



**TARIM ARAZİSİ VERİMLİLİĞİ ALGORİTMASININ
BAŞLANGIÇ POPÜLASYONUNUN
KAOTİK HARİTALARLA OLUŞTURULMASI**

Meral KAYA

**Yüksek Lisans Tezi
Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Prof. Dr. Ahmet Bedri ÖZER
AĞUSTOS-2019**

T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

TARIM ARAZİSİ VERİMLİLİĞİ ALGORİTMASININ
BAŞLANGIÇ POPÜLASYONUNUN
KAOTİK HARİTALARLA OLUŞTURULMASI

YÜKSEK LİSANS TEZİ

Meral KAYA

(082129105)

Tezin Enstitüye Verildiği Tarih: 16 Temmuz 2019

Tezin Savunulduğu Tarih: 2 Ağustos 2019

Tez Danışmanı : Prof. Dr. Ahmet Bedri ÖZER (Fırat Üniversitesi)

Diğer Jüri Üyeleri : Doç. Dr. İlhan AYDIN (Fırat Üniversitesi)

Dr. Öğr. Üyesi Soner KIZILOLUK (Munzur
Üniversitesi)

AĞUSTOS-2019

ÖNSÖZ

Bu çalışmamda bilgisiyle, görüşleriyle ve sabrıyla bana yardımcı olan tez danışmanım Sayın Prof. Dr. Ahmet Bedri Özer ‘e teşekkürlerimi sunmak istiyorum. Ayrıca yardımlarını benden esirgemeyen Dr. Öğr. Üyesi Soner Kızılloluk ‘a teşekkür ederim.

Hayatımın her alanında ilgi, anlayış ve her türlü desteği ile yanımda olan sevgili eşim Caner Kaya ‘ya ve kızlarım İdil ve Zeynep ‘e ayrıca teşekkür ediyorum.

Meral KAYA
ELAZIĞ-2019

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ	II
İÇİNDEKİLER.....	III
ÖZET	V
SUMMARY	VI
ŞEKİLLER LİSTESİ	VII
TABLolar LİSTESİ	VIII
KISALTMALAR.....	IX
SEMBOLLER LİSTESİ	X
1. GİRİŞ	1
1.1. Genel Bilgiler.....	1
1.2. Literatür Taraması.....	2
1.3. Tezin Amacı ve Kapsamı	3
2. OPTİMİZASYON.....	4
2.1. Sezgisel Optimizasyon Algoritmaları	5
2.1.1. Bakteriyel Besin Arama Algoritması	8
2.1.2. Yerçekimsel Arama Algoritması	8
2.1.3. Gri Kurt Optimizasyon Algoritması	9
2.1.4. Ateş Böceği Algoritması	9
2.1.5. Emperyalist Yarışmacı Algoritması	10
3. TARIM ARAZİSİ VERİMLİLİĞİ ALGORİTMASI.....	11
3.1. Tarım Arazisi Verimliliği Algoritması	11
3.1.1. Başlangıç Popülasyonunun Oluşturulması	13
3.1.2. Toprak Kalitesinin Belirlenmesi	15
3.1.3. Yerel ve Genel Hafızaların Güncellenmesi	16
3.1.4. Her Bölümdeki Toprak Kalitesinin Değerlendirilmesi	17
3.1.5. Toprağın Birleşimi ve Son Koşullar	18
4. KAOTİK HARİTALI TARIM ARAZİSİ VERİMLİLİĞİ ALGORİTMASI	19
4.1. Kaotik Haritalar	19
4.2. Kaotik Haritalı Tarım Arazisi Verimliliği Algoritması	23
4.2.1. Tüm Boyutların Kaotik Haritalar ile Üretildiği Birinci Yaklaşım	23
4.2.2. Tüm Popülasyonun ve Tüm Boyutların Kaotik Haritalar ile Üretildiği İkinci Yaklaşım	24

4.2.3.	Tüm Popülasyonun Tek Boyutlu Şekilde Kaotik Haritalar ile Üretildiği Üçüncü Yaklaşım	25
5.	KALİTE TESTİ FONKSİYONLARI İLE YAPILAN DENEYLER.....	26
5.1.	Kalite Testi Fonksiyonları.....	26
5.1.2.	Sphere Fonksiyonu.....	27
5.1.3.	Alpine Fonksiyonu	27
5.1.4.	Griewank Fonksiyonu	28
5.1.5.	Levy Fonksiyonu	29
5.1.6.	Quintic Fonksiyonu.....	30
5.1.7.	Rastrigin Fonksiyonu	31
5.1.8.	Rosenbrock Fonksiyonu.....	32
5.2.	Kaotik Haritalı Tarım Arazisi Verimliliği Algoritmalarının Test Sonuçları	33
5.2.1.	Tüm Boyutların Kaotik Haritalar ile Üretildiği Birinci Yaklaşım Test Sonuçları	33
5.2.2.	Tüm Popülasyonun ve Tüm Boyutların Kaotik Haritalar ile Üretildiği İkinci Yaklaşım Test Sonuçları	40
5.2.3.	Tüm Popülasyonun Tek Boyutlu Şekilde Kaotik Haritalar ile Üretildiği Üçüncü Yaklaşım Test Sonuçları	47
6. SONUÇ	55	
KAYNAKLAR.....	56	
ÖZGEÇMİŞ	59	

ÖZET

Optimizasyon; belirli koşullar altında bir problemin olabilecek bütün çözümleri arasından en iyi çözümü seçme işi olarak tanımlanmaktadır. Optimizasyon problemlerinin çözümünde kullanılan klasik yöntemler, gerçek hayat problemlerinin çözümünde yetersiz kalabilmektedir. Ayrıca problemin boyutuyla orantılı olarak, çözüm süreleri ve çözüm karmaşıklığı aşırı artmaktadır. Klasik yöntemlerin bu dezavantajları, sezgisel araştırma yöntemlerinin gelişmesine yol açmıştır. Bilgisayarın gelişimine paralel olarak son kırk yıldır çalışılmakta olan sezgisel araştırma yöntemleri çoğunlukla doğadan ilham alır. Doğadaki fiziksel süreçler ve yasalar ile canlıların, kusursuz bir tasarıma sahip olması, davranışlarında ve yiyecek ararken en iyi ve en hızlı çözümü bulmaları bilim adamlarının araştırmalarına konu olmakta ve doğadan esinlenen algoritmaların artışına katkı da bulunmaktadır.

Bu tez çalışmasında yakın zamanda önerilen Tarım Arazisi Verimliliği Algoritması detaylı olarak ele alınmıştır. Tarım Arazisi Verimliliği Algoritmasının başlangıç popülasyonu üretilirken kaotik haritalar kullanılmıştır. Kaotik haritalı başlangıç popülasyonları 3 farklı yaklaşım ile oluşturulmuştur. Her yaklaşımda 10 adet kaotik harita kullanılmıştır. Oluşturulan her bir kaotik haritalı algoritma 8 adet kalite testi fonksiyonu ile test edilmiştir.

Yapılan testler sonucunda tüm popülasyonun ve tüm boyutların kaotik haritalar ile üretildiği ikinci yaklaşım da ve popülasyonun tek boyutlu olarak kaotik haritalar ile üretildiği üçüncü yaklaşım ile alınan sonuçlarda genel olarak standart Tarım Arazisi Algoritması ile benzer ya da daha iyi sonuçlar elde edildiği görülmüştür. İkinci yaklaşımda özellikle Çadır (Tent) kaotik haritasının uygulandığı Tarım Arazisi Verimliliği algoritmasının hemen her boyut ve iterasyon için daha başarılı sonuçlar ürettiği gözlemlenmiştir. Üçüncü yaklaşımda ise kaotik haritalı algoritmaların çoğunda başarılı sonuçlar elde edilmiştir.

Anahtar Kelimeler: Sezgisel Optimizasyon Algoritmaları, Tarım Arazisi Verimliliği Algoritması, Kaotik Haritalar

SUMMARY

Generating Initial Population of Farmland Fertility Algorithm with Chaotic Maps

Optimization is defined as choosing the best solution among all possible solutions of a problem under certain conditions. Classical methods used to solve optimization problems may be insufficient to solve real life problems. In addition, proportional to the size of the problem, the solution times and the complexity of the solution is increasing. These disadvantages of classical methods have led to the development of metaheuristic research methods. Parallel to the development of computers, metaheuristic research methods that have been studied for the last four decades are inspired by nature. It is the subject of scientists' researches that living species have an optimum design in the ecosystem and find the optimum solution in their behavior and foraging behaviors, the physical process and laws of nature, contributes to the increase of nature-inspired algorithms.

In this thesis, the recently proposed Farmland Fertility Algorithm is discussed in detail. Chaotic maps were used to generate the initial population of the Farmland Fertility Algorithm. Initial populations with chaotic maps were formed by 3 different methods. 10 chaotic maps were used in each method. Each chaotic map algorithm was tested with 8 quality test functions.

As a result of the tests, it was observed that the results of the approach of Generating the Whole Population and all dimensions with chaotic maps and the third approach in which the population was produced with one dimensional chaotic maps were generally similar or better with the Farmland Fertility Algorithm. In the second approach, it has been observed that the Farmland Fertility Algorithm, in which the tent (Tent) chaotic map is applied, produces more successful results for almost all dimensions and iterations. In the third approach, successful results were obtained in most of the chaotic map algorithms.

Key Words: Meta-Heuristic Optimization Algorithms, Farmland Fertility Algorithm, Chaotic Maps

ŞEKİLLER LİSTESİ

Sayfa No

Şekil 2.1. Sezgisel yöntemler	6
Şekil 2.2. GKO algoritması hiyerarşi zinciri	9
Şekil 3.1. Tarım arazisi, yerel hafıza ve genel hafıza örneği	12
Şekil 3.2. Tarım arazisi verimliliği algoritmasında arama uzayının bölümlendirmesi	14
Şekil 3.3. Tarım arazisi verimliliği algoritması akış şeması	16
Şekil 5.1 Ackley Fonksiyonu	26
Şekil 5.2. Sefere Fonksiyonu	27
Şekil 5.3. Alpine Fonksiyonu	28
Şekil 5.4. Griewank Fonksiyonu	29
Şekil 5.5. Levy Fonksiyonu	30
Şekil 5.6. Quintic Fonksiyonu	31
Şekil 5.7. Rastrigin Fonksiyonu	32
Şekil 5.8. Rosenbrock Fonksiyonu	33
Şekil 5.9. Birinci yaklaşım test sonuçları	38
Şekil 5.10. Birinci yaklaşım 100 iterasyon yakınsama grafikleri	39
Şekil 5.11. Birinci yaklaşım 500 iterasyon yakınsama grafikleri	39
Şekil 5.12. İkinci yaklaşım test sonuçları	45
Şekil 5.13. İkinci yaklaşım 100 iterasyon yakınsama grafikleri	46
Şekil 5.14. İkinci yaklaşım 500 iterasyon yakınsama grafikleri	47
Şekil 5.15. Üçüncü yaklaşım test sonuçları	52
Şekil 5.16. Üçüncü yaklaşım 100 iterasyon yakınsama grafikleri	53
Şekil 5.17. Üçüncü yaklaşım 500 iterasyon yakınsama grafikleri	54

TABLÖLAR LİSTESİ

	<u>Sayfa No</u>
Tablo 4.1. Birinci yaklaşımda kullanılan rastgele sayılar	24
Tablo 4.2. İkinci yaklaşım da kullanılan rastgele sayılar	24
Tablo 4.3. Üçüncü yaklaşımda kullanılan rastgele sayılar	25
Tablo 5.1. 100 iterasyon ve 10 boyut sayısı ile 50 test sonucu ortalaması	34
Tablo 5.2. 500 iterasyon ve 10 boyut sayısı ile 50 test sonucu ortalaması	34
Tablo 5.3. 100 iterasyon ve 20 boyut sayısı ile 50 test sonucu ortalaması	35
Tablo 5.4. 500 iterasyon ve 20 boyut sayısı ile 50 test sonucu ortalaması	36
Tablo 5.5. 100 iterasyon ve 30 boyut sayısı ile 50 test sonucu ortalaması	37
Tablo 5.6. 500 iterasyon ve 30 boyut sayısı ile 50 test sonucu ortalaması	37
Tablo 5.7. 100 iterasyon ve 10 boyut sayısı ile 50 test sonucu ortalaması	40
Tablo 5.8. 500 iterasyon ve 10 boyut sayısı ile 50 test sonucu ortalaması	41
Tablo 5.9. 100 iterasyon ve 20 boyut sayısı ile 50 test sonucu ortalaması	41
Tablo 5.10. 500 iterasyon ve 20 boyut sayısı ile 50 test sonucu ortalaması	42
Tablo 5.11. 100 iterasyon ve 30 boyut sayısı ile 50 test sonucu ortalaması	43
Tablo 5.12. 500 iterasyon ve 30 boyut sayısı ile 50 test sonucu ortalaması	43
Tablo 5.13. 100 iterasyon ve 10 boyut sayısı ile 50 test sonucu ortalaması	48
Tablo 5.14. 500 iterasyon ve 10 boyut sayısı ile 50 test sonucu ortalaması	48
Tablo 5.15. 100 iterasyon ve 20 boyut sayısı ile 50 test sonucu ortalaması	49
Tablo 5.16. 500 iterasyon ve 20 boyut sayısı ile 50 test sonucu ortalaması	49
Tablo 5.17. 100 iterasyon ve 30 boyut sayısı ile 50 test sonucu ortalaması	50
Tablo 5.18. 500 iterasyon ve 30 boyut sayısı ile 50 test sonucu ortalaması	51

KISALTMALAR

FFA	: Farmland Fertility Algorithm(Tarım Arazisi Verimliliği Algoritması)
TAV	: Tarım Arazisi Verimliliği
TAVA	: Tarım Arazisi Verimliliği Algoritması
BFO	: Bacterial Foraging Optimization (Bakteriyel Besin Arama Optimizasyonu)
PSO	: Particle Swarm Optimization (Parçacık Sürü Optimizasyonu)
CSA	: Crow Search Algorithm (Karga Arama Algoritması)
FOA	: Fruit Fly Algorithm (Meyve Sineği Algoritması)
GWO	: Grey Wolf Algorithm (Gri Kurt Optimizasyon Algoritması)
WOA	: Whale Optimization Algorithm (Balina Optimizasyon Algoritması)
FA	: Fire Fly Algorithm (Ateş Böceği Algoritması)
KH	: Krill Herd Algorithm (Krill Sürüsü Algoritması)
BSO	: Bat Swarm Algorithm (Yarasa Sürüsü Algoritması)
GKOA	: Gri Kurt Optimizasyon Algoritması

SEMBOLLER LİSTESİ

N	: Çözüm uzayındaki popülasyon sayısı
k	: Tarım arazisi bölüm sayısı
n	: Her bölümde bulunan çözüm sayısı
j	: Boyut sayısı
Myerel	: Yerel hafıza
Mgenel	: Genel hafıza
$\alpha, \beta, Q,$	: 0 ile 1 arasında bir sayı
X_{ij}	: Popülasyondaki bir çözüm
h	: Ondalık bir sayı
Fit_bölüm	: Her bölümdeki çözümlerin ortalama kalitesi



1. GİRİŞ

1.1. Genel Bilgiler

Bir problemin mevcut tüm çözümleri arasından en iyi çözümü seçme işlemine optimizasyon denilmektedir. Optimizasyon matematikte, mühendislikte, ekonomide, endüstride, işletmelerde ve daha birçok alanda büyük önem taşımaktadır. Problemin matematiksel ifadesi ile modelleme oluşmaktadır. Son yıllarda birçok alanda karar verme sistemlerinin de önemi hızla artmaktadır. Karar verme sürecinde matematiksel modelleme, hesaplama ve algoritmaların becerisi de büyük önem taşımaktadır [1].

Optimizasyon problemlerinin çözümü için klasik ve sezgisel yöntemler bulunmaktadır. Problemlerin çözümünde klasik yöntemlerin yetersiz kalması ile bilim insanları ve araştırmacılar sezgisel yöntemlere yönelmişlerdir.

Bilgisayarın gelişimine paralel olarak 1970'li yıllardan bu yana çalışılmakta olan sezgisel yöntemler doğadan ilham almaktadır. Doğadaki fiziksel süreçler ve yasalar ile canlıların, kusursuz bir tasarıma sahip olması, davranışlarında ve yiyecek ararken en iyi ve en hızlı çözümü bulmaları bilim adamlarının araştırmalarına konu olmakta ve doğadan esinlenen algoritmaların artışına katkı da bulunmaktadır [2]. Literatür de çok fazla sezgisel algoritma bulunmaktadır. Sezgisel yöntemler, klasik yöntemlere göre kolay dönüşebilme, yüksek hesaplama gücü, yerel minimumları aşabilme ve farklı alanlardaki geniş problem yelpazesine cevap verebilme gibi birçok avantaja sahiptir [3]. Sezgisel algoritmalar fizik tabanlı, sosyal tabanlı, müzik tabanlı, sürü tabanlı ve biyoloji tabanlı olmak üzere farklı gruplara ayrıla bilinmektedir [4].

Sezgisel algoritmaların çoğunlukla bir başlangıç popülasyonu bulunmaktadır. Başlangıç popülasyonu oluşturulurken çözümler rastgele dağıtılmaktadır. Algoritmaların sonucunda çıkan en iyi değer çözüm olarak kabul edilmektedir. Son yıllarda sezgisel algoritmaların performansını artırmak için rastgele sayı dizileri yerine kaotik sayı dizileri yoğun bir şekilde kullanılmıştır ve başarılı sonuçlar elde edilmiştir [5,6] Yapılan çalışmalar ile kaotik haritalar, yakınsama hızının artırılması ve yerel optimumdan kaçınma açısından algoritmaların performansını artıran en iyi yöntemlerden biri olmuştur [6].

1.2. Literatür Taraması

Literatürdeki bir çok sezgisel algoritmanın performansını artırmak için kaotik haritalar uygulanmıştır. Bu tezde yeni bir meta sezgisel algoritma olan Tarım Arazisi Verimliliği Algoritmasının başlangıç popülasyonları üretilirken kaotik haritalar kullanılmıştır. Bu bölümde literatürde bulunan bazı kaotik haritalı sezgisel algoritmalara yer verilmiştir.

Gandomi ve Yong [7] Yarasa Algoritmasına kaotik haritalar uygulanmıştır. Çalışmada farklı kaotik haritalarla kıyaslama problemleri üzerine detaylı çalışmalar yapılmıştır. Sonuçlar, bazı kaotik haritalı YA 'ların standart YA 'dan daha iyi performans sergilediğini göstermiştir.

Sayed ve arkadaşları [8] Yusufçuk Algoritmasının yakınsama özelliğini ve verimliliğini artırmak amacıyla Kaotik Yusufçuk Algoritması önerilmiştir. Deneyler Gauss Kaotik haritasının Yusufçuk Algoritmasının performansını önemli ölçüde artırmıştır.

Rizk-Allah ve arkadaşları [9] Kesirli optimizasyon problemlerini çözmek için Kaotik CSA 'yı önermişlerdir. Yapılan çalışmada önerilen Kaotik CSA 'nın performansının standart CSA ya göre daha iyi olduğu yirmi tanınmış kesirli kıyaslama probleminde doğrulanmıştır.

Mitic ve arkadaşları [10] tarafından yapılan çalışmada FOA 'nın performansını geliştirmek için on dört farklı kaotik harita uygulanmış CFOA on adet kalite testi fonksiyonu ile test edilmiştir. Yapılan testler sonucunda Çebişev(Chebyshev) haritalı FOA başarılı sonuçlar vermiştir.

Kohli ve Arora [11] tarafından önerilen Kaotik haritalı GWO algoritması on farklı kaotik harita ve 13 kalite testi kullanılarak yakınsama hızı test edilmiştir. Yapılan çalışmada Kaotik GWO kısıtlı optimizasyon problemlerinde, standart GWO 'dan daha başarılı sonuçlar vermiştir.

Kaur ve Arora [12] WOA 'nın karşılaştığı yavaş yakınsama sorununu çözmek için kaotik haritaları WOA 'ya uygulamışlardır ve kaotik haritaların WOA 'nın performansını artırdığı gözlemlenmiştir. Özellikle Çadır haritada başarılı sonuçlar alınmıştır.

Gandomi ve arkadaşları [13] FA 'nın performansını artırmak için on iki farklı kaotik haritalarla detaylı testler yapılmıştır. Sonuçlar, bazı kaotik FA 'ların standart FA 'dan kesinlikle daha başarılı olduğunu göstermektedir.

Wang ve arkadaşları [14] KH 'nın küresel yakınsama hızını artırmak için Kaotik haritalı CKH algoritmasını çeşitli kaotik haritalar ile test etmişlerdir. Test fonksiyonları ile yapılan testlerde CKH 'nın standart KH 'dan daha başarılı sonuçlar verdiği görülmüştür.

Tanyıldızı ve Demir [15] Altın Sinüs Algoritmasının yerel optimuma takılma riskini azaltmak için kaotik haritaların Altın Sinüs Algoritması üzerindeki etkilerini incelemişlerdir. Yapılan testler özellikle Çebişev ve Sinüzoidal haritaların başarılı olduğunu göstermiştir.

Rezaee Jordehi [16] tarafından yapılan çalışmada BSO 'nun yerel optimumda takılması sorununu çözmek için kaotik haritalı BSO önerilmiştir. Önerilen kaotik haritalı BSO algoritmalarının standart BSO ya başarılı sonuçlar verdiği görülmüştür.

1.3. Tezin Amacı ve Kapsamı

Bu tez çalışmasında yakın zamanda bulunmuş olan sezgisel optimizasyon algoritmalarından Tarım Arazisi Verimliliği Algoritması (Farmland Fertility Algorithm) ayrıntılı olarak incelenmiştir. Algoritmanın başlangıç popülasyonunu üretmek için üç farklı yaklaşım ile farklı kaotik haritalar kullanılmıştır. Oluşturulan kaotik haritalı algoritmalar kalite testi fonksiyonları ile test edilerek, algoritmanın performansı izlenmiştir.

