

T.C.  
BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
ELEKTRONİK VE BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**CMAC SİNİR AĞI ÖĞRENME YÖNTEMİ KULLANILARAK YENİ MELEZ GSA  
VE PSO OPTİMİZASYON ALGORİTMALARININ GELİŞTİRİLMESİ VE  
BAŞARIM ANALİZİ**

DOKTORA TEZİ

NAZMİYE EBRU BULUT

TEZ DANIŞMANI  
DOÇ. DR. EMRE DANDIL

BİLECİK, 2025

10734240

T.C.  
BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
ELEKTRONİK VE BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**CMAC SINIR AĞI ÖĞRENME YÖNTEMİ KULLANILARAK YENİ MELEZ GSA  
VE PSO OPTİMİZASYON ALGORİTMALARININ GELİŞTİRİLMESİ VE  
BAŞARIM ANALİZİ**

DOKTORA TEZİ

NAZMİYE EBRU BULUT

TEZ DANIŞMANI  
DOÇ. DR. EMRE DANDIL

BİLECİK, 2025

10734240

## BEYAN

“CMAC Sinir Ağı Öğrenme Yöntemi Kullanılarak Yeni Melez GSA ve PSO Optimizasyon Algoritmalarının Geliştirilmesi ve Başarım Analizi ” adlı doktora tezinin hazırlık ve yazımı sırasında bilimsel araştırma ve etik kurallarına uyduğumu, başkalarının eserlerinden yararlandığım bölümlerde bilimsel kurallara uygun olarak atıfta bulunduğumu, kullandığım verilerde herhangi bir tahrifat yapmadığımı, tezin herhangi bir kısmının Bilecik Şeyh Edebali Üniversitesi veya başka bir üniversitede başka bir tez çalışması olarak sunulmadığını, aksinin tespit edileceği muhtemel durumlarda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

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

Nazmiye Ebru BULUT

05/08/2025

İmza

.....

## ÖN SÖZ

Bu tez sürecinde bilgi birikimiyle bana yardımcı olan değerli danışmanım Doç. Dr. Emre DANDIL'a, desteklerini hiçbir zaman esirgemeyen ve akademik hayatım boyunca yol göstericim Prof. Dr. Alpaslan DUYSAK'a, tezimin bilimsel olarak ilerlemesinde katkı sağlayan kıymetli hocam Prof. Dr. Uğur YÜZGEÇ'e, katkıları ve emeklerinden dolayı şükranlarımı sunarım. Ayrıca bilgi ve tecrübesi ile akademi camiasında yol arkadaşlığı yaptığım Öğr. Gör. Dr. Elif YILDIRIM'a teşekkürlerimi sunarım.

Son olarak eğitim-öğretim hayatım boyunca maddi, manevi beni destekleyen DUYSAK ailesinin değerli üyelerine ve her zaman yanımda olarak yükümü hafifleten, beni cesaretlendiren saygıdeğer eşim Ali BULUT'a sonsuz teşekkür ve saygılarımı sunarım.

**Nazmiye Ebru BULUT**

**2025**

## ÖZET

### CMAC SINIR AĞI ÖĞRENME YÖNTEMİ KULLANILARAK YENİ MELEZ GSA VE PSO OPTİMİZASYON ALGORİTMALARININ GELİŞTİRİLMESİ VE BAŞARIM ANALİZİ

Son zamanlarda karmaşık ve zorlu problemlerin çözümünde kullanılmak üzere geliştirilmiş metasezgisel algoritmaların sayısı oldukça fazladır. Bu algoritmaların birçoğu, doğal olayların, sistemlerin veya canlı türlerinin davranışlarından ilham alarak mühendislik problemleri için en iyi çözümü bulmayı amaçlamaktadır. Yeni algoritmaların bazıları en iyi çözüm arayışında, iki veya daha fazla algoritmanın veya metotların hibritleştirilmesi ile meydana gelmektedir. Bu tez çalışmasında, karmaşık ve zorlu mühendislik problemlerin çözümünde etkili ve verimli sonuçlar elde etmek için iki yeni algoritma geliştirilmiştir. Bunlardan birincisi, Newton'un evrensel çekim yasasına dayanarak geliştirilen Yerçekimsel Arama Algoritması (GSA) ile bir yapay sinir ağı yöntemi olan Serebellar Model Eklem/Artikülasyon Denetleyicisinin (CMAC) hibritleştirilmesine dayanan algoritmadır. İkincisi ise, kuş ve balık sürülerinin hayvan toplulukları içerisinde yiyecek bulma davranışlarından esinlenerek sunulmuş Parçacık Sürü Optimizasyonu (PSO) algoritması ile CMAC yapay sinir ağıyla hibritleştirilmesine dayanan algoritmadır. Geliştirilen GSA ve PSO algoritmaların varyantları sırasıyla CMACGSA ve CMACPSO olarak adlandırılmıştır. Önerilen CMACGSA algoritmasında GSA'daki parçacıkların / ajanların kütlelerini hesaplamak için farklı bir öğrenme prosedürü kullanılmaktadır. CMAC sinir ağı, GSA parçacıklarının / ajanlarının kütlelerini hesaplamak için klasik GSA'ya entegre edilmiştir. Klasik GSA'da ajanlar başlangıçta rastgele olarak konumlandırılırken CMACGSA'da CMAC'in başlangıçta giriş değerlerini kuantalama işlemi CMACGSA'nın başlangıç kriteri olarak uygulanmıştır. Ek olarak, CMACGSA algoritmasına literatürde var olan üç farklı mekanizma eklenerek iyileştirmeler yapılmıştır. Bu mekanizmalardan ilki, yerel minimuma takılmasını önlemek için popüler yöntemlerden biri olan Lévy mutasyonu, ikinci mekanizma ise, yineleme sırasında arama sınırlarının aşılmasını önlemek için sınır kontrol yöntemidir. Sonuncusu, yineleme sırasında belirli sayıda iyileşme olmazsa, ajanların konumlarını rastgele yeniden belirleyen hata (sayaç) işleme mekanizmasıdır. Geliştirilen diğer algoritma olan CMACPSO'da ise, klasik PSO algoritmasındaki hız güncelleme mekanizmasında bulunan iki rastgele değişken yerine, dinamik öğrenme yeteneğine sahip değişkenler kullanılmıştır. Ayrıca algoritmanın başlangıç adımında CMAC'in kuantalama işlemi uygulanmış ve sınır kontrol mekanizması eklenmiştir.

CMACGSA ve CMACPSO algoritmalarının performanslarını değerlendirmek ve klasik GSA ve PSO algoritmalarına göre üstünlüğünü incelemek için, standartlaştırılmış 2014 Evrimsel Hesaplama Kongresi tarafından sunulan CEC 2014 test paketi kullanılmıştır. Tezin deneysel analiz kısmında CEC 2014 test paketinden seçilen farklı türde toplam dört fonksiyon için iki boyutta (2D) sonuçlar elde edilmiştir. Ayrıca ek olarak, CEC 2014 test paketi için farklı boyutlarda (10D, 30D, 50D) da elde edilen sonuçlar değerlendirilmiştir. Elde edilen sonuçlar, klasik GSA ve PSO algoritmaları ile kıyaslanmıştır. Ayrıca, 30D’de elde edilen sonuçlar en güncel yöntemler (SOTA) ve önceki çalışmalarda bildirilen GSA ve PSO varyantları ile karşılaştırılmıştır. Karşılaştırma sonuçlarına göre, CMACGSA ve CMACPSO, geliştirilen öğrenme yeteneklerinden dolayı üstün performans sergilemişlerdir. Son olarak, CMACGSA ve CMACPSO’nun kapsamını ve gerçek dünya problemlerindeki uygulanabilirliğini incelemek amacıyla, bu algoritmalar farklı türde mühendislik problemleriyle test edilmiştir. Bu problemler; kaynaklı giriş problemi, basınçlı kap tasarımı ve üç çubuklu makas tasarımı problemleridir. Bu problemlerin çözümü, mevcut algoritmalar ile karşılaştırılmış ve sonuç olarak, CMACGSA ve CMACPSO algoritmalarının mevcut bazı algoritmaların gerisinde kaldığı, klasik GSA ve PSO algoritmalarına göre ise daha başarılı performans gösterdiği görülmüştür.

**Anahtar Kelimeler:** Sezgisel Optimizasyon, Metasezgisel Algoritmalar, CMAC, GSA, PSO, Mühendislik Problemleri

## ABSTRACT

### DEVELOPMENT AND PERFORMANCE ANALYSIS OF NOVEL HYBRID GSA AND PSO OPTIMIZATION ALGORITHMS USING CMAC NEURAL NETWORK LEARNING METHOD

Recently, a number number of metaheuristic algorithms have been developed to solve complex and challenging problems. Many of these algorithms aim to find the best solution for engineering problems inspired by the behavior of natural phenomena, systems or species. Some of these new algorithms are hybrids of two or more algorithms or methods used to find the best solution. This thesis presents two new algorithms developed to efficiently and effectively solve complex and challenging engineering problems. The first algorithm is based on the combination of the Gravitational Search Algorithm (GSA), which is based on Newton's law of universal gravitation, and the Cerebellar Model Articulation Controller (CMAC), which is an artificial neural network method. The second is based on the combination of the Particle Swarm Optimization (PSO) algorithm, which is inspired by the foraging behavior of flocks of birds and schools of fish, and the CMAC neural network. The variants of the GSA and PSO algorithms developed are named CMACGSA and CMACPSO, respectively. The CMACGSA algorithm uses a different learning procedure to calculate the masses of the GSA's particles/agents. The CMAC neural network is integrated into the classical GSA to calculate the masses of the GSA's particles/agents. In the classical GSA, the agents are randomly positioned initially, whereas in the CMACGSA, the initialization process of the CMAC, which quantizes the input values, is applied as the initial criterion. Additionally, the CMACGSA algorithm is improved by incorporating three mechanisms from the literature. The first is the Lévy mutation, a popular method for avoiding local minima. The second is the bounds-checking method, which prevents exceeding the search bounds during iteration. The third is an error (counter) handling mechanism that randomly re-determines the positions of the agents if a certain number of improvements do not occur during iteration. In another developed algorithm, CMACPSO, variables with dynamic learning capability are used instead of the two random variables in the rate update mechanism of the classical PSO algorithm. Additionally, CMAC's quantization process is applied in the initialization step of the algorithm, and a boundary control mechanism is added.

To evaluate the performance of the CMACGSA and CMACPSO algorithms and examine their superiority over the classical GSA and PSO algorithms, the CEC 2014 standardized

test suite, presented at the 2014 Congress on Evolutionary Computing, was used. The experimental analysis section of the thesis presents two-dimensional (2D) results for four different types of functions selected from the CEC 2014 test suite. Additionally, the results obtained in different dimensions (10D, 30D, and 50D) are evaluated. These results are then compared with those of the classical GSA and PSO algorithms. Additionally, the 30D results are compared with state-of-the-art (SOTA) methods and GSA and PSO variants reported in previous studies. The comparison results show that CMACGSA and CMACPSO have superior performance due to their improved learning capabilities. Finally, to examine the scope and applicability of CMACGSA and CMACPSO to real-world problems, these algorithms were tested on different types of engineering problems. These problems include the welded beam problem, pressure vessel design, and three-bar truss design. The solutions to these problems are compared with those of existing algorithms, and the results show that the CMACGSA and CMACPSO algorithms outperform some existing algorithms, as well as the classical GSA and PSO algorithms.

**Keywords:** Heuristic Optimization, Metaheuristic Algorithms, CMAC, GSA, PSO, Engineering Problems

## İÇİNDEKİLER

	Sayfa
ÖN SÖZ.....	i
ÖZET.....	ii
ABSTRACT .....	iv
İÇİNDEKİLER.....	vi
TABLolar LİSTESİ.....	viii
ŞEKİLLER LİSTESİ.....	x
KISALTMALAR VE SİMGELER LİSTESİ.....	xii
1. GİRİŞ.....	1
1.1. Tezin Amacı ve Hedefleri.....	5
1.2. Tez Çalışmasının Katkısı .....	6
1.3. Tezin Organizasyonu .....	6
2. LİTERATÜR TARAMASI.....	8
2.1. GSA ile Yapılan Çalışmalar .....	8
2.2. PSO ile Yapılan Çalışmalar .....	11
2.3. Serebellar Model Eklem Kontrolörü (CMAC) ile Yapılan Çalışmalar .....	14
3. MATERYAL ve YÖNTEM .....	17
3.1. Serebellar Model Eklem Kontrolörü (CMAC).....	17
3.2. Yerçekimsel Arama Algoritması (GSA).....	19
3.3. Parçacık Sürü Optimizasyonu (PSO) .....	22
3.4. İyileştirilmiş CMAC Tabanlı GSA (CMACGSA) .....	24
3.5. PSO Tabanlı CMAC Yapısına Entegre CMACPSO.....	30
4. DENEYSEL ÇALIŞMALAR .....	33
4.1. Yakınsama Analizi .....	34

<b>4.2. Arama Geçmişi Analizi .....</b>	<b>36</b>
<b>4.3. Kıyaslama Fonksiyonları .....</b>	<b>38</b>
<b>4.4. Karmaşıklık Analizi .....</b>	<b>38</b>
<b>4.5. CEC 2014 Karşılaştırma Sonuçları .....</b>	<b>40</b>
<b>4.5.1. CEC 2014 10D Boyutta Karşılaştırma Sonuçları .....</b>	<b>40</b>
<b>4.5.2. CEC 2014 30D Boyutta Karşılaştırma Sonuçları .....</b>	<b>44</b>
<b>4.5.3. CEC 2014 50D Boyutta Karşılaştırma Sonuçları .....</b>	<b>47</b>
<b>4.5.4. CMACGSA ve CMACPSO Algoritmalarının SOTA Algoritmaları İle Karşılaştırılması .....</b>	<b>51</b>
<b>4.5.5. CMACGSA'nın GSA Varyantları ile Karşılaştırılması .....</b>	<b>53</b>
<b>4.5.6. CMACPSO'nun PSO Varyantları İle Karşılaştırılması .....</b>	<b>55</b>
<b>4.6. Mühendislik Tasarım Problemleri.....</b>	<b>56</b>
<b>4.6.1. Kaynaklı Kiriş Tasarımı .....</b>	<b>57</b>
<b>4.6.2. Basınçlı Kap Tasarımı .....</b>	<b>62</b>
<b>4.6.3. Üç Çubuklu Makas Tasarımı.....</b>	<b>65</b>
<b>5. SONUÇLAR.....</b>	<b>70</b>
<b>KAYNAKÇA.....</b>	<b>72</b>
<b>EKLER .....</b>	<b>80</b>

## TABLolar LİSTESİ

	Sayfa
<b>Tablo 1.1.</b> Kategorilerine Göre Mevcut Metasezgisel Algorİtmalar .....	1
<b>Tablo 4.1.</b> CMACGSA ve GSA Algorİtmaları İÇin CEC 2014 10D İle Test SonuÇları...41	41
<b>Tablo 4.2.</b> CMACPSO ve PSO Algorİtmaları İÇin CEC 2014 10D İle Test SonuÇları ....42	42
<b>Tablo 4.3.</b> CMACPSO ve CMACGSA Algorİtmaları İÇin CEC 2014 10D İle Test SonuÇları.....43	43
<b>Tablo 4.4.</b> Algorİtmaların CEC 2014 10D Kıyaslama Özeti .....	44
<b>Tablo 4.5.</b> CMACGSA ve GSA Algorİtmaları İÇin CEC 2014 30D İle Test SonuÇları...44	44
<b>Tablo 4.6.</b> CMACPSO ve PSO Algorİtmaları İÇin CEC 2014 30D İle Test SonuÇları ....45	45
<b>Tablo 4.7.</b> CMACPSO ve CMACGSA Algorİtmaları İÇin CEC 2014 30D İle Test SonuÇları.....46	46
<b>Tablo 4.8.</b> CMACGSA ve CMACPSO Algorİtmaların CEC 2014 30D Kıyaslama Özeti .....	47
<b>Tablo 4.9.</b> CMACGSA ve GSA Algorİtmaları İÇin CEC 2014 50D İle Test SonuÇları...48	48
<b>Tablo 4.10.</b> CMACPSO ve PSO Algorİtmaları İÇin CEC 2014 50D İle Test SonuÇları ..49	49
<b>Tablo 4.11.</b> CMACPSO ve CMACGSA Algorİtmaları İÇin CEC 2014 50D İle Test SonuÇları.....50	50
<b>Tablo 4.12.</b> Algorİtmaların CEC 2014 50D Kıyaslama Özeti .....	51
<b>Tablo 4.13.</b> SOTA Algorİtmaların Parametreleri .....	51
<b>Tablo 4.14.</b> CMACGSA ve CMACPSO'nun diĐer Metasezgisel Algorİtmaların CEC 2014 30D İle Test SonuÇları.....52	52
<b>Tablo 4.15.</b> GSA'nın Varyantlarının Parametreleri.....53	53
<b>Tablo 4.16.</b> CMACGSA Ve GSA'nın Varyantlarının CEC 2014 30D İle Test SonuÇları .....	53
<b>Tablo 4.17.</b> PSO Algorİtmasının Varyantlarının Parametreleri.....55	55
<b>Tablo 4.18.</b> CMACPSO ve PSO'nun Varyantlarının CEC 2014 30D İle Test SonuÇları 55	55
<b>Tablo 4.19.</b> WBD Problemi İÇin Karşılařtırılan Popüler Metasezgisel Algorİtmaların Parametreleri.....57	57

<b>Tablo 4.20.</b> WBD Probleminin Parametreleri .....	57
<b>Tablo 4.21.</b> CMACGSA ve diğ er Metasezgisel Algoritmaların WBD Probleminin Sonuçları.....	59
<b>Tablo 4.22.</b> CMACPSO ve diğ er Metasezgisel Algoritmaların WBD Probleminin Sonuçları.....	60
<b>Tablo 4.23.</b> CMACGSA ve diğ er Metasezgisel Algoritmaların PVD Probleminin Sonuçları.....	63
<b>Tablo 4.24.</b> CMACPSO ve diğ er Metasezgisel Algoritmaların PVD Probleminin Sonuçları.....	64
<b>Tablo 4.25.</b> TBT Probleminin Parametreleri .....	66
<b>Tablo 4.26.</b> CMACGSA ve diğ er Metasezgisel Algoritmaların TBT Probleminin Sonuçları.....	67
<b>Tablo 4.27.</b> CMACPSO ve diğ er Metasezgisel Algoritmaların TBT Probleminin Sonuçları.....	68

## ŞEKİLLER LİSTESİ

	Sayfa
Şekil 3.1. Giriş Vektörü $x_i$ 'den Çıkış Vektörü $y_i$ kadar ki CMAC Eşleşmesi .....	18
Şekil 3.2. S Arama Uzayında Nesnelere Arasındaki Mesafe ve Yerçekimi Kuvveti .....	20
Şekil 3.3. Klasik GSA'nın Akış Şeması.....	22
Şekil 3.4. PSO Algoritmasının Akış Şeması .....	24
Şekil 3.5. CMACGSA Modelinin Mimarisi.....	25
Şekil 3.6. Giriş Sinyaline Rastgele Atılmış Ajanlar.....	27
Şekil 3.7. CMACGSA'da Kuantalama Seviyelerine göre Atanan Ajanlar .....	27
Şekil 3.8. CMACGSA Algoritmasının Akış Şeması.....	29
Şekil 4.1. Seçilen Dört Fonksiyonun 3 Boyutlu İz Düşümü (a) F1 Fonksiyonu, (b) F10 Fonksiyonu, (c) F15 Fonksiyonu, (d) F27 Fonksiyonu .....	34
Şekil 4.2. CMACGSA ve GSA Algoritmalarının Yakınsama Sonuçları .....	35
Şekil 4.3. CMACPSO ve PSO Algoritmalarının Yakınsama Sonuçları .....	36
Şekil 4.4. CMACGSA ve GSA'nın Arama Geçmişi Analizi.....	37
Şekil 4.5. CMACPSO ve PSO'nun Arama Geçmişi Analizi .....	38
Şekil 4.6. Kaynaklı Kiriş Tasarım Problemi .....	58
Şekil 4.7. Kaynaklı Kiriş Tasarım Probleminden Elde Edilen Sonuçlar (a) Klasik GSA ve CMACGSA'nın Yakınsama Eğrisi, (b) CMACGSA ve diğer Algoritmaların Yakınsama Eğrisi, (c) CMACGSA ve diğer Algoritmaların Kutu Grafiği .....	60
Şekil 4.8. Kaynaklı Kiriş Tasarım Probleminden Elde Edilen Sonuçlar (a) Klasik PSO ve CMACPSO'nun Yakınsama Eğrisi, (b) CMACPSO ve diğer Algoritmaların Yakınsama Eğrisi, (c) CMACPSO ve diğer Algoritmaların Kutu Grafiği.....	61
Şekil 4.9. Basınçlı Kap Tasarım Problemi .....	62
Şekil 4.10. Basınçlı Kap Tasarım Probleminden Elde Edilen Sonuçlar (a) Klasik GSA ve CMACGSA'nın Yakınsama Eğrisi, (b) CMACGSA ve diğer Algoritmaların Yakınsama Eğrisi, (c) CMACGSA ve diğer Algoritmaların Kutu Grafiği .....	64

<b>Şekil 4.11.</b> Basınçlı Kap Tasarımı Probleminden Elde Edilen Sonuçlar (a) Klasik PSO ve CMACPSO'nun Yakınsama Eğrisi, (b) CMACPSO ve diğer Algoritmaların Yakınsama Eğrisi, (c) CMACPSO ve diğer Algoritmaların Kutu Grafiği.....	65
<b>Şekil 4.12.</b> Üç Çubuklu Makas Tasarımı .....	66
<b>Şekil 4.13.</b> Üç Çubuklu Makas Tasarımı Probleminden Elde Edilen Sonuçlar (a) Klasik GSA ve CMACGSA'nın Yakınsama Eğrisi, (b) CMACGSA ve diğer Algoritmaların Yakınsama Eğrisi, (c) CMACGSA ve diğer Algoritmaların Kutu Grafiği .....	68
<b>Şekil 4.14.</b> Üç Çubuklu Makas Tasarımı Probleminden Elde Edilen Sonuçlar (a) Klasik PSO ve CMACPSO'nun Yakınsama Eğrisi, (b) CMACPSO ve diğer Algoritmaların Yakınsama Eğrisi, (c) CMACPSO ve diğer Algoritmaların Kutu Grafiği.....	69

## KISALTMALAR VE SİMGELER LİSTESİ

<b>ABC</b>	:Yapay Arı Kolonisi (Artificial Bee Colony)
<b>ACO</b>	:Karıncı Kolonisi Optimizasyonu (Ant Colony Optimization)
<b>ALO</b>	:Karıncı Aslanı Optimizasyonu (Ant Lion Optimization)
<b>BAT</b>	:Yarasa Algoritması (Bat Algorithm)
<b>BBBC</b>	:Büyük Patlama Büyük Çöküş (Big Bang Big Crunch)
<b>BBO</b>	:Biyocoğrafya Tabanlı Optimizasyon (Biogeography Based Optimization)
<b>BHO</b>	:Kara Delik Optimizasyonu (Black Hole Optimization)
<b>BSO</b>	:Beyin Fırtınası Optimizasyonu (Brain Storm Optimization)
<b>CA</b>	:Kültürel Algoritmalar (Cultural Algorithms)
<b>CEC2014</b>	:2014 IEEE Evrimsel Hesaplama Kongresi (2014 IEEE Congress on Evolutionary Computation)
<b>CEC2017</b>	:IEEE Evrimsel Hesaplama Kongresi 2017 (IEEE Congress on Evolutionary Computation 2017)
<b>CFO</b>	:Merkezi Kuvvet Optimizasyonu (Central Force Optimization)
<b>CGO</b>	:Kaos Oyunu Optimizasyonu (Chaos Game Optimization)
<b>CMAC</b>	:Serebellar Model Eklem/Artikülasyon Denetleyicisi (Cerebellar Model Articulation Controller)
<b>COA</b>	:Guguk Kuşu Arama (Cuckoo Search)
<b>CS</b>	:Coati Algoritması (Coati Optimization Algorithm)
<b>CSO</b>	:Çaprazlama Optimizasyon Algoritması (Crossover Optimization Algorithm)
<b>CSO</b>	:Kedi Sürü Optimizasyonu (Cat Swarm Optimization)
<b>CSS</b>	:Yüklü Sistem Araması (Charged System Search)
<b>DA</b>	:Yusufoçuk Algoritması (Dragonfly Algorithm)
<b>DE</b>	:Diferansiyel Evrim (Differential Evolution)
<b>EHO</b>	:Fil Sürü Optimizasyonu (Elephant Herding Optimization)

<b>EP</b>	:Evrimsel Programlama (Evolutionary Programming)
<b>FA</b>	:Ateş Böceği Algoritması (Firefly Algorithm)
<b>FPA</b>	:Çiçek Tozlaşma Algoritması (Flower Pollination Algorithm)
<b>GA</b>	:Genetik Algoritmalar (Genetic Algorithms)
<b>GSA</b>	:Yerçekimsel Arama Algoritması (Gravitational Search Algorithm)
<b>GSO</b>	:Galaktik Sürü Optimizasyonu (Galactic Swarm Optimization)
<b>GWO</b>	:Gri Kurt Optimizasyonu (Grey Wolf Optimization)
<b>HBO</b>	:İnsan Davranışına Dayalı Optimizasyon (Human Behavior-Based Optimization)
<b>HHO</b>	:Harris Şahinleri Optimizasyonu (Harris Hawk Optimization)
<b>HOA</b>	:Yürüyüş Optimizasyon Algoritması (Hiking Optimization Algorithm)
<b>HS</b>	:Uyum Arama (Harmony Search)
<b>MFO</b>	:Güve Alev Optimizasyonu (Moth-Flame Optimization)
<b>OIO</b>	:Optik Esinli Optimizasyon (Optics Inspired Optimization)
<b>PSO</b>	:Parçacık Sürüsü Optimizasyonu (Particle Swarm Optimization)
<b>PVD</b>	:Basıncılı Kap Tasarımı (Pressure Vessel Design)
<b>RO</b>	:Işın Optimizasyonu (Ray Optimization)
<b>SBA</b>	:Sosyal Tabanlı Algoritma (Social Based Algorithm)
<b>SCA</b>	:Sinüs Kosinüs Algoritması (Sine Cosine Algorithm)
<b>SFLA</b>	:Karıştırılmış Kurbağa Sıçrama Algoritması (Shuffled Frog-Leaping Algorithm)
<b>SKF</b>	:Simüle Edilmiş Kalman Filtresi
<b>SLO</b>	:Sosyal Öğrenme Optimizasyonu (Social Learning Optimization)
<b>SOSA</b>	:Güneş Sistemi Algoritması (Solar System Optimization Algorithm)
<b>SOTA</b>	:En İleri/Güncel Düzey (State of The Art)

<b>SSA</b>	:Salp Sürü Algoritması (Salp Swarm Algorithm)
<b>Std</b>	:Standart Sapma (Standart Deviation)
<b>TBT</b>	:Üç Çubuklu Makas Tasarımı (Three Bar Truss Design)
<b>TLBO</b>	:Öğretim-Öğrenme Tabanlı Optimizasyon (Teaching-Learning-Based Optimization)
<b>TOA</b>	:Takım Optimizasyon Algoritması (Team Optimization Algorithm)
<b>WBD</b>	:Kaynaklı Kiriş Tasarımı (Welded Beam Design)
<b>WOA</b>	:Balina Optimizasyon Algoritması (Whale Optimization Algorithm)
<b>WWO</b>	:Su Dalgası Optimizasyonu (Water Wave Optimization)
<b>YSA</b>	:Yapay Sinir Ağları (Artificial Neural Network)

## 1. GİRİŞ

Optimizasyon algoritmaları, bir problemde belirli parametreleri kullanarak en iyi sonucu elde etmeyi amaçlayan matematiksel yöntemlerdir. Literatürdeki mevcut yöntemler, karmaşık problemlerin çözümünde etkili sonuçlar sağlama yeteneğinden dolayı birçok çalışmaya konu olmuştur (Amiri vd., 2025). Literatürde geçerliliğini kanıtlamış farklı türde optimizasyon algoritmaları bulunmaktadır. Bu algoritmalar temel olarak doğa olaylarından, canlı davranışlarından, sistemlerden ve basit süreçlerden ilham alınarak geliştirilmiştir (Rodan vd., 2025). Mevcut algoritmalar; sürü zekası tabanlı, fizik / matematik tabanlı, evrimsel tabanlı ve insan davranışı tabanlı gibi farklı alt gruplara ayrılabilir. Sürü zekasına dayalı algoritmalar, hayvan topluluklarının (kuş, balık, karınca, arı vb.) sürü içerisindeki sosyal davranışlarını taklit ederek geliştirilmiş metasezgisel yöntemlerden biridir. Sürü zekası algoritmalarında bireyler arasında iş birliği ve bilgi aktarımı bulunmaktadır. Fizik / matematik tabanlı algoritmalarda, fiziksel yasalar (yer çekim kuvveti, elektromanyetik alan kuvveti), matematiksel kurallar veya sistemlerin davranışları model olarak geliştirilmiştir. Evrim temelli algoritmalar ise, biyolojideki mutasyon ve çaprazlama gibi doğal evrim kurallarını ilham almaktadır. Son olarak insan tabanlı algoritmalar ise, insanların grup halinde veya bireysel olarak düşünme, öğrenme ve karar verme süreçlerinden esinlenmektedir. Bu alt gruplarda bilinen başlıca metasezgisel algoritmalar Tablo 1.1’de özet olarak sunulmuştur.

**Tablo 1.1.**Kategorilerine Göre Mevcut Metasezgisel Algoritmalar

Kategori	Yıl	Algoritma	Açıklama
Sürü Zekası Tabanlı	1995	Parçacık Sürüsü Optimizasyonu (PSO)(Kennedy & Eberhart, 1995)	Kuşların ve balıkların sürü davranışları
	2006	Karınca Kolonisi Optimizasyonu (ACO)(Dorigo vd., 2007)	Karıncaların yiyecek arama davranışları
	2005	Yapay Arı Kolonisi (ABC)(Karaboga, 2005)	Arıların yiyecek bulma davranışı
	2014	Gri Kurt Optimizasyonu (GWO)(Mirjalili vd., 2014)	Gri kurtların beslenme davranışı
	2016	Balina Optimizasyon Algoritması (WOA)(Mirjalili & Lewis, 2016)	Kambur balinaların beslenme davranışı
	2015	Fil Güdüsü Optimizasyonu (EHO)(Wang vd., 2015)	Fil sürüsünün davranışı
	2019	Harris Şahinleri Optimizasyonu (HHO)(Heidari vd., 2019)	Şahinlerin saldırı stratejilerini modelleri
	2010	Yarasa Algoritması (BAT)(Yang, 2010)	Yarasaların avlanma davranışı
	2009	Guguk Kuşu Arama (COA)(Yang & Deb, 2009)	Guguk kuşlarının davranışları
	2009	Ateş Böceği Algoritması (FA)(Yang, 2009)	Okyanuslardaki ateş böceklerinin yanıp sönen ışığının davranışı
	2023	Coati Algoritması (COA)(Dehghani vd., 2023)	Koatilerin doğal davranışları
	2017	Salp Sürü Algoritması (SSA)(Mirjalili vd., 2017)	Salp zincirlerinin okyanustaki hareketi
	2008	Biyocoğrafya Tabanlı Optimizasyon (BBO) (Simon, 2008)	Doğal habitatların tür dağılımı ve göç süreçleri
	2016	Yusufçuk Algoritması (DA)(Mirjalili, 2016a)	yusufçukların avlanma ve sürü davranışlarını

Tablo 1.1.(devamı)

Kategori	Yıl	Algoritma	Açıklama
Fizik/Matematik Tabanlı	2006	Kedi Sürü Optimizasyonu (CSO) (Chu vd., 2006)	Kedilerin avlanma ve dinlenme davranışı
	2015	Karınca Aslanı Optimizasyonu (ALO) (Mirjalili, 2015a)	Karınca aslanı ile karıncalar arasındaki av-avcı ilişkisi
	2015	Güve Alev Optimizasyonu (MFO) (Mirjalili, 2015b)	Güvelerin ışık kaynaklarına yönelik davranışı
	2009	Yerçekimsel Arama Algoritması (GSA)(Rashedi vd., 2009)	Newton'un yerçekimi yasası
	2012	Çiçek Tozlaşma Algoritması (FPA) (Yang, 2012)	Çiçeksiz bitkilerin üreme davranışı
	2020	Güneş Sistemi Algoritması (SOSA)(Zitouni vd., 2020)	Gezegenlerin, uyduların ve asteroidlerin çekim kuvveti
	2013	Kara Delik Optimizasyonu (BHO)(Hatamlou, 2013)	Kara delik çekiminin etkisi
	2021	Kaos Oyunu Optimizasyonu (CGO)(Talatahari & Azizi, 2021)	Kaotik sistemleri
	2015	Optik Esinli Optimizasyon (OIO)(Kashan, 2015)	Işık davranışı ve optik yasaları
	2010	Yüklü Sistem Araması (CSS)(Ali Kaveh & Talatahari, 2010)	Coulomb yasası (elektriksel yükler)
	2012	Işın Optimizasyonu (RO)(A Kaveh & Khayatad, 2012)	Işık ışınlarının yayılımı (optik fiziği)
	2016	Sinüs Kosinüs Algoritması (SCA)(Mirjalili, 2016b)	Trigonometrik fonksiyonların etkisi
	2007	Merkezi Kuvvet Optimizasyonu (CFO)(Formato, 2007)	Kütlelerin merkezi çekim noktasına doğru fiziksel hareketi
	2015	Su Dalgası Optimizasyonu (WWO) (Zheng, 2015)	Suyun yapısı
Evrimsel Tabanlı	2006	Büyük Patlama Büyük Çöküş (BBBC)(Erol & Eksin, 2006)	Evrenin oluşumu
	2016	Galaktik sürü optimizasyonu GSO (Muthiah-Nakarajan & Noel, 2016)	Evrendeki galaksilerin yapısından ve hiyerarşik düzeni
	1992	Genetik Algoritmalar (GA)(Holland, 1992)	Charles Darwin'in evrim teorisi mutasyon ve seçimi modeli
	1999	Evrimsel Programlama (EP)(Yao vd., 1999)	Evrimsel süreci
Evrimsel Tabanlı	2011	Diferansiyel Evrim (DE)(Rocca vd., 2011)	Diferansiyel işlem ve adaptasyon süreci
	2014	Çaprazlama Optimizasyon Algoritması (CSO) (Meng vd., 2014)	Konfüçyüs'ün altın oran ve genetik algoritmasındaki çaprazlama operatöründen esinlenilmiştir.
İnsan Tabanlı	2011	Öğretim-Öğrenme Tabanlı Optimizasyon (TLBO)(Rao vd., 2011)	Öğretim ve öğrenme süreci
	2021	Takım Optimizasyon Algoritması (TOA)(Dehghani & Trojovský, 2021)	Ekip içindeki işbirliği ve sosyal etkileşim süreçleri
	2001	Uyum Arama (HS)(Geem vd., 2001)	Müzikteki doğaçlama süreci taklit eder
	2024	Yürüyüş Optimizasyon Algoritması (HOA)(Oladejo vd., 2024)	Yürüyüş ortamını taklit eder
	2013	Sosyal Tabanlı Algoritma (SBA)(Ramezani & Lotfi, 2013)	Bireylerin sosyal etkileşim, işbirliği süreci
	1994	Kültürel algoritmalar (CA)(Reynolds, 1994)	Bireylerin adaptasyon davranışı
	2011	Beyin Fırtınası Optimizasyonu (BSO) (Shi, 2011)	Bireylerin grup içinde fikir üretme süreci
	2016	Sosyal Öğrenme Optimizasyonu (SLO) (Liu vd., 2016)	İnsan ve hayvanların sosyal öğrenme davranışı
2017	İnsan Davranışına Dayalı Optimizasyon (HBO)(Ahmadi, 2017)	Bireylerin karar verme, öğrenme, taklit etme gibi davranışları	

Klasik optimizasyon algoritmalarına kıyasla metasezgisel algoritmalar, esnek yapıları ve problemlere uygulanabilir olması sayesinde birçok bilim insanının dikkatini çekmiştir. Metasezgisel algoritmaların temel amacı, verilen parametrelere göre problemlere etkili ve en iyi çözümü sunmasıdır. Genellikle optimizasyon algoritmalarında başlangıçta arama alanında rastgele parçacıklar / ajanlar konumlandırılmaktadır. Bu parçacıklar / ajanlar her yinelemede yer değiştirerek çözüm uzayını araştırır ve yineleme sonunda optimum sonuca ulaşması beklenir. Fakat algoritmaların en iyi çözümü arama sürecinde bazı zorluklarla karşılaşılabilir. Bunlar;

- Yerel optimuma takılma yani erken yakınsama olarak bilinen, arama sürecinin erken aşamalarından küresel çözüme ulaşmadan yerel optimumlara takılması (Rodan vd., 2025),
- Parametre seçim zorluğu,
- Çözüm uzayını tarama esnasında keşif kabiliyetinin düşük olması,
- Yüksek hesaplama karmaşıklığı,
- Optimizasyon algoritmaları bazı problemler için etkili iken bazıları için yetersizdir.

Karşılaşılan yukarıdaki zorluklar algoritmaların performanslarını olumsuz etkileyebilir. Algoritmaların performanslarını iyileştirmek ve etkili sonuçlar elde etmek için bilim insanları tarafından yeni yöntemler ve hibrit algoritmalar önerilmektedir. Bir algoritmanın zayıflıkları, başka bir algoritmanın güçlü yönleri ile hibritleştirilerek önerilen metasezgisel algoritmalar bulunmaktadır. Hibritleştirme sürecinde son zamanlarda hızla gelişen makine öğrenmesi yöntemleri de kullanılmaktadır. Makine öğrenimi algoritmaları karmaşık yapılar için kolayca giriş-çıkış ilişkilerini öğrenebilmekte ve problemlere çözüm üretebilmektedir. Makine öğrenimi algoritmaları ayrıca veri içindeki ilişkileri ve yapıları öğrenme / ayırt etme yeteneğine de sahiptir. Bu sayede kendilerine geniş bir yelpazede uygulama alanı bulmuşlardır.

Makine öğrenmesinin alt dalı olan yapay sinir ağları (YSA), karmaşık verileri işleme ve öğrenme sürecinde etkili bir yöntemdir. YSA, insan beyninin fonksiyonlarını taklit eden ve öğrenme yeteneğine sahip bir mekanizmadır. İnsan beyninin Serebellum (beyincik) isimli bölgesinden esinlenerek geliştirilen CMAC, 1975 yılında Albus tarafından önerilmiştir (Albus, 1975a). Klasik YSA'lardan farklı olarak CMAC giriş sinyallerini belirli bölümlere bölmekte ve her bir bölüme belirli bir ağırlık değeri atamaktadır. Böylece, klasik YSA modellerinden daha hızlı çalışmaktadır. Ayrıca,

CMAC'in kolay öğrenilebilen ve genelleştirme yapısı sayesinde, birbirine yakın ortak ağırlık değerine sahip ajanlar arasında bilgi aktarımı bulunmaktadır. CMAC'in sahip olduğu bu avantajlarına karşın eski bir YSA yapısı olmasından dolayı sınırlı sayıda çalışmaya konu olmuştur.

Bu tez çalışmasında, iki farklı metasezgisel algoritma incelenmiştir. Bunlarından ilki, Newton'un yerçekimi ve hareket yasasından ilham olarak sunulmuş fizik tabanlı GSA'dır. Diğer bir algoritma ise kuş ve balık sürülerinin yiyecek ararken sergilediği sosyal davranışlarından esinlenerek literatüre kazandırılan PSO'dur. Literatürde kendilerine yer bulmuş bu algoritmalarından PSO, hızlı yakınsama yeteneği; GSA ise arama uzayındaki etkili keşif kapasitesi nedeniyle, tez çalışmasında tercih edilmiştir. Tez kapsamında, CMAC sinir ağı modeli hibritleştirilerek; GSA'nın varyantı olan CMACGSA ve PSO'nun varyantı olan CMACPSO isimli iki yeni algoritma geliştirilmiştir.

GSA'nın optimizasyon problemlerine uygulanabilirliği birçok araştırmacının dikkatini çekmiştir. Bunun yanı sıra, GSA'nın yerel minimuma takılma riski, sınırlı keşif yeteneği, düşük yakınsama hızı ve hafızasının olmaması algoritmanın kısıtları arasındadır (Mittal vd., 2021). Bu kısıtları aşmak ve GSA'nın performansını iyileştirmek için CMAC yapay sinir ağının gelişmiş özellikleri GSA'ya entegre edilmiştir. GSA'da arama uzayındaki her bir parçacık ajan olarak kabul edilmektedir. Ajanlar, arama uzayına rastgele olarak konumlandırılır. Ajanlar arama uzayında bazı bölgelerde yoğun olarak konumlandırılırken, bazı bölgelerde ise seyrek olarak konumlandırılmaktadır. Bu durum, problemin çözümünü olumsuz yönde etkileyebilir. Bu sorunu aşmak için CMAC'in başlangıç koşulu olan giriş değerlerine kuantalama işlemi uygulanmıştır. Böylece, her bir ajan arama uzayına eşit olarak bölünmüştür. Önerilen başlangıç koşulu, tez çalışmasının özgün yönlerinden bir tanesidir. Diğer bir özgün yönü ise, CMAC'e ait ortak ağırlık değerleri sayesinde bilgi aktarımının olmasıdır. CMACGSA'nın gelişim aşamasında oluşan bazı kısıtları ortadan kaldırmak için üç farklı mekanizma eklenmiştir. Bunlardan ilki, yerel minimuma takılmasını önlemek için Lévy mutasyonudur ve bu mutasyonun amacı arama uzayında çeşitliliği arttırmaktır. Diğer bir mekanizma, algoritmanın yineleme sürecinde ajanların belirlenen yeni konumlarının arama sınırları içinde kalmalarını sağlamak için kullanılan sınır kontrol yöntemidir. Son olarak, hata işleme mekanizması sayesinde ajanların konumlarında belirli yineleme boyunca bir iyileşme olmazsa yeniden konumlandırılmasıdır. Böylece, CMACGSA'nın daha iyi sonuçlar vermesi beklenir.

