



**T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI**

**EEG SİNYALLERİNDEN GÜRÜLTÜNÜN TEMİZLENMESİ
İÇİN HARMONİ ARAMA ALGORİTMASI TEMELLİ
YÖNTEMLERİN GELİŞTİRİLMESİ**

Yüksek Lisans Tezi

Serhat Celil İLERİ

Danışman
Dr. Öğr. Üyesi Sercan DEMİRCİ

SAMSUN
2023

T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI



EEG SİNYALLERİNDEN GÜRÜLTÜNÜN TEMİZLENMESİ
İÇİN HARMONİ ARAMA ALGORİTMASI TEMELLİ
YÖNTEMLERİN GELİŞTİRİLMESİ

Yüksek Lisans Tezi

Serhat Celil İLERİ

Danışman
Dr. Öğr. Üyesi Sercan DEMİRCİ

SAMSUN
2023

TEZ KABUL VE ONAYI

Serhat Celil İLERİ tarafından, Dr. Öğr. Üyesi Sercan DEMİRCİ danışmanlığında hazırlanan “EEG SİNYALLERİNDEN GÜRÜLTÜNÜN TEMİZLENMESİ İÇİN HARMONİ ARAMA ALGORİTMASI TEMELLİ YÖNTEMLERİN GELİŞTİRİLMESİ” başlıklı bu çalışma, jürimiz tarafından 25.8.2023 tarihinde yapılan sınav sonucunda oy birliği ile başarılı bulunarak Yüksek Lisans Tezi olarak kabul edilmiştir.

	Unvanı Adı Soyadı Üniversitesi Ana Bilim/Ana Sanat Dalı	Sonuç
Başkan	Doç. Dr. Tansel DÖKEROĞLU Çankaya Üniversitesi Yazılım Mühendisliği Bölümü	☒ Kabul ☐ Ret
Üye	Dr. Öğr. Üyesi Sercan DEMİRCİ Ondokuz Mayıs Üniversitesi Bilgisayar Mühendisliği Ana Bilim Dalı	☒ Kabul ☐ Ret
Üye	Dr. Öğr. Üyesi Durmuş Özkan ŞAHİN Ondokuz Mayıs Üniversitesi Bilgisayar Mühendisliği Ana Bilim Dalı	☒ Kabul ☐ Ret

Bu tez, Enstitü Yönetim Kurulunca belirlenen ve yukarıda adları yazılı jüri üyeleri tarafından uygun görülmüştür.

Prof. Dr. Ahmet TABAK
Enstitü Müdürü

BİLİMSEL ETİĞE UYGUNLUK BEYANI

Hazırladığım Yüksek Lisans tezinin bütün aşamalarında bilimsel etiğe ve akademik kurallara riayet ettiğimi, çalışmada doğrudan veya dolaylı olarak kullandığım her alıntıya kaynak gösterdiğimi ve yararlandığım eserlerin Kaynaklar'da gösterilenlerden oluştuğunu, her unsurun enstitü yazım kılavuzuna uygun yazıldığını ve TÜBİTAK Araştırma ve Yayın Etiği Kurulu Yönetmeliği'nin 3. bölüm 9. maddesinde belirtilen durumlara aykırı davranılmadığını taahhüt ve beyan ederim.

Etik Kurul Gerekli mi ?

Evet ☐

Hayır ☒

05/09/2023

Serhat Celil İLERİ

TEZ ÇALIŞMASI ÖZGÜNLÜK RAPORU BEYANI

Tez Başlığı: EEG SİNYALLERİNDEN GÜRÜLTÜNÜN TEMİZLENMESİ İÇİN HARMONİ ARAMA ALGORİTMASI TEMELLİ YÖNTEMLERİN GELİŞTİRİLMESİ

Yukarıda başlığı belirtilen tez çalışması için şahsım tarafından 03.07.2023 tarihinde intihal tespit programından alınmış olan özgünlük raporu sonucunda;

Benzerlik oranı : %5

Tek kaynak oranı : %1 çıkmıştır.