İkinci bölümde optimizasyon ve sezgisel optimizasyon algoritmaları hakkında bilgi verilmiştir. Üçüncü bölümde Tarım Arazisi Verimliliği Algoritması detaylı olarak anlatılmıştır. Dördüncü bölüm Kaotik Haritalar ve Kaotik haritaların Tarım Arazisi Verimliliği Algoritmasına uygulanmasını kapsamaktadır. Beşinci bölüm de kalite testi fonksiyonları ile yapılan tüm deneylere ve altıncı bölüm de sonuçlara yer verilmiştir.

2. OPTİMİZASYON

Tarih öncesi zamanlardan buyana, insanlar kullandıkları sistemlerin performansını artırmak için optimizasyona büyük ilgi duyarlar [17]. Optimizasyon; belirli koşullar altında bir problemin olabilecek bütün çözümleri arasından en iyi çözümü seçme işi olarak tanımlanmaktadır [4].

Matematikte ya da matematiksel programlamada optimizasyon terimi; belirli bir aralıktaki gerçek veya tamsayı değerlerini bir gerçel fonksiyonu minimize veya maksimize etmek amacıyla seçip fonksiyona yerleştirerek herhangi bir probleme çözüm bulma işlemlerinin tümünü ifade eder [18].

Mühendislikte ise optimizasyon terimi; bir sistemde var olan işgücü, zaman, süreçler, hammadde, kapasite, ekipman gibi kaynakların en verimli şekilde kullanılarak maliyetin en aza indirilmesi, kârın maksimum seviyeye çıkartılması, tam kapasite kullanımının sağlanması ve en yüksek verimin alınması gibi belirli amaçlara ulaşmayı sağlayan bir teknoloji olarak tanımlanmaktadır [19].

Optimizasyonun bileşenleri; karar değişkenleri, amaç fonksiyonu ve sınırlayıcılarıdır. Önceden tanımlı değerleri olan karar değişkenlerinin, optimizasyonun performansı üzerinde önemli etkisi bulunmaktadır. Karar değişkenlerinin aldığı değerler, sınırlayıcılar ile belirli bir aralıkta tutulmakta olup, karar değişkenleri ile sınırlayıcılar bir problemi çözmek için gerekli olan amaç fonksiyonunu oluştururlar. Optimizasyon işleminde, amaç fonksiyonun maksimum ya da minimum en iyi çözümünü bulmak hedeflenmektedir [4]. Geçmiş dönemlerden bu yana, karşılaşılan problemlerin çözülebilmesi için birbirinden farklı optimizasyon teknikleri geliştirilmiş ve farklı farklı problemlere uygulanmıştır. Karşılaşılan problemlerin çözümünde en çok kullanılan matematiksel yöntemler, diğer bir ifadeyle klasik yöntemler, uygulanabilirliğinin zor ve performansının yeterli olmamasından ötürü bilim insanlarını başka optimizasyon yöntemleri bulmaya yöneltmiştir [20]. Bilim insanları doğadaki olaylardan esinlenerek klasik optimizasyon yöntemlerine alternatif olarak bir çok algoritma önermişlerdir. Doğadaki olaylardan esinlenerek oluşturulan bu optimizasyon algoritmalarına sezgisel optimizasyon algoritmaları denilmektedir.

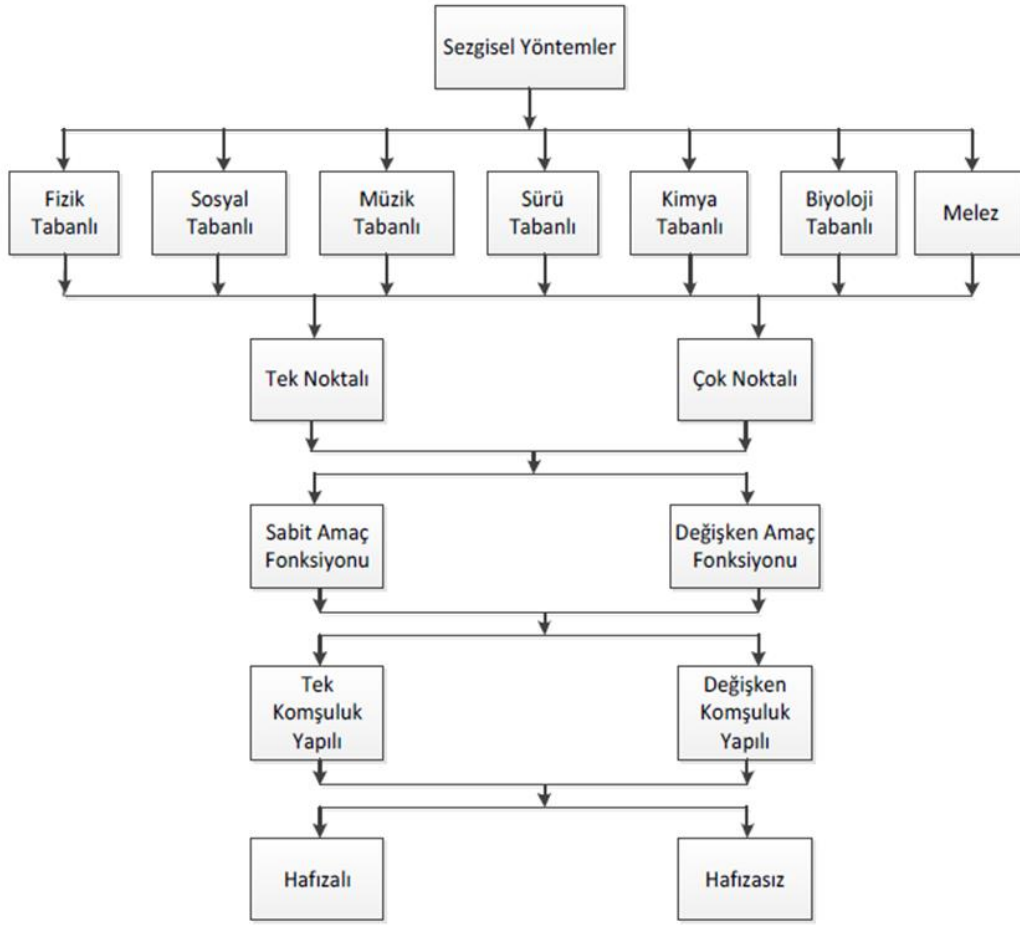
Karmaşık problemleri çözmek için literatürde çok fazla sezgisel optimizasyon algoritması önerilmiştir.

2.1. Sezgisel Optimizasyon Algoritmaları

Optimizasyon problemlerini çözmek amacıyla kullanılan klasik yöntemler, gerçek hayat problemlerinin çözülmesinde istenilen düzeyde olamamıştır. Problemlerin çok boyutlu olması, problemlerde boyut arttıkça karmaşıklığın artması ve çözüm sürelerinin çok uzamasından kaynaklanan sorunlardan ötürü sezgisel algoritmalarla yönelim gittikçe artış göstermiştir. Sezgisel algoritmaların birçoğu doğadaki fiziksel yasaları veya süreçleri temel alırken, büyük bir çoğunluğu da canlı varlıkların sosyal etkileşimlerini örneklemiştir [2].

Doğa bir çok araştırmaya ilham olmaktadır ve sezgisel optimizasyon algoritmaları çoğunlukla doğadan esinlenmiştir. Günümüzde büyük boyutlu problemlerinin çözülmesi için çoğunlukla sezgisel optimizasyon algoritmaları tercih edilmektedir. Sezgisel optimizasyon algoritmaları kesin çözümü garanti edemezler, optimum çözüm elde edilir [21]. Zeki ve sezgisel yaklaşımların zor ve büyük boyutlu problemlerin çözümünde kullanılması üzerine oldukça fazla araştırma yapılmıştır. 1970 yılından bu yana gelişmeye başlayan sezgisel algoritmaların kullanımı son yıllarda şaşırtıcı bir şekilde artmıştır [2]. Sezgisel algoritmalar, optimizasyon problemlerindeki karmaşıklığın artışından ve uygun çözümleri sağlamada matematiksel yöntemlerin yetersiz kalmasından ötürü optimizasyon problemleri için daha uygun olmaktadır. Doğadaki canlıların evrimlerini ve davranışlarını inceleyerek sunulan sezgisel algoritmalar optimizasyon problemleri için icat edilmiştir [22]. Sezgisel optimizasyon algoritmaları, dönüşümleri kolay olduğundan, türev gerektirmediğinden, yerel minimumları aşabildiğinden ve farklı alanlardaki geniş problem yelpazesine cevap verebildiğinden dolayı mühendislik uygulamalarında sıkça rastlanmaya başlanmıştır [23].

Sezgisel algoritmalar doğadaki olayları, fiziksel veya biyolojik davranışları taklit ederek problemleri çözerler. Bu algoritmalarla genel olarak doğa tabanlı algoritmalar diyebiliriz. Sezgisel algoritmalar Şekil 2.1 deki gibi detaylı gruplandırılabilir.



Şekil 2.1. Sezgisel yöntemler [4]

Sezgisel algoritmalarla gerek duyulmasının sebepleri aşağıdaki gibidir.

- Optimizasyon problemi kesin çözümü bulma işleminin tanımlanamadığı bir yapıya sahip olabilir.
- Anlaşılabilirlik açısından sezgisel algoritmalar karar verici açısından çok daha basit olabilir.
- Sezgisel algoritmalar, öğrenme amaçlı ve kesin çözümü bulma işleminin bir parçası olarak kullanılabilir.
- Matematik formülleriyle yapılan tanımlamalarda genellikle gerçek dünya problemlerinin en zor tarafları (hangi amaçlar ve hangi sınırlamalar kullanılmalı, hangi alternatifler test edilmeli, problem verisi nasıl toplanmalı) ihmal edilir. Model parametrelerini belirleme aşamasında kullanılan verinin hatalı olması, sezgisel yaklaşımın üretebileceği alt optimal çözümden daha büyük hatalara sebep olabilir.

Bir problem için geliştirilmiş bir sezgisel algoritma, aşağıdaki faktörler göz önüne alınarak değerlendirilebilmektedir.

- **Çözüm Kalitesi ve Hesaplama Zamanı:** Çözüm kalitesi ve hesaplama zamanı bir algoritmanın etkinliğinin değerlendirilmesi için önemli kriterlerdir. Bundan dolayı bir algoritma, ayarlanabilir parametreler setine sahip olmalı ve bu parametreler kullanıcıya önemlilik açısından hesaplama maliyeti ile çözüm kalitesi arasında bir vurgulamanın yapılabilmesine imkân vermelidir. Diğer bir deyişle, çözüm kalitesi ile hesap zamanı arasındaki ilişki kontrol edilebilmelidir.
- **Kod Basitliği ve Gerçeklenebilirlik:** Algoritma prensipleri basit olmalı ve genel olarak uygulanabilir olmalıdır. Bu durum problem yapısı ile ilgili başlangıçta çok az bilgiye sahip olunması halinde bile algoritmanın yeni alanlara kolaylıkla uygulanabilmesini sağlar.
- **Esneklik:** Algoritmalar modelde, sınırlamalarda ve amaç fonksiyonlarında yapılacak değişiklikleri kolayca karşılayabilmelidir.
- **Dinçlik:** Yöntem, başlangıç çözümünün seçimine bağlı olmaksızın her zaman yüksek kaliteli, kabul edilebilir çözümleri üretebilme kabiliyetine sahip olmalıdır.
- **Basitlik ve Analiz Edilebilirlik:** Karmaşık algoritmalar, esneklik ve çözüm kalitesi açısından basit algoritmalarından daha zor analiz edilebilmektedir. Algoritma kolayca analiz edilebilir olmalıdır.
- **Etkileşimli Hesaplama ve Teknoloji Değişimleri:** Algoritma içinde insan-makine etkileşimini kullanma fikri çoğu sistemde yaygın olarak gerçekleştirilmektedir. Herkesçe bilindiği gibi iyi bir kullanıcı ara yüzü herhangi bir bilgisayar sistemini veya algoritmayı daha çekici yapmaktadır. Bunun en önemli avantajı çözümlerin grafiksel olarak sergilenebilmesidir [3].

Genel amaçlı sezgisel yöntemler temelde 3 ana gruba ayrılmaktadır. Bunlar Fizik, Sosyal ve Biyoloji tabanlı gruplardır. Bunların birleşiminden de melez gruplar oluşmaktadır. Tek bir çözüm ile başlayan algoritma tek noktalı, bir popülasyon ile başlayan algoritma ise çok noktalı olarak ifade edilmektedir. Amaç fonksiyonu değişen algoritmalar değişken amaç fonksiyonlu, sabit olanlar ise sabit amaç fonksiyonlu algoritmalarlardır. Çoğunlukla tek bir komşuluk yapısında çalışan sezgisel algoritmalar, arama işlemini değiştirerek birden fazla yöntemle başka çözüm alanlarına erişmek isteyebilir. Algoritmanın çalışması esnasında herhangi bir değişken veya çözüm muhafaza ediliyorsa hafızalı, edilmiyorsa hafızasız algoritma olarak tanımlanmaktadır.

2.1.1. Bakteriyel Besin Arama Algoritması

Bakteriyel Besin Arama Algoritması (BBAO) popülasyon tabanlı meta sezgisel bir algoritmadır. Temeli E.Coli adı verilen birçok canlının bağırsağında yaşayan bakterinin minimum enerji harcayarak besine ulaşma çabasına dayanmakta olup, karmaşık mühendislik problemlerini çözmek için geliştirilmiş bir hesaplama tekniğidir. BBAO yerel minimumdan kurtulabilme, çok parametrelili optimizasyon problemleri ile rahatlıkla çalışabilme, uygulanması basit ve anlaşılması kolay olma özellikleri ile mühendislik problemlerinin çözümünde başarılı bir şekilde kullanılan ve ilgi gören bir algoritmadır. Bakteriyel besin arama algoritması ilk defa 2002 yılında Pasino tarafından uygulanmıştır.

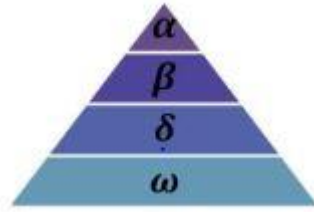
2.1.2. Yerçekimsel Arama Algoritması

Yerçekimsel Arama Algoritması(YAA) Rashedi ve arkadaşları tarafından Newtonun yerçekimi kanunu prensiplerine göre ortaya çıkarılmış bir algoritmadır. Algoritmanın temeli Newton'un hareket ve yerçekimi kanunlarına dayanmaktadır [24]. Algoritma da her nesnenin kendi kütlesi bulunmakta ve her nesne yerçekimi kuvveti ile birbirlerini çeker. Bu çekim esnasında kütlesi en ağır olana doğru bir hareket olur. Her bir nesne bir çözümü göstermektedir ve bu çözümlerin ağırlıkları performansları olarak kabul edilir. Kütlesi en ağır olana doğru gerçekleşen hareket tamamlandığında, ağırlığı en yüksek olan nesneye ait çözüm problemin optimum çözümü olarak kabul edilmektedir. [25].

2.1.3. Gri Kurt Optimizasyon Algoritması

Gri kurt optimizasyon algoritması (GKO) doğadaki gri kurtların liderlik hiyerarşisini ve avlanma mekanizmasını taklit eder. Liderlik hiyerarşisini simüle etmek için alfa, beta, delta ve omega gibi dört gri kurt türü kullanılır [26].

GKO algoritmasında liderlik hiyerarşisi Şekil 2.2 de gösterilmektedir. Alfa grubu kurtlar, en baskın ve sürüye hakim kurtları temsil eder. Beta grubu kurtlar, alfa kurtlar ile diğer kurtlar arasında iletişimi sağlayan kurtlardan oluşur. Omega grubu kurtlar, alfa grubu kurtlar tarafından seçilen ve en alt seviyede yer alan kurtlardır. Delta grubu kurtlar, omega kurtlardan baskın olarak alfa ve beta kurtlar tarafından seçilen kurtlardır [27].



Şekil 2.2. GKO algoritması hiyerarşi zinciri

GKO algoritmasının temel bölümleri avı çevreleme, avlama ve avına saldırmadan oluşmaktadır. Algoritmada alfa grubu kurtlar en iyi çözümü ifade etmektedir.

2.1.4. Ateş Böceği Algoritması

Ateş böceği algoritması 2008 yılında Xin-She Yang tarafından geliştirilen ve ateş böceklerinin parlaklık ve hareket yönlerini baz alan bir optimizasyon algoritmasıdır. Algoritmada ateş böceklerinin cinsiyetleri aynı kabul edilir. Ateş böceklerinin çekiciliği parlaklığı ile doğru orantılı olup, parlaklığı az olan ateşböcekleri parlaklığı yüksek olan ateş böceklerine doğru yönelirler [28]. Ateş böceği algoritmasında, verimli optimal çözümler elde etmek için, verilen bir optimizasyon probleminin amaç fonksiyonu ateş böceği sürüsüne parlak ve daha çekici yerlere gitmede yardım eden yanıp sönen ışık yada ışık şiddeti ile ilişkili olmaktadır. Ateş böceklerinin birbirlerini çekmeleri algoritmanın temelini oluşturmaktadır [29].

2.1.5. Emperyalist Yarışmacı Algoritması

2007 yılında Atashpaz-Gagari ve Lucas tarafından önerilen emperyalist yarışmacı algoritmasının başlangıç popülasyonunu dünyadaki ülkeler oluşturur. Başlangıç popülasyonundaki en iyi ülkelerin birkaçı emperyalist olmak için seçilir ve diğer ülkeler bu seçilen ülkelerin kolonileri olurlar. Koloniler uygun emperyalistlere doğru hareket ederler. Emperyalistin ve kendine ait kolonilerin gücü imparatorluğun gücünü oluşturur. İmparatorluklar arasında bir yarış başlar. Bu yarış süresince güçlü imparatorluklar gücünü artırmaya devam eder, zayıf olanlar ise yok olurlar. Tek bir imparatorluk kalana kadar yarış devam eder. Son aşamada tek imparatorluk kaldığında diğer tüm ülkeler bu imparatorluğun bir kolonisi olurlar ve bu ideal yeni dünya da koloniler ve emperyalist aynı güce ve konuma sahip olacaktır [30,31]

2.1.6. Parçacık Sürü Optimizasyonu Algoritması

Parçacık Sürü Optimizasyon Algoritması sürü zekasına dayalı bir optimizasyon algoritmasıdır. 1995–1996 yıllarında sosyolog-psikolog James Kennedy ve elektrik mühendisi Russel Eberhart tarafından ilk olarak ortaya atılmıştır [4]. PSO kuş ve balık sürülerinin yiyecek arama davranışlarından esinlenerek geliştirilmiştir [31]. Sürü zekasına dayanan algorithmada her bireye parçacık, parçacıklardan oluşan topluluğa da sürü denir [20].

PSO algoritmasında bir dizi sözde parçacık vardır. Her birinin arama alanında bir konumu ve her birinin bir uygunluk değeri vardır. Parçacıklar, çözüm uzayında her boyutta ne kadar hızda hareket ettiğini, şuna kadar elde edilen en iyi uygunluk değerini ve bu değere ulaştığı koordinat bilgilerini hafıza da tutmaktadır. Her ilerlemede, her boyuttaki hızının ve yönünün yeni hali, komşularının en iyi koordinatları ve parçacığın kendi en iyi koordinat bilgilerinin birleşiminden oluşacaktır [4].

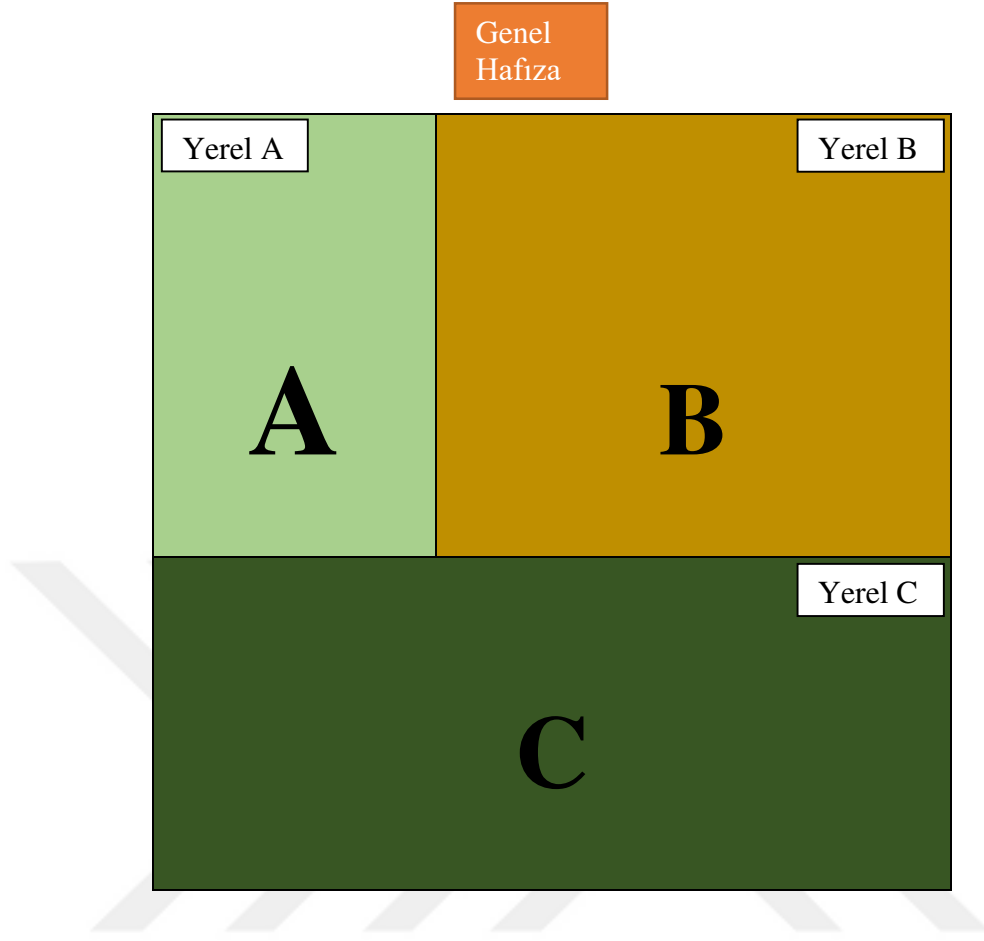
PSO algoritması sürü zekası temelli olduğu için bilimsel ve mühendislik alanlarında geniş bir kullanım alanına sahiptir.