PSO'nun basit ve uygulanabilir yapısı, düşük parametre sayısı, gelişmeye ve iyileştirilmeye açık yapısı sayesinde araştırma konusu olmuştur. Ancak yerel minimumda takılıp kalma, parametre seçiminde zorluk, hesaplama maliyeti ve başlangıç popülasyonun rastgele belirlenmesi gibi kısıtları bulunmaktadır. Bu tez çalışmasında, PSO'nun çeşitli kısıtlarının üstesinden gelmek ve performansını iyileştirmek amacıyla CMAC'in gelişmiş özellikleri PSO'ya entegre edilerek CMACPSO isimli yeni algoritma geliştirilmiştir. CMACPSO'da, PSO'nun hız güncelleme mekanizmasında kullanılan rastgele sayılar yerine, öğrenme yeteneğine sahip dinamik iki değişken kullanılmıştır. Böylece yineleme sürecinde CMAC'in ağırlıkları güncellenirken hız denklemindeki iki değişkene öğrenme yeteneği kazandırılmıştır. Klasik PSO algoritmasında, başlangıçta arama uzayına parçacıklar rastgele konumlandırılırken, CMACPSO'da CMAC'in başlangıç koşulu olan kuantalama işlemi uygulanmıştır. Bu işlem sayesinde, parçacıklar arama uzayına dengeli şekilde konumlandırılmaktadır. Ek olarak, CMACPSO'ya sınır kontrol mekanizması da eklenmiştir. Bu mekanizmanın amacı, parçacıkların yineleme sırasında sınır ihlali yapmasını engellemektir. Eğer parçacıklar arama sınırını geçerse, yeniden konumları hesaplanmaktadır.

Tez çalışmasının deneysel çalışmalar bölümünde, CMACGSA ve CMACPSO algoritmalarının performanslarını değerlendirmek için standartlaştırılmış ve farklı türde problemlerden oluşan CEC2014 test paketi kullanılmıştır. CMACGSA ve CMACPSO algoritmalarının kapsamı incelemek için CEC 2014 test paketinden farklı türde seçilen problemler ile 2 boyutta (2D) sonuçları detaylıca incelenmiştir. Ek olarak test paketi farklı boyutlarda (10D, 30D, 50D) çalıştırılarak sonuçlar elde edilmiştir. Elde edilen sonuçlara göre sırasıyla CMACGSA, klasik GSA ile CMACPSO, PSO ile kıyaslanmıştır. Ayrıca, tez çalışmasında, CEC2014 30D ile elde edilen sonuçlar hem literatürde var olan diğer metasezgisel algoritmalar ile hem de kendi varyantları ile kıyaslanmıştır. Son olarak, CMACGSA ve CMACPSO'nun günlük hayatta uygulanabilirliğini incelemek için popüler üç farklı mühendislik problemi ile test edilmiştir. Bu problemler sırasıyla; kaynaklı giriş tasarımı, basınçlı kap tasarımı ve üç çubuklu makas tasarımıdır.

### **1.1. Tezin Amacı ve Hedefleri**

Algoritmaların başka bir algoritma veya yöntemlerle hibritleştirilmesinin temel amacı mevcut algoritmanın kısıtlılıklarını başka bir algoritma ya da yöntemin güçlü yönleri ile gidermektedir. Bu tez çalışmasında, CMAC'in öğrenme, genelleştirme, karar verme gibi güçlü yönlerini GSA ve PSO algoritmalarına entegre ederek, optimizasyon problemlerine karşı hızlı, etkili ve güvenilir sonuçlar elde edilmesi amaçlanmıştır.

Mühendislik problemlerinde kullanılmak üzere, GSA ve PSO algoritmalarının CMAC yapay sinir ağı ile entegre iki yeni algoritma geliştirilmesi tezin temel hedeflerinden biridir. Bu kapsamda, GSA ve PSO algoritmalarının var olan kısıtlarının üstesinden gelmek için CMAC'in gelişmiş özellikleri ile entegre iki yeni algoritma geliştirilmesi hedeflenmiştir.

## **1.2. Tez Çalışmasının Katkısı**

Literatürde GSA ve PSO algoritmalarının performanslarını iyileştirmek için diğer algoritmalarla hibritleştirilmiş farklı çalışmalar mevcuttur. Bunun yanı sıra, GSA ve PSO'nun özellikle YSA'lar olmak üzere çeşitli yaklaşımlarla birleştirilmesi üzerine çalışmalar mevcut olmasına rağmen kısıtlıdır. Ayrıca CMAC'in, diğer algoritmalarla hibritleştirilmesi üzerine yapılan çalışmalar sınırlı sayıdadır. Bu tez çalışmasının özgün yönlerinden birisi, CMAC yapay sinir ağı ve GSA'nın birlikte kullanıldığı ilk çalışmalardan birisi olması ve bu kapsamda literatüre önemli bir katkısının olmasıdır. Ek olarak, literatürde optimizasyon problemlerine yönelik CMAC'ın PSO ile entegre edilmiş ve farklı mekanizmalar eklenmiş bir çalışma olmaması da tezin özgün diğer yönlerinden biridir. Ayrıca, tez çalışmasında sunulan CMAC sinir ağının ortaya koyduğu öğrenme modeli kullanılarak metasezgisel algoritmalarla ilk kez CMAC tabanlı bir öğrenme yeteneği kazandırılmıştır. Bu yenilikçi yaklaşım, tezin özgünlüğünü pekiştirmekte ve GSA ve PSO algoritmalarının uygulanabilirliğini ve performansını artırarak önemli bir iyileştirme sağlamaktadır.

## **1.3. Tezin Organizasyonu**

Tez çalışması 5 bölümden oluşmaktadır. Bunlar;

Bölüm 1'de mevcut olan metasezgisel algoritmaların geliştirilme aşamaları, bilim dünyasına olan katkıları, YSA'ların özellikleri, YSA ile metasezgisel algoritmalarının hibritleştirilmesinin gerekçeleri hakkında bilgi verilmiştir. Ayrıca tezin amacı, katkıları sunulmuştur.

Bölüm 2'de tez konusu hakkında detaylı literatür çalışması yer almaktadır. GSA ve PSO algoritmalarının performanslarını iyileştirmek için geliştirilen yöntemlerden bahsedilmiştir. GSA ve PSO algoritmalarının başka bir ya da daha fazla algoritma ile hibritleştirilmesi ile sunulan uygulamalar incelenmiştir. Ayrıca, GSA ve PSO'nun yapısında yer alan parametre değerlerine çeşitli yaklaşımlar ile sunulan çalışmalar ve GSA ve PSO'nun YSA ile hibritleştirilmesi hakkındaki çalışmalar anlatılmıştır. Son olarak, CMAC ile olan mevcut çalışmalar hakkında bilgi verilmiştir.

Bölüm 3'te GSA ve PSO algoritmalarının çalışma prensibi ve mekanizması detaylı olarak incelenmiştir. Bu bölümde CMAC'in yapısı, öğrenme süreci, hız ve genelleme yeteneği açıklanmıştır. Ayrıca, tezin ana çıktıları olan ve geliştirilen CMACGSA ve CMACPSO algoritmalarının çalışma mekanizmaları, tasarımları, uygulanabilirlikleri hakkında detaylı bilgiler verilmiştir.

Bölüm 4'te tez çalışması kapsamında yürütülen deneysel çalışmalar yer almaktadır. CMACGSA ve CMACPSO'nun performansları ve geçerliliklerini değerlendirmek için kullanılan CEC 2014 kıyaslama fonksiyonlarının özellikleri ile açıklanmıştır. CEC 2014 test paketinden seçilen 4 adet fonksiyon için detaylı analiz yapılmıştır. Ayrıca CEC 2014 test paketindeki fonksiyonlardan farklı boyutlarda (10D, 30D, 50D) sonuçlar elde edilmiştir. 30D elde edilen sonuçlar, SOTA algoritmaları ve hem GSA'nın hem de PSO'nun diğer varyantları ile karşılaştırılmıştır. CMACPSO ve CMACGSA'nın günlük hayatta uygulanabilirliğini değerlendirmek için mühendislik tasarım problemlerinden olan kaynaklı kirişi tasarımı, basınçlı kap tasarımı ve üç çubuklu makas tasarımı problemleri kullanılmıştır.

Bölüm 5'de tez kapsamında elde edilen sonuçlar yer almaktadır. Bu bölümde CMACGSA ve CMACPSO algoritmalarının genel değerlendirmesi, literatüre ve gelecekte yapılacak olan çalışmalara katkısı vurgulanmıştır.

## 2. LİTERATÜR TARAMASI

Bu bölümde, tez çalışmasında ele alınan GSA, PSO ve CMAC sinir ağı ile yapılan güncel çalışmalar hakkında bilgi verilmiştir. Bölüm üç alt başlıktan oluşmaktadır. Bunlar GSA ile yapılan çalışmalar, PSO ile yapılan çalışmalar ve CMAC ile yapılan çalışmalardır.

### 2.1. GSA ile Yapılan Çalışmalar

GSA algoritmasının literatüre kazandırılmasından bu yana araştırmacılar tarafından performansını iyileştirmek için farklı yöntemler geliştirilmiştir. Bu yöntemlerden biri, GSA'nın yapısında ve işleme mekanizmasında çeşitli değişiklikler yapılmasıdır. Çalışmalardan bazıları; Rashedi ve Saryazdi tarafından sunulan (Rashedi vd., 2010) ikili (binary) kodlamalı GSA (BGSA), Saeidi-Khabisi ve tarafından önerilen (Saeidi-Khabisi & Rashedi, 2012) bulanık denetleyicilerle genişletilmiş GSA (FGSA), Precup ve arkadaşlarının çalışması (Precup vd., 2012) uyarlamalı GSA (GGSA) ve Sheikhan ve Jadid tarafından önerilen (Sheikhan & Jadidi, 2014) modifiye edilmiş GSA (MGSA) olarak sıralanabilir. Bu kapsamda önerilen diğer çalışmalar aşağıda kısaca özetlenmiştir.

Mirjalili ve Hashim tarafından (Mirjalili & Hashim, 2010) PSO'nun sömürü özelliğini GSA'nın keşif özelliği ile birleştirerek PSO-GSA isimli algoritma önerilmiştir. Geliştirilen algoritma 23 adet karşılaştırma fonksiyonları ile test edilmiştir. PSO-GSA algoritmasının standart PSO ve GSA göre daha hızlı optimuma ulaştığı gözlemlenmiştir.

Mirjalili ve Lewis (Mirjalili & Lewis, 2014) tarafından GSA'nın yineleme sırasında yerel minimumlardan kaçınmak ve sömürü aşamalarını hızlandırmak için en iyi kütle (gbest) arşivleyerek, gbest kılavuzlu GGSA önerilmiştir. GGSA'yı CEC 2005 ve iki farklı mühendislik problemi ile test etmişlerdir.

Geliştirilmiş GSA varyantlarından genellikle yerçekimi sabiti, kbest, hız ve konum gibi parametrelerde değişikliklerle iyileştirmeler yapılmıştır. Mittal ve arkadaşları tarafından (Mittal vd., 2016) kaotik Kbest'i (en iyi uygunluk sayısı) kullanan yeni CKGSA önerilmiştir.

Pelusi ve arkadaşları tarafından (Pelusi vd., 2020) yerçekimi sabiti ve ivme katsayısı hiperbolik bir fonksiyona yerleştirilerek, istenmeyen erken yakınsamayı önlemek için Hiperbolik Yerçekimsel Arama Algoritması (HGSA) önerilmiştir. Önerilen algoritma CEC 2019 test fonksiyonları ve üç çubuklu kafes mühendislik problemi ile test edilmiştir.

Motameni ve Nematzadeh (Alirezanejad vd., 2021) tarafından öğrenme yeteneğine sahip yerçekimi sabiti olan GSA-LA sunulmuştur. Test fonksiyonları ile sayısal analizleri yapılmış ve evrim tabanlı algoritmalar ile karşılaştırılmıştır. Sonuç olarak, mevcut algoritmalara göre iyi performans gösterdiğini belirtmişlerdir.

Guo ve arkadaşları tarafından (Guo vd., 2023) GSA'nın keşif ve sömürü yeteneği artırmak için uyarlanabilir konum kılavuzlu yerçekimi arama algoritması disGSA önerilmiştir. Önerilen algoritmanın performansı CEC2017 kıyaslama fonksiyonları ve GSA'nın varyantları ile test edilmiştir.

GSA'nın performansını iyileştirmek ve etkili çözümler elde etmek için yapılan iyileştirmelerden bir diğeri, mevcut algoritmalar veya YSA ile hibritleştirilmesidir. Geliştirilmiş / iyileştirilmiş ya da hibrit GSA algoritmaları birçok mühendislik problemi ve uygulama alanlarında kullanılmıştır. Armaghani ve Mohamad tarafından sunulan (Harandizadeh vd., 2020) kayanın çekme dayanımının tahmini, Mosa'nın önerdiği veri madenciliği (Mosa, 2019), Singh ve Deep çalışmalarında (Singh & Deep, 2017) yapı mühendisliği, Bahrololoum ve Saeed çalışmasında (Bahrololoum vd., 2012) sınıflandırma, Özyön ve Yaşar tarafından (Özyön & Yaşar, 2018) elektrik enerjisi üretim sistemleri uygulama alanlarında geliştirilmiş GSA'lar önerilmiştir.

Naik ve Panda tarafından (Naik & Panda, 2015) CS'nin sömürü yeteneği ile GSA'nın keşif yeteneğinin birleştirilmesi ile CS-GSA algoritması önerilmiştir. CS-GSA'nın performansı benchmark test fonksiyonları ile test edilmiş ve elde edilen sonuçlar klasik CS ve GSA ile karşılaştırılmıştır. Sonuç olarak, CS-GSA'nın klasik algoritmalara göre daha iyi sonuçlar verdiği gözlemlenmiştir.

Bir diğer çalışmada, Ahmadi (Ahmadi, 2017) tarafından sensör düğümlerinin verimini artırmak için Benzetimli Tavlama Algoritması (SA) ile GSA hibritleştirilerek GSA-SA önerilmiştir

Muhammad ve arkadaşları tarafından (Muhammad vd., 2017) simüle edilmiş kalman filtresi SKF ile melezleştirilmiş SKF-GSA algoritma önerilmiştir. Algoritmalar CEC 2014 test paketi ile değerlendirilmiştir. Sonuç olarak, SKF, GSA ve SKF-GSA algoritmaları bu test fonksiyonları yardımıyla karşılaştırılmıştır.

Bohat ve Arya (Bohat & Arya, 2018) tarafından, GSA'nın ajanlar arasındaki sosyal ilişkisi sayesinde Rehberli Yerçekimsel Arama Algoritması (GG-GSA) önerilmiştir. Algoritmada GSA'nın kuvvet hesaplama işlemi sırasındaki küresel en iyi çözümün (gbest) rehberlik yapması ile keşif ve sömürü dengesi korunmuştur. Önerilen

algoritmanın performansı 23 popüler test fonksiyonu ve CEC 2014 kıyaslama paketi ile değerlendirilmiştir.

Zhao ve arkadaşları (Zhao vd., 2018) tarafından GSA'nın yakınsama hızını arttırmak ve keşif ile sömürü arasındaki dengeyi sağlamak için GSA ve DE'ye dayalı SGSADDE isimli yeni algoritma önerilmiştir. Çalışmada, DE'nin çaprazlama ve mutasyon operatörlerinden yararlanılmıştır. Bu algoritma CEC 2017 kıyaslama fonksiyonları ile incelenmiş ve elde edilen sonuçlar GSA'nın varyantları ile karşılaştırılmıştır. Hibrit algoritmanın iyi sonuçlar verdiği raporlanmıştır.

Sattar ve Cheetar (Sattar & Cheetar, 2019) tarafından, tek amaçlı optimizasyon problemini çözmek için BAT ve GSA'nın hibritleştirilmesi ile elde edilen GSABAT algoritması önerilmiştir.

Garg tarafından (Garg, 2019) karma değişkenli kısıtlı doğrusal olmayan optimizasyon problemleri için GA'nın seçim, çaprazlama ve mutasyon operatörleri GSA'ya entegre ederek GSA-GA isimli algoritma önerilmiştir. Önerilen algoritmanın performansını doğrulamak için çalışma, dokuz mühendislik tasarım probleminden elde edilen verileri kullanarak CS, PSO, ABC, gibi mevcut algoritmalarla karşılaştırılmıştır. Sonuçlar, önerilen GSA-GA'nın performansının diğer algoritmalarla daha iyi olduğunu göstermiştir.

Özyön ve arkadaşları tarafından (Özyön vd., 2019) GSA ve Artımlı Sosyal Öğrenme (ISL) algoritmasının birleşimine dayanan artımlı GSA (IGSA) önerilmiştir. Bu algoritmada, ilk popülasyondan başlayarak belirli adımlarda GSA'ya yeni ajanlar eklenir ve ajan sayısı maksimum olana kadar süreç devam eder. Popülasyona ajan ekleme sürecinde üç farklı yaklaşım ekleyerek, üç IGSA varyantı önermişlerdir. Önerilen algoritmalar kıyaslama fonksiyonları kullanılarak klasik GSA ile karşılaştırılmış ve IGSA'ların klasik GSA'dan daha iyi sonuçlar verdiği gözlemlenmiştir.

Jiang ve Li tarafından (Jiang, Jiang, vd., 2020) GSA'nın kaotik haritalarda keşif yeteneğini geliştirmek için önerilen Kaotik Yerçekimi Arama Algoritması (CGSA) ve SCA hibritleştirilerek, Sinüs Kaotik Yerçekimi Arama Algoritması (SCGSA) önerilmiştir. CGSA yerel optimum takılmasını önlemek için sinüs fonksiyonlarından yararlanılmıştır. SCGSA ve CGSA, CEC 2014 test fonksiyonları ve bazı mühendislik problemleri ile karşılaştırılmıştır. Önerilen SCGSA'nın iyi sonuçlar verdiği gözlemlenmiştir.

Bir başka çalışmada, Jiang ve arkadaşları (Jiang, Yang, vd., 2020) tarafından Kaotik Yerçekimi Arama Algoritmasına (CGSA) sinüs fonksiyonu ile rastgelelik ve denge mekanizması eklenerek, denge ayarlama tabanlı kaotik yerçekimi arama algoritması BA-GSA önerilmiştir. Algoritmanın keşif yeteneğini değerlendirmek için CEC 2014 kıyaslama fonksiyonları ve uygulanabilirliğini incelemek için mühendislik tasarım problemlerini kullanmışlardır. BA-CGSA hem CGSA hem de sota algoritmaları ile kıyaslanmıştır. BA-CGSA çoğu karşılaştırımda iyi performans göstermesine rağmen, geliştirilmeye ihtiyaç olduğu değerlendirilmiştir.

Shehadeh tarafından (Shehadeh, 2021) Sperm Sürüsü Optimizasyonu (SSO) ve GSA'nın birleştirilmesi ile HSSOGSA algoritması önerilmiştir. Algoritmada SSO'nun sömürü yeteneği ile GSA'nın keşif yeteneğinin üstünlüğünden yararlanılmıştır. Önerilen HSSOGSA'nın performansı CEC 2017 test fonksiyonları ile hem klasik GSA hem de klasik SSO ile karşılaştırılmış ve HSSOGSA'nın daha iyi sonuçlar verdiği doğrulanmıştır.

Neamah ve arkadaşları (Neamah vd., 2022) tarafından GSA'nın erken yakınsama sorunun üstesinden gelmek için Diferansiyel Evrim (DE) algoritmasının avantajlı yönlerini kullanarak geliştirilmiş Yerçekimsel Evrim Algoritması (GEA) önerilmiştir. Önerilen algoritma CEC 2014 test fonksiyonları ile test edilmiştir. Elde edilen sonuçlar klasik GSA, DE, WOA ile kıyaslanmıştır. GEA'nın sömürü hızının diğer algoritmalara göre hızlı olduğu görülmüştür.

## **2.2. PSO ile Yapılan Çalışmalar**

PSO algoritması basit yapısı, az parametre sayısı sayesinde araştırmacıların dikkatini çekmiş ve çok fazla çalışmaya konu olmuştur. Araştırmacılar PSO'nun erken yakınsama kısıtını çözmek için ciddi çaba sarf etmişlerdir. Çalışmalarda PSO'nun performansını daha verimli hale getirmek için çeşitli PSO varyantları sunulmuştur. PSO'nun parametrelerinde yapısal değişiklikler yapılarak önerilen çalışmalar bulunmaktadır. Ayrıca, literatürde başka algoritmaların güçlü yönleri ile hibritleştirilmesine dayanan çalışmalar da yer almaktadır (Shami vd., 2022). Tezin bu kısmında PSO ile yapılan çalışmalar aşağıda incelenmiştir.

Şenel ve arkadaşları tarafından (Şenel vd., 2019) PSO'nun yerel minimuma düşmesini önlemek için GWO'nun keşif yeteneğini kullanan PSO-GWO önerilmiştir. Önerilen algoritma 5 kıyaslama fonksiyonu ve 3 gerçek dünya problemleri ile test edilmiş ve çeşitli yöntemlerle karşılaştırılmıştır.

Garg (Garg, 2016) tarafından kısıtlı optimizasyon problemlerini çözmek için PSO'nun keşif ve sömürü dengesine GA operatörlerini ekleyerek PSO-GA algoritması sunulmuştur. Önerilen PSO-GA'nın performansı mühendislik tasarım problemleri ile test edilmiştir.

Chen ve arkadaşları tarafından (J.-F. Chen vd., 2015) PSO'nun yakınsama yeteneği ile CS'nin keşif özelliği kullanılarak, PSOCS algoritması sunulmuştur. Çalışmada, FNN'nin öğrenme özelliğini PSOCS'ye eklemişlerdir. PSOCS-FNN, PSO-FNN ve CS-FNN, iki mühendislik problemi ile karşılaştırılmıştır.

Al Bataineh ve Manacek tarafından (Al Bataineh & Manacek, 2022) kalp hastalıklarının teşhisi için eğitilmiş çok katmanlı bir algılayıcı MLP-PSO önerilmiştir. MLP-PSO'nun diğer algoritmalara kıyasla yüksek bir doğruluk oranına sahip olduğu tespit edilmiştir.

Gou ve arkadaşları tarafından (Gou vd., 2017) PSO'nun parçacıkları duygusal durumlarındaki farklılıklarına göre arama alanında farklı gruplara ayrılarak çeşitlendirilmiştir. Parçacıkların bireysel farklılıklarına dayalı PSO ile DE'nin hibritleştirilmesiyle yeni IDE-PSO sunulmuştur. Algoritmanın performansı CEC 2013 test paketi ile kıyaslanmıştır.

Sedighizadeh ve arkadaşları tarafından (Sedighizadeh vd., 2021) PSO'nun hız güncelleme mekanizmasına iki yeni terim eklenerek Genelleştirilmiş Parçacık Sürü Optimizasyonu (GEPSO) sunulmuştur. GEPSO'nun performansını değerlendirmek için 16 adet standart kıyaslama fonksiyonu kullanılmıştır.

Zhang ve arkadaşları tarafından (Zhang vd., 2007) ileri beslemeli sinir ağlarının ağırlıklarını eğitmek için PSO'nun küresel arama yeteneği ve geri yayılım algoritmasının yerel arama yeteneğinin hibritleştirilmesiyle PSO-BP sunulmuştur. Sonuç olarak, önerilen algoritmanın karmaşık problemlerin çözümünde makul olduğunu tespit etmişlerdir.

Cuong-Le ve arkadaşları tarafından (Cuong-Le vd., 2022) yapısal sağlık izleme SHM ve Tahribatsız Hasar Tespiti yapılarını güvence altına almak için PSO ve destek vektör makinesinin (SVM) birleşimi olan PSO-SVM önerilmiştir. SVM gereksiz giriş parametrelerini ortadan kaldırmak ve PSO'nun arama yeteneği sayesinde hasarlı bölgeleri sınıflandırmak için kullanılmıştır. PSO-SVM ve diğer makine öğrenmesi modellerine kıyasla etkili sonuçlar sağlamıştır.

A Kaveh ve arkadaşları tarafından (A Kaveh vd., 2014) Kırlangıç Sürü Optimizasyonu (SSO) algoritmasının arama yeteneğini geliştirmek için Hibrit Parçacık Kırlangıç Sürü Optimizasyonu (HPSSO) hibrit algoritması önerilmiştir. Çalışmada, PSO'nun her popülasyon alt kolonilere bölüldüğünde, her alt koloninin iyi parçacıkların deneyimlerinden yararlanması hedeflenmiştir. Geliştirilen algoritma farklı gruplardan oluşan matematiksel optimizasyon problemlerinde test edilmiştir. Sayısal sonuçlara göre, HPSSO algoritması, PSO ve PSO'nun varyantlarına kıyasla daha başarılı sonuçlar sergilemiştir.

K. Chen ve arkadaşları tarafından (K. Chen vd., 2018) PSO'nun arama sürecindeki çeşitliliğinin az olması ve küresel arama yeteneği ile yerel arama yeteneği arasındaki dengeyi sağlamakta yaşanan zorlukların üstesinden gelmek için doğrusal olmayan dinamik ivme katsayıları, lojistik haritalama ile yeni parçacık konum güncellenmesi eklenerek geliştirilmiş A-PSO algoritması önerilmiştir. Yeni algoritmanın çözüm kalitesini test etmek için 20 standart optimizasyon problemi kullanılmıştır.

W.-N. Chen ve arkadaşları tarafından (W.-N. Chen vd., 2012) PSO'nun hızlı yakınsama özelliğini kaybetmeden erken yakınsama zorluğunu aşmak için doğadaki yaşlanma ve lider mekanizmasından esinlenerek geliştirilmiş ALC-PSO algoritması önerilmiştir. ALC-PSO algoritmasının etkinliğini değerlendirmek için 17 adet test fonksiyonu kullanılmıştır. Elde edilen test sonuçları PSO'nun varyantları ile kıyaslanmıştır.

Sun ve arkadaşları tarafından (Sun vd., 2010) kör kaynak ayırma (BSS) problemini çözmek için PSO algoritmasının bir varyantı olan küme kılavuzuna sahip CGPSO algoritması önerilmiştir. Önerilen algoritma CGPSO'nun hem iyi hem de kötü koşullu karışım matrislerinde yüksek performans gösterdiği tespit edilmiştir.

Mendes ve arkadaşları tarafından (Mendes vd., 2004) PSO'nun sosyal bileşen özelliğine farklı yaklaşımda bulunan, iki algoritma karşılaştırılmıştır. Bu algoritmalar Kanonik PSO ile Tam bilgili Parçacık Sürü Optimizasyonu (FIPS) olarak belirlenmiştir. En iyi çözüme ulaşma sürecinde Kanonik PSO algoritması sadece en iyi komşulardan etkilenirken, FIPS'te her komşu eşit katkı sağlamıştır. Karşılaştırma sonuçlarına göre FIPS'in daha esnek ve etkili sonuçlar sergilediği görülmüştür.

Peram ve Mohan tarafından (Peram vd., 2003) önerilen FDRPSO algoritması klasik PSO'nun erken yakınsama problemini çözmek için parçacıkların yakınındaki en iyi

uygunluk deęerine sahip paracıklara doęru hareket etme yaklařımına dayanmaktadır. Karmařık ve ok modlu problemlerin özümünde PSOFDR etkili özümler saęlamıřtır.

### **2.3. Serebellar Model Eklem Kontrolörü (CMAC) ile Yapılan alıřmalar**

Literatürde optimizasyon algoritmaları ve YSA'nın hibritleřtirilmesi ile yapılan ok az sayıda alıřma bulunmaktadır. Genellikle YSA'ların eęitiminde parametrelerin iyileřtirilmesi için optimizasyon algoritmaları kullanılmaktadır. Mevcut alıřmalarda geliřtirilen modeller bazı mühendislik problemlerinde kullanılmıřtır (Özden & İřeri, 2023) (Bas vd., 2021; M.-L. Huang & Chou, 2019; Nezamoddini & Gholami, 2019). Arařtırmacılar tarafından CMAC'in giriş / ıkıř yapısında, öęrenme veya depolama modelinde, öęrenme algoritmasının yapısında, aęın parametre seiminde deęiřiklikler ve iyileřtirmeler yapılarak sunulmuř alıřmalar bulunmaktadır (Li vd., 2022).

Chao ve arkadařları tarafından (Chao vd., 2019) bulanık mantıkla birleřtirilmiř FCMAC, bir dięer alıřmada ise Tian ve arkadařları tarafından (Tian vd., 2013) alıřmalarında CMAC'in bellek sorununu özmek için KCMAC önerilmiřtir. Tao ve arkadařları (Tao vd., 2003) alıřmalarında depolama modelinde deęiřiklikler yaparak, görüntü verisi sıkıřtırmak için bellek sınırlı CA-CMAC önermiřtir. Cotter ve Guillerm (Cotter & Guillerm, 1992) alıřmalarında öęrenme parametresinin seimini incelemiřlerdir. Lane ve arkadařları (Lane vd., 1992) alıřmalarında B-spline fonksiyonlarını kullanan daha yüksek dereceli bir CMAC önermiřlerdir.

CMAC klasik YSA'lardan farklı olarak hızlı öęrenme, iyi genelleme, bilgi depolama gibi yeteneklere sahiptir (Hsu vd., 2009). Ayrıca klasik YSA'lardan farklı olarak giriş sinyallerini belirli bölümlere ayırılır ve her bir bölüme aęırlık deęerleri atar. Bu özellięi sayesinde klasik yapay sinir aęlarından daha hızlı alıřır. Bu özellięinden dolayı CMAC, genellikle gerek zamanlı kontrol sistemleri gibi uygulama alanlarında kullanılmaktadır (Li vd., 2022). CMAC'in uygulama alanları, Kim ve Lewis (Kim & Lewis, 2000) tarafından önerilen alıřmada kontrol sistemleri ve robotik; Kolcz ve Allinson (Kolcz & Allinson, 1994) alıřmalarında sinyal iřleme; Shiraishi ve arkadařlarının (Shiraishi vd., 1995) alıřmalarında yakıt enjeksiyon sistemleri; Tsai ve Yeh tarafından (Tsai & Yeh, 2009) yapılan alıřmada endüksiyon motor sürücülerinin kontrolü; Palacios ve arkadařları tarafından (Palacios vd., 2006) yapılan alıřmada veri madencilięi ve Bucak ve Baki (Bucak & Baki, 2010) tarafından gerekleřtirilen alıřmada karacięer hastalıęı teřhis sistemleridir.

Qin ve arkadaşları tarafından (Qin vd., 2004) CMAC'in ağırlıkları güncellemesinde kullanılması için Yinelemeli En Küçük Kareler (RLS) algoritması tabanlı CMAC-RLS modeli sunulmuştur. CMAC-RLS'nin performansı değerlendirmek için doğrusal olmayan kontrol sistemleri simüle edilmiştir.

Teddy ve arkadaşları tarafından (Teddy vd., 2007) CMAC'in bellek alanının daha verimli kullanılması için hiyerarşik kümelenmiş adaptif kuantalama yapan HCAQ-CMAC modeli sunulmuştur.

Jan ve Hung tarafından (Jan & Hung, 2001) tek boyutlu CMAC'e ağaç yapısı eklenerek MS\_CMAC modeli geliştirilmiştir. Ek olarak ikinci dereceden spline şeması kullanarak, yüksek mertebeli HMS\_CMAC modeli önerilmiştir. Bu model ağırlıklar güncellendikten sonra doğruluğunu kontrol etmektedir ve iyi bir genelleme yeteneği sunmaktadır.

Kong ve arkadaşları (Kong vd., 2020) tarafından CMAC ve GA algoritmaları kullanılarak, kükürt giderme ve ekonomik maliyet oranı için iki girdili tahmin modeli önerilmiştir.

Lee ve arkadaşları (Lee vd., 2004) tarafından GA tabanlı sağlam öğrenme kredi atama modeli olan GCA-CMAC önerilmiştir. Önerilen algoritma bazı matematik fonksiyonları ile test edilmiş ve klasik CMAC, CA-CMAC, bulanık CMAC (FCMAC) ve GCA-CMAC yöntemleri ile karşılaştırılmıştır. Sonuç olarak önerilen algoritmanın diğer CMAC'lere göre daha iyi sonuç verdiği tespit edilmiştir.

Tu ve Cao tarafından (Tu & Cao, 2012) kalıcı mıknatıslı senkron motor için PSO ile CMAC'in hibritleştirilmesi ile PSO-CMAC önerilmiştir. Önerilen algoritma PSO'nun bireyler arasındaki rastgele öğrenmesi simüle edilirken, CMAC bir bireyin kendi kendine öğrenmesini simüle edilmiştir. PSO-CMAC'in performansını değerlendirmek için bazı deneyler yürütülmüştür.

Moosavi ve arkadaşları (Moosavi vd., 2023) tarafından yüksek çözünürlüklü toprak nem haritalaması için optimize edilmiş PSO-CMAC ve PSO-GMDH yöntemleri geliştirilmiştir. Çalışma sonuçlarına göre, SM tahmininde hem küresel hem de yerel yaklaşımları PSO-CMAC'in PSO-GMDH'den daha iyi performans gösterdiğini tespit etmişlerdir.

Özetle literatürde, CMAC'in güçlü yapısını kullanarak metasezgisel algoritmalarla hibritleştirilmesi dayanan çalışmalar oldukça sınırlıdır. Bildiğimiz kadarıyla, CMAC'in GSA ile birlikte kullanıldığı ilk çalışma, bu tez çalışması kapsamında geliştirilen

CMACGSA algoritmasıdır ve literatüre katkısı gelişmiş özelliklerinden dolayı oldukça yüksektir. CMAC yapay sinir ağı ve PSO'nun birlikte olduğu mevcut çalışmalar bulunmaktadır. Fakat optimizasyon problemlerinde kullanılmak üzere geliştirilmiş dinamik öğrenme yeteneğine sahip, tez çalışmasında geliştirilen CMACPSO algoritması benzeri bir çalışma da literatürde bulunmamaktadır. Dolayısıyla, CMAC yapısının öğrenme modelinin, metasezgisel algoritmalara entegre edilmesi tez çalışmasının yenilikçi yönünü göstermektedir. Sonuç olarak, çalışmanın özgünlüğü pekiştirilmiş ve GSA'nın performansının iyileştirilmesi sağlanmıştır (Bulut vd., 2025).

### 3. MATERYAL ve YÖNTEM

Tez çalışmasının bu bölümünde tezin ana konularından biri olan CMAC'ın çalışma prensibi, yapısı ve özellikleri hakkında bilgi verilmiştir. Daha sonrasında sırasıyla GSA ve PSO algoritmalarının çalışma mekanizmaları, parametreleri ve matematiksel ifadeleri açıklanmıştır. Tezin çıktıları olan iyileştirilmiş CMAC tabanlı GSA (CMACGSA) ile CMAC'ın öğrenme yeteneğine sahip PSO tabanlı CMACPSO algoritmaları detaylı olarak sunulmuştur.

#### 3.1. Serebellar Model Eklem Kontrolörü (CMAC)

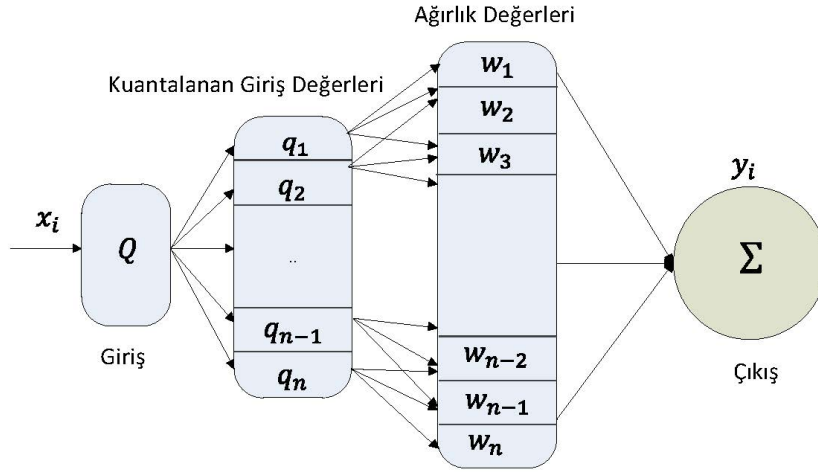
YSA'ların çalışma prensibi insan beyninin nörolojik sinirsel yapısından esinlenerek geliştirilmiştir. Günümüzde birçok bilimsel çalışmalarda ve uygulama alanlarında YSA modelleri kullanılmaktadır. YSA'lar kullandıkları öğrenme algoritmalarına, topolojik yapılarına, kullanılan transfer fonksiyonlarına veya katman sayılarına göre farklı formlarda olabilmektedir.

Bu tez çalışmasında, YSA modellerinden biri olan Serebellar Model Eklem/Artikülasyon Denetleyicisi (Cerebellar Model Articulation Controller, CMAC) sinir ağı incelenmiştir. Klasik YSA modellerinden farklı bir mekanizmaya sahip olan CMAC, 1975'te Albus (Albus, 1975b) tarafından geliştirilmiştir. Serebellum kelimesi beyincik anlamına gelmektedir. Serebellum kaslardan, eklemlerden, uzuvlardan, deriden gelen bilgileri ve geri bildirimleri alır ve kayıt eder (Tan vd., 2022). Ayrıca serebellum vücudumuzun hareket kontrolünü, motor ve öğrenme becerisi de rol almaktadır. Albus, serebellum sinirsel fonksiyonundan esinlenerek bu modeli önermiştir. Model bilgiyi hafızada tutma, hızlı öğrenme ve iyi bir genelleştirme gibi yeteneklere sahiptir (S. Huang vd., 2024). Ayrıca tek katmanlı yapıya sahip olan CMAC yapısı "Look-up table (arama tablosu)" modeline göre çalışmaktadır (Tao vd., 2003). Modelde benzer girişler değerlerine sahip olan vektörler benzer çıkış değerlerine sahiptir (Abdelhameed vd., 2010). CMAC, ileri beslemeli sinir ağı modeli olmasından dolayı elde edilen bilgiyi giriş değerlerinden, çıkış değerlerine kadar taşımaktadır. Giriş değerlerine karşılık gelen adresler, aktif hale getirilmektedir ve çıkış alanına kadar ulaştırılmaktadır. CMAC mimarisi, kullandığı ağırlık dağılımı ve güncelleme yöntemi açısından diğer YSA yapılarına göre farklılıklar içermektedir. Bu farklılıklar;

- i. Giriş uzayının belli seviyelere bölünmesi (kuantalama işlemi),
- ii. Kuantalama seviyeleri ile ağırlıklar arasında CMAC algoritmasına özel bağlantı yapısı,

- iii. Kullanılan öğrenme algoritması,
- iv. Genelleştirme ve hızlı öğrenme yeteneğidir.

CMAC'te giriş değerlerinden çıkış değerlerine kadar iki eşleşme işlemi uygulanmaktadır. Birinci eşleşme, giriş değerlerine kuantalama işlemi uygulayarak belirli seviyelere bölünmesidir. İkinci eşleşme ise, her seviyeye ağırlık değeri atanmasıdır. Şekil 3.1'de bu eşleşmeler, tek boyutlu giriş vektörü  $x_i$ 'den tek boyutlu çıktı vektörü  $y_i$ 'ye sunulmaktadır.



**Şekil 3.1.** Giriş Vektörü  $x_i$ 'den Çıkış Vektörü  $y_i$  kadar ki CMAC Eşleşmesi

Şekil 3.1'deki gibi giriş değerlerini  $q_1$  den  $q_n$ 'e kadar sınır aralığında olduğunu varsayalım. CMAC modelinde başlangıçta  $x_i$  giriş değerlerine kuantalama işlemi uygulanmaktadır. Böylece,  $x_i$  verisi belli bir sayıda seviyeye bölünür. Kuantalama için  $x_i$  giriş değerlerinin en yüksek ve en düşük değerleri bulunur. Denklem 3.1'de CMAC'in kuantalama işleminin matematiksel ifadesi yer almaktadır. Burada  $k$  indeksi kuantalama seviyesini göstermekle birlikte, giriş değerleri arasındaki en düşük ve en yüksek değerler sırasıyla  $x_{\min}$  ve  $x_{\max}$  olarak ifade edilmektedir. Denklem 3.1'deki  $\Delta$  kuantalama çözümünü temsil etmektedir. Kuantalama seviyesi olan  $q_k$  0 ile  $(q_{\max} - 1)$  arasındadır. Denklem 3.2'deki  $q_{\max}$  seviye sayısını yani sinyalin kaç parçaya bölüneceğini ifade etmektedir. Sinyal en kadar yüksek sayıda seviyeye bölünürse o kadar çok anlamlı veri kullanılmış olacaktır.

$$q_k = \frac{x_i - x_{\min}}{\Delta} \quad (3.1)$$

$$\Delta = \frac{x_{\max} - x_{\min}}{q_{\max}} \quad (3.2)$$

CMAC yapısındaki ikinci eşleşme olan her seviye için belli sayıda ağırlık ataması, kuantalama seviyeleri  $q_k$ 'lar ile  $w_n$  ağırlık değerleri arasında uygulanmaktadır. Her kuantalama seviyesinde  $A^*$  kadar ağırlıklar ile eşlenmektedir. Burada  $A^*$  aktif hale getirilen bellek sayısını temsil etmektedir. Şekil 3.1'de her kuantalama seviye için 3 adet ağırlık değeri atanmıştır;

Şekil 3.1'deki  $q_1$  girişi  $w_1, w_2, w_3$  olarak üç adet  $A^*$  ile eşleşmiştir. Denklem 3.3 ve Denklem 3.4'te sırasıyla  $q_1$  ve  $q_2$  girişleri için üç adet ağırlık değerlerinin matematiksel gösterimi bulunmaktadır.

$$Q_1 = w_1, w_2, w_3 \quad (3.3)$$

veya,

$$Q_2 = w_2, w_3, w_4 \quad (3.4)$$

Dolayısı ile bir seviyeden diğerine geçerken belli sayıda ortak ağırlık değişkeni sayısı korunmaktadır. Denklem 3.3 ve Denklem 3.4'te ardışık iki giriş seviyeleri için iki adet ortak ağırlık değerine sahiptir. Bu ortak ağırlık değerleri  $w_2$  ve  $w_3$  olarak belirlenmiştir. Ortak ağırlık kullanımı, CMAC'ın yerel öğrenme kapasitesini yansıtırken, tüm ağırlık değerlerinin birlikte değerlendirilmesi küresel öğrenme ve genelleme yeteneğini temsil etmektedir.

