yavaşladığı için yerel minimumdan kurtulması zorlaşmaktadır. Algoritmada her döngüde ajanların yeni pozisyonlarını belirlemek için yerçekimi parametresini zamana bağlı üstel fonksiyon ile her döngüde azalan şekilde belirlendiği daha önceki bölümde belirtilmiştir. Ancak bu değişim aramanın bu bölgeden kurtulmasına engel olamayabilir.

Şekil 2.9'da Ek-2'deki F11 fonksiyonu için çalışılmış olup, yukarıda belirtilen arama süreci içerisinde yerel minimuma takılma veya kötü arama durumunun grafiksel olarak gözlemlenmesine yer verilmiştir. $G_0=100$, $\alpha=10$, döngü=100, $N=30$, boyut=30 değişkenleri kendileri için en uygun değerler olarak seçilmiştir ve deney süresince değiştirilmemiştir. Belli döngü aralıklarında en iyi değerde hiç değişim olmadığı görülmektedir. Örneğin Şekil 2.9 (a) 1000 döngü çalıştırılması durumunda 20 kez en 15 döngü boyunca en iyi değer değişmediği görülmektedir. Şekil 2.9. (b) de 100 döngü için yapılan çalışmada 30-40, 45-60 65-80 ve 85-100 aralıklarında her döngü için en iyi değer 15 döngü boyunca hiç değişmemiştir. Bu durum için 100 döngülük bir aramada 30 döngüde yerel minimuma takılma veya kötü bölge de arama olduğu söylenir.

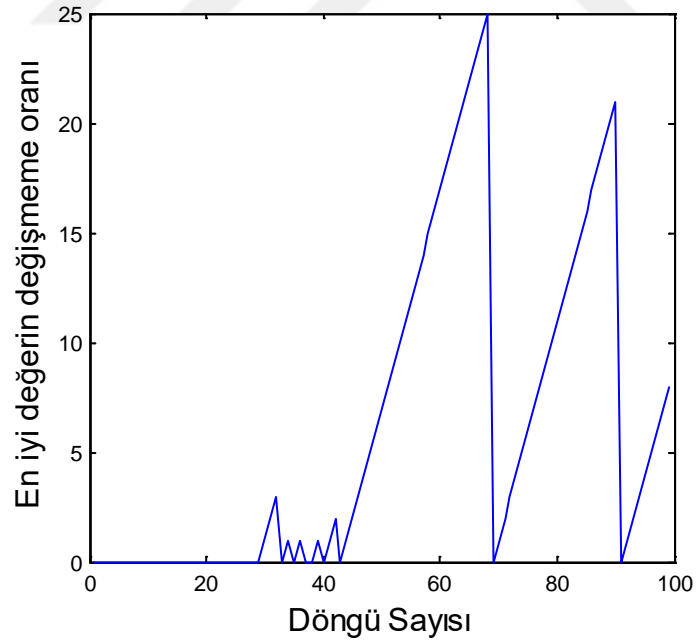
Aynı şekilde Şekil 2.10 (a) da Ek-1'deki F6 fonksiyonu 1000 döngü için çalıştırılmıştır. En iyi değer 250. döngüden sonra hiç değişmediği, yani YAA algoritmasının 750 döngü boyunca yerel minimum noktasından çıkamadığı görülmüştür. YAA'nın dezavantajı F6 fonksiyonu için bu eğri davranışında açıkça belirlenmiştir. Aynı fonksiyon bu kez Şekil 2.10 (b) de 100 döngü için çalıştırılmıştır. Bu çalışma da 40-70 ve 70-95 aralıklarında yerel minimumun hiç değişmediği görülmektedir.

Yukarıdaki yerel minimuma yaklaşarak orada takılmış ajanları çıkarmak ve yeni arama bölgelerine yönlendirmek için ilk önce yerel minimuma takılma durumunun tespiti gerekir. Şekil 2.11'de yer alan sözel kod da yerel minimuma takılma veya kötü bölge de arama durumunun tespiti için yapılan algoritmaya yer verilmiştir. Bu algoritmaya göre döngü içerisinde küresel en iyi değer değişmiyor ise *Takılma_say* değeri bir artar ve değişmeme durumu ardı sıra döngülerde de gerçekleşir ise *Takılma_say* değeri artmaya devam eder. Bu durumda *Takılma_say* değerine bakarak kaç döngüde değişim olmadığı, kaç döngü için yerel minimuma takıldığı görülebilir. Eğer her döngüde en iyi küresel

değere yeni atama yapılıyor ise *Takılma_say* parametresi sıfırlanmaktadır. Bunun anlamı arama istenildiği gibi olumlu seyrini devam ettiriyor demektir.



(a)



(b)

Şekil 2.10. F6 için küresel en iyi değerin değışmeme durumunun gözlemlenmesi.

(a) Döngü=1000 de değışmeme durumu (b) Döngü=100 değışmeme durumu

```

for döngü=1 den maksimum döngüye
    Uygunluk değerlerini hesapla; Uygunluk değerlerinin en küçüğünü bul best olarak yaz;
    if döngü=1 then
        best değerini küresel Fbest değerine yaz;
    end if
    if best<Fbest then (En küçük bulma işlemi için)
        Fbest:=best; Takılma_say değerini sıfırla;
    else
        Takılma_say değerini bir artır;
    end if
    Takılma_say değerini sayac dizinine kaydet;
end for

```

Şekil 2.11. Yerel minimuma takılma durumunun tespiti.

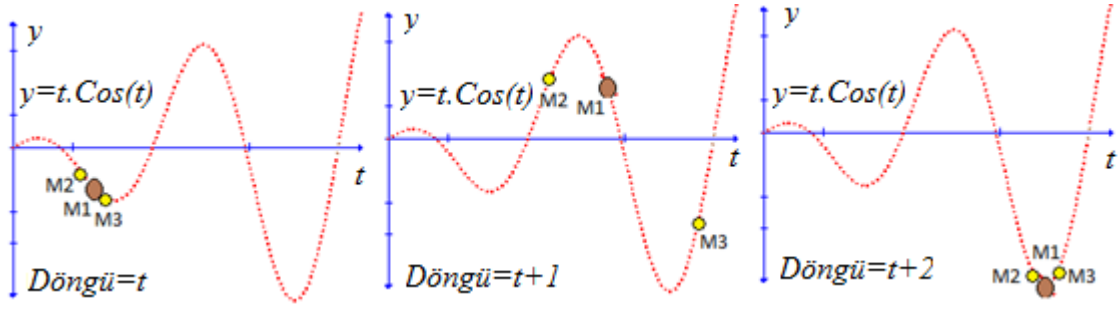
Problemin tespitini yaptıktan sonra, yerel minimumdan kaçınmak ve arama algoritmasını dinamik hale getirmek için yerçekimi sabiti üzerinde küçük kaotik sarsıntı oluşturarak, ajanların hızları ve pozisyonları değiştirilmiş olur. Şekil 2.12'de görüldüğü gibi, algoritmanın son arama evrelerinde en iyi değer her döngü için değişmesi istenmez. Çünkü son evrelerde hassas arama yapılacaktır ve en iyi değer değişmeyebilir, gayet normaldir. Bundan dolayı, son döngüler içerisinde *Takılma_say* değerinin büyümesi hususu dikkate alınmaması gerekir.

```

for döngü=1 den maksimum döngüye
    if Takılma_say >= Eşik ve Döngü <(%75*maksimum döngü ) then
        Yerçekimi sabitini Kaotik Sarsıntı yöntemi ile hesapla;
         $G(t) = Sarsıntı(Döngü) + G0 * exp(-alfa * t/T);$ 
    else
        Yerçekimi sabitini Klasik yöntem ile hesapla;
         $G(t) = G0 * exp(-alfa * t/T);$ 
    end if
end for

```

Şekil 2.12. Kaotik sarsıntı oluşturma sözel kodu.



Şekil 2.13. Kaotik sarsıntı sonucu yerel minimumdan kurtulma durumu.

Eğer döngü sayısı %75den küçük ise yani aramanın ilk ve orta evrelerinde bulunuluyorsa ve en iyi sonuç değeri Eşik sayısı kadar değişmemiş ise kaotik sarsıntı oluşturulur. Dolayısıyla, aramayı farklı bölgelere götürme işlemi sağlanmış olur. Eğer bu şartların dışında ise normal arama gerçekleşiyor var olan durumu devam ettirilir. Yerel minimuma takılma durumunda ve kötü arama gerçekleştiğinde kaotik sarsıntı oluşturularak bu durumdan kurtulmak için gerekli ajan hızları sağlanmış olacaktır. Problemin çözümü $y = t.Cos(t)$ bir boyutlu uygunluk fonksiyonu üzerinde Şekil 2.13'de canlandırılmıştır. Hatırlanırsa yerel minimuma takılmış ve ajanların hızları orta döngülerde yerel minimumdan çıkabilecek büyüklükte değil idi. Kaotik sarsıntı oluşmuş $G(t)$ ve ajan hızları büyüyerek sağ tarafta yer alan küresel minimum değeri yakalanmıştır.

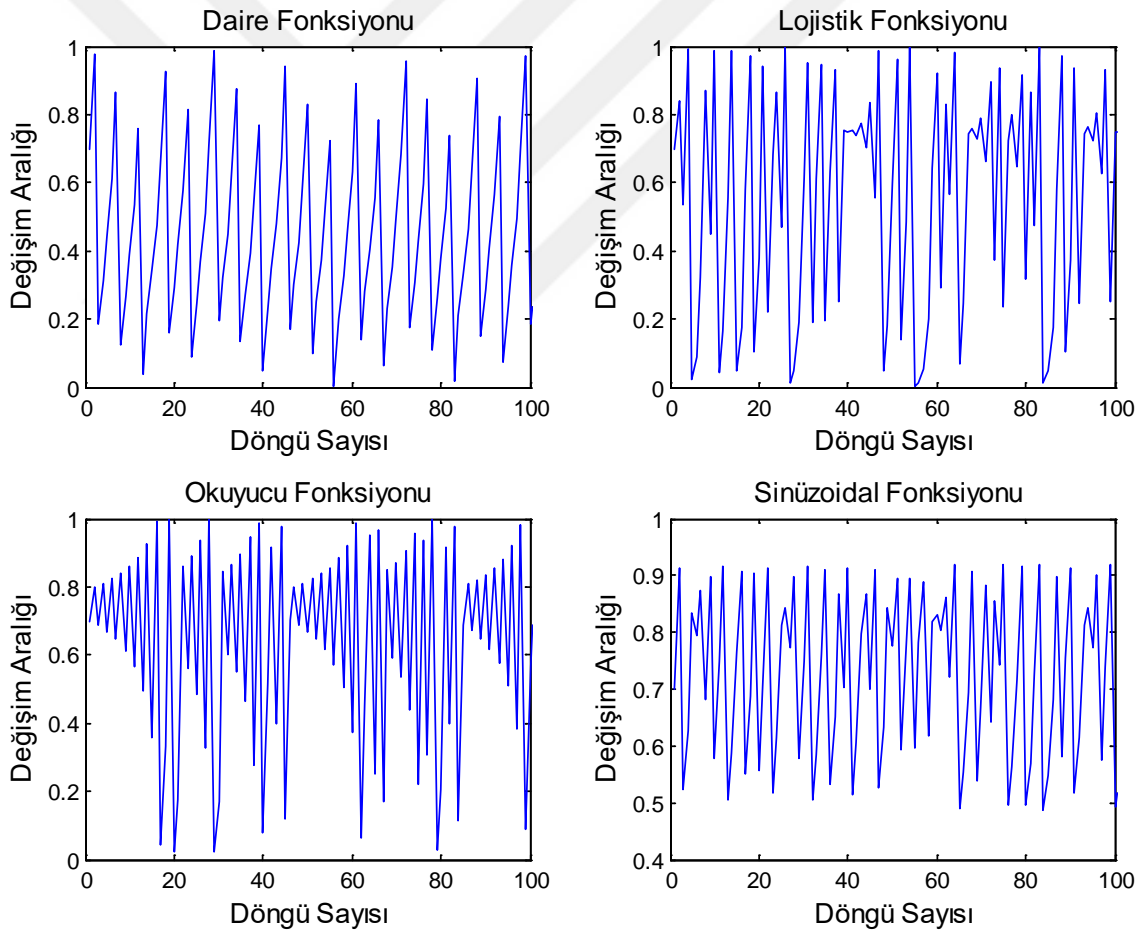
Kaotik doğrusal olmayan deterministik karmaşık davranışlar olarak tanımlanır [90]. Bir kaotik işlemin tanımlaması ilk olarak Lorenz tarafından 1963 yılında yapılmış olup, güneş desenlerinin modellenmek için Lorenz Çezp Edicisi olarak bilinen doğrusal olmayan diferansiyel eşitlik sistemlerinde kullanılmıştır [91]. Sinyal işleme, Kaos kontrolü, güvenlik iletişimi gibi bazı yeni mühendislik uygulamalarında literatür içinde kullanılmıştır [92-94].

En uygun değerinin değişmemesi durumundaki kaotik sarsıntı uygulanacağı döngüde Denklem (2.17)'deki yerçekimi sabitini ifadesi yerine Denklem (2.30) uygulanır.

$$G(t) = Sarsıntı + G_0 \cdot \exp\left(-\alpha \cdot \frac{t}{T}\right) \quad (2.30)$$

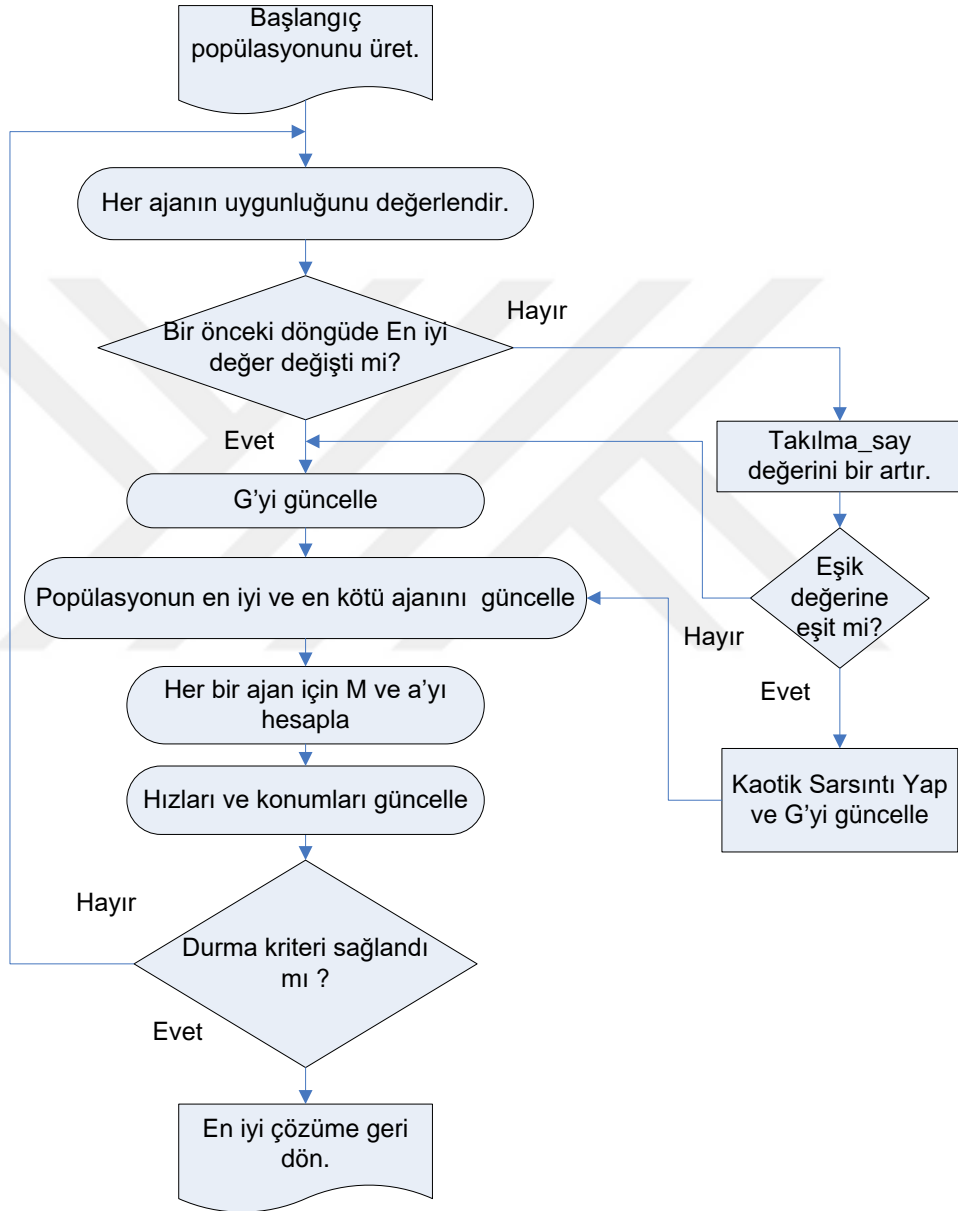
Çizelge 2.14. Algoritmada kullanılan kaotik fonksiyonlar.

NO	ADI	KAOTİK HARİTA	ARALIK
1	Daire	$X_{i+1} = \text{mod} \left(X_i + b - \left(\frac{a}{2\pi} \right) \sin(2\pi X_k), 1 \right)$ $a = 0.5 \text{ and } b = 0.2$	(0,1)
2	Lojistik	$X_{i+1} = aX_i(1 - X_i), \quad a = 4$	(0,1)
3	Okuyucu	$X_{i+1} = \mu(7.86X_i - 23.31X_i^2 + 28.75X_i^3 - 13.302875X_i^4)$ $\mu = 1.07$	(0,1)
4	Sinüzoidal	$X_{i+1} = aX_i^2 \sin(\pi X_i), \quad a = 2.3$	(0,1)



Şekil 2.14. Kaotik haritaların görünüşleri.

Yukarıdaki Denklem (2.30)'da "*Sarsıntı*" olarak adlandırılan ve çalışma içerisinde kullanılan kaotik haritalar Çizelge 2.14 de sunulmuştur. Literatürde yaygın olarak kullanılan daire, lojistik, okuyucu ve sinüzoidal fonksiyonlar tercih edilmiştir [95-98]. Çizelge 2.14 içinde kullanılan kaotik haritalar içerisinde rastgele özelliğini sağlayan komponentler yer almamıştır.



Şekil 2.15. Kaotik sarsıntı operatörü akış şeması.

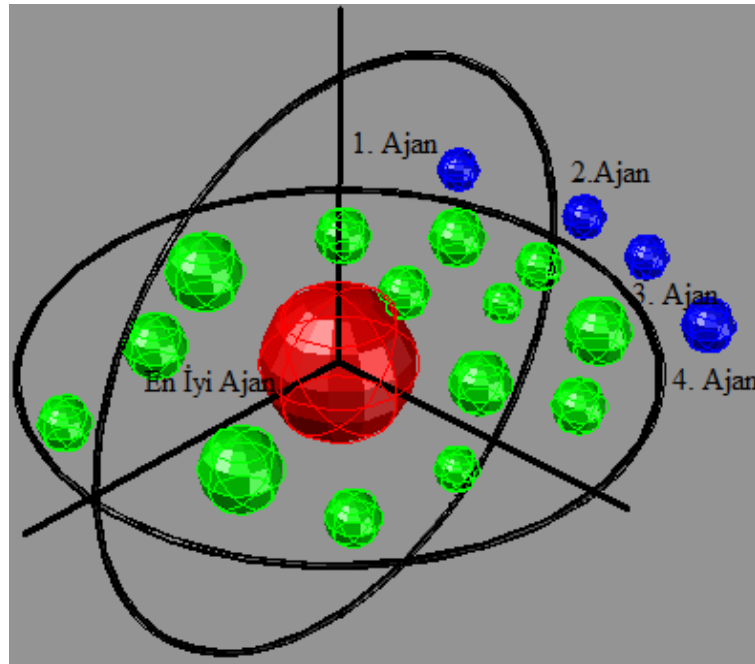
Şekil 2.14'de kaotik davranışları sağlayabilen deterministik sistemlerin çıkış grafiklerine yer verilmiştir. Şekil 2.14 içerisinde eşitliklerin kaotik davranışları oldukça

belirgindir. Kaotik haritaların kümesi başlangıç noktası tümü için 0,7 iken farklı davranışlar ile seçilebilir. Başlangıç noktası 0 ve 1 arasında herhangi bir sayı seçilebilir.

Şekil 2.15'de KSYAA operatörünün akış şemasına yer verilmiştir. Şekil 2.11 ve Şekil 2.12'de kaotik sarsıntı için sözel kodlara yer verilmiştir ve bu kodlarını standart algoritmanın hangi evrelerinde yer alacakları yukarıdaki akış şemasında belirtilmiştir. Öncelikle her döngünün başında bir önceki döngüde en iyi sonuç değeri değişip değişmediği sorgulanmaktadır. Böylece yerel minimuma takılıp takılmadığı konusunda fikir alınmaktadır. $G(t)$ 'nin güncellenmesi en iyi (*best*) ve en kötü ajanın (*worst*) güncellemelerinden ayrılmış ve bir önceki evrede yapılmıştır. *Takılma_say* değeri eşik değerden eşit ve büyük olduğu durumda kaotik sarsıntı oluşturulur, $G(t)$ güncellenir. Daha sonra standart YAA da ki temel akış içerisinde kaldığı yerden devam eder.

2.3.2. Kurtulma Hız Operatörü

Bu tez çalışmasında geliştirilen kurtulma hız operatörünün YAA uygulanması sonucu elde edilen yeni yöntem Kurtulma Hızlı Yerçekimi Arama Algoritması (KHYYAA) olarak adlandırılmıştır. YAA algoritmasında, ajanların hızları ve pozisyonlarını belirleyen en önemli etki eşitlik Denklem (2.11)'de belirtildiği gibi ilgili ajana etkiyen toplam kuvvettir. Kuvvet hesabında Denklem (2.25)'de verildiği üzere kütlesi büyük olanlar ve ilgili ajana mesafesi yakın olanların katkısı diğerlerine göre etkileri fazladır.

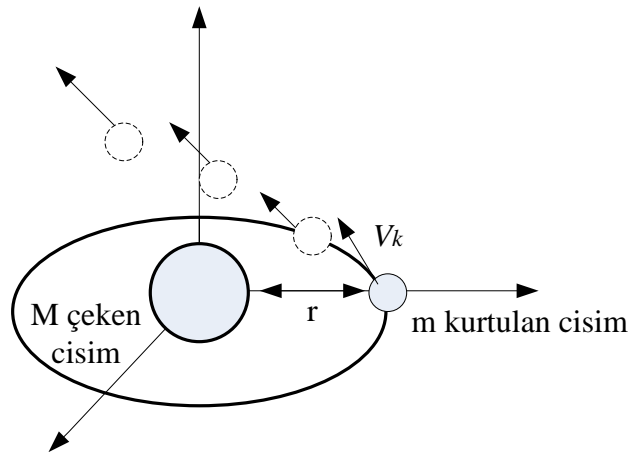


Şekil 2.16. Ajanların sembolik hareketlerinin gösterimi.

Şekil 2.16'da kütlesi büyük olan en iyi ajanın diğer ajanları çektiği ve grupsal hareket ettiklerini gösteren iki boyutlu sembolik yaklaşım verilmiştir. Bu grupsal davranışın uzak bölgelerinde yer alan dört ajanın var olduğunu varsayalım. Uzakta yer alan bu dört ajanın Denklem (2.8) ve Denklem (2.25)'e göre etkiyen toplam kuvvetleri hesaplandığında mesafeden dolayı toplam kuvvetlerinin düşük olduğu, dolayısıyla Denklem (2.13) ve Denklem (2.14)'e göre hızlarının da düşük olacağı tespit edilmiştir. Bu durumda hızlarının yavaş olmasından dolayı kötü bölgelerdeki aramalarına devam ederler, optimizasyon sonucuna katkılarını azalması anlamına gelir.

Sezgisel tekniklerin temel özelliklerinden biri olan grupsal davranış hareketi YAA da mevcuttur. Ancak ajanlar arasındaki mesafe büyükse ve kütesinin küçük olmasından dolayı grupsal davranışın dışında kalma eğilimi artar. YAA için geçmiş yıllarda yapılan çalışmaların çoğunda ajanların hız güncellemeleri ve problemin çözüm önerileri sunulmuştur [73, 99-100].

Fizikte kurtulma hızı, üç boyutlu bir uzayda bulunan cismin kendisini etkileyen kütle çekim alanından kurtulabilmesi için ulaşması gereken minimum sürati ifade eder. Örneğin bir uzay aracının dünyanın yerçekimi kuvvetinden kurtulabilmesi için ihtiyacı olan hıza denir. Kurtulma hızına sahip cisim, kendisini çekmeye çalışan cisme doğru geri düşmez veya o cisim etrafında bir yörünge üzerinde hareket etmez [101]. Tek merkezli, basit bir çekim alanından kurtulma durumunda kurtulma hızı, cismin sahip olduğu kinetik enerjinin kütle çekimsel potansiyel enerjiye (eksi) eşit olduğu andaki değeridir [102].



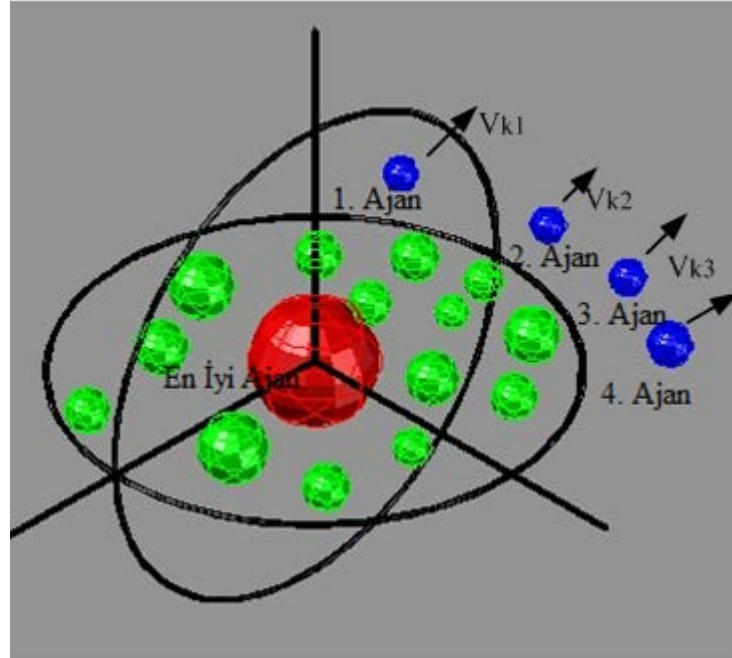
Şekil 2.17. Tek merkezli basit çekim alanındaki kurtulma hızı.

Şekil 2.17'de tek merkez olarak M cisminin çekim alanından bir m cisminin kurtulmasını sembolize etmektedir. Sembolik şekle göre herhangi bir cisim kurtulma hızına sahip olamaz ise M çekim alanından çıkamaz. Minimum gerekli olan kurtulma hızı için, kaçan cismin kinetik enerjisi çeken cismin çekimsel potansiyel enerjisine Denklem (2.31)'deki gibi eşitlenir.

$$\frac{1}{2}mV_k^2 = \frac{GMm}{r} \quad (2.31)$$

$$V_k = \sqrt{\frac{2GM}{r}} \quad (2.32)$$

Burada V_k kurtulma hızı, G kütleçekim sabitini, M kaçılan cismin kütlesi, m kaçan cismin kütlesi, r cismin merkezi ile kurtulma hızının hesaplandığı nokta arasındaki mesafeyi temsil etmektedir.