3. TARIM ARAZİSİ VERİMLİLİĞİ ALGORİTMASI

3.1. Tarım Arazisi Verimliliği Algoritması

Tarım Arazisi Verimliliği Algoritması, doğadaki tarım arazilerinin verimliliğinden esinlenerek, 2018 yılında Human Shayanfar ve Farhad Soleimanian Gharehchopogh tarafından önerilen yeni bir meta sezgisel algoritmadır. Human Shayanfar ve Farhad Soleimanian Gharehchopogh tarafından yapılan çalışmada Tarım Arazisi Verimliliği Algoritması, Yapay Arı Koloni Algoritması, Ateş Böceği Algoritması, Harmoni Arama Algoritması, Parçacık Sürü Optimizasyonu, Diferansiyel Evrim Algoritması ve Yarasa Algoritması ile karşılaştırılmıştır. Çalışmada yapılan simülasyonlarda, Tarım Arazisi Verimliliği algoritmasının, diğer algoritmalara göre çoğu kez daha erken yakınsadığı gösterilmiştir. Küçük boyutlu matematiksel optimizasyon problemlerinde, Tarım Arazisi Verimliliği algoritmasının güçlü bir meta sezgisel algoritma gibi davrandığı ve aynı şekilde büyük boyutlu problemlerde de performanslı olduğu belirtilmektedir. Yeni bir meta sezgisel algoritma olan Tarım Arazisi Verimliliği algoritmasının tüm optimizasyon problemlerini iyi ve doğru bir şekilde çözebileceği iddia edilmektedir [23].

Tarım Arazisi Verimliliği algoritmasında, çiftçilerin tarım arazisini toprak kalitesi açısından farklı bölümlere ayırmaları simüle edilir (Şekil 3.1). Tarım arazisinin her bir bölümü farklı bir toprak kalitesine sahiptir. Toprak kalitesini artırmak ve böylece daha yüksek verimlilikle daha iyi ve daha kaliteli ürünler üretmek amacıyla kimyasal, organik, hayvansal veya yeşil gübre gibi toprak kalitesini artıran herhangi bir malzeme kullanılmaktadır. Kullanılan bu materyaller sonucuna toprak kalitesi artarsa, tarım alanından elde edilecek ürün miktarı ve kalitesi artacaktır.



Şekil 3.1. Tarım arazisi, yerel hafıza ve genel hafıza örneği

Çiftçiler tarım arazisini her ziyaret ettiklerinde önceki toprak kalitesinden daha iyi bir kalite elde etmek için toprağa malzeme eklemeye devam ederler. Ayrıca, gelecek ziyaretlerinde, her bir bölümdeki toprak kalitesi bilindiği için, ilgili bölümün toprak kalitesini artırmak için gereken malzemelerin miktarı ve türüne önceden karar verebilirler. Sonuçta bir tarım arazisinde farklı kalitede olan toprak tipleri bulunabilir ve malzeme ve gübre ekleme vb. işlemler ile toprak kalitesi artırılabilir.

Çiftçiler tarım arazisini her bir ziyaretlerinde, her bölüm için ilgili değişiklikleri uygularlar ve her bölüm için toprak kalitesindeki değişimleri not ederler. Sonraki ziyaretlerinde, elde ettikleri bu bilgileri arazinin diğer bölümlerine de uygularlar.

Tarım arazilerindeki her bir bölümün toprak kalitesini belirledikten sonra, en kötü kaliteye sahip bölümün toprak kalitesini değiştirmek için en yüksek kalite ve miktardaki malzemeyi kullanırlar. Bu nedenle, en kötü kaliteye sahip olan bölüm en çok değişikliğe uğrayacak olan bölümdür.

Algoritmaya göre, araziye ait iki farklı hafıza bulunur: yerel ve genel hafıza. Yerel hafıza, her bir bölüme ait geçmiş dönemde uygulanan en iyi çözümleri tutar. Genel hafıza ise, tüm arazideki bilinen en iyi çözümleri tutar. Çiftçiler bu hafızaların her ikisini de sonraki ziyaretler için kullanırlar. Her ziyarette, çiftçiler bulunduğu bölümdeki toprağa materyal eklemeye çalışır ve daha sonra genel hafızadaki en iyilerden biri veya yerel hafızadaki en iyilerden biriyle birleştirerek daha kaliteli toprak üretebilirler. Bu algoritmada, tarım arazisinin en kötü kaliteye sahip bölümüne farklı davranılmaktadır. En kötü kalitedeki toprağa, genel hafızadaki en iyi çözümler uygulanmaktadır. Diğer bölümler içinse, var olan tüm çözümlerden rastlantısal olarak bir tanesi seçilip birleştirilir ve böylece daha iyi sonuçlar elde edilmeye çalışılır.

Sonraki aşamada ise, alanlara ait topraklar karıştırılarak kombinasyon elde edilir. Her bir ziyarette, yerel hafızadaki ya da genel hafızadaki en iyi sonuca ait toprak kullanılarak karışım elde edilir. Bu aşamada yerel ya da genel hafızanın kullanımına rastlantısal olarak karar verilir.

3.1.1. Başlangıç Popülasyonunun Oluşturulması

Tüm sezgisel algoritmalarda, ilk aşama probleme ilişkin çözümlerin hazırlanmasından oluşur. Bu aşamada oluşturulacak başlangıç popülasyonun sayısı, tarım arazisinin bölüm sayısı ve her bölümdeki mevcut çözüm sayısının çarpımından oluşmaktadır.

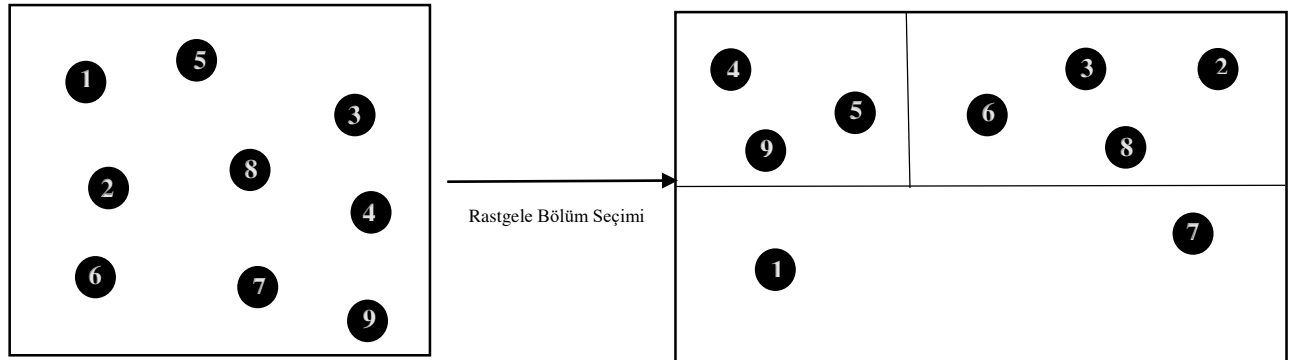
$$N = k * n \quad (3.1)$$

Denklem (3.1) de N çözüm uzayındaki popülasyon sayısını belirtir. k optimizasyon problemine göre tarım arazisinin bölüneceği bölüm sayısını ifade eder. Optimizasyon problemine göre bölüm sayısı değiştirilebilir. Tarım arazisi alanı k bölüme ayrılmış ve her bölümde n tane çözüm bulunmaktadır. Her bir bölümdeki çözüm sayısını ifade eden n sayısı değişebilir ve tamsayıdır. Çözüm uzayının rastgele üretilmesi için Denklem (3.2) kullanılmaktadır.

$$X_{ij} = L_j + rand(0,1) \times (U_j - L_j) \quad (3.2)$$

U_j ve L_j X değişkeninin üst ve alt sınırlarıdır. $rand(0,1)$ fonksiyonu, 0 ile 1 arasında rasgele bir sayı üretir. j değeri, X'in boyut sayısını ifade eder. i ise 1 ile N arasındaki sayıyı göstermektedir. Tarım arazilerinin bölüm sayısı dikkate alınmaksızın bu aşama tüm çözüm uzayındaki mevcut çözümler üzerinde gerçekleştirilir. Böylece her bir çözümün uygunluğu çözüm uzayında belirtilir.

Tarım Arazisi Verimliliği algoritmasında, bölüm sayısını ifade eden k değeri büyük önem taşımaktadır. Algoritmayı diğer meta sezgisel algoritmalarından farklı kılan k değişkenidir. Gerçek dünyada çiftçiler arazilerini Şekil 3.1 'de de görüldüğü gibi dikdörtgen veya kare şekillerle diğer alanlardan ayrılırlar. Ancak meta sezgisel algoritmalarda k 'nın sonsuz olma ihtimali bulunmaktadır, bu da önemli bir problemidir. Arama alanını meta sezgisel algoritma da dikdörtgen, kare veya dairesel şekilde bölmek istiyorsak, meta sezgisel algoritmaların rastgelelik ilkesine uygun olmalıdır. Sonuç olarak arama alanını meta sezgisel algoritmalarda bölmenin en uygun yolu arama alanını rastgele paylaşmaktır. Tarım arazisi verimliliği algoritmasında çözümler ilk olarak Şekil 3.3 de ki gibi rastgele oluşturulur ve her bir çözüm bölümlerden birine rastgele atanır.



Şekil 3.2. Tarım arazisi verimliliği algoritmasında arama uzayının bölümlendirmesi

Sahayanfar ve Gharenchopogh yaptıkları çalışmada bir optimizasyon problemi olarak kabul edilebilecek arama alanı sayısını belirten k nin sınırlı bir sabit değer $2 \leq k \leq 8$ aralığında olması gerektiğini ispatlamışlardır [22]. Algoritmada keşif ve üretkenlik arasında bir denge kurmak için k ve Q değişkeni kullanılmaktadır.

3.1.2. Toprak Kalitesinin Belirlenmesi

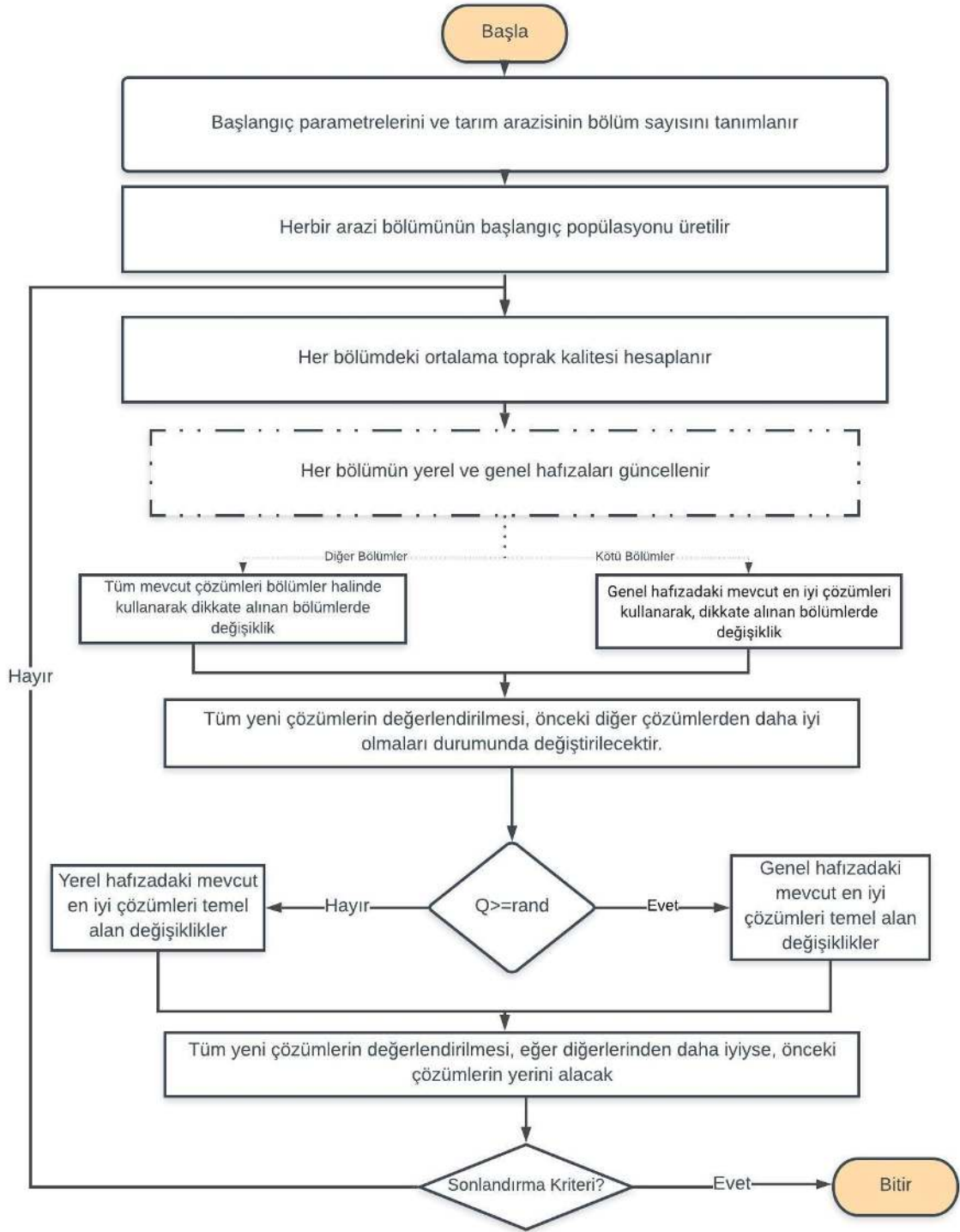
Başlangıç popülasyonu oluşturulduktan sonra tarımsal alan bölümlerinin her birinin kalitesi Denklem (3.3) e göre hesaplanır.

$$Bölüm_s = x(a_j), a = n * (s - 1) : n * ss = \{1, 2, \dots, k\}, j = \{1, 2, \dots, 4\} \quad (3.3)$$

Denklem (3.3) 'de x arama alanındaki tüm çözümlere eşittir. j , x değişkeninin boyutunu ifade eder. Bu denklem ile, çözüm uzayı alanlar arasında paylaştırılır.

$$Fit_{bölüm_s} = Ortalama \left(bölüm_s \text{deki} BütünFit(X_{ij}) \right), \\ s = \{1, 2, \dots, k\}, i = \{1, 2, \dots, n\} \quad (3.4)$$

Denklem (3.4) de görüldüğü gibi her bir bölümdeki çözümlerin ortalaması alınarak o bölüme ait ortalama toprak kalitesi değeri bulunur. $fit_{bölüm}$, tarım alanlarının her bir bölümü için çözümlerin ortalama kalitesini ifade eder.



Şekil 3.3. Tarım arazisi verimliliği algoritması akış şeması

3.1.3. Yerel ve Genel Hafızaların Güncellenmesi

Tarım arazilerinin her bir bölümünün çözümlerini ve her bir bölümün ortalamasını belirledikten sonra Şekil 3.2 ‘de de görüldüğü gibi bölümlerin her birinin yerel ve genel

hafızası güncellenmektedir. Her bölümün en iyi çözümü bölümün yerel hafızasında, tüm çözümlerden en iyi çözüm ise genel hafızada saklanır.

$$M_{yerel} = round(t * n) \quad 0.1 < t < 1 \quad (3.5)$$

$$M_{genel} = round(t * N) \quad 0.1 < t < 1 \quad (3.6)$$

Denklem (3.5) 'te M_{yerel} yerel hafızadaki çözümlerin, Denklem (3.6) 'da M_{genel} ise genel hafızadaki çözümlerin sayısını gösterir. Yerel ve genel hafızalardaki uygunluğa göre çözümler yerleştirilir ve hafızalar güncellenir. Yerel ve genel hafızalar güncellendikten sonra en iyi ve en kötü bölümler belirlenir.

3.1.4. Her Bölümdeki Toprak Kalitesinin Değerlendirilmesi

Bu aşamada her bir bölümün toprak kalitesi Denklem (3.4) ile belirlendikten sonra, en kötü kalitede olan bölümde büyük değişiklikler olacaktır. Tarım arazilerinin en kötü bölümündeki tüm çözümler Denklem (3.7) ve Denklem (3.8) ile genel çözümlerden biri ile birleştirilecektir.

$$h = \alpha * rand(-1.1) \quad (3.7)$$

$$X_{new} = h * (X_{ij} - X_{MGenel}) + X_{ij} \quad (3.8)$$

X_{MGenel} genel hafızadaki mevcut çözümler arasında rastgele bir çözümdür. α algoritmanın başında tanımlanması gereken 0 ile 1 arasında bir sayıdır. X_{ij} tarım arazisinin en kötü bölümünde değişiklikleri uygulamak için seçilen bir çözümdür. Denklem (3.7) deki h değişkeni denkleme göre hesaplanabilecek ondalık bir sayıdır. Denklem (3.8) de uygulanan değişikliklerle yeni bir çözüm olan X_{new} elde edilir. Tarım arazisinin en kötü bölümünde değişiklikler yapıldıktan sonra, diğer bölümler mevcut çözümler ile birleştirilmelidir.

$$h = \beta * rand(0.1) \quad (3.9)$$

$$X_{new} = h * (X_{ij} - X_{uj}) + X_{ij} \quad (3.10)$$

Denklem (3.9)' da yer alan β algoritmanın başında tanımlanması gereken 0 ile 1 arasında bir sayıdır. X_{uj} tüm arama alanındaki mevcut çözümler arasından rastgele bir çözümdür. X_{ij} en kötü bölüm hariç değişiklikleri uygulamak için seçilen bir çözümdür. Denklem (3.10) 'a göre X_{new} ise uygulanan değişiklikler ile elde edilen yeni bir çözümdür.

3.1.5. Toprağın Birleşimi ve Son Koşullar

Bu aşamada, çiftçiler her bir bölüme ait toprağı, rastlantısal olarak ilgili bölüme ait yerel hafızadaki en iyi çözümle ya da şimdiye kadarki bilinen en iyi çözümle birleştirirler.

$$H = \begin{cases} X_{new} = X_{ij} + \omega_1 * (X_{ij} - Eniyi_{genel}(b)) & .Q > rand \\ X_{New} = X_{ij} + rand(0,1) * (X_{ij} - Eniyi_{yerel}(b)) & .else \end{cases} \quad (3.11)$$

$$\omega_1 = \omega_1 * R_v \quad . \quad 0 < R_v < 1 \quad (3.12)$$

Denklem (3.11) de iki yöntem yeni bir çözüm oluşturabilir. Denklem de yer alan Q 0 ile 1 arasında bir değer alır ve algoritmanın başında tanımlanması gereken bir parametredir. Q parametresi, Denklem (3.11)'de gösterildiği gibi yerel ya da global çözümlerin kullanımına rastlantısal olarak karar verirken kullanılır. ω_1 algoritmanın bir parametresidir, başlangıçta tanımlanmalıdır ve algoritmanın her tekrarında kademeli olarak azalmaktadır. X_{ij} değişiklikleri uygulamak için tüm bölümlerden seçilen bir çözümdür. Sonuç olarak X_{new} uygulanan değişikliklere göre elde edilen yeni bir çözümdür.

4. KAOTİK HARİTALI TARIM ARAZİSİ VERİMLİLİĞİ ALGORİTMASI

4.1. Kaotik Haritalar

Kaos teorisi, başlangıç koşullarına duyarlı doğrusal olmayan dinamik sistemleri ifade etmektedir. Başlangıç koşullarındaki küçük değişiklikler, sistemin sonucuna yüksek oranda etki eder [15]. Yapılan çalışmalarda, rastgele sayı dizileri yerine kaotik sayı dizileri, karmaşık problemleri modellemede, sayısal analiz yöntemlerinde, görüntü şifrelemede, kriptolojide ve sezgisel optimizasyon yöntemlerinde sıkça kullanılmıştır ve başarılı sonuçlar elde edilmiştir [5,6]. Kaotik haritalar, algoritmaların performansını hem yerel optimumdan kaçınma hem de yakınsama hızı açısından geliştiren en iyi yöntemlerden biridir [6]. Sezgisel algoritmalarda kaotik haritalar ile yerel optimuma takılma riski azaltılabilmektedir. Optimizasyon algoritmalarında yerel optimuma takılma riskinden kaçınmak için kaotik haritalar sıklıkla kullanılmıştır. Bu şekilde algoritmaların performansları artırılmıştır [15].

Kaotik sistemler başlangıç değerlerine aşırı bağımlıdır. Kaotik bir sistem birbirine çok yakın iki farklı başlangıç noktasından başlatılırsa bu küçük farklılık zamanla üstel olarak artar. Ancak kaotik olmayan bir sistemde ise fark zamanla doğrusal olarak artan bir hataya dönüşebilmektedir. Kaotik sistemlerin hesaplama maliyetleri düşüktür. Üretilen sayılar için fazla depolama alanı kullanılmamalı ve istenen bir doğruluğa ulaşmak için fazla zamana gereksinim duyulmamalıdır. Yapılan çalışmalar kaotik sayı dizilerinin üretilmelerinin ve depolanmalarının kolay ve hızlı olduğunu göstermektedir [4]. Kaotik algoritmalar ile yapılan sistemlerin hızı daha yüksek olmaktadır.