CMAC sinir ağının çıkış değerlerinin hesaplanması, Denklem 3.5'te sunulmuştur. Burada  $Q$  kuantalama seviyelerinden ağırlıkları ifade etmektedir.

$$y_i = \sum_i Q_i w \quad (3.5)$$

CMAC sinir ağı eğitilirken Denklem 3.6'daki matematiksel işlemler uygulanmaktadır. Ağırlıkların güncellenmesi işlemi Denklem 3.6'da verilmiştir. Denklem 3.7'de yer alan  $\eta$  öğrenme katsayısını,  $e(k)$  hata fonksiyonunu ve  $k$  yineleme sayısını göstermektedir.

$$w_j(k+1) = w_j(k) + \Delta w_j(k+1) \quad (3.6)$$

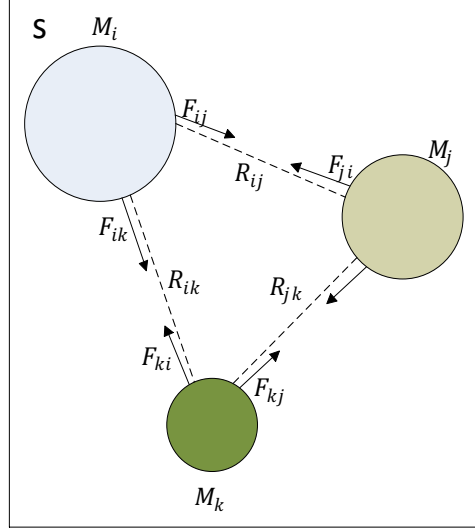
$$\Delta w_j(k+1) = \eta \Delta e(k) \quad (3.7)$$

$$\Delta e(k) = \frac{1}{A^*} e(k) \quad (3.8)$$

### 3.2. Yerçekimsel Arama Algoritması (GSA)

GSA, 2009 yılında Rashedi ve arkadaşları (Rashedi et al., 2009) tarafından Newton'un evrensel çekim kanundan yola çıkarak esinlenilmiş metasezgizel

algoritmalarından biridir. GSA'ya göre uzayda kütlesi olan her nesne diğer nesnelere karşı çekim kuvveti uygulamaktadır. Nesnelere bu kuvvet sayesinde birbirleri arasında etkileşim halinde olup, birbirlerini itip çekme işlemlerini gerçekleştirmektedir. Şekil 3.2'de S arama uzayında bulunan 3 nesnenin birbirlerine uyguladıkları etkileşim kuvveti gösterilmektedir.



**Şekil 3.2.** S Arama Uzayında Nesnelere Arasındaki Mesafe ve Yerçekimi Kuvveti

GSA'da arama uzayında her bir nesne ajan olarak tanımlanmaktadır. Ajanların konumu problemin çözüm uzayını temsil ederken, ajanların arama uzayında yer değiştirerek en iyi sonuca ulaşması beklenmektedir. Başlangıçta, N ajan kümesinin konumları, arama alanı S'de rastgele olarak konumlandırılır. Denklem 3.9'da  $x_i^d$  indeksi  $d$ . boyutta  $i$ . kümenin konumunu,  $n$  arama alanının boyutunu ifade etmektedir.

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n), i = 1, 2, 3, \dots, N \quad (3.9)$$

Ajanlar arasındaki çekim kuvvetinin matematiksel gösterimi Denklem 3.10'da sunulmuştur. Ajanlar arasındaki kuvvet sayesinde daha küçük kütleye sahip olan ajan, daha büyük kütleye sahip olan ajana doğru ivmelenme eğilimindedir. Denklem 3.10'da bulunan  $t$  yinelemesi sayısı,  $G(t)$  yerçekimi sabiti,  $i$  ve  $j$  ajanlarının kütleleri sırasıyla  $M_{pi}(t)$  ve  $M_{aj}(t)$  olarak ifade edilmektedir. Denklemde kütleler arasındaki Öklid mesafesi  $R_{ij}(t)$  ve anlamsızlığı engellemek için çok küçük değer olan  $\epsilon$  kullanılmıştır. Kütleler arasında varsayılan kuvvetin yönü  $[x_j^d(t) - x_i^d(t)]$  olarak temsil edilmektedir. Denklem 3.11'de yerçekimi sabitinin hesaplanması işlemi yer almaktadır. Burada  $\alpha$  yineleme boyunca kütle

çekim sabitinin azalma oranını etkileyen bir kontrol faktörünü,  $t_m$  maksimum yineleme sayısını temsil etmektedir. Öklid mesafe hesabı Denklem 3.12’de yer almaktadır.

$$F_{ij}^d(t) = G(t) \frac{M_{pi}(t).M_{aj}(t)}{R_{ij}(t)+\varepsilon} [x_j^d(t) - x_i^d(t)] \quad (3.10)$$

$$G(t) = G_0 e^{-\alpha \left(\frac{t-1}{t_m-1}\right)} \quad (3.11)$$

$$R_{ij}(t) = \sqrt{\sum_{k=1}^n (X_i^k(t) - X_j^k(t))^2} \quad (3.12)$$

Yineleme boyunca kuvvetten etkilenen ajanlar hareket ederek konumlarını güncellemekte ve yineleme sonunda çözümü temsil eden sabit bir konuma ulaşmaktadır. Ajanlar arasındaki kuvvet sayesinde daha küçük kütleyle sabit olan ajan daha büyük ajana doğru ivmelenme eğilimindedir. Denklem 3.13’te  $i$  ajanın  $t$  yinelemesinde  $d$  boyutta kuvvetten doğan ivmenin matematiksel ifadesi yer almaktadır. İvme formülü aynı zamanda Denklem 3.14’teki gibi yazılmaktadır.

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \quad (3.13)$$

$$a_i^d(t) = \sum_{j=1, i \neq j}^N r_j G(t) \frac{M_j(t)}{R_{ij}(t)+\varepsilon} [x_j^d(t) - x_i^d(t)] \quad (3.14)$$

İvmesi bilinen ajanın, mevcut konumu eklenerek bir sonraki hızı Denklem 3.15’teki gibi hesaplanmaktadır. Burada  $r_i$  [0,1] arasında rastgele bir değerdir. Ajanın mevcut konumu  $v_i^d(t)$  ve ivmesi  $a_i^d(t)$  olarak ifade edilmektedir.

$$v_i^d(t+1) = r_i \times v_i^d(t) + a_i^d(t) \quad (3.15)$$

Son olarak yeni hızı  $v_i^d(t+1)$  ve mevcut konumu  $x_i^d(t)$  olarak bilinen ajanın bir sonraki konumu Denklem 3.16’da sunulmuştur.

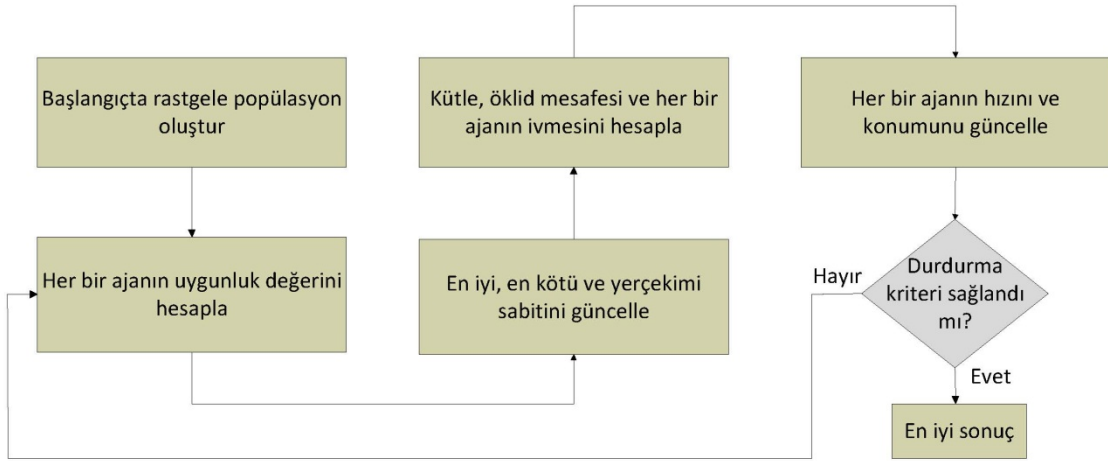
$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (3.16)$$

Yineleme boyunca kütlesi en iyi olan ajanın hedeflenen optimum değere doğru hareket etmesi beklenmektedir. Dolayısıyla ajanların kütleleri Denklem 3.17 ve Denklem 3.18’deki gibi sürekli olarak güncellenir. Burada  $f_i(t)$ ,  $i$  ajanın  $t$  yinelemesinde uygunluk değerini,  $f_w(t)$  en kötü ajanın değerini ve  $f_b(t)$  en iyi ajanın değerini ifade etmektedir.

$$m_i(t) = \frac{f_i(t) - f_w(t)}{f_b(t) - f_w(t)} \quad (3.17)$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)} \quad (3.18)$$

GSA'da ajanların başlangıç konumları rastgele olarak oluşturulmaktadır. Her bir ajanın uygunluk değeri hesaplanmaktadır. Ajanlar arasında en iyi ve en kötü ajan, Öklid mesafesi, yerçekimi sabiti, her bir ajanın kuvveti ve ivmesi hesaplanmaktadır. Bir sonraki adımda her bir ajanın hızı ve konumu güncellenmektedir. Bu işlem en iyi sonucu elde edene kadar yani durdurma şartı gerçekleşene kadar devam etmektedir. Böylece yineleme sonunda en iyi çözüm elde edilmesi beklenmektedir. Şekil 3.3'te GSA'nın akış şeması sunulmuştur.



Şekil 3.3. Klasik GSA'nın Akış Şeması

### 3.3. Parçacık Sürü Optimizasyonu (PSO)

PSO, Kenedy ve Eberhart tarafından 1995 yılında kuş, balık böcek sürülerinin yiyecek arama davranışlarından esinlenerek geliştirilmiş popülasyon tabanlı metasezgisel algoritmalarından biridir (Kennedy & Eberhart, 1995). Doğada bir kuş sürüsü yiyeceğe en yakın konumda olan lider kuşu izleyerek yiyecek kaynağına doğru uçmaktadır. PSO'nun çalışma prensibi sürü zekasına dayanmaktadır. PSO'da sürüdeki her birey parçacık olarak adlandırılır ve çözüm kümesinin birer üyesidir. Algoritmada her parçacık durdurma koşulu gerçekleşene kadar kendi en iyi konumunu ve tüm parçacıkların ulaştığı en iyi çözümü takip etmektedir. Parçacıklar arasında konum bilgisinin aktarımı söz konusudur. PSO'nun problemlere uygulanması ve kodlanması basit olmasından dolayı çalışmalara konu olmuştur. Algoritmanın üç temel kontrol parametresi bulunmaktadır. Bunlar atalet ağırlığı, bilişsel ve sosyal bileşenler olarak ifade edilmektedir. Bu parametrelerde yapılan değişiklikler PSO'nun performansını doğrudan etkilemektedir. Ayrıca keşif ve sömürü arasındaki dengeyi koruma da rol oynamaktadır. Algoritmanın

keşif aşaması parçacıkların arama alanını keşfetmesi, sömürü aşaması iyi çözümler sunan bölgeye odaklanmasıdır.

PSO'nun en iyi çözümü bulmak için izlediği adımlar aşağıdaki gibidir.

- 1- Başlangıçta parçacıkları rastgele olarak konumlandırılır.
- 2- Parçacıkların önceki en iyi çözümleri ve sürüdeki en iyi çözümleri kullanarak yeni hızları ve konumları güncellenir.
- 3- Parçacıkların uygunluk değerleri ve küresel en iyi değerleri güncellenir.
- 4- Durdurma koşulu gerçekleşene kadar adımlar tekrarlanır.

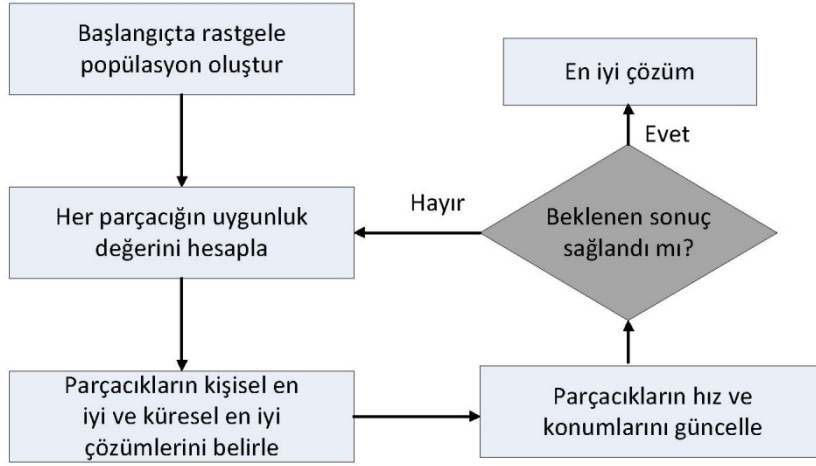
PSO'da parçacıklar, parçacıkların önceki en iyi konumları ve popülasyondaki en iyi çözümün konumlarına göre hareket etmektedir. Bu hareket esnasında iki bileşenden yararlanılmaktadır. Birincisi bilişsel bileşen, bireyin kendi en iyi konumuna doğru hareket etmesi ve ikincisi sosyal bileşen, popülasyon içerisindeki en iyi parçacığa doğru hareket etme durumudur. Denklem 3.19'da PSO'nun hız güncelleme formülü sunulmuştur. Denklemde her parçacığın mevcut hızı  $v_i^t$ , kendi en iyi konumuna olan uzaklığını ve popülasyondaki en iyi konuma olan uzaklığını dikkate alınarak güncellenmektedir. Denklem 3.19'da  $w$  eylemsizlik ağırlığı,  $c_1$  ve  $c_2$  öğrenme katsayıları,  $r_1$  ve  $r_2$   $[0,1]$  arasındaki rastgele değerleri,  $p_i^t$  i. parçacığın kişisel en iyi konumu,  $g^t$  tüm parçacıkların küresel en iyi konumu ve  $x_i^t$  parçacığın mevcut konumunu ifade etmektedir. Eylemsizlik indisi parçacığın önceki hızını koruyarak arama alanın keşfetmesini sağlamaktadır. Denklemde bilişsel bileşen  $c_1 \cdot r_1 \cdot (p_i^t - x_i^t)$  olarak ifade edilirken sosyal bileşen  $c_2 \cdot r_2 \cdot (g^t - x_i^t)$  olarak açıklanmaktadır.

$$v_i^{t+1} = w \cdot v_i^t + c_1 \cdot r_1 \cdot (p_i^t - x_i^t) + c_2 \cdot r_2 \cdot (g^t - x_i^t) \quad (3.19)$$

Parçacığın yeni konumunu hesaplamak için Denklem 3.19'daki güncellenen yeni hız değeri ile mevcut konumun  $x_i^t$  toplanmasıyla elde edilmektedir. Denklem 3.20'de hızı güncellenen parçacığın yeni konum formülü sunulmuştur.

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (3.20)$$

Şekil 3.4'te PSO'nun akış şeması sunulmuştur.

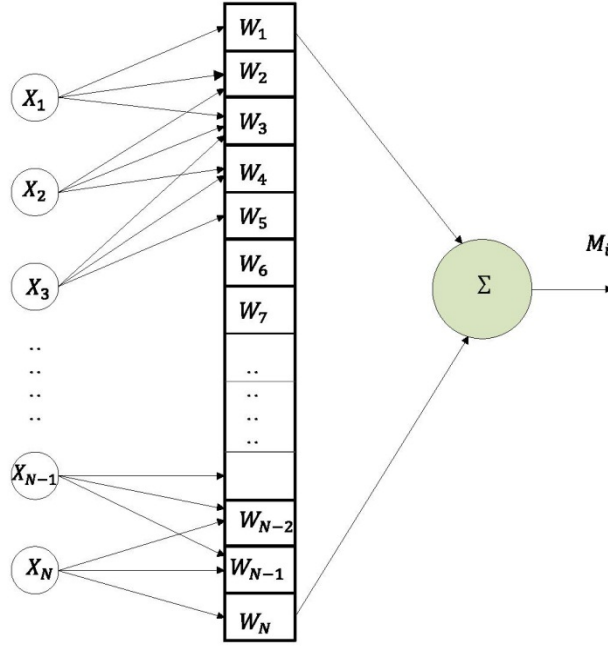


**Şekil 3.4.** PSO Algoritmasının Akış Şeması

### 3.4. İyileştirilmiş CMAC Tabanlı GSA (CMACGSA)

Klasik GSA'nın bazı kısıtlayıcı ve dezavantajlı yönleri bulunmaktadır. Bunlardan ilki karmaşık ve çok modlu fonksiyonlar için yerel minimuma takılıp kalmasıdır ve bu durum optimum sonuca ulaşmayı zorlaştırmaktadır. GSA başlangıçta geniş bir arama alanında arama yaparken yineleme sayısı arttıkça çözümler arasındaki çeşitlilik azalabilir; ajanlar birbirlerine yaklaştıkça algoritmanın keşfetme yeteneği düşebilir. Algoritmanın parametre seçimi performansını etkilemekte ve doğru parametre seçimi zaman alıcı olmaktadır. Ayrıca problemin boyutu arttıkça hesaplama maliyeti artmakta ve algoritmanın verimliliği düşmektedir. GSA'da başlangıçta ajanların konumlarının rastgele olarak belirlenmesi, elde edilen çözümler arasında tutarsızlığa sebep olmaktadır. Ayrıca GSA, bazı mühendislik ve optimizasyon problemlerinin çözümü için uygun olmayabilir. Tez çalışmasında GSA'nın bilinen kısıtlı yönlerini gidermek ve GSA'nın verimliliğini artırmak için CMAC tabanlı iyileştirilmiş CMACGSA algoritması geliştirilmiştir.

Temel olarak, CMAC modelinin öğrenme yeteneği, GSA yapısı içindeki yerçekimi ve eylemsizlik kütlelerini hesaplamak için kullanılmaktadır. Klasik GSA'nın aksine, ajanların kütleleri bir öğrenme sürecinden geçen ağırlıklarla  $w$  hesaplanmaktadır. Yani, ajanların kütleleri CMAC yapısına entegre edilmiş bir öğrenme mekanizması sayesinde dinamik olarak güncellenmektedir.



**Şekil 3.5.** CMACGS A Modelinin Mimarisi

Şekil 3.5'te CMAC modelinin mimari yapısı sunulmuştur. Burada her bir giriş değerine 3 adet ağırlık karşılık gelmektedir. Komşu olan giriş değeri ortak iki adet ağırlığa sahiptir.

CMAC yapısı içerisinde bulunan  $A^*$  indeksi kullanıcı tarafından seçilen ağırlık sayısını ifade etmektedir. Her ajan  $A^*$  sayısı kadar ağırlık değerini aktif etmekte ve her ajanın kütlesi  $A^*$  sayısı kadar ağırlık değerinin toplamına karşılık gelmektedir. Toplam ağırlık sayısı  $N + A^* - 1$  olarak hesaplanmaktadır. Burada  $N$  arama ajanı sayısını ifade etmektedir. Tez çalışmasında, modelin yükünü çok artırmamak için  $A^*=3$  olarak belirlenmiştir. Şekil 3.5'te gösterilen  $X_1, X_2, X_3, \dots, X_{N-1}, X_N$  kadar giriş değerlerini yani ajanları ifade etmektedir. Ajanlara karşılık gelen ağırlık değerleri  $W_1, W_2, W_3, \dots, W_{N-1}, W_N$  olarak gösterilmiştir. Şekil 3.5'de  $X_1$  ajanı  $W_1, W_2, W_3$  kadar ağırlık değerini aktif ederken,  $X_2$  ajanı  $W_2, W_3, W_4$  kadar ağırlık değerini aktif etmektedir. Örneğin  $X_1$  ve  $X_2$  ajanların kütlelerinin hesabı sırasıyla Denklem 3.21 ve Denklem 3.22'deki gibi hesaplanmaktadır.

$$M_1 = W_1 + W_2 + W_3 \quad (3.21)$$

$$M_2 = W_2 + W_3 + W_4 \quad (3.22)$$

Kütle hesaplanmasından da anlaşıldığı gibi ardışık  $A^*$  kadar ağırlık ve  $A^* - 1$  ortak ağırlığı paylaşmaktadır. Bu özellik sayesinde ajanlar arasında bilgi paylaşımını bulunmaktadır. Aynı zamanda çok yakın ajanlar (sıralı ajanlar) benzer çıktılar üretmesine olanak sağlanmaktadır. Yapı içerisinde bulunan farklı çıktılar ise paylaşılmamış

ağırlıkları ifade etmektedir. Dolayısıyla ortak ağırlık değerleri olan ajanların birbirlerini optimum bulma konusunda etkilemiş olması olasıdır.

CMACGSA'daki ajanların kütlelerini hesaplamak için Denklem 3.23 ve 3.24'deki matematiksel ifadeler kullanılmaktadır.

$$M_i(t) = \sum_{k=i}^{i+A^*} W_k(t) \quad (3.23)$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)} \quad (3.24)$$

Kütle hesaplanmasının ardından  $i$ . ajanın ivmesi  $a_i^d(t)$  Denklem 3.25'teki gibi hesaplanmaktadır. Sonrasında  $i$ . ajanın hızı  $v_i^d(t+1)$  ve konumu  $x_i^d(t+1)$  klasik GSA'daki Denklem 3.15 ve Denklem 3.16'daki formüller kullanılarak güncellenmektedir.

$$a_i^d(t) = \sum_{j=1, i \neq j}^N G(t) \frac{M_j(t)}{R_{ij}(t)+\epsilon} [x_j^d(t) - x_i^d(t)] \quad (3.25)$$

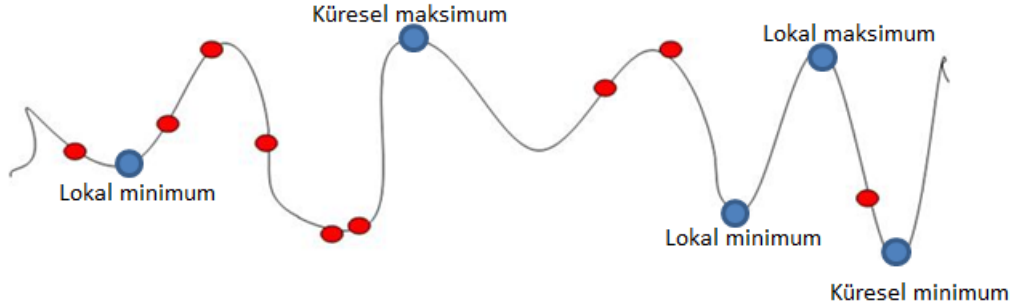
Her yinelemenin sonunda kütle hesaplaması için kullanılan ağırlık değerlerinin eğitimi Denklem 3.26'daki formül kullanılarak hesaplanmaktadır. Burada  $\eta$  öğrenme katsayısı,  $f_w$  en kötü ajanın uygunluk değeri,  $f_b$  en iyi ajanın uygunluk değerini temsil etmektedir. Eğitim sürecinde her bir ajanın ağırlıklarını uygunluk değerlerine göre güncelleme işlemi uygulanmaktadır. Ağırlık güncellemeleri öğrenme oranı ve ajanın uygunluk değerleri arasındaki farklara göre hesaplanmaktadır. En iyi ajan için ağırlık değerleri sabit kalırken, en kötü ajanın ise ağırlıkları en büyük hata oranında güncellenmektedir.

$$W_i(t+1) = W_i(t) + \eta \frac{f_b(t) - f_i(t)}{f_b(t) - f_w(t)} \cdot \frac{1}{A^*} \quad (3.26)$$

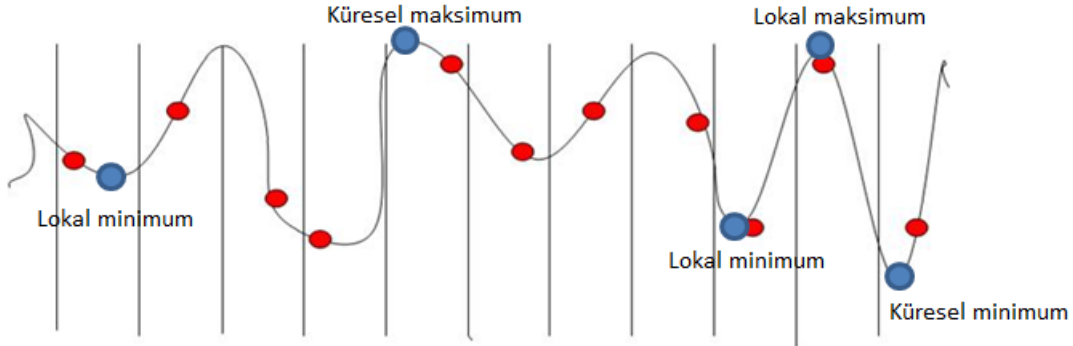
Optimizasyon algoritmalarında, arama uzayındaki her parçacık/ ajan çözüm kümesinin üyeleri olarak simgelenmektedir. Parçacıklar/ ajanlar başlangıçta arama uzayına rastgele olarak konumlandırılmaktadır. Fakat geliştirilen CMACGSA'da başlangıçta ajanlar GSA'daki gibi rastgele olarak konumlandırılmaz. Her ajan başlangıçta CMAC sinir ağının giriş değerlerine uyguladığı kuantalama işlemi ile arama uzayında bölümlendirilmektedir. Böylelikle ağın giriş desenlerinin daha verimli şekilde eşleşmesi sağlanmaktadır. Kuantalama işlemi yapılırken popülasyon sayısı, alt ve üst sınırlar, problemin boyutu gibi faktörler dikkate alınmaktadır.

Şekil 3.6'da klasik GSA'da çözüm uzayına ajanların rastgele konumlandırılması gösterilmektedir. Sinyalin bazı bölgelerinde ajanlar yoğun olarak konumlandırılırken, bazı bölgelerde seyrek olarak konumlandırıldığı gözlemlenmiştir. Şekil 3.7'de CMACGSA

yapısında çözüm uzayına ajanların eşit seviyelere kuantalama işlemi uygulanarak konumlandırıldığı gösterilmektedir. Böylelikle her seviyeyi bir ajan temsil etmektedir. Kuantalama seviyeleri ve ağırlıklar arasında bağlantı sağlanmaktadır. Bu bağlantı sayesinde ajanların en iyi çözüm değerine hızlı ve daha kolay ulaşması beklenmektedir.



**Şekil 3.6.** Giriş Sinyaline Rastgele Atılmış Ajanlar



**Şekil 3.7.** CMACGSA'da Kuantalama Seviyelerine göre Atanan Ajanlar

CMACGSA'nın gelişim aşamasında erken yakınsama, yerel minimuma takılma gibi sorunlarla karşılaşmıştır. Bu sorunların üstesinden gelmek için literatürdeki bazı mekanizmalardan veya diğer algoritmaların güçlü yönlerinden yararlanılarak iyileştirmeler yapılmıştır. CMACGSA'yı geliştirirken üç farklı yenilik algoritmaya dahil edilmiştir. Bunlardan ilki, L-SHADE algoritmasının bir fonksiyonu olan sınır kontrol mekanizmasıdır (Tanabe & Fukunaga, 2014). Algoritmanın yineleme sürecinde ajanların yeni konumları arama sınırlarını aşılabilir ya da sınır dışına çıkma durumuyla karşılaşılabılır. Sınır kontrol mekanizması sayesinde CMACGSA'daki yineleme sırasında ajanların yeni konumlarının arama sınırlarını aşıp aşmadığı kontrol edilmektedir. Mekanizmada bir ajanın yeni konumu eğer alt ve üst sınırları ihlal ederse ajanın önceki yinelemedeki konumu ile sınırlar değerinin ortalaması alınarak konumu tekrar hesaplanmaktadır. Hesaplama işleminin matematiksel ifadesi Denklemi 3.27 ve Denklem 3.28'de yer almaktadır.

$$v_{i,j} = \frac{x_{i,j} + x_{l,j}}{2}, \text{ Eğer } v_{i,j} < x_{l,j} \quad (3.27)$$

$$v_{i,j} = \frac{x_{i,j} + x_{u,j}}{2}, \text{ Eğer } v_{i,j} > x_{u,j} \quad (3.28)$$

Denklem 3.27’de ve Denklem 3.28’de yer alan  $x_{l,j}$  ve  $x_{u,j}$  indeksleri j. boyuttaki ajanın alt ve üst sınırları ifade edilmektedir. Ajanın j. boyuttaki mevcut konumu  $x_{i,j}$  güncellenen konumu  $v_{i,j}$  olarak simgelenmiştir.

CMACGSA’ya eklenen diğer bir yenilik, Lévy Flight mekanizmasına dayalı mutasyon operatörüdür. CMACGSA’yı geliştirme aşamasında her yineleme sürecinden sonra yerel minimuma takılıp kalma sorununu aşmak için bu operatör eklenmiştir. Operatörün amacı çözüm uzayında çeşitliliği arttırmaktır. Mutasyon operatörü Denklem 3.29 ve Denklem 3.30’daki gibi tanımlanmıştır. Denklem 3.30’da yer alan  $x_{i,j}$  ajanın konumunu,  $u_i$  ve  $v_i$  rastgele vektörleri temsil etmektedir. Levy dağılımının parametresi  $\beta = 1.5$  olarak tanımlanmıştır. Normal dağılımlı  $u$  vektörü  $\sigma$  ile ölçeklendirilerek rastgele olarak oluşturulmuştur. Bu vektör mutasyonun yönünü ve büyüklüğünü temsil etmektedir. Diğer bir normal dağılımlı  $v$  vektörü ise Lévy dağılımına göre adım boyutunu ayarlamak için oluşturulmuştur.

$$\sigma = \left( \frac{\Gamma(1+\beta) \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \beta 2^{\frac{\beta-1}{2}}} \right)^{\frac{1}{\beta}} \quad (3.29)$$

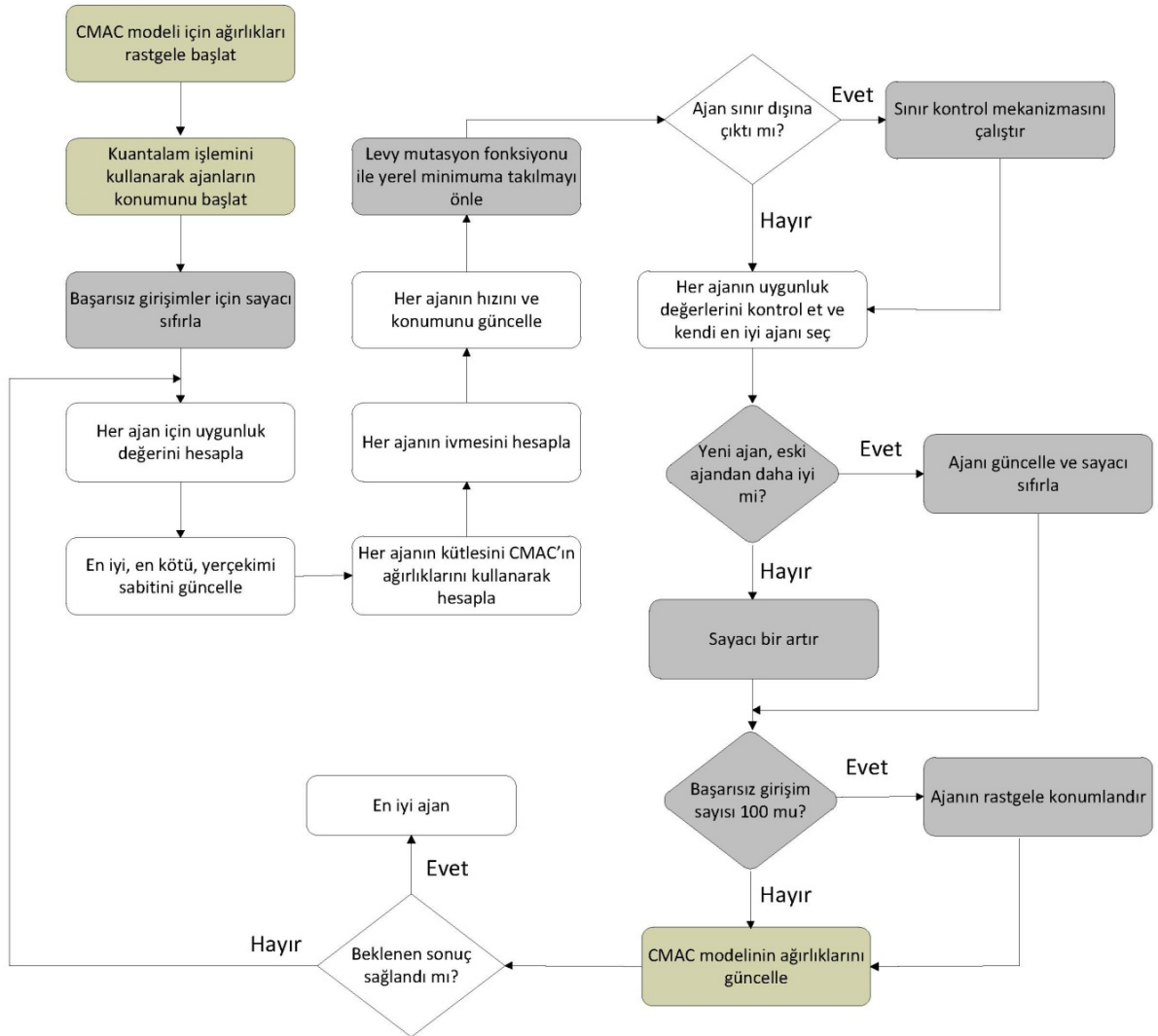
$$x_{i,j} = x_{i,j} \left( 1 + \frac{u_i}{|v_i|^{1/\beta}} \right) \quad (3.30)$$

CMACGSA’ya eklenen son yenilik ise sayaç mekanizmasıdır. Sayaç mekanizması sayesinde bir ajanın konumu belirli sayıda yineleme boyunca iyileşmezse bu ajan arama uzayında rastgele olarak yeniden konumlandırılmaktadır. Böylece arama uzayının daha verimli şekilde keşfedilmesi hedeflenmektedir. Başlangıçta her ajanın başarısızlık sayaç değeri sıfır olarak ayarlanmıştır. Eğer i. ajanın uygunluk değeri  $f_i(t)$  en iyi değer olan  $f_b(t)$ ’den iyi değilse, başarısızlık sayacının değeri bir artırılır. Sayaç değeri toplam 100 olduğunda ajanın konumu Denklem 3.31’deki gibi rastgele olarak güncellenmektedir ve sayaç değeri yeniden sıfırlanmaktadır. Denklem 3.31’deki  $r_i$  değeri  $[0,1]$  arasında rastgele bir sayıdır.

$$x_{i,j} = x_{i,j} + r_i(x_{u,j} - x_{l,j}) \quad (3.31)$$

Özetle klasik GSA’nın başlangıç koşulunda ajanlar çözüm uzayına rastgele olarak konumlandırılırken, CMACGSA’da ajanlar eşit şekilde dağıtılarak seyrek ve yoğun

dağılımın önüne geçilmektedir. Böylece algoritmanın başlangıç noktasında bir iyileştirilme yapılmıştır. CMAC'ın algoritmaya entegrasyonu şu şekildedir: Öğrenme sürecine rehberlik eden aktif ajanlar sayesinde kütlelerin hesaplaması yapılmaktadır. Böylece ajanların davranışlarında dinamik bir öğrenme gerçekleşmesi ve birbirleri ile uyumlu şekilde hareket etmesi hedeflenmiştir. Ajanların yerel minimumlara takılmasını önlemek için Lévy mutasyon fonksiyonu eklenmiştir. Ajanları alt ve üst sınırlar arasında tutmak için sınır kontrol mekanizması algoritmaya dahil edilmiştir. Son olarak, sayaç mekanizması sayesinde belirli sayıda başarısız girişimde bulunan ajanların konumları rastgele olarak değiştirilmiştir. Bu sayede ajanların arama uzayında yeni konumlar keşfetmesi hedeflenmiştir.



**Şekil 3.8.** CMACGSA Algoritmasının Akış Şeması

CMAC'ın ağırlık güncelleme işlemi ajanlar arasındaki uygunluk farkları ile tespit edilmektedir. Daha iyi performans gösteren ajan, öğrenme modeline katkı sağlayarak

optimum çözüme doğru yakınsama eğilimindedir. Bu eğilim, küresel olarak diğer ajanları etkileyerek algoritmanın verimliliğine olumlu etki etmektedir. Şekil 3.8’de CMACGSA’nın akış şemasını gösterilmektedir.

### 3.5. PSO Tabanlı CMAC Yapısına Entegre CMACPSO

PSO basit ve esnek yapısı sayesinde çok sayıda uygulamalarda ve mühendislik problemlerinde etkili çözümler sunmasından dolayı literatürde kendine yer bulmuştur. Bu tez çalışmasında, klasik PSO algoritması üzerinde çeşitli iyileştirmeler yapılarak CMACPSO isminde yeni algoritma geliştirilmiştir. Yapılan iyileştirmelerden biri PSO’nun hız güncelleme denkleminde yer alan rastgele parametrelere  $r_1$  ve  $r_2$ ’ye dinamik öğrenme yeteneği eklenerek  $r_1^*$  ve  $r_2^*$  olarak kullanılmıştır. Her iki dinamik değişken için sunulan denklemlerde  $w_1^{(k,i)}$  ve  $w_2^{(k,i)}$  k parçacığı için CMAC modelinin i. ağırlığını ifade etmektedir. Denklem 3.32’de PSO hız güncelleme formülü yer almaktadır. Dinamik öğrenme yeteneği eklenerek değiştirilen  $r_1^*$  ve  $r_2^*$  değişkenleri Denklem 3.33 ve Denklem 3.34’te yer almaktadır.

$$v_i^{t+1} = w \cdot v_i^t + c_1 \cdot r_1 \cdot (p_i^t - x_i^t) + c_2 \cdot r_2 \cdot (g^t - x_i^t) \quad (3.32)$$

$$r_1^{(k)} = \frac{1}{A} \sum_{i=1}^A w_1^{(k,i)}, w_1 \sim u(0,1), w_1 \in \mathbb{R}^{(A \cdot P_s) \times D} \quad (3.33)$$

$$r_2^{(k)} = \frac{1}{A} \sum_{i=1}^A w_2^{(k,i)}, w_2 \sim u(0,1), w_2 \in \mathbb{R}^{(A \cdot P_s) \times D} \quad (3.34)$$

Her iki dinamik değişken için sunulan denklemlerde  $w_1^{(k,i)}$  ve  $w_2^{(k,i)}$  k parçacığı için CMAC modellerinin i. ağırlığını ifade etmektedir. Her parçacık için ağırlık sayısı A, parçacık sayısı  $P_s$ , problemin boyutu D olarak ifade edilmiştir. Burada dinamik öğrenme yeteneğine sahip iki değişken için CMAC modelinin ağırlık güncelleme mekanizması kullanılmaktadır. Başlangıç ağırlıkları rastgele olarak belirlenmektedir. Her yineleme sonunda CMAC modelinin ağırlıkları Denklem 3.35 ve Denklem 3.36’daki gibi güncellenmektedir.

$$w_1^{(k,i)} \leftarrow w_1^{(k,i)} + \eta * \frac{f(X_{gbest}) - f(X_{pbest}^k)}{f(X_{gbest}) - f(X_{worst}^k)} * \frac{1}{A} \quad (3.35)$$

$$w_2^{(k,i)} \leftarrow w_2^{(k,i)} + \eta * \frac{f(X_{gbest}) - f(X_{pbest}^k)}{f(X_{gbest}) - f(X_{worst}^k)} * \frac{1}{A} \quad (3.36)$$

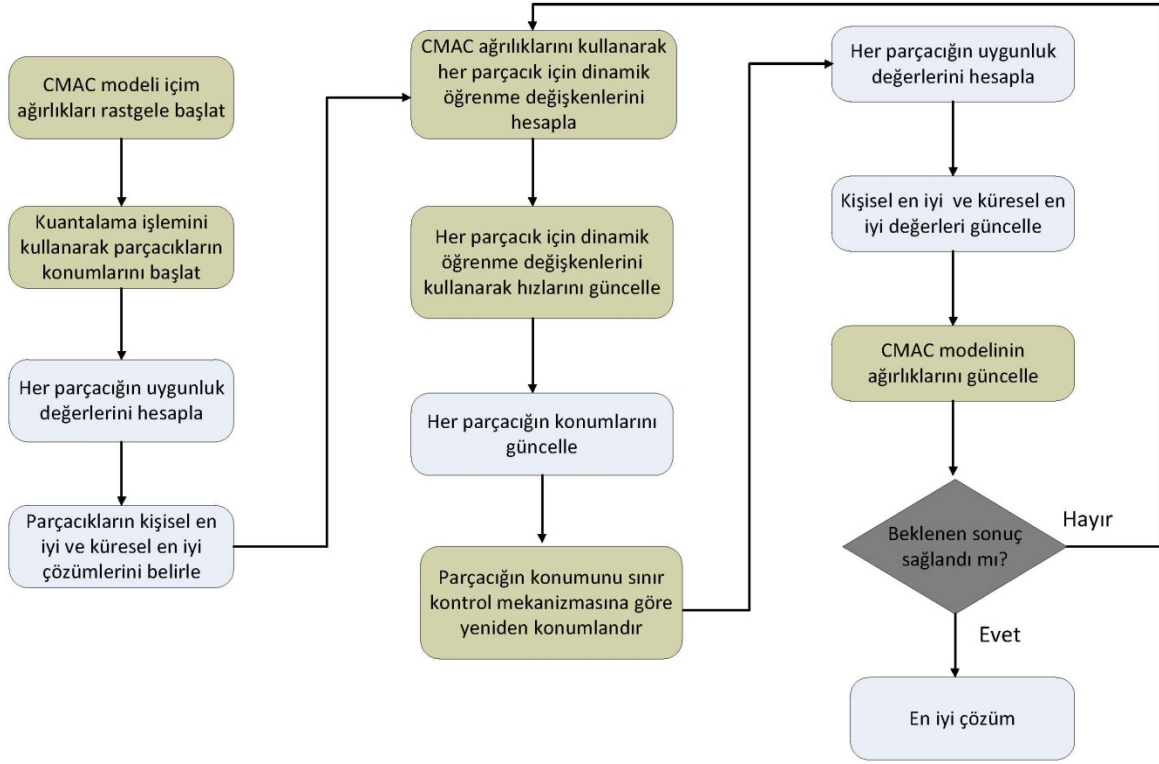
Burada  $\eta$  öğrenme oranı,  $f(X_{gbest})$  en iyi parçacığın uygunluk değeri,  $f(X_{worst}^k)$  en kötü parçacığın uygunluk değeri ve  $f(X_{pbest}^k)$  k. parçacığın kendi en iyi uygunluk değeri olarak ifade etmektedir. CMACPSO’daki dinamik öğrenme değişkenleri

algoritmanın hız güncelleme denkleminde yer aldığı için, keşif ve sömürü arasındaki dengeyi etkilemesinde önemli ölçüde rol almaktadır. Optimizasyon algoritmalarında başlangıçta parçacıklar arama uzayına rastgele olarak konumlandırılırken, CMACPSO’da CMAC sınır ağının başlangıç koşulu olarak parçacıklara kuantalama işlemi uygulanmaktadır. Böylece parçacıkları arama uzayına rastgele konumlandırmak yerine önceden bölümlenmiş aralıklara yerleştirilmektedir. CMAC sınır ağının kuantalama süreci ve ağırlık eşleştirme yapısının klasik PSO’ya entegre edilmesi algoritmanın yenilikçi yönünü göstermektedir. Yineleme sırasında parçacıkların yeni konumlarının arama uzayının sınırlarını aşmasını engellemek için L-SHADE algoritmasının yapısında yer alan sınır kontrol mekanizması eklenmiştir (Tanabe & Fukunaga, 2014). Bu mekanizma bir parçacığın yineleme sırasında belirlenen alt ve üst sınırlarını ihlal etmesine karşın parçacığın konumunu arama sınırları içerisinde yeniden güncellemektedir. Güncelleme işleminde parçacığın kendi en iyi değeri ile sınır değerleri arasındaki ortalaması alınarak hesaplanmaktadır. Hesaplama formülü Denklemi 3.37 ve Denklem 3.38’deki gibi ifade edilmektedir.

$$x_i^j = \frac{p_i^j + x_L^j}{2}, \text{ EĞER } x_i^j < x_L^j \quad (3.37)$$

$$x_i^j = \frac{p_i^j + x_U^j}{2}, \text{ EĞER } x_i^j > x_U^j \quad (3.38)$$

Burada  $i$ . parçacık sayısı,  $j$  problemin boyutu,  $x_L$  arama uzayının alt sınırı,  $x_U$  arama uzayının üst sınırı ve  $p$  parçacığın kendi en iyi değeri olarak ifade edilmektedir. Sınır kontrol mekanizması sayesinde parçacıkların sınırlar içerisinde kalması amaçlanmaktadır. CMACPSO algoritmasının akış şeması Şekil 3.9’da sunulmuştur.