05/09/2023

Dr. Öğr. Üyesi Sercan DEMİRCİ

ÖZET

EEG SİNYALLERİNDEN GÜRÜLTÜNÜN TEMİZLENMESİ İÇİN HARMONİ ARAMA ALGORİTMASI TEMELLİ YÖNTEMLERİN GELİŞTİRİLMESİ

Serhat Celil İLERİ

Ondokuz Mayıs Üniversitesi

Lisansüstü Eğitim Enstitüsü

Bilgisayar Mühendisliği Ana Bilim Dalı

Yüksek Lisans, Eylül/2023

Danışman: Dr. Öğr. Üyesi Sercan DEMİRCİ

Elektro-ensefalogram (EEG) taraması insan beynindeki elektriksel aktivitenin, kişinin kafatasına yerleştirilen çok sayıda elektrod vasıtasıyla ölçülmesine dayanan bir tıbbi görüntüleme yöntemidir. Beyinde gerçekleşen işlevler sırasında nöronlar arası meydana gelen elektriksel aktivite beyin sağlığı açısından anlamlı bilgiler edinmeyi sağlar. EEG taraması uygun koşullarda ve uzmanlar tarafından yönlendirilerek gerçekleşse de ölçülen sinyaller dış etkenlerden veya ölçümü yapılan kişinin vücudundaki diğer aktivitelerden kaynaklanan gürültülere maruz kalır. Bu gürültülerin temizlenmesi sinyallerin yorumlanması açısından önem taşımaktadır.

Harmoni Arama (HS), insan kulağına gelen farklı seslerin bir araya gelerek oluşturduğu uyumun verdiği musiki tatmini matematiksel olarak modelleyerek en uygun çözümü üretmeyi amaçlayan bir yarı-sezgisel algoritmadır. Zaman içerisinde birçok araştırmacı tarafından HS algoritmasının parametrelerinin daha iyi belirlenmesi ve doğaçlama işleyişinin değiştirilmesi amacı ile çok sayıda HS varyantı literatüre sunulmuştur.

Bu tez kapsamında, HS varyantlarının bir büyük veri optimizasyon problemi olan EEG sinyallerinin gürültülerden temizlenmesi ve yaygın kıyaslama fonksiyonları üzerindeki başarımları ele alınmıştır. Mevcut HS varyantlarının yanı sıra Trigonometrik Harmoni Arama (TRI-HS) adı verilen yeni bir varyant geliştirilmiş ve belirlenen problemler üzerinde diğer varyantlar ile performans değerlendirmesi yapılmıştır.

Anahtar Sözcükler: EEG, Harmoni arama algoritması, Büyük veri optimizasyonu

ABSTRACT

DEVELOPMENT OF THE HARMONY SEARCH ALGORITHM BASED TECHNIQUES FOR REMOVING NOISE FROM EEG SIGNALS

Serhat Celil İLERİ

Ondokuz Mayıs University

Institute of Graduate Studies

Department of Computer Engineering

Master, September/2023

Supervisor: Asst. Prof. Dr. Sercan DEMİRCİ

Electro-encephalogram scanning is a medical imaging technique based on measurement of the electrical activity in human brain by deploying many electrodes to the skull of patient. Inter-neurons electrical activity during the brain functions, provides meaningful information on the health of the brain. Even though EEG scanning is made in appropriate conditions and directed by experts, measured signals are exposed to the external factors or other activities of the body of the patient. Cleaning the signals from the noises are crucial for interpretation.

Harmony Search (HS) algorithm is a meta-heuristic algorithm that aims to produce the most appropriate solution by mathematically modeling the musical satisfaction given by the harmony created by the combination of different sounds in the human ear. Over time, many HS variants have been presented to the literature by many researchers in order to better adjust the parameters and change the improvisation mechanism of HS.

In this thesis, the performance of HS variants on noise removal from EEG signals which is a big data optimization problem and common benchmark functions is discussed. Besides the existing HS variants, a new variant called Trigonometric Harmony Search (TRI-HS) is developed and its performance is evaluated on dedicated problems with other variants.