Kaotik haritalar, karmaşık problemleri modellemede, sayısal analiz yöntemlerinde, görüntü şifrelemede, kriptolojide ve sezgisel optimizasyon yöntemlerinde sıkça kullanılmaktadır.

a. Lojistik Harita

En sık kullanılan ve anlaşılması kolay kaotik haritalardandır. Lojistik haritalar rastgele sayı üretiminde, imge şifreleme ve optimizasyon yöntemlerinde kullanılmaktadır [32]. Denklemi (4.1) deki gibi ifade edilmektedir.

$$X_{n+1} = \lambda X_n(1 - X_n) \quad (4.1)$$

Burada sırasıyla $X_n \in (0,1)$ ve λ sistem değişkeni ve parametresi, n ise yineleme sayısıdır. Böylece, bir başlangıç değeri x_0 ve bir parametre k alarak, $\{X_n\}_{n=0}^{\infty}$ serisi hesaplanır.

b. Çadır(Tent) Harita

(0,1) aralığında sayılar üreten Çadır harita Logistic haritaya benzerliği ile bilinmektedir. Denklemi (4.2) deki gibi ifade edilmektedir [33].

$$X_{n+1} = \begin{cases} \frac{X_n}{0.7} & , X_n < 0.7 \\ \frac{10}{3}(1 - X_n) & X_n \geq 0.7 \end{cases} \quad (4.2)$$

c. Çember(Circle) Harita

Çember haritanın denklemi (4.3) deki gibi ifade edilmektedir.

$$X_{n+1} = X_n + b - \left(\frac{a}{2\pi}\right) \sin(2\pi n) \bmod (1) \quad (4.3)$$

$a = 0.5$ ve $b = 0.2$ olduğunda Çember harita (0,1) aralığında kaotik sayı dizileri üretir [33].

d. Gauss Harita

(0,1) aralığında kaotik diziler üreten Gauss haritaya ait denklemler (4.4a) ve (4.4b) teki gibi ifade edilmektedir [7].

$$X_{n+1} = \begin{cases} 0 & X_n = 0 \\ \frac{1}{X_n \bmod(1)} & \text{Veya} \end{cases} \quad (4.4a)$$

$$\frac{1}{X_n \bmod(1)} = \frac{1}{X_n} - \left[\frac{1}{X_n} \right] \quad (4.4b)$$

e. Çebişev(Chebysev) Harita

Çebişev haritası denklem (4.5) teki gibi ifade edilir [7].

$$X_{n+1} = \cos(\cos^{-1}(X_n)) \quad (4.5)$$

f. Sinüs(Sine) Harita

Sinüs haritaya ait denklem (4.6) da ifade edilmiştir [16].

$$X_{n+1} = 2.3 (X_n)^{2 \sin(\pi x_n)} \quad (4.6)$$

g. Sinüzoidal Yenileyici

Sinüzoidal yineleyici denklemi (4.7a) da verilmiştir.

$$X_{n+1} = aX_n^2 \sin(\pi X_n) \quad (4.7a)$$

$a = 2.3$ ve $X_0 = 0.7$ değerleri alınarak denklem (4.7b) deki gibi daha basit hale dönüştürülebilir [7].

$$X_{n+1} = \sin(\pi X_n) \quad (4.7b)$$

h. Singer Harita

Singer Harita Denklemi Denklem (4.8) 'de verilmiştir.

$$X_{n+1} = a(7.86X_n - 23.31X_n^2 + 28.75X_n^3 - 13.3X_n^4) \quad (4.8)$$

a 0.9 ile 1.08 aralığında kabul edilmektedir [16].

ı. Parçalı(Piecewise) Harita

Parçalı haritaya ait matematiksel formül Denklem (4.9) 'da verilmiştir.

$$X_{n+1} = \begin{cases} \frac{X_n}{P} & 0 \leq X_n < P \\ \frac{X_n - P}{0.5 - P} & P \leq X_n < \frac{1}{2} \\ \frac{1 - P - X_n}{0.5 - P} & \frac{1}{2} \leq X_n < 1 - P \\ \frac{1 - X_n}{P} & 1 - P \leq X_n < 1 \end{cases} \quad (4.9)$$

P 0 ile 0.5 arasında kontrol parametresi olduğunda ve $X \in (0,1)$ ise $P \neq 0$ [7].

i. İteratif Harita

İteratif haritaya ait matematiksel formül Denklem (4.10) 'da belirtilmektedir.

$$X_{n+1} = \sin\left(\frac{a\pi}{X_n}\right) \quad (4.10)$$

Denklemda $a \in (0,1)$ şeklinde alınabilir [7].

4.2. Kaotik Haritalı Tarım Arazisi Verimliliği Algoritması

Tarım Arazisi Verimliliği algoritmasında, başlangıç popülasyonunun her bir parçacığına ait boyut değerleri, $\text{rand}(0,1)$ komutu ile rastgele üretilen sayılar kullanılarak Denklem (3.2)'deki gibi üretilmektedir.

Bu tez çalışmasında $X_{ij} = L_j + \text{rand}(0,1) \times (U_j - L_j)$ denklemindeki (Denklem 3.2) $\text{rand}(0,1)$ komutu yerine aşağıda detayları açıklanan üç farklı yaklaşım ile kaotik haritalar uygulanmıştır.

4.2.1. Tüm Boyutların Kaotik Haritalar ile Üretildiği Birinci Yaklaşım

Bu yaklaşım da, her bir parçacığa ait boyut değerleri hesaplanırken kaotik haritalar kullanılmıştır. Parçacığa ait ilk boyut değeri, rastgele üretilen 0 ve 1 arasındaki bir sayı kullanılarak Denklem (3.2) ile hesaplanmıştır. 2. ve sonrasındaki değerler için ise, Denklem (3.2)'de kullanmak üzere yeniden rastgele sayılar üretilmemiştir. Bunun yerine, ilk boyut için üretilen rastgele sayı kullanılarak kaotik haritalar ile yeni sayılar elde edilmiş ve Denklem (3.2) de bu sayılar kullanılmıştır. Her bir parçacık için bu işlem tekrarlanmıştır.

Başlangıç popülasyonun da bulunan her bir parçacığın, her bir boyutu için Denklem (3.2)'de bulunan $\text{rand}(0,1)$ yerine kaotik haritaların kullanımı Tablo 4.1 'de detaylı olarak verilmiştir.

Tablo 4.1. Birinci yaklaşımda kullanılan rastgele sayılar

	Boyut 1	Boyut 2	...	...	Boyut Nvar
Parçacık 1	$rand_1$	$Kaotik(rand_1)_2$	...	...	$Kaotik(rand_1)_{Nvar}$
Parçacık 2	$rand_2$	$Kaotik(rand_2)_2$	...	...	$Kaotik(rand_2)_{Nvar}$
...	...	...	...	...	...
...	...	...	...	...	...
Parçacık N	$rand_N$	$Kaotik(rand_N)_2$	...	...	$Kaotik(rand_N)_{Nvar}$

4.2.2. Tüm Popülasyonun ve Tüm Boyutların Kaotik Haritalar ile Üretildiği İkinci Yaklaşım

Bu yaklaşım da, ilk parçacığın tüm boyutları, TAV algoritmasındaki gibi rastgele sayılar kullanılarak Denklem (3.2) ile üretilmiştir. İkinci ve sonrasındaki parçacıkların boyut değerlerinin hesaplanması için ise, yeniden rastgele sayı üretilmemiş, ilk parçacığın ilgili boyut değeri hesaplanırken kullanılan rastgele sayı baz alınarak kaotik haritalar ile üretilen yeni sayılar kullanılmıştır. Her boyut için bu işlem tekrarlanmıştır.

Başlangıç popülasyonunda bulunan her bir parçacığın, her bir boyutu için Denklem (3.2)'de bulunan $rand(0.1)$ yerine kaotik haritaların kullanımı Tablo 4.2 'de detaylı olarak verilmiştir.

Tablo 4.2. İkinci yaklaşım da kullanılan rastgele sayılar

	Boyut 1	Boyut 2	...	...	Boyut Nvar
Parçacık 1	$rand_1$	$rand_2$	...	...	$rand_{Nvar}$
Parçacık 2	$Kaotik(rand_1)_2$	$Kaotik(rand_2)_2$	...	...	$Kaotik(rand_{Nvar})_2$
...	...	...	...	...	...
...	...	...	...	...	...
Parçacık N	$Kaotik(rand_1)_N$	$Kaotik(rand_2)_N$	...	...	$Kaotik(rand_{Nvar})_N$

4.2.3. Tüm Popülasyonun Tek Boyutlu Şekilde Kaotik Haritalar ile Üretildiği Üçüncü Yaklaşım

Bu yaklaşımda, başlangıç popülasyonunun ilk parçacığı oluşturulurken, Denklem (3.2) de hesaplanan boyut değeri, parçacığın tüm boyutlarına atanır. Böylece ilk parçacığın tüm boyut değerleri birbirine eşit olur. ($X_{1j} = X_{1(j+1)}$)

İkinci ve sonrasındaki parçacıklar hesaplanırken ise, yeniden rastgele sayı üretilmez, ilk parçacık için üretilen rastgele sayı kullanılarak kaotik haritalar ile oluşturulan yeni sayılar kullanılır ve Denklem (3.2) ile elde edilen değerler tüm boyutlara atanır.

Bu yöntemi 4.2.2. yönteminden ayıran temel fark, ilk boyut için yapılan işlemin her boyut için tekrarlanmaması, tüm boyut değerlerinin birbirine eşit olmasıdır. ($X_{ij} = X_{i(j+1)}$)

Başlangıç popülasyonunda bulunan her bir parçacığın, her bir boyutu için Denklem (3.2)'de bulunan $rand(0.1)$ yerine kaotik haritaların kullanımı Tablo 4.3.'de detaylı olarak verilmiştir.

Tablo 4.3. Üçüncü yaklaşımda kullanılan rastgele sayılar

	Boyut 1	Boyut 2	...	...	Boyut Nvar
Parçacık 1	$rand_1$	$rand_1$	...	...	$rand_1$
Parçacık 2	$Kaotik(rand_1)_2$	$Kaotik(rand_1)_2$	...	...	$Kaotik(rand_1)_2$
...	...	...	...	...	...
...	...	...	...	...	...
Parçacık N	$Kaotik(rand_1)_N$	$Kaotik(rand_1)_N$	...	...	$Kaotik(rand_1)_N$

5. KALİTE TESTİ FONKSİYONLARI İLE YAPILAN DENEYLER

5.1. Kalite Testi Fonksiyonları

Optimizasyon algoritmalarının performansını ölçmek için literatürde birçok kalite testi fonksiyonu bulunmaktadır. Kalite test fonksiyonları yerel optimum sayıları ve sıklıklarına göre sınıflandırılmaktadır [23]. Geliştirilen algoritmanın performansını ölçmek için 8 adet kalite test fonksiyonu seçilmiştir.

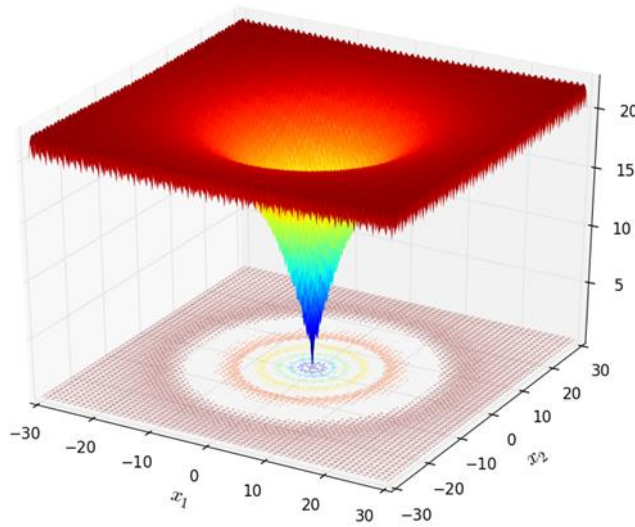
5.1.1. Ackley Fonksiyonu

Optimizasyon problemlerini test etmek için sıklıkla kullanılan çok modlu bir test fonksiyonudur. Matematiksel formülü Denklemi (5.1)' de, grafiği ise şekil (5.1) de verilmiştir.

$$20 + e - 20 \exp \left(-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2} \right) - \exp \left(\frac{1}{d} \sum_{i=1}^d \cos(2\pi x_i) \right) \quad (5.1)$$

Sınırlar $-32 \leq x \leq 32$

$x^* = (0,0,...,0)$ ve $f(x^*) = 0$



Şekil 5.1 Ackley Fonksiyonu

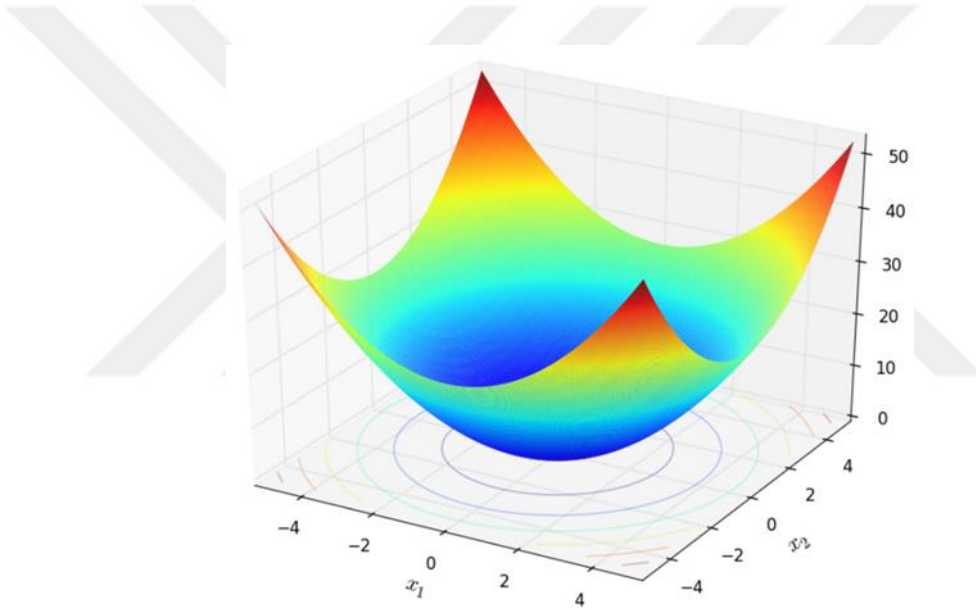
5.1.2. Sphere Fonksiyonu

Global minimum dışında n kadar yerel minimuma da sahip, sürekli, dış bükey ve tek modlu bir test fonksiyonudur [23]. Matematiksel Formülü Denklem (5.2) 'de, grafiği ise Şekil (5.2) de verilmiştir.

$$f(x) = \sum_{i=1}^n x_i^2 \quad (5.2)$$

Sınırlar $-5,12 \leq x_i \leq 5,12$

$x_i = 0$, ve $i=1,...,n$, için Global minimum $f(x) = 0$



Şekil 5.2. Sefphere Fonksiyonu

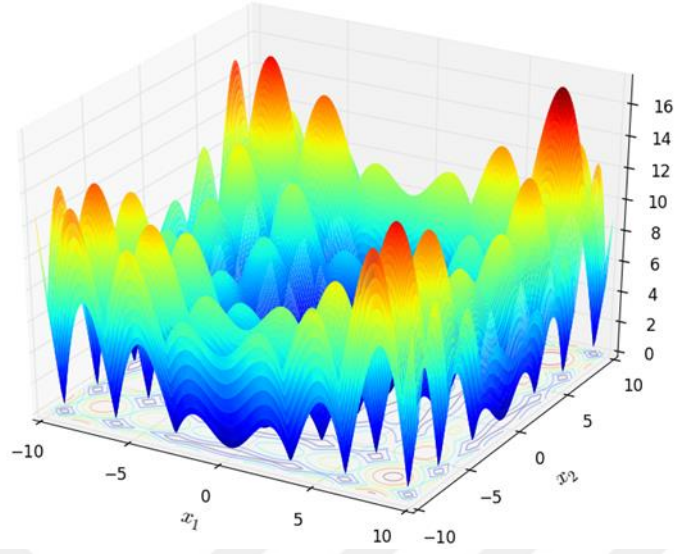
5.1.3. Alpine Fonksiyonu

Matematiksel fonksiyonu Denklem (5.3) de verilen Alpine fonksiyonu birden fazla yerel optimum noktası bulunan çok modlu bir fonksiyondur. Fonksiyonun grafiği ise Şekil (5.3) te verilmiştir.

$$\sum_{i=1}^d |x_i \sin(x_i) + 0.1 x_i| \quad (5.3)$$

Sınırlar $-5,12 \leq x_i \leq 5,12$

$x_i = 0$, ve $i=1,...,n$, için Global minimum $f(x) = 0$



Şekil 5.3. Alpine Fonksiyonu

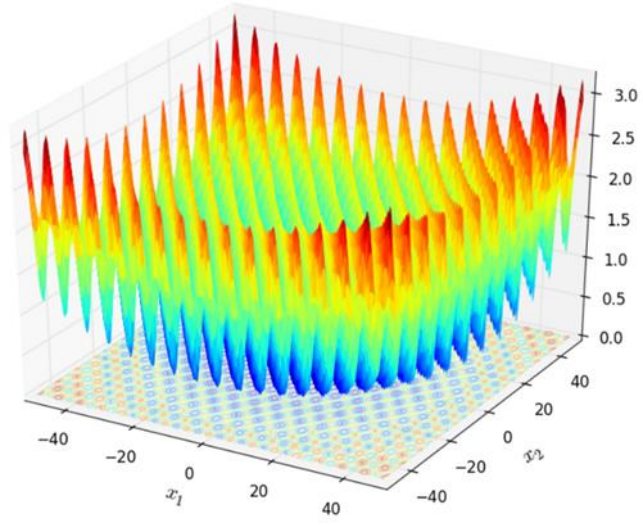
5.1.4. Griewank Fonksiyonu

Griewank Fonksiyonu boyutu arttıkça katlanarak artan yerel optimuma sahiptir. Denklem (5.4) te matematiksel fonksiyonu, Şekil 5.4 te ise grafiği gösterilmektedir.

$$\sum_{i=1}^d \left(\frac{x_i^2}{4000} \right) - \prod_{i=1}^d \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1 \quad (5.4)$$

Sınırlar $-10 \leq x \leq 10$

$x^* = (0,0,...,0)$ ve $f(x^*) = 0$



Şekil 5.4. Griewank Fonksiyonu

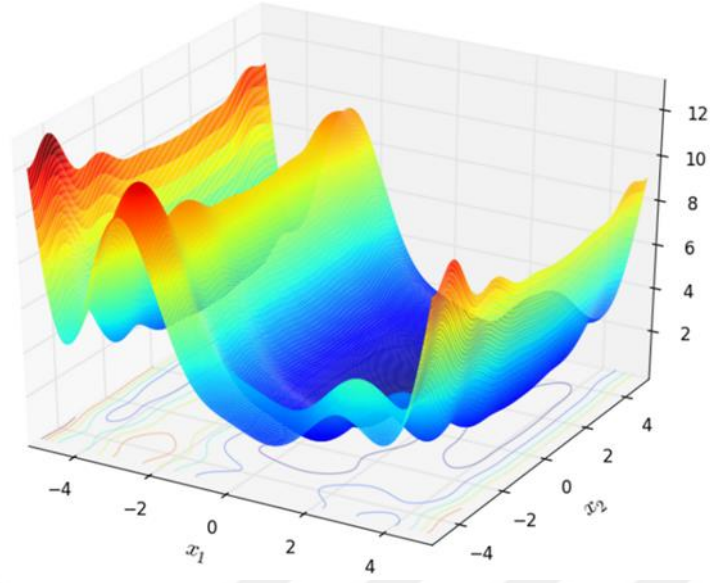
5.1.5. Levy Fonksiyonu

Çok modlu bir fonksiyon olan Levy fonksiyonunun matematiksel ifadesi Denklem (5.5) de, grafiği ise Şekil 5.5 te verilmiştir.

$$\sin^2(\pi y_1) + \sum_{i=1}^{d-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2, y_i = 1 + \frac{x_i - 1}{4} \quad (5.5)$$

Sınırlar $-10 \leq x_i \leq 10$

$x^* = (0, 0, \dots, 0)$ ve $f(x^*) = 0$



Şekil 5.5. Levy Fonksiyonu

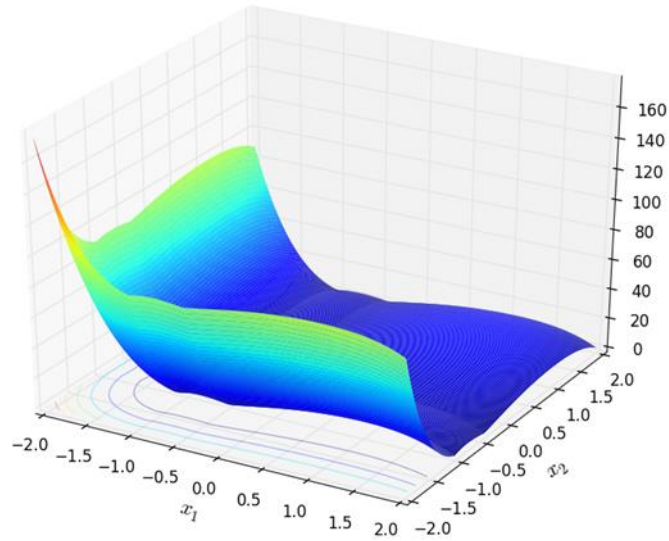
5.1.6. Quintic Fonksiyonu

Matematiksel ifadesi Denklem (5.6) da verilen fonksiyon çok modlu yapıya sahiptir. Fonksiyonun grafiği Şekil (5.6)' da verilmiştir.