**Şekil 3.9.** CMACPSO Algoritmasının Akış Şeması

Özetle CMACPSO’da CMAC yapısına ait ağırlık değerleri rastgele olarak başlatılırken, PSO’nun parçacıklarına CMAC sinir ağının kuantalama işlemi uygulanmaktadır. Yineleme sırasında parçacıkların uygunluk değerleri, kişisel en iyi çözümü  $p_i$  ve popülasyon içerisindeki en iyi çözüm  $g$  olarak belirlenmektedir. PSO’nun hız güncelleme sürecine CMAC tabanlı dinamik öğrenme yeteneğine sahip  $r_1$  ve  $r_2$  dahil edilerek hesaplanmaktadır. Hızları güncellenen parçacıkların yeni konumları belirlenmektedir. Yeni konumları belirlenen parçacıklara sınır kontrol mekanizması ile sınır ihlali olup olmadığı kontrol edilmektedir. Sınırı aşan parçacıklar Denklem 3.37 ve Denklem 3.38’deki formüller kullanılarak yeniden konumlandırılır. Parçacıkların konumları ve uygunluk değerleri güncellendikten sonra en iyi ve en kötü performans gösteren parçacıklar belirlenmektedir. Optimizasyon boyunca CMAC’in ağırlıkları en kötü parçacığın uygunluk değeri, kişisel en iyi çözümleri ve küresel en iyi çözümleri dikkate alarak dinamik olarak güncellenmektedir. Algoritmanın durdurma koşulu gerçekleşene kadar yineleme devam etmekte ve sonunda parçacıkların en iyi çözüme ulaşması beklenmektedir.

#### 4. DENEYSEL ÇALIŞMALAR

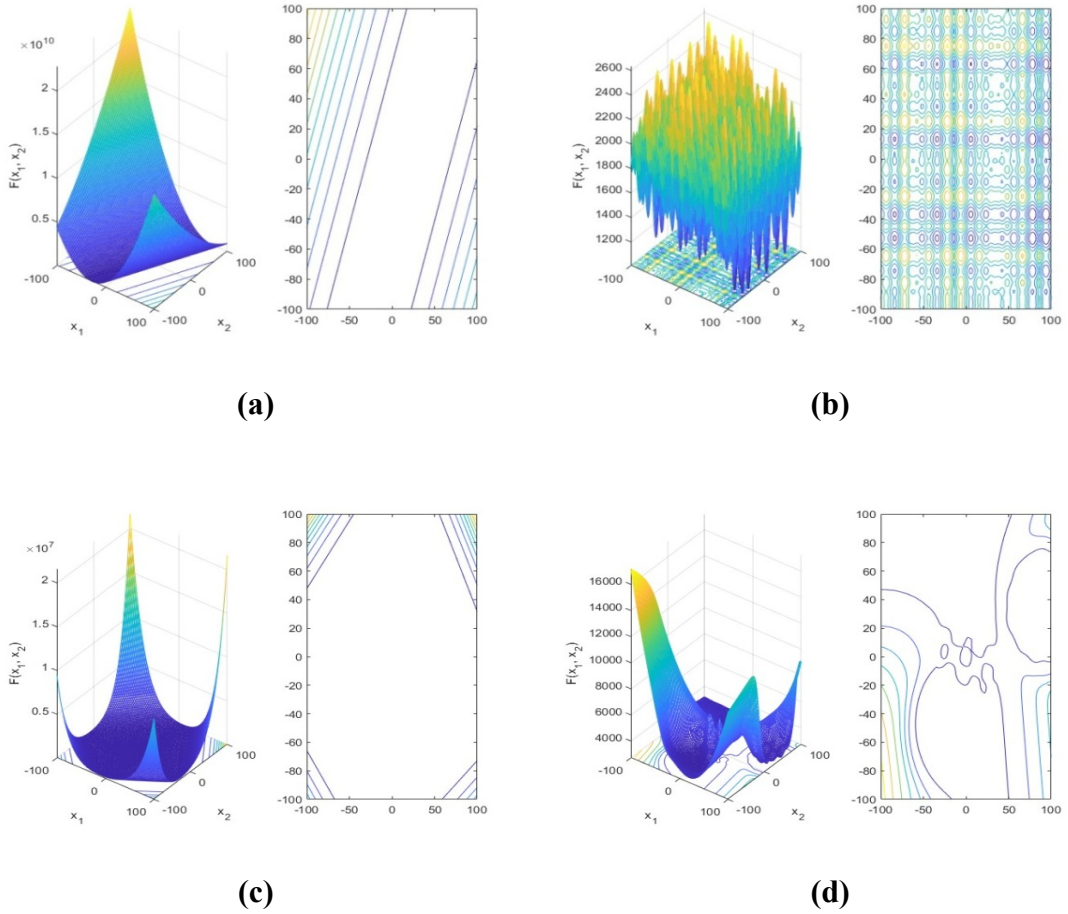
Tez kapsamındaki deneysel çalışmalarda, geliştirilen CMACGSA ve CMACPSO algoritmalarının performanslarını değerlendirmek ve klasik GSA ve PSO'ya göre üstünlüğünü anlamak için test fonksiyonları kullanılmıştır. CMACGSA ve CMACPSO algoritmalarını klasik GSA ve PSO algoritmalarına göre adil bir karşılaştırma yapmak için tüm analizlerde aynı parametreler kullanılmıştır. GSA için parametreler şu şekildedir; popülasyon sayısı 20,  $\alpha$  sabit katsayı 20, başlangıçta yerçekimi sabiti  $G_0$  100, eplison tolerance  $\varepsilon = 1e - 6$  maksimum yineleme sayısı (iterasyon) 1000 olarak belirtilmiştir. CMACGSA'ya özgü parametreler öğrenme oranı  $\eta$  0,001, öğrenme dönemi (epok)  $\alpha=20$ , kuantalama işlemi sırasındaki ağırlık sayısı  $A^* = 3$ , sayaç numarası 100 olarak tanımlanmıştır. CMACPSO'ya ait parametreler şu şekildedir; popülasyon sayısı 100,  $A^* = 3$ , yineleme sayısı 1000.

Ek olarak CEC 2014 test fonksiyonları farklı boyutlarda (10D, 30D, 50D) 51 kez çalıştırılmış ve elde edilen sonuçlar sunulmuştur. Ayrıca CMACGSA ve CMACPSO algoritmalarının 30D boyutunda elde edilen sonuçları, daha önceki çalışmalarda sunulan popüler meta sezgisel algoritmalarla ve hem GSA'nın hem de PSO'nun mevcut varyantları ile karşılaştırılmıştır. Deneysel çalışmalar Intel i7 işlemcili, 2.3Ghz hızında CPU'ya, 8GB RAM'e sahip bir bilgisayarda MATLAB R2022A yazılımı ortamında gerçekleştirilmiştir.

CMACGSA ve CMACPSO algoritmalarının performanslarını değerlendirmek ve geçerliliklerini ölçmek için standartlaştırılmış CEC 2014 test paketinden dört farklı kategoride dört adet kıyaslama fonksiyonu seçilmiştir. Bu fonksiyonlardan 2 boyutta elde edilen sonuçlara göre yakınsama eğrisi ve arama geçmişi analizleri detaylıca sunulmuştur. Bu fonksiyonlar Fonksiyon 1 (F1), Fonksiyon 10 (F10), Fonksiyon 15 (F15), Fonksiyon 27 (F27)'dir.

- Fonksiyon 1 (Rotated High Conditioned Elliptic Function): Bu fonksiyon tek bir küresel optimuma sahip tek modlu fonksiyondur. Şekil 4.1(a)'de F1 fonksiyonun üç boyutlu iz düşümü sunulmuştur.
- Fonksiyon 10 (Shifted Schwefel's Function): Basit, çok modlu (birden fazla yerel minimum noktasına sahip) olan F10 fonksiyonu Schwefel fonksiyonuna dayanmaktadır. Şekil 4.1(b)'de F10 fonksiyonun üç boyutlu iz düşümü yer almaktadır.

- Fonksiyon 15 (Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function): Bu fonksiyon Griewank ve Rosenbrock fonksiyonlarının birleştirilmesi ile oluşturulmuş ve içi içe geçmiş fonksiyondur. Fonksiyonda sırasıyla Griewank fonksiyonu ve Rosenbrock fonksiyonu uygulanır. Şekil 4.1(c)'de F15 fonksiyonun üç boyutlu iz düşümü sunulmuştur.
- Fonksiyon 27 (Composition Function 5 (N=5)): Bileşim Fonksiyonu 5 adet fonksiyonun birleştirilmesi ile oluşturulmuştur. Çok modlu ve yüksek karmaşıklığa sahip bu fonksiyon, algoritmaların hem yerel hem de küresel becerilerini ölçmektedir. Şekil 4.1(d)'de F27 fonksiyonun üç boyutlu iz düşümü bulunmaktadır.



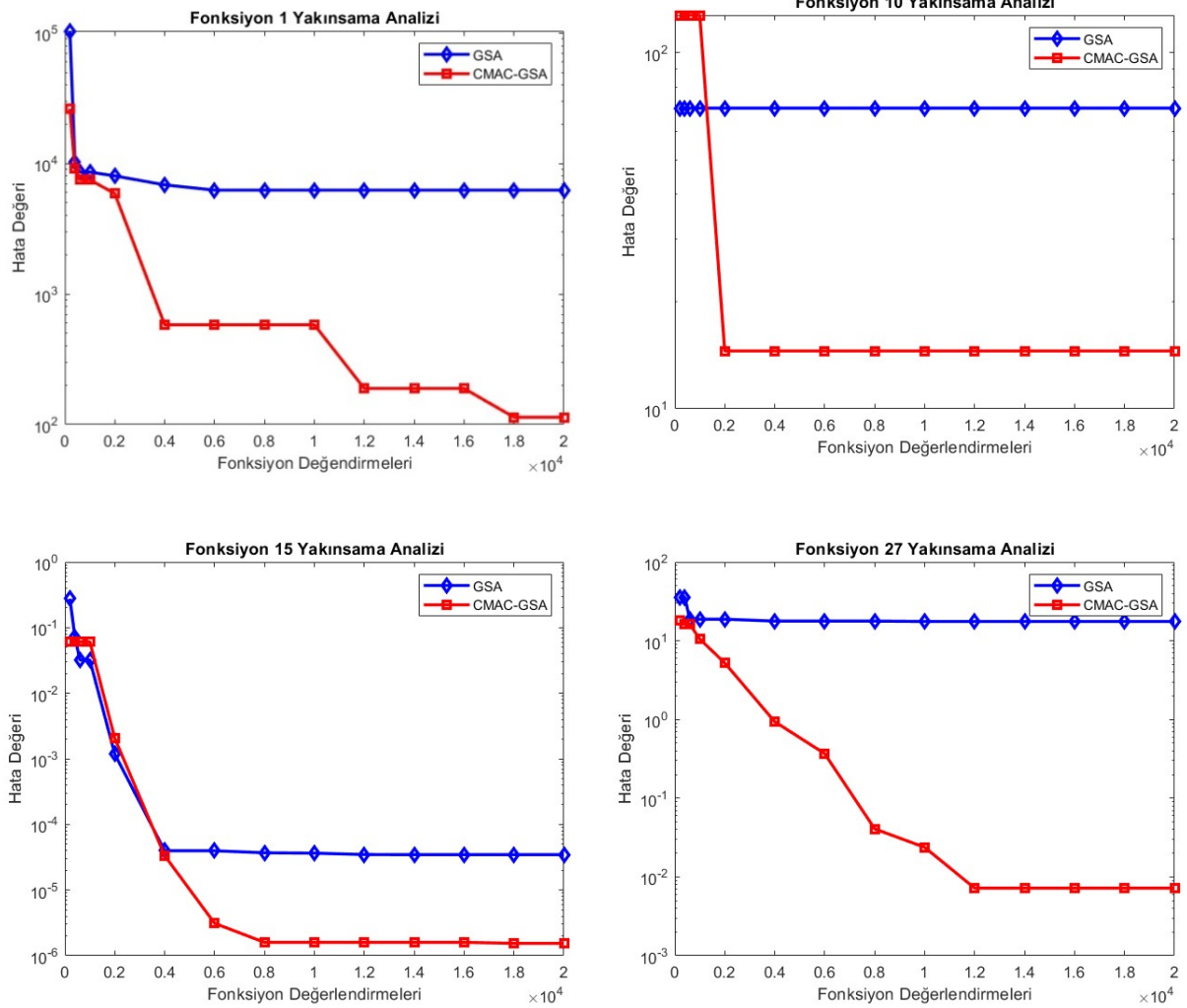
**Şekil 4.1.** Seçilen Dört Fonksiyonun 3 Boyutlu İz Düşümü (a) F1 Fonksiyonu, (b) F10 Fonksiyonu, (c) F15 Fonksiyonu, (d) F27 Fonksiyonu

#### 4.1. Yakınsama Analizi

Bu analiz, bir algoritmanın zaman içerisinde (yineleme boyunca) en iyi çözümlere yaklaşma hızını değerlendirmek için kullanılmaktadır. Yakınsama analizi algoritmanın

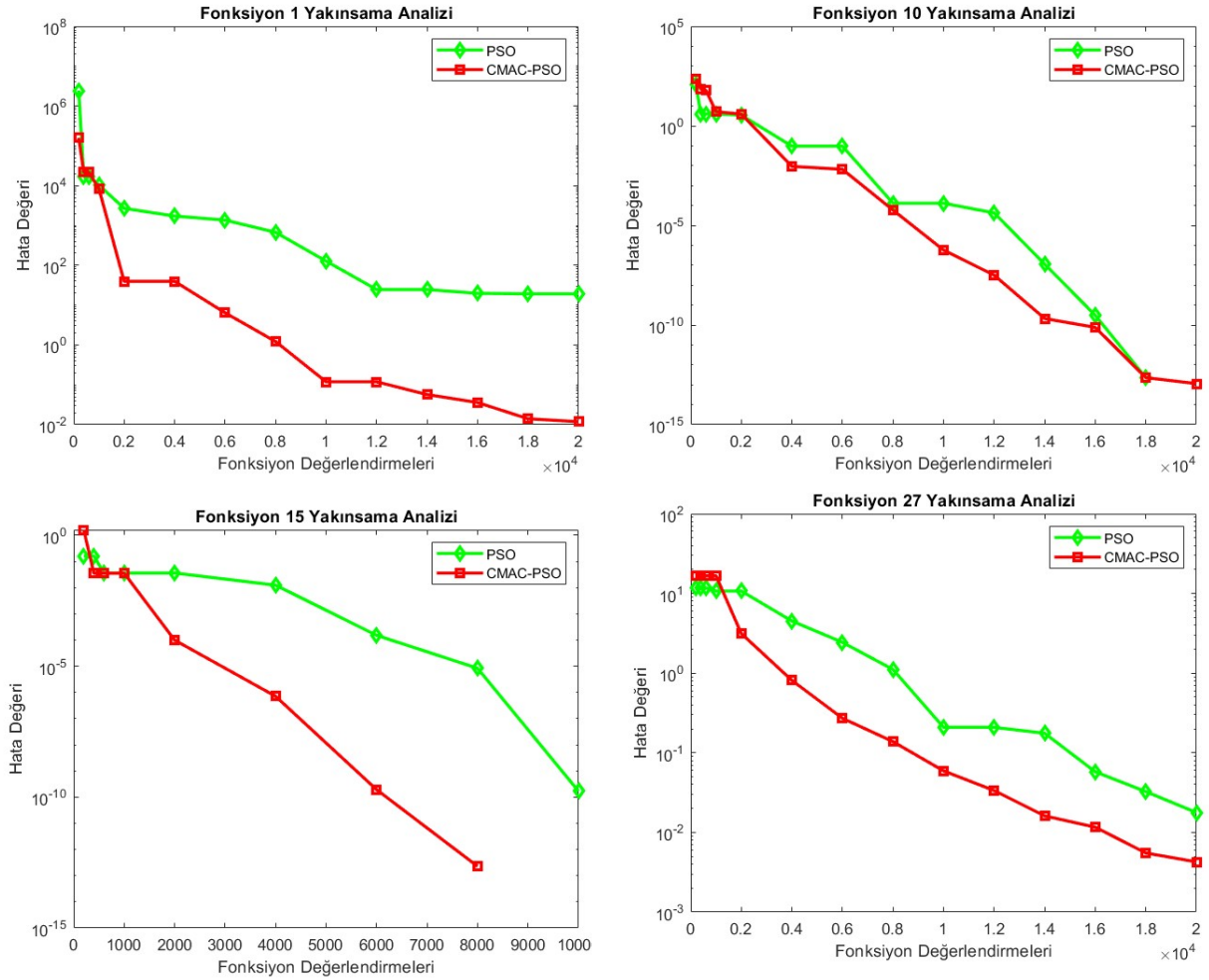
amaç fonksiyonun en iyi sonuca ne kadar yaklaştığı hakkında bilgi vermesinden dolayı önemli bir analizdir. Yakınsama grafiğinde yatay ve dikey eksenler sırasıyla “en iyi uygunluk değeri” ve iterasyon sayısı bulunmaktadır.

Şekil 4.2’de CMACGSA ve klasik GSA algoritmalarının F1, F10, F15, F27 fonksiyonları için yakınsama eğrileri gösterilmektedir. CMACGSA’nın dört kıyaslama fonksiyonu içinde klasik GSA’ya göre daha hızlı yakınsama yeteneğine sahip olduğu görülmektedir. Klasik GSA’nın keşif aşamasında sabit ilerleme, ya da erken yakınsamaya takıldığı görülmektedir. Fonksiyon 15 için klasik GSA başlangıçta CMACGSA ile benzer sonuçlar vermesine rağmen yakınsama durmuştur. Fonksiyon 27 için klasik GSA sabit bir çizgide ilerlerken, CMACGSA kararlı bir şekilde azalmaktadır. Genel olarak CMACGSA çok modlu, rotasyonlu ve bileşik fonksiyonlar için klasik GSA’ya göre üstün başarı göstermiştir.



Şekil 4.2. CMACGSA ve GSA Algoritmalarının Yakınsama Sonuçları

Şekil 4.3'te CMACPSO ve klasik PSO'nun F1, F10, F15, F27 fonksiyonları için yakınsama eğrileri gösterilmektedir. Genel olarak CMACPSO'nun dört farklı türde fonksiyon içinde klasik PSO'ya göre daha hızlı yakınsama yeteneğine sahip olduğu görülmektedir. F1 için CMACPSO algoritması PSO'ya kıyasla daha hızlı ve istikrarlı bir şekilde yakınsama grafiği çizerken, F10'da hem CMACPSO hem de PSO benzer yakınsama eğrisine sahiptir. F15 için CMACPSO algoritması PSO'ya göre erken yakınsama elde etmiştir.

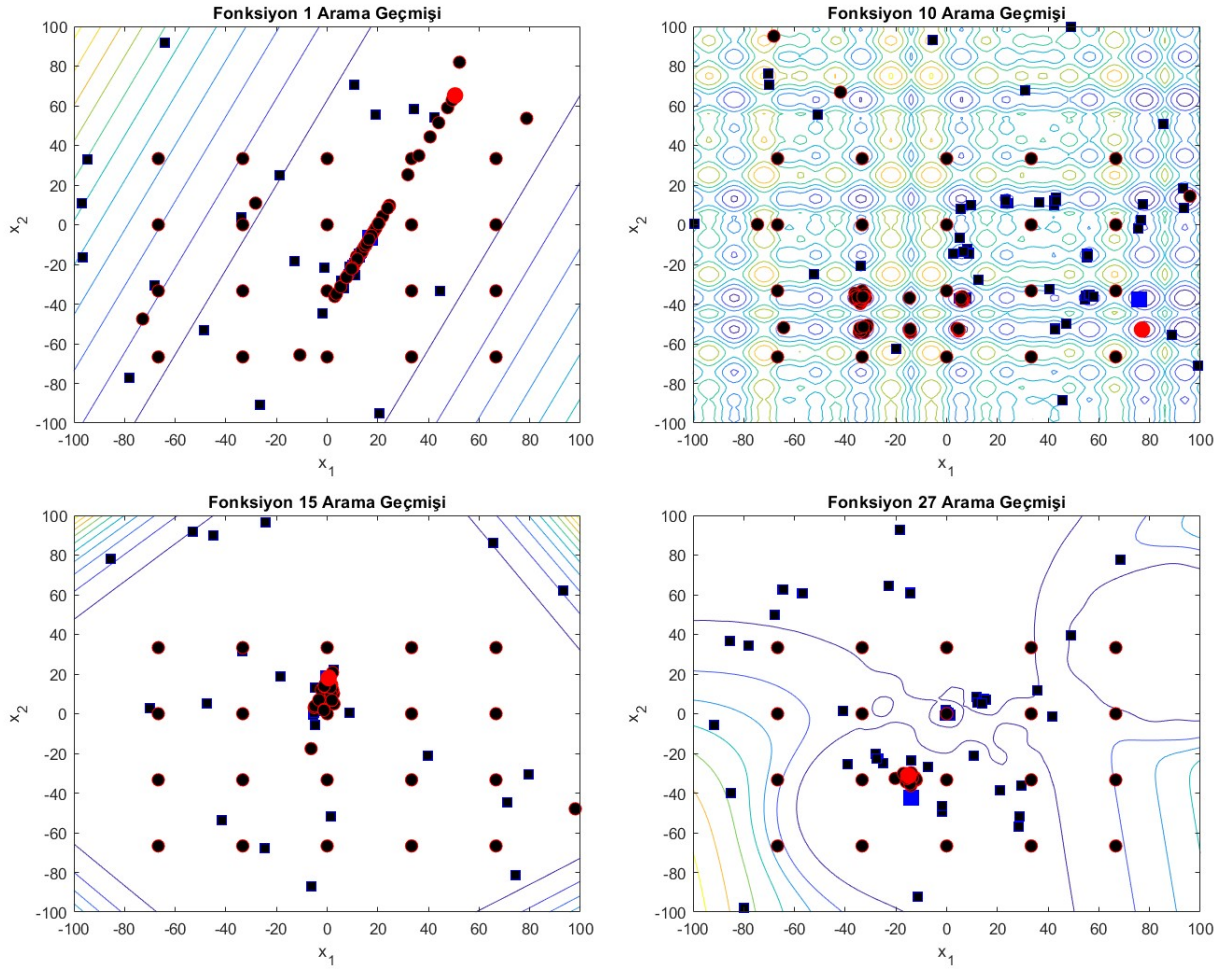


Şekil 4.3. CMACPSO ve PSO Algoritmalarının Yakınsama Sonuçları

#### 4.2. Arama Geçmişi Analizi

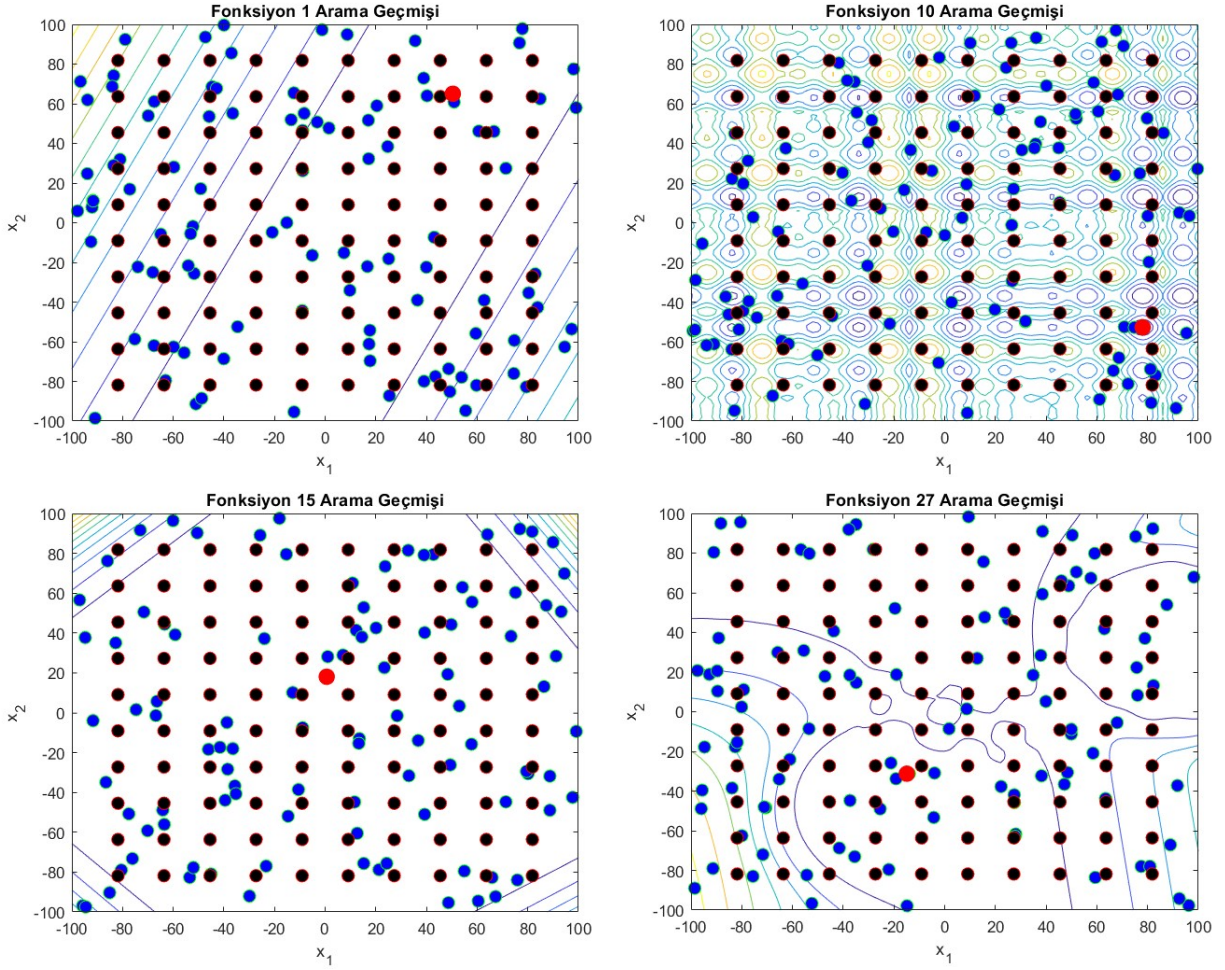
Arama geçmişi analizi, algoritmanın optimum çözüme ulaşma esnasında arama alanında nasıl yol izlediği ve çözüm uzayını nasıl taradığı hakkında bilgiler vermektedir. Temelde algoritmanın problemin çözümünde arama uzayının tamamını taraması beklenir. Ek olarak bu analiz algoritmanın keşif ve sömürü davranışlarını incelemektedir.

Şekil 4.4'te CMACGSA ve klasik GSA algoritmalarının 100 yenileme hızında F1, F10, F15, F27 fonksiyonları için arama geçmişi grafikleri sunulmuştur. Şekilde CMACGSA ve GSA sırasıyla kırmızı daire ve mavi kare şekillerinde simgelenmiştir. Şekil 4.4'te genel olarak CMACGSA arama alanına dağılmış ve etkili bir arama stratejisi ile optimum sonuçlara yaklaşmıştır. Klasik GSA arama alanının geneline dağılmış küresel optimuma ulaşmakta yetersiz kalmıştır.



Şekil 4.4. CMACGSA ve GSA'nın Arama Geçmişi Analizi

Şekil 4.5'te CMACPSO ve klasik PSO algoritmalarının 100 yenileme hızında F1, F10, F15, F27 fonksiyonları için arama geçmişi grafikleri sunulmuştur. Şekilde CMACPSO ve PSO sırasıyla siyah daireler ve mavi daireler olarak simgelenmiştir. Kırmızı noktalar en iyi çözümü ifade etmektedir. Şekil 4.5'te genel olarak CMACPSO algoritması arama alanına daha yoğun ve sistematik olarak dağılmış ve etkili bir arama stratejisi oluşturmuştur. Klasik PSO algoritması ise arama alanına rastgele olarak dağılmış küresel optimuma ulaşmakta yetersiz kalmıştır.



Şekil 4.5. CMACPSO ve PSO' nun Arama Geçmişi Analizi

### 4.3. Kıyaslama Fonksiyonları

Tezin deneysel çalışmasını gerçekleştirmek için IEEE Evrimsel Hesaplama Kongresi tarafından hazırlanan standartlaştırılmış CEC 2014 test seti kullanılmıştır (Liang vd., 2013). Dört temel gruptan oluşan CEC 2014 test paketi EK-1'de sunulmuştur. Toplamda 30 fonksiyondan oluşan test seti tek modlu, çok modlu, hibrit, bileşim olmak üzere 4 farklı gruba ayrılmıştır. Arama uzayı genellikle  $[-100, 100]$  olarak belirlenmiş olup 10D, 30D, 50D boyutlarında test edilmektedir. Bu tez çalışmasında, farklı boyutlardan elde edilen sonuçlar sunulmuştur.

### 4.4. Karmaşıklık Analizi

Algoritmaların karmaşıklık analizi, bir algoritmanın parametre büyüklüğüne bağlı olarak performanslarını değerlendirmek için kullanılan teorik bir yöntemdir. Bir başka deyişle algoritmaların zaman ve bellek kullanımı açısından verimi belirlemek için kullanılmaktadır. CMACGSA ve CMACPSO'nun sırasıyla karmaşıklık analizi şu şekildedir:

CMACGSA'nın her yineleme sırasında zaman karmaşıklığının işlem sırası aşağıdaki gibidir. Algoritmanın başlangıç parametreleri; ajan sayısı (N), problem boyutu (D) olmak üzere,

- i. Kütlelerin güncelleme işlemi sırasında tüm ajanlar (N:arama alanı sayısı) için yapılmaktadır. Zaman Karmaşıklığı  $O(N)$
- ii. Yerçekimi sabitinin hesaplanması iki tane iç içe döngü ile yapılmaktadır. Zaman Karmaşıklığı  $O(N^2)$
- iii. İvmenin hesaplanması işlemi iki adet iç içe döngü ile yapılmaktadır. Zaman Karmaşıklığı  $O(N^2)$
- iv. Her ajan için uygunluk değeri hesaplanması (D boyutta) ve mutasyon işlemini uygulanmaktadır. Zaman Karmaşıklığı  $O(N * D)$
- v. Eğitim işleminde iki adet döngü bulunmaktadır. Bunlar Dış döngü 20 adımda iç döngü N adımda gerçekleştirilir. Zaman Karmaşıklığı  $O(20 * N)$

Toplam zaman karmaşıklığı tüm adımların birleştirilmesi ile elde edilmektedir. CMACGSA'nın zaman karmaşıklığı  $O(20N^2 + N(D + 20 + 1))$  olarak tespit edilmiştir. Zaman karmaşıklığını sadeleştirilmiş hali ise  $O(N^2)$  olarak ortaya çıkmaktadır. Özetle CMACGSA'nın zaman karmaşıklığı klasik GSA ile aynıdır.

CMACPSO'nun her yineleme sırasında zaman karmaşıklığının işlem sırası aşağıdaki gibidir. Algoritmanın başlangıç parametreleri; ajan sayısı (N), problem boyutu D, CMAC'in ajan sayısı (A) olmak üzere,

- i. CMAC ağırlıkları rastgele olarak başlatılmaktadır ve parçacıklar için kuantalama işlemi uygulanmaktadır. Zaman Karmaşıklığı  $O(N * A * D)$
- ii. Parçacıkların uygunluk değerleri hesaplanır, bireylerin en iyi ve küresel en iyi değerler tespit edilmektedir. Zaman karmaşıklığı  $O(N)$
- iii. N parçacık için hız ve konum güncellenmektedir. CMAC tabanlı rastgele değişkenler ( $r_1, r_2$ ) hesaplanır. Zaman karmaşıklığı  $O(N * A * D)$
- iv. Her parçacığın yeni konumu ve uygunluk değerleri hesaplanır. Eğer parçacıklar sınır ihlali yaparsa sınır kontrol mekanizması işlemi uygulanır bu işlemi zaman karmaşıklığı  $O(N)$  olarak hesaplanmaktadır.
- v. Bireylerin en iyi ve küresel en iyi çözümleri  $O(N)$  karmaşıklığında güncellenmektedir.
- vi. CMACPSO'nun yapısında yer alan CMAC sinir ağının öğrenme mekanizmasındaki zaman karmaşıklığı  $O(N * A * D)$  olarak hesaplanır.

Toplamda tüm işlemler T yineleme boyunca çalıştığı varsayılarak  $O(T(N * A * D + N * D + N))$  olarak hesaplanabilir. Bu ifade sadeleştirildiğinde  $O(T * N * A * D)$  olarak belirlenir. CMACPSO'nun klasik PSO'ya kıyasla CMAC'in ağırlıklarının hesaplanması işleminden dolayı ek hesaplama maliyeti bulunmaktadır. CMACPSO öğrenme yeteneğinden dolayı PSO'ya göre daha üstün sonuçlar elde etmesi beklenmektedir.

#### **4.5. CEC 2014 Karşılaştırma Sonuçları**

Çalışmanın bu bölümünde CMACGSA ve CMACPSO algoritmalarının performanslarını daha iyi analiz etmek için CEC 2014 test paketini farklı boyutlarda (10D, 30D, 50D) elli bir kez çalıştırılmıştır. CEC 2014 test paketine ait (10D, 30D, 50D) boyutlu problemler üzerinde elde edilen sonuçlar tablolarda detaylı olarak sunulmuştur. Tablo verilerinde en iyi, en kötü, medyan, ortalama ve standart sapma değerleri yer almaktadır.

Karşılaştırmalar üç farklı grupta incelenmiştir. Bunlardan ilki CMACGSA ve GSA arasında, ikinci olarak CMACPSO ve PSO algoritmaları arasında ve son olarak CMACGSA VE CMACPSO algoritmaları arasında kıyaslanmıştır. Tablolarda en iyi ve etkili çözümleri belirtmek için kalın yazı tipi kullanılmıştır. Üç farklı gruptan elde edilen bulguların bulunduğu sonuçlar özet tablo şeklinde sunulmuştur.

##### **4.5.1. CEC 2014 10D Boyutta Karşılaştırma Sonuçları**

Tablo 4.1'de CMACGSA ve GSA'nın CEC 2014 test paketinin 10 boyutta elde edilen sonuçlar bulunmaktadır. Tablo 4.1'de yer alan sonuçlar incelendiğinde CMACGSA en iyi, en kötü ve ortalama değerler için 30 problemde 20'sinde üstün başarı elde etmiş, medyan değerler için 18 tanesinde başarılı sonuçlar vermiştir. Standart sapma için CMACGSA ve GSA benzer sonuçlar elde etmiştir.

**Tablo 4.1.** CMACGSA ve GSA Algoritmaları için CEC 2014 10D ile Test Sonuçları

No	GSA					CMACGSA				
	En iyi	En kötü	Medyan	Ortalama	Std	En iyi	En kötü	Medyan	Ortalama	Std
1	<b>42.622</b>	<b>39620</b>	<b>4383.1</b>	<b>7480.9</b>	<b>7462.5</b>	1176.6	3.75e+5	27569	45917	70803
2	<b>1331.3</b>	<b>7662.2</b>	<b>3255</b>	<b>3413.4</b>	<b>1573.9</b>	4535.1	44912	14274	15123	6808.4
3	<b>707.2</b>	<b>8714</b>	<b>4039.3</b>	<b>4297.4</b>	<b>1936.6</b>	4894.9	17357	10233	9979.2	3022.6
4	<b>0.0241</b>	<b>34.781</b>	<b>34.781</b>	<b>33.418</b>	6.812	34.783	34.793	34.788	34.788	<b>0.0022</b>
5	20.189	<b>20.515</b>	<b>20.345</b>	20.354	<b>0.0682</b>	<b>20.137</b>	20.527	20.36	<b>20.348</b>	0.0751
6	3.2621	7.8805	5.6291	5.5606	<b>0.9799</b>	<b>0.2247</b>	<b>4.9173</b>	<b>1.8229</b>	<b>2.0526</b>	1.2785
7	<b>0.0425</b>	0.5025	<b>0.0961</b>	<b>0.1534</b>	0.1299	0.1101	<b>0.4805</b>	0.311	0.2997	<b>0.0829</b>
8	22.889	77.98	33.834	39.186	13.876	<b>14.941</b>	<b>44.007</b>	<b>22.906</b>	<b>23.338</b>	<b>5.9024</b>
9	18.912	59.88	29.852	30.468	<b>7.3829</b>	<b>10.957</b>	<b>50.739</b>	<b>21.896</b>	<b>26.389</b>	11.775
10	757.92	2237.8	1759.2	1663	348.97	<b>673.07</b>	<b>1639.9</b>	<b>1265.4</b>	<b>1219.9</b>	<b>193.4</b>
11	<b>469.26</b>	1941.7	1440.5	1322.9	383.25	664.7	<b>1412</b>	<b>1160.2</b>	<b>1135.6</b>	<b>180.22</b>
12	<b>0.5727</b>	1.556	1.1752	1.1721	0.1965	0.7823	<b>1.5477</b>	<b>1.1541</b>	<b>1.1598</b>	<b>0.1750</b>
13	<b>0.0705</b>	0.2836	0.1502	0.1552	0.0502	0.0888	<b>0.2333</b>	<b>0.1421</b>	<b>0.1465</b>	<b>0.0359</b>
14	0.2396	0.4698	0.4144	0.4035	<b>0.0477</b>	<b>0.1194</b>	<b>0.4647</b>	<b>0.3809</b>	<b>0.3721</b>	0.0557
15	<b>0.7590</b>	<b>2.4871</b>	<b>1.6493</b>	<b>1.6365</b>	0.354	1.1815	2.7349	1.9796	1.9694	<b>0.3034</b>
16	2.5933	3.5738	3.1317	3.1121	<b>0.2584</b>	<b>1.8767</b>	<b>3.0469</b>	<b>2.5645</b>	<b>2.5458</b>	0.2952
17	593	<b>4127.2</b>	<b>1058.5</b>	<b>1349.9</b>	<b>716.64</b>	<b>578.7</b>	4.00e+5	9419.2	8.73e+5	1.23e+5
18	2059.8	7910.2	5335.9	5336.9	1208.5	<b>280.54</b>	<b>4470.4</b>	<b>2710.5</b>	<b>2556</b>	<b>973.95</b>
19	1.662	<b>4.747</b>	3.2525	3.1975	<b>0.6077</b>	<b>1.5975</b>	4.7593	<b>3.2144</b>	<b>2.9296</b>	0.7395
20	311.57	4095.8	1759.2	1943.2	955.05	<b>94.177</b>	<b>2381.4</b>	<b>698.82</b>	<b>874.25</b>	<b>575.82</b>
21	<b>65.303</b>	<b>2724.1</b>	<b>583.38</b>	<b>691.13</b>	<b>555.54</b>	250.61	10255	1826.2	1968.2	1569.5
22	143.64	235.92	<b>168.14</b>	168.08	<b>21.063</b>	<b>99.555</b>	<b>186.7</b>	173.73	<b>164.98</b>	22.582
23	<b>201.57</b>	<b>329.46</b>	<b>202.15</b>	<b>222.03</b>	<b>46.799</b>	202.8	<b>329.46</b>	329.46	287.51	59.913
24	189.56	<b>203.62</b>	202.13	200.77	<b>3.2101</b>	<b>110.93</b>	203.98	<b>145.41</b>	<b>149.11</b>	27.996
25	191.06	200.2	<b>199.64</b>	<b>199.24</b>	<b>1.4242</b>	<b>189.42</b>	<b>200.17</b>	200.17	199.36	2.2838
26	100.1	178.7	104.3	106.13	10.862	<b>100.06</b>	<b>103.01</b>	<b>100.72</b>	<b>101.1</b>	<b>0.9779</b>
27	261.74	746.64	573.21	572.51	110.33	<b>184.66</b>	<b>443.59</b>	<b>365.21</b>	<b>367.15</b>	<b>46.789</b>
28	935.97	2233.8	1523.8	1582.4	317.83	<b>112.28</b>	<b>439.92</b>	<b>116.21</b>	<b>231.96</b>	<b>147.77</b>
29	5.66e+5	1.11e+8	3.46e+7	3.39e+7	2.50e+7	<b>3.40e+2</b>	<b>1.89e+5</b>	<b>5.53e+2</b>	<b>7.90e+3</b>	<b>30768</b>
30	2826.3	1.21e+5	15127	29473	30890	<b>855.07</b>	<b>2886.2</b>	<b>2009.7</b>	<b>1971.9</b>	<b>407.67</b>

Tablo 4.2’de CMACPSO ve PSO algoritmalarının 10 boyutta elde edilen sonuçlar, bulunmaktadır. Tablo 4.2’de yer alan sonuçlar incelendiğinde CMACPSO en iyi değerler için 30 problemde 18’inde, en kötü değerler için 17’sinde, medyan değerler için 20’sinde ve ortalama değerler için 30 problemde 19’ unda üstün başarılı sonuçlar elde edilmiştir. PSO standart sapma değeri için 30 problemde 17 tanesinde başarılı olmuştur.