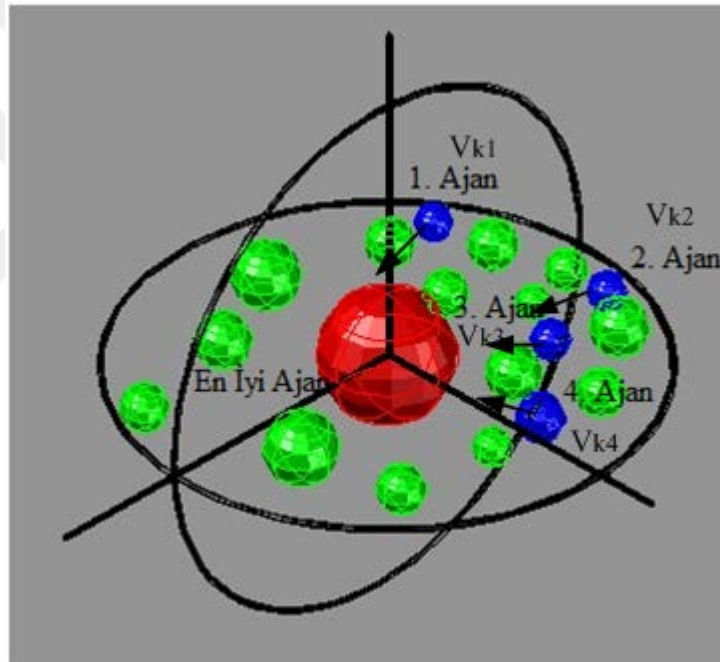


Şekil 2.18. Grupsal hareketinin dışındaki ajanlara kurtulma hızı uygulanmış gösterimi.

Denklem (2.32)'deki kurtulma hızını, Şekil 2.15'de sembolik gösterilmiş dört ajana klasik hızları ile toplamları şeklinde güncellenerek uygulandığında Şekil 2.18'de gösterildiği gibi bu ajanlar grupsal davranıştan daha fazla uzaklaşacaktır. Bu durum ise

katkılarının daha da azalması anlamına gelir. Yapılan ilk deneysel çalışmalarda pozitif katkılı veya klasik hızın hiç kullanılmadığı durumlarda standart YAA sonuçlarından daha kötü sonuçlar elde edilmiştir. Bu yüzden uzakta kalan ajanlara pozitif kurtulma hızı eklendiğinde Şekil 2.18'de ki sembolik gösterimde ki gibi grupsal davranışın dışına çıkma eğilimi artmıştır.

Grupsal davranışın dışında veya uzakta kalmış ajanların hızlarını artırmak ve grup içerisine düşürmek için yukarıdaki kurtulma hızının negatif yönde eklenmesi ile sağlanabileceği önerilmiştir. Böylece arama içerisinde sürü ve grupsal yaklaşımı mükemmelleştirme çalışması gerçekleştirilmiştir. Şekil 2.20'de önerinin sonucunun sembolik gösterimi verilmiştir.



Şekil 2.20. Dışta kalmış ajanlara negatif kurtulma hızı uygulanmış gösterimi.

Problemin çözümünde ilk önce, klasik YAA da ki K_{best} yaklaşımına benzer bir metot kullanılacaktır. Her bir döngüde uzakta kalan ajan sayısını belirleyen ve zamanla azalan h_{best} parametresi kurtulma hızına tabii olacak ajan bilgisini vermektedir. Denklem (2.32) KHYAA göre güncellenirse aşağıdaki hız denklemleri elde edilir.

$$V_k = -\sqrt{\frac{2.G(t).Mb}{MesEn}} \quad (2.33)$$

Eşitlik deki Mb , bir önceki döngüde ajanlar arasındaki en büyük kütle, $MesEn$ ise ajanların en iyi ajana olan mesafelerini gösteren büyükten küçüğe doğru sıralanmış bir diziyi temsil eder. $G(t)$ ise klasik YAA algoritmasındaki yerçekimi parametresi özelliklerini koruyarak burada da kullanılmıştır.

```

En iyi uygun değere sahip ajanı bul
for ajan=1 den maksimum ajana
    İlgili ajanın fitness değerini bul
    if Eğer ajanın fitness değeri en küçük değer ise then
        En iyi ajan değerine ata
    end if
end for
 $a_i^d(t+1) = \frac{F_i^d(t)}{M_{ii}(t)}$  İvmeyi hesapla
 $v_i^d(t+1) = rand_i * v_i^d(t) + a_i^d(t+1)$  Hız güncellemesini klasik yolla yap
Son döngüde kurtulma hızına tabii olacak ajan sayısı final_don değerini gir.
Kurtulma hızına tabii olacak hbest değerini hesapla,  $V_k$  hızlarını sıfırla
En iyi ajana olan mesafeleri büyükten küçüğe sırala ve MesEn dizisine sıra ile at
Maksimum kütle  $Mb = \max(M)$  hesapla;
for ajan=1 den maksimum ajana
    if Eğer ilgili ajan En iyi değere sahip ajan ise then
        for ajan=1 den maksimum hbest kadar
            j değerine hbest içerisindeki ajanı ata;
             $V_{kurtulma} = -\text{sqrt}((2 * G * Mb) / MesEn(1, j));$ 
             $V_k(j, :) = (rand(1, boyut) * V_{kurtulma});$ 
        end for
    end if
end for
for ajan=1 den maksimum ajana
    if Grupsal davranışın dışında kalan uzak ajan ise then
        Kurtulma hızına sahip ajanlara klasik hızı ekle  $v_i^d(t+1) = v_i^d(t+1) + V_k(t)$ 
    else
        Klasik hızı kullan  $v_i^d(t+1)$ 
    end if
end for
Ajanların pozisyonlarını güncelle  $x_i^d(t+1) = x_i^d(t) + v_i^d(t+1)$ 

```

Şekil 2.21. KHYAA sözel kodu.

Denklem (2.34)'de bu uygulamaya giren ajanların hızları klasik yolla hesaplandıktan sonra kurtulma hızları ile toplanarak son nihai hızları elde edilir. Uygulamaya girmeyen ajanlar için ise kurtulma hızı hesaplanmaz sadece Denklem (2.14)'deki klasik yol kullanılır.

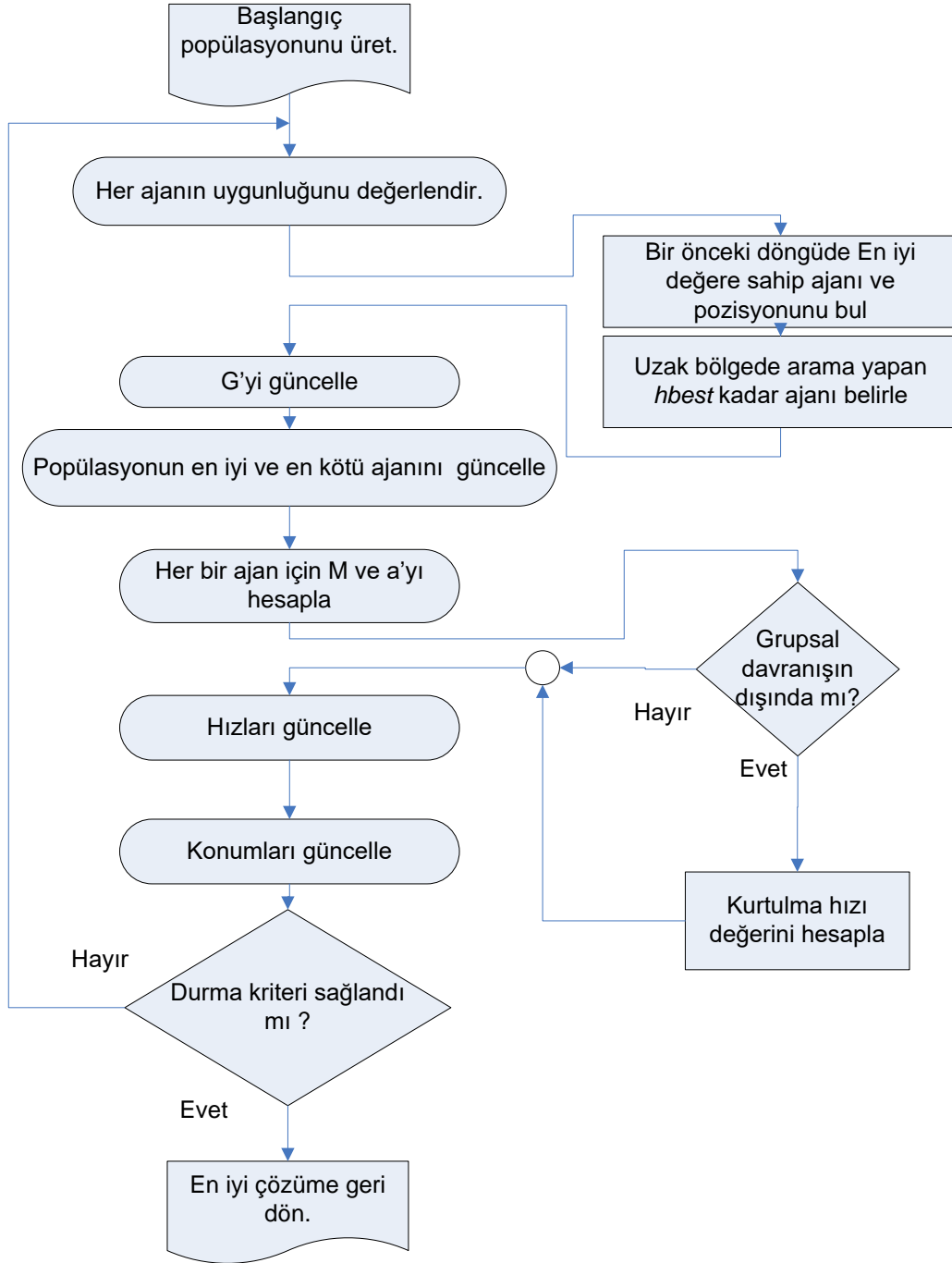
$$v_i^d(t+1) = v_i^d(t) + V_k(t) \quad (2.34)$$

KHYAA da ajanların hızları ve pozisyonları belirleyen algoritmanın sözel kodu Şekil 2.20'de verilmiştir. İlk önce *hbest* parametresi ile kurtulma hızına tabii olacak her döngüdeki ajan sayısı belirlenmiştir. Ajanlar içerisinde bir önceki döngüde en iyi ajan olan "En" ajanına sıra geldiğinde diğer ona uzak olan *hbest* sayısı kadar ajanların kurtulma hızları hesaplanmıştır. Diğer ajanlar için ise klasik hız hesaplaması yapılmıştır.

KHYAA'nın standart YAA algoritmasındaki akış da ne gibi değişiklikler yaptığını daha iyi görmek için Şekil 2.21'deki akış şemasına yer verilmiştir. Akış şemasına göre ilk döngünün başında *hbest* uzakta kalan ajan sayısı ve bir önceki döngüdeki en iyi değere sahip ajanın değeri ve diğer ajanlara olan mesafe bilgileri alınarak standart YAA ana akışına devam edilir.

Klasik YAA dan ayıran en önemli farklardan biri de hız ve pozisyon güncellemelerinin ayrı işlem bölmelerinde yapılmasıdır. Hız güncellemelerinden önce ilgili ajanın uzakta olup olmadığı sorgusu yapılır. Eğer uzakta yer alan ajan grubuna girmiyor ise klasik YAA gibi hız güncellemesi yapılır. Uzakta ajan grubuna giriyor ise Denklem (2.33) ve Denklem (2.34)'de yer alan kurtulma hızı ve klasik hıza eklenme işlemleri gerçekleştirilir. İki sınıfa ayrılmış ajan grubu pozisyon güncellemesine beraber girerek ana akışa devam edilir.

Şekil 2.21'de yer alan uzakta ajan sayısı olan *hbest* değerini çok yüksek vermek (Örneğin başlangıçta % 50 ajan) yapılan deneysel çalışmalar sonucunda olumsuz sonuçlara neden olduğu görülmüştür. Bu yüzden *hbest* değerini toplam ajan sayısının %20 sini geçmeyecek şekilde vermek yerinde olacaktır. KHYAA da başka yeni bir parametre olarak *MesEn* yer almaktadır. En iyi değere sahip ajan grupsal davranışın merkezi gibi kabul edilmiş, diğer ajanların bu ajana olan uzaklığı öklid mesafesi alınarak hesaplanmıştır. Tek boyut içerisindeki uzaklığından çok genel anlamda fikir vermesi için tercih edilmiştir.



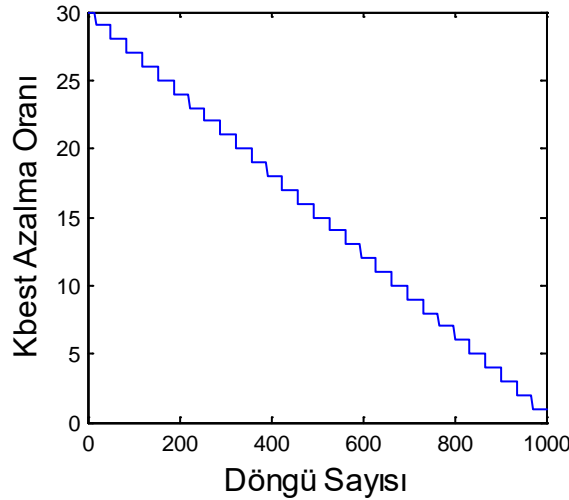
Şekil 2.21. KHYAA akış şeması.

2.3.3. En İyi Ajan Operatörü

Bu tez çalışmasında geliştirilen en iyi ajan operatörünün YAA uygulanması sonucu elde edilen yeni yöntem En İyi Ajan Yerçekimi Arama Algoritması (EİAYAA) olarak adlandırılmıştır. YAA optimizasyon algoritmasında kütleler arasında bir çekim kuvvetini esas alır ve en büyük kütleyle sahip nesne daha yavaş hareket eder. Çekim

kuvveti cisimlerin kütleleri ile doğru aralarındaki mesafenin karesi ile ters orantılıdır. Ayrıca çekim kuvvetinin yönü büyük cisme doğrudur.

Newton'un yerçekimi yasasındaki kütleyle sahip nesnelere YAA arama yapan ajanlar şeklinde dönüştürülmüştür. Her döngüde en iyi sonuç değerine sahip olan ajanın kütlesi Denklem (2.19) ve Denklem (2.20) yer göre en büyük değere sahip olur. YAA algoritmasında rastgelelik özelliği vermek için Denklem (2.8)'de d boyutu içerisindeki i ajanının üzerindeki toplam kuvvet hesaplanırken iki şekilde gerçekleştirilir. Birincisinde Rashedi, Denklem (2.11)'de görüldüğü gibi bütün ajanların i ajanına etkileyen kuvvetlerinin toplamları şeklinde ifade edilmiştir. İkincisinde Denklem (2.25)'de, her döngüde ajan sayısını azaltarak arama ve sömürüyü kontrol ederek YAA'nın performansı artırılmak istenmiştir. $Kbest$ zamanın bir fonksiyonudur, başlangıçta bütün ajanlar ilgili ajana kuvvet uygulayacak ve zaman geçtikçe her döngüde $Kbest$ doğrusal olarak azalacaktır. En son döngüde programda belirtilecek son_ajan kadar ajan, ilgili ajana etkiyecektir [4]. $Kbest$ azalma oranı aşağıdaki Denklem (2.26)'da ve Şekil 2.22'de görsel olarak gösterilmiştir. Aramanın ilk döngüsünde kuvvet hesabına bütün ajanlar katılırken, ilerleyen döngülerde kütlesi büyük ajanlardan seçilen ajanlar kuvvet hesabına girmektedir.



Şekil 2.22. $Kbest$ azalma oranı.

Denklem (2.25)'e göre, bir sonraki döngüde, bu ajana etki eden toplam kuvveti bulmak için $Kbest$ yaklaşımına göre kendisine etki eden ajanlar en iyi kütleyle sahip ajanlar arasından seçilmektedir. Buda bu çok büyük kütleli ajanın diğerleri gibi ivmesi, hızının

büyük olması ve büyük oranda pozisyon deęiřtirmesi anlamına gelir. Halbuki en iyi sonuç deęerine sahip olmasından kaynaklanan kütleinin büyük olması çok yavaş hareket etmesi gerekirdi [104].

```

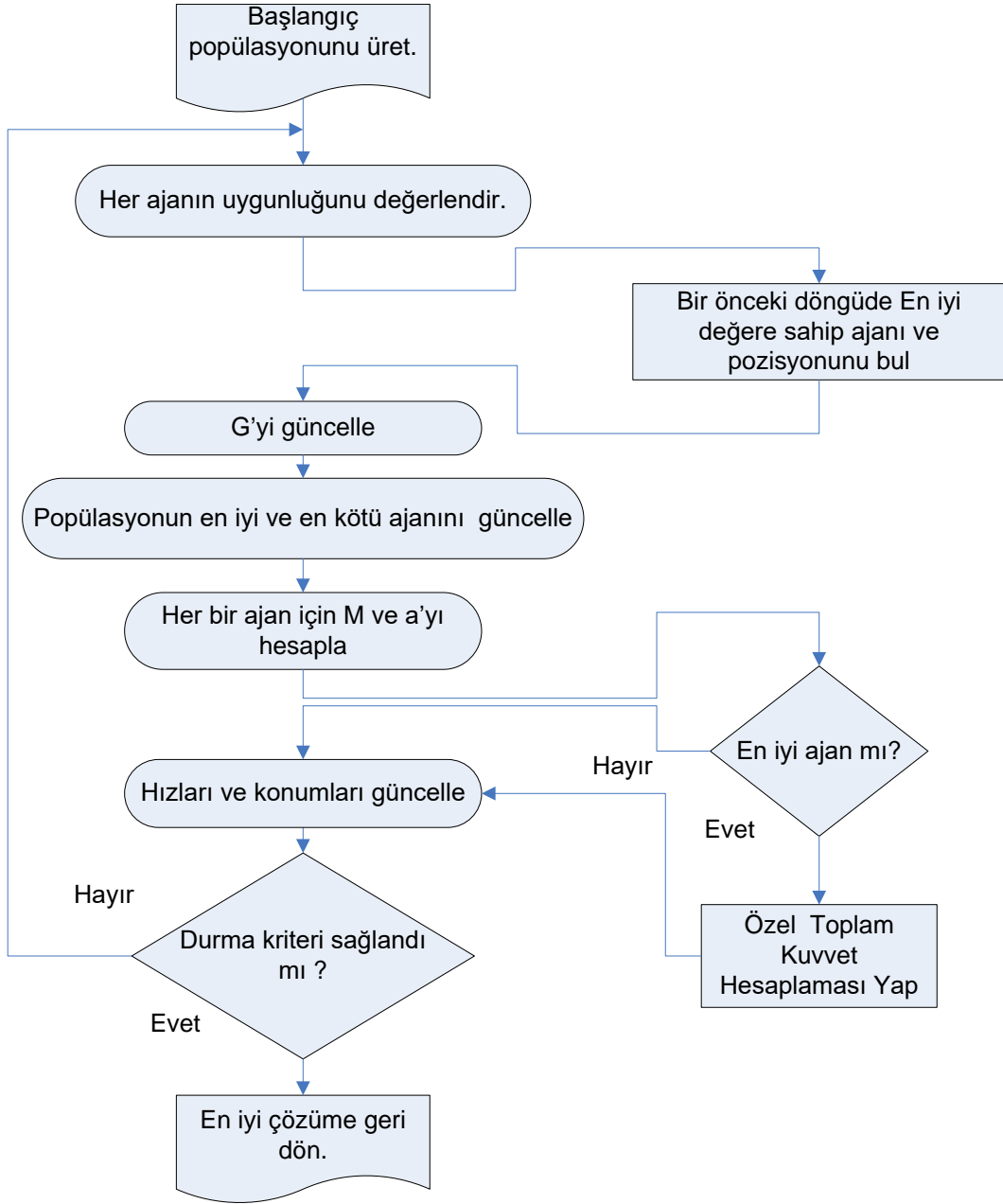
En iyi uygun deęere sahip ajanı bul
for ajan=1 den maksimum ajana
  İlgili ajanın fitness deęerini bul
  if Eđer en küçük deęer ise then
    En iyi ajan deęerine ata
  end if
end for
Bir sonraki döngüde Kbest ajan sayısını hesapla
Bir sonraki döngüde Kworst ajan sayısını hesapla.
for ajan=1 den maksimum ajana
  if Eđer ilgili ajan En iyi deęere sahip ajan ise then
    for ajan=1 den maksimum Kworst kadar
       $F_i^d(t) = \sum_{j \in Kworst, j \neq i}^N rand_j F_{ij}^d(t)$ 
    end for
  else
    for ajan=1 den maksimum Kbest kadar
       $F_i^d(t) = \sum_{j \in Kbest, j \neq i}^N rand_j F_{ij}^d(t)$ 
    end for
  end if
end for

```

Şekil 2.23. EİAYAA sözel kodu.

Yukarıda tespiti yapılan sakıncayı ortadan kaldırmak için, en iyi sonuç deęerine sahip ajanın bir sonraki döngüde *Kbest* yaklaşımının tam tersi özel bir uygulama olan *Kworst* uygulaması amaçlanmıştır. Denklem (2.34) *Kworst* formülünde temel olarak fazla bir deęişiklik yoktur, sadece uygulanacak ajanlar deęişmiştir.

$$Kworst = son_ajan + \left(1 - \frac{t}{T}\right) \cdot (K_4 - son_ajan) \quad (2.34)$$



Şekil 2.24. En iyi ajan operatörü akış şeması.

Denklem (2.34) deki, K_4 yüzde sabiti değeri 100, t arama içerisindeki döngü değeri, T arama maksimum döngü sayısı ve son_ajan son döngüde kuvvet hesabına girecek ajan sayısı değeridir. En iyi ajanın etkiyen toplam kuvvet hesabı ise Denklem (2.35) deki gibi güncellenir.

$$F_i^d(t) = \sum_{j \in K_{worst}, j \neq i}^N rand_j F_{ij}^d(t) \quad (2.35)$$

Kworst yaklaşımında bu ajana etkiyen en kötü kütleye sahip ajanları devreye sokarak toplam kuvveti, dolayısıyla hızı düşük olacaktır. Çok küçük oranlarda pozisyon değiştirmesi ile en iyi sonuç değerine yakınsama özelliğini artıracaktır.

Öncelikle, bir önceki döngüde uygunluk değerleri arasından en küçük değeri sahip ajan indisini *En* adlı değişkene atılır. Daha sonra Şekil 2.23'de sözel kodda görüldüğü gibi, *En* indisli en iyi ajana geldiğinde diğerlerinin tam tersi bir sıralama yaparak kütlesi küçük olanların işleme girmesi sağlanır [104]. En iyi ajanının toplam kuvveti hesaplanırken kütlesi küçük olan ajanlar katılırken, diğer ajanlarda *Kbest* yaklaşımı devam ettirilerek, kütlesi büyük olanlar katılımı sağlanmıştır.

Şekil 2.24'de yer alan EİAYAA algoritmasındaki akış şemasına göre döngülerin başında en iyi değere sahip ajan belirlenmekte ve ileri ki safhalar da ki hız ve konum güncellemelerinde özel bir uygulamaya sokulmaktadır. En iyi ajana özel hız hesaplaması yapılması, toplam kuvvet bulunurken kütlesi düşük ajanlar devreye sokularak hızının düşürülmesi ile gerçekleştirilmiştir.



3. BULGULAR VE TARTIŞMA

Bu tez çalışmasında önerilen operatörler olan KSYAA, KHYAA ve EİAYAA'nın performans testleri için karşılaştırma fonksiyonları kullanılmıştır. Ek1-3'de görüldüğü gibi n , fonksiyonun boyut sayısını verir ve $S \subseteq \mathbb{R}^n$ arama uzayı gerçekte sayılar kümesinin bir alt kümesi şeklinde tanımlanır. Köşeli parantez içerisindeki oran ise o fonksiyonun arama uzayı içerisindeki sınırlarını gösterir. Ek-1'de verilen yedi fonksiyon tek modlu fonksiyonlardır, optimizasyonun sonuç değerinden çok algoritmanın yakınsama hızını vermesi için tercih edilmiştir. Ek-2 çok modlu fonksiyonlar olup boyut sayısı yüksektir. Bu fonksiyonların içerisinde çok sayıda yerel minimum noktası vardır. Bundan dolayı bu fonksiyonların çalıştırılmasındaki amaç, küresel en iyi değere ne kadar yaklaştığına ve yerel minimuma takılıp takılmadığına bakmaktır. Ek-3'deki fonksiyonlar çok modlu fonksiyonlar grubunda olup, boyut sayısı ve sınır aralığı ikinci gruba göre düşüktür. Yerel minimum sayısı azdır. Bundan dolayı bu fonksiyon grubunda çalışırken döngü sayısı az tutulur, işlem zamanı ve her çalıştırma için en iyi değere yakınlık kriterine bakılır [105].

YAA içerisindeki değişkenler $G_0=100$, $\alpha=20$, $N=30$, boyut=30 olarak alınmıştır. Ek-1 ve Ek-2'deki fonksiyonlar için döngü=1000, Ek-3 için döngü=500 ve boyut her fonksiyonda değişkendir. Belirtilen bu değerler en uygun değerler olarak seçilmiş olup, deney süresince değiştirilmemiştir. Üzerinde çalışılan her bir fonksiyon için KHYAA, KSYAA EİAYAA ve standart YAA 30'ar kez çalıştırılmıştır. Her bir 30 çalıştırma sonucunda en iyi uygunluk değeri, sonuçların ortalaması ve ortanca değerleri alınmıştır.

Elde edilen operatörlerin performanslarının test edilmesi için karşılaştırma tekniği en önemli metottur. Bunun için literatürde önemli ve kullanım alanı geniş olan beş çalışmaya yer verilmiştir. İlki, standart Lipschitzian uygulaması matematiksel yöntemlerden biri sayılmıştır ve direkt adı altında karşılaştırma tablolarında kullanılmıştır [106]. Sezgisel algoritmalarından olan GA, temelde doğal evrim sürecini örnek olarak geliştirilen bir arama ve optimizasyon yöntemidir. John Holland tarafından icat edilen algoritmanın daha sonraki yıllarda geliştirilmiş çeşitleri mevcuttur. Bunlardan biri olan GA versiyonunu karşılaştırma algoritması olarak eklenmiştir [107]. PSO 1995'te Dr. Eberhart ve Dr. Kennedy tarafından geliştirilmiş, kuş sürülerinin sosyal davranışlarını esas alır [108]. Son olarak da standart YAA referans çalışma

olarak alınmıştır [4].

Elde edilen sonuçların değerlendirilmesinde ikinci yaklaşım olarak Wilcoxon işaretli sıra testi kullanılmıştır. Bu testteki amaç, standart YAA ve geliştirilen operatörlerden elde edilen sayısal verilerin ortalamalarının karşılaştırılmasıdır. Wilcoxon işaretli sıra testi güçlü bir test olup, farklar arasındaki işarete (artı mı eksi mi olduğuna) bakmayıp, farkın miktarını da dikkate alması en önemli özelliklerindedir.

Her iki eşleşmiş veri arasındaki fark hesaplanır. Farkı sıfır olan veriler dikkate alınmaz. Daha sonra bu farklar artı (+) veya eksi (-) olmalarına göre gruplandırılır. Elde edilen veriler mutlak değerine göre küçükten büyüğe doğru sıraya konur. En büyük farkın sırasına n denir. İki veya daha fazla farkın eşit olması halinde sıralarının ortalaması alınır ve çıkan sonuç numara olarak verilir. Sıfır hipotezinin doğru olabilmesi için artı ve eksi farkların sıralamaların toplamı sıfır olmalıdır [109,110]. Wilcoxon işaretli sıra testi aşağıdaki akışa göre gerçekleştirilmiştir.