$$\sum_{i=1}^d |x_i^5 - 3x_i^4 + 4x_i^3 + 2x_i^2 - 10x_i - 4| \quad (5.6)$$

Sınırlar $-10 \leq x_i \leq 10$

$x^* = (-1 \text{ veya } 2)$ ve $f(x^*) = 0$



Şekil 5.6. Quintic Fonksiyonu

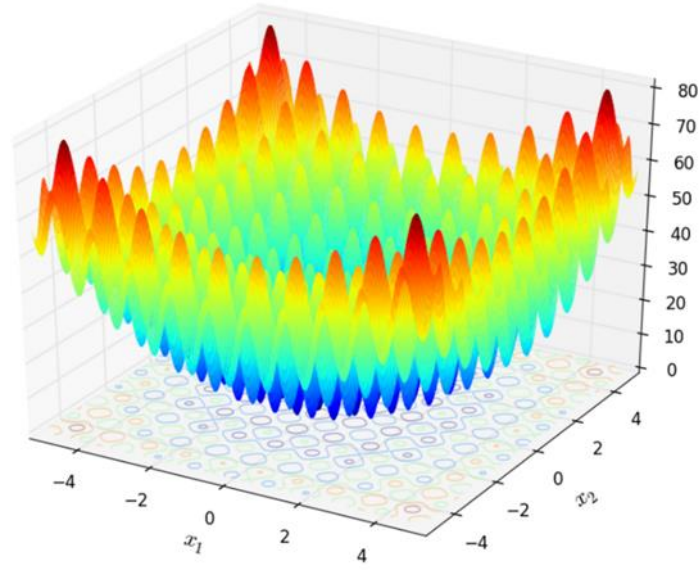
5.1.7. Rastrigin Fonksiyonu

Matematiksel ifadesi Denklem (5.7) da verilen fonksiyon çok modlu yapıya sahiptir. Fonksiyonun grafiği Şekil (5.7)' de verilmiştir.

$$10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)] \quad (5.7)$$

Sınırlar $-5.12 \leq x \leq 5.12$

$x^* = (0,0...0)$ ve $f(x^*) = 0$



Şekil 5.7. Rastrigin Fonksiyonu

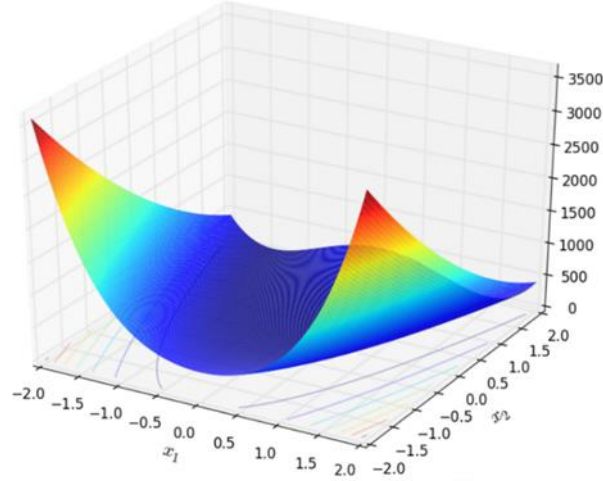
5.1.8. Rosenbrock Fonksiyonu

Tepe noktalarının keskinliği ve dar tepecikleri bulunduğundan ötürü global optimuma yakınsamak zordur. Matematiksel ifadesi Denklem (5.8) de, grafiği ise Şekil 5.8 de verilmiştir.

$$\sum_{i=1}^{d-1} [100(x_i^2 - x_{i+1})^2 + (1 - x_i^2)] \quad (5.8)$$

Sınırlar $-2.048 \leq x \leq 2.048$

$x^* = (1, 1 \dots 1)$ ve $f(x^*) = 0$



Şekil 5.8. Rosenbrock Fonksiyonu

5.2. Kaotik Haritalı Tarım Arazisi Verimliliği Algoritmalarının Test Sonuçları

Bölüm 4.2 de önerilen her üç yaklaşım 10, 20 ve 30 boyutlu olarak 100 ve 500 iterasyon sayıları ile 50 kez test edilmiştir. Testler 8 farklı kalite testi fonksiyonu ile gerçekleştirilmiştir. Test fonksiyonlarının tüm sonuçları her üç yaklaşım için de aşağıda detaylı olarak verilmiştir.

5.2.1. Tüm Boyutların Kaotik Haritalar ile Üretildiği Birinci Yaklaşım Test Sonuçları

50 bağımsız çalıştırma ile test edilen sonuçlar Tablo 5.1, Tablo 5.2, Tablo 5.3, Tablo 5.4, Tablo 5.5 ve Tablo 5.6 da gösterilmektedir. İlgili harita ve fonksiyon için daha başarılı olan değerler koyu punto ile belirtilmiştir.

Tablo 5.1. 100 iterasyon ve 10 boyut sayısı ile 50 test sonucu ortalaması

	Sphere	Rastrigin	Rosenbrock	Griewank	Ackley	Levy	Alpine	Quintic
TAV	4,365E-06	2,192E+01	7,618E+00	1,846E-01	3,295E-03	1,002E-03	7,214E-02	3,076E+00
Chebyshev	9,171E-06	2,591E+01	7,857E+00	2,650E-01	5,555E-03	4,893E-03	1,497E-01	2,409E+00
Circle	4,434E-06	2,343E+01	7,620E+00	1,529E-01	3,555E-03	1,941E-03	7,831E-02	2,564E+00
Gauss/ mouse	5,095E-06	2,393E+01	7,772E+00	2,013E-01	3,661E-03	2,318E-03	8,005E-02	2,563E+00
Iterative	5,734E-06	2,307E+01	8,123E+00	1,966E-01	3,768E-03	1,410E-03	8,505E-02	3,363E+00
Logistic	5,911E-06	2,375E+01	8,313E+00	2,543E-01	4,305E-03	1,501E-03	8,644E-02	4,233E+00
Piecewise	4,362E-06	2,388E+01	8,091E+00	1,856E-01	3,728E-03	9,591E-04	6,192E-02	3,033E+00
Sine	5,911E-06	2,324E+01	7,968E+00	2,377E-01	4,207E-03	1,898E-03	9,267E-02	3,980E+00
Singer	8,222E-06	2,481E+01	8,053E+00	2,480E-01	6,246E-03	2,655E-03	3,722E-01	6,736E+00
Sinusoidal	5,532E-06	2,236E+01	7,963E+00	2,227E-01	3,982E-03	1,211E-03	1,084E-01	4,252E+00
Tent	4,705E-06	2,356E+01	7,680E+00	1,416E-01	3,913E-03	2,904E-03	9,500E-02	2,536E+00

Tablo 5.1.'de görülebildiği gibi, 100 iterasyon ve 10 boyut ile test edildiğinde Quintic fonksiyonu sonuçlarında Chebyshev, Circle, Gauss/Mouse, Piecewise ve Tent haritalı TAV algoritmaları daha iyi sonuçlar vermiştir. Bununla birlikte; Iterative, Logistic, Sine, Singer ve Sinusoidal haritalı TAV algoritmaları hiçbir fonksiyonda standart TAV algoritmasından daha iyi sonuç vermemiştir.

Tablo 5.2. 500 iterasyon ve 10 boyut sayısı ile 50 test sonucu ortalaması

	Sphere	Rastrigin	Rosenbrock	Griewank	Ackley	Levy	Alpine	Quintic
TAV	4,668E-34	3,493E+00	5,829E+00	2,895E-02	4,299E-15	7,324E-30	7,073E-06	9,108E-04
Chebyshev	1,785E-33	4,213E+00	6,009E+00	7,782E-02	4,228E-15	1,843E-28	1,124E-05	7,503E-04
Circle	6,393E-34	3,951E+00	6,038E+00	1,977E-02	4,228E-15	5,566E-29	3,938E-06	3,030E-04
Gauss/ mouse	8,558E-34	4,286E+00	5,987E+00	3,772E-02	4,086E-15	1,415E-29	7,173E-06	3,799E-04
Iterative	6,697E-34	3,730E+00	5,694E+00	3,697E-02	4,157E-15	4,209E-30	2,623E-06	6,687E-04
Logistic	7,060E-34	4,527E+00	5,308E+00	5,971E-02	4,228E-15	4,474E-30	1,141E-05	3,624E-03
Piecewise	4,260E-34	4,165E+00	5,825E+00	3,314E-02	4,157E-15	8,194E-30	5,768E-06	8,443E-04
Sine	9,037E-34	4,375E+00	5,546E+00	4,480E-02	4,015E-15	1,302E-28	4,288E-06	1,488E-03
Singer	1,354E-33	4,768E+00	1,703E+00	5,309E-02	4,299E-15	2,949E-29	2,001E-05	2,524E-02
Sinusoidal	6,506E-34	3,832E+00	5,221E+00	4,529E-02	4,228E-15	1,376E-29	4,283E-06	4,430E-03
Tent	8,103E-34	4,217E+00	5,931E+00	1,773E-02	4,157E-15	6,665E-28	1,606E-05	7,818E-04

Tablo 5.2. incelendiğinde, aynı boyut sayısı ile iterasyon sayısı 100'den 500'e çıkarıldığında, genel olarak kaotik haritaların başarılarının arttığı gözlemlenebilmektedir. Örneğin Ackley fonksiyonu 100 iterasyonda hiç iyi sonuç vermezken, 500 iterasyona çıkıldığında Singer hariç tüm haritalarda daha iyi sonuç vermiştir. Iterative ve Piecewise haritaları 5 farklı fonksiyonda, Circle benzer şekilde 4 farklı fonksiyonda daha iyi performans göstermiştir.

Tablo 5.3. 100 iterasyon ve 20 boyut sayısı ile 50 test sonucu ortalaması

	Sphere	Rastrigin	Rosenbrock	Griewank	Ackley	Levy	Alpine	Quintic
TAV	8,031E-02	1,209E+02	1,962E+02	1,439E-01	6,972E-01	2,157E+00	6,523E+00	5,407E+01
Chebyshev	1,378E-01	1,239E+02	2,041E+02	1,625E-01	9,372E-01	3,040E+00	7,803E+00	4,998E+01
Circle	8,775E-02	1,208E+02	1,404E+02	1,366E-01	7,339E-01	2,368E+00	6,278E+00	4,744E+01
Gauss/ mouse	9,134E-02	1,194E+02	1,673E+02	1,519E-01	7,095E-01	2,554E+00	7,130E+00	5,167E+01
Iterative	9,327E-02	1,206E+02	1,926E+02	1,427E-01	6,868E-01	2,238E+00	6,705E+00	5,801E+01
Logistic	9,867E-02	1,235E+02	2,263E+02	1,554E-01	7,592E-01	2,106E+00	7,129E+00	6,247E+01
Piecewise	7,942E-02	1,210E+02	1,953E+02	1,589E-01	7,116E-01	2,047E+00	6,351E+00	6,103E+01
Sine	1,007E-01	1,192E+02	2,059E+02	1,596E-01	7,777E-01	2,300E+00	7,226E+00	5,913E+01
Singer	1,477E-01	1,278E+02	5,104E+02	1,736E-01	1,128E+00	3,051E+00	9,182E+00	8,027E+01
Sinusoidal	9,447E-02	1,209E+02	2,592E+02	1,422E-01	7,484E-01	3,126E+00	6,949E+00	6,724E+01
Tent	9,103E-02	1,213E+02	2,071E+02	1,390E-01	7,273E-01	2,032E+00	7,064E+00	5,455E+01

Tablo 5.3.'de 100 iterasyon ve 20 boyut ile çalıştırmanın sonuçları bulunmaktadır. Tablo 5.1.'de bulunan, 10 boyut ile çalıştırmaya kıyasla, genel olarak kaotik haritaların başarı sayısının arttığı gözlemlenebilmektedir. Örneğin Circle haritası Griewank ve Quintic fonksiyonlarındaki başarısını sürdürürken; Rastrigin, Rosenbrock ve Alpine fonksiyonlarında da daha iyi sonuçlar üretmeye başlamıştır. Yine Iterative ve Piecewise haritaları da 4 farklı test fonksiyonu için daha iyi sonuçlar üretmiştir. Diğer taraftan; Singer haritası hiçbir fonksiyon için TAVA 'dan daha iyi sonuç üretmemiştir.

Tablo 5.4. 500 iterasyon ve 20 boyut sayısı ile 50 test sonucu ortalaması

	Sphere	Rastrigin	Rosenbrock	Griewank	Ackley	Levy	Alpine	Quintic
TAV	8,105E-15	4,091E+01	1,651E+01	3,057E-05	1,023E-07	4,240E-08	1,166E-02	1,351E+00
Chebyshev	1,155E-14	4,275E+01	1,648E+01	2,299E-05	1,381E-07	6,720E-07	1,510E-02	7,069E-01
Circle	8,907E-15	4,649E+01	1,655E+01	3,225E-05	1,093E-07	8,487E-07	1,334E-02	6,377E-01
Gauss/ mouse	6,800E-15	4,487E+01	1,652E+01	3,330E-05	1,027E-07	4,420E-07	1,156E-02	8,658E-01
Iterative	8,063E-15	4,629E+01	1,661E+01	4,474E-05	1,095E-07	3,284E-08	1,227E-02	1,567E+00
Logistic	9,507E-15	4,664E+01	1,653E+01	4,478E-05	1,062E-07	2,779E-08	1,327E-02	1,908E+00
Piecewise	7,641E-15	4,606E+01	1,651E+01	4,829E-04	1,009E-07	2,901E-08	1,201E-02	1,493E+00
Sine	8,535E-15	4,379E+01	1,657E+01	2,572E-05	1,081E-07	1,897E-08	1,326E-02	1,600E+00
Singer	1,382E-14	4,632E+01	1,403E+01	1,968E-05	4,430E-07	9,204E-07	2,231E-02	6,003E+00
Sinusoidal	9,212E-15	4,496E+01	1,676E+01	1,596E-05	1,159E-07	3,046E-07	1,101E-02	2,721E+00
Tent	6,958E-15	4,374E+01	1,661E+01	3,121E-05	1,100E-07	8,151E-08	1,745E-02	1,227E+00

Tablo 5.4. ve Tablo 5.3. karşılaştırıldığında, 20 boyut için iterasyon sayısının 100'den 500'e çıkması ile kaotik haritaların performansında istikrarlı bir değişiklik tespit edilememiştir. Bununla birlikte test fonksiyonlarının sonuçlarında anlamlı değişiklikler görülmektedir. Örneğin 100 iterasyonda 5 farklı kaotik harita, Rastrigin fonksiyonu ile daha iyi değerler üretirken, 500 iterasyonda herhangi bir kaotik harita Rastrigin fonksiyonu ile standart TAV algoritmasından daha iyi sonuç üretememiştir. Diğer taraftan, 100 iterasyonda Sphere fonksiyonu ile sadece Piecewise haritası iyi sonuç üretirken, 500 iterasyonda Piecewise ile birlikte 3 farklı kaotik harita da daha iyi sonuç vermeye başlamıştır.

Tablo 5.5. 100 iterasyon ve 30 boyut sayısı ile 50 test sonucu ortalaması

	Sphere	Rastrigin	Rosenbrock	Griewank	Ackley	Levy	Alpine	Quintic
TAV	3,093E+00	2,494E+02	5,634E+03	3,822E-01	3,114E+00	1,690E+01	1,897E+01	2,420E+02
Chebyshev	4,257E+00	2,570E+02	6,574E+03	4,344E-01	3,394E+00	2,734E+01	2,121E+01	3,542E+02
Circle	3,056E+00	2,472E+02	4,795E+03	3,802E-01	3,130E+00	2,070E+01	1,857E+01	2,588E+02
Gauss/ mouse	3,438E+00	2,504E+02	5,600E+03	3,769E-01	3,168E+00	2,272E+01	1,875E+01	2,959E+02
Iterative	3,520E+00	2,513E+02	6,046E+03	3,836E-01	3,127E+00	1,524E+01	1,956E+01	2,347E+02
Logistic	3,551E+00	2,443E+02	5,836E+03	3,802E-01	3,181E+00	1,439E+01	1,994E+01	2,772E+02
Piecewise	3,235E+00	2,501E+02	5,122E+03	3,864E-01	3,095E+00	1,699E+01	1,873E+01	2,607E+02
Sine	3,416E+00	2,506E+02	5,150E+03	3,983E-01	3,181E+00	1,699E+01	2,021E+01	2,738E+02
Singer	4,741E+00	2,581E+02	1,216E+04	4,570E-01	3,562E+00	1,714E+01	2,297E+01	4,646E+02
Sinusoidal	3,737E+00	2,548E+02	6,767E+03	3,875E-01	3,186E+00	1,891E+01	1,964E+01	2,885E+02
Tent	3,374E+00	2,520E+02	5,250E+03	3,780E-01	3,163E+00	1,478E+01	2,057E+01	2,650E+02

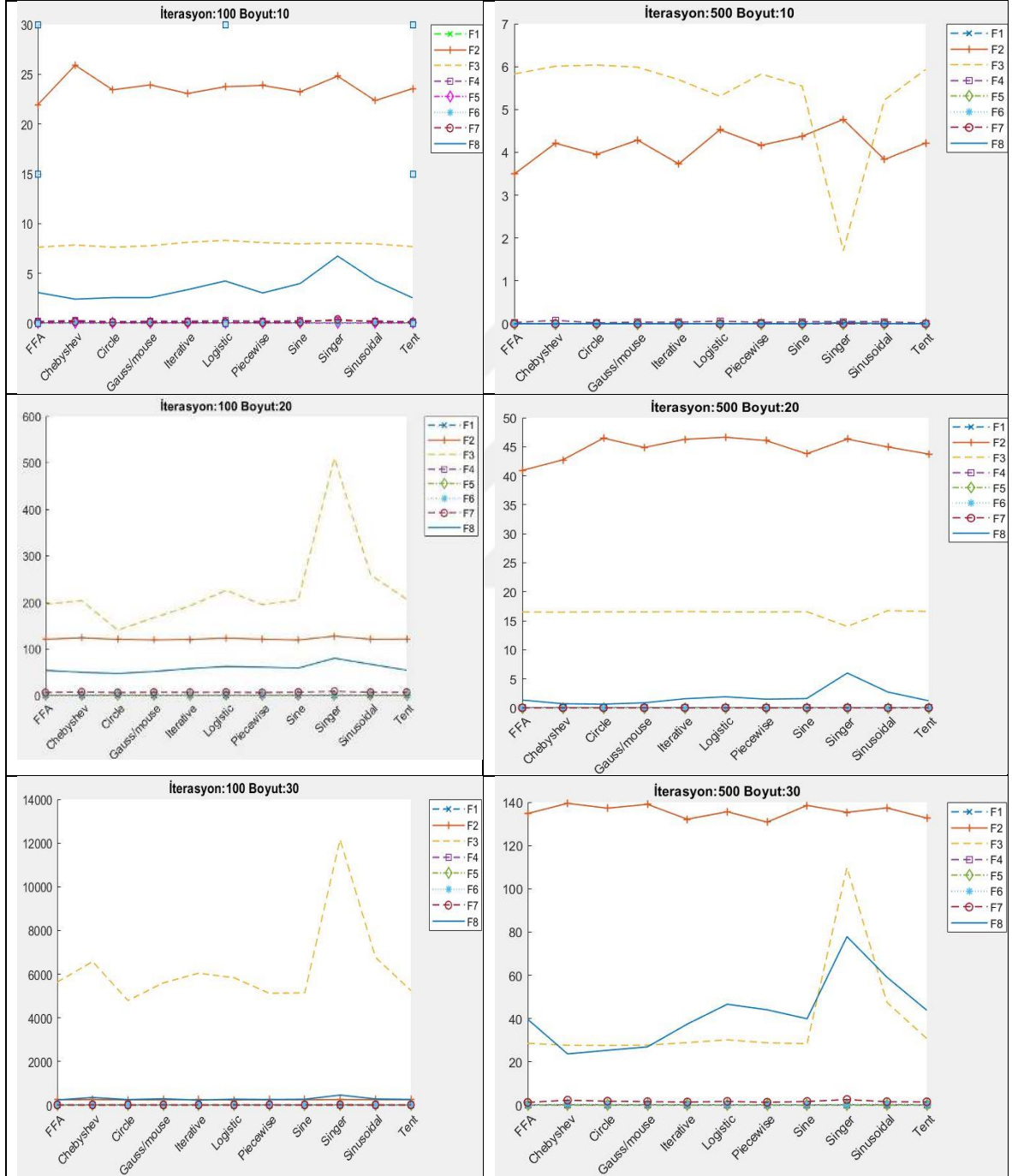
Tablo 5.6. 500 iterasyon ve 30 boyut sayısı ile 50 test sonucu ortalaması

	Sphere	Rastrigin	Rosenbrock	Griewank	Ackley	Levy	Alpine	Quintic
TAV	7,482E-08	1,348E+02	2,859E+01	3,504E-04	2,467E-04	2,311E-01	1,280E+00	3,969E+01
Chebyshev	9,704E-08	1,395E+02	2,767E+01	4,345E-04	3,184E-04	4,488E-01	2,303E+00	2,368E+01
Circle	6,673E-08	1,372E+02	2,760E+01	3,872E-04	2,603E-04	3,222E-01	1,832E+00	2,536E+01
Gauss/ mouse	6,839E-08	1,390E+02	2,772E+01	3,601E-04	2,425E-04	3,628E-01	1,637E+00	2,695E+01
Iterative	7,187E-08	1,321E+02	2,892E+01	3,791E-04	2,683E-04	1,854E-01	1,447E+00	3,747E+01
Logistic	7,894E-08	1,356E+02	3,020E+01	3,955E-04	2,744E-04	1,569E-01	1,734E+00	4,664E+01
Piecewise	6,416E-08	1,308E+02	2,881E+01	3,544E-04	2,580E-04	1,660E-01	1,346E+00	4,407E+01
Sine	7,271E-08	1,385E+02	2,840E+01	3,639E-04	2,638E-04	1,895E-01	1,707E+00	3,993E+01
Singer	1,004E-07	1,353E+02	1,099E+02	3,630E-04	6,140E-04	2,709E-01	2,602E+00	7,788E+01
Sinusoidal	7,186E-08	1,374E+02	4,753E+01	2,996E-04	2,655E-04	6,080E-01	1,583E+00	5,915E+01
Tent	7,251E-08	1,327E+02	3,077E+01	3,913E-04	2,533E-04	2,018E-01	1,538E+00	4,385E+01