**Tablo 4.2.** CMACPSO ve PSO Algoritmaları için CEC 2014 10D ile Test Sonuçları

No	PSO					CMACPSO				
	En iyi	En kötü	Medyan	Ortalama	Std	En iyi	En kötü	Medyan	Ortalama	Std
1	1.33e+5	4.09e+6	4.30e+5	9.27e+5	9.96e+5	<b>1.04e+2</b>	<b>3.36e+4</b>	<b>1.92e+3</b>	<b>2.68e+3</b>	<b>4.63e+3</b>
2	<b>4.39e-1</b>	1.18e+4	9.86e+3	7.95e+3	4.05e+3	1.60e+0	<b>3.56e+3</b>	<b>8.93e+2</b>	<b>1.18e+3</b>	<b>9.58e+2</b>
3	3.25e-1	2.90e+4	3.87e+2	2.02e+3	4.67e+3	<b>1.28e-1</b>	<b>1.79e+3</b>	<b>4.42e+1</b>	<b>2.31e+2</b>	<b>4.01e+2</b>
4	8.42e-1	1.26e+2	3.48e+1	3.07e+1	1.62e+1	<b>1.79e-2</b>	<b>3.48e+1</b>	<b>3.48e+1</b>	<b>2.71e+1</b>	<b>1.40e+1</b>
5	2.00e+1	<b>2.04e+1</b>	<b>2.02e+1</b>	<b>2.02e+1</b>	<b>1.04e-1</b>	<b>2.00e+1</b>	2.05e+1	2.03e+1	2.03e+1	1.19e-1
6	1.94e-3	8.51e+0	2.74e+0	3.05e+0	1.91e+0	<b>8.54e-9</b>	<b>6.09e+0</b>	<b>1.71e+0</b>	<b>1.95e+0</b>	<b>1.59e+0</b>
7	3.20e-2	1.93e+1	3.88e+0	4.98e+0	4.70e+0	<b>2.95e-2</b>	<b>4.37e+0</b>	<b>1.25e-1</b>	<b>2.10e-1</b>	<b>5.98e-1</b>
8	<b>9.95e-1</b>	2.43e+1	1.13e+1	1.11e+1	4.81e+0	2.98e+0	<b>2.09e+1</b>	<b>9.33e+0</b>	<b>9.63e+0</b>	<b>3.52e+0</b>
9	<b>2.98e+0</b>	<b>4.11e+1</b>	1.59e+1	<b>1.79e+1</b>	<b>8.01e+0</b>	4.97e+0	4.61e+1	1.59e+1	1.79e+1	9.21e+0
10	<b>1.52e+1</b>	8.09e+2	<b>2.59e+2</b>	<b>2.64e+2</b>	<b>1.84e+2</b>	1.56e+1	<b>7.57e+2</b>	2.69e+2	3.11e+2	1.90e+2
11	3.06e+1	<b>1.11e+3</b>	<b>4.10e+2</b>	<b>4.20e+2</b>	<b>2.26e+2</b>	<b>3.02e+1</b>	1.68e+3	6.52e+2	6.50e+2	2.90e+2
12	<b>4.43e-2</b>	<b>9.36e-1</b>	<b>1.39e-1</b>	<b>2.01e-1</b>	<b>1.56e-1</b>	6.63e-2	1.36e+0	6.13e-1	5.93e-1	3.21e-1
13	<b>8.90e-2</b>	<b>5.23e-1</b>	2.99e-1	<b>3.06e-1</b>	<b>9.06e-2</b>	1.03e-1	5.59e-1	<b>1.83e-1</b>	<b>2.14e-1</b>	9.38e-2
14	1.12e-1	2.40e+0	7.43e-1	6.12e-1	4.49e-1	<b>1.11e-1</b>	<b>1.07e+0</b>	<b>2.49e-1</b>	<b>3.13e-1</b>	<b>2.08e-1</b>
15	<b>2.63e-1</b>	4.33e+1	1.02e+0	2.70e+0	8.23e+0	2.88e-1	<b>3.06e+0</b>	<b>8.74e-1</b>	<b>1.03e+0</b>	<b>5.81e-1</b>
16	<b>1.57e+0</b>	<b>3.45e+0</b>	<b>2.56e+0</b>	<b>2.53e+0</b>	<b>4.00e-1</b>	1.91e+0	3.99e+0	2.98e+0	2.91e+0	4.01e-1
17	3.12e+2	1.86e+4	8.07e+3	8.70e+3	6.46e+3	<b>2.50e+2</b>	<b>8.96e+3</b>	<b>2.17e+3</b>	<b>2.98e+3</b>	<b>2.23e+3</b>
18	1.15e+2	3.74e+4	<b>3.73e+3</b>	1.01e+4	1.26e+4	<b>2.82e+1</b>	<b>2.27e+4</b>	5.28e+3	<b>6.32e+3</b>	<b>5.68e+3</b>
19	1.51e+0	7.11e+0	4.40e+0	4.47e+0	<b>1.31e+0</b>	<b>1.16e-1</b>	<b>5.87e+0</b>	<b>2.40e+0</b>	<b>2.60e+0</b>	1.47e+0
20	2.26e+1	<b>2.27e+3</b>	<b>1.48e+2</b>	<b>3.09e+2</b>	<b>4.61e+2</b>	<b>5.84e+0</b>	7.84e+3	5.80e+2	1.39e+3	2.00e+3
21	<b>1.37e+1</b>	2.98e+4	2.96e+2	1.03e+3	4.14e+3	1.86e+1	<b>8.54e+2</b>	<b>2.42e+2</b>	<b>2.67e+2</b>	<b>1.73e+2</b>
22	<b>3.91e-1</b>	<b>5.92e+1</b>	<b>2.75e+1</b>	<b>3.16e+1</b>	<b>1.28e+1</b>	8.77e+0	2.11e+2	3.72e+1	8.07e+1	7.20e+1
23	<b>3.29e+2</b>	3.61e+2	3.29e+2	3.36e+2	9.66e+0	<b>3.29e+2</b>	<b>3.29e+2</b>	<b>3.29e+2</b>	<b>3.29e+2</b>	<b>1.72e-13</b>
24	1.10e+2	<b>1.35e+2</b>	1.24e+2	<b>1.24e+2</b>	<b>6.43e+0</b>	<b>1.08e+2</b>	1.47e+2	<b>1.24e+2</b>	1.24e+2	8.77e+0
25	1.25e+2	<b>2.03e+2</b>	2.01e+2	1.89e+2	<b>2.63e+1</b>	<b>1.16e+2</b>	2.03e+2	<b>2.01e+2</b>	<b>1.87e+2</b>	2.73e+1
26	1.00e+2	1.01e+2	1.00e+2	1.00e+2	<b>1.57e-1</b>	<b>1.00e+2</b>	<b>1.01e+2</b>	<b>1.00e+2</b>	<b>1.00e+2</b>	1.64e-1
27	<b>1.68e+0</b>	5.07e+2	4.01e+2	3.33e+2	<b>1.66e+2</b>	2.05e+0	<b>4.99e+2</b>	<b>3.42e+2</b>	<b>2.60e+2</b>	1.79e+2
28	3.88e+2	<b>6.77e+2</b>	4.91e+2	4.96e+2	<b>7.11e+1</b>	<b>4.55e-13</b>	7.96e+2	<b>4.85e+2</b>	<b>4.80e+2</b>	1.10e+2
29	3.67e+2	<b>3.55e+6</b>	7.61e+2	<b>1.79e+5</b>	<b>6.53e+5</b>	<b>2.55e+2</b>	3.63e+6	<b>4.62e+2</b>	2.50e+5	8.12e+5
30	<b>4.28e+2</b>	<b>2.38e+3</b>	<b>7.64e+2</b>	<b>8.60e+2</b>	<b>3.92e+2</b>	4.92e+2	3.02e+3	8.88e+2	1.02e+3	5.65e+2

Tablo 4.3'te CMACPSO ve CMACGSA algoritmalarını 10 boyutta elde edilen karşılaştırmalı sonuçları sunulmaktadır. Tablo 4.3'te yer alan sonuçlar incelendiğinde CMACPSO'nun CMACGSA'ya kıyasla üstün performans sergilediği görülmüştür. Özellikle en iyi değerler için 30 problemde 27'sinde, medyan değerler için 24'ünde ve ortalama değerler için 23'ünde başarılı performans göstermiştir. Bulgulara göre

CMACPSO'nun CMACGSA'ya göre problemlere karşı daha istikrarlı ve etkili çözümler üretme yeteneğine sahip olduğu gözlemlenmiştir.

**Tablo 4.3.** CMACPSO ve CMACGSA Algoritmaları için CEC 2014 10D ile Test Sonuçları

No	CMACPSO					CMACGSA				
	En iyi	En kötü	Medyan	Ortalama	Std	En iyi	En kötü	Medyan	Ortalama	Std
1	1.04e+2	3.36e+4	1.92e+3	2.68e+3	4.63e+3	1176.6	3.75e+5	27569	45917	70803
2	1.60e+0	3.56e+3	8.93e+2	1.18e+3	9.58e+2	4535.1	44912	14274	15123	6808.4
3	1.28e-1	1.79e+3	4.42e+1	2.31e+2	4.01e+2	4894.9	17357	10233	9979.2	3022.6
4	1.79e-2	3.48e+1	3.48e+1	2.71e+1	1.40e+1	34.783	<b>34.793</b>	<b>34.788</b>	34.788	<b>0.0022</b>
5	2.00e+1	2.05e+1	2.03e+1	2.03e+1	1.19e-1	20.137	20.527	20.36	20.348	<b>0.0751</b>
6	8.54e-9	6.09e+0	1.71e+0	1.95e+0	1.59e+0	0.2247	<b>4.9173</b>	1.8229	2.0526	<b>1.2785</b>
7	2.95e-2	4.37e+0	1.25e-1	2.10e-1	5.98e-1	0.1101	<b>0.4805</b>	0.311	0.2997	<b>0.0829</b>
8	2.98e+0	2.09e+1	9.33e+0	9.63e+0	3.52e+0	14.941	44.007	22.906	23.338	5.9024
9	4.97e+0	4.61e+1	1.59e+1	1.79e+1	9.21e+0	10.957	50.739	21.896	26.389	11.775
10	1.56e+1	7.57e+2	2.69e+2	3.11e+2	1.90e+2	673.07	1639.9	1265.4	1219.9	193.4
11	3.02e+1	1.68e+3	6.52e+2	6.50e+2	2.90e+2	664.7	<b>1412</b>	1160.2	1135.6	<b>180.22</b>
12	6.63e-2	1.36e+0	6.13e-1	5.93e-1	3.21e-1	0.7823	1.5477	1.1541	1.1598	<b>0.1750</b>
13	1.03e-1	5.59e-1	1.83e-1	2.14e-1	9.38e-2	<b>0.0888</b>	<b>0.2333</b>	<b>0.1421</b>	<b>0.1465</b>	<b>0.0359</b>
14	1.11e-1	1.07e+0	2.49e-1	3.13e-1	2.08e-1	0.1194	<b>0.4647</b>	0.3809	0.3721	<b>0.0557</b>
15	2.88e-1	3.06e+0	8.74e-1	1.03e+0	5.81e-1	1.1815	<b>2.7349</b>	1.9796	1.9694	<b>0.3034</b>
16	1.91e+0	3.99e+0	2.98e+0	2.91e+0	4.01e-1	<b>1.8767</b>	<b>3.0469</b>	<b>2.5645</b>	<b>2.5458</b>	<b>0.2952</b>
17	2.50e+2	8.96e+3	2.17e+3	2.98e+3	2.23e+3	578.7	4.00e+5	9419.2	8.73e+5	1.23e+5
18	2.82e+1	2.27e+4	5.28e+3	6.32e+3	5.68e+3	280.54	<b>4470.4</b>	<b>2710.5</b>	<b>2556</b>	<b>973.95</b>
19	1.16e-1	5.87e+0	2.40e+0	2.60e+0	1.47e+0	1.5975	<b>4.7593</b>	3.2144	2.9296	<b>0.7395</b>
20	5.84e+0	7.84e+3	5.80e+2	1.39e+3	2.00e+3	94.177	<b>2381.4</b>	698.82	<b>874.25</b>	<b>575.82</b>
21	1.86e+1	8.54e+2	2.42e+2	2.67e+2	1.73e+2	250.61	10255	1826.2	1968.2	1569.5
22	8.77e+0	2.11e+2	3.72e+1	8.07e+1	7.20e+1	99.555	<b>186.7</b>	173.73	164.98	<b>22.582</b>
23	3.29e+2	3.29e+2	3.29e+2	3.29e+2	1.72e-13	<b>202.8</b>	329.46	329.46	<b>287.51</b>	59.913
24	1.08e+2	1.47e+2	1.24e+2	1.24e+2	8.77e+0	110.93	203.98	145.41	149.11	27.996
25	1.16e+2	2.03e+2	2.01e+2	1.87e+2	2.73e+1	189.42	<b>200.17</b>	<b>200.17</b>	199.36	<b>2.2838</b>
26	1.00e+2	1.01e+2	1.00e+2	1.00e+2	1.64e-1	100.06	103.01	100.72	101.1	0.9779
27	2.05e+0	4.99e+2	3.42e+2	2.60e+2	1.79e+2	184.66	<b>443.59</b>	365.21	367.15	<b>46.789</b>
28	4.55e-13	7.96e+2	4.85e+2	4.80e+2	1.10e+2	112.28	<b>439.92</b>	<b>116.21</b>	<b>231.96</b>	147.77
29	2.55e+2	3.63e+6	4.62e+2	2.50e+5	8.12e+5	3.40e+2	<b>1.89e+5</b>	5.53e+2	<b>7.90e+3</b>	<b>30768</b>
30	4.92e+2	3.02e+3	8.88e+2	1.02e+3	5.65e+2	855.07	<b>2886.2</b>	2009.7	1971.9	<b>407.67</b>

CEC 2014 test paketinin 10D boyutlu problemler için elde edilen sonuçlarının kıyaslama özeti Tablo 4.4'te sunulmuştur. Tablo verileri incelendiğinde, CMACGSA'nın GSA'ya kıyasla daha iyi, CMACPSO'nun PSO'ya kıyasla daha başarılı sonuçlar elde ettiği tespit edilmiştir. Son olarak tez kapsamında geliştirdiğimiz CMACGSA ile

CMACPSO algoritmaları kıyaslandığında CMACPSO'nun çoğu problemde CMACGSA'ya kıyasla daha üstün başarı elde ettiği görülmüştür.

**Tablo 4.4.** Algoritmaların CEC 2014 10D Kıyaslama Özeti

Boyut	Algoritmalar					
10D	GSA	CMACGSA	PSO	CMACPSO	CMACPSO	CMACGSA
En iyi	10	<b>20</b>	12	<b>18</b>	<b>27</b>	3
En kötü	10	<b>20</b>	13	<b>17</b>	13	<b>17</b>
Medyan	12	<b>18</b>	10	<b>20</b>	<b>24</b>	6
Ortalama	10	<b>20</b>	11	<b>19</b>	<b>23</b>	7
Standart	<b>15</b>	<b>15</b>	17	13	12	<b>18</b>

#### 4.5.2. CEC 2014 30D Boyutta Karşılaştırma Sonuçları

Tablo 4.5'te CMACGSA ve GSA algoritmalarının 30 boyutta (30D) elde edilen sonuçları gösterilmektedir. Tablo 4.5'te yer alan sonuçlar incelendiğinde CMACGSA'nın GSA'ya kıyasla 30 problemde en iyi değerler için 16'sında, en kötü değerler için 17'sinde, medyan ve ortalama değerler için 19'unda ve standart sapma değerleri için 18'inde üstün başarı sergilediği görülmektedir.

**Tablo 4.5.** CMACGSA ve GSA Algoritmaları için CEC 2014 30D ile Test Sonuçları

No	GSA					CMACGSA				
	En iyi	En kötü	Medyan	Ortalama	Std	En iyi	En kötü	Medyan	Ortalama	Std
1	<b>7.88e+4</b>	<b>1.38e+6</b>	1.11e+6	<b>1.02e+6</b>	<b>3.27e+5</b>	5.44e+5	3.28e+6	<b>1.02e+6</b>	1.38e+6	9.85e+5
2	<b>1.33e+4</b>	<b>3.11e+4</b>	<b>2.07e+4</b>	<b>2.11e+4</b>	<b>4.10e+3</b>	3.34e+4	8.19e+4	6.37e+4	6.04e+4	1.37e+4
3	2.91e+4	6.89e+4	5.02e+4	4.98e+4	7.63e+3	<b>2.34e+4</b>	<b>3.94e+4</b>	<b>3.14e+4</b>	<b>3.14e+4</b>	<b>5.77e+3</b>
4	<b>0</b>	<b>74.53</b>	<b>66.56</b>	<b>44.11</b>	29.98	89.284	135.11	127.63	119.07	<b>16.927</b>
5	20.83	21.03	20.94	20.94	<b>0.05</b>	<b>20.767</b>	<b>20.988</b>	<b>20.920</b>	<b>20.913</b>	0.0621
6	30.04	44.14	40.04	38.88	<b>3.78</b>	<b>15.899</b>	<b>31.108</b>	<b>27.694</b>	<b>25.034</b>	5.3301
7	<b>0.02</b>	<b>0.05</b>	<b>0.04</b>	<b>0.04</b>	<b>0.01</b>	0.0766	0.1720	0.1417	0.1396	0.0274
8	121.4	335.49	306.49	266.25	79.55	<b>119.84</b>	<b>257.91</b>	<b>248.81</b>	<b>236.12</b>	<b>39.786</b>
9	140.3	330.82	159.21	165.55	33.91	<b>114.49</b>	<b>135.57</b>	<b>130.43</b>	<b>130.34</b>	<b>6.0094</b>
10	5979.1	8371.2	7679.2	7646.6	442.76	<b>4805.7</b>	<b>5967.1</b>	<b>5306.6</b>	<b>5397.1</b>	<b>368.05</b>
11	6727.9	8388.6	7961.5	7853.2	<b>375.64</b>	<b>5773.8</b>	<b>6836.7</b>	<b>6605.0</b>	<b>6452.5</b>	379.75
12	<b>1.57</b>	2.93	2.51	2.48	0.3	1.6838	<b>2.5626</b>	<b>2.1982</b>	<b>2.2051</b>	<b>0.2639</b>
13	0.37	0.56	0.47	0.47	0.04	<b>0.2383</b>	<b>0.3537</b>	<b>0.3170</b>	<b>0.3129</b>	<b>0.0333</b>
14	<b>0.21</b>	<b>0.34</b>	<b>0.27</b>	<b>0.28</b>	<b>0.03</b>	0.3182	0.4205	0.3597	0.3634	0.0306
15	<b>1.62</b>	<b>9.88</b>	<b>2.83</b>	<b>3.07</b>	1.17	11.391	13.360	12.327	12.302	<b>0.6686</b>
16	12.46	13.38	13.01	12.97	0.18	<b>12.138</b>	<b>12.555</b>	<b>12.362</b>	<b>12.343</b>	<b>0.1347</b>
17	<b>1.28e+4</b>	<b>1.09e+5</b>	<b>5.06e+4</b>	<b>5.14e+4</b>	<b>2.61e+4</b>	4.52e+4	2.61e+5	1.57e+5	1.46e+5	6.56e+4
18	<b>225.98</b>	<b>516.18</b>	<b>303.62</b>	<b>333.87</b>	<b>79.8</b>	686.80	1609.02	978.235	1058.556	303.83
19	<b>10.28</b>	<b>16.07</b>	<b>12.48</b>	12.43	<b>1.12</b>	11.678	16.448	14.039	<b>14.047</b>	1.3651

Tablo 4.5. (devamı)

No	GSA					CMACGSA				
	En iyi	En kötü	Medyan	Ortalama	Std	En iyi	En kötü	Medyan	Ortalama	Std
20	1.04e+4	6.57e+4	2.15e+4	2.41e+4	9.88e+3	<b>7.56e+3</b>	<b>1.92e+4</b>	<b>1.16e+4</b>	<b>1.23e+4</b>	<b>3.92e+3</b>
21	2.53e+4	1.63e+5	1.03e+5	1.01e+5	3.21e+4	<b>1.30e+4</b>	<b>5.75e+4</b>	<b>2.36e+4</b>	<b>2.93e+4</b>	<b>1.46e+4</b>
22	405.69	1245.89	805.24	848.63	171.51	<b>397.61</b>	<b>868.95</b>	<b>631.18</b>	<b>620.36</b>	<b>148.57</b>
23	<b>203.23</b>	<b>204.49</b>	<b>204.04</b>	<b>204</b>	0.24	315.25	315.25	315.25	315.25	<b>0.0006</b>
24	<b>200.21</b>	<b>200.35</b>	<b>200.29</b>	<b>200.29</b>	<b>0.03</b>	223.64	238.26	224.04	225.64	4.2914
25	<b>200.05</b>	<b>200.07</b>	<b>200.06</b>	<b>200.06</b>	<b>0.01</b>	206.97	213.08	211.49	210.81	1.8621
26	106.06	<b>200.01</b>	183.29	169.03	34.12	<b>105.10</b>	200.04	<b>107.23</b>	<b>119.92</b>	<b>28.916</b>
27	865.35	1870.8	1625.7	1536.9	260.13	<b>509.17</b>	<b>566.32</b>	<b>549.47</b>	<b>543.02</b>	<b>18.755</b>
28	<b>1293.6</b>	8559.8	7152.2	6658.2	1712.3	2688.5	<b>3282.5</b>	<b>2798.3</b>	<b>2937.7</b>	<b>233.28</b>
29	2.33e+8	5.73e+8	4.95e+8	4.89e+8	6.27e+7	<b>6038.7</b>	<b>10284</b>	<b>8624.37</b>	<b>8720.35</b>	<b>1309.4</b>
30	1.48e+6	8.68e+6	4.29e+6	4.21e+6	1.45e+6	<b>1567.2</b>	<b>7184.3</b>	<b>3366.1</b>	<b>3389.7</b>	<b>1485.7</b>

Tablo 4.6’da CMACPSO ve PSO algoritmalarının 30 boyutta (30D) elde edilen sonuçları bulunmaktadır. Tabloda yer alan sonuçlar incelendiğinde, CMACPSO’nun 30 problemde en iyi değerler için 23’ünde, en kötü değerler için 20’sinde, medyan ve standart sapma değerleri için 23’ünde ve ortalama değerler için 22’sinde daha başarılı sonuçlar elde edilmiştir.

Tablo 4.6. CMACPSO ve PSO Algoritmaları için CEC 2014 30D ile Test Sonuçları

No	PSO					CMACPSO				
	En iyi	En kötü	Medyan	Ortalama	Std	En iyi	En kötü	Medyan	Ortalama	Std
1	1.18e+7	2.62e+8	5.44e+7	6.92e+7	5.45e+7	<b>9.23e+4</b>	<b>3.88e+7</b>	<b>1.80e+6</b>	<b>4.92e+6</b>	<b>8.72e+6</b>
2	1.12e+9	1.56e+10	9.40e+9	9.61e+9	3.46e+9	<b>2.96e+0</b>	<b>6.81e+8</b>	<b>2.42e+3</b>	<b>1.35e+7</b>	<b>9.54e+7</b>
3	4.48e+2	3.75e+4	8.59e+3	1.05e+4	8.02e+3	<b>5.31e-1</b>	<b>1.24e+3</b>	<b>1.16e+2</b>	<b>2.69e+2</b>	<b>3.45e+2</b>
4	9.92e+1	1.37e+3	4.55e+2	5.42e+2	3.37e+2	<b>6.34e+1</b>	<b>2.48e+2</b>	<b>8.25e+1</b>	<b>1.01e+2</b>	<b>3.94e+1</b>
5	<b>2.05e+1</b>	<b>2.10e+1</b>	<b>2.08e+1</b>	<b>2.08e+1</b>	1.08e-1	2.06e+1	2.11e+1	2.10e+1	2.09e+1	<b>1.00e-1</b>
6	1.18e+1	2.96e+1	1.95e+1	1.98e+1	3.48e+0	<b>6.35e+0</b>	<b>1.91e+1</b>	<b>1.21e+1</b>	<b>1.25e+1</b>	<b>3.24e+0</b>
7	1.59e+1	1.72e+2	8.75e+1	8.90e+1	3.39e+1	<b>3.01e-7</b>	<b>2.01e+1</b>	<b>1.97e-2</b>	<b>2.97e+0</b>	<b>5.30e+0</b>
8	<b>3.42e+1</b>	<b>1.30e+2</b>	<b>7.48e+1</b>	<b>8.04e+1</b>	<b>2.20e+1</b>	4.94e+1	1.51e+2	9.44e+1	9.69e+1	2.21e+1
9	<b>7.29e+1</b>	<b>2.09e+2</b>	<b>1.19e+2</b>	<b>1.23e+2</b>	<b>3.18e+1</b>	8.59e+1	2.26e+2	1.63e+2	1.54e+2	3.63e+1
10	1.31e+3	<b>4.08e+3</b>	2.75e+3	<b>2.69e+3</b>	<b>5.68e+2</b>	<b>1.28e+3</b>	5.13e+3	<b>2.54e+3</b>	2.75e+3	9.51e+2
11	<b>2.02e+3</b>	<b>4.83e+3</b>	<b>3.29e+3</b>	<b>3.35e+3</b>	<b>5.75e+2</b>	2.34e+3	7.30e+3	4.72e+3	4.74e+3	1.36e+3
12	<b>1.44e-1</b>	<b>2.31e+0</b>	<b>9.40e-1</b>	<b>1.06e+0</b>	6.39e-1	7.90e-1	3.35e+0	2.26e+0	2.19e+0	<b>5.45e-1</b>
13	5.16e-1	3.83e+0	8.35e-1	1.32e+0	9.84e-1	<b>3.54e-1</b>	<b>1.04e+0</b>	<b>6.16e-1</b>	<b>6.38e-1</b>	<b>1.50e-1</b>
14	1.30e+0	6.89e+1	2.37e+1	2.57e+1	1.85e+1	<b>3.06e-1</b>	<b>1.41e+0</b>	<b>1.06e+0</b>	<b>1.04e+0</b>	<b>2.16e-1</b>
15	2.20e+1	2.31e+4	6.69e+2	2.13e+3	3.78e+3	<b>2.70e+0</b>	<b>4.71e+1</b>	<b>6.07e+0</b>	<b>8.34e+0</b>	<b>8.80e+0</b>
16	<b>9.72e+0</b>	<b>1.25e+1</b>	<b>1.15e+1</b>	<b>1.15e+1</b>	5.27e-1	1.15e+1	1.32e+1	1.23e+1	1.23e+1	<b>4.13e-1</b>
17	2.57e+5	1.55e+7	1.68e+6	3.04e+6	3.44e+6	<b>5.25e+3</b>	<b>5.40e+5</b>	<b>7.14e+4</b>	<b>8.40e+4</b>	<b>8.37e+4</b>
18	6.13e+2	2.58e+8	1.21e+4	1.06e+7	5.06e+7	<b>1.52e+2</b>	<b>2.60e+4</b>	<b>4.57e+3</b>	<b>8.39e+3</b>	<b>8.60e+3</b>

Tablo 4.6. (devamı)

No	PSO					CMACPSO				
	En iyi	En kötü	Medyan	Ortalama	Std	En iyi	En kötü	Medyan	Ortalama	Std
19	8.45e+0	1.17e+2	1.58e+1	2.72e+1	2.69e+1	<b>5.93e+0</b>	<b>1.14e+2</b>	<b>1.03e+1</b>	<b>1.41e+1</b>	<b>1.83e+1</b>
20	8.70e+2	1.36e+4	3.83e+3	4.27e+3	2.83e+3	<b>1.68e+2</b>	<b>8.60e+3</b>	<b>1.06e+3</b>	<b>1.67e+3</b>	<b>1.67e+3</b>
21	4.03e+4	9.88e+5	2.68e+5	3.79e+5	2.57e+5	<b>4.06e+3</b>	<b>2.49e+5</b>	<b>3.92e+4</b>	<b>5.08e+4</b>	<b>4.62e+4</b>
22	6.26e+1	1.03e+3	4.93e+2	4.79e+2	2.36e+2	<b>4.66e+1</b>	<b>9.36e+2</b>	<b>3.83e+2</b>	<b>3.93e+2</b>	<b>1.72e+2</b>
23	3.20e+2	4.08e+2	3.45e+2	3.48e+2	1.72e+1	<b>3.15e+2</b>	<b>3.27e+2</b>	<b>3.15e+2</b>	<b>3.16e+2</b>	<b>2.41e+0</b>
24	2.32e+2	2.91e+2	2.47e+2	2.53e+2	1.59e+1	<b>2.24e+2</b>	<b>2.48e+2</b>	<b>2.36e+2</b>	<b>2.36e+2</b>	<b>6.52e+0</b>
25	2.06e+2	2.35e+2	2.13e+2	2.15e+2	6.85e+0	<b>2.03e+2</b>	<b>2.12e+2</b>	<b>2.05e+2</b>	<b>2.06e+2</b>	<b>2.24e+0</b>
26	1.01e+2	<b>1.05e+2</b>	1.03e+2	1.03e+2	<b>9.67e-1</b>	<b>1.00e+2</b>	2.00e+2	<b>1.01e+2</b>	<b>1.03e+2</b>	1.40e+1
27	4.04e+2	1.14e+3	8.29e+2	7.27e+2	2.79e+2	<b>4.01e+2</b>	<b>1.04e+3</b>	<b>6.78e+2</b>	<b>6.28e+2</b>	<b>2.28e+2</b>
28	<b>9.55e+2</b>	<b>1.94e+3</b>	<b>1.31e+3</b>	<b>1.38e+3</b>	<b>2.53e+2</b>	9.78e+2	2.76e+3	1.35e+3	1.45e+3	4.00e+2
29	3.28e+4	<b>2.32e+7</b>	2.84e+6	5.64e+6	<b>5.57e+6</b>	<b>1.29e+3</b>	2.78e+7	<b>2.13e+4</b>	<b>4.68e+6</b>	7.25e+6
30	4.03e+3	2.78e+5	4.45e+4	6.70e+4	6.86e+4	<b>1.83e+3</b>	<b>2.12e+5</b>	<b>1.07e+4</b>	<b>2.55e+4</b>	<b>3.71e+4</b>

Tablo 4.7’de CMACPSO ve CMACGSA algoritmalarının 30 boyutta (30D) elde edilen sonuçları sunulmuştur. Tablo 4.7’de yer alan sonuçlar incelendiğinde CMACPSO’nun 30 problemde en iyi değerler için 27’sinde, medyan değerler için 18’inde ve ortalama değerler için 16’sında üstün performans gösterdiği tespit edilmiştir. CMACGSA algoritması 30 problemde en kötü değerler için 22’sinde ve standart sapma değerleri için 25’inde daha başarılı sonuçlara sahiptir.

Tablo 4.7. CMACPSO ve CMACGSA Algoritmaları için CEC 2014 30D ile Test Sonuçları

No	CMACPSO					CMACGSA				
	En iyi	En kötü	Medyan	Ortalama	Std	En iyi	En kötü	Medyan	Ortalama	Std
1	<b>9.23e+4</b>	3.88e+7	1.80e+6	4.92e+6	8.72e+6	5.44e+5	<b>3.28e+6</b>	<b>1.02e+6</b>	<b>1.38e+6</b>	<b>9.85e+5</b>
2	<b>2.96e+0</b>	6.81e+8	<b>2.42e+3</b>	1.35e+7	9.54e+7	3.34e+4	<b>8.19e+4</b>	6.37e+4	<b>6.04e+4</b>	<b>1.37e+4</b>
3	<b>5.31e-1</b>	<b>1.24e+3</b>	<b>1.16e+2</b>	<b>2.69e+2</b>	<b>3.45e+2</b>	2.34e+4	3.94e+4	3.14e+4	3.14e+4	5.77e+3
4	<b>6.34e+1</b>	2.48e+2	<b>8.25e+1</b>	<b>1.01e+2</b>	3.94e+1	89.284	<b>135.11</b>	127.63	119.07	<b>16.927</b>
5	<b>2.06e+1</b>	2.11e+1	2.10e+1	<b>2.09e+1</b>	1.00e-1	20.767	<b>20.988</b>	<b>20.920</b>	20.913	<b>0.0621</b>
6	<b>6.35e+0</b>	<b>1.91e+1</b>	<b>1.21e+1</b>	<b>1.25e+1</b>	<b>3.24e+0</b>	15.899	31.108	27.694	25.034	5.3301
7	<b>3.01e-7</b>	2.01e+1	<b>1.97e-2</b>	2.97e+0	5.30e+0	0.0766	<b>0.1720</b>	0.1417	<b>0.1396</b>	<b>0.0274</b>
8	<b>4.94e+1</b>	<b>1.51e+2</b>	<b>9.44e+1</b>	<b>9.69e+1</b>	<b>2.21e+1</b>	119.84	257.91	248.81	236.12	39.786
9	<b>8.59e+1</b>	2.26e+2	1.63e+2	1.54e+2	3.63e+1	114.49	<b>135.57</b>	<b>130.43</b>	<b>130.34</b>	<b>6.0094</b>
10	<b>1.28e+3</b>	<b>5.13e+3</b>	<b>2.54e+3</b>	<b>2.75e+3</b>	9.51e+2	4805.7	5967.1	5306.6	5397.1	<b>368.05</b>
11	<b>2.34e+3</b>	7.30e+3	<b>4.72e+3</b>	<b>4.74e+3</b>	1.36e+3	5773.8	<b>6836.7</b>	6605.0	6452.5	<b>379.75</b>
12	<b>7.90e-1</b>	3.35e+0	2.26e+0	<b>2.19e+0</b>	5.45e-1	1.6838	<b>2.5626</b>	<b>2.1982</b>	2.2051	<b>0.2639</b>
13	3.54e-1	1.04e+0	6.16e-1	6.38e-1	1.50e-1	<b>0.2383</b>	<b>0.3537</b>	<b>0.3170</b>	<b>0.3129</b>	<b>0.0333</b>
14	<b>3.06e-1</b>	1.41e+0	1.06e+0	1.04e+0	2.16e-1	0.3182	<b>0.4205</b>	<b>0.3597</b>	<b>0.3634</b>	<b>0.0306</b>

Tablo 4.7. (devamı)

No	CMACPSO					CMACGSA				
	En iyi	En kötü	Medyan	Ortalama	Std	En iyi	En kötü	Medyan	Ortalama	Std
15	2.70e+0	4.71e+1	6.07e+0	8.34e+0	8.80e+0	11.391	13.360	12.327	12.302	0.6686
16	1.15e+1	1.32e+1	1.23e+1	1.23e+1	4.13e-1	12.138	12.555	12.362	12.343	0.1347
17	5.25e+3	5.40e+5	7.14e+4	8.40e+4	8.37e+4	4.52e+4	2.61e+5	1.57e+5	1.46e+5	6.56e+4
18	1.52e+2	2.60e+4	4.57e+3	8.39e+3	8.60e+3	686.80	1609.02	978.235	1058.556	303.83
19	5.93e+0	1.14e+2	1.03e+1	1.41e+1	1.83e+1	11.678	16.448	14.039	14.047	1.3651
20	1.68e+2	8.60e+3	1.06e+3	1.67e+3	1.67e+3	7.56e+3	1.92e+4	1.16e+4	1.23e+4	3.92e+3
21	4.06e+3	2.49e+5	3.92e+4	5.08e+4	4.62e+4	1.30e+4	5.75e+4	2.36e+4	2.93e+4	1.46e+4
22	4.66e+1	9.36e+2	3.83e+2	3.93e+2	1.72e+2	397.61	868.95	631.18	620.36	148.57
23	3.15e+2	3.27e+2	3.15e+2	3.16e+2	2.41e+0	315.25	315.25	315.25	315.25	0.0006
24	2.24e+2	2.48e+2	2.36e+2	2.36e+2	6.52e+0	223.64	238.26	224.04	225.64	4.2914
25	2.03e+2	2.12e+2	2.05e+2	2.06e+2	2.24e+0	206.97	213.08	211.49	210.81	1.8621
26	1.00e+2	2.00e+2	1.01e+2	1.03e+2	1.40e+1	105.10	200.04	107.23	119.92	28.916
27	4.01e+2	1.04e+3	6.78e+2	6.28e+2	2.28e+2	509.17	566.32	549.47	543.02	18.755
28	9.78e+2	2.76e+3	1.35e+3	1.45e+3	4.00e+2	2688.5	3282.5	2798.3	2937.7	233.28
29	1.29e+3	2.78e+7	2.13e+4	4.68e+6	7.25e+6	6038.7	10284	8624.37	8720.35	1309.4
30	1.83e+3	2.12e+5	1.07e+4	2.55e+4	3.71e+4	1567.2	7184.3	3366.1	3389.7	1485.7

CEC 2014 test paketinin 30D boyutlu problemler için elde edilen sonuçlarının kıyaslama özeti tablo 4.8’de sunulmuştur. Tablo verilerine göre, CMACGSA’nın GSA’ya göre daha iyi performans sergilediği gözlenmiştir. Ek olarak, CMACPSO’nun PSO’ya kıyasla daha başarılı sonuçlar elde ettiği görülmüştür. Son olarak, tez kapsamında geliştirilen CMACGSA ile CMACPSO algoritmalarının sonuçları kıyaslandığında, CMACPSO’nun 30 problemde en iyi, medyan ve ortalama değerler için CMACGSA’ya kıyasla üstün başarı elde edilmiştir.

Tablo 4.8. CMACGSA ve CMACPSO Algoritmaların CEC 2014 30D Kıyaslama Özeti

Boyut	Algoritmalar					
	GSA	CMACGSA	PSO	CMACPSO	CMACPSO	CMACGSA
30D						
En iyi	14	16	7	23	27	3
En kötü	13	17	10	20	8	22
Medyan	11	19	7	23	18	12
Ortalama	11	19	8	22	16	14
Standart	12	18	7	23	5	25

#### 4.5.3. CEC 2014 50D Boyutta Karşılaştırma Sonuçları

Tablo 4.9’da CMACGSA ve GSA algoritmalarının 50 boyutta (50D) elde edilen sonuçları sunulmuştur. Tablo 4.9’da yer alan sonuçlar incelendiğinde, CMACGSA’nın

GSA'ya kıyasla 30 problemde en kötü, medyan ve ortalama değerler için 19'unda üstün başarı sergilediği gözlenmiştir. GSA algoritması 30 problemin en iyi değerler için 18'inde CMACGSA'yı geride bırakmıştır. Standart sapma değerleri için her iki algoritma da benzer sonuçlar üretmiştir.