Keywords: EEG, Harmony search algorithm, Big data optimization

İÇİNDEKİLER

TEZ KABUL VE ONAYI.....	i
BİLİMSEL ETİĞE UYGUNLUK BEYANI	ii
TEZ ÇALIŞMASI ÖZGÜNLÜK RAPORU BEYANI.....	ii
ÖZET	iii
ABSTRACT	iv
İÇİNDEKİLER	v
SİMGELER VE KISALTMALAR	vi
ŞEKİLLER DİZİNİ	ix
TABLolar DİZİNİ.....	xi
1. GİRİŞ.....	1
1.1. Çalışmanın Amacı	2
1.2. Problem Tanımı	3
1.3. Katkılar	3
1.4. Tezin Organizasyonu	4
2. GENEL BİLGİLER.....	5
2.1. Optimizasyon Algoritmaları.....	5
2.1.1. Deterministik Optimizasyon Algoritmaları.....	7
2.1.2. Stokastik Optimizasyon Algoritmaları.....	7
2.1.2.1. Sezgisel Algoritmalar	7
2.1.2.2. Yarı-Sezgisel Algoritmalar	8
2.1.2.3. Üst-Sezgisel Algoritmalar	8
2.1.3. Optimizasyon Amaç Fonksiyonları.....	8
2.1.3.1. Tek Amaçlı Optimizasyon	9
2.1.3.2. Çok Amaçlı Optimizasyon	9
2.1.4. EEG Sinyallerinin Gürültüden Temizlenmesi Problemi.....	10
3. MATERYALLER VE YÖNTEMLER	13
3.1. Harmoni Arama (HS) Algoritması	13
3.1.1. Harmoni Arama Algoritması Varyantları	16
3.1.1.1. Geliştirilmiş Harmoni Arama Algoritması (IHS)	19
3.1.1.2. Küresel En İyi Harmoni Arama Algoritması (GHS)	20
3.1.1.3. Değiştirilmiş Harmoni Arama Algoritması (MHS).....	21
3.1.1.4. Kendi-Uyarlamalı Küresel En İyi Harmoni Arama Algoritması (SGHS) ..	21
3.1.1.5. Kaşif Harmoni Arama Algoritması (EHS).....	22
3.1.1.6. Yeni Bir Küresel Harmoni Arama Algoritması (NGHS)	23
3.1.1.7. Önerilen HS (PHS) ve İyileştirilen Önerilen HS Algoritması (IPHS)..	23
3.1.1.8. Zeki Ayarlı Harmoni Arama Algoritması (ITHS)	23
3.1.1.9. Dinamik Kontrol Parametrelili Harmoni Arama Algoritması (NDHS)..	24
3.1.1.10.Kendinden-Uyarlamalı Harmoni Arama Algoritması (NSHS).....	25
3.1.1.11.Dinamik Kendinden-Uyarlamalı Harmoni Arama Algoritması (DSHS)	25
3.1.1.12.Parametre Uyarlamalı Harmoni Arama Algoritması (PAHS)	27
3.1.1.13.Küresel Dinamik Harmoni Arama Algoritması (GDHS).....	27
3.1.1.14.Kendinden Uyarlamalı Geliştirilmiş HS Algoritması (SIHS)	29
3.1.1.15.Bulanık Harmoni Arama Algoritması (FHS).....	29
3.1.1.16.Yerel Karşıt Tabanlı Kendinden Uyarlamalı Küresel HS (LHS).....	30
3.1.1.17.Yeni Bir Geliştirilmiş Harmoni Arama Algoritması (NEHS)	34
3.1.1.18.Bilgi Geri Beslemeli Kendinden-Uyarlamalı HS (IFSHS)	34
3.1.1.19.En İyi - En Kötü - Ortalama HS Algoritması (BWMHS)	34
3.1.1.20.Doğrusal Dinamik Alan ile İyileştirilmiş Diferansiyel Tabanlı HS	35