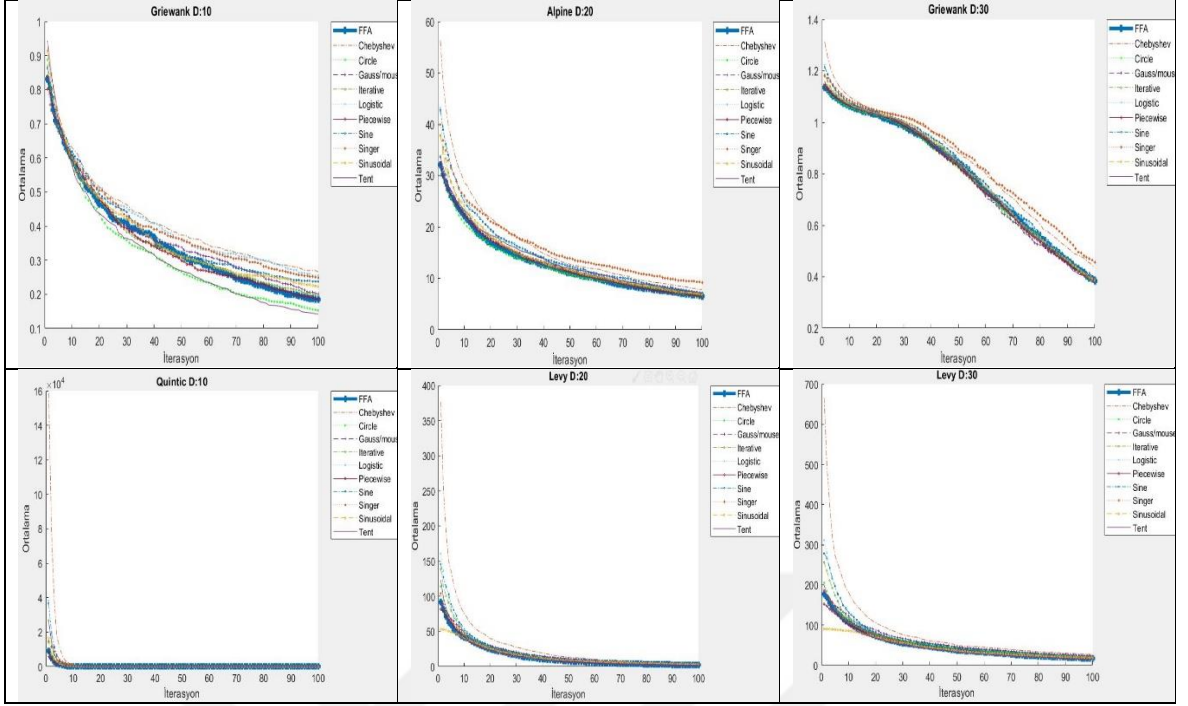
Tablo 5.5. ve Tablo 5.6. incelendiğinde, haritalar bazında ya da fonksiyonlar bazında orantılı bir değişiklik gözlemlenememiştir. Örneğin 100 iterasyona göre, 500 iterasyonda Sphere fonksiyonunun sonuçları iyileşirken; Griewank fonksiyon sonuçları gerilemiştir. Diğer taraftan Ackley fonksiyonunun sonuçlarında anlamlı bir değişiklik olmamıştır. Haritalarda ise Circle haritasının performansı azalırken Iterative haritasının performansı kısmi olarak artmıştır. Dolayısı ile 20 boyutta olduğu gibi 30 boyutta da haritalar bazında

büyük ve anlamlı değişiklikler gözlemlenemezken, fonksiyonlar bazında daha belirgin değişiklikler gözlemlenmiştir.

Bu yaklaşıma ait her boyut ve iterasyon için hesaplanan sonuçların grafik gösterimleri Şekil 5.9. da yer almaktadır.

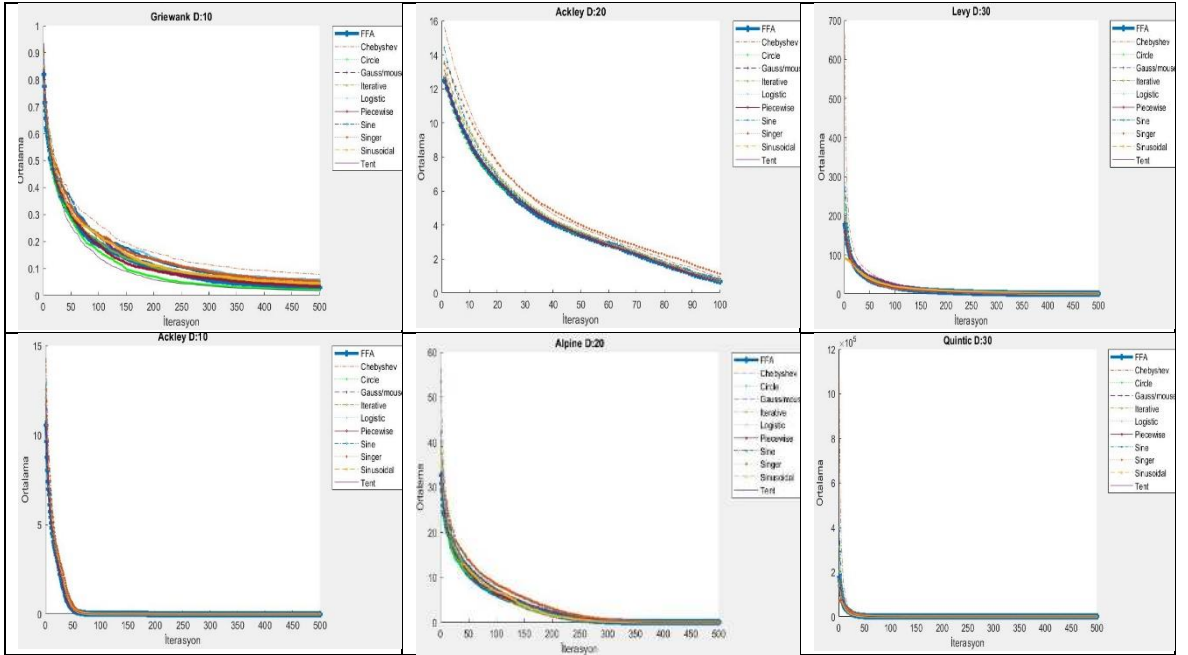


Şekil 5.9. Birinci yaklaşım test sonuçları



Şekil 5.10. Birinci yaklaşım 100 iterasyon yakınsama grafikleri

Şekil 5.10 da Griewank fonksiyonu ile yapılan 10 boyutlu testlerde bazı kaotik haritalı algoritmalarda daha iyi performans görülmektedir. Ayrıca Şekil 5.11 de, 50 farklı çalıştırma ile her iterasyon sonucunun ortalaması alınarak oluşturulan yakınsama grafiklerinin sonuçları bulunmaktadır.



Şekil 5.11. Birinci yaklaşım 500 iterasyon yakınsama grafikleri

5.2.2. Tüm Popülasyonun ve Tüm Boyutların Kaotik Haritalar ile Üretildiği İkinci Yaklaşım Test Sonuçları

50 bağımsız çalıştırma ile test edilen sonuçlar Tablo 5.7, Tablo 5.8, Tablo 5.9, Tablo 5.10, Tablo 5.11 ve Tablo 5.12 de gösterilmektedir. İlgili harita ve fonksiyon için daha başarılı olan değerler koyu punto ile belirtilmiştir.

Tablo 5.7. 100 iterasyon ve 10 boyut sayısı ile 50 test sonucu ortalaması

	Sphere	Rastrigin	Rosenbrock	Griewank	Ackley	Levy	Alpine	Quintic
TAV	4,977E-06	2,495E+01	7,828E+00	1,985E-01	3,514E-03	1,606E-03	5,665E-02	3,269E+00
Chebyshev	7,548E-06	2,438E+01	7,784E+00	2,554E-01	5,263E-03	4,364E-03	1,722E-01	2,288E+00
Circle	3,922E-06	2,343E+01	7,636E+00	1,563E-01	3,439E-03	2,095E-03	6,517E-02	2,215E+00
Gauss/								
mouse	4,619E-06	2,297E+01	7,664E+00	2,188E-01	3,868E-03	1,765E-03	9,585E-02	2,708E+00
Iterative	4,995E-06	2,434E+01	7,701E+00	2,094E-01	3,743E-03	1,561E-03	1,279E-01	3,600E+00
Logistic	5,178E-06	2,220E+01	7,915E+00	2,144E-01	3,923E-03	1,280E-03	1,306E-01	3,601E+00
Piecewise	4,242E-06	2,269E+01	7,784E+00	1,816E-01	3,593E-03	9,652E-04	6,214E-02	3,033E+00
Sine	6,302E-06	2,254E+01	7,696E+00	2,480E-01	4,266E-03	1,899E-03	1,274E-01	3,006E+00
Singer	8,307E-06	2,424E+01	7,225E+00	2,261E-01	5,118E-03	1,934E-03	2,283E-01	5,925E+00
Sinusoidal	1,152E-05	2,323E+01	8,024E+00	2,544E-01	8,467E-03	5,697E-03	3,996E-01	3,208E+00
Tent	1,660E-06	1,494E+00	9,757E-01	4,601E-04	6,334E-03	7,426E-04	1,001E-02	1,382E+00

Bu yaklaşımda, 10 boyut ve 100 iterasyon ile yapılan 50 bağımsız çalıştırmanın sonuçları Tablo 5.7. de bulunmaktadır. Tablo 5.7. incelendiğinde, Rastrigin fonksiyonunun, tüm kaotik haritalar için standart TAVA ‘dan daha iyi sonuçlar ürettiği görülmektedir. Benzer şekilde Rosenbrock fonksiyonu ile de Logistic ve Sinusoidal hariç tüm haritalarda standart TAVA ‘dan daha iyi sonuç alınmıştır.

Haritalar bazında göze çarpan en belirgin fark ise Tent haritasının sonuçlarında ortaya çıkmıştır. Bu harita ile üretilen değerler, Ackley hariç tüm test fonksiyonlarında daha iyi performans göstermiştir. Benzer şekilde Circle ve Piecewise haritaları da iki fonksiyon hariç olmak üzere daha iyi sonuçlar üretmiştir.

Tablo 5.8. 500 iterasyon ve 10 boyut sayısı ile 50 test sonucu ortalaması

	Sphere	Rastrigin	Rosenbrock	Griewank	Ackley	Levy	Alpine	Quintic
TAV	7,007E-34	4,641E+00	5,890E+00	3,078E-02	4,157E-15	9,013E-30	9,828E-06	9,374E-04
Chebyshev	1,639E-33	4,050E+00	6,057E+00	7,772E-02	4,299E-15	2,747E-28	8,007E-06	4,312E-05
Circle	6,108E-34	3,712E+00	5,984E+00	1,981E-02	4,086E-15	1,310E-29	3,760E-06	4,319E-04
Gauss/ mouse	6,654E-34	4,155E+00	5,993E+00	3,688E-02	4,086E-15	1,336E-29	3,993E-06	2,481E-04
Iterative	6,758E-34	4,485E+00	5,741E+00	3,479E-02	4,228E-15	1,296E-29	3,371E-06	6,656E-04
Logistic	1,058E-33	3,947E+00	5,558E+00	4,963E-02	4,086E-15	1,884E-29	5,675E-06	6,180E-04
Piecewise	6,851E-34	4,011E+00	5,759E+00	2,976E-02	4,086E-15	7,855E-30	3,127E-06	4,862E-04
Sine	7,663E-34	3,982E+00	5,831E+00	5,073E-02	4,157E-15	1,236E-29	4,834E-06	6,548E-04
Singer	1,378E-33	4,314E+00	3,264E+00	6,114E-02	4,228E-15	3,817E-29	2,291E-05	9,829E-03
Sinusoidal	1,463E-33	4,709E+00	5,459E+00	7,835E-02	4,228E-15	1,773E-19	1,984E-04	1,379E-03
Tent	2,342E-34	5,081E-03	8,088E-03	1,018E-06	5,009E-15	5,065E-16	7,944E-06	6,773E-04

Tablo 5.8. de 500 iterasyon ve 10 boyut sonuçları incelendiğinde, 100 iterasyon sonuçlarına paralel olduğu gözlemlenebilmektedir. 500 iterasyon testinde, Piecewise haritası performansını arttırıp tüm test fonksiyonlarında daha iyi sonuçlar vermiştir. Diğer taraftan Tent haritası 500 iterasyonda Ackley fonksiyonunun yanında Levy fonksiyonunda da standart TAVA 'ın gerisinde kalmıştır, ancak diğer 6 fonksiyonda hala standart TAVA 'dan daha iyi sonuçlar üretmeye devam etmiştir.

Tablo 5.9. 100 iterasyon ve 20 boyut sayısı ile 50 test sonucu ortalaması

	Sphere	Rastrigin	Rosenbrock	Griewank	Ackley	Levy	Alpine	Quintic
TAV	8,061E-02	1,203E+02	1,910E+02	1,497E-01	6,536E-01	2,228E+00	6,354E+00	5,752E+01
Chebyshev	1,235E-01	1,224E+02	1,914E+02	1,434E-01	9,820E-01	3,146E+00	7,575E+00	4,799E+01
Circle	7,825E-02	1,161E+02	1,506E+02	1,486E-01	6,719E-01	2,330E+00	6,393E+00	5,077E+01
Gauss/ mouse	8,977E-02	1,180E+02	1,718E+02	1,540E-01	6,930E-01	2,603E+00	6,711E+00	5,126E+01
Iterative	8,817E-02	1,202E+02	2,035E+02	1,691E-01	7,459E-01	2,320E+00	6,979E+00	6,047E+01
Logistic	1,033E-01	1,222E+02	2,239E+02	1,457E-01	7,215E-01	1,952E+00	7,424E+00	5,897E+01
Piecewise	8,278E-02	1,177E+02	1,828E+02	1,224E-01	6,866E-01	2,182E+00	6,444E+00	5,602E+01
Sine	9,566E-02	1,197E+02	1,987E+02	1,416E-01	7,434E-01	2,095E+00	7,310E+00	5,845E+01
Singer	1,222E-01	1,247E+02	3,678E+02	1,560E-01	9,751E-01	2,861E+00	8,296E+00	7,324E+01
Sinusoidal	1,274E-01	1,243E+02	2,282E+02	1,606E-01	9,546E-01	3,409E+00	8,198E+00	5,380E+01
Tent	1,019E-02	5,180E+00	4,438E+00	1,973E-03	1,685E-01	1,485E-02	6,278E-02	9,734E+00

Tablo 5.9. ile Tablo 5.7 karşılaştırıldığında, 100 iterasyon için boyut sayısının 10'dan 20'ye yükseltilmesi ile genelde kaotik haritaların başarısında kısmi bir düşüş olduğu anlaşılmaktadır. Bunun bir istisnası olarak ise Tent haritası 20 boyut için tüm test fonksiyonlarında daha iyi sonuç vermiştir.

Tablo 5.10. 500 iterasyon ve 20 boyut sayısı ile 50 test sonucu ortalaması

	Sphere	Rastrigin	Rosenbrock	Griewank	Ackley	Levy	Alpine	Quintic
TAV	8,699E-15	4,365E+01	1,656E+01	1,533E-05	1,019E-07	4,221E-08	1,163E-02	1,437E+00
Chebyshev	1,360E-14	4,896E+01	1,657E+01	2,217E-05	1,343E-07	1,489E-06	1,427E-02	7,397E-01
Circle	8,444E-15	4,518E+01	1,649E+01	1,365E-05	1,090E-07	2,454E-07	1,570E-02	8,731E-01
Gauss/ mouse	8,193E-15	4,268E+01	1,657E+01	1,970E-05	9,680E-08	1,144E-07	1,255E-02	8,900E-01
Iterative	9,723E-15	4,567E+01	1,656E+01	2,369E-05	1,021E-07	1,223E-07	9,952E-03	1,339E+00
Logistic	8,827E-15	4,520E+01	1,648E+01	1,671E-05	1,026E-07	3,121E-08	1,428E-02	1,427E+00
Piecewise	7,356E-15	4,257E+01	1,656E+01	1,783E-05	9,488E-08	4,289E-08	1,189E-02	1,429E+00
Sine	1,087E-14	4,549E+01	1,653E+01	1,642E-05	1,111E-07	5,480E-08	1,366E-02	1,340E+00
Singer	1,063E-14	4,632E+01	1,705E+01	2,648E-05	1,324E-07	7,435E-08	2,627E-02	4,009E+00
Sinusoidal	1,189E-14	4,447E+01	1,645E+01	1,058E-04	1,332E-07	1,176E-04	3,827E-02	1,026E+00
Tent	1,003E-15	8,305E-02	3,316E-01	3,773E-07	3,267E-04	4,750E-05	5,198E-04	5,809E-02

Tablo 5.9 ile Tablo 5.10 karşılaştırıldığında, 100 iterasyon ve 500 iterasyon arasındaki farklılıklar izlenebilir. Her ikisinde de Singer harita hiçbir fonksiyon ile standart TAVA 'dan daha iyi sonuçlar üretememiştir. Diğer haritaların performanslarında ise kısmi iyileşme ya da gerilemeler gözlemlenmiştir.

Tablo 5.11. 100 iterasyon ve 30 boyut sayısı ile 50 test sonucu ortalaması

	Sphere	Rastrigin	Rosenbrock	Griewank	Ackley	Levy	Alpine	Quintic
TAV	3,248E+00	2,533E+02	5,187E+03	3,788E-01	3,026E+00	1,679E+01	1,907E+01	2,633E+02
Chebyshev	4,387E+00	2,535E+02	6,908E+03	4,221E-01	3,367E+00	2,316E+01	2,156E+01	3,145E+02
Circle	2,973E+00	2,501E+02	5,115E+03	3,691E-01	3,130E+00	2,048E+01	1,745E+01	2,536E+02
Gauss/ mouse	3,361E+00	2,517E+02	5,561E+03	3,833E-01	3,169E+00	2,256E+01	1,951E+01	2,526E+02
Iterative	3,369E+00	2,469E+02	4,849E+03	3,772E-01	3,160E+00	1,938E+01	2,004E+01	2,636E+02
Logistic	3,426E+00	2,457E+02	5,781E+03	4,202E-01	3,228E+00	1,567E+01	2,063E+01	2,702E+02
Piecewise	3,280E+00	2,519E+02	5,432E+03	3,566E-01	3,053E+00	1,719E+01	1,960E+01	2,454E+02
Sine	3,436E+00	2,474E+02	5,548E+03	4,041E-01	3,221E+00	1,717E+01	2,023E+01	2,693E+02
Singer	4,095E+00	2,550E+02	8,057E+03	4,288E-01	3,351E+00	1,571E+01	2,265E+01	3,655E+02
Sinusoidal	4,476E+00	2,544E+02	6,148E+03	4,263E-01	3,375E+00	2,381E+01	2,163E+01	3,327E+02
Tent	4,369E-02	8,481E+00	6,814E+00	2,852E-03	2,086E-01	2,920E-02	1,245E-01	1,652E+01

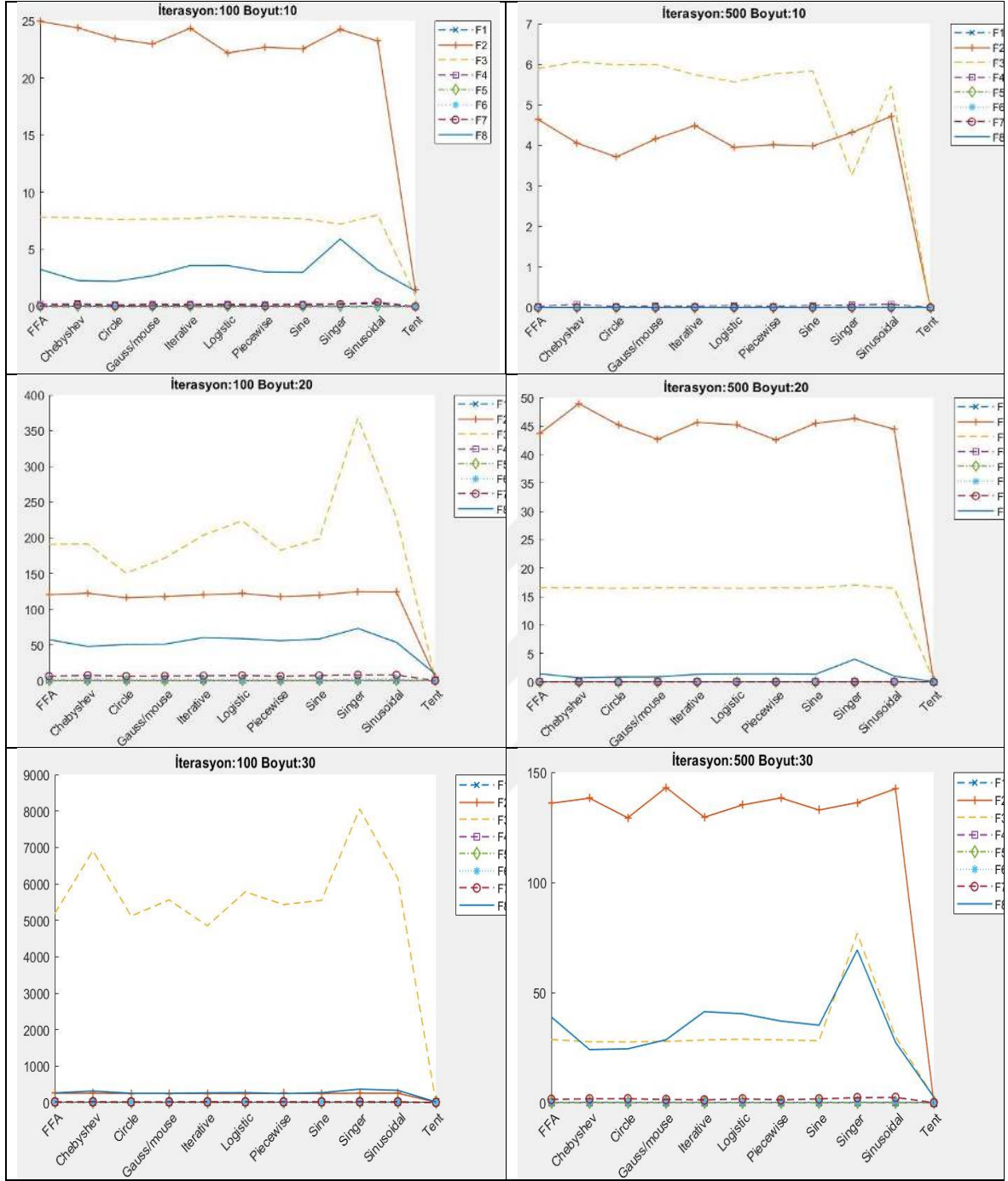
Tablo 5.11 de 30 boyutlu çalıştırma sonuçları incelendiğinde, 20 boyutta olduğu gibi Tent haritasının tüm test fonksiyonlarında standart TAVA 'dan daha iyi sonuçlar ürettiği görülebilmektedir. Benzer şekilde Circle haritasının da iyi olan performansı artarak devam etmiştir. Diğer taraftan Singer ve Sinusoidal haritalarının düşük performansı devam etmektedir. Bu kez Sinusoidal haritası hiçbir fonksiyon ile standart TAVA 'dan daha iyi sonuçlar üretememiştir.