- **Sıfır hipotezi (H_0) ve alternatif hipotezin (H_1) tanımlanması**

H_0 : Ajanların uygunluk değerlerinde klasik ve yeni algoritma arasında fark yoktur

H_1 : Ajanların uygunluk değerlerinde klasik ve yeni algoritma arasında fark vardır.

- **İki çalışmada elde edilen verilerin toplanması**

- **İlgili sıfır hipotezi için test istatistiğinin hesaplanması**

Örneklemede algoritmalar 30 ajan için 30 kere çalıştırılmıştır. Elde edilen her çalıştırmadaki uygunluk değeri birey olarak kaydedilmiştir. Bireyler arasındaki fark daha sonra sıraya konur. Sıfır olan farklar atılır. Bu işlemden sonra pozitif ve negatif farkların sıra değerleri ayrı toplanır. $n' \leq 25$ T testi yapılır, $n' > 25$ olması durumunda Z hesabı yapılır. Bunun için aşağıdaki eşitlik kullanılır.

$$Z = \frac{\left| T - \frac{n'(n'+1)}{4} \right| - \frac{1}{2}}{\left(\frac{n(n'+1)(2n'+1)}{24} \right)} \quad (3.1)$$

Denklem (3.1)'deki n ' değeri yukarıda yapılan sıralamanın en büyük değeridir. Ayrıca T değeri pozitif sıra değerlerinin toplamı ile negatif sıra değerlerinin toplamından hangisi küçükse o olur.

- **Test istatistiğinden elde edilen değerlerin bilinen bir olasılık dağılımı ile karşılaştırılması**

Çıkan Z değeri Çizelge 3.1'deki Wilcoxon istatistik değerler tablosuyla karşılaştırılır. İşlemlerden çıkan z değerine göre çıkış p değeri bulunur.

Çizelge 3.1. Z değeri istatistik değerler tablosu.

z değeri	p değeri
00	1.000
0.1	0.920
3.3	0.001
3.4	0.001
3.5	0.000

- **P değerinin ve sonuçların yorumlanması**

p değeri 0,05 olan önem değeriyle karşılaştırılır. Büyük olması durumunda sıfır hipotezi kabul edilirken tersi durumda büyük fark olduğunu ve dolayısıyla geliştirilen algoritmanın güçlü olduğunu göstermektedir [111].

Önem seviyesi $\alpha=0.05$ seçilmiş klasik YAA ya karşı KSYAA, KHYAA ve EİAYAA tekli karşılaştırma için Wilcoxon testinde p değerleri üretilmiştir. Eğer test sonucunda çıkan p değeri önem seviyesinden küçük veya eşit olarak üretilmiş ise sıfır hipotezinin reddi anlamına gelir. Buda yeni algoritmanın performansının güçlü olduğu sonucu çıkarılır.

Son değerlendirme yaklaşımında her bir 30 çalıştırmanın arkasında standart sapma ve tek döngü için işlem sürelerinin ortalaması alınmıştır. Standart sapma, bir sayı serisindeki sayıların, serinin aritmetik ortalamasından farklarının karelerinin toplamının dizinin eleman sayısının bir eksiğine bölümünün kareköküdür. Standart sapma verilerin kendi içerisinde nasıl bir yayılım gösterdiğini belirtir. Standart sapmanın küçük olması değerlerin birbirine yakın olduğunu, büyük olması ise değerlerin birbirinden uzak olduğunu gösterir. İşlem süresi tek döngü için hesaplanmıştır ve algoritmanın hızı hakkında fikir vermesi açısından önemlidir.

KSYAA, KHYAA ve EİAYAA çalıştırılmış ve her biri için ayrı alt başlık açılmıştır. Yukarıda belirtilen karşılaştırma, Wilcoxon işaretli sıra testi ve standart sapma şeklindeki değerlendirmeler ile algoritmaların performansları incelenmiştir.

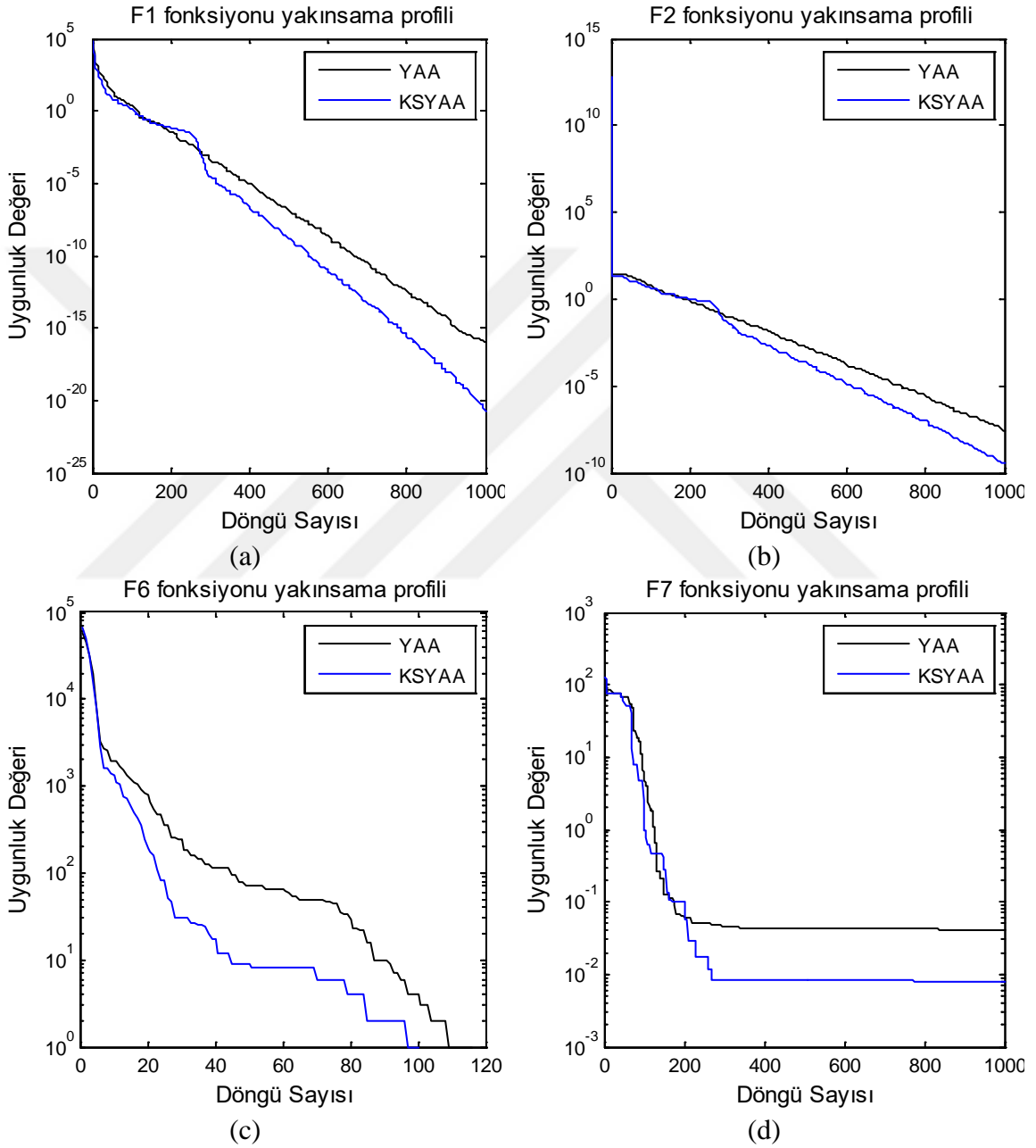
3.1. KSYAA UYGULAMA SONUÇLARI

KSYAA ile literatürde çok sık kullanılan matematiksel yöntem, GA, PSO ve standart YAA algoritmaları Çizelge 3.2'de görüldüğü gibi tek modlu yüksek boyutlu test fonksiyonlarının her biri için karşılaştırma tabloları verilmiştir. F1, F2, F3 ve F4 fonksiyonlarında tablo incelendiğinde iyileşmenin büyük oranlarda olduğu tespit edilmiştir. F5 fonksiyonun minimum değeri elde edildiğinden YAA ile aynı sonuçları vermiştir. F7 fonksiyonunda ise çok az olsa standart uygulamaya göre iyi sonuçlar verdiği görülmektedir.

Çizelge 3.2. KSYAA tek modlu fonksiyonlarının minimize sonuçları.

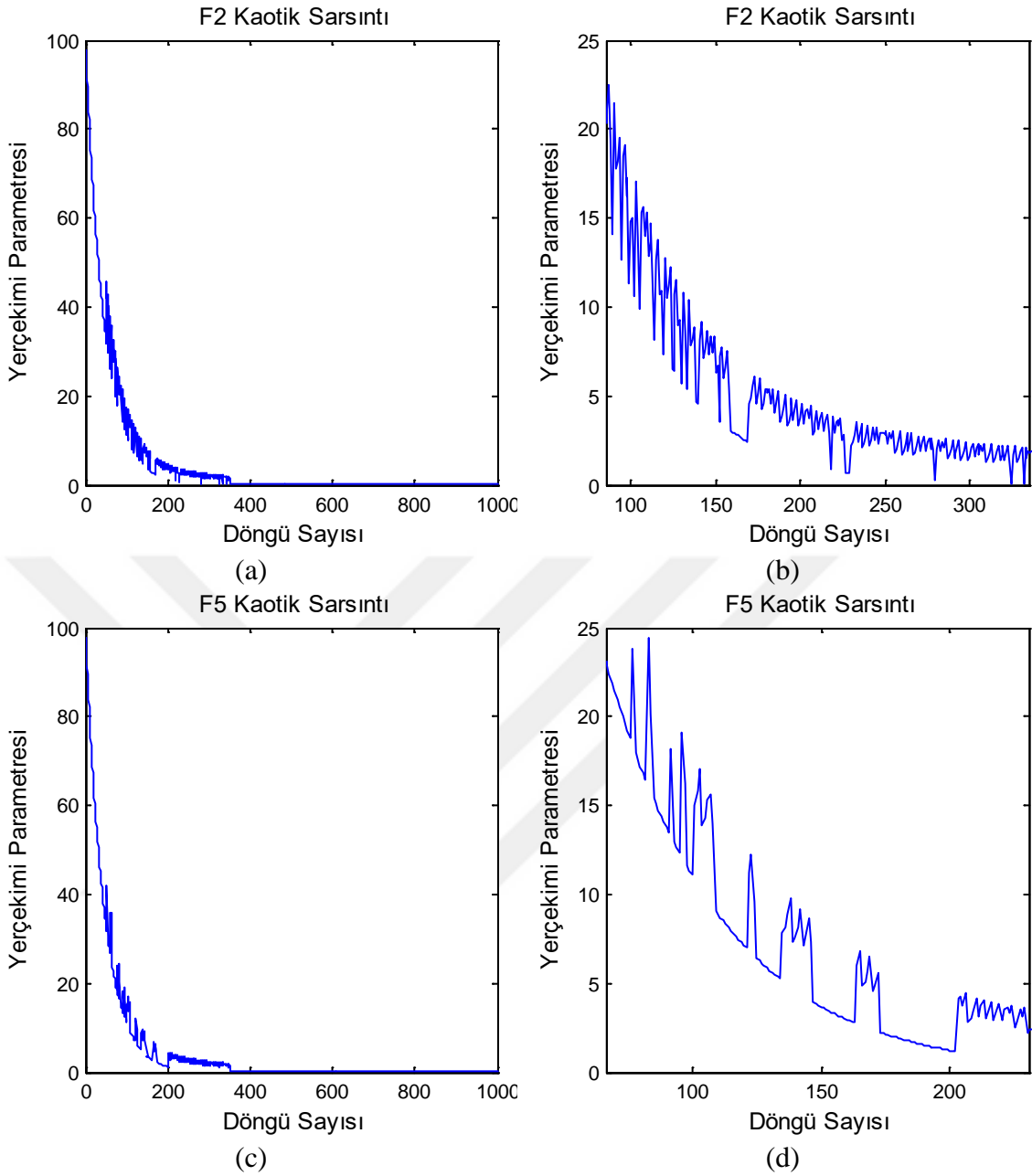
K.F.		Direk [106]	GA [107]	PSO [108]	YAA [4]	KSYAA
f_1	En iyi	Yok	9.6732	1.11×10^{-4}	9.41×10^{-19}	3.48×10^{-22}
	Ortancası	Yok	21.3478	1.4×10^{-3}	2.11×10^{-18}	2.03×10^{-23}
	Ortalaması	Yok	23.1591	2.3×10^{-3}	2.21×10^{-18}	8.10×10^{-16}
f_2	En iyi	Yok	0.7123	6.78×10^{-5}	1.49×10^{-9}	4.75×10^{-11}
	Ortancası	Yok	1.2134	0.0023	2.32×10^{-9}	1.27×10^{-11}
	Ortalaması	Yok	1.1123	2.0021	2.34×10^{-9}	8.11×10^{-8}
f_3	En iyi	Yok	395.78	139.77	98.35	2.25×10^{-18}
	Ortancası	Yok	569.10	226.89	222.21	2.97×10^{-22}
	Ortalaması	Yok	561.68	411.45	238.12	4.77×10^{-15}
f_4	En iyi	Yok	9.4512	5.4789	2.14×10^{-8}	2.51×10^{-11}
	Ortancası	Yok	11.8921	7.6723	3.19×10^{-8}	6.03×10^{-12}
	Ortalaması	Yok	11.6734	8.2144	3.51×10^{-8}	1.62×10^{-8}
f_5	En iyi	28.13	544.87	82.1987	25.69	25.69
	Ortancası	28.13	1110.21	$1.68 \times 10^{+2}$	26.14	26.14
	Ortalaması	28.50	1110.14	$3.7 \times 10^{+5}$	29.63	29.63
f_6	En iyi	Yok	4.1165	6.12×10^{-3}	9.45×10^{-17}	0
	Ortancası	Yok	24.5645	6.68×10^{-3}	2.11×10^{-16}	0
	Ortalaması	Yok	24.0321	0.0009	2.11×10^{-16}	0
f_7	En iyi	10.48	0.0431	0.0298	0.0431	0.07
	Ortancası	10.50	0.0587	0.0512	0.0587	0.06
	Ortalaması	10.99	0.567	0.0412	0.0567	0.51

Şekil 3.1'de F1, F2, F6 ve F7 test fonksiyonları için tek bir çalıştırma sonucunda döngüye bağlı olarak en iyi sonuçlar grafikleri verilmiştir. F1 ve F2 fonksiyonların da yakınsama hızının standart YAA ya göre daha hızlı olduğu ve sonuç değerlerinin daha küçük olduğu görülmektedir. F6 ve F7 fonksiyonlarında yakınsama hızı farkı çok küçüktür, ancak sonuç değerleri standart YAA ya oran ile daha küçüktür.



Şekil 3.1. YAA ile KSYAA algoritmalarının tek modlu fonksiyonlarda yakınsama özelliklerinin karşılaştırılması.

(a) F1 fonksiyonu (b) F2 fonksiyonu (c) F6 fonksiyonu (d) F7 fonksiyonu



Şekil 3.2. KSYAA tek modlu fonksiyonlarda yerçekimi parametresinin değişim eğrisi.

(a) F2 fonksiyonu (b) F2 büyütülmüş (c) F6 fonksiyonu (d) F6 büyütülmüş

Şekil 3.2'de F2 ve F5 fonksiyonları birer kere çalıştırılmış ve döngüye göre yerçekimi sabitinin azalma oranları grafikleri verilmiştir. KSYAA algoritmasının en önemli özelliği yerçekimi parametresinin azalma oranının değişimini kontrol ettiği görülmektedir. F2 fonksiyonu 5-300 döngüleri arasında çok yoğun kaotik sarsıntı oluşturulduğu görülürken F5 fonksiyonun 50-150 döngüleri arasında kaotik sarsıntı oluşturulmuştur. Bu durum tamamen fonksiyonların yerel minimumda takılma durumlarına bağlıdır. Aynı şekilde algoritmanın hassas arama yapma özelliğini

bozmamak için çalışmanın son %60 kısmında kaotik sarsıntı oluşturulmamıştır. Kaotik sarsıntı oluşturulurken, kullanılan kaotik fonksiyonlar yerçekimi sabitini pozitif yönde artıracak şekilde uygulanmıştır. Azaltacak yönde yapılan sarsıntı olumlu sonuçlar vermemiştir.

Çizelge 3.3'de görüldüğü gibi işaretli sıra testi F1 fonksiyonuna uygulanmıştır. Burada dikkat edilecek en önemli husus ikili karşılaştırma yapıldığı için tablodaki p değerlerinden son satırdaki değer dikkate alınacaktır. Elde edilen z değerinin 3.5 değerinden büyük olması ve 0.0000 p değerine sahip olmasından dolayı sıfır hipotezin reddi gerçekleşmiştir. Yani YAA ve KSYAA arasında bu fonksiyon için belirgin farklar vardır ve geliştirilen algoritma güçlüdür.

Çizelge 3.3. KSYAA F1 için Wilcoxon işaretli sıra testi sonucu.

	YAA	KSYAA
Örnekleme Sayısı	30	30
Sıra değerlerinin top	1365.0	465.0
Ortalama sıra	45.5	15.5
T	0.0	900.0
Örnekleme büyüklüğü normal dağılımı kullanmak için yeterince büyük		
Ortalama		450.0
Z	6.6456	6.6456
p-değeri (tekli)		0.00000
p-değeri (ikili)		0.00000

Çizelge 3.4'de sıra testinin bütün parametreleri verilmeyip sadece hipotezi sonucunu belirleyen p değeri verilmiştir. Özellikle F5, F6 ve F7 fonksiyonları sıfır hipotezini doğrularken diğerlerinde standart algoritma ve yeni önerilen algoritma arasında büyük farkların olduğu p eşik değeri 0.05 den küçük olmalarından anlaşılmaktadır.

Çizelge 3.4. KSYAA tek modlu fonksiyonlar için Wilcoxon işaretli sıra testi sonuçları.

K. Fonk.	f_1	f_2	f_3	f_4	f_5	f_6	f_7
p-değeri (2)	0.00000	0.00000	0.00000	0.00002	0.28378	0.31282	0.87663

KSYAA algoritmasında bütün fonksiyonlar için standart sapma daha küçüktür. Bu da 30 çalıştırmanın sonucundaki değerlerin birbirine yakın olduğunu, YAA ya göre her çalıştırma da küresel minimum değerine yaklaşma oranının daha iyi olduğunu

göstermektedir. Çizelge 3.5'deki son iki satır iki uygulamanın işlem zamanı karşılaştırılmasına yer verilmiştir. Eşit döngüler altında işlem süresinin YAA ya göre daha büyük çıkmıştır. Kaotik fonksiyonların eklenmesi ve sarsıntı oluşturulması işlem süresini artırmıştır.

Çizelge 3.5. YAA ve KSYAA arasında tek modlu fonksiyonlar için standart sapma ve işlem süresi karşılaştırması.

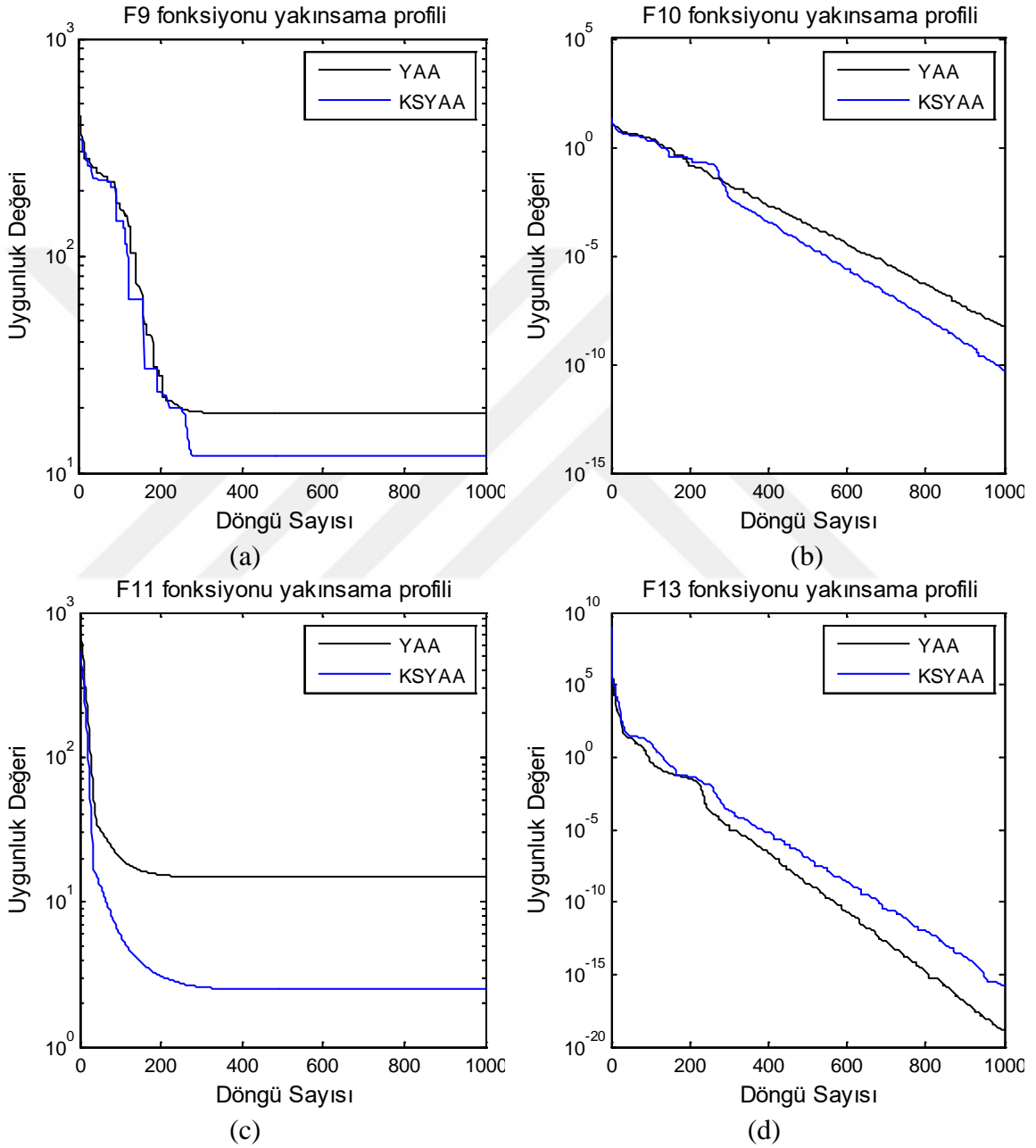
K. F.		f_1	f_2	f_3	f_4	f_5	f_6	f_7
S. S.	YAA	1.10.10-15	4.20.10-8	122.94	1.2021	47.88	0.3051	0.0097
	KSYAA	2.39.10-17	6.90.10-9	117.06	1.0902	42.99	0.1825	0.0094
İ. Z.	YAA	0.00400	0.00426	0.00553	0.00403	0.004	0.00426	0.00437
	KSYAA	0.00411	0.00433	0.00550	0.00393	0.004	0.00443	0.00445

Çizelge 3.6. KSYAA yüksek modlu çok boyutlu fonksiyonlarının minimize sonuçları.

K.F		Direk [106]	GA [107]	PSO [108]	YAA [4]	KSYAA
f_8	En iyi	-2.60×10^3	-1.2×10^4	-1.06×10^4	-3.67×10^3	-9.05×10^3
	Ortancası	-2.60×10^3	-1.2×10^4	-2.79×10^3	-2.91×10^3	-6.99×10^3
	Ortalaması	-1.39×10^3	-1.2×10^4	-9.98×10^3	-2.91×10^3	-6.77×10^3
f_9	En iyi	Yok	3.7689	35.5423	8.98	8.95
	Ortancası	Yok	5.6897	56.5689	15.88	9.94
	Ortalaması	Yok	5.9876	55.1256	15.65	9.94
f_{10}	En iyi	Yok	1.3425	0.0034	2.87×10^{-9}	8.04×10^{-12}
	Ortancası	Yok	2.1342	0.0085	3.87×10^{-9}	2.24×10^{-12}
	Ortalaması	Yok	2.1458	0.0089	3.44×10^{-9}	1.97×10^{-9}
f_{11}	En iyi	Yok	1.0479	0.000611	1.43	0.0073
	Ortancası	Yok	1.1345	0.0076	3.78	0.0474
	Ortalaması	Yok	1.1578	0.0123	3.88	0.0515
f_{12}	En iyi	0.1865	0.0139	0.000667	6.65×10^{-20}	3.80×10^{-24}
	Ortancası	0.1865	0.0378	0.2311	2.54×10^{-19}	4.92×10^{-24}
	Ortalaması	0.8038	0.0562	0.2345	0.0498	5.03×10^{-24}
f_{13}	En iyi	1.97	0.0251	1.3×10^{-31}	5.67×10^{-33}	7.95×10^{-23}
	Ortancası	1.97	0.0367	2.2×10^{-23}	2.54×10^{-32}	9.17×10^{-23}
	Ortalaması	2.57	0.0872	3.11×10^{-18}	2.89×10^{-32}	9.30×10^{-23}

Çok modlu ve yüksek boyutlu fonksiyonlar için Çizelge 3.6 incelendiğinde özellikle F9, F10, F11 ve F12 test fonksiyonlarında sonuçların ve performansın dikkat çekici olduğu görülmektedir. Sadece F13 test fonksiyonunda önerilen operatörün standart YAA ve PSO dan daha kötü sonuçlar elde edilmiştir. Bunun sebebi ilgili test fonksiyonunda

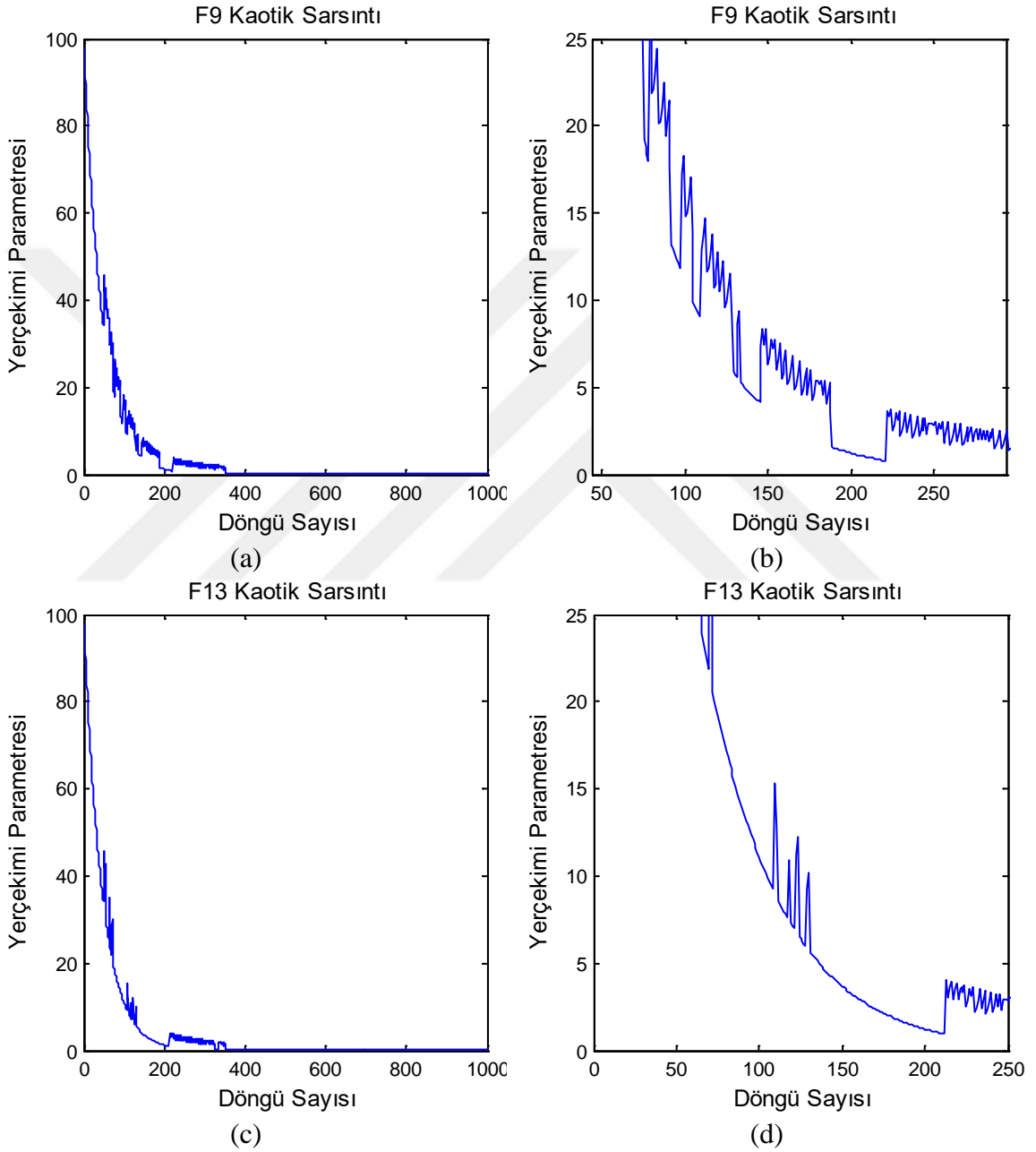
diğer fonksiyonlara göre yerel minimumun fazla olması ve fonksiyonun diğerlerine göre daha karmaşık olması algoritmanın davranışını etkilemiştir. Çok fazla yerel minimuma sahip problemler için çözüm üretilmesi ihtiyacını doğurmuştur. KSYAA operatörünün güçlü bir yanında en iyilerin ortanca ve ortalama değerlerinin en iyi değerine yakınlığı 30 çalışma performansının güçlülüğünü göstermektedir.