**Tablo 4.9.** CMACGSA ve GSA Algoritmaları için CEC 2014 50D ile Test Sonuçları

No	GSA					CMACGSA				
	En iyi	En kötü	Medyan	Ortalama	Std	En iyi	En kötü	Medyan	Ortalama	Std
1	<b>5.61e+5</b>	<b>1.61e+6</b>	<b>9.19e+5</b>	<b>9.44e+5</b>	<b>2.13e+5</b>	2.11e+6	2.77e+6	2.57e+6	2.47e+6	2.43e+5
2	<b>2.76e+4</b>	<b>4.79e+4</b>	4.01e+4	3.99e+4	<b>4.92e+3</b>	4.67e+5	7.46e+5	<b>5.20e+5</b>	<b>5.62e+5</b>	8.73e+4
3	8.05e+4	1.28e+5	1.15e+5	1.13e+5	<b>9.30e+3</b>	<b>4.70e+4</b>	<b>1.09e+5</b>	<b>8.60e+4</b>	<b>8.47e+4</b>	1.53e+4
4	<b>18.33</b>	143.3	<b>89.11</b>	<b>92.74</b>	28.03	98.574	<b>108.92</b>	102.57	103.52	<b>2.9606</b>
5	<b>21</b>	21.19	21.14	21.13	0.04	21.055	<b>21.169</b>	<b>21.118</b>	<b>21.121</b>	<b>0.0322</b>
6	68.37	76.27	73.72	73.51	<b>1.78</b>	<b>26.581</b>	<b>34.161</b>	<b>31.012</b>	<b>30.567</b>	2.5966
7	<b>0.02</b>	<b>0.04</b>	<b>0.03</b>	<b>0.03</b>	<b>0.01</b>	0.5878	0.8429	0.6274	0.6779	0.0967
8	<b>246.76</b>	650.91	611.19	552.53	134.29	460.52	<b>537.41</b>	<b>523.06</b>	<b>513.96</b>	<b>21.520</b>
9	337.3	702.06	366.16	478.8	154.38	<b>266.72</b>	<b>427.70</b>	<b>336.45</b>	<b>334.83</b>	<b>46.215</b>
10	1.31e+4	1.54e+4	1.46e+4	1.45e+4	4.84e+2	<b>1.12e+4</b>	<b>1.21e+4</b>	<b>1.16e+4</b>	<b>1.16e+4</b>	<b>291.21</b>
11	1.34e+4	1.52e+4	1.45e+4	1.45e+4	3.98e+2	<b>1.24e+4</b>	<b>1.34e+4</b>	<b>1.29e+4</b>	<b>1.29e+4</b>	<b>328.19</b>
12	<b>2.46</b>	4.01	3.46	3.37	0.37	2.6129	<b>3.3039</b>	<b>3.0971</b>	<b>3.0767</b>	<b>0.1905</b>
13	0.39	0.49	0.44	0.44	<b>0.02</b>	<b>0.3035</b>	<b>0.4053</b>	<b>0.3244</b>	<b>0.3349</b>	0.0324
14	<b>0.28</b>	0.39	0.33	0.33	<b>0.02</b>	0.2982	<b>0.3552</b>	<b>0.3293</b>	<b>0.3256</b>	0.0222
15	<b>3.75</b>	<b>8.04</b>	<b>5.95</b>	<b>5.96</b>	<b>1.03</b>	7.008	24.365	9.330	13.616	6.5679
16	22.11	22.82	22.65	22.63	<b>0.14</b>	<b>21.276</b>	<b>21.958</b>	<b>21.7093</b>	<b>21.664</b>	0.1945
17	<b>4.15e+4</b>	4.91e+5	<b>1.45e+5</b>	<b>1.86e+5</b>	1.18e+5	8.72e+4	<b>3.87e+5</b>	1.80e+5	2.16e+5	<b>9.10e+4</b>
18	<b>1035.8</b>	<b>3941.3</b>	<b>2658.4</b>	<b>2643.5</b>	<b>745.15</b>	1.65e+4	1.71e+5	4.31e+4	5.02e+4	4.20e+4
19	<b>20.2</b>	<b>30.09</b>	<b>25.15</b>	<b>25.06</b>	<b>1.97</b>	21.939	32.505	26.404	26.662	3.3169
20	1.67e+4	4.99e+4	3.05e+4	3.05e+4	8.09e+3	<b>1.30e+4</b>	<b>2.34e+4</b>	<b>1.53e+4</b>	<b>1.62e+4</b>	<b>3.51e+3</b>
21	<b>1.23e+5</b>	<b>9.92e+5</b>	<b>3.39e+5</b>	<b>3.49e+5</b>	<b>1.59e+5</b>	3.21e+5	1.57e+6	5.62e+5	7.00e+5	3.85e+5
22	<b>1231.4</b>	2482.3	1786.0	1786.5	284.82	1462.0	<b>2004.5</b>	<b>1779.4</b>	<b>1757.8</b>	<b>179.58</b>
23	<b>203.9</b>	<b>205.36</b>	<b>204.86</b>	<b>204.82</b>	0.28	344.44	345.24	344.88	344.79	<b>0.2738</b>
24	<b>200.47</b>	<b>200.83</b>	<b>200.64</b>	<b>200.64</b>	<b>0.06</b>	271.86	281.18	276.42	276.38	3.0000
25	<b>200.08</b>	<b>200.11</b>	<b>200.1</b>	<b>200.1</b>	<b>0.01</b>	228.35	238.08	234.58	234.54	3.0534
26	112.83	<b>200.04</b>	200.02	197.03	<b>13.63</b>	<b>107.79</b>	200.21	<b>129.81</b>	<b>155.21</b>	43.568
27	2517.6	3203.1	2869.3	2868.5	153.17	<b>1422.8</b>	<b>1777.35</b>	<b>1690.4</b>	<b>1657.9</b>	<b>102.36</b>
28	<b>5761.1</b>	16525.1	14575.4	13709.8	2514.6	7842.3	<b>8656.8</b>	<b>8201.1</b>	<b>8256.9</b>	<b>283.49</b>
29	8.17e+8	1.21e+9	1.03e+9	1.03e+9	8.65e+7	<b>1.78e+4</b>	<b>2.53e+4</b>	<b>1.97e+4</b>	<b>2.07e+4</b>	<b>2689.2</b>
30	1.53e+7	4.20e+7	2.96e+7	2.93e+7	5.51e+6	<b>1.36e+4</b>	<b>2.18e+4</b>	<b>1.61e+4</b>	<b>1.67e+4</b>	<b>2514.5</b>

Tablo 4.10'da CMACPSO ve PSO algoritmalarının 50 boyutta (50D) elde edilen sonuçları bulunmaktadır. Tablo 4.10'da yer alan sonuçlar incelendiğinde,

CMACPSO'nun PSO'ya kıyasla 30 problemde en iyi, en kötü, medyan, ortalama ve standart sapma değerleri için üstün başarı sergilediği görülmüştür.

**Tablo 4.10.** CMACPSO ve PSO Algoritmaları için CEC 2014 50D ile Test Sonuçları

No	PSO					CMACPSO				
	En iyi	En kötü	Medyan	Ortalama	Std	En iyi	En kötü	Medyan	Ortalama	Std
1	4.56e+7	7.67e+8	2.38e+8	2.87e+8	1.66e+8	<b>2.81e+6</b>	<b>1.41e+8</b>	<b>1.09e+7</b>	<b>1.98e+7</b>	<b>2.26e+7</b>
2	1.78e+10	5.86e+10	3.75e+10	3.84e+10	8.45e+9	<b>9.98e+4</b>	<b>6.93e+9</b>	<b>1.13e+9</b>	<b>1.51e+9</b>	<b>1.84e+9</b>
3	2.43e+4	8.49e+4	5.46e+4	5.47e+4	1.46e+4	<b>8.24e+2</b>	<b>1.05e+4</b>	<b>4.40e+3</b>	<b>4.70e+3</b>	<b>2.44e+3</b>
4	6.31e+2	8.47e+3	2.62e+3	3.02e+3	1.72e+3	<b>7.88e+1</b>	<b>1.34e+3</b>	<b>2.33e+2</b>	<b>2.72e+2</b>	<b>1.99e+2</b>
5	2.09e+1	2.12e+1	2.11e+1	2.11e+1	<b>6.22e-2</b>	<b>2.09e+1</b>	<b>2.12e+1</b>	<b>2.11e+1</b>	<b>2.11e+1</b>	7.01e-2
6	3.41e+1	5.25e+1	4.24e+1	4.31e+1	<b>4.15e+0</b>	<b>1.88e+1</b>	<b>4.26e+1</b>	<b>3.01e+1</b>	<b>3.01e+1</b>	4.61e+0
7	1.43e+2	4.88e+2	3.09e+2	3.15e+2	9.36e+1	<b>5.12e-1</b>	<b>7.58e+1</b>	<b>9.54e+0</b>	<b>1.66e+1</b>	<b>2.01e+1</b>
8	<b>1.51e+2</b>	<b>3.34e+2</b>	<b>2.24e+2</b>	<b>2.23e+2</b>	4.35e+1	1.69e+2	3.65e+2	2.45e+2	2.48e+2	<b>4.32e+1</b>
9	<b>2.07e+2</b>	<b>3.94e+2</b>	<b>2.82e+2</b>	<b>2.93e+2</b>	4.92e+1	2.82e+2	4.91e+2	3.88e+2	3.98e+2	<b>4.65e+1</b>
10	<b>3.35e+3</b>	<b>8.23e+3</b>	<b>6.23e+3</b>	<b>6.23e+3</b>	<b>1.13e+3</b>	4.56e+3	1.01e+4	7.17e+3	7.00e+3	1.45e+3
11	<b>4.91e+3</b>	<b>8.69e+3</b>	<b>6.95e+3</b>	<b>6.86e+3</b>	<b>7.62e+2</b>	6.47e+3	1.40e+4	1.12e+4	1.11e+4	1.73e+3
12	<b>2.24e-1</b>	<b>3.38e+0</b>	<b>1.28e+0</b>	<b>1.41e+0</b>	8.81e-1	1.72e+0	3.92e+0	2.92e+0	2.91e+0	<b>5.98e-1</b>
13	9.11e-1	5.01e+0	3.57e+0	3.49e+0	9.51e-1	<b>5.27e-1</b>	<b>1.06e+0</b>	<b>7.74e-1</b>	<b>7.85e-1</b>	<b>1.29e-1</b>
14	3.71e+1	1.56e+2	8.62e+1	8.37e+1	2.65e+1	<b>2.70e-1</b>	<b>2.64e+1</b>	<b>1.10e+0</b>	<b>1.66e+0</b>	<b>3.69e+0</b>
15	2.90e+3	9.80e+5	5.17e+4	1.16e+5	1.85e+5	<b>9.57e+0</b>	<b>1.20e+3</b>	<b>3.33e+1</b>	<b>7.58e+1</b>	<b>1.80e+2</b>
16	<b>1.99e+1</b>	<b>2.24e+1</b>	<b>2.14e+1</b>	<b>2.13e+1</b>	6.13e-1	2.10e+1	2.27e+1	2.20e+1	2.20e+1	<b>3.79e-1</b>
17	1.31e+6	6.42e+7	1.09e+7	1.28e+7	1.05e+7	<b>4.96e+4</b>	<b>6.33e+6</b>	<b>4.75e+5</b>	<b>8.92e+5</b>	<b>1.22e+6</b>
18	1.74e+3	1.75e+9	2.83e+8	4.04e+8	3.83e+8	<b>5.27e+2</b>	<b>7.49e+3</b>	<b>3.61e+3</b>	<b>3.64e+3</b>	<b>2.25e+3</b>
19	3.40e+1	3.62e+2	1.21e+2	1.55e+2	8.64e+1	<b>2.09e+1</b>	<b>1.37e+2</b>	<b>3.17e+1</b>	<b>4.14e+1</b>	<b>2.70e+1</b>
20	5.53e+3	6.60e+4	1.76e+4	2.49e+4	1.70e+4	<b>5.40e+2</b>	<b>9.04e+3</b>	<b>1.80e+3</b>	<b>2.25e+3</b>	<b>1.74e+3</b>
21	4.49e+5	1.67e+7	3.53e+6	4.74e+6	3.78e+6	<b>3.69e+4</b>	<b>1.04e+6</b>	<b>1.75e+5</b>	<b>2.38e+5</b>	<b>2.08e+5</b>
22	<b>1.97e+2</b>	2.05e+3	1.36e+3	1.38e+3	3.59e+2	4.13e+2	<b>1.82e+3</b>	<b>1.11e+3</b>	<b>1.10e+3</b>	<b>3.33e+2</b>
23	3.86e+2	7.21e+2	4.93e+2	5.06e+2	7.03e+1	<b>3.44e+2</b>	<b>3.71e+2</b>	<b>3.52e+2</b>	<b>3.53e+2</b>	<b>6.96e+0</b>
24	3.13e+2	4.42e+2	3.48e+2	3.54e+2	3.34e+1	<b>2.71e+2</b>	<b>3.25e+2</b>	<b>2.83e+2</b>	<b>2.86e+2</b>	<b>1.15e+1</b>
25	2.17e+2	2.75e+2	2.42e+2	2.43e+2	1.43e+1	<b>2.06e+2</b>	<b>2.31e+2</b>	<b>2.13e+2</b>	<b>2.15e+2</b>	<b>5.61e+0</b>
26	1.03e+2	4.35e+2	1.06e+2	1.78e+2	1.09e+2	<b>1.01e+2</b>	<b>3.76e+2</b>	<b>1.05e+2</b>	<b>1.49e+2</b>	<b>5.87e+1</b>
27	1.25e+3	1.86e+3	1.62e+3	1.59e+3	1.16e+2	<b>1.13e+3</b>	<b>1.62e+3</b>	<b>1.37e+3</b>	<b>1.37e+3</b>	<b>1.14e+2</b>
28	<b>1.69e+3</b>	4.23e+3	<b>2.54e+3</b>	<b>2.62e+3</b>	5.50e+2	1.73e+3	<b>3.78e+3</b>	2.74e+3	2.77e+3	<b>5.01e+2</b>
29	2.27e+6	<b>9.87e+7</b>	4.84e+7	4.92e+7	<b>2.61e+7</b>	<b>3.54e+3</b>	1.35e+8	<b>4.97e+6</b>	<b>2.75e+7</b>	3.84e+7
30	4.00e+4	<b>9.33e+5</b>	2.05e+5	2.54e+5	<b>1.92e+5</b>	<b>1.25e+4</b>	1.46e+6	<b>6.67e+4</b>	<b>1.08e+5</b>	2.06e+5

Tablo 4.11'de CMACPSO ve CMACGSA algoritmalarının 50 boyutta (50D) elde edilen sonuçları sunulmuştur. Tablo 4.11'de yer alan sonuçlar incelendiğinde, CMACPSO'nun 30 problemde en iyi değerler için 26'sında, ortalama değerler için 16'sında üstün performans sergilediği gözlemlenmiştir. CMACGSA'nın 30 problemde en

kötü değerler için 20'sinde ve standart sapma değerleri için 26'sında başarılı sonuçlara sahiptir. Medyan değerler için her iki algoritma da benzer sonuçlar üretmiştir.

**Tablo 4.11.** CMACPSO ve CMACGSA Algoritmaları için CEC 2014 50D ile Test Sonuçları

No	CMACPSO					CMACGSA				
	En iyi	En kötü	Medyan	Ortalama	Std	En iyi	En kötü	Medyan	Ortalama	Std
1	2.81e+6	1.41e+8	1.09e+7	1.98e+7	2.26e+7	<b>2.11e+6</b>	<b>2.77e+6</b>	<b>2.57e+6</b>	<b>2.47e+6</b>	<b>2.43e+5</b>
2	<b>9.98e+4</b>	6.93e+9	1.13e+9	1.51e+9	1.84e+9	4.67e+5	<b>7.46e+5</b>	<b>5.20e+5</b>	<b>5.62e+5</b>	<b>8.73e+4</b>
3	<b>8.24e+2</b>	<b>1.05e+4</b>	<b>4.40e+3</b>	<b>4.70e+3</b>	<b>2.44e+3</b>	4.70e+4	1.09e+5	8.60e+4	8.47e+4	1.53e+4
4	<b>7.88e+1</b>	1.34e+3	2.33e+2	2.72e+2	1.99e+2	98.574	<b>108.92</b>	<b>102.57</b>	<b>103.52</b>	<b>2.9606</b>
5	<b>2.09e+1</b>	2.12e+1	<b>2.11e+1</b>	<b>2.11e+1</b>	7.01e-2	21.055	<b>21.169</b>	21.118	21.121	<b>0.0322</b>
6	<b>1.88e+1</b>	4.26e+1	<b>3.01e+1</b>	<b>3.01e+1</b>	4.61e+0	26.581	<b>34.161</b>	31.012	30.567	<b>2.5966</b>
7	<b>5.12e-1</b>	7.58e+1	9.54e+0	1.66e+1	2.01e+1	0.5878	<b>0.8429</b>	<b>0.6274</b>	<b>0.6779</b>	<b>0.0967</b>
8	<b>1.69e+2</b>	<b>3.65e+2</b>	<b>2.45e+2</b>	<b>2.48e+2</b>	4.32e+1	460.52	537.41	523.06	513.96	<b>21.520</b>
9	2.82e+2	4.91e+2	3.88e+2	3.98e+2	4.65e+1	<b>266.72</b>	<b>427.70</b>	<b>336.45</b>	<b>334.83</b>	<b>46.215</b>
10	<b>4.56e+3</b>	<b>1.01e+4</b>	<b>7.17e+3</b>	<b>7.00e+3</b>	1.45e+3	1.12e+4	1.21e+4	1.16e+4	1.16e+4	<b>291.21</b>
11	<b>6.47e+3</b>	1.40e+4	<b>1.12e+4</b>	<b>1.11e+4</b>	1.73e+3	1.24e+4	<b>1.34e+4</b>	1.29e+4	1.29e+4	<b>328.19</b>
12	<b>1.72e+0</b>	3.92e+0	<b>2.92e+0</b>	<b>2.91e+0</b>	5.98e-1	2.6129	<b>3.3039</b>	3.0971	<b>3.0767</b>	<b>0.1905</b>
13	5.27e-1	1.06e+0	7.74e-1	7.85e-1	1.29e-1	<b>0.3035</b>	<b>0.4053</b>	<b>0.3244</b>	<b>0.3349</b>	<b>0.0324</b>
14	<b>2.70e-1</b>	2.64e+1	1.10e+0	1.66e+0	3.69e+0	0.2982	<b>0.3552</b>	<b>0.3293</b>	<b>0.3256</b>	<b>0.0222</b>
15	9.57e+0	1.20e+3	3.33e+1	7.58e+1	1.80e+2	<b>7.008</b>	<b>24.365</b>	<b>9.330</b>	<b>13.616</b>	<b>6.5679</b>
16	<b>2.10e+1</b>	2.27e+1	2.20e+1	2.20e+1	3.79e-1	21.276	<b>21.958</b>	<b>21.7093</b>	<b>21.664</b>	<b>0.1945</b>
17	<b>4.96e+4</b>	6.33e+6	4.75e+5	8.92e+5	1.22e+6	8.72e+4	<b>3.87e+5</b>	<b>1.80e+5</b>	<b>2.16e+5</b>	<b>9.10e+4</b>
18	<b>5.27e+2</b>	<b>7.49e+3</b>	<b>3.61e+3</b>	<b>3.64e+3</b>	<b>2.25e+3</b>	1.65e+4	1.71e+5	4.31e+4	5.02e+4	4.20e+4
19	<b>2.09e+1</b>	1.37e+2	3.17e+1	4.14e+1	2.70e+1	21.939	<b>32.505</b>	<b>26.404</b>	<b>26.662</b>	<b>3.3169</b>
20	<b>5.40e+2</b>	<b>9.04e+3</b>	<b>1.80e+3</b>	<b>2.25e+3</b>	<b>1.74e+3</b>	1.30e+4	2.34e+4	1.53e+4	1.62e+4	3.51e+3
21	<b>3.69e+4</b>	<b>1.04e+6</b>	<b>1.75e+5</b>	<b>2.38e+5</b>	<b>2.08e+5</b>	3.21e+5	1.57e+6	5.62e+5	7.00e+5	3.85e+5
22	<b>4.13e+2</b>	<b>1.82e+3</b>	<b>1.11e+3</b>	<b>1.10e+3</b>	3.33e+2	1462.0	2004.5	1779.4	1757.8	<b>179.58</b>
23	<b>3.44e+2</b>	3.71e+2	3.52e+2	3.53e+2	6.96e+0	344.44	<b>345.24</b>	<b>344.88</b>	<b>344.79</b>	<b>0.2738</b>
24	<b>2.71e+2</b>	3.25e+2	2.83e+2	2.86e+2	1.15e+1	271.86	<b>281.18</b>	<b>276.42</b>	<b>276.38</b>	<b>3.0000</b>
25	<b>2.06e+2</b>	<b>2.31e+2</b>	<b>2.13e+2</b>	<b>2.15e+2</b>	5.61e+0	228.35	238.08	234.58	234.54	<b>3.0534</b>
26	<b>1.01e+2</b>	3.76e+2	<b>1.05e+2</b>	<b>1.49e+2</b>	5.87e+1	107.79	<b>200.21</b>	129.81	155.21	<b>43.568</b>
27	<b>1.13e+3</b>	<b>1.62e+3</b>	<b>1.37e+3</b>	<b>1.37e+3</b>	1.14e+2	1422.8	1777.35	1690.4	1657.9	<b>102.36</b>
28	<b>1.73e+3</b>	<b>3.78e+3</b>	<b>2.74e+3</b>	<b>2.77e+3</b>	5.01e+2	7842.3	8656.8	8201.1	8256.9	<b>283.49</b>
29	<b>3.54e+3</b>	1.35e+8	4.97e+6	2.75e+7	3.84e+7	1.78e+4	<b>2.53e+4</b>	<b>1.97e+4</b>	<b>2.07e+4</b>	<b>2689.2</b>
30	<b>1.25e+4</b>	1.46e+6	6.67e+4	1.08e+5	2.06e+5	1.36e+4	<b>2.18e+4</b>	<b>1.61e+4</b>	<b>1.67e+4</b>	<b>2514.5</b>

CEC 2014 test paketinin 50D boyutlu problemler için elde edilen sonuçlarının kıyaslama özeti Tablo 4.12'de sunulmuştur. Tablo verilerine göre, CMACGSA'nın GSA'ya göre daha iyi, CMACPSO'nun PSO'ya kıyasla daha başarılı sonuçlar elde ettiği görülmektedir. Ek olarak tez kapsamında geliştirdiğimiz CMACGSA ile CMACPSO

algoritmaları kıyaslandığında CMACPSO'nun 30 problemde en ve ortalama değerler için CMACGSA'ya kıyasla üstün başarı elde etmiştir.

**Tablo 4.12.** Algoritmaların CEC 2014 50D Kıyaslama Özeti

Boyut	Algoritmalar					
50D	GSA	CMACGSA	PSO	CMACPSO	CMACPSO	CMACGSA
En iyi	<b>18</b>	12	8	<b>22</b>	<b>26</b>	3
En kötü	13	<b>19</b>	8	<b>22</b>	8	<b>20</b>
Medyan	11	<b>19</b>	7	<b>23</b>	<b>15</b>	<b>15</b>
Ortalama	11	<b>19</b>	7	<b>23</b>	<b>16</b>	14
Standart	<b>15</b>	<b>15</b>	6	<b>24</b>	6	<b>26</b>

#### 4.5.4. CMACGSA ve CMACPSO Algoritmalarının SOTA Algoritmaları İle Karşılaştırılması

CMACGSA ve CMACPSO algoritmalarının performanslarını değerlendirmek için WOA(Mirjalili & Lewis, 2016), COA(Yang & Deb, 2009), DA(Mirjalili, 2016a) , SCA(Mirjalili, 2016b) gibi SOTA algoritmaları seçilmiş ve CEC 2014 test paketinde bulunan 30 problem için 30 boyutta (30D) elde edilen sonuçlar ile karşılaştırılmıştır. Karşılaştırma sırasında SOTA algoritmalarının parametre değerleri orijinal makalelerde verilen değerlerle aynı olmasına dikkat edilmiştir. SOTA algoritmaların parametreleri Tablo 4.13'te sunulmuştur. Bu algoritmaların seçilmesinin sebebi CEC 2014 test paketinde 30 boyutta test edilmiş olmasıdır.

**Tablo 4.13.** SOTA Algoritmaların Parametreleri

Algoritmalar	Parametreler
Gri Kurt Optimizasyon Algoritması (WOA)	$b = 1$
Guguk Kuşu Optimizasyonu (COA)	Guguk sayısı=10
Yusuçuk Algoritması (DA)	$w_{min} = 0,4$ $w_{max} = 0,9$
Sinüs Kosinüs Algoritması (SCA)	$a = 2$

Yukarıdaki algoritmaların karşılaştırma sonuçları Tablo 4.14'te sunulmuştur. Tabloda her fonksiyonun alt satırında ortalama değerlerine göre sıralamaları yer almaktadır. Tablonun altındaki satırda 30 test fonksiyonu için her bir algoritmanın sıralamalarının ortalaması yer almaktadır. Ortalama sonuç değerlerinin sıralaması tablonun en alt satırında bulunmaktadır. Tablo sonuçlarına göre, CMACPSO'nun birinci, CMACGSA'nın ikinci sırada olduğu tespit edilmiştir.

**Tablo 4.14.** CMACGSA ve CMACPSO'nun diğer Metasezgisel Algoritmaların CEC 2014 30D ile Test Sonuçları

		WOA	COA	DA	SCA	CMACGSA	CMACPSO
<b>F1</b>	Ortalama	3.24E+07	1.03E+08	4.49E+08	2.24E+08	1.38E+06	4.92E+06
	Sıralama	3	4	6	5	1	2
<b>F2</b>	Ortalama	5.05E+06	2.06E+09	3.67E+10	1.60E+10	60428.06	1.35E+07
	Sıralama	2	4	6	5	1	3
<b>F3</b>	Ortalama	2.97E+04	1.78E+05	1.81E+05	3.79E+04	31412.4	2.69E+02
	Sıralama	2	5	6	4	3	1
<b>F4</b>	Ortalama	176	398	4860	1020	119.07	1.01E+02
	Sıralama	3	4	6	5	2	1
<b>F5</b>	Ortalama	20.4	488	20.8	20.9	20.91	2.09E+01
	Sıralama	1	6	2	3	4	3
<b>F6</b>	Ortalama	34.8	437	37.2	34.4	25.03	1.25E+01
	Sıralama	4	6	5	3	2	1
<b>F7</b>	Ortalama	1.03	701	354	133	0.14	2.97E+00
	Sıralama	2	6	5	4	1	3
<b>F8</b>	Ortalama	176	421	289	236	236.12	9.69E+01
	Sıralama	2	6	5	3	4	1
<b>F9</b>	Ortalama	213	902	301	267	130.34	1.54E+02
	Sıralama	3	6	5	4	1	2
<b>F10</b>	Ortalama	3770	1870	6570	5880	5397.06	2.75E+03
	Sıralama	3	1	6	5	4	2
<b>F11</b>	Ortalama	4500	4980	6920	7040	6452.47	4.74E+03
	Sıralama	1	3	5	6	4	2
<b>F12</b>	Ortalama	1.67	1210	2.21	2.45	2.21	2.19E+00
	Sıralama	1	5	3	4	3	2
<b>F13</b>	Ortalama	0.5	1310	5.14	2.89	0.31	6.38E-01
	Sıralama	2	6	5	4	1	3
<b>F14</b>	Ortalama	0.28	1410	131	41.1	0.36	1.04E+00
	Sıralama	1	6	5	4	2	3
<b>F15</b>	Ortalama	70	3.84E+04	2.85E+05	2820	12.3	8.34E+00
	Sıralama	3	5	6	4	2	1
<b>F16</b>	Ortalama	1.26E+01	1610	1.32E+01	1.28E+01	12.34	1.23E+01
	Sıralama	3	6	5	4	2	1
<b>F17</b>	Ortalama	4.33E+06	6.41E+06	1.61E+07	6.61E+06	146397.3	8.40E+04
	Sıralama	3	4	6	5	2	1
<b>F18</b>	Ortalama	1.55E+04	2.18E+08	6.11E+08	1.85E+08	1058.56	8.39E+03
	Sıralama	3	5	6	4	1	2
<b>F19</b>	Ortalama	44.8	2540	268	90.8	14.05	1.41E+01
	Sıralama	3	6	5	4	1	2
<b>F20</b>	Ortalama	2.04E+04	8.81E+04	2.78E+05	1.31E+04	12315.21	1.67E+03
	Sıralama	4	5	6	3	2	1
<b>F21</b>	Ortalama	9.48E+05	2.51E+05	8.23E+06	1.48E+06	29317.72	5.08E+04
	Sıralama	4	3	6	5	1	2
<b>F22</b>	Ortalama	749	8.21E+04	1.12E+03	754	620.36	3.93E+02
	Sıralama	3	6	5	4	2	1
<b>F23</b>	Ortalama	331	2630	576	370	315.25	3.16E+02
	Sıralama	3	6	5	4	1	2
<b>F24</b>	Ortalama	206	2410	281	201	225.64	2.36E+02
	Sıralama	2	6	5	1	3	4
<b>F25</b>	Ortalama	225	5290	240	227	210.81	2.06E+02
	Sıralama						

Tablo 4.14. (devamı)

		WOA	COA	DA	SCA	CMACGSA	CMACPSO
	Sıralama	3	6	5	4	2	1
F26	Ortalama	100	2620	104	102	119.92	1.03E+02
	Sıralama	1	6	4	2	5	3
F27	Ortalama	905	2720	1090	722	543.02	6.28E+02
	Sıralama	4	6	5	3	1	2
F28	Ortalama	2150	2820	2240	2000	2937.73	1.45E+03
	Sıralama	3	5	4	2	6	1
F29	Ortalama	4.38E+06	1.26E+07	2.15E+07	1.34E+07	8720.35	4.68E+06
	Sıralama	2	4	6	5	1	3
F30	Ortalama	8.20E+04	3.88E+05	4.79E+05	2.48E+05	3389.66	2.55E+04
	Sıralama	3	5	6	4	1	2
Sıralamaların Ortalaması		2.56	5.06	5.16	3.9	2.2	1.93
Genel Sıralama		3	5	6	4	2	1

#### 4.5.5. CMACGSA'nın GSA Varyantları ile Karşılaştırılması

Literatürde sunulan GSA'nın varyantları, CEC 2014 30 boyutta (30D) test sonuçları ile CMACGSA'nın test sonuçları kıyaslanmıştır. Bu varyantlar, SCGSA(Jiang, Jiang, vd., 2020), BA-CGSA(Jiang, Yang, vd., 2020), GEA(Neamah vd., 2022), GG-GSA (Bohat & Arya, 2018) olarak belirlenmiştir. Karşılaştırma algoritmalarına ait parametre değerleri Tablo 4.15'te sunulmuştur. Tablo 4.16'da CEC 2014 30 boyutta elde edilen ortalama değerleri yer almaktadır. Her fonksiyon için satır sonunda kıyaslama sonuçlarının sıralaması belirlenmiştir. Tablonun en alt satırında kıyaslama sıralamaları ortalama değerlerine göre sıralanmıştır. Sıralamalar incelendiğinde, CMACGSA'nın GSA'nın varyantlarına kıyasla daha başarılı sonuçlar elde ettiği tespit edilmiştir.

Tablo 4.15. GSA'nın Varyantlarının Parametreleri

Algoritmalar	Parametreler
Sinüs Kaotik Yerçekimi Arama Algoritması (SCGSA)	$\alpha = 20, G_0 = 100, \varepsilon = 1e - 6, k = 2 * (1 - \frac{t}{T})$
Denge Ayarlama Tabanlı Kaotik Yerçekimi Arama Algoritması (BA-CGSA)	$\alpha = 20, G_0 = 100, \varepsilon = 1e - 6, k = c - c * \frac{t}{T}$
Yerçekimsel evrim algoritması (GEA)	$\alpha = 20, G_0 = 100, \varepsilon = 1e - 6, pCR = 0.5, \beta_0 = 0.95, \lambda = 0.01$
Rehberli Yerçekimsel Arama Algoritması (GG-GSA)	$\alpha = 20, G_0 = 1, \varepsilon = 1e - 6, \dot{C}_1 = (-2t^3 / T^3) + 2, \dot{C}_2 = (-2t^2 / T^3)$

Tablo 4.16. CMACGSA Ve GSA'nın Varyantlarının CEC 2014 30D ile Test Sonuçları

		SCGSA	BA-CGSA	GEA	GG-GSA	CMACGSA
F1	Ortalama	4.74e+6	4.47e+6	6.68e+5	4.49e+5	1.38e+6
	Sıralama	4	4	2	1	3
F2	Ortalama	2.57e+3	8.22e+3	1.31E-03	1.15e+4	6.04e+4
	Sıralama	2	4	1	3	5
F3	Ortalama	1.26e+5	4.17e+4	2.52e+1	4.10e+3	3.14e+4
	Sıralama	7	5	2	3	4

Tablo 4.16. (devamı)

		SCGSA	BA-CGSA	GEA	GG-GSA	CMACGSA
F4	Ortalama	5.36e+2	4.91e+2	1.10e+2	4.63e+2	1.19e+2
	Sıralama	5	4	1	3	2
F5	Ortalama	5.20e+2	5.20e+2	2.00e+1	5.20e+2	2.09e+1
	Sıralama	3	3	1	3	2
F6	Ortalama	6.20e+2	6.16e+2	1.41e+6	6.09e+2	2.50e+1
	Sıralama	4	3	5	2	1
F7	Ortalama	7.00e+2	7.00e+2	6.40E-03	7.00e+2	1.40E-01
	Sıralama	3	3	1	3	2
F8	Ortalama	9.51e+2	9.03e+2	9.50e+6	8.73e+2	2.36e+2
	Sıralama	4	3	5	2	1
F9	Ortalama	1.04e+3	9.99e+2	1.04e+2	978.00	130.34
	Sıralama	5	4	1	3	2
F10	Ortalama	6.59e+3	4.16e+3	2.83e+6	3.30e+3	5.40e+3
	Sıralama	4	2	5	1	3
F11	Ortalama	7.26e+3	4.80e+3	3.33e+3	3.43e+3	6.45e+3
	Sıralama	5	3	1	2	4
F12	Ortalama	1.20e+3	1.20e+3	1.51E-02	1.20e+3	2.21e+0
	Sıralama	3	3	1	3	2
F13	Ortalama	1.30e+3	1.30e+3	4.47E-01	1.30e+3	3.13E-01
	Sıralama	4	3	1	3	2
F14	Ortalama	1.40e+3	1.40e+3	2.85E-01	1.40e+3	3.63E-01
	Sıralama	4	3	1	3	2
F15	Ortalama	1.52e+3	1.51e+3	3.77e+0	1.50e+3	1.23e+1
	Sıralama	5	3	1	4	2
F16	Ortalama	1.62e+3	1.61e+3	1.26e+6	1.61e+3	1.23e+1
	Sıralama	4	3	5	2	1
F17	Ortalama	1.28e+6	2.24e+5	2.14e+5	4.37e+4	1.46e+5
	Sıralama	5	4	3	1	2
F18	Ortalama	2.96e+3	2.34e+3	4.35e+4	2.32e+3	1.06e+3
	Sıralama	4	3	5	2	1
F19	Ortalama	1922.90	1.92e+3	2.71e+1	1.92e+3	1.40e+1
	Sıralama	5	3	2	4	1
F20	Ortalama	4.21e+4	2.72e+4	5.66e+6	1.41e+4	1.23e+4
	Sıralama	4	3	5	2	1
F21	Ortalama	1.32e+6	1.68e+5	3.68e+4	3.78e+4	2.93e+4
	Sıralama	5	4	1	2	3
F22	Ortalama	3.55e+3	3.21e+3	7.24e+6	3.10e+3	6.20e+2
	Sıralama	4	2	5	3	1
F23	Ortalama	2.65e+3	2.62e+3	3.16e+2	2.62e+3	3.15e+2
	Sıralama	5	4	1	3	2
F24	Ortalama	2.66e+3	2.61e+3	2.30e+6	2.60e+3	2.26e+2
	Sıralama	4	2	5	3	1
F25	Ortalama	2.70e+3	2.70e+3	2.16e+2	2.70e+3	2.11e+2
	Sıralama	3	3	2	3	1
F26	Ortalama	2.80e+3	2.79e+3	1.65e+6	2.80e+3	1.20e+2
	Sıralama	4	2	5	3	1
F27	Ortalama	3.73e+3	3.72e+3	6.54e+2	3.33e+3	5.43e+2
	Sıralama	5	4	2	3	1
F28	Ortalama	8.37e+3	6.01e+3	2.39e+6	5.45e+3	2.94e+3
	Sıralama	4	2	5	3	1
F29	Ortalama	1.18e+4	6.53e+3	5.15e+6	4.08e+3	8.72e+3
	Sıralama	4	2	5	1	3
F30	Ortalama	5.76e+4	1.14e+4	23354.46	5.24e+3	3.39e+3
	Sıralama	5	3	4	1	2
<b>Sıralamaların Ortalaması</b>		<b>5.96</b>	<b>4.76</b>	<b>4.13</b>	<b>4.06</b>	<b>3.26</b>
<b>Genel Sıralama</b>		<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>

#### 4.5.6. CMACPSO'nun PSO Varyantları İle Karşılaştırılması

Literatürde sunulan PSO'nun varyantları CEC 2014 test sonuçları ile CMACPSO karşılaştırılmıştır. Karşılaştırma algoritmaları ALCPSO(W.-N. Chen vd., 2012), CGPSO(Sun vd., 2010), PSOFIPS(Mendes vd., 2004), PSOFDR (Peram vd., 2003) olarak belirlenmiştir. Algoritmalara ait parametreler Tablo 4.17'de sunulmuştur. Kıyaslama sonuçları CEC 2014 30 boyutta elde edilen ortalama değerleri Tablo 4.18'de yer almaktadır. Tabloda her fonksiyon için satır sonunda kıyaslama sonuçlarının sıralaması bulunmaktadır. Tablonun en alt satırında kıyaslamalar ortalama değerlerine göre sıralanmıştır. Sıralamalar incelendiğinde, CMACPSO algoritmasının PSO'nun varyantlarına kıyasla ilk sırada olduğu görülmektedir.

**Tablo 4.17.** PSO Algoritmasının Varyantlarının Parametreleri

Algoritmalar	Parametreler
ALCPSO	$w = 0,4, c_1 = c_2 = 2, M = 20, \theta_0 = 60$
CGPSO	$sz = 20, G = 150, P_D = 0.5, a_0 = 0.12, a_1 = 0.3, a_2 = 0.4$
PSOFIPS	-
PSOFDR	$\psi_1 = 1, \psi_2 = 1, \psi_3 = 2$

**Tablo 4.18.** CMACPSO ve PSO'nun Varyantlarının CEC 2014 30D ile Test Sonuçları

		ALCPSO	CGPSO	PSOFIPS	PSOFDR	CMAC-PSO
F1	Ortalama	4.34e+6	9.35e+6	1.02e+7	2.78e+6	4.92e+6
	Sıralama	2	4	5	1	3
F2	Ortalama	2.10e+3	1.59e+8	1.13e+4	1.53e+8	1.35e+7
	Sıralama	1	5	2	4	3
F3	Ortalama	3.91e+2	2.29e+3	6.94e+3	5.40e+2	2.69e+2
	Sıralama	2	4	5	3	1
F4	Ortalama	1.22e+2	6.91e+1	1.27e+2	9.50e+1	1.01e+2
	Sıralama	4	1	5	2	3
F5	Ortalama	2.08e+1	2.10e+1	2.10e+1	2.09e+1	2.09e+1
	Sıralama	1	4	4	3	2
F6	Ortalama	1.69e+1	2.48e+1	6.19e+0	7.67e+0	1.25e+1
	Sıralama	4	5	1	2	3
F7	Ortalama	1.00e-2	2.40e+0	2.56e-3	1.33e+1	2.97e+0
	Sıralama	2	3	1	5	4
F8	Ortalama	2.42e+1	8.79e+1	6.60e+1	1.35e+2	9.69e+1
	Sıralama	1	3	2	5	4
F9	Ortalama	9.93e+1	2.23e+2	1.53e+3	1.55e+2	1.54e+2
	Sıralama	1	4	5	3	2
F10	Ortalama	1.54e+3	5.57e+3	2.87e+3	1.97e+3	2.75e+3
	Sıralama	1	5	4	2	3
F11	Ortalama	4.14e+3	5.80e+3	5.76e+3	3.09e+3	4.74e+3
	Sıralama	2	5	4	1	3
F12	Ortalama	1.89e+0	2.78e+0	2.62e+0	8.12e-1	2.19e+0
	Sıralama	2	5	4	1	3
F13	Ortalama	1.56e+0	1.66e+0	3.46e-1	4.35e-1	6.38e-1
	Sıralama	4	5	1	2	3
F14	Ortalama	1.10e+0	8.90e-1	3.07e-1	8.91e-1	1.04e+0
	Sıralama	5	2	1	3	4
F15	Ortalama	1.10e+1	1.80e+1	1.58e+1	4.49e+0	8.34e+0
	Sıralama	3	5	4	1	2
F16	Ortalama	1.25e+1	1.27e+1	1.18e+1	1.26e+1	1.23e+1
	Sıralama	3	5	1	4	2

Tablo 4.18. (devamı)

		ALCPSO	CGPSO	PSOFIPS	PSOFDR	CMAC-PSO
F17	Ortalama	5.53e+5	3.61e+5	3.76e+5	1.36e+5	8.40e+4
	Sıralama	5	3	4	2	1
F18	Ortalama	8.34e+3	2.44e+6	1.59e+3	3.38e+3	8.39e+3
	Sıralama	3	5	1	2	4
F19	Ortalama	1.80e+1	1.70e+1	1.42e+1	1.57e+1	1.41e+1
	Sıralama	5	4	2	3	1
F20	Ortalama	1.11e+3	4.55e+2	5.81e+3	7.05e+3	1.67e+3
	Sıralama	2	1	4	5	3
F21	Ortalama	7.94e+4	1.36e+5	1.49e+5	6.77e+4	5.08e+4
	Sıralama	3	4	5	2	1
F22	Ortalama	4.82e+2	7.19e+2	2.25e+2	2.27e+2	3.93e+2
	Sıralama	4	5	1	2	3
F23	Ortalama	3.15e+2	2.00e+2	3.17e+2	3.18e+2	3.16e+2
	Sıralama	2	1	4	5	3
F24	Ortalama	2.38e+2	2.00e+2	2.24e+2	2.27e+2	2.36e+2
	Sıralama	5	1	2	3	4
F25	Ortalama	2.11e+2	2.00e+2	2.07e+2	2.06e+2	2.06e+2
	Sıralama	5	1	4	3	2
F26	Ortalama	1.47e+2	1.97e+2	1.70e+2	1.51e+2	1.03e+2
	Sıralama	2	5	4	3	1
F27	Ortalama	6.99e+2	3.14e+2	4.34e+2	7.12e+2	6.28e+2
	Sıralama	4	1	2	5	3
F28	Ortalama	1.75e+3	2.00e+2	3.98e+2	1.48e+3	1.45e+3
	Sıralama	5	1	2	4	3
F29	Ortalama	3.72e+6	5.97e+3	2.14e+2	6.65e+6	4.68e+6
	Sıralama	3	2	1	5	4
F30	Ortalama	1.41e+4	1.11e+4	7.00e+2	1.82e+4	2.55e+4
	Sıralama	3	2	1	4	5
<b>Sıralamaların Ortalaması</b>		<b>2.96</b>	<b>3.36</b>	<b>2.86</b>	<b>3</b>	<b>2.76</b>
<b>Genel Sıralama</b>		<b>3</b>	<b>5</b>	<b>2</b>	<b>4</b>	<b>1</b>

#### 4.6. Mühendislik Tasarım Problemleri

Mühendislik tasarım problemleri çok boyutlu, karmaşık ve kısıtları olan yapıya sahiptir. Bu problemler literatürde algoritmaların geçerliliklerini test etmek, güçlü ve zayıf yönlerini anlamak için sıklıkla kullanılmaktadır. Problemlerde verilen parametreleri kullanarak amaç fonksiyonu için optimum sonuca ulaşılması hedeflenmektedir. Amaç fonksiyonunun, problem türüne göre minimize ya da maksimize olması beklenmektedir. Problemlerin parametreleri kendine özgü olup, parametrelerin kısıtları bulunmaktadır. Tezin bu bölümünde, geliştirilen CMACGSA ve CMACPSO algoritmalarının gerçek dünya problemlerindeki performansını değerlendirmek için mevcut mühendislik tasarım problemlerinden kaynaklı giriş tasarımı, basınçlı kap tasarımı ve üç çubuklu makas tasarımı problemleri seçilmiş ve literatürde yer alan diğer metasezgisel algoritmalarla karşılaştırılmıştır. Bu algoritmalar ABC, BBO, DE, GSA, GWO, PSO, SCA ve SSA olarak seçilmiştir. Her algoritma için popülasyon sayısı 30 olarak belirlenmiştir. Mühendislik problemlerinde kullanılan algoritmaların yapıları gereği rastgelelik içerdiklerinden dolayı daha güvenilir ve istatistiksel açıdan anlamlı sonuçlar elde etmek için

çalıştırılma sayıları 10 olarak belirlenmiştir. Tablo 4.19’da karşılaştırılan algoritmaların parametreleri sunulmuştur.