Tablo 5.12. 500 iterasyon ve 30 boyut sayısı ile 50 test sonucu ortalaması

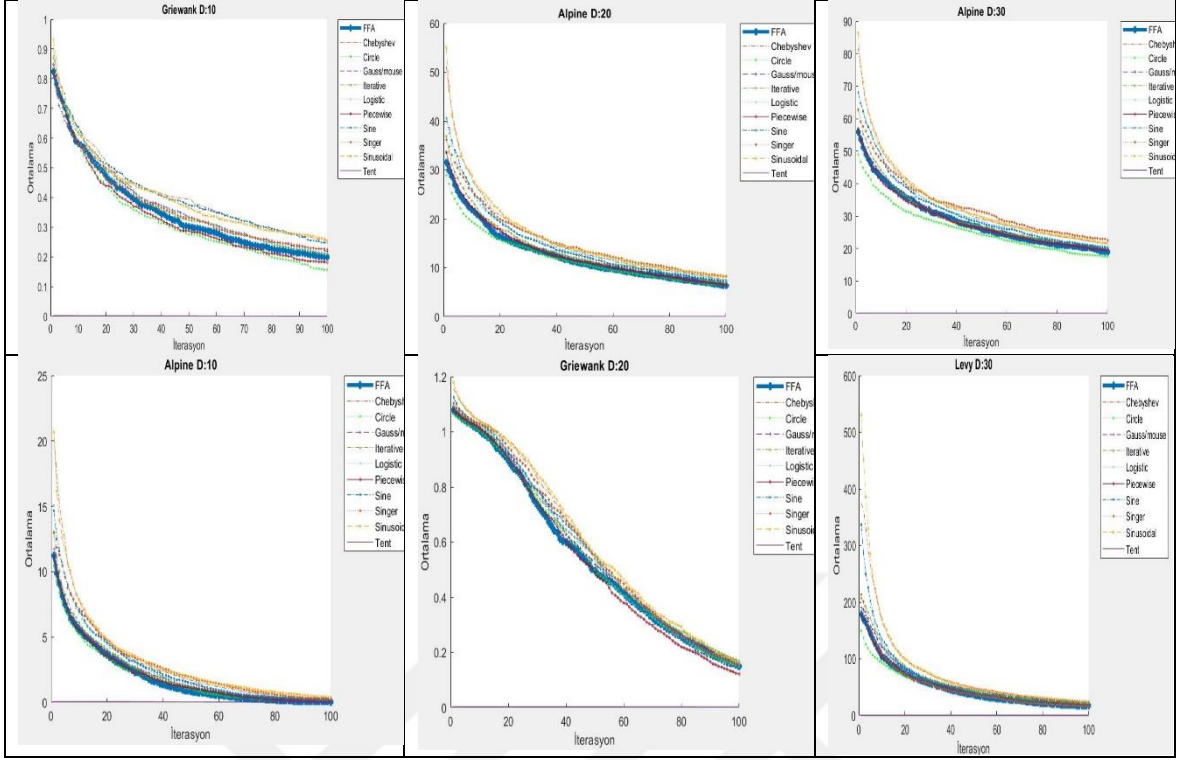
	Sphere	Rastrigin	Rosenbrock	Griewank	Ackley	Levy	Alpine	Quintic
TAV	6,519E-08	1,359E+02	2,865E+01	3,374E-04	2,510E-04	1,761E-01	1,525E+00	3,893E+01
Chebyshev	8,998E-08	1,382E+02	2,768E+01	4,406E-04	3,178E-04	3,694E-01	1,768E+00	2,412E+01
Circle	5,862E-08	1,293E+02	2,761E+01	3,389E-04	2,358E-04	3,293E-01	1,820E+00	2,446E+01
Gauss/ mouse	6,898E-08	1,431E+02	2,782E+01	4,086E-04	2,622E-04	2,880E-01	1,469E+00	2,858E+01
Iterative	7,253E-08	1,296E+02	2,847E+01	3,739E-04	2,492E-04	1,862E-01	1,266E+00	4,134E+01
Logistic	8,549E-08	1,351E+02	2,884E+01	4,054E-04	2,664E-04	1,429E-01	1,760E+00	4,039E+01
Piecewise	6,722E-08	1,383E+02	2,850E+01	3,745E-04	2,509E-04	1,567E-01	1,301E+00	3,708E+01
Sine	6,047E-08	1,328E+02	2,815E+01	4,330E-04	2,813E-04	1,490E-01	1,739E+00	3,520E+01
Singer	8,399E-08	1,362E+02	7,690E+01	3,605E-04	3,180E-04	2,906E-01	2,305E+00	6,934E+01
Sinusoidal	9,103E-08	1,425E+02	2,998E+01	3,314E-04	2,997E-04	4,348E-01	2,432E+00	2,743E+01
Tent	4,243E-09	4,150E-01	2,603E+00	1,673E-05	1,485E-02	1,081E-03	7,245E-03	2,613E+00

500 iterasyon ve 30 boyut ile yapılan testlerin sonuçları Tablo 5.12 üzerinden incelendiğinde, iterasyon sayısının artması ile köklü değişikliklerin olmadığı, kısmi iyileşme ya da gerilemelerin olduğu anlaşılabilmektedir. Bu anlamda 20 boyut için 100 ve 500 iterasyon arasındaki değişiklikler, 30 boyut için de tekrarlanmıştır. Örneğin Tent haritalı TAV algoritması, 100 iterasyon için hem 20, hem de 30 boyut için tüm testlerde daha iyi sonuç üretmiştir. 500 iterasyonda ise; hem 20, hem de 30 boyutta performansı kısmi olarak gerilemiştir.



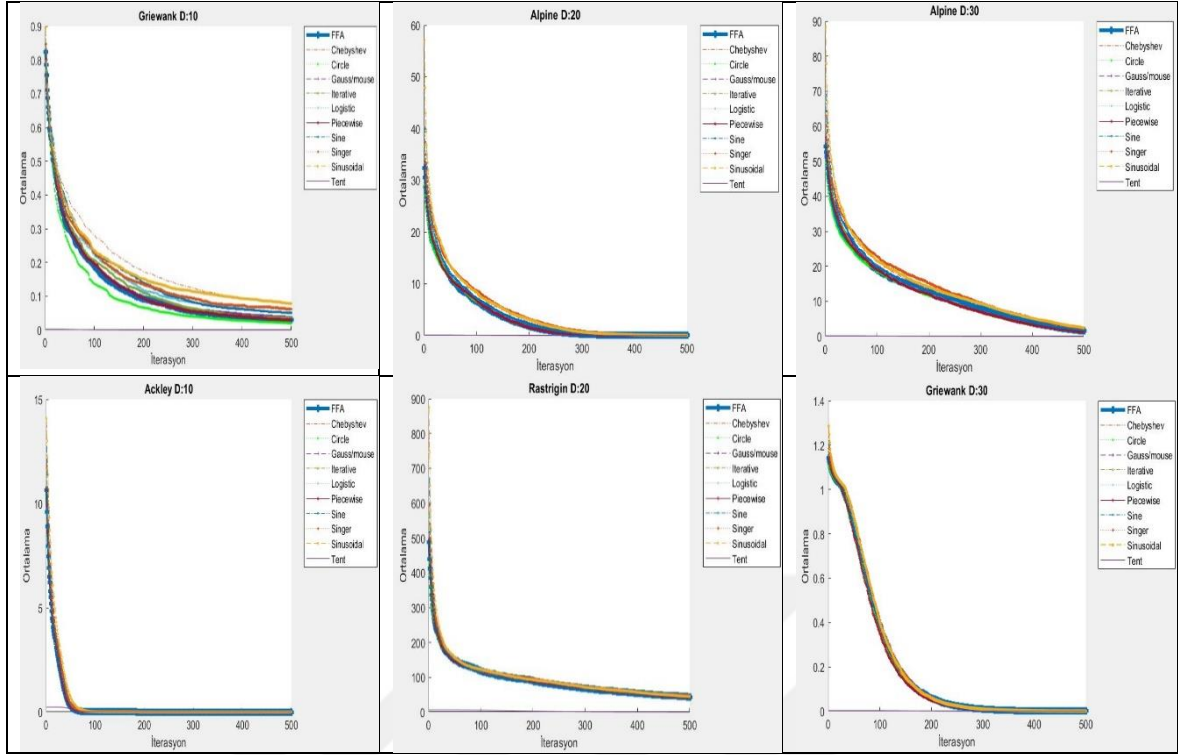


Şekil 5.12. İkinci yaklaşım test sonuçları



Şekil 5.13. İkinci yaklaşım 100 iterasyon yakınsama grafikleri

Bu yaklaşım için genel bir gözlem olarak, Tent haritasının performansı iterasyon sayısı arttıkça bir miktar düşüş gösterse de, bariz bir şekilde daha iyi sonuçlar ürettiği söylenebilir. Sonuçların grafik gösterimleri Şekil 5.12 de yer almaktadır. 50 farklı çalıştırma ile her iterasyon sonucunun ortalaması alınarak yakınsama grafikleri oluşturulmuştur ve Şekil 5.13 ve Şekil 5.14 de gösterilmiştir.



Şekil 5.14. İkinci yaklaşım 500 iterasyon yakınsama grafikleri

5.2.3. Tüm Popülasyonun Tek Boyutlu Şekilde Kaotik Haritalar ile Üretildiği Üçüncü Yaklaşım Test Sonuçları

50 bağımsız çalıştırma ile test edilen sonuçlar Tablo 5.13, Tablo 5.14, Tablo 5.15, Tablo 5.16, Tablo 5.17 ve Tablo 5.18 de gösterilmektedir. İlgili harita ve fonksiyon için daha başarılı olan değerler koyu punto ile belirtilmiştir.

Tablo 5.13. 100 iterasyon ve 10 boyut sayısı ile 50 test sonucu ortalaması

	Sphere	Rastrigin	Rosenbrock	Griewank	Ackley	Levy	Alpine	Quintic
TAV	4,339E-06	2,372E+01	7,791E+00	1,693E-01	3,181E-03	9,319E-04	6,505E-02	3,262E+00
Chebyshev	3,470E-06	1,212E+01	5,431E+00	1,384E-02	5,909E-03	2,617E-03	1,077E-01	1,701E+00
Circle	1,383E-06	2,226E+00	1,690E+00	3,354E-04	6,138E-03	7,596E-04	7,095E-03	6,139E-01
Gauss/ mouse	1,788E-06	6,440E+00	3,441E+00	1,093E-03	6,767E-03	2,294E-03	1,627E-02	1,025E+00
Iterative	2,328E-06	8,189E+00	3,921E+00	6,417E-03	6,556E-03	2,676E-03	3,198E-02	1,136E+00
Logistic	2,059E-06	7,348E+00	3,227E+00	4,419E-03	6,300E-03	4,093E-03	2,882E-02	1,469E+00
Piecewise	1,617E-06	4,850E+00	3,632E+00	6,840E-04	6,859E-03	2,062E-03	9,227E-03	1,079E+00
Sine	2,224E-06	6,160E+00	4,318E+00	3,204E-03	5,934E-03	4,256E-03	3,144E-02	1,165E+00
Singer	2,179E-06	6,715E+00	3,141E+00	2,291E-03	6,934E-03	1,741E-03	1,991E-02	1,736E+00
Sinusoidal	2,547E-04	1,487E+01	5,521E+00	7,398E-02	7,411E-01	4,625E+00	2,661E-01	2,829E+00
Tent	1,524E-06	1,646E+00	9,041E-01	4,601E-04	6,930E-03	7,549E-04	9,706E-03	0,000E+00

Tablo 5.13. incelendiğinde, bu yönteme ait 100 iterasyon ve 10 boyut testlerinde Ackley ve Levy fonksiyonları dışında, neredeyse tüm haritaların daha iyi sonuç verdiği görülmektedir. Örneğin Rastrigin, Rosenbrock, Griewank ve Quintic fonksiyonları ile tüm kaotik haritalar daha iyi sonuç vermiştir. Bununla birlikte, Ackley fonksiyonu ile hiçbir kaotik harita daha iyi sonuç verememiştir.

Tablo 5.14. 500 iterasyon ve 10 boyut sayısı ile 50 test sonucu ortalaması

	Sphere	Rastrigin	Rosenbrock	Griewank	Ackley	Levy	Alpine	Quintic
TAV	6,155E-34	3,727E+00	5,769E+00	3,130E-02	4,086E-15	7,520E-30	3,718E-06	1,603E-04
Chebyshev	5,306E-34	2,919E+00	3,238E+00	7,055E-04	4,512E-15	9,345E-20	6,703E-03	7,855E-04
Circle	2,107E-34	3,913E-01	1,317E-01	1,589E-07	1,226E-14	5,253E-16	2,162E-06	4,054E-05
Gauss/ mouse	1,986E-34	1,294E+00	1,060E+00	1,827E-06	6,200E-14	1,996E-13	5,114E-03	6,337E-04
Iterative	2,218E-34	3,244E+00	2,009E+00	4,359E-06	1,197E-14	7,273E-10	4,447E-03	1,380E-03
Logistic	3,475E-34	2,183E+00	1,631E+00	6,711E-06	3,634E-14	2,357E-16	6,386E-03	4,266E-03
Piecewise	1,875E-34	1,792E+00	1,530E+00	5,718E-07	6,803E-14	5,004E-14	1,127E-03	8,152E-04
Sine	3,013E-34	3,030E+00	1,814E+00	6,418E-06	5,436E-15	1,860E-14	7,479E-03	3,190E-03
Singer	2,175E-34	1,972E+00	1,760E+00	4,904E-06	5,933E-15	1,601E-15	5,417E-03	1,151E-03
Sinusoidal	2,101E-32	3,219E+00	2,590E+00	7,329E-02	3,150E-13	3,714E-05	4,196E-03	4,040E-03
Tent	1,796E-34	5,371E-03	9,430E-03	3,412E-07	7,212E-15	9,595E-18	2,312E-05	0,000E+00

10 boyut için iterasyon sayısı arttırılınca, Alpine ve Quintic fonksiyonları ile yapılan testlerin performansının düştüğü görülürken, Sphere, Rastrigin, Rosenbrock ve Griewank fonksiyonlarında ise iyi sonuçların devam ettiği görülebilmektedir.

Tablo 5.15. 100 iterasyon ve 20 boyut sayısı ile 50 test sonucu ortalaması

	Sphere	Rastrigin	Rosenbrock	Griewank	Ackley	Levy	Alpine	Quintic
TAV	8,345E-02	1,201E+02	1,997E+02	1,548E-01	6,707E-01	2,053E+00	6,305E+00	5,716E+01
Chebyshev	3,152E-02	3,254E+01	3,981E+01	2,843E-02	6,207E-01	3,676E-01	5,715E-01	1,771E+01
Circle	8,120E-03	8,076E+00	6,057E+00	2,341E-03	1,942E-01	2,217E-02	2,346E-02	3,717E+00
Gauss/ mouse	1,166E-02	1,229E+01	1,125E+01	7,609E-03	4,330E-01	7,118E-02	1,203E-01	7,134E+00
Iterative	1,719E-02	1,883E+01	1,582E+01	9,963E-03	5,969E-01	1,267E-01	1,294E-01	9,506E+00
Logistic	1,973E-02	2,370E+01	1,971E+01	1,107E-02	4,819E-01	1,839E-01	2,236E-01	9,729E+00
Piecewise	1,509E-02	1,576E+01	1,285E+01	9,863E-03	3,468E-01	7,456E-02	9,843E-02	6,673E+00
Sine	1,917E-02	2,336E+01	2,398E+01	1,006E-02	4,188E-01	1,907E-01	2,287E-01	1,089E+01
Singer	1,606E-02	2,121E+01	1,519E+01	9,800E-03	3,694E-01	1,189E-01	1,444E-01	1,037E+01
Sinusoidal	4,287E-01	7,224E+01	5,586E+02	1,325E-01	3,235E+00	3,405E+01	2,669E+00	4,202E+01
Tent	9,417E-03	5,205E+00	4,449E+00	1,982E-03	1,662E-01	1,446E-02	5,970E-02	0,000E+00

Bu yaklaşıma ait 20 boyut 100 iterasyon testi Tablo 5.15. de incelendiğinde, Sinusoidal hariç tüm kaotik haritaların, tüm test fonksiyonlarında daha iyi sonuç ürettiği görülmektedir.

Tablo 5.16. 500 iterasyon ve 20 boyut sayısı ile 50 test sonucu ortalaması

	Sphere	Rastrigin	Rosenbrock	Griewank	Ackley	Levy	Alpine	Quintic
TAV	8,344E-15	4,052E+01	1,656E+01	2,384E-05	1,060E-07	1,749E-08	1,147E-02	1,247E+00
Chebyshev	3,233E-15	1,821E+01	9,516E+00	3,324E-06	7,040E-05	1,041E-04	7,202E-02	3,089E-01
Circle	8,423E-16	1,691E+00	8,508E-01	3,466E-07	4,946E-04	5,693E-05	3,009E-04	3,028E-02
Gauss/ mouse	1,541E-15	7,956E+00	3,950E+00	6,360E-07	5,454E-04	2,261E-04	1,828E-03	6,542E-02
Iterative	1,642E-15	8,429E+00	4,310E+00	1,119E-06	3,548E-04	3,934E-04	2,291E-02	8,406E-02
Logistic	2,016E-15	1,427E+01	7,571E+00	1,208E-06	2,958E-04	2,663E-04	3,140E-03	1,226E-01
Piecewise	1,399E-15	5,468E+00	5,736E+00	5,247E-07	3,631E-04	1,125E-04	6,988E-04	5,945E-02
Sine	1,815E-15	5,959E+00	6,161E+00	1,088E-06	4,833E-04	1,487E-04	2,248E-02	1,106E-01
Singer	2,070E-15	9,275E+00	3,412E+00	5,229E-07	5,635E-04	8,207E-05	6,082E-03	1,566E-01
Sinusoidal	2,934E-14	3,158E+01	7,418E+00	1,268E-05	3,461E-01	1,044E+01	2,764E-01	6,598E-01
Tent	1,074E-15	8,783E-02	2,965E-01	3,805E-07	6,490E-04	3,883E-05	5,315E-04	0,000E+00

20 boyut için iterasyon sayısı 500 ile test edildiğinde, Ackley ve Levy fonksiyonlarında hiçbir kaotik haritanın daha iyi sonuç vermediği görülmüştür. Diğer tüm test fonksiyonlarında ise kaotik haritaların başarısı devam etmektedir.