Şekil 3.3. YAA ile KSYAA çok modlu fonksiyonlarda yakınsama özelliklerinin karşılaştırılması.

(a) F9 fonksiyonu (b) F10 fonksiyonu (c) F11 fonksiyonu (d) F13 fonksiyonu

Şekil 3.3'de F9, F10, F11 ve F13 fonksiyonlarının döngü ilerledikçe en iyilerin ortalamalarının değişimleri verilmiştir. Küresel değeri yakalamada ve küresel değere yaklaşım hızında ve hassas arama özelliği bakımından standart YAA dan daha iyi olduğu ilk üç grafikte görülmektedir. F13 için yukarıda belirtilen olumsuz durum çıkış grafiği üzerinde de belirgin bir şekilde görülmektedir.



Şekil 3.4. KSYAA çok modlu fonksiyonlarda yerçekimi parametresini değişim eğrisi.

(a) F2 fonksiyonu (b) F2 büyütülmüş (c) F6 fonksiyonu (d) F6 büyütülmüş

Şekil 3.4'de F9 ve F13 fonksiyonları birer kere çalıştırılmış ve döngüye göre yerçekimi sabitinin azalma oranları grafikleri verilmiştir. Şekil 3.4 (a)'daki F9 fonksiyonu 5-300 döngüleri arasında çok yoğun kaotik sarsıntı oluşturulmasından dolayı sonuç değerlerine olumlu yansımıştır. Ancak Şekil 3.4 (c)'deki F13 fonksiyonun çok az kaotik sarsıntı oluşturulmuştur. Bundan dolayı da yerel minimumdan çıkma durumu özelliği bakımından diğer fonksiyonlardaki başarı elde edilememiştir. Bu fonksiyonun kaotik sarsıntı oluşturma eşik değeri düşürülmesine rağmen yerel minimumdan veya kötü arama bölgesinden çıkması sağlanamamıştır. Çizelge 3.7'de görüldüğü gibi işaretli sıra testi F8 fonksiyonuna uygulanmıştır. Burada YAA ve KSYAA arasında bu fonksiyon için belirgin farklar vardır ve geliştirilen algoritma güçlüdür. Tablonun en altındaki iki karşılaştırma p değeri 0.05 değerinden küçük çıkmıştır.

Çizelge 3.7. KSYAA F8 için Wilcoxon işaretli sıra testi sonucu.

	YAA	KSYAA
Örnekleme Sayısı	30	30
Sıra değerlerinin top	1171.0	659.0
Ortalama sıra	39.0	22.0
T	194.0	706.0
Örnekleme büyüklüğü normal dağılımı kullanmak için yeterince büyük		
Ortalama		450.0
Z	3.7774	3.7774
p-değeri (tekli)		0.00008
p-değeri (ikili)		0.00016

Çizelge 3.8'de sıra testinin bütün parametreleri verilmeyip sadece hipotezi sonucunu belirleyen p değerlerine yer verilmiştir. Bütün fonksiyonlarda sıfır hipotezinin reddi gerçekleşmiştir. F13 fonksiyonunda da belirgin farklar vardır, Çizelge 3.9'daki standart sapma değerinden de anlaşıldığı gibi bu farklar standart algoritma lehinedir.

Çizelge 3.8. KSYAA çok modlu için Wilcoxon işaretli sıra testi sonuçları.

K. Fonk.	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}
p-değeri (2)	0.00016	0.00000	0.00037	0.00000	0.00002	0.00000

Çizelge 3.9 KSYAA algoritmasında F13 fonksiyonu dışındaki fonksiyonlar için standart sapma daha küçüktür. Değerlerin birbirine yakın olduğunu, YAA ya göre her çalıştırma da küresel minimum değerine yaklaşma oranının daha iyi olduğu tespiti yapılabilir.

Çizelge 3.9. YAA ve KSYAA arasında standart sapma ve işlem süresi karşılaştırması.

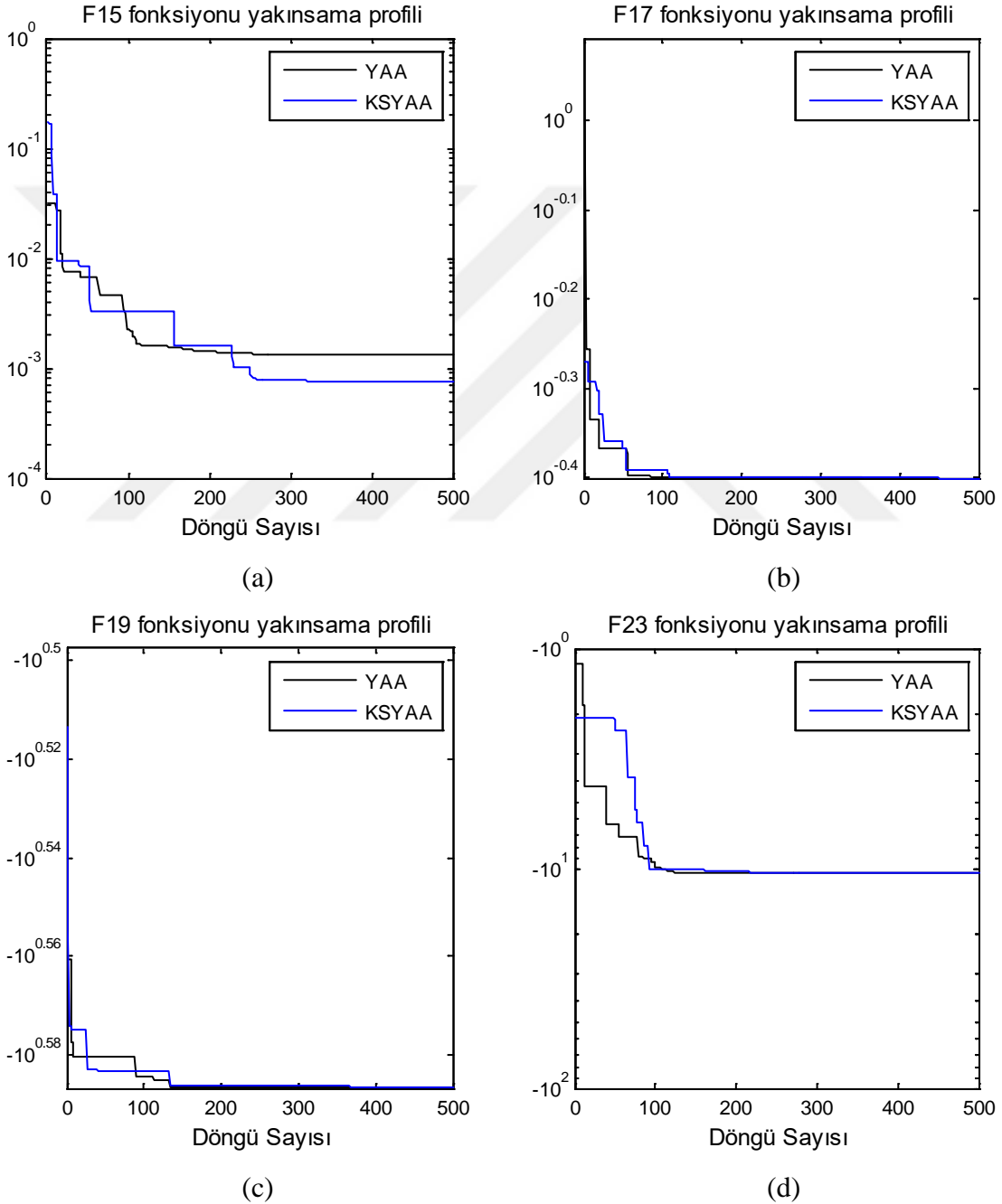
K. F.		f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}
S. S.	YAA	7.68.10+2	8.300	1.08.10-9	4.9012	0.228	0.0047
	KSYAA	7.07.10+2	6.640	1.08.10-11	0.0295	0.0316	1.4383
İ. Z.	YAA	0.00408	0.00405	0.00411	0.00403	0.0051	0.0049
	KSYAA	0.00419	0.00435	0.00411	0.00493	0.0053	0.0055

Çizelge 3.10. KSYAA yüksek modlu düşük boyutlu minimize sonuçları.

K.F.		Direk [106]	GA [107]	PSO [108]	YAA [4]	KSYAA
f_{14}	En iyi	0.998	0.9980	0.9980	0.9980	0.9980
	Ortancası	0.998	0.9980	0.9980	3.30	0.9980
	Ortalaması	6.85	0.9980	0.9980	4.72	1.1965
f_{15}	En iyi	3.7×10^{-3}	0.0011	3.07×10^{-4}	0.0016	3.84×10^{-4}
	Ortancası	3.7×10^{-3}	0.0017	7.02×10^{-4}	0.0022	0.00103
	Ortalaması	0.0662	0.0040	0.0028	0.0018	0.00107
f_{16}	En iyi	-1.0316	-1.0313	-1.0316	-1.0316	-1.0316
	Ortancası	-1.0316	-1.0315	-1.0316	-1.0316	-1.0316
	Ortalaması	-0.9735	-1.0313	-1.0316	-1.0316	-1.0316
f_{17}	En iyi	55.60	0.3979	0.3979	0.3979	0.3979
	Ortancası	55.60	0.3980	0.3979	0.3979	0.3979
	Ortalaması	57.53	0.3996	0.3979	0.3979	0.3979
f_{18}	En iyi	3.00	3.00	3.00	3.00	3.00
	Ortancası	3.00	5.70	3.00	3.00	3.00
	Ortalaması	13.09	5.75	3.00	3.00	3.00
f_{19}	En iyi	-1.8996	-3.8628	-3.8628	-3.8628	-3.8628
	Ortancası	-1.8996	-3.8628	-3.8628	-3.8628	-3.8628
	Ortalaması	-1.8334	-3.8627	-3.8628	-3.8628	-3.8628
f_{20}	En iyi	-1.1238	-3.3220	-3.3220	-3.3220	-3.3220
	Ortancası	-1.1238	-3.3217	-3.2031	-3.3220	-3.3220
	Ortalaması	-0.7505	-3.3098	-3.2369	-3.3220	-3.3180
f_{21}	En iyi	-6.8405	-10.1527	-10.1532	-10.1532	-10.1532
	Ortancası	-6.8405	-2.6824	-5.1008	-10.1486	-10.1530
	Ortalaması	-3.6348	-5.6605	-5.7496	-9.9541	-7.4141
f_{22}	En iyi	-7.0871	-10.4029	-10.4029	-10.4029	-10.4029
	Ortancası	-7.0871	-10.3932	-10.4029	-10.4029	-10.4029
	Ortalaması	-4.3715	-7.3421	-9.1118	-10.4008	-10.4029
f_{23}	En iyi	-7.2170	—	-10.5364	-10.5364	-10.5364
	Ortancası	-7.2170	-4.5054	-10.5364	-10.5364	-10.5364
	Ortalaması	-4.4568	-6.2541	-9.7634	-10.5364	-10.5364

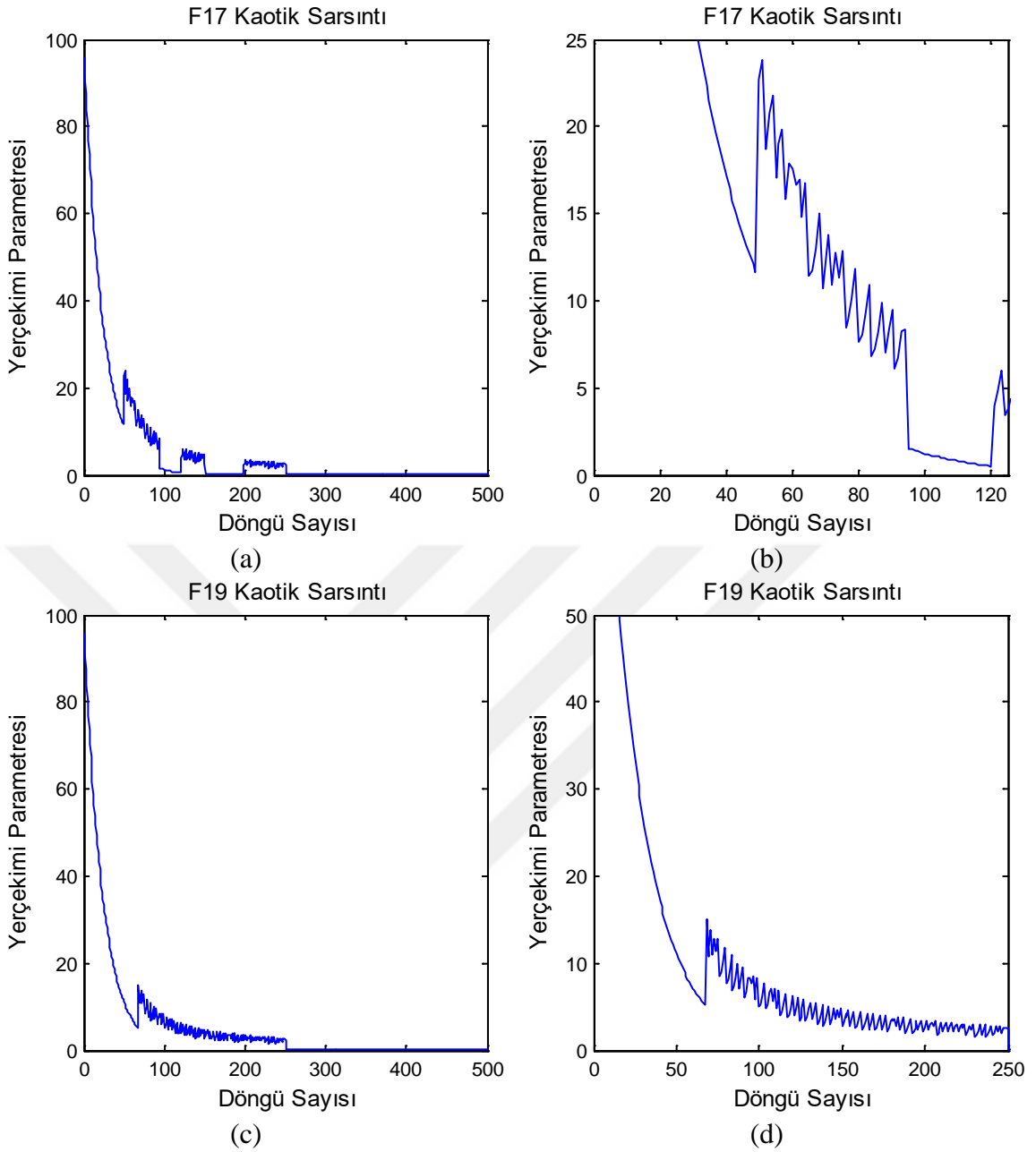
Çizelge 3.10'da çok modlu düşük boyutlu test fonksiyonlarında önerilen çalışmanın sonuçları verilmiştir. Diğer test fonksiyonlarında olduğu gibi iyi performans

göstermiştir, fonksiyonların özelliklerinden dolayı olumlu sonuçlar çok küçük oranlarda gerçekleşmiştir. İki algoritma arasındaki büyük değişim F14 ve F15 fonksiyonlarında vardır. Şekil 3.5 (b), (c) ve (d)'de F17, F19 ve F23 fonksiyonları için iki algoritmanın sonuç değerlerinin aynı olduğunu grafiksel olarak göstermektedir. Ancak F17 ve F19 için istenilen sonuca daha önce ulaşıldığı görülmektedir. Şekil 3.5 (a)'da ise iyileşmenin çok az da olsa KSYAA operatörünün lehinedir.



Şekil 3.5. YAA ile KSYAA yakınsama özelliklerinin karşılaştırılması.

(a) F16 fonksiyonu (b) F17 fonksiyonu (c) F19 fonksiyonu (d) F23 fonksiyonu



Şekil 3.6. KSYAA uygulanması sonrasında yerçekimi parametresinin değişim eğrisi.

(a) F2 fonksiyonu (b) F2 büyütülmüş (c) F6 fonksiyonu (d) F6 büyütülmüş

Şekil 3.6'da F17 ve F19 fonksiyonları birer kere çalıştırılmış ve döngüye göre yerçekimi sabitinin azalma oranları grafikleri verilmiştir. Her iki fonksiyon içinde çok sık oranda 50-250 döngüleri arasında çok yoğun kaotik sarsıntı oluşturulduğu görülmektedir. Çizelge 3.11'de görüldüğü gibi işaretli sıra testi F16 fonksiyonuna uygulanmıştır. Burada YAA ve KSYAA arasında bu fonksiyon için belirgin farkın olmadığı p değerinin 0.05 değerinden çok büyük çıkmasından anlaşılmaktadır. Aynı şekilde Çizelge 3.12'deki F17 fonksiyonu için söylenebilir. Bu fonksiyonun özelliği ise her iki

algoritmadaki verilerin birebir aynı olmasından kaynaklanan durumdur. Bundan dolayı da p değeri tekli karşılaştırma için 1.00000 ve ikili karşılaştırma için 2.00000 değerlerine sahip olup çok büyüktür. Bu iki fonksiyon içinde sıfır hipotezi doğrulanmıştır.

Çizelge 3.11. F16 fonksiyonu için Wilcoxon işaretli sıra testi sonucu.

	YAA	KSYAA
Örnekleme Sayısı	30	30
Sıra değerlerinin top	886.5	943.5
Ortalama sıra	29.6	31.4
T	478.5	421.5
Örnekleme büyüklüğü normal dağılımı kullanmak için yeterince büyük		
Ortalama		450.0
Z	0.7953	0.7953
p-değeri (tekli)		0.21322
p-değeri (ikili)		0.42644

Çizelge 3.12. F17 fonksiyonu için Wilcoxon işaretli sıra testi sonucu.

	YAA	KSYAA
Örnekleme Sayısı	30	30
Sıra değerlerinin top	915.0	915.0
Ortalama sıra	30.5	30.5
T	450.0	450.0
Örnekleme büyüklüğü normal dağılımı kullanmak için yeterince büyük		
Ortalama		450.0
Z	-Sonsuz	-Sonsuz
p-değeri (tekli)		1.00000
p-değeri (ikili)		2.00000

Çizelge 3.13'de sıra testinin bütün parametreleri verilmeyip sadece hipotezi sonucunu belirleyen p değerlerine yer verilmiştir. Sadece F14 fonksiyonunda sıfır hipotezinin reddi gerçekleşirken, diğerlerinde sıfır hipotezi gerçekleşmiştir. Bunun sebebine geçmeden evvel Çizelge 3.10 dikkatlice incelendiğinde bu fonksiyonlar için YAA ve KSYAA sonuçlarının hepsinin aynı veya çok yakın olduğu görülecektir. Bundan dolayı z değeri çok küçük çıkararak p değerinin 1.00000 olması ile sonuçlanmıştır. Önem seviyesi $\alpha=0.05$ den büyüktür sıfır hipotez doğrulanmıştır, iki algoritma arasında fark yoktur.

Çizelge 3.13. KSYAA çok modlu düşük boyutlu fonksiyonları için Wilcoxon işaretli sıra testi sonuçları.

K. F.	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}	$f_{19}-f_{20}$	f_{21}	$f_{22}-f_{23}$
p (2)	0.00065	0.25188	0.42644	2.00000	0.7411	2.00000	0.3816	2.00000

Çizelge 3.14'deki KSYAA operatöründeki fonksiyonlar için standart sapma değerleri daha küçüktür. Değerlerin birbirine yakın olduğunu, YAA ya göre her çalıştırma da küresel minimum değerine yaklaşma oranının daha iyi olduğu tespiti yapılabilir. Tek bir döngü için işlem süresinin büyük olması kaotik hesaplamalardan kaynaklanmaktadır. Ancak yakınsama hızı standart algoritmaya göre yüksek olmasından dolayı daha düşük döngülerde çalıştırarak bu dezavantaj ortadan kaldırılabilir.

Çizelge 3.14. YAA ve KSYAA arasında çok modlu düşük boyutlu fonksiyonları için standart sapma ve süre karşılaştırması.

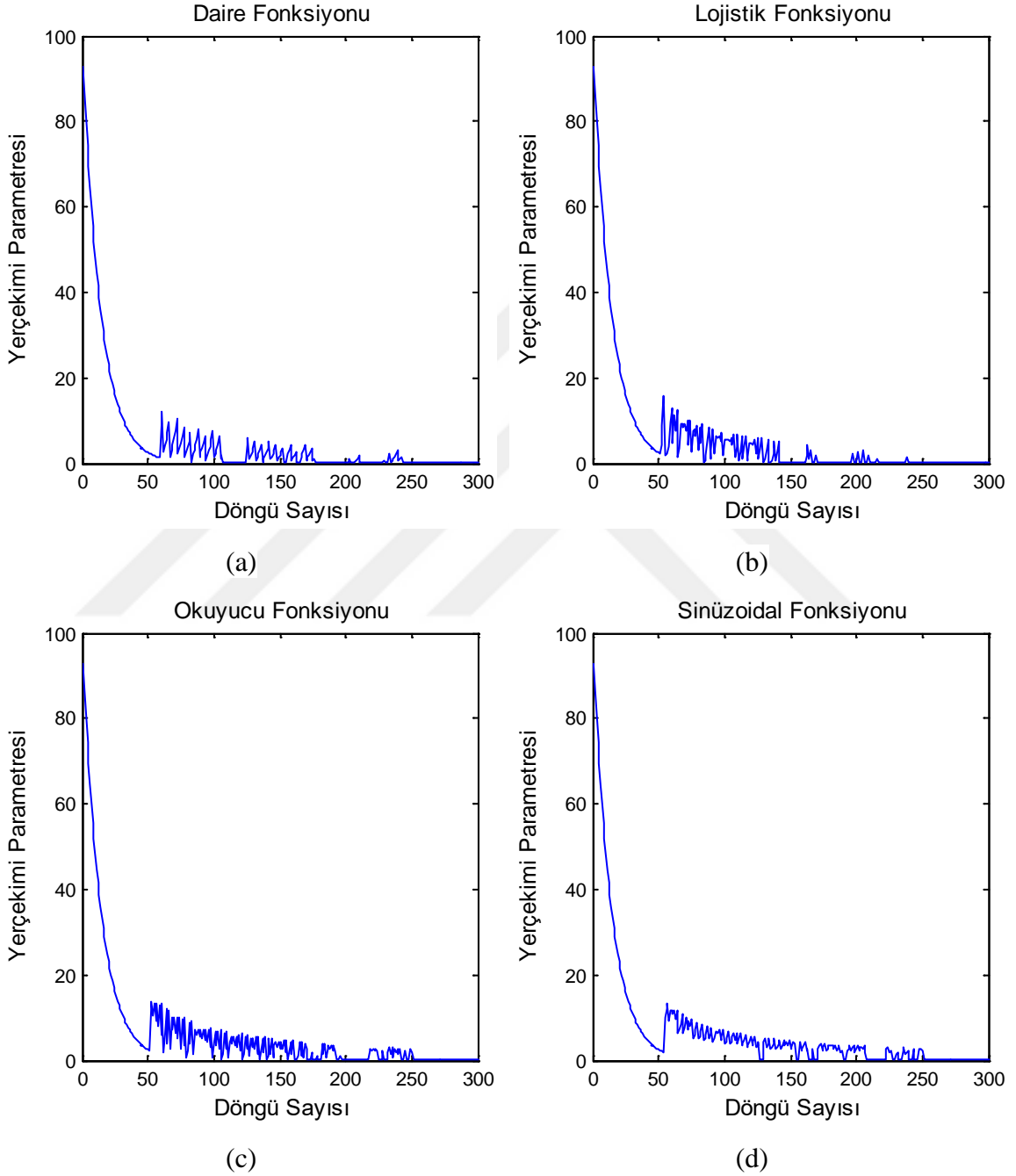
K. F.	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}	f_{21}	f_{22}	f_{23}
S. S.	2.7102	0.00045	4.6.10-16	5.1.10-15	0	0	0	3.2544	0	0
	0.5466	0.00038	4.5.10-16	3.6.10-15	0	0	0	3.1524	0	0
İ. Z.	0.0049	0.0030	0.0027	0.0027	0.002	0.002	0.002	0.0029	0.002	0.002
	0.0051	0.0035	0.0029	0.0029	0.003	0.002	0.003	0.0033	0.003	0.003

Çizelge 3.15. Kaotik fonksiyonların performans sonuçları.

f_9 $N=30$ $Döngü=300$ $\alpha=20$ $G_0=100$ 50-250 kaotik sarsıntı	Daire	Lojistik	Okuyucu	Sinüzoidal
En iyi	20.7469	19.7570	16.9612	19.3050
En iyilerin Ortancası	37.4443	39.4024	34.8264	33.2083
En iyilerin Ortalaması	38.0403	38.1632	34.5006	34.0181
Standart Sapma	9.4863	9.9964	9.4526	10.1380
İşlem Zamanı	0.0044	0.0042	0.0043	0.0044

KSYAA algoritmasında döngüler ilerledikçe en iyi sonuç değeri değişmiyorsa ajanlar yerel minimuma takılmış veya kötü arama bölgesindedir. Bu durumda yerçekimi parametresi üzerinde kaotik sarsıntı oluşturarak, ajanların pozisyonları daha büyük oranda değiştirilmiştir. Kaotik sarsıntıları daha iyi görebilmek ve inceleyebilmek için referans olarak F9 fonksiyonu seçilmiş ve döngü sayısı 300 gibi küçük bir değer de

tutulmuştur. Her bir kaotik fonksiyon için 50-250 aynı aralık için sarsıntı oluşturulmuştur. Çizelge 3.15'deki Okuyucu kaotik fonksiyonunun sonuçları dikkat çekicidir. Bundan dolayı da yukarıdaki bütün deneysel çalışmalarda Okuyucu fonksiyonu tercih edilmiştir.



Şekil 3.7. KSYAA kaotik fonksiyonlarının sarsıntı eğrileri.