**Tablo 4.19.** WBD Problemi için Karşılaştırılan Popüler Metasezgisel Algoritmaların Parametreleri

Algoritmalar	Parametreler
Yapay Arı Koloni (ABC)	deneme sayısı = $0.6 * D * Pop$
Biyocoğrafya Tabanlı Optimizasyon (BBO)	$a = 0,9, p_m = 0.1, sigma = 0.02 * (U_b - U_l)$
Diferansiyel Evrim (DE)	$\beta_{min} = 0.2, \beta_{max} = 0.8, Pcr = 0.2$
Genetik Algoritma (GA)	$p_c = 0.95, p_m = 0.1$
Yerçekimi Arama Algoritması (GSA)	$a = 20, G_0 = 100, \varepsilon = 1e - 6$
Gri Kurt Optimizasyonu(GWO)	$a = 2 - t(2/T)$
Parçacık Sürü Optimizasyonu (PSO)	$c_1 = 2, c_2 = 2$
Sinüs Kosinüs Algoritması (SCA)	$a = 2$
Salp Sürü Algoritması (SSA)	$c_1 = 2 \exp(-(4t/T)^2)$

#### 4.6.1. Kaynaklı Kiriş Tasarımı

Kaynaklı kiriş tasarımı problemi, yapı mühendisliğinin de sıklıkla kullanılan kaynak kirişlerinin maliyetini en aza indirmeyi amaçlayan tasarım problemlerinden biridir (Zou vd., 2011). Kaynaklama işlemi metalik parçaların ısıtılarak, basınçlı veya basınçsız birleştirilme işlemidir. Bu problem optimizasyon algoritmalarının performanslarını değerlendirmek için çok kez kullanılmıştır. Problemin dört tasarım değişkeni vardır. Bunlar kiriş kalınlığı ( $x_1$ ), yüksekliği ( $x_2$ ), kaynak kalınlığı ( $x_3$ ) ve uzunluğu ( $x_4$ ) olarak ifade edilmektedir. Değişkenler sabit aralıkta ve pozitif sayılardır. Probleme ait matematiksel ifadeler Denklem 4.1-4.11’de yer almaktadır. Kaynaklı kiriş tasarımının çizimi Şekil 4.6’da sunulmuştur.

Probleme ait formüllerde  $\Phi$  amaç fonksiyonu, M kirişin eğilme momentini, R nötr eksene olan uzaklığı, J kutup atalet momentini,  $\sigma$  eğilme gerilimini,  $\Delta$  uç sapmasını,  $P_c$  burkulma yükünü,  $\tau_p$  ve  $\tau_{pp}$  kesme geriliminin 1. ve 2. bileşenlerini ve  $\tau$  toplam kesme gerilimini ifade etmektedir. Probleme ait parametreler Tablo 4.20’de gösterilmiştir.

**Tablo 4.20.** WBD Probleminin Parametreleri

Açıklama	Parametre
Uygulanan Uç yükü	$P = 6000lb$
Young’ın modülü	$E = 30 \times 10^6 psi$
Kiriş için kesme modülü	$G = 12 \times 10^6 psi$
Konsol uzunluğu	$L = 14inch$
Kaynak tasarım stresi	$\tau = 13600psi$
Normal tasarım stresi	$\sigma = 3000psi$

$$\sigma = \left( \frac{\Gamma(1+\beta) \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \beta 2^{\frac{\beta-1}{2}}}\right)^{\frac{1}{\beta}} \quad (4.1)$$

$$M = P(L_c + \frac{x_2}{2}) \quad (4.2)$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2} \quad (4.3)$$

$$J = 2\sqrt{2}x_1x_2R^2 \quad (4.4)$$

$$\Phi = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2) \quad (4.5)$$

$$\sigma = \frac{6PL_c}{x_3^2x_4} \quad (4.6)$$

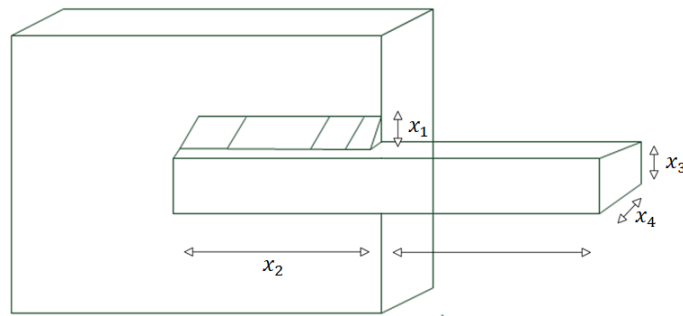
$$\Delta = \frac{4PL_c^3}{Ex_3^3x_4} \quad (4.7)$$

$$P_c = 4.013E \sqrt{\frac{x_2^3x_4^6}{36} \frac{1-x_3\sqrt{\frac{E}{4G}}}{L_c^2}} \quad (4.8)$$

$$\tau_p = \frac{P}{\sqrt{2}x_1x_2} \quad (4.9)$$

$$\tau_{pp} = \frac{M \cdot R}{J} \quad (4.10)$$

$$\tau = \sqrt{\tau_p^2 + \frac{2\tau_p\tau_{pp}x_2}{2R}} + \tau_{pp}^2 \quad (4.11)$$



**Şekil 4.6.** Kaynaklı Kiriş Tasarım Problemi

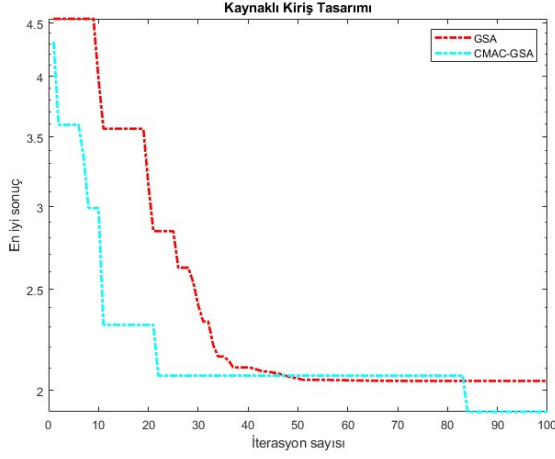
Tablo 4.21’de WBD probleminin CMACGSA ve diğer metasezgisel algoritmalar ile sonuçları karşılaştırılmıştır. Tablodaki en iyi değerlere göre PSO (1.6977) ile birinci sırada iken GWO (1.6978) ile ikinci sıradadır. CMACGSA için en iyi değer (1.9082) ile olarak gösterilmiştir. Genel olarak PSO ve GWO düşük çözümler üretmekte ve tutarlı

sonular verirken CMACGSA ise klasik GSA'ya kıyasla WBD problemi iin umut verici sonular elde etmiřtir.

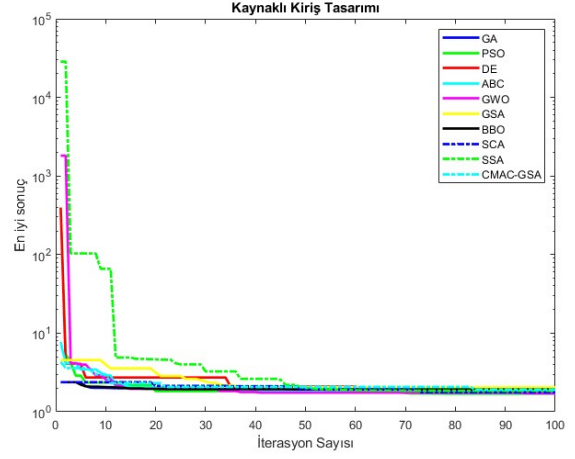
**Tablo 4.21.** CMACGSA ve dięer Metasezgisel Algoritmaların WBD Probleminin Sonuları

Algoritma	özüm				Maliyet				
	$x_1$	$x_2$	$x_3$	$x_4$	En iyi	En kötü	Medyan	Ortalama	Standart
<b>GA</b>	0.2022	3.5333	8.4084	0.2376	1.8450	3.6806	2.5013	2.6848	0.5971
<b>PSO</b>	0.2049	3.2640	9.0520	0.2057	1.6977	1.8965	1.7069	1.7317	0.0606
<b>DE</b>	0.2169	3.1770	9.9664	0.2176	1.9571	2.2219	2.1033	2.0949	0.0968
<b>ABC</b>	0.1747	4.3112	9.1855	0.2065	1.8166	2.1563	1.9144	1.9645	0.1218
<b>GWO</b>	0.2056	3.2595	9.0478	0.2057	1.6978	1.7296	1.7074	1.7103	0.0100
<b>GSA</b>	0.2966	2.4484	7.6867	0.2966	2.0424	3.2792	2.8482	2.7212	0.4398
<b>BBO</b>	0.1870	3.8924	8.1776	0.2515	1.9207	2.9974	2.6626	2.5302	0.4042
<b>SCA</b>	0.1952	3.4905	9.3547	0.2061	1.7691	2.1159	1.9859	1.9764	0.0931
<b>SSA</b>	0.2105	3.4400	8.3871	0.2388	1.8490	2.9062	2.0128	2.0939	0.3120
<b>CMACGSA</b>	0.1891	3.9672	9.2266	0.2196	1.9082	2.2548	2.0009	2.0146	0.1001

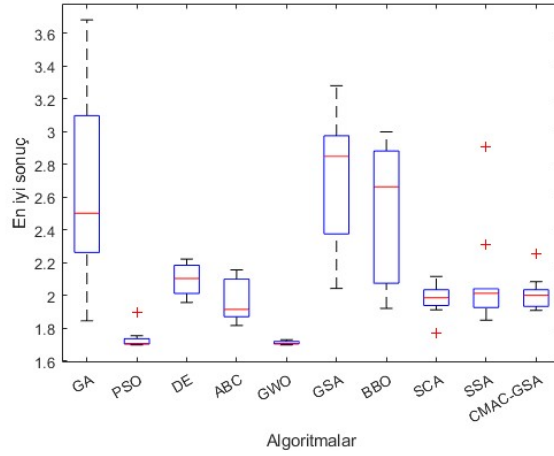
řekil 4.7'de WBD problemi iin metasezgisel algoritmalar ve CMACGSA' nun yakınsama eęrileri ve kutu grafikleri sunulmuřtur. řekil 4.7(a) incelendięinde CMACGSA'nın daha hızlı yakınsadıęı ve GSA'ya kıyasla daha düşük en iyi özüme ulařtıęı tespit edilmiřtir. řekil 4.7(b)'de CMACGSA ile dięer metasezgisel algoritmalarının yakınsama eęri grafikleri yer almaktadır. Grafięe göre, CMACGSA'nın dięer algoritmalara kıyasla erken yinelemede hızlı bir düşüř sergiledięi gözlemlenmektedir. řekil 4.7(c)'de CMACGSA ile dięer metasezgisel algoritmaların en iyi özüm deęerlerinin kutu grafikleri sunulmuřtur. Grafikler incelendięinde, CMACGSA'nın hem özüm kalitesi hem de tutarlılık bakımından dięer metasezgisel algoritmalara kıyasla daha üstün bir performansa sahip olduęu görülmüřtür.



(a)



(b)



(c)

**Şekil 4.7.** Kaynaklı Kiriş Tasarımı Probleminde Elde Edilen Sonuçlar (a) Klasik GSA ve CMACGSA'nın Yakınsama Eğrisi, (b) CMACGSA ve diğer Algoritmaların Yakınsama Eğrisi, (c) CMACGSA ve diğer Algoritmaların Kutu Grafiği

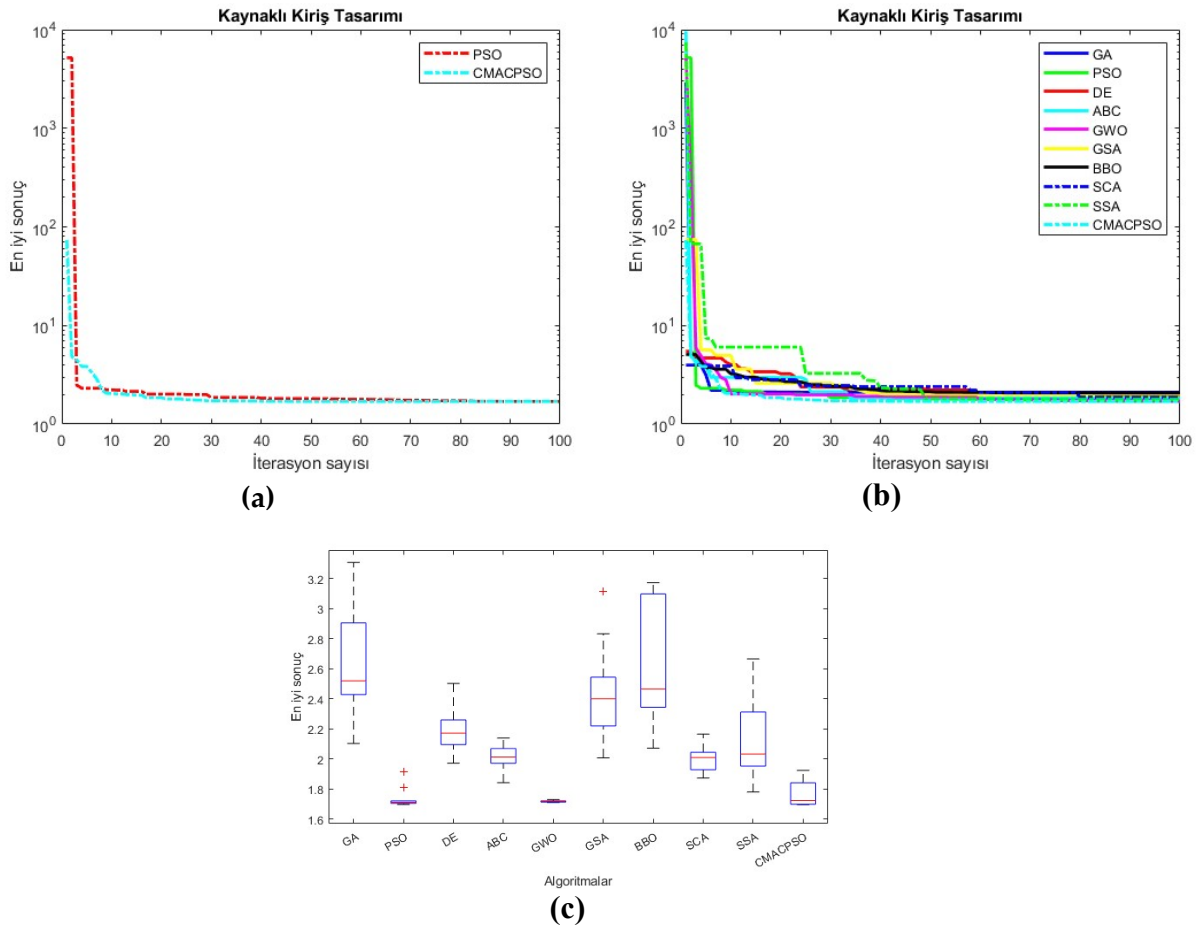
**Tablo 4.22.** CMACPSO ve diğer Metasezgisel Algoritmaların WBD Probleminin Sonuçları

Algoritma	Çözüm				Maliyet				
	$x_1$	$x_2$	$x_3$	$x_4$	En iyi	En kötü	Medyan	Ortalama	Standart
ABC	0.1770	4.3321	8.8330	0.2172	2.1031	3.3079	2.5203	2.6278	0.3496
BBO	0.3088	2.3979	7.4279	0.3105	1.6971	1.9135	1.7120	1.7392	0.0694
DE	0.2374	2.8838	8.6401	0.2554	1.9719	2.5030	2.1723	2.1882	0.1531
GA	0.3032	2.5046	7.2155	0.3227	1.8419	2.1394	2.0133	2.0063	0.0884
GSA	0.1739	4.6356	8.1336	0.2541	1.7104	1.7300	1.7181	1.7187	0.0063
GSA	0.1997	3.4284	9.0370	0.2058	2.0078	3.1139	2.4003	2.4448	0.3271
PSO	0.2049	3.2682	9.0414	0.2058	2.0721	3.1732	2.4667	2.6117	0.3990
SCA	0.1949	3.6356	9.3432	0.2171	1.8732	2.1656	2.0100	2.0004	0.0863
SSA	0.1990	3.1662	9.8441	0.2020	1.7810	2.6660	2.0330	2.1078	0.2671
CMACPSO	0.2055	3.2568	9.0412	0.2057	1.6960	1.9243	1.7233	1.7626	0.0820

Tablo 4.22'de WBD probleminin CMACPSO ve diğer metasezgisel algoritmalar ile sonuçları karşılaştırılmıştır. Tablo incelendiğinde, en iyi değerlere göre CMACPSO

(1.6960)'nun diğer metasezgisel algoritmalarla kıyasla daha üstün performansa sahip olduğu görülmüştür. Sonuç olarak, CMACPSO'nun klasik PSO'ya kıyasla daha tutarlı ve etkili çözümler sergilediği tespit edilmiştir.

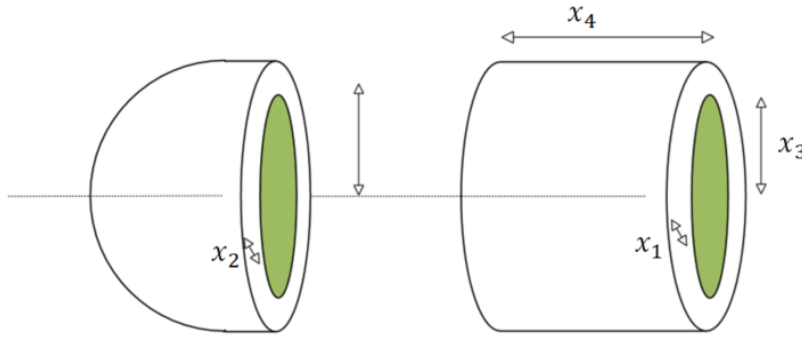
Şekil 4.8'de WBD problemi için metasezgisel algoritmalar ve CMACPSO'nun yakınsama eğrileri ve kutu grafikleri sunulmuştur. Şekil 4.8(a) incelendiğinde CMACPSO'nun PSO'ya kıyasla daha hızlı yakınsadığı ve en iyi sonuca ulaştığı gözlemlenmiştir. Şekil 4.8(b)'de CMACPSO algoritması ile diğer metasezgisel algoritmalarının yakınsama eğri grafikleri yer almaktadır. CMACPSO'nun yinelemenin başında diğer algoritmalarla kıyasla en iyi çözüm değerlerine ulaştığı gözlemlenmiştir. Şekil 4.8(c)'de CMACPSO ile diğer metasezgisel algoritmalarla ilişkin en iyi çözüm değerlerinin kutu grafikleri sunulmuştur. Grafik incelendiğinde, CMACPSO'nun PSO ve GWO'nun gerisinde kalırken, diğer algoritmaları kıyasla daha iyi sonuçlar verdiği görülmektedir.



**Şekil 4.8.** Kaynaklı Kiriş Tasarımı Probleminden Elde Edilen Sonuçlar (a) Klasik PSO ve CMACPSO'nun Yakınsama Eğrisi, (b) CMACPSO ve diğer Algoritmaların Yakınsama Eğrisi, (c) CMACPSO ve diğer Algoritmaların Kutu Grafiği

#### 4.6.2. Basınçlı Kap Tasarımı

Basınçlı Kap Tasarımı (PVD), yüksek basınçta çalışır ve sıvıları ya da gazları bir yerden başka bir yere taşıyan, işleyen ve depolayan mühendislik problemlerinden biridir. Problem dört parametreden oluşur. Bunlar, kabuk kalınlığı ( $x_1$ ), kafa kalınlığı ( $x_2$ ), iç yarıçap ( $x_3$ ) ve silindir kabın uzunluğu ( $x_4$ ) ve bu parametreleri kullanarak, toplam üretim maliyetini en aza indirmeyi amaçlanmaktadır. Problemin çizimi Şekil 4.9'da sunulmuştur.



Şekil 4.9. Basınçlı Kap Tasarımı Problemi

Probleme ait matematiksel ifadeler Denklem 4.12-4.16 arasında sunulmuştur. Problemin maliyet fonksiyonunu  $F(x)$  olarak ifade edilmektedir. Problemin kısıtları  $G_1, G_2, G_3$  ve  $G_4$  fonksiyonlarıdır.

$$\min F(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.166x_1^2x_4 + 19.84x_1^2x_3 \quad (4.12)$$

$$G_1 = -x_1 + 0.0193x_3 \quad (4.13)$$

$$G_2 = -x_3 + 0.00954x_3 \quad (4.14)$$

$$G_3 = -\pi x_3^2 x_4 - (4/3)\pi x_3^3 + 129600 \quad (4.15)$$

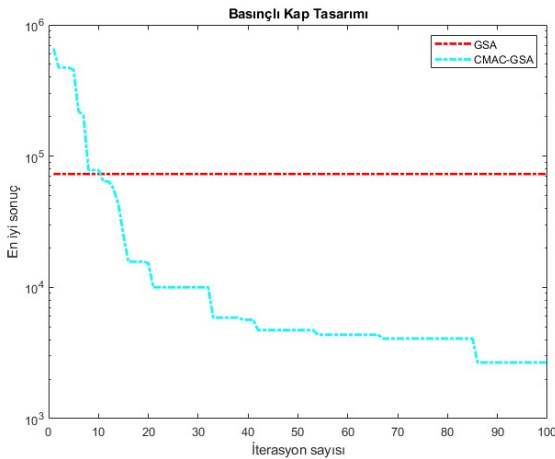
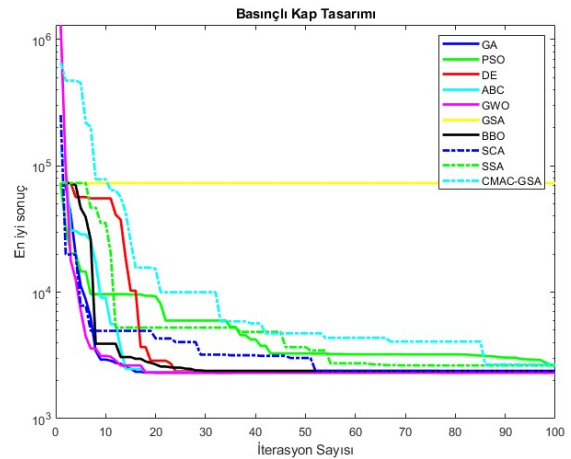
$$G_4 = x_4 - 240 \quad (4.16)$$

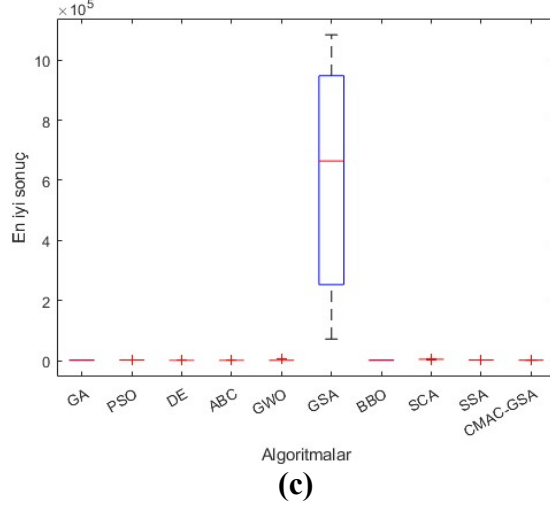
Tablo 4.23'te CMACGSA ve diğer metasezgisel algoritmalarının PVD probleminin sonuçları sunulmuştur. Tablo verilerine göre, CMACGSA'nın performansı diğer algoritmalara kıyasla rekabetçi fakat karışık bir performans sergilemektedir. CMACGSA çözüm kalitesi ve tutarlılık açısından GSA'yı geride bırakmıştır. Fakat, en iyi değerler açısından bakıldığında çoğu algoritmadan daha yüksek değere sahiptir.

**Tablo 4.23.** CMACGSA ve diğer Metasezgisel Algoritmaların PVD Probleminin Sonuçları

Algoritmalar	Çözüm				Maliyet				
	$x_1$	$x_2$	$x_3$	$x_4$	En iyi	En kötü	Medyan	Ortalama	Standart
GA	1.0969	0	65.226	10.00	2302.7	3611	3089.8	2925.4	525
PSO	1.0828	0.0073	64.636	16.274	2598.3	3638.3	3509.4	3342.5	392.25
DE	1.0935	0	65.225	10.000	2302.5	2309.2	2302.8	2303.8	2.2
ABC	1.0932	0	65.231	10.000	2303.1	2310.5	2303.7	2304.8	2.43
GWO	1.1032	0	65.236	10.002	2304.7	6123.3	2356.6	3094.6	1582.42
GSA	3.0668	5.744	73.784	21.89	7.31e+4	1.08e+6	6.64e+5	6.05e+5	3.61e+5
BBO	1.0755	0	64.477	13.270	2386.3	4.e+3	3151.8	3070.9	4.e+2
SCA	1.1713	0	65.402	10.000	2383.2	6156.4	6072.1	5346.8	1561.1
SSA	1.0172	0	62.074	24.296	2635.7	3637.3	3625.6	3523.1	312.41
CMACGSA	1.1683	0	66.873	10.459	2679.1	4185	3085.7	3192.8	477.4

Şekil 4.10'da PVD problemi için metasezgisel algoritmalar ve CMACGSA'nın yakınsama eğrileri ve kutu grafikleri verilmiştir. Şekil 4.10(a)'da CMACGSA ilk yinelemede hızlı bir yakınsama göstermiş ve en iyi çözüme kısa sürede ulaşmıştır. GSA ise sabit bir yakınsama eğilimi göstermiştir en iyi çözüme ulaşmada zorluklar yaşamıştır. Şekil 4.10(b)'de CMACGSA ile diğer metasezgisel algoritmalarının yakınsama eğri grafikleri gösterilmektedir. CMACGSA diğer algoritmalarla kıyasla tutarlı bir davranış sergilemiştir. Yakınsama eğrilerine göre DE, PSO ve GWO algoritmaları başarılı sonuçlar elde etmiştir. Sonuç olarak CMACGSA'nın diğer metasezgisel algoritmalarla karşı rekabetçi bir yaklaşım sergilediği gözlemlenmiştir. Şekil 4.10(c)'de CMACGSA ile diğer metasezgisel algoritmalarının kutu grafikleri sunulmuştur. Grafikler incelendiğinde, CMACGSA'nın performansının GSA'ya kıyasla oldukça yüksek olduğu görülmüştür.

**(a)****(b)**



**Şekil 4.10.** Basıncılı Kap Tasarımı Probleminden Elde Edilen Sonuçlar (a) Klasik GSA ve CMACGSA'nın Yakınsama Eğrisi, (b) CMACGSA ve diğer Algoritmaların Yakınsama Eğrisi, (c) CMACGSA ve diğer Algoritmaların Kutu Grafiği

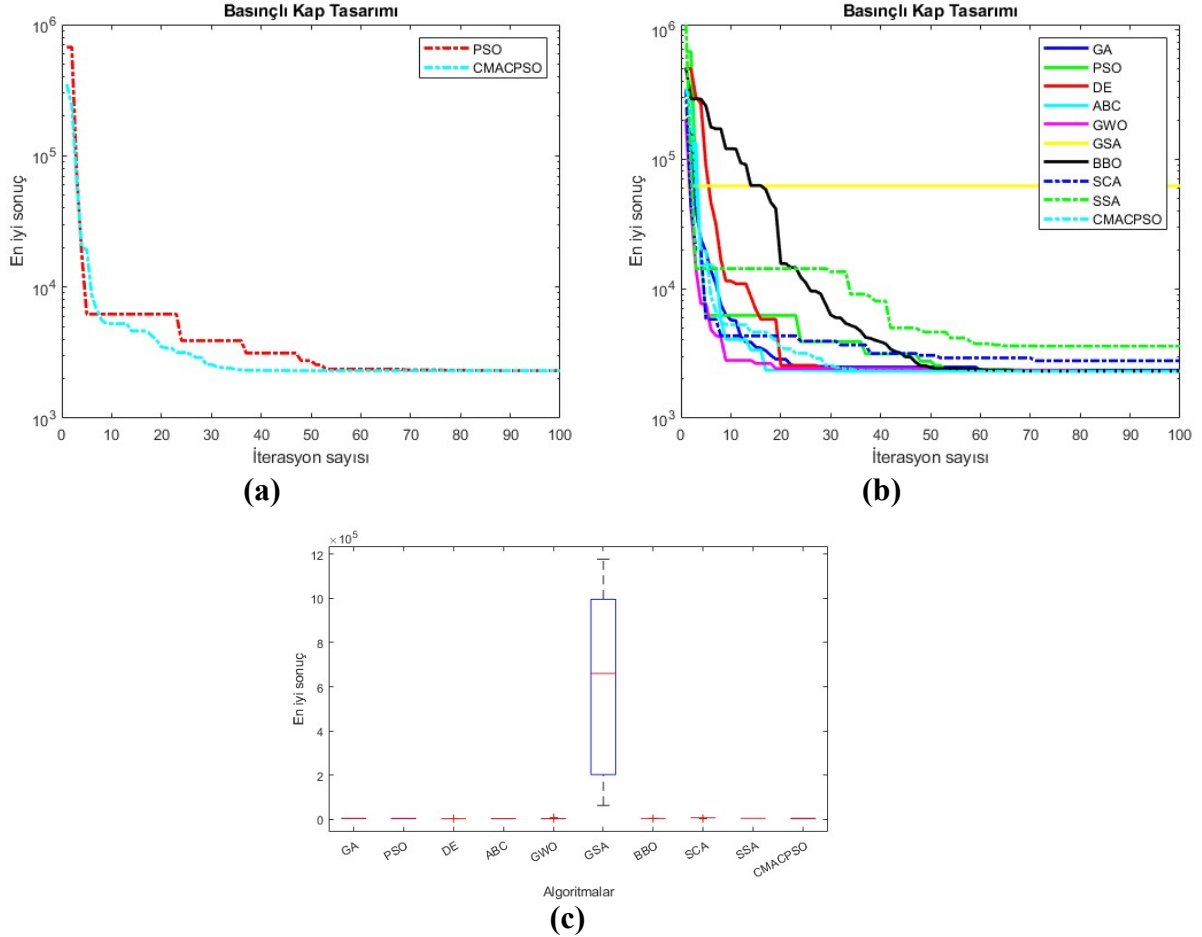
Tablo 4.24'te CMACPSO ve diğer metasezgisel algoritmalarının PVD probleminin sonuçları sunulmuştur. Tablo verilerine göre, CMACPSO (2302.50)'nun en iyi değerler açısından ABC, DE, GA, GWO, SCA ve SSA gibi popüler algoritmalara kıyasla karşılaştırılabilir performansa sahip olduğu tespit edilmiştir. Sonuç olarak, CMACPSO'nun klasik PSO'ya kıyasla çözüm kalitesinin ve güvenilirliğinin üstün olduğu görülmüştür.

**Tablo 4.24.** CMACPSO ve diğer Metasezgisel Algoritmaların PVD Probleminin Sonuçları

Algoritmalar	Çözüm				Maliyet				
	$x_1$	$x_2$	$x_3$	$x_4$	En iyi	En kötü	Medyan	Ortalama	Standart
ABC	1.0952	0.0000	65.2260	10.0030	2302.80	2371.50	2309.20	2324.00	27.749
BBO	1.0943	0.0000	65.2720	10.0000	2307.00	3632.80	3270.30	3132.80	426.44
DE	1.0936	0.0000	65.2250	10.0000	2302.50	2306.80	2302.60	2303.20	1.3642
GA	1.1513	0.0000	65.2310	10.0000	2340.70	3631.00	3095.60	3057.30	503.99
GSA	2.2287	1.5489	94.4890	190.8700	6.19e+4	1.18e+6	6.60e+5	6.11e+5	4.06e+5
GWO	1.0950	0.0000	65.2320	10.0000	2303.20	6055.60	2306.30	2838.50	1203.30
PSO	1.0975	0.0000	65.2260	10.0010	2302.90	3638.70	3493.10	3179.00	609.30
SCA	1.2393	0.0000	67.0280	12.4400	2775.50	6139.60	6092.10	5762.80	1049.90
SSA	0.6209	0.0000	46.3180	130.5300	3595.50	3636.90	3630.10	3624.50	13.948
CMACPSO	1.0936	0.0000	65.2250	10.0000	2302.50	3624.40	2315.50	2819.60	663.22

Şekil 4.11'de PVD problemi için metasezgisel algoritmalar ve CMACPSO'nun yakınsama eğrileri ve kutu grafikleri sunulmuştur. Şekil 4.11(a) incelendiğinde, CMACPSO'nun daha hızlı ve sonrasında sabit şekilde en iyi sonuca ulaştığı görülmektedir. Grafiğe göre CMACPSO algoritmasının performansı PSO'ya kıyasla daha iyi olduğu izlenmektedir. Şekil 4.11(b)'de CMACPSO algoritması ile diğer metasezgisel algoritmalarının yakınsama eğri grafikleri yer almaktadır. CMACPSO'da diğer

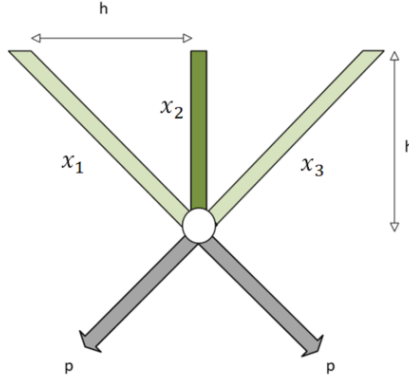
algoritmalarla kıyasla daha iyi sonuçlar elde etmiştir. Yakınsama eğrilerine göre GSA, BBO ve SSA gibi algoritmalar düşük performans göstermiştir. Şekil 4.11(c)'de CMACPSO algoritması ile diğer metasezgisel algoritmalarının kutu grafikleri sunulmuştur. Grafikler incelendiğinde, CMACPSO diğer metasezgisel algoritmalarla rekabetçi ve güvenilir sonuçlar ürettiği görülmektedir.



**Şekil 4.11.** Basınçlı Kap Tasarımı Probleminden Elde Edilen Sonuçlar (a) Klasik PSO ve CMACPSO'nun Yakınsama Eğrisi, (b) CMACPSO ve diğer Algoritmaların Yakınsama Eğrisi, (c) CMACPSO ve diğer Algoritmaların Kutu Grafiği

#### 4.6.3. Üç Çubuklu Makas Tasarımı

Üç çubuklu makas tasarımı (TBT) problemi, literatürde sıklıkla kullanılan mühendislik problemlerinden biridir. Probleme ait olan üç parametre, minimum maliyet ağırlıklarını bulmayı amaçlamaktadır. Problemin çizimi Şekil 4.12'de yer almaktadır. Çapraz olarak konumlandırılmış  $x_1$  ve  $x_3$  değerleri üzerinden kafes elemanlarının her birinin gerilimi göz önüne alındığında, kafes hacminin optimum değer olması beklenir.



Şekil 4.12. Üç Çubuklu Makas Tasarımı

Probleme ait matematiksel ifadeler Denklem 4.17-4.21 arasında yer almaktadır. Problemin maliyet fonksiyonunu  $F(x)$  olarak ifade edilmektedir. Problemin kısıtları  $G_1$ ,  $G_2$  ve  $G_3$  fonksiyonları olarak yer almaktadır. Tablo 4.25'te problemin parametreleri sunulmuştur.

$$(x_1 = x_3) \quad (4.17)$$

$$\min F(x) = L * (2\sqrt{2x_1} + x_2) \quad (4.18)$$

$$G_1 = \frac{\sqrt{2x_1} + x_2}{\sqrt{2x_1^2} + 2x_1x_2} P - RU \quad (4.19)$$

$$G_2 = \frac{x_2}{\sqrt{2x_1^2} + 2x_1x_2} P - RU \quad (4.20)$$

$$G_3 = \frac{1}{x_1 + \sqrt{2x_2}} P - RU \quad (4.21)$$

Tablo 4.25. TBT Probleminin Parametreleri

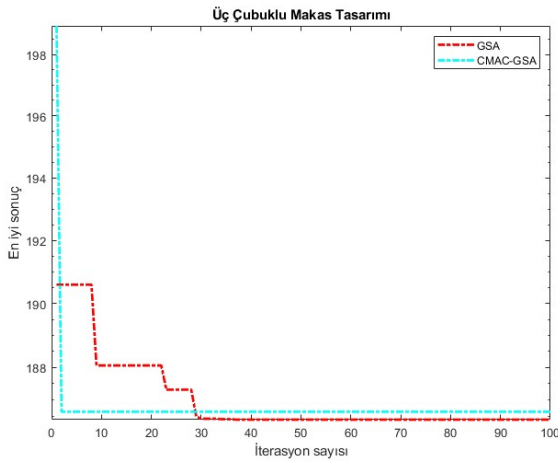
Açıklama	Parametre
Çubuk uzunluğu	$L = 100$
Yük	$P = 2$
İzin verilen gerilim uzunluğu	$RU = 2$

Tablo 4.26'da CMACGSA ve diğer metasezgisel algoritmaların TBT probleminin sonuçları yer almaktadır. Tablo verileri incelendiğinde, CMACGSA'nın (186.630) en iyi değer bakımından, diğer popüler algoritmalara (186.386) yakın sonuç verdiği görülmektedir. Buna göre, CMACGSA'nın kararlılığı ve tekrarlanabilirliği yüksek senaryolar için uygun olduğu anlaşılmaktadır.

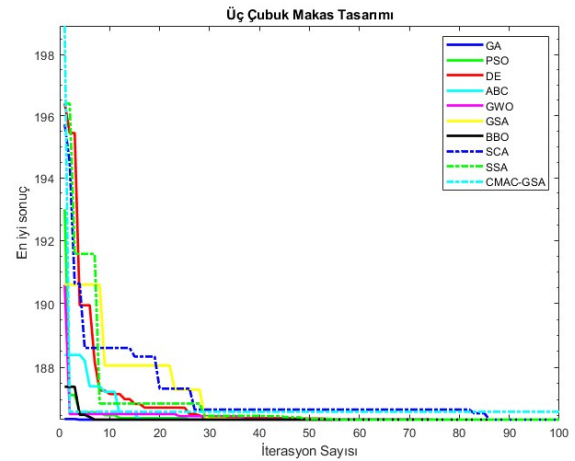
**Tablo 4.26.** CMACGSA ve diğer Metasezgisel Algoritmaların TBT Probleminin Sonuçları

Algoritmalar	Çözüm		Maliyet				
	$x_1$	$x_2$	En iyi	En kötü	Medyan	Ortalama	Standart
GA	0.7870	0.2877	186.386	191.176	186.486	187.203	1.516
PSO	0.7868	0.2880	186.386	186.386	186.386	186.386	0.000
DE	0.7868	0.2880	186.386	186.387	186.386	186.386	0.000
ABC	0.7867	0.2883	186.386	186.398	186.387	186.388	0.003
GWO	0.7870	0.2877	186.386	186.400	186.387	186.389	0.004
GSA	0.7859	0.2898	186.387	188.612	187.071	187.204	0.715
BBO	0.7866	0.2884	186.386	193.877	186.585	187.395	2.299
SCA	0.7862	0.2893	186.386	187.773	186.612	186.728	0.432
SSA	0.7868	0.2880	186.386	186.386	186.386	186.386	0.000
CMACGSA	0.7857	0.2857	186.630	186.630	186.630	186.630	0.000

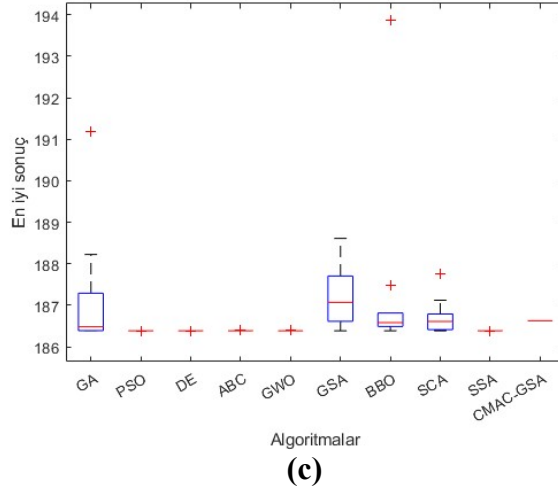
Şekil 4.13'te TBT problemi için metasezgisel algoritmalar ve CMACGSA'nın yakınsama eğrileri ve kutu grafikleri sunulmuştur. Şekil 4.13(a) incelendiğinde CMACGSA ilk yinelemelerde en iyi çözüme ulaşmış ve sonrasında sabit kalmıştır. Sonuç olarak, CMACGSA, GSA'ya kıyasla daha hızlı ve etkili yakınsama yeteneğine sahiptir. Şekil 4.13(b)'te CMACGSA ile diğer metasezgisel algoritmalarının yakınsama eğri grafikleri yer almaktadır. CMACGSA diğer algoritmalara kıyasla hızlı yakınsama ve tutarlı davranış sergilemiştir. Yakınsama eğrilerine göre DE, PSO, GWO ve SSA algoritmaları erken yakınsama eğrisi çizmelerine rağmen CMACGSA'ya kıyasla yüksek sonuçlar elde etmiştir. Şekil 4.13(c)'te CMACGSA ile diğer metasezgisel algoritmalarının kutu grafikleri sunulmuştur. Grafikler incelendiğinde, CMACGSA'nın performansı GSA, GA, BBO ve SCA gibi algoritmalara kıyasla oldukça yüksektir.