Tablo 5.17. 100 iterasyon ve 30 boyut sayısı ile 50 test sonucu ortalaması

	Sphere	Rastrigin	Rosenbrock	Griewank	Ackley	Levy	Alpine	Quintic
TAV	3,240E+00	2,503E+02	4,942E+03	3,593E-01	3,113E+00	1,775E+01	1,889E+01	2,440E+02
Chebyshev	6,993E-01	5,854E+01	5,650E+01	7,965E-02	1,729E+00	6,147E-01	1,405E+00	2,896E+01
Circle	9,594E-02	1,278E+01	8,781E+00	4,579E-03	2,403E-01	4,861E-02	4,639E-02	6,311E+00
Gauss/ mouse	2,324E-01	2,863E+01	2,611E+01	2,097E-02	7,286E-01	2,149E-01	3,139E-01	1,251E+01
Iterative	2,252E-01	3,393E+01	2,583E+01	3,389E-02	8,172E-01	2,166E-01	3,709E-01	1,695E+01
Logistic	3,745E-01	3,674E+01	3,584E+01	2,893E-02	8,396E-01	3,903E-01	2,609E-01	1,684E+01
Piecewise	2,832E-01	2,937E+01	1,974E+01	2,154E-02	6,462E-01	3,870E-01	4,196E-01	1,287E+01
Sine	2,745E-01	3,800E+01	3,819E+01	3,348E-02	1,134E+00	5,821E-01	4,477E-01	1,845E+01
Singer	3,269E-01	3,248E+01	2,195E+01	3,824E-02	8,403E-01	2,507E-01	3,417E-01	1,479E+01
Sinusoidal	6,239E+00	1,648E+02	1,391E+04	2,521E-01	4,439E+00	7,496E+01	1,140E+01	2,006E+03
Tent	4,339E-02	8,467E+00	6,814E+00	2,817E-03	2,096E-01	2,866E-02	1,247E-01	0,000E+00

Tablo 5.17. incelendiğinde, 100 iterasyon için 30 boyut sonuçlarının, Tablo 5.15. de bulunan 20 boyut sonuçları ile paralel olduğu görülebilmektedir. 30 boyut testinde de 20 boyutta olduğu gibi Sinusoidal hariç tüm kaotik haritalar, tüm test fonksiyonu ile standart TAVA 'ya göre daha iyi sonuçlar vermiştir.

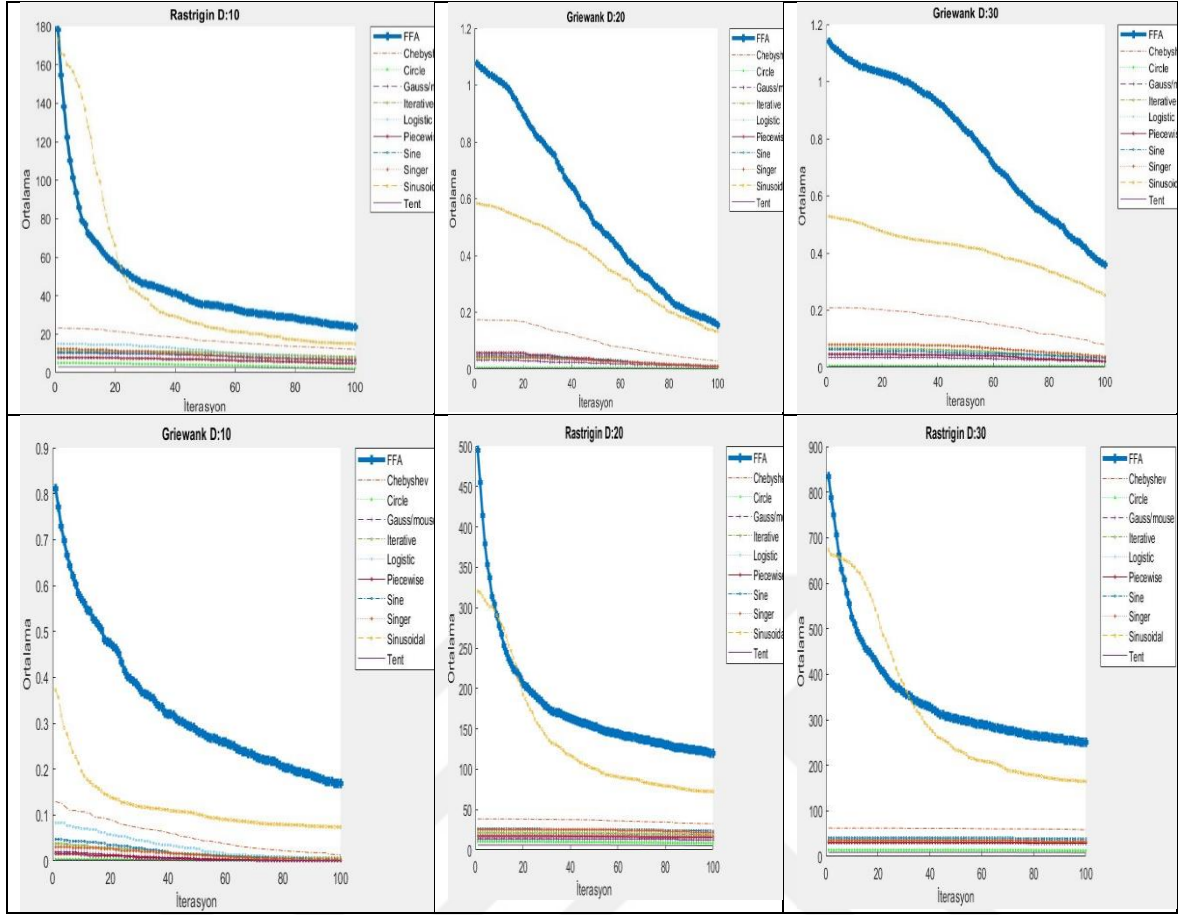
Tablo 5.18. 500 iterasyon ve 30 boyut sayısı ile 50 test sonucu ortalaması

	Sphere	Rastrigin	Rosenbrock	Griewank	Ackley	Levy	Alpine	Quintic
TAV	6,082E-08	1,414E+02	2,857E+01	3,430E-04	2,545E-04	2,593E-01	1,340E+00	4,046E+01
Chebyshev	1,397E-08	2,639E+01	1,907E+01	6,530E-05	1,554E-02	9,783E-03	5,849E-02	7,010E+00
Circle	3,613E-09	3,829E+00	5,218E+00	1,315E-05	1,405E-02	2,295E-03	6,180E-03	8,048E-01
Gauss/ mouse	7,114E-09	1,484E+01	1,471E+01	2,062E-05	2,129E-02	4,820E-03	3,156E-02	1,316E+00
Iterative	1,051E-08	1,277E+01	1,425E+01	2,938E-05	2,112E-02	1,546E-02	2,108E-02	3,726E+00
Logistic	9,180E-09	2,093E+01	1,268E+01	2,858E-05	1,809E-02	1,601E-02	3,011E-02	3,395E+00
Piecewise	5,868E-09	7,697E+00	1,669E+01	2,405E-05	2,111E-02	4,167E-03	9,019E-03	1,954E+00
Sine	9,102E-09	1,333E+01	1,647E+01	3,622E-05	1,748E-02	1,346E-02	7,094E-02	3,264E+00
Singer	6,779E-09	1,210E+01	1,240E+01	3,468E-05	6,211E-02	8,959E-03	2,672E-02	5,367E+00
Sinusoidal	1,422E-07	6,824E+01	1,379E+01	2,432E-04	7,371E-01	3,678E+01	1,676E+00	1,226E+01
Tent	4,525E-09	4,300E-01	2,661E+00	1,511E-05	1,588E-02	1,243E-03	7,099E-03	0,000E+00

Tablo 5.18. de 500 iterasyon ve 30 boyut testlerinin sonuçları bulunmaktadır. Buna göre Ackley fonksiyonu testlerinde hiçbir kaotik harita standart TAVA 'dan daha iyi sonuç üretememiştir. Diğer boyut ve iterasyon testlerinde olduğu gibi, 500 iterasyon ve 30 boyut için de Sinusoidal hariç diğer haritaların genel olarak daha iyi sonuçlar verdiği gözlemlenmiştir.

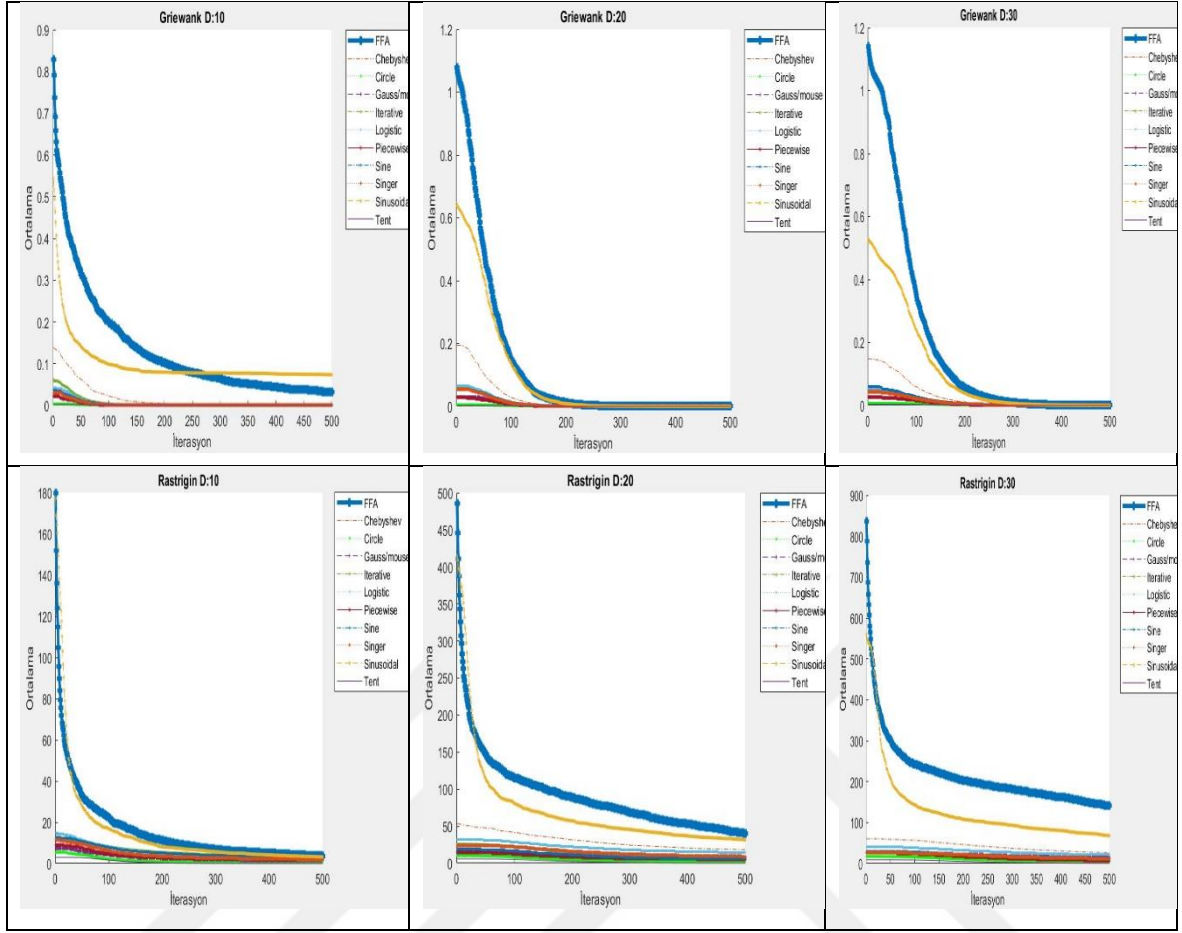


Şekil 5.15. Üçüncü yaklaşım test sonuçları



Şekil 5.16. Üçüncü yaklaşım 100 iterasyon yakınsama grafikleri

Şekil 5.16 ‘da bu yaklaşımın 100 iterasyon ve 50 bağımsız çalıştırma ile elde edilen sonuçlarının yakınsama grafiklerinde neredeyse tüm kaotik haritalı TAV algoritmalarının standart TAV algoritmalarından daha başarılı olduğu bariz bir şekilde görülebilmektedir.



Şekil 5.17. Üçüncü yaklaşım 500 iterasyon yakınsama grafikleri

Bu yaklaşımın tüm sonuçları incelendiğinde, boyut sayısı arttıkça standart TAV algoritmasına göre başarısının arttığı, iterasyon sayısı arttıkça ise başarısının kısmi olarak azaldığı ancak tüm senaryolarda standart TAVA 'ya göre daha iyi sonuçlar elde edilebildiği görülmüştür. Sonuçların grafik gösterimleri Şekil 5.15 de yer almaktadır. 50 farklı çalıştırma ile her iterasyon sonucunun ortalaması alınarak yakınsama grafikleri oluşturulmuştur ve Şekil 5.16 ve Şekil 5.17 de gösterilmiştir.

6. SONUÇ

Tarım Arazisi Verimliliği algoritmasının başlangıç popülasyonunun kaotik haritalar ile üretilmesi amacı ile üç farklı yaklaşım benimsenmiş ve her üç yaklaşım da 50 farklı bağımsız çalıştırma ile 10, 20 ve 30 boyut ile hem 100 hem de 500 iterasyon sayıları ile test edilmiştir. Yapılan testler sonucunda;

Tüm boyut değerlerinin kaotik haritalar ile üretildiği birinci yaklaşımda genel olarak standart Tarım Arazisi Verimliliği algoritması daha iyi sonuçlar verirken, sadece belirli kaotik haritalar, bazı kalite testi fonksiyonu sonuçlarına göre standart Tarım Arazisi Verimliliği algoritmasından daha iyi sonuçlar elde edilebilmektedir.

Tüm popülasyonun ve tüm boyut değerlerinin kaotik haritalar ile üretildiği ikinci yaklaşımda alınan sonuçlarda genel olarak standart Tarım Arazisi Algoritması ile benzer ya da daha iyi sonuçlar elde edildiği görülmüştür. Özellikle Çadır (Tent) kaotik haritasının uygulandığı Tarım Arazisi Verimliliği algoritmasının hemen her boyut ve iterasyon için daha başarılı sonuçlar ürettiği gözlemlenmiştir.

Tüm popülasyonun tek boyutlu şekilde kaotik haritalar ile üretildiği üçüncü yaklaşım ile alınan sonuçların genel olarak daha iyi performansa sahip olduğu görülmüştür. Bu yaklaşım da, tüm boyut değerleri eşit olduğu için, boyut sayısının artmasından etkilenmemekte ve Tarım Arazisi Verimliliği algoritmasına göre performansı artmaktadır.

KAYNAKLAR

- [1] Cigal, T., 2018. Sürekli zamanlı kaotik sistem tabanlı balina optimizasyon algoritması'nın geliştirilmesi, *Yüksek lisans Tezi*, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elazığ.)
- [2] ERDOĞMUŞ, P. (2016). Doğadan esinlenen optimizasyon algoritmaları ve optimizasyon algoritmalarının optimizasyonu. *Düzce üniversitesi bilim ve teknoloji dergisi*, 4(1).
- [3] D. Karaboğa, *Yapay Zeka Optimizasyon Algoritmaları*, Nobel Yayın Dağıtım. 2004.
- [4] Alataş, B.,2007. Kaotik haritalı parçacık sürü optimizasyonu algoritmaları geliştirme, *Doktora Tezi*, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elazığ.)
- [5] Altay, E. V., & Alatas, B. (2019). Bird swarm algorithms with chaotic mapping. *Artificial Intelligence Review*, 1-42.
- [6] Demir, F. B., Tuncer, T., & Kocamaz, A. F. Lojistik-Gauss Harita Tabanlı Yeni Bir Kaotik Sürü Optimizasyon Yöntemi. *Anatolian Science-Bilgisayar Bilimleri Dergisi*, 47-53.
- [7] S. Saremi, S. Mirjalili, and A. Lewis, “Biogeography-based optimisation with chaos,” *Neural Comput. Appl.*, vol. 25, no. 5, pp. 1077–1097, 2014.
- [8] A. H. Gandomi and X. S. Yang, “Chaotic bat algorithm,” *J. Comput. Sci.*, vol. 5, no. 2, pp. 224–232, 2014.
- [9] G. I. Sayed, A. Tharwat, and A. E. Hassanien, “Chaotic dragonfly algorithm: an improved metaheuristic algorithm for feature selection,” *Appl. Intell.*, vol. 49, no. 1, pp. 188–205, 2019.
- [10] R. M. Rizk-Allah, A. E. Hassanien, and S. Bhattacharyya, “Chaotic crow search algorithm for fractional optimization problems,” *Appl. Soft Comput. J.*, vol. 71, pp. 1161–1175, 2018.
- [11] M. Mitić, N. Vuković, M. Petrović, and Z. Miljković, “Chaotic fruit fly optimization algorithm,” *Knowledge-Based Syst.*, vol. 89, pp. 446–458, 2015.
- [12] M. Kohli and S. Arora, “Chaotic grey wolf optimization algorithm for constrained optimization problems,” *J. Comput. Des. Eng.*, vol. 5, no. 4, pp. 458–472, 2018.
- [13] G. Kaur and S. Arora, “Chaotic whale optimization algorithm,” *J. Comput. Des. Eng.*, vol. 5, no. 3, pp. 275–284, 2018.
- [14] A. H. Gandomi, X. S. Yang, S. Talatahari, and A. H. Alavi, “Firefly algorithm with

- chaos,” *Commun. Nonlinear Sci. Numer. Simul.*, vol. 18, no. 1, pp. 89–98, 2013.
- [15] G. G. Wang *et al.*, “Chaotic Krill Herd algorithm,” *Commun. Nonlinear Sci. Numer. Simul.*, vol. 274, no. 1, pp. 17–34, 2013.
- [16] E. Tanyildizi and G. Demir, “Nümerik Optimizasyon için Kaotik Altın Sinüs Algoritması Chaotic Gold Sine Algorithm for Numerical Optimization,” vol. 31, no. 1, pp. 91–97, 2019.
- [17] A. Rezaee Jordehi, “Chaotic bat swarm optimisation (CBSO),” *Appl. Soft Comput. J.*, vol. 26, pp. 523–530, 2015.
- [18] P. Statement *et al.*, “Models for Decision Making,” pp. 1–8.
- [19] M. Öztürk, “ÇOK AŞAMALI TEDARİK ZİNCİRİ OPTİMİZASYONU PROBLEMİNİN YAYILAN AĞAÇ TABANLI GENETİK ALGORİTMA İLE ÇÖZÜMÜ,” 2008.
- [20] M. Türkay, “International Institute for Applied Systems Analysis (IIASA),” *Encycl. Oper. Res. Manag. Sci.*, pp. 805–806, 2013.
- [21] Y. Özsağlam and M. Çunkaş, “Optimizasyon Problemlerinin Çözümü için Parçacık Sürü Optimizasyonu Algoritması,” *Politek. Derg.*, vol. 11, no. 4, pp. 299–305, 2008.
- [22] I. Fister, X. S. Yang, J. Brest, and D. Fister, “A brief review of nature-inspired algorithms for optimization,” *Elektroteh. Vestnik/Electrotechnical Rev.*, vol. 80, no. 3, pp. 116–122, 2013.
- [23] H. Shayanfar and F. S. Gharehchopogh, “Farmland fertility: A new metaheuristic algorithm for solving continuous optimization problems,” *Appl. Soft Comput. J.*, vol. 71, pp. 728–746, 2018.
- [24] T. Tanyıldızı, Erkan;Cigal, “Kaotik Haritalı Balina Optimizasyon Algoritmaları Erkan TANYILDIZI 1* , Tuncay CİGAL 2 1,” vol. 29, no. 1, pp. 309–319, 2017.
- [25] Kazak, N.,2011. Geliştirilmiş yerçekimsel arama algoritması, *Yüksek Lisans Tezi*, Bilecik Üniversitesi, Fen Bilimleri Enstitüsü, Bilecik.
- [26] D. Demirel and A. Karci, “Fizik Tabanlı Meta-Sezgisel Optimizasyon Algoritmalarının Performans Karşılaştırılması,” no. September, 2018.
- [27] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer,” *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014.
- [28] F. A. ŞENEL, A. S. YÜKSEL, F. GÖKÇE, and T. YİĞİT, “Gri kurt optimizasyon algoritması ile iki boyutlu dizilim yazılımının geliştirilmesi,” *Balıkesir Üniversitesi Fen Bilim. Enstitüsü Derg.*, vol. 20, no. 2, pp. 1–14, 2018.

- [29] Karaarslan, E., & Zengin, K. (2016). Ateş Böceği Algoritması ile Haftalık Ders Programı Hazırlama. *EEB 2016 Elektrik-Elektronik ve Bilgisayar Sempozyumu*, 11-13.
- [30] Akyol, S., 2013. *Güncel sürü zekâsi algoritmalarıyla sınıflandırma kurallarının keşfi*, Yüksek Lisans Tezi, Tunceli Üniversitesi, Fen Bilimleri Enstitüsü, Tunceli.
- [31] Kızılloluk, S. ve Alataş, B., 2012. Sosyal tabanlı güncel sezgisel optimizasyon algoritmaları, Cumhuriyet Üniversitesi İktisadi ve İdari Bilimler Dergisi, 13, 2, 39-56.
- [32] Altunbey, F. ve Alataş, B. 2015. Sosyal ağ analizi için sosyal tabanlı yapay zekâ optimizasyon algoritmalarının incelenmesi. *International Journal of Pure and Applied Sciences*, 1(1), 33-52.
- [33] Q. Bai, “Analysis of Particle Swarm Optimization Algorithm,” *Comput. Inf. Sci.*, vol. 3, no. 1, pp. 180–184, 2010.
- [34] A. A. Heidari, R. Ali Abbaspour, and A. Rezaee Jordehi, “An efficient chaotic water cycle algorithm for optimization tasks,” *Neural Comput. Appl.*, vol. 28, no. 1, pp. 57–85, 2017.

ÖZGEÇMİŞ

1984 yılında Diyarbakır'da doğan Meral KAYA ilköğretim ve lise eğitimini Diyarbakır'da tamamladı. 2002 yılında Diyarbakır Anadolu Lisesini bitirdikten sonra 2003 yılında Mersin Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği bölümüne yerleşti. 2007 yılında Mersin Üniversitesi Bilgisayar Mühendisliği bölümünden mezun oldu. 2009 yılında Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim dalında yüksek lisansa başladı ve halen devam etmektedir. 2010 yılında Bayındırlık ve İskan Bakanlığına Mühendis olarak atandı. 2017 yılından bu yana Çevre ve Şehircilik Bakanlığı, Coğrafi Bilgi Sistemleri Genel Müdürlüğü, Yazılım Teknolojileri Daire Başkanlığında Veri Yönetimi Şube Müdür Vekili olarak görev yapmaktadır.