(a)Circle eğrisi (b) Logistic eğrisi (c) Singer eğrisi (d) Sinüzoidal eğrisi

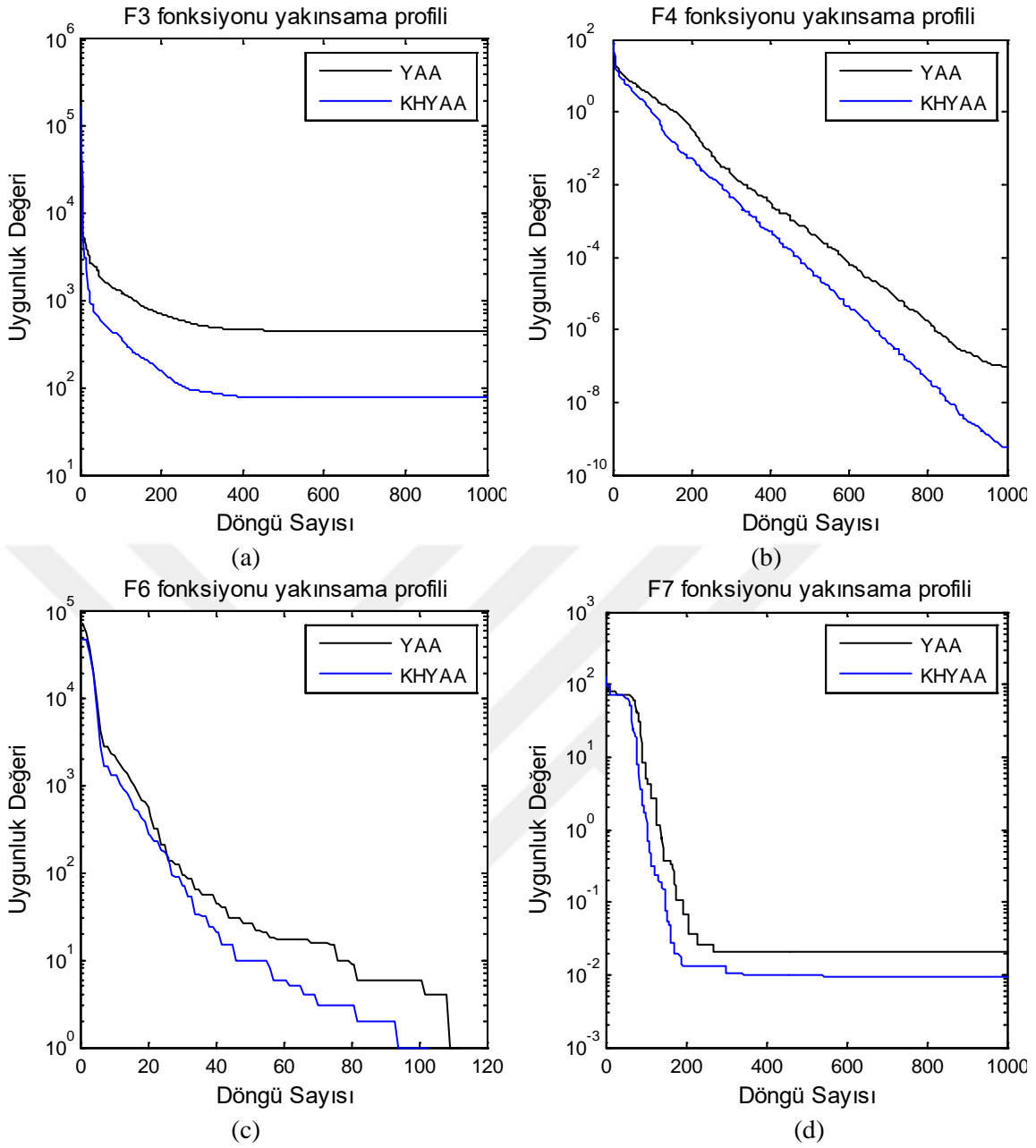
Şekil 3.7 uygulama çalışmasında ve operatörde tercih edilen kaotik fonksiyonların aynı şartlar altındaki çıkış eğrileri verilmiştir. Şekil 3.7 (b) ve (c)'deki Okuyucu ve Lojistik fonksiyonlarının $G(t)$ değerini değiştirme noktasında daha değişken oldukları görülmektedir. Bu özellik de bu iki kaotik haritanın katkılarının daha iyi olması ile sonuçlanmıştır.

3.2. KHYAA UYGULAMA SONUÇLARI

Grupsal davranışın dışında veya uzakta kalmış ajanların hızlarını kurtulma hızı yaklaşımı uygulayarak, bu ajanların hızlarını güncellemek ve grup davranışı sağlamak amaç edinilmiştir. Bu operatörde özellikle grup davranışı dışında kalan ajanların tespiti ve bu ajanlara kurtulma hızının negatif yönde eklenmesi ile sağlanabileceği önerilmiştir. KHYAA operatörünün performansını değerlendirmek ve etkinliğini kanıtlamak için, Yao ve arkadaşlarının çalışmasında kullanılan 23 test fonksiyonundan faydalanılmıştır [112].

Çizelge 3.16. KHYAA tek modlu fonksiyonların minimize sonuçları.

K.F.		Direk [106]	GA [107]	PSO [108]	YAA [4]	KHYAA
f_1	En iyi	Yok	9.6732	1.11×10^{-4}	9.41×10^{-19}	2.63×10^{-21}
	Ortancası	Yok	21.3478	1.4×10^{-3}	2.11×10^{-18}	3.96×10^{-21}
	Ortalaması	Yok	23.1591	2.3×10^{-3}	2.21×10^{-18}	4.18×10^{-21}
f_2	En iyi	Yok	0.7123	6.78×10^{-5}	1.49×10^{-9}	1.95×10^{-10}
	Ortancası	Yok	1.2134	0.0023	2.32×10^{-9}	2.94×10^{-10}
	Ortalaması	Yok	1.1123	2.0021	2.34×10^{-9}	3.01×10^{-10}
f_3	En iyi	Yok	395.78	139.77	98.35	52.29
	Ortancası	Yok	569.10	226.89	222.21	117.066
	Ortalaması	Yok	561.68	411.45	238.12	116.48
f_4	En iyi	Yok	9.4512	5.4789	2.14×10^{-8}	4.2×10^{-11}
	Ortancası	Yok	11.8921	7.6723	3.19×10^{-8}	6.37×10^{-11}
	Ortalaması	Yok	11.6734	8.2144	3.51×10^{-8}	0.16405
f_5	En iyi	28.1	544.87	82.1987	25.69	25.02
	Ortancası	28.1	1110.21	$1.68 \times 10^{+2}$	26.14	26.11
	Ortalaması	28.5	1110.14	$3.7 \times 10^{+5}$	29.63	28.31
f_6	En iyi	Yok	4.1165	6.12×10^{-3}	9.45×10^{-17}	0
	Ortancası	Yok	24.5645	6.68×10^{-3}	2.11×10^{-16}	0
	Ortalaması	Yok	24.0321	0.0009	2.11×10^{-16}	0
f_7	En iyi	10.4	0.0431	0.0298	0.0431	0.00384
	Ortancası	10.5	0.0587	0.0512	0.0587	0.01133
	Ortalaması	10.9	0.567	0.0412	0.0567	0.01335



Şekil 3.8. YAA ile KHYAA yakınsama özelliklerinin karşılaştırılması.

(a) F3 fonksiyonu (b) F4 fonksiyonu (c) F6 fonksiyonu (d) F7 fonksiyonu

KHYAA ile literatürde çok sık kullanılan matematiksel yöntem, GA, PSO, standart YAA Çizelge 3.16'da görüldüğü gibi tek modlu yüksek boyutlu test fonksiyonlarının her biri için karşılaştırma şeklinde verilmiştir. İyileşme ve farkın çok büyük olduğu F1 ve F4 fonksiyonlarında göze çarparken, KHYAA operatöründe F6 fonksiyonun minimum değerini yakaladığı görülmektedir. Şekil 3.8'de F3, F4, F6 ve F7 test fonksiyonları için tek bir çalıştırma sonucunda döngüye bağlı olarak en iyi sonuçlar grafikleri verilmiştir. Özellikle Şekil 3.8 (b)'deki F4 fonksiyonunda yakınsama hızının standart YAA ya göre daha hızlı olduğu ve sonuç değerlerinin daha küçük olduğu

görülmektedir. Ayrıca Şekil 3.8 (c)'deki F6 fonksiyonunun çok küçük döngülerde fonksiyonun minimum değerini yakalamıştır. Şekil 3.8 (a)'da yer alan F3 fonksiyonunda yakınsama hızı ve sonuç değer farkının çok küçük olduğu tespitleri yapılmıştır.

Çizelge 3.17'de F5 fonksiyonu için farkın çok fazla olmadığını kanıtlayan işaretli sıra testi parametreleri ve sonuçları verilmiştir. Elde edilen z değerinin 3.5 değerinden küçük ve çıkan 0.21702 p sonucunun eşik değeri olan 0.05 değerinden büyük olmasından dolayı sıfır hipotezi gerçekleşmiştir. Yani YAA ve KHYAA arasında bu fonksiyon için belirgin farklar yoktur. Diğer fonksiyonlar için uygulama parametre değerleri verilmeyip, sadece sonuç değerleri Çizelge 3.18'de gösterilmiştir. Buna göre F5 ve F6 fonksiyonları dışındaki bütün fonksiyonlar için sıfır hipotezini reddi gerçekleşmiştir. Standart algoritma ve KHYAA yeni önerilen algoritma arasında büyük farkların olduğu p eşik değeri 0.05 den küçük olmalarından anlaşılmaktadır.

Çizelge 3.17. KHYAA F5 için Wilcoxon işaretli sıra testi sonucu.

	YAA	KHYAA
Örnekleme Sayısı	30	30
Sıra değerlerinin top	9999.0	831.0
Ortalama sıra	33.3	27.7
T	366.0	534.0
Örnekleme büyüklüğü normal dağılımı kullanmak için yeterince büyük		
Ortalama		450.0
Z	1.2345	1.2345
p-değeri (tekli)		0.10851
p-değeri (ikili)		0.21702

Çizelge 3.18. YAA ve KHYAA tek modlu Wilcoxon işaretli sıra testi sonuçları.

K. Fonk.	f_1	f_2	f_3	f_4	f_5	f_6	f_7
p-değeri (2)	0.00000	0.00000	0.00000	0.00000	0.21702	0.33371	0.00000

Çizelge 3.19 KHYAA da bütün fonksiyonlar için standart sapma daha küçük olması 30 çalıştırmanın sonucundaki kendi arasındaki değerlerin birbirine daha yakın olduğunu göstermektedir. KHYAA uygulama sonuçlarının, YAA ya göre her çalıştırma da küresel minimum değerine yakınlığı ve benzer sonuçlarının çıkma oranı yüksektir. Eşit döngüler altında işlem süresinin YAA ya göre daha büyük çıkmasının sebebi ise grupsal davranışın dışındaki bütün ajanlar için ek olarak kurtulma hızı hesaplaması

yapılmasıdır. Ancak Şekil 3.8'deki grafikler dikkatlice incelendiğinde daha küçük döngülerde daha iyi sonuçlar alındığı ve bu yüzden işlem süresinin döngü sayısını az vererek giderilebilir.

Çizelge 3.19. YAA ve KHYAA arasında tek modlu fonksiyonlar için standart sapma ve işlem süresi karşılaştırması.

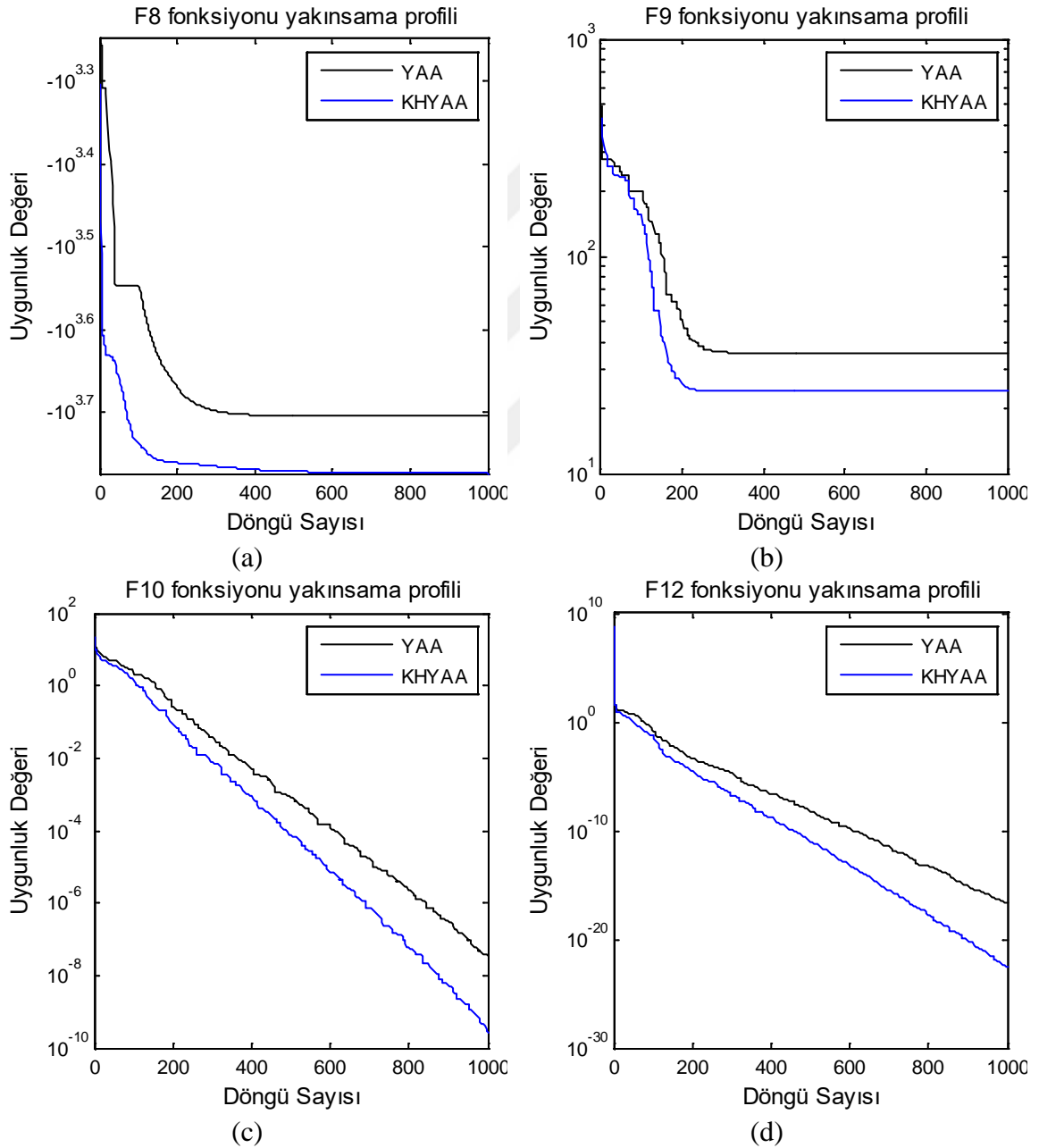
K. F.		f_1	f_2	f_3	f_4	f_5	f_6	f_7
S. S.	YAA	2.23.10-17	7.57.10-9	167.07	1.98084	22.72	0.182	0.01604
	KHYAA	3.96.10-21	4.70.10-11	36.439	0.44323	11.10	0	0.0062
İ. Z.	YAA	0.00397	0.00410	0.005756	0.0040	0.00	0.00	0.0041
	KHYAA	0.01724	0.01709	0.019809	0.01714	0.017	0.016	0.01704

Çok modlu ve yüksek boyutlu fonksiyonlar için KHYAA sonuçları Çizelge 3.20 de yer verilmiştir. Karşılaştırma yapıldığında özellikle F8, F10 ve F12 test fonksiyonlarında sonuçların ve performansın dikkat çekici olduğu görülmektedir. F12 fonksiyonunda daha net görüldüğü üzere, ortalamaların klasik algoritmalara göre en iyi değere daha yakın olduğu tespiti yapılmıştır.

Çizelge 3.20. KHYAA yüksek modlu çok boyutlu fonksiyonların minimize sonuçları.

K.F.		Direk [106]	GA [107]	PSO [108]	YAA [4]	KHYAA
f_8	En iyi	$-2.60 \times 10^{+3}$	$-1.2 \times 10^{+4}$	$-1.06 \times 10^{+4}$	-3.67×10^3	-7.68×10^3
	Ortancası	$-2.60 \times 10^{+3}$	$-1.2 \times 10^{+4}$	$-2.79 \times 10^{+3}$	-2.91×10^3	-6.22×10^3
	Ortalaması	$-1.39 \times 10^{+3}$	$-1.2 \times 10^{+4}$	$-9.98 \times 10^{+3}$	-2.91×10^3	-6.19×10^3
f_9	En iyi	Yok	3.7689	35.5423	8.98	8.87
	Ortancası	Yok	5.6897	56.5689	15.88	9.64
	Ortalaması	Yok	5.9876	55.1256	15.65	9.98
f_{10}	En iyi	Yok	1.3425	0.0034	2.87×10^{-9}	1.72×10^{-11}
	Ortancası	Yok	2.1342	0.0085	3.87×10^{-9}	2.55×10^{-11}
	Ortalaması	Yok	2.1458	0.0089	3.44×10^{-9}	2.58×10^{-11}
f_{11}	En iyi	Yok	1.0479	0.000611	1.43	1.1453
	Ortancası	Yok	1.1345	0.0076	3.78	1.4254
	Ortalaması	Yok	1.1578	0.0123	3.88	1.6897
f_{12}	En iyi	0.1865	0.0139	0.000667	6.65×10^{-20}	3.59×10^{-24}
	Ortancası	0.1865	0.0378	0.2311	2.54×10^{-19}	4.24×10^{-24}
	Ortalaması	0.8038	0.0562	0.2345	0.0498	8.12×10^{-24}
f_{13}	En iyi	1.97	0.0251	1.34×10^{-31}	5.67×10^{-33}	5.18×10^{-23}
	Ortancası	1.97	0.0367	2.2×10^{-23}	2.54×10^{-32}	1.12×10^{-22}
	Ortalaması	2.57	0.0872	3.11×10^{-18}	2.89×10^{-32}	8.43×10^{-22}

Şekil 3.9'da F8, F9, F10 ve F12 fonksiyonlarının döngü ilerledikçe en iyilerin ortalamalarının değişimleri verilmiştir. Küresel değere yaklaşım hızında ve elde edilen sonuç değeri bakımından standart YAA dan daha iyi olduğu grafikleri verilen bütün fonksiyonlarda görülmektedir. Örneğin Şekil 3.9 (a) F8 fonksiyonu KHYAA küresel değere 200'lü döngülerde yaklaşırken ve YAA küresel değere 300'lerde yaklaşmaktadır. Ayrıca 0-100 aralığında önerilen algoritmanın yakınsama hızının çok yüksek olduğu grafikteki ani düşüş ile açıklanabilir.



Şekil 3.9. YAA ile KHYAA çok modlu yakınsama özelliklerinin karşılaştırılması.

(a) F8 fonksiyonu (b) F9 fonksiyonu (c) F10 fonksiyonu (d) F12 fonksiyonu

Çizelge 3.21. F10 fonksiyonu için Wilcoxon işaretli sıra testi sonucu.

	YAA	KHYAA
Örnekleme Sayısı	30	30
Sıra değerlerinin top	1365.0	465.0
Ortalama sıra	45.5	15.5
T	0.0	900.0
Örnekleme büyüklüğü normal dağılımı kullanmak için yeterince büyük		
Ortalama		450.0
Z	6.6457	6.6457
p-değeri (tekli)		0.00000
p-değeri (ikili)		0.00000

Çizelge 3.21'de görüldüğü gibi işaretli sıra testi F10 fonksiyonuna uygulanmıştır. YAA ve KHYAA arasında bu fonksiyon için belirgin farklar vardır ve geliştirilen algoritma güçlüdür. Tablonun en altındaki iki karşılaştırma p değeri 0.05 değerinden küçük çıkmıştır. Bir başka ifade ile, Çizelge 3.21'de pozitif ve negatif farkların sıra değerleri ayrı toplanmış ve $n' > 25$ olmasından dolayı z hesabı yapılmıştır. z değerinin 6.6457 çıkmasından dolayı aynı şekilde sıfır hipotezinin reddi gerçekleşmiştir denir.

Çizelge 3.22. KHYAA çok modlu fonksiyonlar için Wilcoxon işaretli sıra testi sonuçları.

K. Fonk.	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}
p-değeri (2)	0.86499	0.00020	0.00000	0.00106	0.00180	0.00000

Çizelge 3.23. YAA ve KHYAA arasında çok modlu fonksiyonlar için standart sapma ve işlem süresi karşılaştırması

K. F.		f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}
S. S.	YAA	7.69.10+2	7.2145	8.98.10-10	4.59857	0.4919	0.1365
	KHYAA	6.17.10+2	6.0364	4.85.10-12	1.32564	0.2228	0.0181
İ. Z.	YAA	0.00417	0.00415	0.00412	0.00428	0.0049	0.0048
	KHYAA	0.01689	0.02468	0.01118	0.01145	0.0120	0.0120

Çizelge 3.22'de sıra testinin bütün parametreleri verilmeyip sadece hipotezi sonucunu belirleyen p değerlerine yer verilmiştir. F8 fonksiyonu dışında bütün fonksiyonlarda sıfır hipotezinin reddi gerçekleşmiştir. F8 fonksiyonunda önerilen algoritma güçlüdür,

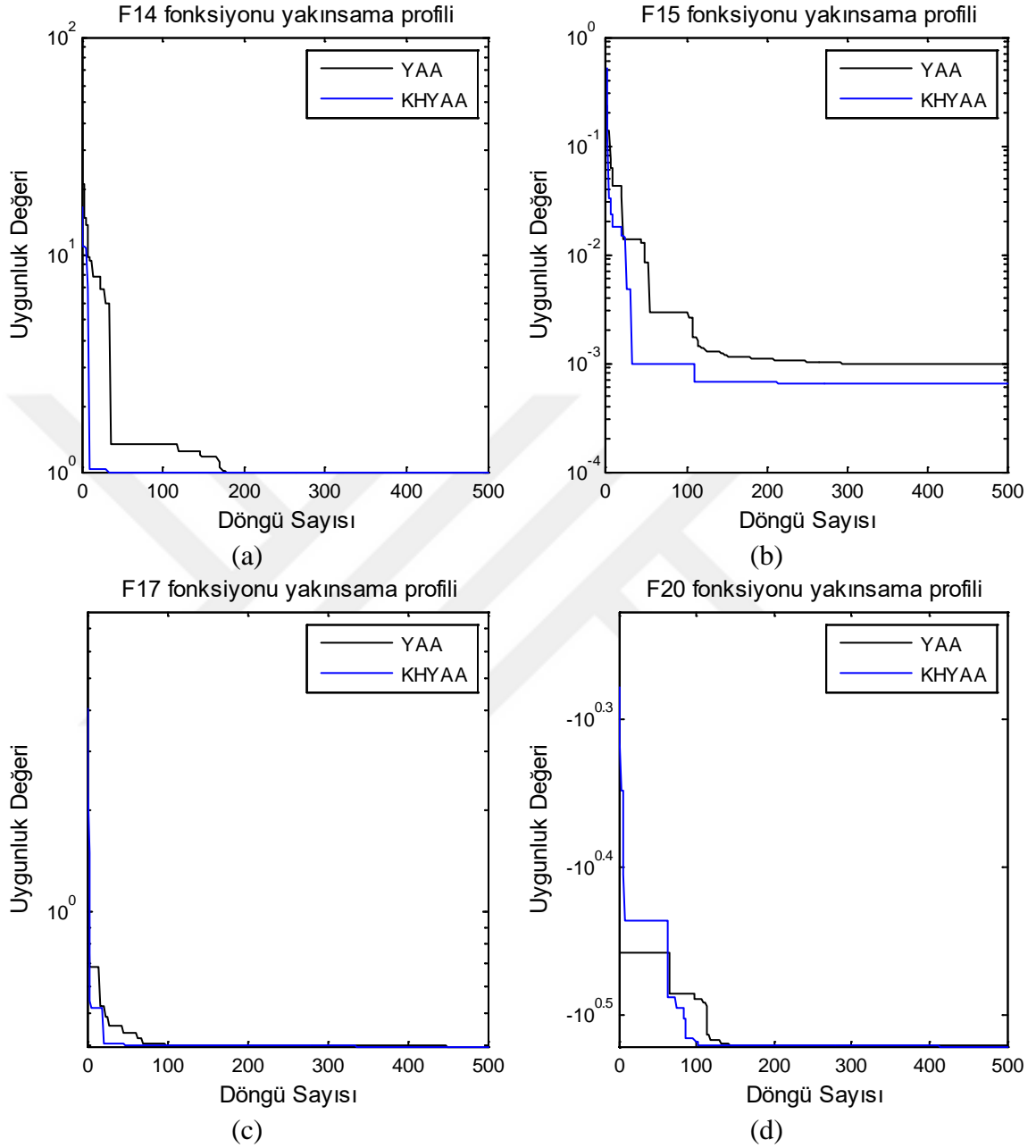
fakat fonksiyon çıktılarının negatif olması ve YAA yakın sonuçlar vermesinden dolayı sıfır hipotezi gerçekleşmiştir.

Çizelge 3.24. KHYAA yüksek modlu düşük boyutlu fonksiyonların minimize sonuçları.

K.F.		Direk [106]	GA [107]	PSO [108]	YAA [4]	KHYAA
f_{14}	En iyi	0.998	0.9980	0.9980	0.9980	0.9980
	Ortancası	0.998	0.9980	0.9980	3.30	1.9920
	Ortalaması	6.85	0.9980	0.9980	4.72	2.5051
f_{15}	En iyi	3.7×10^{-3}	0.0011	3.07×10^{-4}	0.0016	3.20×10^{-4}
	Ortancası	3.7×10^{-3}	0.0017	7.02×10^{-4}	0.0022	7.15×10^{-4}
	Ortalaması	0.0662	0.0040	0.0028	0.0018	7.69×10^{-4}
f_{16}	En iyi	-1.0316	-1.0313	-1.0316	-1.0316	-1.0316
	Ortancası	-1.0316	-1.0315	-1.0316	-1.0316	-1.0316
	Ortalaması	-0.9735	-1.0313	-1.0316	-1.0316	-1.0316
f_{17}	En iyi	55.60	0.3979	0.3979	0.3979	0.3979
	Ortancası	55.60	0.3980	0.3979	0.3979	0.3979
	Ortalaması	57.53	0.3996	0.3979	0.3979	0.3979
f_{18}	En iyi	3.00	3.00	3.00	3.00	3.00
	Ortancası	3.00	5.70	3.00	3.00	3.00
	Ortalaması	13.09	5.75	3.00	3.00	3.00
f_{19}	En iyi	-1.8996	-3.8628	-3.8628	-3.8628	-3.8628
	Ortancası	-1.8996	-3.8628	-3.8628	-3.8628	-3.8628
	Ortalaması	-1.8334	-3.8627	-3.8628	-3.8628	-3.8628
f_{20}	En iyi	-1.1238	-3.3220	-3.3220	-3.3220	-3.3220
	Ortancası	-1.1238	-3.3217	-3.2031	-3.3220	-3.3220
	Ortalaması	-0.7505	-3.3098	-3.2369	-3.3220	-3.2625
f_{21}	En iyi	-6.8405	-10.1527	-10.1532	-10.1532	-10.1532
	Ortancası	-6.8405	-2.6824	-5.1008	-10.1486	-10.1531
	Ortalaması	-3.6348	-5.6605	-5.7496	-9.9541	-7.2196
f_{22}	En iyi	-7.0871	-10.4029	-10.4029	-10.4029	-10.4029
	Ortancası	-7.0871	-10.3932	-10.4029	-10.4029	-10.4029
	Ortalaması	-4.3715	-7.3421	-9.1118	-10.4008	-10.4029
f_{23}	En iyi	-7.2170	-10.5364	-10.5364	-10.5364	-10.5364
	Ortancası	-7.2170	-4.5054	-10.5364	-10.5364	-10.5364
	Ortalaması	-4.4568	-6.2541	-9.7634	-10.5364	-10.5364

Çizelge 3.23'deki standart sapma ve işlem süresi yönünden KHYAA ve YAA karşılaştırmalarına yer verilmiştir. KHYAA da özellikle F10 ve F11 fonksiyonların standart değerlerin standart YAA göre ne kadar çok küçük olduğu görülmektedir. Yani bu iki fonksiyonda KHYAA sonuçları en iyi değere çok yakındırlar. İşlem süresi bakımında bütün fonksiyonlar için yaklaşık üç katı kadar KHYAA operatörünün daha

uzun sürdüğü tespiti yapılmıştır. Tek modlu test fonksiyonlarında söylendiği gibi uzaktaki ajanların ek olarak kurtulma hızı hesaplamaları işlem süresini artırmıştır.