(a)



(b)



**Şekil 4.13.** Üç Çubuklu Makas Tasarımı Probleminde Elde Edilen Sonuçlar (a) Klasik GSA ve CMACGSA'nın Yakınsama Eğrisi, (b) CMACGSA ve diğer Algoritmaların Yakınsama Eğrisi, (c) CMACGSA ve diğer Algoritmaların Kutu Grafiği

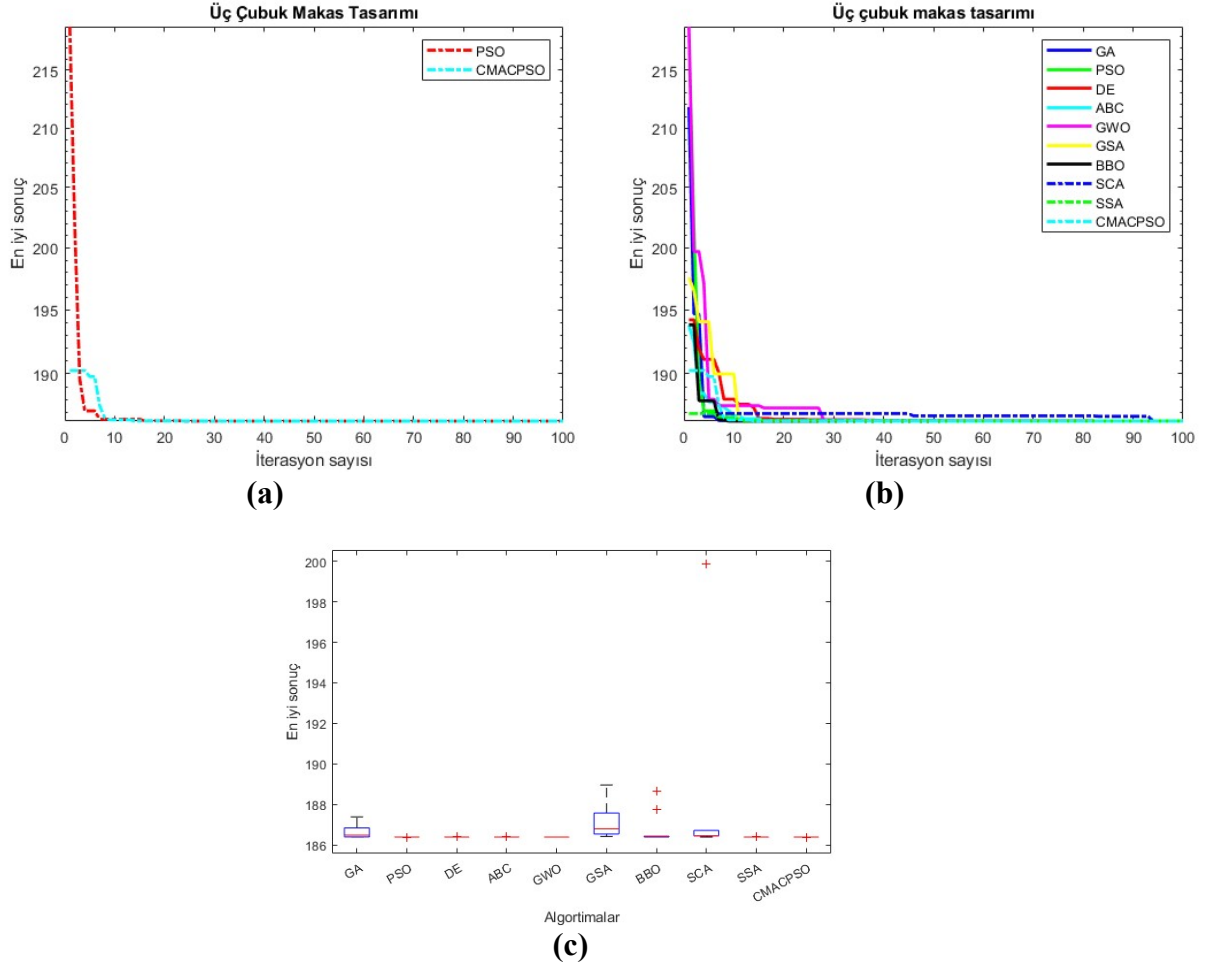
Tablo 4.27'de CMACPSO ve diğer metasezgisel algoritmaların TBT probleminin sonuçları yer almaktadır. Tablo verileri incelendiğinde, CMACPSO'nun TBT problemi için diğer algoritmalara kıyasla performansının nispeten rekabetçi olduğu görülmüştür. Bu da, CMACPSO'nun kararlılığı ve tekrarlanabilirliği yüksek senaryolar için daha uygun olduğu göstermektedir.

**Tablo 4.27.** CMACPSO ve diğer Metasezgisel Algoritmaların TBT Probleminin Sonuçları

Algoritmalar	Çözüm		Maliyet				
	$x_1$	$x_2$	En iyi	En kötü	Medyan	Ortalama	Standart
<b>ABC</b>	0,7872	0,2875	0,1117	0,1683	0,1146	0,1247	0,0201
<b>BBO</b>	0,7864	0,2890	0,0315	0,0460	0,0338	0,0361	0,0055
<b>DE</b>	0,7868	0,2880	0,0331	0,0464	0,0351	0,0370	0,0044
<b>GA</b>	0,7869	0,2880	0,1197	0,1910	0,1277	0,1340	0,0213
<b>GSA</b>	0,7940	0,2748	0,0461	0,0834	0,0471	0,0520	0,0115
<b>GWO</b>	0,7869	0,2879	0,0094	0,0247	0,0098	0,0121	0,0049
<b>PSO</b>	0,7869	0,2880	0,0889	0,1364	0,0908	0,0979	0,0152
<b>SCA</b>	0,7850	0,2917	0,0090	0,0171	0,0094	0,0103	0,0024
<b>SSA</b>	0,7869	0,2880	0,0109	0,0194	0,0113	0,0122	0,0026
<b>CMACPSO</b>	0,7869	0,2880	0,6010	0,7034	0,6125	0,6300	0,0344

Şekil 4.14'te TBT problemi için metasezgisel algoritmalar ve CMACPSO'nun yakınsama eğrileri ve kutu grafikleri yer almaktadır. Şekil 4.14(a) incelendiğinde, CMACPSO'da yapılan yenilikler sayesinde klasik PSO'ya kıyasla başarılı sonuçlar elde etmiş ve yakınsama hızını artırmıştır. Şekil 4.14(b)'te CMACPSO ile diğer metasezgisel algoritmalarının yakınsama eğri grafikleri sunulmaktadır. CMACPSO diğer algoritmalara kıyasla hızlı yakınsama ve tutarlı davranış sergilemiştir. Yakınsama eğrilerine göre GSA, BBO ve SCA daha yavaş ve kötü sonuçlar elde ederek CMACPSO'nun gerisinde

kalmıştır. Şekil 4.14(c)'te CMACPSO algoritması ile diğer metasezgisel algoritmalarının kutu grafikleri gösterilmektedir. Grafik incelendiğinde, CMACPSO'nun performansı GA, GSA ve BBO gibi algoritmalara kıyasla oldukça yüksektir. Genel olarak CMACPSO istikrarlı, güvenilir ve rekabetçi performans sergilemektedir.



**Şekil 4.14.** Üç Çubuklu Makas Tasarımı Probleminden Elde Edilen Sonuçlar (a) Klasik PSO ve CMACPSO'nun Yakınsama Eğrisi, (b) CMACPSO ve diğer Algoritmaların Yakınsama Eğrisi, (c) CMACPSO ve diğer Algoritmaların Kutu Grafiği

## 5. SONUÇLAR

Bu tez çalışmasında, CMAC sinir ağı ile entegre edilmiş ve iyileştirilmiş GSA'nın varyantı olan CMACGSA ve PSO'nun varyantı olan CMACPSO isimli iki yeni algoritma geliştirilmiştir. Öncelikle, CMAC'in başlangıç adımı olan kuantalama işlemi yeni bir yaklaşım olarak GSA'nın ajanlarına uygulanmıştır. Ek olarak, CMAC'in genelleme ve kolay öğrenme yeteneğinden yararlanarak, algoritmaların karmaşık optimizasyon problemlerine karşı etkili ve verimli çözümler sunmasına katkı sağlanmıştır. Ayrıca algoritmaların erken yakınsama ve yerel minimuma takılması gibi sorunların üstesinden gelmek için çeşitli mekanizmalar eklenerek gerekli iyileştirme yapılmıştır. Klasik GSA'nın performansını daha da iyileştirmek için önerdiğimiz algoritmaya, üç farklı mekanizma eklenmiştir. Bu mekanizmalar sınır kontrol sistemi Lévy mutasyon operatörü ve sayaç sistemidir. Bu mekanizmalar sayesinde CMACGSA'nın karmaşık optimizasyon problemlerine karşı sağlamlığı ve uygulanabilirliği gözlenmiştir.

PSO algoritması, optimizasyon problemlerinde etkili çözümler sunması ile tercih edilen bir algoritma olmasına karşın çeşitli kısıtları bulunmaktadır. Bu kısıtları çözmek için tezin diğer bir amacı olarak klasik PSO algoritmasında çeşitli iyileştirmeler yapılmıştır. Öncelikle CMAC'in kuantalama işlemi PSO'nun başlangıç koşulu olarak parçacıklara uygulanmıştır. Böylece arama uzayında keşif çeşitliliği artırılması amaçlanmaktadır. Daha sonra PSO'nun hız güncelleme formülünde yer alan rastgele iki değişken yerine dinamik öğrenme yeteneğine sahip iki değişken kullanılmıştır. Bu değişkenler CMAC'in ağırlıklarını kullanarak güncellenmektedir. Son olarak, L-SHADE'den esinlenen bir sınır kontrol mekanizması CMACPSO algoritması içinde uygulanmıştır. Konumu güncellenen parçacıkların arama uzayı sınırlarını ihmal etmesinin önüne geçilmiştir. Böylelikle PSO algoritmasının erken yakınsama, başlangıçta popülasyon çeşitliliğinin zayıflığı gibi sorunları en aza indirgenmesi hedeflenmiştir.

CMACGSA ve CMACPSO algoritmalarının performansını daha iyi anlamak ve uygulanabilirliğini incelemek için hem standart test fonksiyonları hem de gerçek dünya mühendislik problemleri kullanılmıştır. Deneysel çalışmalar kapsamında, CEC 2014 test paketi farklı boyutlarda (10D, 30D, 50D) sistematik olarak test edilmiştir. Böylece algoritmaların düşük ve yüksek boyutlardaki sonuçları elde edilmiştir. Elde edilen sonuçlar incelendiğinde, CMACGSA ve CMACPSO algoritmalarının GSA ve PSO algoritmalarına kıyasla daha tutarlı ve güvenilir sonuçlar verdiği gözlemlenmiştir. CMACGSA ve CMACPSO algoritmalarının CEC 2014 test paketinin farklı boyutlarında elde edilen

sonuçlarına göre, CMACPSO'nun CMACGSA'ya kıyasla daha kararlı ve etkili sonuçlar sergilediği gözlemlenmiştir. Ayrıca, CEC 2014 30D boyutunda GSA ve PSO algoritmalarının literatürde var olan mevcut versiyonları ile karşılaştırılmıştır. Karşılaştırma sonuçlarına göre, CMACGSA'nın GSA varyantlarına göre, CMACPSO'nun PSO varyantlarına göre daha üstün bir başarı elde ettiği tespit edilmiştir. Geliştirdiğimiz algoritmaların geçerliliğini daha iyi değerlendirebilmek için mevcut popüler olan diğer algoritmalar ile CEC 2014 test fonksiyonları 30D boyutta karşılaştırılmıştır. Elde edilen sonuçlarına göre CMACPSO birinci sırada iken, CMACGSA ikinci sırada yer almaktadır. Deneysel çalışmanın son bölümünde CMACGSA ve CMACPSO algoritmalarının pratik uygulamalara uygunluğunu değerlendirmek amacıyla popüler olan mühendislik optimizasyon problemlerinden üç tanesi kullanılmıştır. Bu problemler kaynaklı kiriş tasarımı, basınçlı kap tasarımı ve son olarak üç çubuklu makas tasarımıdır. Problemlerin hem kısıtları olması hem de tasarım değişkenlerinin çokluğu sebebiyle zorluk seviyesi yüksektir. Elde edilen sonuçlar incelendiğinde, CMACGSA ve CMACPSO'nun standart test fonksiyonlarına benzer şekilde mühendislik problemleri için de özellikle GSA ve PSO'ya kıyasla daha başarılı sonuçlar elde ettiği gözlemlenmiştir. Çalışmanın en önemli kısıtı CMAC'in ağırlıklarının hesaplanması işleminden dolayı ek hesaplama maliyeti gerektirmesidir. Dolayısıyla geliştirilen CMACGSA ve CMACPSO algoritmaları zaman açısından daha yavaş çalışmaktadır.

Sonuç olarak, bu tez çalışmasında, CMAC sinir ağlarının öğrenme yeteneğinin kullanılmasıyla hibritleştirilen iki yeni algoritmanın (CMACGSA ve CMACPSO) özellikle karmaşık optimizasyon problemlerinin çözümlerinde kararlı ve tutarlı sonuçlara sahip olduğu görülmüştür. Bu kapsamda, tez çalışmasında geliştirilen algoritmalar, özellikle literatürdeki diğer klasik algoritmaların eksikliklerinin ve sınırlılıklarının aşılması için öğrenme yeteneğinin kullanılmasının iyi bir alternatif olduğunu gösteren öncü bir çalışma niteliğindedir.

## KAYNAKÇA

- Abdelhameed, Magdy, Kassem, Ahmed, & Cairo, Egypt Benha.** (2010). *CMAC NEURAL NETWORK: MODELING, SIMULATION, AND A COMPARATIVE STUDY OF LEARNING ALGORITHMS*. Paper presented at the Proceedings of 11th Middle Eastern Simulation Multiconference (EUROSIS MESM 2010).
- Ahmadi, Seyed-Alireza.** (2017). Human behavior-based optimization: a novel metaheuristic approach to solve complex optimization problems. *Neural Computing and Applications*, 28(Suppl 1), 233-244.
- Al Bataineh, Ali, & Manacek, Sarah.** (2022). MLP-PSO hybrid algorithm for heart disease prediction. *Journal of Personalized Medicine*, 12(8), 1208.
- Albus, James S.** (1975a). Data storage in the cerebellar model articulation controller (CMAC).
- Albus, James S.** (1975b). A new approach to manipulator control: The cerebellar model articulation controller (CMAC).
- Alirezanejad, Mehdi, Enayatifar, Rasul, Motameni, Homayun, & Nematzadeh, Hossein.** (2021). GSA-LA: gravitational search algorithm based on learning automata. *Journal of Experimental & Theoretical Artificial Intelligence*, 33(1), 109-125.
- Amiri, Roya, Tahmouresi, Amirhossein, Momenaei Kermani, Vahid, Mirjalili, Seyedali, & Majrouhi Sardroud, Javad.** (2025). Tower crane location optimization problem: a comprehensive metaheuristic algorithm approach. *Evolutionary Intelligence*, 18(2), 44.
- Bahrololoum, Abbas, Nezamabadi-Pour, Hossein, Bahrololoum, Hamid, & Saeed, Masoud.** (2012). A prototype classifier based on gravitational search algorithm. *Applied Soft Computing*, 12(2), 819-825.
- Bas, Eren, Egrioglu, Erol, & Karahasan, Ozlem.** (2021). A Pi-Sigma artificial neural network based on sine cosine optimization algorithm. *Granular Computing*, 1-8.
- Bohat, Vijay Kumar, & Arya, KV.** (2018). An effective gbest-guided gravitational search algorithm for real-parameter optimization and its application in training of feedforward neural networks. *Knowledge-based systems*, 143, 192-207.
- Bucak, İhsan Ömür, & Baki, Semra.** (2010). Diagnosis of liver disease by using CMAC neural network approach. *Expert Systems with Applications*, 37(9), 6157-6164.
- Bulut, Nazmiye Ebru, Dandil, Emre, Yuzgec, Ugur, & Duysak, Alpaslan.** (2025). CMACGSA: Improved Gravitational Search Algorithm Based on Cerebellar Model Articulation Controller for Optimization. *IEEE Access*.
- Chao, Fei, Zhou, Dajun, Lin, Chih-Min, Yang, Longzhi, Zhou, Changle, & Shang, Changjing.** (2019). Type-2 fuzzy hybrid controller network for robotic systems. *IEEE Transactions on Cybernetics*, 50(8), 3778-3792.
- Chen, Jeng-Fung, Do, Quang Hung, & Hsieh, Ho-Nien.** (2015). Training artificial neural networks by a hybrid PSO-CS algorithm. *Algorithms*, 8(2), 292-308.
- Chen, Ke, Zhou, Fengyu, Wang, Yugang, & Yin, Lei.** (2018). An ameliorated particle swarm optimizer for solving numerical optimization problems. *Applied Soft Computing*, 73, 482-496.

- Chen, Wei-Neng, Zhang, Jun, Lin, Ying, Chen, Ni, Zhan, Zhi-Hui, Chung, Henry Shu-Hung, Li, Yun, & Shi, Yu-Hui.** (2012). Particle swarm optimization with an aging leader and challengers. *IEEE transactions on evolutionary computation*, 17(2), 241-258.
- Chu, Shu-Chuan, Tsai, Pei-Wei, & Pan, Jeng-Shyang.** (2006). *Cat swarm optimization*. Paper presented at the Pacific Rim international conference on artificial intelligence.
- Cotter, Neil E, & Guillermin, Thierry J.** (1992). The CMAC and a theorem of Kolmogorov. *Neural networks*, 5(2), 221-228.
- Cuong-Le, Thanh, Nghia-Nguyen, Trong, Khatir, Samir, Trong-Nguyen, Phuoc, Mirjalili, Seyedali, & Nguyen, Khuong D.** (2022). An efficient approach for damage identification based on improved machine learning using PSO-SVM. *Engineering with computers*, 1-16.
- Dehghani, Mohammad, Montazeri, Zeinab, Trojovská, Eva, & Trojovský, Pavel.** (2023). Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems. *Knowledge-based systems*, 259, 110011.
- Dehghani, Mohammad, & Trojovský, Pavel.** (2021). Teamwork optimization algorithm: A new optimization approach for function minimization/maximization. *Sensors*, 21(13), 4567.
- Dorigo, Marco, Birattari, Mauro, & Stutzle, Thomas.** (2007). Ant colony optimization. *IEEE computational intelligence magazine*, 1(4), 28-39.
- Erol, Osman K, & Eksin, Ibrahim.** (2006). A new optimization method: big bang–big crunch. *Advances in engineering software*, 37(2), 106-111.
- Formato, Richard.** (2007). Central force optimization: a new metaheuristic with applications in applied electromagnetics. *Progress in electromagnetics research*, 77, 425-491.
- Garg, Harish.** (2016). A hybrid PSO-GA algorithm for constrained optimization problems. *Applied Mathematics and Computation*, 274, 292-305.
- Garg, Harish.** (2019). A hybrid GSA-GA algorithm for constrained optimization problems. *Information sciences*, 478, 499-523.
- Geem, Zong Woo, Kim, Joong Hoon, & Loganathan, Gobichettipalayam Vasudevan.** (2001). A new heuristic optimization algorithm: harmony search. *simulation*, 76(2), 60-68.
- Gou, Jin, Lei, Yu-Xiang, Guo, Wang-Ping, Wang, Cheng, Cai, Yi-Qiao, & Luo, Wei.** (2017). A novel improved particle swarm optimization algorithm based on individual difference evolution. *Applied Soft Computing*, 57, 468-481.
- Guo, Anjing, Wang, Yirui, Guo, Lijun, Zhang, Rong, Yu, Yang, & Gao, Shangce.** (2023). An adaptive position-guided gravitational search algorithm for function optimization and image threshold segmentation. *Engineering Applications of Artificial Intelligence*, 121, 106040.
- Harandizadeh, Hooman, Armaghani, Danial Jahed, & Mohamad, Edy Tonnizam.** (2020). Development of fuzzy-GMDH model optimized by GSA to predict rock tensile strength based on experimental datasets. *Neural Computing and Applications*, 32, 14047-14067.

- Hatamlou, Abdolreza.** (2013). Black hole: A new heuristic optimization approach for data clustering. *Information sciences*, 222, 175-184.
- Heidari, Ali Asghar, Mirjalili, Seyedali, Faris, Hossam, Aljarah, Ibrahim, Mafarja, Majdi, & Chen, Huiling.** (2019). Harris hawks optimization: Algorithm and applications. *Future generation computer systems*, 97, 849-872.
- Holland, John H.** (1992). Genetic algorithms. *Scientific american*, 267(1), 66-73.
- Hsu, Chun-Fei, Chung, Chao-Ming, Lin, Chih-Min, & Hsu, Chia-Yu.** (2009). Adaptive CMAC neural control of chaotic systems with a PI-type learning algorithm. *Expert Systems with Applications*, 36(9), 11836-11843.
- Huang, Mei-Ling, & Chou, Yueh-Ching.** (2019). Combining a gravitational search algorithm, particle swarm optimization, and fuzzy rules to improve the classification performance of a feed-forward neural network. *Computer methods and programs in biomedicine*, 180, 105016.
- Huang, Shaojia, Xu, Tao, Chen, Jiaqing, Huang, Jiajia, Wang, Zhikun, Ke, Ya, & Yung, Wing Ho.** (2024). Advancements in Cerebellar Modeling and Its Practical Applications: A Comprehensive Review. *IEEE Access*.
- Jan, Jiun-Chi, & Hung, Shih-Lin.** (2001). High-order MS CMAC neural network. *IEEE Transactions on Neural Networks*, 12(3), 598-603.
- Jiang, Jianhua, Jiang, Ran, Meng, Xianqiu, & Li, Keqin.** (2020). SCGSA: A sine chaotic gravitational search algorithm for continuous optimization problems. *Expert Systems with Applications*, 144, 113118.
- Jiang, Jianhua, Yang, Xi, Meng, Xianqiu, & Li, Keqin.** (2020). Enhance chaotic gravitational search algorithm (CGSA) by balance adjustment mechanism and sine randomness function for continuous optimization problems. *Physica A: Statistical Mechanics and its Applications*, 537, 122621.
- Karaboga, Dervis.** (2005). *An idea based on honey bee swarm for numerical optimization*. Retrieved from
- Kashan, Ali Husseinzadeh.** (2015). A new metaheuristic for optimization: optics inspired optimization (OIO). *Computers & operations research*, 55, 99-125.
- Kaveh, A, Bakhshpoori, T, & Afshari, E.** (2014). An efficient hybrid particle swarm and swallow swarm optimization algorithm. *Computers & structures*, 143, 40-59.
- Kaveh, A, & Khayatazad, M.** (2012). A new meta-heuristic method: ray optimization. *Computers & structures*, 112, 283-294.
- Kaveh, Ali, & Talatahari, Siamak.** (2010). A novel heuristic optimization method: charged system search. *Acta mechanica*, 213(3), 267-289.
- Kennedy, James, & Eberhart, Russell.** (1995). *Particle swarm optimization*. Paper presented at the Proceedings of ICNN'95-international conference on neural networks.
- Kim, Young H, & Lewis, Frank L.** (2000). Optimal design of CMAC neural-network controller for robot manipulators. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 30(1), 22-31.
- Kolcz, A, & Allinson, NM.** (1994). Application of the CMAC input encoding scheme in the N-tuple approximation network. *IEE Proceedings-Computers and Digital Techniques*, 141(3), 177-183.

- Kong, Zhiwei, Zhang, Yong, Wang, Xudong, Xu, Yueyang, & Jin, Baosheng.** (2020). Prediction and optimization of a desulphurization system using CMAC neural network and genetic algorithm. *Journal of Environmental Engineering and Landscape Management*, 28(2), 74-87.
- Lane, Stephen H, Handelman, David A, & Gelfand, Jack J.** (1992). Theory and development of higher-order CMAC neural networks. *IEEE Control Systems Magazine*, 12(2), 23-30.
- Lee, Zne-Jung, Wang, Yan-Ping, & Su, Shun-Feng.** (2004). A genetic algorithm based robust learning credit assignment cerebellar model articulation controller. *Applied Soft Computing*, 4(4), 357-367.
- Li, Xin, Fang, Gui, & Qiu, Ya.** (2022). *Review of the cerebellar model articulation controller*. Paper presented at the 2022 41st Chinese Control Conference (CCC).
- Liang, Jing J, Qu, Bo Y, & Suganthan, Ponnuthurai N.** (2013). Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*, 635(2), 2014.
- Liu, Zhi-Zhong, Chu, Dian-Hui, Song, Cheng, Xue, Xiao, & Lu, Bao-Yun.** (2016). Social learning optimization (SLO) algorithm paradigm and its application in QoS-aware cloud service composition. *Information sciences*, 326, 315-333.
- Mendes, Rui, Kennedy, James, & Neves, José.** (2004). The fully informed particle swarm: simpler, maybe better. *IEEE transactions on evolutionary computation*, 8(3), 204-210.
- Meng, An-bo, Chen, Yu-cheng, Yin, Hao, & Chen, Si-zhe.** (2014). Crisscross optimization algorithm and its application. *Knowledge-based systems*, 67, 218-229.
- Mirjalili, Seyedali.** (2015a). The ant lion optimizer. *Advances in engineering software*, 83, 80-98.
- Mirjalili, Seyedali.** (2015b). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-based systems*, 89, 228-249.
- Mirjalili, Seyedali.** (2016a). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*, 27, 1053-1073.
- Mirjalili, Seyedali.** (2016b). SCA: a sine cosine algorithm for solving optimization problems. *Knowledge-based systems*, 96, 120-133.
- Mirjalili, Seyedali, Gandomi, Amir H, Mirjalili, Seyedeh Zahra, Saremi, Shahrzad, Faris, Hossam, & Mirjalili, Seyed Mohammad.** (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in engineering software*, 114, 163-191.
- Mirjalili, Seyedali, & Hashim, Siti Zaiton Mohd.** (2010). *A new hybrid PSOGSA algorithm for function optimization*. Paper presented at the 2010 international conference on computer and information application.
- Mirjalili, Seyedali, & Lewis, Andrew.** (2014). Adaptive gbest-guided gravitational search algorithm. *Neural Computing and Applications*, 25, 1569-1584.

- Mirjalili, Seyedali, & Lewis, Andrew.** (2016). The whale optimization algorithm. *Advances in engineering software*, 95, 51-67.
- Mirjalili, Seyedali, Mirjalili, Seyed Mohammad, & Lewis, Andrew.** (2014). Grey wolf optimizer. *Advances in engineering software*, 69, 46-61.
- Mittal, Himanshu, Pal, Raju, Kulhari, Ankur, & Saraswat, Mukesh.** (2016). *Chaotic kbest gravitational search algorithm (ckgsa)*. Paper presented at the 2016 ninth international conference on contemporary computing (IC3).
- Mittal, Himanshu, Tripathi, Ashish, Pandey, Avinash Chandra, & Pal, Raju.** (2021). Gravitational search algorithm: a comprehensive analysis of recent variants. *Multimedia Tools and Applications*, 80, 7581-7608.
- Moosavi, Vahid, Karami, Ayoob, & Aliramaee, Ramyar.** (2023). High-resolution soil moisture mapping using PSO-based optimized cerebellar model articulation controller (CMAC). *Science of The Total Environment*, 857, 159493.
- Mosa, Mohamed Atef.** (2019). Real-time data text mining based on Gravitational Search Algorithm. *Expert Systems with Applications*, 137, 117-129.
- Muhammad, Badaruddin, Ibrahim, Zuwairie, Mat Jusof, Mohd Falfazli, Ab Aziz, Nor Azlina, Abd Aziz, Nor Hidayati, & Mokhtar, Norrima.** (2017). *A hybrid simulated Kalman filter-gravitational search algorithm (SKF-GSA)*. Paper presented at the International conference on artificial life and robotics.
- Muthiah-Nakarajan, Venkataraman, & Noel, Mathew Mithra.** (2016). Galactic swarm optimization: a new global optimization metaheuristic inspired by galactic motion. *Applied Soft Computing*, 38, 771-787.
- Naik, Manoj Kumar, & Panda, Rutuparna.** (2015). A new hybrid CS-GSA algorithm for function optimization. *positions*, 1(1), 12.
- Neamah, Hasanain Jalil, Almobarqaa, Ali M, & Abdulhusien, Zainab Ali.** (2022). Gravitational evaluation algorithm for global optimization problem. *International Journal of Nonlinear Analysis and Applications*, 13(2), 345-359.
- Nezamoddini, Nasim, & Gholami, Amirhosein.** (2019). *Integrated genetic algorithm and artificial neural network*. Paper presented at the 2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC).
- Oladejo, Sunday O, Ekwe, Stephen O, & Mirjalili, Seyedali.** (2024). The Hiking Optimization Algorithm: A novel human-based metaheuristic approach. *Knowledge-based systems*, 296, 111880.
- Özden, Ayşenur, & İşeri, İsmail.** (2023). COOT optimization algorithm on training artificial neural networks. *Knowledge and Information Systems*, 65(8), 3353-3383.
- Özyön, Serdar, & Yaşar, Celal.** (2018). Gravitational search algorithm applied to fixed head hydrothermal power system with transmission line security constraints. *Energy*, 155, 392-407.
- Özyön, Serdar, Yaşar, Celal, & Temurtaş, Hasan.** (2019). Incremental gravitational search algorithm for high-dimensional benchmark functions. *Neural Computing and Applications*, 31, 3779-3803.
- Palacios, Francisco, Li, Xiaou, & Rocha, Luis E.** (2006). *Data mining based on CMAC neural networks*. Paper presented at the 2006 3rd International Conference on Electrical and Electronics Engineering.

- Pelusi, Danilo, Mascella, Raffaele, Tallini, Luca, Nayak, Janmenjoy, Naik, Bighnaraj, & Deng, Yong.** (2020). Improving exploration and exploitation via a hyperbolic gravitational search algorithm. *Knowledge-based systems, 193*, 105404.
- Peram, Thanmaya, Veeramachaneni, Kalyan, & Mohan, Chilukuri K.** (2003). *Fitness-distance-ratio based particle swarm optimization*. Paper presented at the Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No. 03EX706).
- Precup, Radu-Emil, David, Radu-Codrut, Petriu, Emil M, Preitl, Stefan, & Radac, Mircea-Bogdan.** (2012). Novel adaptive gravitational search algorithm for fuzzy controlled servo systems. *IEEE Transactions on Industrial Informatics, 8*(4), 791-800.
- Qin, Ting, Chen, Zonghai, Zhang, Haitao, Li, Sifu, Xiang, Wei, & Li, Ming.** (2004). A learning algorithm of CMAC based on RLS. *Neural Processing Letters, 19*, 49-61.
- Ramezani, Fatemeh, & Lotfi, Shahriar.** (2013). Social-based algorithm (SBA). *Applied Soft Computing, 13*(5), 2837-2856.
- Rao, R Venkata, Savsani, Vimal J, & Vakharia, Dipakkumar P.** (2011). Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-aided design, 43*(3), 303-315.
- Rashedi, Esmat, Nezamabadi-Pour, Hossein, & Saryazdi, Saeid.** (2009). GSA: a gravitational search algorithm. *Information sciences, 179*(13), 2232-2248.
- Rashedi, Esmat, Nezamabadi-Pour, Hossein, & Saryazdi, Saeid.** (2010). BGSA: binary gravitational search algorithm. *Natural computing, 9*, 727-745.
- Reynolds, Robert G.** (1994). *An introduction to cultural algorithms*. Paper presented at the Proceedings of the third annual conference on evolutionary programming.
- Rocca, Paolo, Oliveri, Giacomo, & Massa, Andrea.** (2011). Differential evolution as applied to electromagnetics. *IEEE Antennas and Propagation Magazine, 53*(1), 38-49.
- Rodan, Ali, Al-Tamimi, Abdel-Karim, Al-Alnemer, Loai, Mirjalili, Seyedali, & Tiño, Peter.** (2025). Enzyme action optimizer: a novel bio-inspired optimization algorithm. *The Journal of Supercomputing, 81*(5), 686.
- Saeidi-Khabisi, Fatemeh-sadat, & Rashedi, Esmat.** (2012). *Fuzzy gravitational search algorithm*. Paper presented at the 2012 2nd International eConference on Computer and Knowledge Engineering (ICCKE).
- Sattar, HA, & Cheetar, Alaa.** (2019). A new strategy based on GSABAT to solve single objective optimization problem. *International Journal of Swarm Intelligence Research (IJSIR), 10*(3), 1-22.
- Sedighizadeh, Davoud, Masehian, Ellips, Sedighizadeh, Mostafa, & Akbaripour, Hossein.** (2021). GEPSO: A new generalized particle swarm optimization algorithm. *Mathematics and Computers in Simulation, 179*, 194-212.
- Shami, Tareq M, El-Saleh, Ayman A, Alswaitti, Mohammed, Al-Tashi, Qasem, Summakieh, Mhd Amen, & Mirjalili, Seyedali.** (2022). Particle swarm optimization: A comprehensive survey. *IEEE Access, 10*, 10031-10061.

- Shehadeh, Hisham A.** (2021). A hybrid sperm swarm optimization and gravitational search algorithm (HSSOGSA) for global optimization. *Neural Computing and Applications*, 33(18), 11739-11752.
- Sheikhan, Mansour, & Jadidi, Zahra.** (2014). Flow-based anomaly detection in high-speed links using modified GSA-optimized neural network. *Neural Computing and Applications*, 24, 599-611.
- Shi, Yuhui.** (2011). *Brain storm optimization algorithm*. Paper presented at the Advances in Swarm Intelligence: Second International Conference, ICSI 2011, Chongqing, China, June 12-15, 2011, Proceedings, Part I 2.
- Shiraishi, Hitoshi, Ipri, Susan L, & Cho, D-ID.** (1995). CMAC neural network controller for fuel-injection systems. *IEEE Transactions on Control Systems Technology*, 3(1), 32-38.
- Simon, Dan.** (2008). Biogeography-based optimization. *IEEE transactions on evolutionary computation*, 12(6), 702-713.
- Singh, Amarjeet, & Deep, Kusum.** (2017). Hybridizing gravitational search algorithm with real coded genetic algorithms for structural engineering design problem. *Opsearch*, 54, 505-536.
- Sun, Tsung-Ying, Liu, Chan-Cheng, Tsai, Shang-Jeng, Hsieh, Sheng-Ta, & Li, Kan-Yuan.** (2010). Cluster guide particle swarm optimization (CGPSO) for underdetermined blind source separation with advanced conditions. *IEEE transactions on evolutionary computation*, 15(6), 798-811.
- Şenel, Fatih Ahmet, Gökçe, Fatih, Yüksel, Asım Sinan, & Yiğit, Tuncay.** (2019). A novel hybrid PSO–GWO algorithm for optimization problems. *Engineering with computers*, 35, 1359-1373.
- Talatahari, Siamak, & Azizi, Mahdi.** (2021). Chaos game optimization: a novel metaheuristic algorithm. *Artificial Intelligence Review*, 54(2), 917-1004.
- Tan, Ning, Li, Chaoyuan, Yu, Peng, & Ni, Fenglei.** (2022). Two model-free schemes for solving kinematic tracking control of redundant manipulators using CMAC networks. *Applied Soft Computing*, 126, 109267.
- Tanabe, Ryoji, & Fukunaga, Alex S.** (2014). *Improving the search performance of SHADE using linear population size reduction*. Paper presented at the 2014 IEEE congress on evolutionary computation (CEC).
- Tao, Ted, Lu, Hung-Ching, Hsu, Chau-Yun, & Hung, Ta-Hsiung.** (2003). The one-time learning hierarchical CMAC and the memory limited CA-CMAC for image data compression. *Journal of the Chinese Institute of Engineers*, 26(2), 133-145.
- Teddy, Sintiani Dewi, Lai, EM-K, & Quek, Chai.** (2007). Hierarchically clustered adaptive quantization CMAC and its learning convergence. *IEEE Transactions on Neural Networks*, 18(6), 1658-1682.
- Tian, Ken, Guo, Ben, Liu, Guoqing, Mitchell, Ian, Cheng, Dansong, & Zhao, Wei.** (2013). KCMAC-BYY: Kernel CMAC using Bayesian Ying–Yang learning. *Neurocomputing*, 101, 24-31.
- Tsai, Cheng-Hung, & Yeh, Ming-Feng.** (2009). Application of CMAC neural network to the control of induction motor drives. *Applied Soft Computing*, 9(4), 1187-1196.
- Tu, Ji, & Cao, Shaozhong.** (2012). PMSM Driver Based on Hybrid Particle Swarm Optimization and CMAC. *Physics Procedia*, 33, 983-990.

- Wang, Gai-Ge, Deb, Suash, & Coelho, Leandro dos S.** (2015). *Elephant herding optimization*. Paper presented at the 2015 3rd international symposium on computational and business intelligence (ISCBI).
- Yang, Xin-She.** (2009). *Firefly algorithms for multimodal optimization*. Paper presented at the International symposium on stochastic algorithms.
- Yang, Xin-She.** (2010). A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)* (pp. 65-74): Springer.
- Yang, Xin-She.** (2012). *Flower pollination algorithm for global optimization*. Paper presented at the International conference on unconventional computing and natural computation.
- Yang, Xin-She, & Deb, Suash.** (2009). *Cuckoo search via Lévy flights*. Paper presented at the 2009 World congress on nature & biologically inspired computing (NaBIC).
- Yao, Xin, Liu, Yong, & Lin, Guangming.** (1999). Evolutionary programming made faster. *IEEE transactions on evolutionary computation*, 3(2), 82-102.
- Zhang, Jing-Ru, Zhang, Jun, Lok, Tat-Ming, & Lyu, Michael R.** (2007). A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training. *Applied Mathematics and Computation*, 185(2), 1026-1037.
- Zhao, Fuqing, Xue, Feilong, Zhang, Yi, Ma, Weimin, Zhang, Chuck, & Song, Houbin.** (2018). A hybrid algorithm based on self-adaptive gravitational search algorithm and differential evolution. *Expert Systems with Applications*, 113, 515-530.
- Zheng, Yu-Jun.** (2015). Water wave optimization: a new nature-inspired metaheuristic. *Computers & operations research*, 55, 1-11.
- Zitouni, Farouq, Harous, Saad, & Maamri, Ramdane.** (2020). The solar system algorithm: a novel metaheuristic method for global optimization. *IEEE Access*, 9, 4542-4565.
- Zou, Dexuan, Liu, Haikuan, Gao, Liqun, & Li, Steven.** (2011). A novel modified differential evolution algorithm for constrained optimization problems. *Computers & mathematics with applications*, 61(6), 1608-1623.

## EKLER

### EK-1: CEC 2014 Test Fonksiyonları

Grup	Fonksiyon No	Fonksiyon Adı	$f(x^*)$
Unimodal Functions	F1	Rotated High Conditioned Elliptic Function	100
	F2	Rotated Bent Cigar Function	200
	F3	Rotated Discus Fonksiyon	300
Simple Multimodal Functions	F4	Shifted and Rotated Rosenbrock's Fonksiyon	400
	F5	Shifted and Rotated Ackley's Fonksiyon	500
	F6	Shifted and Rotated Weierstrass Fonksiyon	600
	F7	Shifted and Rotated Griewank's Fonksiyon	700
	F8	Shifted Rastrigin's Fonksiyon	800
	F9	Shifted and Rotated Rastrigin's Fonksiyon	900
	F10	Shifted Schwefel's Fonksiyon	1000
	F11	Shifted and Rotated Schwefel's Fonksiyon	1100
	F12	Shifted and Rotated Katsuura Fonksiyon	1200
	F13	Shifted and Rotated HappyCat Fonksiyon	1300
	F14	Shifted and Rotated HGBat Fonksiyon	1400
	F15	Shifted and Rotated Expanded Griewank's plus Rosenbrock's Fonksiyon	1500
	F16	Shifted and Rotated Expanded Scaffer's F6 Fonksiyon	1600
Hybrid Fonksiyonlar	F17	Hybrid Function 1 (N=3)	1700
	F18	Hybrid Function 2 (N=3)	1800
	F19	Hybrid Function 3 (N=4)	1900
	F20	Hybrid Function 4 (N=4)	2000
	F21	Hybrid Function 5 (N=5)	2100
	F22	Hybrid Function 6 (N=5)	2200
Composition Fonksiyonları	F23	Composition Fonksiyon 1 (N=5)	2300
	F24	Composition Fonksiyon 2 (N=3)	2400
	F25	Composition Fonksiyon 3 (N=3)	2500
	F26	Composition Fonksiyon 4(N=5)	2600
	F27	Composition Fonksiyon 5 (N=5)	2700
	F28	Composition Fonksiyon 6 (N=5)	2800
	F29	Composition Fonksiyon 7(N=3)	2900
	F30	Composition Fonksiyon 8(N=3)	3000