Şekil 3.10. YAA ile KHYAA algoritmasının çok modlu düşük boyutlu fonksiyonlarda yakınsama özelliklerinin karşılaştırılması.

(a) F14 fonksiyonu (b) F16 fonksiyonu (c) F17 fonksiyonu (d) F20 fonksiyonu

Çizelge 3.24'de boyut sayısı ve sınır aralığı ikinci gruba göre düşük olan yüksek modlu düşük boyutlu test fonksiyonlarında KHYAA operatörünün davranışı incelenmiştir. Bilindiği gibi bu fonksiyonlarda yerel minimum sayısı azdır. Bundan dolayı bu

fonksiyon grubunda KHYAA sonuçları YAA sonuçlarına çok yakın çıkmış veya olumlu sonuçlar çok küçük oranlarda gerçekleşmiştir. İki algoritma arasındaki büyük değişim ve iyileşme F14 ve F15 fonksiyonlarında çok açık görülmektedir. Diğer fonksiyonlarda da olumlu iyileşmeler görülmüştür. Ancak sonuçların virgülden sonra dört basamak devam ettirilmesinden dolayı tablo içinde görülmemektedir. Sadece F21 fonksiyonunun ortalama değeri YAA göre daha kötü bir değer çıkmıştır.

Şekil 3.10'da F14, F15, F17 ve F20 fonksiyonları için KHYAA ve YAA karşılaştırmalı yakınsama grafiklerine yer verilmiştir. Şekil 3.10 (c)'deki F17 ve Şekil 3.10 (d)'deki F20 fonksiyonlarının iki algoritma arasındaki yakınlığı temsil ederken Şekil 3.10 (a) F14 ve Şekil 3.10 (b) F15 grafikleri ise olumlu gelişmeleri göstermektedir. Ayrıca KHYAA operatörünün bu iki fonksiyonda daha iyi sonuç üretmesinin yanında istenilen sonuca daha önce ulaşıldığı tespiti yapılmıştır.

Çizelge 3.25. KHYAA F17 için Wilcoxon işaretli sıra testi sonucu.

	YAA	KSYAA
Örnekleme Sayısı	30	30
Sıra değerlerinin top	915.0	915.0
Ortalama sıra	30.5	30.5
T	450.0	450.0
Örnekleme büyüklüğü normal dağılımı kullanmak için yeterince büyük		
Ortalama		450.0
Z	-Sonsuz	-Sonsuz
p-değeri (tekli)		1.00000
p-değeri (ikili)		2.00000

Çizelge 3.26. KHYAA F15 fonksiyonu için Wilcoxon işaretli sıra testi sonucu.

	YAA	KSYAA
Örnekleme Sayısı	30	30
Sıra değerlerinin top	1176.0	654.0
Ortalama sıra	39.2	21.8
T	189.0	711.0
Örnekleme büyüklüğü normal dağılımı kullanmak için yeterince büyük		
Ortalama		450.0
Z	3.8513	3.8513
p-değeri (tekli)		0.00006
p-değeri (ikili)		0.00012

Çizelge 3.25'de görüldüğü gibi işaretli sıra testi F17 fonksiyonuna uygulanmış sonuçlarının koyulmasının sebebi, YAA ve KHYAA arasında bu fonksiyon için belirgin farkın olmadığı p değerinin 0.05 değerinden çok büyük çıkmasını göstermektedir. Sıfır hipotezin reddi olarak da Çizelge 3.26'daki F15 fonksiyonu parametrelerinin ayrıntılı gösterimine yer verilmiştir. Bu fonksiyonun özelliği ise her iki algoritmadaki verilerin birbirlerinde çok farklı ve önerilen operatörün daha güçlü olduğu görülmektedir. Çizelgedeki p değeri ikili karşılaştırma için 0.00012 değerine sahip olup çok küçüktür.

Çizelge 3.27'de sıra testinin bütün parametreleri verilmeyip sadece hipotezi sonucunu belirleyen p değerlerine yer verilmiştir. Sadece F15 fonksiyonunda sıfır hipotezinin reddi gerçekleşirken, diğerlerinde sıfır hipotezi gerçekleşmiştir. Sebebi ise düşük boyutlu fonksiyonlar olmasından dolayı iki çalıştırma arasında fark yoktur.

Çizelge 3.27. KHYAA çok modlu düşük boyutlu fonksiyonlar için Wilcoxon işaretli sıra testi sonuçları.

K. F.	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}	f_{21}	$f_{22}-f_{23}$
p (2)	0.811	0.00012	0.42644	2.00000	0.7411	2.00000	0.974	0.1612	2.00000

Çizelge 3.28 KHYAA operatöründeki fonksiyonlar için standart sapma daha küçüktür. Değerlerin birbirine yakın olduğunu, YAA ya göre her çalıştırma da küresel minimum değerine yaklaşma oranının daha iyi olduğu tespiti yapılabilir. Tek bir döngü için işlem süresi diğer iki tip test fonksiyon grubuna göre küçüktür. Bunun sebebi döngü sayısı ve düşük boyuttur. Yine aynı şekilde kurtulma hızı hesaplamalarından dolayı yaklaşık YAA ya göre üç kat daha fazla bir süre harcanmıştır. Bunun için önerilen operatörde çalışırken yakınsama hızının güçlü olmasından dolayı döngü sayısının düşük verilmesi durumunda işlem süresi dezavantajı ortadan kalkacaktır.

Çizelge 3.28. YAA ve KHYAA arasında standart sapma ve işlem süresi karşılaştırması.

K. F.	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}	f_{21}	f_{22}	f_{23}
S. S.	2.9413	0.0000	4.7.10 ⁻¹⁶	0	0	0	0.060	3.4296	0	0
	2.5051	0.0000	3.9.10 ⁻¹⁶	0	0	0	0.0601	3.2993	0	0
İ. Z.	0.004	0.0029	0.0028	0.002	0.002	0.002	0.003	0.0031	0.002	0.002
	0.0114	0.0089	0.0086	0.008	0.008	0.008	0.009	0.009	0.008	0.008

3.3. EİAYAA UYGULAMA SONUÇLARI

YAA da ajanının üzerindeki toplam kuvvet hesaplanırken kütlesi büyük olanlar seçilir ve her döngüde ajan sayısı azaltılır. Büyük kütleli ajanların hesaplamaya katılması en iyi değere sahip ajan için büyük oranda pozisyon değiştirmesi anlamına gelir. Halbuki en iyi sonuç değerine sahip olmasından kaynaklanan kütlelerinin büyük olması çok yavaş hareket etmesi gerekirdi.

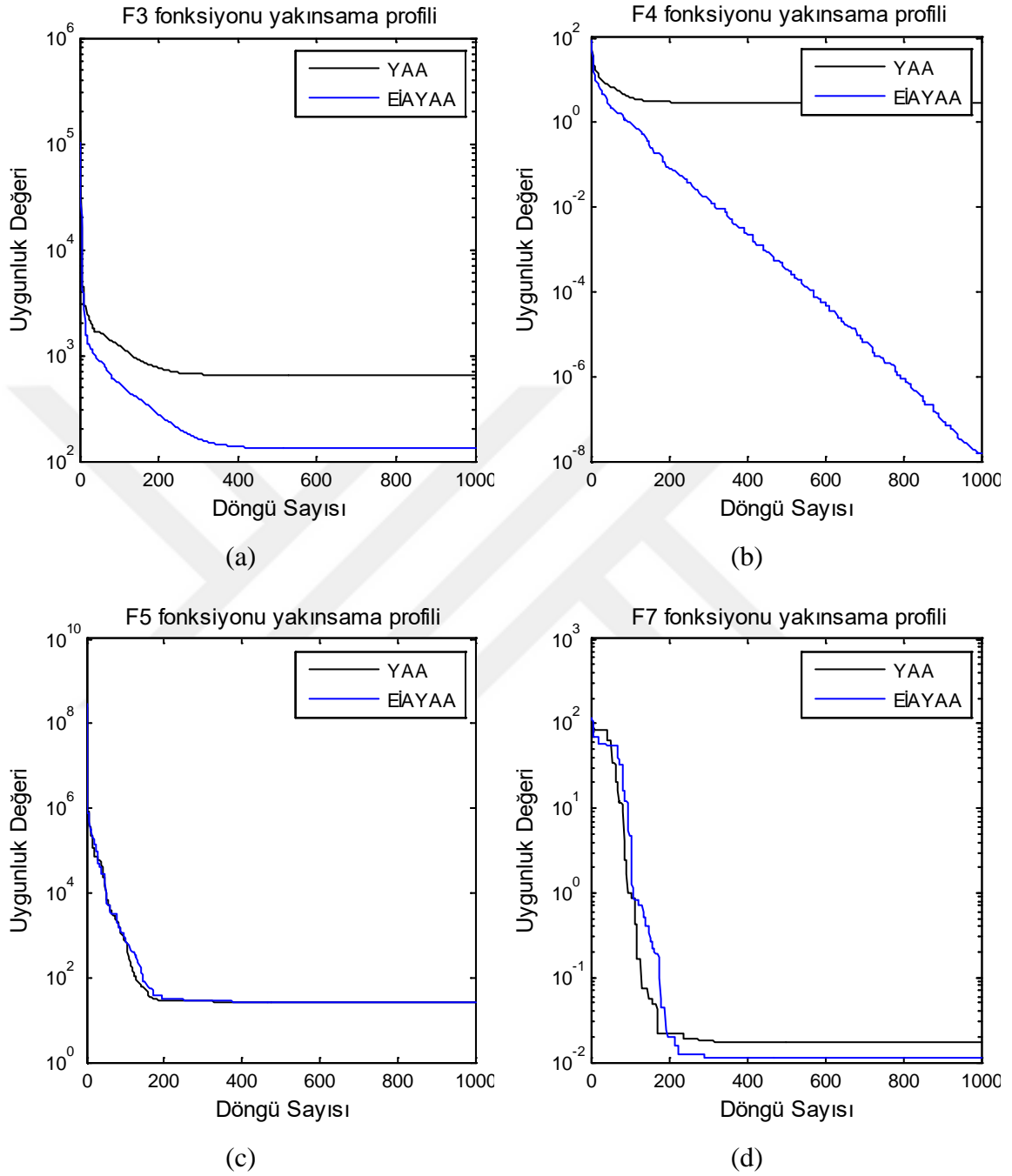
Çizelge 3.29. EİAYAA tek modlu fonksiyonların minimize sonuçları.

K.F.		Direk [106]	GA [107]	PSO [108]	YAA [4]	EİAYAA
f_1	En iyi	Yok	9.6732	1.11×10^{-4}	9.41×10^{-19}	2.72×10^{-21}
	Ortancası	Yok	21.3478	1.4×10^{-3}	2.11×10^{-18}	6.27×10^{-21}
	Ortalaması	Yok	23.1591	2.3×10^{-3}	2.21×10^{-18}	6.61×10^{-21}
f_2	En iyi	Yok	0.7123	6.78×10^{-5}	1.49×10^{-9}	1.74×10^{-10}
	Ortancası	Yok	1.2134	0.0023	2.32×10^{-9}	3.14×10^{-10}
	Ortalaması	Yok	1.1123	2.0021	2.34×10^{-9}	3.32×10^{-10}
f_3	En iyi	Yok	395.78	139.77	98.35	61.22
	Ortancası	Yok	569.10	226.89	222.21	136.73
	Ortalaması	Yok	561.68	411.45	238.12	161.33
f_4	En iyi	Yok	9.4512	5.4789	2.14×10^{-8}	9.90×10^{-11}
	Ortancası	Yok	11.8921	7.6723	3.19×10^{-8}	1.01×10^{-10}
	Ortalaması	Yok	11.6734	8.2144	3.51×10^{-8}	1.62×10^{-10}
f_5	En iyi	28.13	544.87	82.1987	25.69	25.28
	Ortancası	28.13	1110.21	$1.68 \times 10^{+2}$	26.14	26.71
	Ortalaması	28.50	1110.14	$3.7 \times 10^{+5}$	29.63	27.71
f_6	En iyi	Yok	4.1165	6.12×10^{-3}	9.45×10^{-17}	0
	Ortancası	Yok	24.5645	6.68×10^{-3}	2.11×10^{-16}	0
	Ortalaması	Yok	24.0321	0.0009	2.11×10^{-16}	0
f_7	En iyi	10.48	0.0431	0.0298	0.0431	0.0054
	Ortancası	10.50	0.0587	0.0512	0.0587	0.0121
	Ortalaması	10.99	0.567	0.0412	0.0567	0.0127

En iyi sonuç değerine sahip ajanın bir sonraki döngüde toplam kuvvetini bulurken kuvvet hesabı için kütlesi küçük ajanları devreye sokma prensibini esas alan EİAYAA operatörü önerilmiştir.

Bu algoritma tek boyutlu test fonksiyonlarına uygulanarak, literatürde çok sık kullanılan matematiksel yöntem, GA, PSO ve standart YAA algoritmaları ile Çizelge 3.29'da görüldüğü gibi karşılaştırılmıştır. F1 ve F2 fonksiyonlarında tablo incelendiğinde iyileşmenin büyük oranlarda olduğu tespit edilmiştir. F5 fonksiyonun minimum değeri

elde edildiğinden YAA ile aynı sonuçları vermiştir. F7 fonksiyonunda ise çok az olsa standart uygulamaya göre iyi sonuçlar verdiği görülmektedir.



Şekil 3.11. YAA ile EİAYAA tek modlu fonksiyonlarda yakınsama özelliklerinin karşılaştırılması.

(a) F3 fonksiyonu (b) F4 fonksiyonu (c) F6 fonksiyonu (d) F7 fonksiyonu

Şekil 3.11'de F3, F4, F5 ve F7 test fonksiyonları için tek bir çalıştırma sonucunda döngüye bağlı olarak en iyi sonuçlar grafikleri verilmiştir. Şekil 3.11 (c) F5

fonksiyonunda standart YAA ile aynı sonuçları verdiği, Şekil 3.11 (d) F7 de ise EİAYAA operatörünün az da olsa iyi performans ile sonuçlandığı eğrilerden anlaşılmaktadır. Şekil 3.11 (b) F4 fonksiyonundaki iyileşmenin büyüklüğü ve yakınsama hızının kalitesi algoritmanın güçlülüğünü ortaya koymaktadır. EİAYAA operatörü F4 fonksiyonunda 0-200 döngüleri arasında yakınsamasının çok hızlı olduğu grafiğin ani inişinden anlaşılmaktadır. Standart YAA ise F4 daha yavaş ve yumuşak bir iniş gerçekleştirmiştir. Şekil 3.11 (a)'daki F3 fonksiyonunun 500. döngülerde en iyi değere yaklaşması dışında davranışı benzerdir.

Çizelge 3.30'da görüldüğü gibi işaretli sıra testi F5 fonksiyonuna uygulanmış ve ikili karşılaştırma yapıldığı için tablodaki p değerlerinden son satırdaki değer dikkate alınmıştır. Çizelge 3.29'da F5 fonksiyonun en iyi, en iyilerin ortancası ve ortalaması değerlerine bakıldığında YAA ile EİAYAA arasında fark neredeyse hiç yok denecek kadar azdır. Bundan dolayı, elde edilen z değeri 3.5 değerinden küçük ve p değeri 0.42896 çıkmıştır, sıfır hipotezi gerçekleşmiştir. YAA ve EİAYAA arasında bu fonksiyon için belirgin fark yoktur.

Çizelge 3.30. EİAYAA F5 için Wilcoxon işaretli sıra testi sonucu.

	YAA	EİAYAA
Örnekleme Sayısı	30	30
Sıra değerlerinin top	969.0	861.0
Ortalama sıra	32.3	28.7
T	396.0	504.0
Örnekleme büyüklüğü normal dağılımı kullanmak için yeterince büyük		
Ortalama		450.0
Z	0.7910	0.7910
p-değeri (tekli)		0.21448
p-değeri (ikili)		0.42896

Çizelge 3.31. EİAYAA tek modlu Wilcoxon işaretli sıra testi sonuçları.

K. Fonk.	f_1	f_2	f_3	f_4	f_5	f_6	f_7
p-değeri (2)	0.00000	0.00000	0.00000	0.00124	0.42896	0.00028	0.00000

Çizelge 3.31'de sıra testinin bütün parametreleri verilmeyip sadece hipotezi sonucunu belirleyen p değeri verilmiştir. Tek modlu test fonksiyonları arasında sadece F5 fonksiyonunda sıfır hipotezini doğrularken diğerlerinde standart algoritma ve yeni

önerilen algoritma arasında büyük farkların olduğu p eşik değeri 0.05 den küçük olmalarından dolayı, EİAYAA operatörünün performansı memnun edicidir.

Çizelge 3.32. YAA ve EİAYAA arasında tek modlu fonksiyonlar için standart sapma ve işlem süresi karşılaştırması.

B. F.		f_1	f_2	f_3	f_4	f_5	f_6	f_7
S. S.	YAA	2.48.10 ⁻¹⁷	4.96.10 ⁻⁹	159.01	1.20	10.856	0.3051	0.0088
	EİAYAA	2.54.10 ⁻²¹	8.02.10 ⁻¹¹	93.12	1.08	4.96	0.1825	0.0050
İ. Z.	YAA	0.0039	0.0042	0.0056	0.004	0.0042	0.00426	0.0043
	EİAYAA	0.0073	0.0072	0.0134	0.007	0.0080	0.00443	0.0111

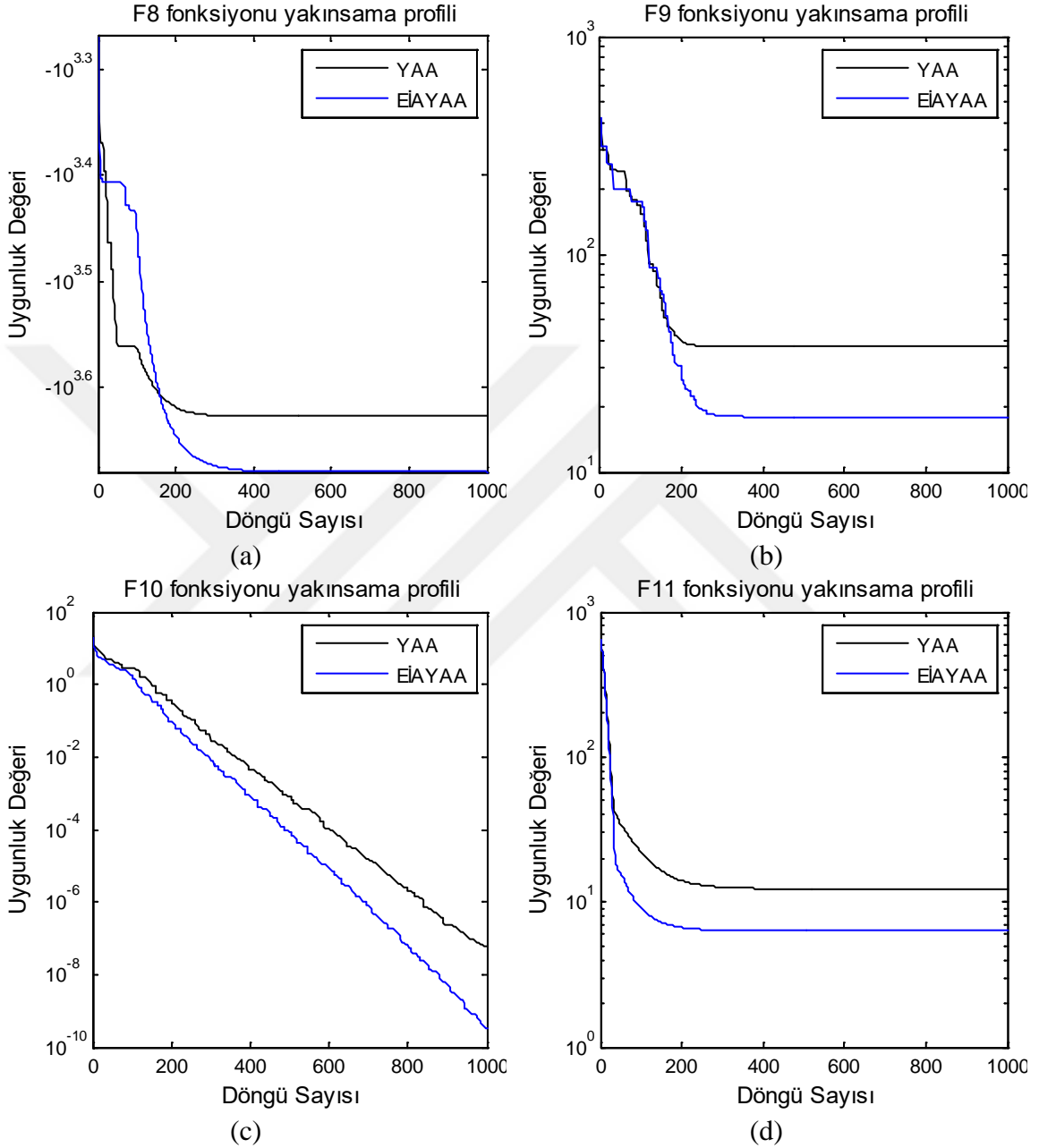
Çizelge 3.32 EİAYAA operatörünün bütün fonksiyonlarda çalıştırılması sonucunda standart sapma değeri standart YAA göre daha küçüktür. Bu da 30 çalıştırmanın sonucunda değerlerin en iyi sonuç değerine ve birbirine yakın olduğu göstermektedir. Sonuçların birbirine yakınlığı operatörün güçlü özelliklerindedir.

Çizelge 3.33. EİAYAA yüksek modlu çok boyutlu minimize sonuçları.

K.F.		Direk [106]	GA [107]	PSO [108]	YAA [4]	EİAYAA
f_8	En iyi	-2.60×10 ⁺³	-1.2×10 ⁺⁴	-1.06×10 ⁺⁴	-3.67×10 ³	-7.22×10 ³
	Ortancası	-2.60×10 ⁺³	-1.2×10 ⁺⁴	-2.79×10 ⁺³	-2.91×10 ³	-6.04×10 ³
	Ortalama	-1.39×10 ⁺³	-1.2×10 ⁺⁴	-9.98×10 ⁺³	-2.91×10 ³	-5.87×10 ³
f_9	En iyi	Yok	3.7689	35.5423	8.98	7.95
	Ortancası	Yok	5.6897	56.5689	15.88	15.61
	Ortalama	Yok	5.9876	55.1256	15.65	15.72
f_{10}	En iyi	Yok	1.3425	0.0034	2.87×10 ⁻⁹	6.48×10 ⁻¹¹
	Ortancası	Yok	2.1342	0.0085	3.87×10 ⁻⁹	9.14×10 ⁻¹¹
	Ortalama	Yok	2.1458	0.0089	3.44×10 ⁻⁹	9.29×10 ⁻¹¹
f_{11}	En iyi	Yok	1.0479	0.000611	1.43	0
	Ortancası	Yok	1.1345	0.0076	3.78	1.1826
	Ortalama	Yok	1.1578	0.0123	3.88	1.0981
f_{12}	En iyi	0.1865	0.0139	0.000667	6.65×10 ⁻²⁰	2.66×10 ⁻²²
	Ortancası	0.1865	0.0378	0.2311	2.54×10 ⁻¹⁹	1.32×10 ⁻²¹
	Ortalama	0.8038	0.0562	0.2345	0.0498	5.44×10 ⁻²⁰
f_{13}	En iyi	1.97	0.0251	1.34×10 ⁻³¹	5.67×10 ⁻³³	2.43×10 ⁻²¹
	Ortancası	1.97	0.0367	2.2×10 ⁻²³	2.54×10 ⁻³²	4.82×10 ⁻²¹
	Ortalama	2.57	0.0872	3.11×10 ⁻¹⁸	2.89×10 ⁻³²	6.30×10 ⁻²¹

Çok modlu ve yüksek boyutlu fonksiyonlar için Çizelge 3.33 incelendiğinde özellikle F10, F11 ve F12 test fonksiyonlarında sonuçların ve performansın dikkat çekici olduğu

görülmektedir. Diğer önerilen algoritma sonuçlarında olduğu gibi sadece F13 test fonksiyonunda önerilen operatör EİAYAA da standart YAA ve PSO dan daha kötü sonuçlar elde edilmiştir.



Şekil 3.12. YAA ile EİAYAA çok modlu fonksiyonlarda yakınsama özelliklerinin karşılaştırılması.

(a) F8 fonksiyonu (b) F9 fonksiyonu (c) F10 fonksiyonu (d) F11 fonksiyonu

Şekil 3.12'de F8, F9, F10 ve F11 fonksiyonlarının döngü ilerledikçe en iyilerin ortalamalarının değişimleri verilmiştir. Küresel değeri yakalamada ve küresel değere

yaklaşım hızında ve hassas arama özelliği bakımından standart YAA dan daha iyi olduğu ilk dört grafikte görülmektedir. Örneğin Şekil 3.12 (a) F8 fonksiyonunda EİAYAA operatörü küresel minimum değerine 200 ü döngülerde yaklaşırken, YAA algoritmasında 250'li döngülerde yaklaşmaktadır. En iyi sonuç değerini verme açısından Şekil 3.12 (c) F10 fonksiyonunu gözlem altına alacak olursak, EİAYAA 10^{-11} değerlerine ulaşırken YAA algoritmasında 10^{-8} değerlerinde kalmıştır.

Çizelge 3.34. EİAYAA F10 için Wilcoxon işaretli sıra testi sonucu.

	YAA	EİAYAA
Örnekleme Sayısı	30	30
Sıra değerlerinin top	1365.0	465.0
Ortalama sıra	45.5	15.5
T	0.0	900.0
Örnekleme büyüklüğü normal dağılımı kullanmak için yeterince büyük		
Ortalama		450.0
Z	6.6456	6.6456
p-değeri (tekli)		0.00000
p-değeri (ikili)		0.00000

Çizelge 3.34'de görüldüğü gibi işaretli sıra testi F10 fonksiyonuna uygulanmıştır. Burada YAA ve EİAYAA arasında bu fonksiyon için belirgin farklar vardır ve geliştirilen algoritma güçlüdür. Tablonun en altındaki iki karşılaştırma p değeri 0.05 değerinden küçük çıkmıştır.

Çizelge 3.35. EİAYAA çok modlu Wilcoxon işaretli sıra testi sonuçları.

B. Fonk.	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}
p-değeri (2)	0.29727	0.00000	0.00000	0.00000	0.00002	0.00001

Çizelge 3.35'de ise, sıra testinin bütün parametreleri verilmeyip sadece hipotezi sonucunu belirleyen p değerlerine yer ayrılmıştır. F8 fonksiyonu hariç, diğer fonksiyonlarda sıfır hipotezinin reddi gerçekleşmiştir. F8'deki iyileşme çok küçük oranda olmasından dolayı sıralı test içerisinde olumlu yansımamıştır.

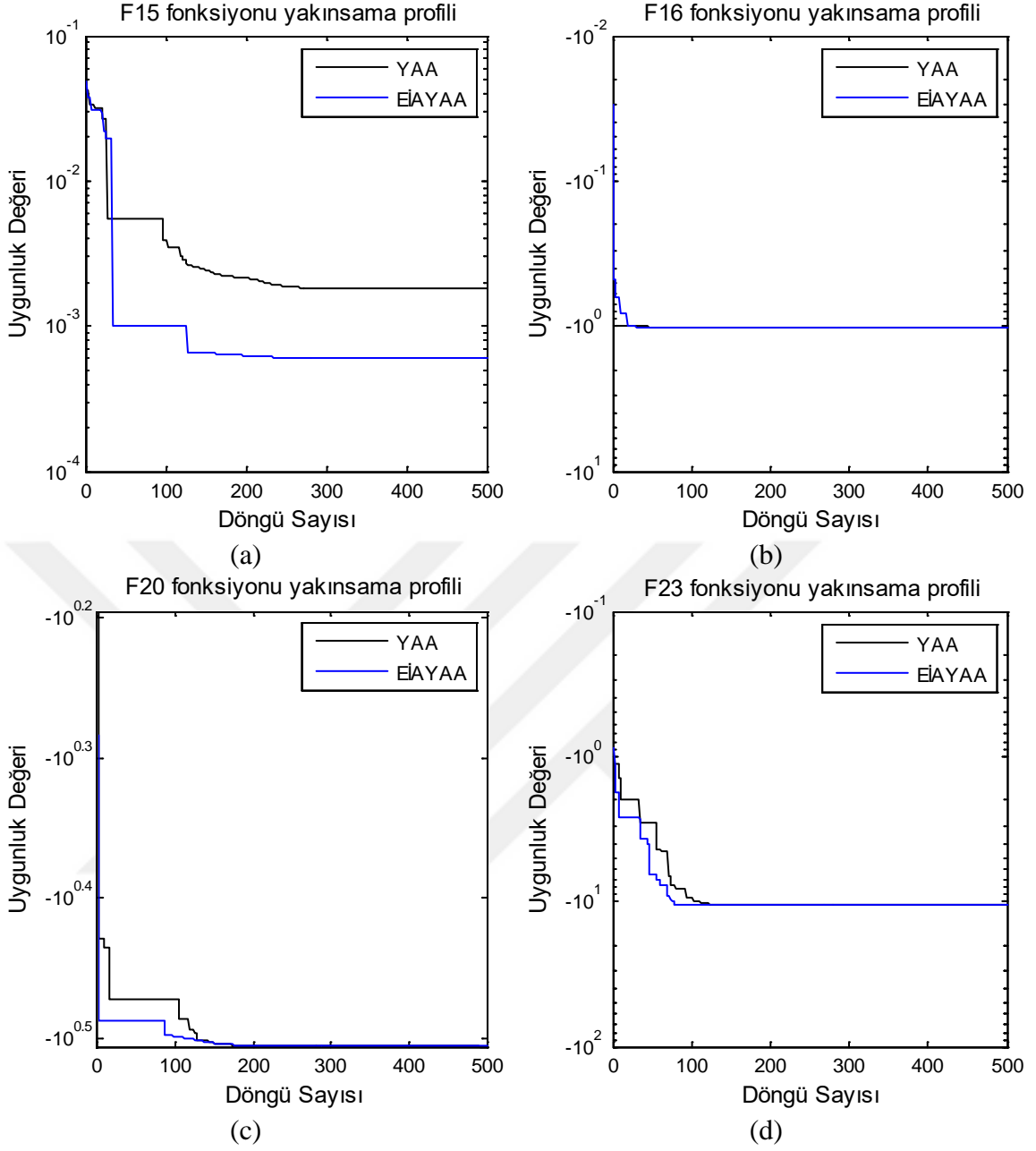
Çizelge 3.36 EİAYAA operatöründe standart sapma değerinin küçük çıkmasından dolayı, değerlerin birbirine yakın olduğunu, YAA ya göre her çalıştırma da küresel minimum değerine yaklaşma oranının daha iyi olduğu tespiti yapılabilir.

Çizelge 3.36. YAA ve EİAYAA arasında çok modlu fonksiyonlar için standart sapma ve işlem süresi karşılaştırması.

K. F.		f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}
S. S.	YAA	6.66.10+2	8.300	1.08.10-9	5.6294	0.228	0.0047
	EİAYAA	6.07.10+2	4.034	1.10.10-11	0.6934	0.0636	0.0044
İ. Z.	YAA	0.0041	0.0040	0.0041	0.0042	0.0051	0.0051
	EİAYAA	0.0071	0.0083	0.0076	0.0072	0.0084	0.0090

Çizelge 3.37. EİAYAA düşük boyutlu fonksiyonların minimize sonuçları.

K.F.		Direk [106]	GA [107]	PSO [108]	YAA [4]	EİAYAA
f_{14}	En iyi	0.998	0.9980	0.9980	0.9980	0.9980
	Ortancası	0.998	0.9980	0.9980	3.30	2.5090
	Ortalaması	6.85	0.9980	0.9980	4.72	3.6231
f_{15}	En iyi	3.7×10^{-3}	0.0011	3.07×10^{-4}	0.0016	3.36×10^{-4}
	Ortancası	3.7×10^{-3}	0.0017	7.02×10^{-4}	0.0022	0.00134
	Ortalaması	0.0662	0.0040	0.0028	0.0018	0.00159
f_{16}	En iyi	-1.0316	-1.0313	-1.0316	-1.0316	-1.0316
	Ortancası	-1.0316	-1.0315	-1.0316	-1.0316	-1.0316
	Ortalaması	-0.9735	-1.0313	-1.0316	-1.0316	-1.0316
f_{17}	En iyi	55.60	0.3979	0.3979	0.3979	0.3979
	Ortancası	55.60	0.3980	0.3979	0.3979	0.3979
	Ortalaması	57.53	0.3996	0.3979	0.3979	0.3979
f_{18}	En iyi	3.00	3.00	3.00	3.00	3.00
	Ortancası	3.00	5.70	3.00	3.00	3.00
	Ortalaması	13.09	5.75	3.00	3.00	3.00
f_{19}	En iyi	-1.8996	-3.8628	-3.8628	-3.8628	-3.8628
	Ortancası	-1.8996	-3.8628	-3.8628	-3.8628	-3.8628
	Ortalaması	-1.8334	-3.8627	-3.8628	-3.8628	-3.8628
f_{20}	En iyi	-1.1238	-3.3220	-3.3220	-3.3220	-3.3220
	Ortancası	-1.1238	-3.3217	-3.2031	-3.3220	-3.3220
	Ortalaması	-0.7505	-3.3098	-3.2369	-3.3220	-3.2862
f_{21}	En iyi	-6.8405	-10.1527	-10.1532	-10.1532	-10.1532
	Ortancası	-6.8405	-2.6824	-5.1008	-10.1486	-10.1530
	Ortalaması	-3.6348	-5.6605	-5.7496	-9.9541	-7.4141
f_{22}	En iyi	-7.0871	-10.4029	-10.4029	-10.4029	-10.4029
	Ortancası	-7.0871	-10.3932	-10.4029	-10.4029	-10.4029
	Ortalaması	-4.3715	-7.3421	-9.1118	-10.4008	-10.4029
f_{23}	En iyi	-7.2170	-10.5364	-10.5364	-10.5364	-10.5364
	Ortancası	-7.2170	-4.5054	-10.5364	-10.5364	-10.5364
	Ortalaması	-4.4568	-6.2541	-9.7634	-10.5364	-10.5364



Şekil 3.13. YAA ve EİAYAA arasında çok modlu düşük boyutlu fonksiyonlarda yakınsama özelliklerinin karşılaştırılması.

(a) F16 fonksiyonu (b) F18 fonksiyonu (c) F20 fonksiyonu (d) F23 fonksiyonu

Çizelge 3.37'de çok modlu düşük boyutlu test fonksiyonlarında önerilen çalışmanın sonuçları verilmiştir. Çok modlu düşük boyutlu fonksiyonlarda yerel minimum noktası çok fazladır, ancak boyutların küçük değer verilmesinden dolayı, işlem süresi, yakınsama hızı ve arama sonuçları diğer iki fonksiyon tipine göre literatürdeki sonuçlara yakın veya eşittir. Bundan dolayı, bu tip fonksiyon özelliklerinden dolayı olumlu sonuçlar çok küçük oranlarda gerçekleşmiştir. Çizelgeye bakıldığında, sonuçları

virgülden sonra dört basamak ilerlettiğimiz için ve sonuçlarında onlu basamaklarda farklılık gösterdiği için küçük olumlu sonuçları yakalamak zordur. YAA ve EİAYAA arasındaki büyük değişimleri F14 ve F15 fonksiyonlarında görmek mümkündür. F20 ve F21 fonksiyonları sadece en iyilerin ortalaması karşılaştırmasında standart YAA ya göre daha kötü sonuçlar vermiştir. Negatif değer veren sonuçlarda böyle küçük bir dezavantajın olduğu tespiti yapılmıştır.

Şekil 3.13 de F15 ve F23 için EİAYAA algoritmasının daha iyi sonuç ürettiği ve F23 fonksiyonunda yakınsama hızının daha iyi olduğu görülmektedir. Bunun yanında Şekil 3.13 (b) F16 fonksiyonun grafiği iki algoritmanın sonuç değerleri aynıdır. Bunun sebebini düşük boyutlarda çalışılması ve ilgili fonksiyonun gerçek minimum değerinin iki algoritma tarafından yakalanmasıdır. Şekil 3.13 (c) F20 fonksiyonunun grafiğinde ise çok küçük oranlarda olumlu iyileşmenin olduğu görülmektedir.

Çizelge 3.38. EİAYAA F18 için Wilcoxon işaretli sıra testi sonucu.

	YAA	EİAYAA
Örnekleme Sayısı	30	30
Sıra değerlerinin top	955.0	875.0
Ortalama sıra	31.8	29.2
T	410.0	490.0
Örnekleme büyüklüğü normal dağılımı kullanmak için yeterince büyük		
Ortalama		450.0
Z	0.5849	0.5849
p-değeri (tekli)		0.27932
p-değeri (ikili)		0.55865

Çizelge 3.39. EİAYAA F15 için Wilcoxon işaretli sıra testi sonucu.

	YAA	EİAYAA
Örnekleme Sayısı	30	30
Sıra değerlerinin top	1077.0	753.0
Ortalama sıra	35.9	25.1
T	288.0	612.0
Örnekleme büyüklüğü normal dağılımı kullanmak için yeterince büyük		
Ortalama		450.0
Z	2.3877	2.3877
p-değeri (tekli)		0.00848
p-değeri (ikili)		0.01695

Çizelge 3.38'de işaretli sıra testi F18 fonksiyonuna uygulanmıştır. Her iki algoritmadaki verilerin birbirlerine çok yakın olmasından kaynaklanan durumdan dolayı p değeri ikili karşılaştırma için 0.55865 gibi çok büyük bir değere sahiptir, sıfır hipotezi doğrulanmıştır. Çizelge 3.39'da ise yukarıdaki durumun tam zıddı olarak, sıfır hipotezin reddini gerçekleştirildiği F15 fonksiyonuna yer verilmiştir. Burada YAA ve EİAYAA arasında belirgin farkın ve iyileşmenin olduğunun göstergesi p değerinin 0.05 değerinden çok küçük çıkması gösterilir.

Çizelge 3.40. EİAYAA çok modlu düşük boyutlu fonksiyonlar için Wilcoxon işaretli sıra testi sonuçları.

K. F.	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}	$f_{19}-f_{20}$	f_{21}	f_{22}	f_{23}
p-(2)	0.8295	0.01695	0.487	2.00000	0.55865	2.00000	0.351	2.0000	0.26236

Çizelge 3.40'da sıra testinin bütün parametreleri verilmeyip sadece hipotezi sonucunu belirleyen p değerlerine yer verilmiştir. Sadece F15 fonksiyonunda sıfır hipotezinin reddi gerçekleşirken, diğerlerinde sıfır hipotezi gerçekleşmiştir. Bunun sebebi ise Çizelge 3.40'da değerlerin aynı veya çok yakın olduğu ve bundan dolayı z değeri çok küçük çıkarak p değerinin 2.00000 olması ile sonuçlanmıştır. Önem seviyesi $\alpha=0.05$ den büyüktür sıfır hipotez doğrulanmıştır, iki algoritma arasında fark yoktur.

Çizelge 3.41. YAA ve EİAYAA arasında standart sapma ve işlem süresi karşılaştırması.

K. F.	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}	f_{21}	f_{22}	f_{23}
S. S.	2.7206	0.0004	4.9.10-16	0	0	0	0	3.2544	0	0
	2.2655	0.0000	4.1.10-16	0	0	0	0	3.1524	0	0
İ. Z.	0.0048	0.0030	0.0027	0.002	0.002	0.002	0.002	0.0029	0.002	0.002
	0.0077	0.0052	0.0049	0.004	0.005	0.004	0.005	0.0033	0.003	0.003

Çizelge 3.41 EİAYAA operatörü sonuçlarında standart sapma daha küçüktür, sonuçlar birbirine yakındır. Ayrıca, YAA ya göre her çalışma da küresel minimum değerine yaklaşma oranının daha iyidir. Tek bir döngü için işlem süresi YAA a göre büyüktür, ancak KHYAA göre süreler daha kısadır.

4. SONUÇLAR VE ÖNERİLER

Bugüne kadar yapılan YAA uygulamalarında yerel minimuma takılma, grupsal davranış bozukluğu ve hassas arama kabiliyet yetersizliğinden dolayı mühendislik bakış açısı ile bu algoritmalara uyumluluk ve sağlamlık eklenmesi arzu edilir hale gelmiştir. Bu bağlamda, yapılacak olan çalışma yöntemi ve planı içerisinde, ilk olarak YAA'nın farklı parametreler için test fonksiyonlarına karşı gösterdiği olumlu veya olumsuz davranışlar incelenmiş ve dezavantajları ortaya konmuştur. Buradan elde edilen sonuçlar ışığında, yerel minimuma takılma durumunu ortadan kaldıran KSYAA, grupsal davranış özelliğini artıran KHYAA ve hassas arama özelliğini sağlayan EİAYAA operatörleri gerçekleştirilmiştir.

Parametrelerin incelenmesi aşamasında, G_0 değerinin 100 ile 1000 aralığında en iyi sonuçlar vermiştir. Yerçekimi parametresinin azalma şeklini belirleyen α parametresinin çok küçük veya çok büyük verilmesi optimizasyon sonuçlarına iyi bir şekilde yansımamıştır. Azalma α parametresi 10-20 aralığında en iyi sonuçlar elde edildiği görülmüştür. Ajan sayısı değerlendirmesinde ise 30 ve 100 gibi orta değerli ajan sayılarında kayda değer en iyi sonuçlar elde edilmesinin yanında, geçen sürenin normal olduğu söylenebilir. Ajan sayısının çok az verilmesi yerele takılma, çok büyük değerler verilmesi de optimizasyon süresinin çok uzun olması ile sonuçlanmıştır. Kullanıcı geçen zaman parametresini düşünerek uygulamalarda kısa zamanda sonuca ulaşmak isterse döngü sayısını 500 civarında, geçen sürenin kullanıcı tarafından önemi olmayan uygulamalarda ise döngü sayısının 1000 ve üzeri verilmesinin doğru olacağı sonucuna varılmıştır. İlgili ajana etkiyen kuvvet hesabında, K_{best} kullanılması durumunda optimizasyon süresinin yaklaşık olarak yarıya düştüğünü ve özellikle en iyi sonuçları K_{best} kullanılarak elde edildiği görülmüştür. Ancak azalma oranının farklı yaklaşımlar uygulayarak incelenmesi ayrı bir çalışma konusu olarak değerlendirilmiştir. Önce hızlı daha sonraki döngülerde yavaş K_{best} düzenlemesi tercih edilebilir. Son olarak, son_ajan katsayısında, küresel en küçük değere giderken son döngülerde sadece kütlesi çok büyük olan ajanların devreye sokulması, yaklaşım hızı açısından önemlidir. Yaklaşım hızını düşürür, son döngülerde hassas ince arama yapar. Bu yüzden son_ajan yüzde değerlerinin toplam ajan sayısının % 10 gibi küçük değerler verilmesi gerekir.

Elde edilen operatörlerin performanslarının test edilmesi için karşılaştırma tekniği en önemli metottur. Bunun için literatürde önemli ve kullanım alanı geniş olan beş çalışmaya yer verilmiştir. İlki, standart Lipschitzian uygulaması matematiksel yöntemlerden biri sayılmıştır ve direk adı altında karşılaştırma tablolarında kullanılmıştır. Sezgisel algoritmalarından olan GA, PSO ve son olarak da standart YAA referans çalışma olarak alınmıştır.

Elde edilen sonuçların değerlendirilmesinde ikinci yaklaşım olarak Wilcoxon işaretli sıra testi kullanılmıştır. Bu testteki amaç, standart YAA ve geliştirilen operatörlerden elde edilen sayısal verilerin ortalamalarının karşılaştırılmasıdır. Wilcoxon işaretli sıra testi güçlü bir test olup, farklar arasındaki işarete (artı mı eksi mi olduğuna) bakmayıp, farkın miktarını da dikkate alması en önemli özelliklerindedir.

Son değerlendirme yaklaşımında her bir 30 çalışmanın arkasında standart sapma ve tek döngü için işlem sürelerinin ortalaması alınmıştır. Standart sapma verilerin kendi içerisinde nasıl bir yayılım gösterdiğini belirtir. Standart sapmanın küçük olması değerlerin birbirine yakın olduğunu, büyük olması ise değerlerin birbirinden uzak olduğunu gösterir. İşlem süresi tek döngü için hesaplanmıştır ve algoritmanın hızı hakkında fikir vermesi açısından önemlidir.

Önerisi yapılan ilk operatör olan KSYAA da, döngüler ilerledikçe en iyi sonuç değeri değişmiyorsa ajanlar yerel minimuma takılmış veya kötü arama bölgesindedir. Bu durumda yerçekimi parametresi üzerinde kaotik sarsıntı oluşturarak, ajanların pozisyonları daha büyük oranda değiştirilmiştir. Kaotik sarsıntı oluşturulurken, kullanılan kaotik fonksiyonlar yerçekimi parametresini pozitif yönde artıracak şekilde uygulanmıştır. Azaltacak yönde yapılan sarsıntı olumlu sonuçlar vermemiştir.

KSYAA da kullanılan Daire, Lojistik, Okuyucu ve Sinüzoidal kaotik fonksiyonlar 50-250 döngüleri arasında sarsıntı oluşturulmasına izin verilmiş, son %60 kısmında izin verilmemiştir. Bunun sebebi olarak algoritmanın hassas arama yapma özelliğini bozmamak olarak belirtilmiştir. Kaotik fonksiyonlardan Okuyucu kaotik fonksiyonunun sonuçları dikkat çekicidir ve bundan dolayı da bütün deneysel çalışmalarda bu fonksiyon tercih edilmiştir.

KSYAA algoritmasının tek ve çok modlu test fonksiyonlarının her biri için karşılaştırma tablolarına yer verilmiştir. Temel karşılaştırma tabloları incelendiğinde

iyileşmenin büyük oranlarda olduğu tespit edilmiştir. Grafiksel gösterimde incelemeler yaparak bazı fonksiyonların yakınsama hızı yönünde standart YAA ya göre daha hızlı olduğu ve sonuç değerlerinin daha küçük olduğu sonucu çıkartılmıştır.

KSYAA'nın Wilcoxon işaretli sıra testi sonuçlarına göre F1-3, F5, F8-14 toplam 11 fonksiyonda sıfır hipotezinin reddi gerçekleşirken diğer fonksiyonlarda sıfır hipotezi gerçekleşmiştir. Sıfır hipotezi gerçekleşen bu fonksiyonlar için YAA ve KSYAA sonuçlarının hepsinin aynı veya çok yakın olduğu görülecektir. Bundan dolayı z değeri çok küçük çıkarak p değerinin 2.00000 olması ile sonuçlanmıştır. Sıfır hipotezi gerçekleşen fonksiyonlarda algoritmanın kötü olduğu anlamına gelmez, sadece iyileşme küçük oranlarda gerçekleşmiştir.

KSYAA operatöründe F1-6, F9-12 fonksiyonları için standart sapma çok küçük, F19-23 aralığındaki fonksiyonlarda eşit, F7-8, F14-18 değerlerin birbirine yakın olduğu fakat az da olsa önerilen operatörün iyi olduğu belirlenmiştir. Sadece F13 fonksiyonunda kötü sonuçlar elde edilmiştir. Tek bir döngü için işlem süresinin büyük olması kaotik hesaplamalardan kaynaklanmaktadır. Ancak yakınsama hızı standart algoritmaya göre yüksek olmasından dolayı daha düşük döngülerde çalıştırarak bu dezavantaj ortadan kaldırılabılır.

Sonuç olarak KSYAA yerçekimi parametresi üzerinde kaotik sarsıntı oluşturarak, ajanların pozisyonları daha büyük oranda değiştirilebilir ve bu durum aramaya olumlu yansıdığı görülmüştür. KSYAA standart YAA algoritmasına göre performansı yerel minimuma takılmama ve kolay çıkma noktasında performansı yüksektir.

Önerilen ikinci operatör olan KHYAA, grupsal davranışın dışında veya uzakta kalmış ajanların hızlarını artırmak ve grup içerisine düşürmek için kurtulma hızının negatif yönde eklenmesi ile sağlanabileceği gerçekleştirilmiştir. Böylece arama içerisinde sürü ve grupsal yaklaşımı mükemmelleştirme çalışması sağlanmıştır. Elde edilen sonuçlar standart YAA algoritması ile karşılaştırılmış ve değerlendirilmiştir.

KHYAA ilk olarak 23 karşılaştırma test fonksiyonlarına uygulanmış ve literatürde yaygın olarak kullanılan sezgisel ve matematiksel algoritmalar ile karşılaştırılmıştır. Bu karşılaştırmaya göre F1-4, F6, F9-12 ve F15 fonksiyonlarında çok büyük farklar ile iyi sonuçlar vermiştir. Grafiksel incelemede ise F4, F6-8, F10 ve F14-15 fonksiyonlarında yakınsama hızının yüksek olduğu görsel olarak ispatlanmıştır.

Wilcoxon işaretli sıra testi sonuçlarına göre KHYAA F1-4, F7, F9-13 ve F15 fonksiyonlarında sıfır hipotezinin reddi gerçekleşerek 0.05 olan eşik değerin altında çıkmıştır. Diğer fonksiyonlarda iyileşmeni çok küçük oranlarda olmasından dolayı sıfır hipotezi gerçekleşmiştir.

KHYAA verilerin kendi içerisinde nasıl bir yayılım gösterdiğini belirten standart sapma özelliği ile de dikkat çekicidir. Bilindiği gibi standart sapmanın küçük çıkması tercih edilir, çünkü küçük olması değerlerin birbirine yakın olduğunu ve her 30 çalışmada da düzgün çalıştığını gösterir. F1-6, F9-13, F14-16 ve F20-21 fonksiyonlarında standart sapma değeri standart YAA değerlerine göre küçük çıkmıştır, her çalışmada en iyi sonuçlara yaklaşma oranının yüksek olduğunu göstermektedir.

Sonuç olarak, KHYAA grupsal davranışın uzak bölgelerinde yer alan ajanlara en iyi ajana doğru kurtulma hızları uygulanmıştır. Böylece arama içerisinde sürü ve grupsal yaklaşımı mükemmelleştirme çalışması gerçekleştirilmiştir ve bu durum aramaya olumlu yansıdığı görülmüştür.

Tez çalışmasında gerçekleştirilen son operatör EİAYAA da temel yaklaşım en iyi sonuç değerine sahip ajanın kütlelerinin büyük olmasından dolayı çok yavaş hareket ettirmeyi sağlamaktır. En iyi kütleyle sahip ajanların yavaş hareket etmelerini sağlamak için etki eden kuvvet hesabında düşük kütleli ajanlar önerilmiştir.

Bu algoritma 7 tek modlu, 6 çok modlu yüksek boyutlu ve 10 adet çok modlu düşük boyutlu test fonksiyonlarına uygulanarak, literatürde çok sık kullanılan matematiksel yöntem, GA, PSO ve standart YAA algoritmaları ile karşılaştırılmıştır. F1-3, F6-7, F10-12 ve F14-15 fonksiyonlarındaki veriler incelendiğinde iyileşmenin büyük oranlarda olduğu tespit edilmiştir. Grafikselleştirme çalışması bize algoritmanın yakınsama hızı hakkında fikir vermektedir. Bu incelemeye göre F4, F10, F11 ve F23 fonksiyonlarında EİAYAA operatörünün çok erken sonuca gittiğini göstermiştir. Yakınsama hızı yönünden önerilen operatör güçlüdür.

Alınan verilerin birebir ve sıralı karşılaştırmasını yapan Wilcoxon işaretli sıra testi EİAYAA sonuçları F1-4, F6-7, F9-13 ve F15 fonksiyonlarında sıfır hipotezinin reddi gerçekleşmiştir. Bu fonksiyonlardan sadece F13 fonksiyonunda standart YAA lehine sıfır hipotezinin reddi gerçekleşmiştir. Yani F13 fonksiyonunda standart YAA verileri daha iyi çıkmıştır.

EİAYAA standart sapma daha küçüktür, sonuçlar birbirine yakındır. Ayrıca, YAA ya göre her çalıştırma da küresel minimum değerine yaklaşma oranının daha iyidir. Tek bir döngü için işlem süresi YAA a göre büyüktür, ancak KHYAA göre süreler daha kısadır. Sonuç olarak EİAYAA döngüler ilerledikçe en iyi değerlere sahip ajanların kütleleri, dolayısıyla hızları yavaşlatılarak hassas arama gerçekleştirilmiştir ve performansı standart YAA göre güçlüdür.

Mühendisliğin optimizasyon tabanlı uygulamalarında, kullanılması gereken temel yapı, hız ve gerçek değere yakınsama özelliğidir. Gerçekleştirilen üç yeni operatör kullanıldığı tüm mühendislik uygulamalarını çok daha esnek ve kullanışlı kılacaktır. Tez sonunda ortaya çıkacak bu esnek ve kullanışlı optimizasyon algoritma yapısı ile, ülkemizdeki mühendislik uygulamalarında tercih edilmesi ve geliştirilmesi sağlanacaktır. Yerel minimuma takılmayan kaotik sarsıntılı KSYAA, grupsal davranış hareketini çok iyi sağlayan kurtulma hızlı KHYAA ve hassas arama özelliği artırılmış EİAYAA olmak üzere üç önemli özellik kazanılmıştır. Bu özellikler sayesinde gerçek yaşam problemi çok daha doğru temsil edilebilir. Tezde elde edilen sonuçlar, konu ile ilgili SCI kapsamındaki dergilerde yayınlanacak ve ülkemizin üretmiş olduğu SCI indeksli bilimsel yayınlara önemli katkılarda bulunulacaktır.

5. KAYNAKLAR

- [1] Reeves C. R., *Modern heuristic techniques for combinatorial problems*, John Wiley & Sons, Inc., (1993).
- [2] Cura T., *Modern sezgisel teknikler ve uygulamaları*. Papatya Yayıncılık Eğitim, (2008).
- [3] Salhi, S., *Heuristic search methods*, Mahwah, NJ: Erlbaum, (1998).
- [4] Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S., GSA: a gravitational search algorithm, *Information sciences*, 179(13) (2009) 2232-2248.
- [5] Karaboğa D., *Yapay Zeka Optimizasyon Algoritmaları*, Nobel Yayın Dağıtım, (2014).
- [6] Hamm S., Lesellier E., Bleton J., & Tchaplal A., Optimization of headspace solid phase microextraction for gas chromatography/mass spectrometry analysis of widely different volatility and polarity terpenoids in olibanum, *Journal of Chromatograph*, (2003).
- [7] Boyd S., Vandenberghe L., *Convex optimization*. Cambridge university press, (2004).
- [8] Modica E., Maximum/minimum problems solved using an algebraic way, *Teaching Mathematics and its Applications*, 29(1) (2010) 41-47.
- [9] Ferguson T. S., Who solved the secretary problem?, *Statistical science*, (1989) 282-289.
- [10] Sethi S. P., Thompson G. L., What is Optimal Control Theory?, *Springer US*, (2000) 1-22.
- [11] Peajcariac J. E., Tong Y. L., *Convex functions, partial orderings, and statistical applications*, Academic Press, (1992).
- [12] Dyer J. S., Fishburn P. C., Steuer R. E., Wallenius J., Zionts S., Multiple criteria decision making, multiattribute utility theory: the next ten years, *Management science*, 38(5) (1992) 645-654.
- [13] Dantzig, G. B., *Linear programming and extensions*, Princeton university press, (1998).

- [14] Markowitz, H., The optimization of a quadratic function subject to linear constraints, *Naval research logistics Quarterly*, 3(1-2) (1956) 111-133.
- [15] Bellman, R., Dynamic programming and Lagrange multipliers, *Proceedings of the National Academy of Sciences*, 42(10) (1956) 767-769.
- [16] Mitchell, M., Forrest, S., & Holland, J. H., The royal road for genetic algorithms: Fitness landscapes and GA performance, *In Proceedings of the first european conference on artificial life Cambridge: The MIT Press. (1992)* (245-254).
- [17] Öztürk S., Kırılgan ve Dayanıklı Resim Damgalama Tekniklerinden Başarımının Zeki Optimizasyon Yöntemleriyle Artırılması, *Doktora Tezi*, Erciyes Üniversitesi, (2009).
- [18] Lee, K. Y., El-Sharkawi, M. A., *Modern heuristic optimization techniques: theory and applications to power systems*, John Wiley & Sons, (2008).
- [19] Glover, F., Tabu search-part I., *ORSA Journal on computing*, 1(3) (1989) 190-206.
- [20] Glover, F., Tabu search-part II., *ORSA Journal on computing*, 2(1) (1990) 4-32.
- [21] Kirkpatrick, S., Vecchi, M. P., Optimization by simulated annealing, *science*, 220(4598) (1983) 671-680.
- [22] Tan, K. C., Li, Y., Murray-Smith, D. J., Sharman, K. C., System identification and linearisation using genetic algorithms with simulated annealing, *In Genetic Algorithms in Engineering Systems: Innovations and Applications*, (1995).
- [23] Eberhart, R. C., Kennedy, J., A new optimizer using particle swarm theory, *In Proceedings of the sixth international symposium on micro machine and human science*, 1 (1995) 39-43.
- [24] Dorigo, M., Maniezzo, V., & Coloni, A., Ant system: optimization by a colony of cooperating agents, *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 26(1) (1996) 29-41.
- [25] Krishnanand, K. N., Ghose, D., Detection of multiple source locations using a glowworm metaphor with applications to collective robotics, *In Swarm Intelligence Symposium SIS 2005. Proceedings 2005 IEEE*, (2005) 84-91.
- [26] Chu, S. C., Tsai, P. W., Pan, J. S., Cat swarm optimization, *In PRICAI 2006: Trends in artificial intelligence*, (2006) 854-858.

- [27] Başbuğ, S., Bakteriyeel Besin Arama Algoritması ile Lineer Anten Dizilerinin Diyagram Sıfırlaması, *Yüksek Lisans Tezi*, Erciyes Üniversitesi, (2008).
- [28] Jiang, M., Yuan, D., & Cheng, Y., Improved artificial fish swarm algorithm, *In Natural Computation, 2009. ICNC'09. Fifth International Conference on IEEE*, 4 (2009) 281-285.
- [29] Yang, X. S., Firefly algorithms for multimodal optimization, *In Stochastic algorithms: foundations and applications*, (2009) 169-178.
- [30] Liu, C., Yan, X., Liu, C., Wu, H., The wolf colony algorithm and its application, *Chinese Journal of Electronics*, 20(2) (2011) 212-216.
- [31] Keskin, M. Ö., Çukurova yöresi kuvarsit işletmelerinde bilgisayar destekli şev dizaynı ve şev stabilize analizleri, *Doktora Tezi*, Çukurova Üniversitesi, (2008).
- [32] Khajehzadeh, M., Taha, M. R., El-Shafie, A., Eslami, M., A modified gravitational search algorithm for slope stability analysis, *Engineering Applications of Artificial Intelligence*, 25(8) (2012) 1589-1597.
- [33] Altschul, S. F., Boguski, M. S., Gish, W., Wootton, J. C., Issues in searching molecular sequence databases, *Nature genetics*, 6(2) (1994) 119-129.
- [34] González-Álvarez, D. L., Vega-Rodríguez, M. A., Gómez-Pulido, J. A., & Sánchez-Pérez, J. M., Applying a multiobjective gravitational search algorithm (MO-GSA) to discover motifs, *In Advances in Computational Intelligence*, (2011) 372-379.
- [35] Askari, H., Zahiri, S. H., Decision function estimation using intelligent gravitational search algorithm, *International journal of machine learning and cybernetics*, 3(2) (2012) 163-172.
- [36] Rashedi, E., Nezamabadi-Pour, H., A stochastic gravitational approach to feature based color image segmentation, *Engineering Applications of Artificial Intelligence*, 26(4) (2013) 1322-1332.
- [37] Sun, G., Zhang, A., A hybrid genetic algorithm and gravitational search algorithm for image segmentation using multilevel thresholding, *In Pattern Recognition and Image Analysis*, (2013) 707-714.
- [38] Pei, J., Liu, X., Pardalos, P. M., Fan, W., Yang, S., Wang, L., Application of an

- effective modified gravitational search algorithm for the coordinated scheduling problem in a two-stage supply chain, *The International Journal of Advanced Manufact*, 70(1-4) (2014) 335-348.
- [39] Doraghinejad, M., Nezamabadi-Pour, H., Mahani, A., Channel assignment in multi-radio wireless mesh networks using an improved gravitational search algorithm, *Journal of Network and Computer Applications*, 38 (2014) 163-171.
- [40] Yuan, X., A hybrid differential evolution method for dynamic economic dispatch with valve-point effects, *Experts systems with applications*, 36 (2009) 4042-4048.
- [41] Swain, R.K., Sahu, N.C., Hota, P.K., Gravitatioanal search algorithm for optimal economic dispatch, *Procedia Technology*, 6 (2012) 411-419.
- [42] Duman, S., Yörükeren, N., Altas, İ. H., A novel modified hybrid PSOGSA based on fuzzy logic for non-convex economic dispatch problem with valve-point effect, *Elektrical Power and Energy Systems*, 64 (2015) 121-135.
- [43] Talaq, J.H., El-Hawary, F., El-hawary, M.E., A Summary of environmental economical dispatch algorithms, *IEEE Trans. on PWRS*, 9 (1993) 1508-1516.
- [44] Zehar K., Sayah, S., Optimal power flow with environmental constraint using a fast successive linear programming algorithm: Application to the algerian power system, *Energy Conversion and Management*, 49 (2008) 3361-3365.
- [45] Özyön, S., Yaşar, C., Temurtaş, H., Parçacık sürü optimizasyon algoritmasının termik birimlerden oluşan çevresel ekonomik güç dağıtım problemlerine uygulanması, *Elazığ 6th International Advanced Technologies Symposium*, (2011) 175-180.
- [46] Shaw, B., Mukherjee, V., Ghoshal, S.P., A novel opposition-based gravitational search algorithm for combined economic and emission dispatch problems of power systems, *Electrical Power and Energy Systems*, 35 (2012) 21-33.
- [47] Güvenç, U., Sönmez, Y., Duman, S., Yörükeren, N., *Combined economic and emission dispatch solution using gravitaitonal search algorithm*, *Scientia Iranica*, 19(6) (2012) 1754-1762.
- [48] Mondal, S., Bhattacharya, A., Dey, S.H., Multi-objective economic emission load dispatch solution using gravitational search algorithm and considering wind power penetration, *Electrical Power and Energy Systems*, 44 (2013) 282-292.

- [49] Jiang, S., Ji, Z., Shen, Y., A novel hybrid particle swarm optimization and gravitation search algorithm for solving economic emission load dispatch problems with various practical constraints, *Electrical Power and Energy Systems*, 55 (2014) 628-644.
- [50] Xia, X., Elaiw, AM., Optimal dynamic economic dispatch of generation: a review, *Electr Power Syst Res*, 80 (2010) 975-986.
- [51] Swain, R.K., Meher, K.C., Mishra, U.C, Dynamic economic dispatch using hybrid gravitational search algorithm, *2nd International Conference on Power, Control and Embedded Systems*, (2012).
- [52] Yuan, X., A novel approach for economic dispatch of hydrothermal system via gravitational search algorithm, *Applied Mathematics and Computation*, 247 (2014) 535-546.
- [53] Shams, M., Rashedi, E., & Hakimi, A., Clustered-gravitational search algorithm and its application in parameter optimization of a low noise amplifier, *Applied Mathematics and Computation*, 258 (2015) 436-453.
- [54] Xu, Y., Zhou, J., Xue, X., Fu, W., Zhu, W., Li, C., An adaptively fast fuzzy fractional order PID control for pumped storage hydro unit using improved gravitational search algorithm, *Energy Conversion and Management*, 111 (2016) 67-78.
- [55] Aziz, M. S. I., Nawawi, S. W., Sudin, S., Wahab, N. A., Faramarzi, M., Yusof, M. A. M., Gravitational Search Algorithm Optimization for PID Controller Tuning in Waste-water Treatment Process, *Jurnal Teknologi*, 73(3) (2015).
- [56] Elsis, M., Soliman, M., Aboelela, M. A. S., Mansour, W., Dual Proportional Integral Controller of Two-Area Load Frequency Control Based Gravitational Search Algorithm, *Telkomnika Indonesian Journal of Electrical Engineering*, 15(3) (2015) 397-406.
- [57] Moura Oliveira, P. B., Pires, E. S., Novais, P., Design of Posicast PID control Systems using a gravitational search algorithm, *Neurocomputing*, 167 (2015) 18-23.
- [58] Sarafrazi, S., Nezamabadi-Pour, H., & Saryazdi, S., Disruption: a new operator in gravitational search algorithm, *Scientia Iranica*, 18(3) (2011) 539-548.
- [59] Li, C., Zhou, J., Xiao, J., & Xiao, H., Parameters identification of chaotic system by

- chaotic gravitational search algorithm, *Chaos, Solitons & Fractals*, 45(4) (2012) 539-547.
- [60] Han, X., Chang, X., A chaotic digital secure communication based on a modified gravitational search algorithm filter, *Information Sciences*, 208 (2012) 14-27.
- [61] Yazdani, S., Nezamabadi-pour, H., Kamyab, S., A gravitational search algorithm for multimodal optimization, *Swarm and Evolutionary Computation*, 14 (2014) 1-14.
- [62] Mirjalili, S., Hashim, S. Z. M., A new hybrid PSO-GSA algorithm for function optimization, In Computer and information application (ICCIA), 2010 international conference on IEEE, (2010) 374-377.
- [63] Yin, M., Hu, Y., Yang, F., Li, X., Gu, W., A novel hybrid K-harmonic means and gravitational search algorithm approach for clustering, *Expert Systems with Applications*, 38(8) (2011) 9319-9324.
- [64] Guo, Z., A Hybrid Optimization Algorithm Based on Artificial Bee Colony and Gravitational Search Algorithm, *International Journal of Digital Content Technology & its Applications*, 6(17) (2012).
- [65] Hatamlou, A., Abdullah, S., Nezamabadi-pour, H., A combined approach for clustering based on K-means and gravitational search algorithms, *Swarm and Evolutionary Computation*, 6 (2012) 47-52.
- [66] Han, X., Quan, L., Xiong, X., Wu, B., Facing the classification of binary problems with a hybrid system based on quantum-inspired binary gravitational search algorithm and K-NN method, *Engineering Applications of Artificial Intelligence*, 26(10) (2013) 2424-2430.
- [67] Gu, B., Pan, F., Modified gravitational search algorithm with particle memory ability and its application, *International Journal of Innovative Computing, Information and Control*, 9(11) (2013) 4531-4544.
- [68] Liu, Y., Ma, L., Improved gravitational search algorithm based on free search differential evolution, *Systems Engineering and Electronics, Journal of*, 24(4) (2013) 690-698.
- [69] Sarafrazi, S., Nezamabadi-pour, H., Facing the classification of binary problems with a GSA-SVM hybrid system, *Mathematical and Computer Modelling*, 57(1)

(2013) 270-278.

- [70] Tsai, H. C., Tyan, Y. Y., Wu, Y. W., Lin, Y. H., Gravitational particle swarm, *Applied Mathematics and Computation*, 219(17) (2013) 9106-9117.
- [71] Jiang, S., Ji, Z., Shen, Y., A novel hybrid particle swarm optimization and gravitational search algorithm for solving economic emission load dispatch problems with various practical constraints, *International Journal of Electrical Power & Energy*, 55 (2014) 628-644.
- [72] Han, X., Quan, L., Xiong, X., & Wu, B., Diversity enhanced and local search accelerated gravitational search algorithm for data fitting with B-splines, *Engineering with Computers*, 31(2) (2015) 215-236.
- [73] Ibrahim, I., Ibrahim, Z., Ahmad, H., Jusof, M. F. M., Yusof, Z. M., Nawawi, S. W., Mubin, M., An assembly sequence planning approach with a rule-based multi-state gravitational search algorithm, *The International Journal of Advanced Manufacturing*, 79(5-8) (2015) 1363-1376.
- [74] Niknam, T., Golestaneh, F., Malekpour, A., Probabilistic energy and operation management of a microgrid containing wind/photovoltaic/fuel cell generation and energy storage devices based on point estimate method and self-adaptive gravitational search algorithm, *Energy*, 43(1) (2012) 427-437.
- [75] Sombra, A., Valdez, F., Melin, P., & Castillo, O., A new gravitational search algorithm using fuzzy logic to parameter adaptation, *In Evolutionary Computation (CEC), 2013 IEEE Congress on IEEE*, (2013) 1068-1074.
- [76] Niknam, T., Narimani, M., Azizipanah-Abarghooee, R., & Bahmani-Firouzi, B., Multiobjective optimal reactive power dispatch and voltage control: a new opposition-based self-adaptive modified gravitational search algorithm, *Systems Journal, IEEE*, 7(4) (2013) 742-753.
- [77] Kumar, J. V., Kumar, D. V., Edukondalu, K., Strategic bidding using fuzzy adaptive gravitational search algorithm in a pool based electricity market, *Applied Soft Computing*, 13(5) (2013) 2445-2455.
- [78] Li, C., Li, H., & Kou, P., Piecewise function based gravitational search algorithm and its application on parameter identification of AVR system, *Neurocomputing*, 124 (2014) 139-148.

- [79] Newton, I., *The Principia: mathematical principles of natural philosophy*, Univ of California Press, (1999).
- [80] Kragh, H., Gravitation and the earth sciences: the contributions of Robert Dicke, *arXiv preprint arXiv,1501.04293*(2015).
- [81] Kreuzer, L. B., Experimental measurement of the equivalence of active and passive gravitational mass, *Physical Review*, 169(5) (1968) 1007.
- [82] Ciufolini, I., Wheeler, J.A., *Gravitation and inertia*, Princeton universitypress, (1995).
- [83] Giancoli, D. C., *Physics: principles with applications (Vol. 5)*, Upper Saddle River, NJ, USA:: Prentice Hall, (1998).
- [84] Akıncı, Ü., *Fonksiyon optimizasyonunda genetik algoritmalar*, Ders Notları, (2006).
- [85] Güvenç, U., Katırcıoğlu, F., GSA Algoritmasının Değişkenlerinin İncelenmesi Ve En Uygun Değerlerin Tespiti, *İleri Teknoloji Bilimleri Dergisi*, 4(1), (2015).
- [86] Sarafrazi, S., Nezamabadi-Pour, H., Saryazdi, S., Disruption: a new operator in gravitational search algorithm, *Scientia Iranica*, 18(3) (2011) 539-548.
- [87] Han, X., Chang, X., A chaotic digital secure communication based on a modified gravitational search algorithm filter, *Information Sciences*, 208 (2012) 14-27.
- [88] Mirjalili, S., Hashim, S. Z. M., A new hybrid PSO-GSA algorithm for function optimization, *In Computer and information application (ICCIA), 2010 international conference on IEEE*, (2010) 374-377.
- [89] Gu, B., Pan, F., Modified gravitational search algorithm with particle memory ability and its application, *International Journal of Innovative Computing, Information and Control*, 9(11) (2013) 4531-4544.
- [90] Wang, G., He, S., A quantitative study on detection and estimation of weak signals by using chaotic Duffing oscillators, *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 50(7) (2003) 945-953.
- [91] Lorenz, E. N., Deterministic nonperiodic flow, *Journal of the atmospheric sciences*, 20(2) (1963) 130-141.
- [92] Ott, E., Grebogi, C., Yorke, J. A., Controlling chaos, *Physical review letters*, 64(11)

- (1990) 1196.
- [93] Liao, T. L., Tsai, S. H., Adaptive synchronization of chaotic systems and its application to secure communications, *Chaos, Solitons & Fractals*, 11(9) (2000) 1387-1396.
- [94] Haykin, S., Li, X. B., Detection of signals in chaos, *Proceedings of the IEEE*, 83(1) (1995) 95-122.
- [95] Yang, L., Chen, T-L., Application of chaos in genetic algorithms, *Commun Theor Phys*, 38 (2002) 168-172.
- [96] Zhenyu, G., Jia L., Gao X., Liu J., Wu F., Self-adaptive chaos differential evolution, *Advances in natural computation*, (2006) 972-975.
- [97] Simon D., Biogeography-based optimization, *IEEE Trans Evol Comput*, 12 (2008) 702-713.
- [98] Du, D, Simon, D ve Ergezer, M., Biogeography-based optimization combined with evolutionary strategy and immigration refusal, *IEEE international conference on systems*, (2009) 997-1002.
- [99] Khajehzadeh, M., Taha, M. R., El-Shafie, A., Eslami, M., A modified gravitational search algorithm for slope stability analysis, *Engineering Applications of Artificial Intelligence*, 25(8) (2012) 1589-1597.
- [100] Jiang, S., Ji, Z., Shen, Y., A novel hybrid particle swarm optimization and gravitational search algorithm for solving economic emission load dispatch problems with various practical constraints, *International Journal of Electrical Power & Energy*, 55 (2014) 628-644.
- [101] Brown, R. G., *Introductory physics I: elementary mechanics*, URI Durham, (2013).
- [102] <http://tr.wikipedia.org>, Erişim Tarihi: 27 Şubat 2016.
- [103] <http://www.brighthub.com/science/space/articles/117429.aspx>, Erişim Tarihi: 27 Şubat 2016.
- [104] Güvenç, U., Katırcıoğlu, F., En iyi ajana özel davranış: Geliştirilmiş yerçekimi arama algoritması, *Ecjse Journal of Science and Engineering*, 3(1) (2016) 143-153.
- [105] Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K., Chen, Y. P., Auger, A., Tiwari, S., *Problem definitions and evaluation criteria for the CEC 2005 special session on*

real-parameter optimization, KanGAL report, **(2005)**.

- [106] Jones, D. R., Perttunen, C. D., & Stuckman, B. E., Lipschitzian optimization without the Lipschitz constant, *Journal of Optimization Theory and Applications*, 79(1) **(1993)** 157-181.
- [107] Haupt, R. L., Haupt, S. E., *Practical genetic algorithms*, John Wiley & Sons, **(2004)**.
- [108] Tsai, H. C., Tyan, Y. Y., Wu, Y. W., & Lin, Y. H., Gravitational particle swarm, *Applied Mathematics and Computation*, 219(17) **(2013)** 9106-9117.
- [109] Garcia, S., A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the cec'2005 special on real parameter optimization, *Journal of Heuristics*, 15(6) **(2009)** 617-644.
- [110] Wilcoxon, F., Individual comparisons by ranking methods, *Biometrics*, 1(6) **(1945)** 80-83.
- [111] Zhang, A., A hybrid genetic algorithm and gravitational search algorithm for global optimization, *Neural Network World*, 25 (3) **(2015)** 53-73.
- [112] Yao, X., Liu, Y. ve Lin, G., Evolutionary programming made faster, *IEEE Trans. Evol. Comput*, 3 **(1999)** 82-102.

6. EKLER

EK-1. TEK MODLU KARŞILAŞTIRMA TEST FONKSİYONLARI

NO	TEK MODLU TEST FONKSİYONLARI	BOYUT
1	$F_1(X) = \sum_{i=1}^n X_i^2$	$[-100,100]^n$
2	$F_2(X) = \sum_{i=1}^n X_i + \prod_{i=1}^n X_i $	$[-10,10]^n$
3	$F_3(X) = \sum_{i=1}^n \left(\sum_{j=1}^i X_j \right)^2$	$[-100,100]^n$
4	$F_4(X) = \max\{ X_i , 1 \leq i \leq n\}$	$[-100,100]^n$
5	$F_5(X) = \sum_{i=1}^{n-1} [100(X_{i+1} - X_i^2)^2 + (X_i - 1)^2]$	$[-30,30]^n$
6	$F_6(X) = \sum_{i=1}^n ([X_i + 0.5])^2$	$[-100,100]^n$
7	$F_7(X) = \sum_{i=1}^n iX_i^4 + \text{random}[0,1)$	$[-1.28,1.28]^n$

EK-2. ÇOK MODLU KARŞILAŞTIRMA TEST FONKSİYONLARI

NO	ÇOK MODLU KARŞILAŞTIRMA TEST FONKSİYONLARI	BOYUT
8	$F_8(X) = \sum_{i=1}^n -X_i \sin(\sqrt{ X_i })$	$[-500,500]^n$
9	$F_9(X) = \sum_{i=1}^n [X_i^2 - 10 \cos(2\pi X_i) + 10]$	$[-5.12,5.12]^n$
10	$F_{10}(X) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n X_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi X_i)\right) + 20 + e$	$[-32,32]^n$
11	$F_{11}(X) = \frac{1}{4000} \sum_{i=1}^n X_i^2 - \prod_{i=1}^n \cos\left(\frac{X_i}{\sqrt{i}}\right) + 1$	$[-600,600]^n$
12	$F_{12}(X) = \frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\}$ $+ \sum_{i=1}^n u(X_i, 10, 100, 4)$ $y_i = 1 + \frac{X_i + 1}{4}$ $u(X_i, a, k, m) = \begin{cases} k(X_i - a)^m X_i > a \\ 0 & -a < X_i < a \\ k(-X_i - a)^m X_i < -a \end{cases}$	$[-50,50]^n$
13	$F_{13}(X) = 0.1 \left\{ \sin^2(3\pi X_1) \right.$ $+ \sum_{i=1}^n \left(\sum_{i=1}^n (X_i - 1)^2 [1 + \sin^2(3\pi X_i + 1)] (X_n - 1)^2 [1 \right.$ $\left. + \sin^2(2\pi X_n) \right\} + \sum_{i=1}^n u(X_i, 5, 100, 4)$	$[-50,50]^n$

EK-3. ÇOK MODLU DÜŞÜK BOYUTLU KARŞILAŞTIRMA TEST FONKSİYONLARI

NO	ÇOK MODLU DÜŞÜK BOYUTLU TEST FONKSİYONLARI	BOYUT
14	$F_{14}(X) = \left(\frac{1}{500} + \sum_{j=1}^{n25} \frac{1}{\sum_{i=1}^2 (X_i - a_{ij})^6} \right)^{-1}$	$[-65.53, 65.53]^2$
15	$F_{15}(X) = \sum_{i=1}^{11} \left[a_i - \frac{X_1(b_i^2 + b_i(X_2))}{b_i^2 + b_i \cdot X_3 + X_4} \right]^2$	$[-5, 5]^4$
16	$F_{16}(X) = 4X_1^2 - 2.1X_1^4 + \frac{1}{3}X_1^6 + X_1X_2 - 4X_2^2 + 4X_2^4$	$[-5, 5]^2$
17	$F_{17}(X) = \left(X_2 - \frac{5.1}{4\pi^2}X_1^2 + \frac{5}{\pi}X_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos X_1 + 10$	$[-5, 10] \times [0, 15]$
18	$F_{18}(X) = [1 + (X_1 + X_2 + 1)^2(19 - 14X_1 + 3X_1^2 - 14X_2 + 6X_1X_2 + 3X_2^2)] \\ * [30 + (2X_1 - 3X_2)^2 * (18 - 32X_1 + 12X_1^2 + 48X_2 \\ - 36X_1X_2 + 27X_2^2)]$	$[-5, 5]^2$
19	$F_{19}(X) = - \sum_{i=1}^4 c_i \exp\left(- \sum_{j=1}^3 a_{ij}(X_j - p_{ij})^2\right)$	$[0, 1]^3$
20	$F_{20}(X) = - \sum_{i=1}^4 c_i \exp\left(- \sum_{j=1}^6 a_{ij}(X_j - p_{ij})^2\right)$	$[0, 1]^6$
21	$F_{21}(X) = - \sum_{i=1}^5 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	$[0, 10]^4$
22	$F_{22}(X) = - \sum_{i=1}^7 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	$[0, 10]^4$
23	$F_{23}(X) = - \sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	$[0, 10]^4$

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : KATIRCIOĞLU, Ferzan
Uyruğu : T.C.
Doğum tarihi ve yeri : 08.01.1974 DÜZCE
Telefon : 0 505 6758390
Faks : 0 380 5240097
E-posta : ferzankatircioglu@duzce.edu.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Yüksek Lisans	Abant İ.B.Ü. Elektrik Eğitimi	2007
Lisans	Marmara Ü. Elektrik Öğretmenliği	1996
Lise	Kastamonu Teknik Lisesi	1992

İş Deneyimi

Yıl	Yer	Görev
2011- Halen	D.Ü. Düzce MYO	Öğretim Görevlisi
2009-2011	D.Ü. Düzce MYO	Müdür Yardımcısı
2006-2011	D.Ü. Düzce MYO	Öğretim Görevlisi
2000-2006	A.İ.B.Ü. Düzce MYO	Öğretim Görevlisi
1998-2000	Cizre Endüstri M. L.	Elektrik Öğretmeni
1996-1998	Çerkeş 19 Mayıs ÇPL	Elektrik Öğretmeni

Yabancı Dil

İngilizce (ÜDS/KPDS/TOEFL : ÜDS 68.75)

Yayınlar

1. Güvenç, U., Katircioğlu, F., En iyi ajana özel davranış: Geliştirilmiş yerçekimi arama algoritması. *Ecjse Journal of Science and Engineering*,3(1), (2016) pp.143-153
2. Güvenç, U., Katircioğlu, F., GSA Algoritmasının Değişkenlerinin İncelenmesi Ve En Uygun Değerlerin Tespiti, *İleri Teknoloji Bilimleri Dergisi*, 4(1), (2015).
3. Koyuncu, I., Cetin, O., Katircioğlu, F., Tuna, M., Edge dedection application with FPGA based Sobel operator. *In Signal Processing and Communications Applications Conference (SIU), 2015 23th* (pp. 1829-1832), IEEE,(2015).

4. Demirci, R., Katircioglu, F., Segmentation of color images based on relation matrix.
In 2007 IEEE 15th Signal Processing and Communications Applications,(2007).