3.1.2. Trigonometrik Harmoni Arama Algoritması (TRI-HS)	36
3.1.2.1. Dinamik Sinüzoidal HMCR	37
3.1.2.2. Küresel Ton Ayarlama.....	39
3.1.2.3. Rekabet Seçim Şeması	39
4. BULGULAR VE DEĞERLENDİRME.....	42
4.1. Kıyaslama Amaç Fonksiyonları	42
4.2. Deney Parametreleri.....	45
4.3. Deneysel Çıktılar.....	49
4.3.1. Kıyaslama Fonksiyonları Deneysel Çıktıları	49
4.3.2. EEG Optimizasyonu Deneysel Çıktıları.....	61
5. SONUÇLAR VE ÖNERİLER	77
KAYNAKLAR	79
ÖZGEÇMİŞ	82



SİMGELER VE KISALTMALAR

HS	: Harmoni Arama (Harmony Search)
BCI	: Beyin-Bilgisayar Arayüzü (Brain-Computer Interface)
EEG	: Elektro-Ensefalogram
IPA	: İmmün Plazma Algoritması (Immune Plasma Algorithm)
EA	: Evrimsel Algoritmalar (Evolutionary Algorithms)
TS	: Tabu Arama (Tabu Search)
ABC	: Yapay Arı Kolonisi (Artificial Bee Colony)
FWA	: Havai Fişek Algoritması (Fireworks Algorithm)
HM	: Harmoni Hafızası (Harmony Memory)
HMCR	: Harmoni Hafızasından Yararlanma Oranı (HM Consideration Rate)
PAR	: Ton Ayarlama Oranı (Pitch Adjusting Rate)
bw	: Bant genişliği (Bandwidth)
PSO	: Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)
DE	: Diferansiyel Gelişim (Differential Evolution)
OBE	: Karşıt Tabanlı DE (Opposition Based DE)
ACO	: Karınca Koloni Optimizasyonu (Ant Colony Optimization)
GSO	: Grup Arama Optimizasyonu (Group Search Optimization)
IHS	: Geliştirilmiş HS (Improved HS)
GHS	: Küresel En-İyi HS (Global-Best HS)
MHS	: Değiştirilmiş HS (Modified HS)
SGHS	: Kendi-Uyarlamalı Küresel-En İyi HS (Self-Adaptive Global-Best HS)
EHS	: Kaşif HS (Explorative HS)
NGHS	: Yeni Bir Küresel HS (A Novel Global HS)
PHS	: Önerilen HS (Proposed HS)

IPHS	: İyileştirilen Önerilen HS (Improving Proposed HS)
ITHS	: Zeki Ayarlı HS (Intelligent Tuned HS)
NDHS	: Dinamik Kontrol Parametreli HS (HS Algorithm with Dynamic Parameters)
NSHS	: Yeni Kendinden-Uyarlamalı HS (Novel Self-Adaptive HS)
DSHS	: Dinamik Kendinden-Uyarlamalı HS (Dynamic Self-Adaptive HS)
PAHS	: Parametre Uyarlamalı HS (Parameter Adaptive HS)
GDHS	: Küresel Dinamik HS (Global Dynamic HS)
SIHS	: Kendinden-Uyarlamalı Geliştirilmiş HS (Self-Adaptive Improved HS)
FHS	: Bulanık HS (Fuzzy HS)
LHS	: Yerel Karşıt Tabanlı Kendinden-Uyarlamalı Küresel HS (Local Opposition-Based Self-Learning Global HS)
NEHS	: Yeni Bir Geliştirilmiş HS (A New Enhanced HS)
IFSHS	: Bilgi Geri Beslemeli Kendinden-Uyarlamalı HS (Information Feedback Self-Adaptive HS)
BWMHS	: En İyi-En Kötü-Ortalama HS (Best-Worst-Mean HS)
ID-HS-LDD	: Doğrusal Dinamik Alan ile İyileştirilmiş Diferansiyel Tabanlı HS (Improved Differential-Based HS Algorithm with Linear Dynamic Domain)
TRI-HS	: Trigonometrik HS (Trigonometric HS)

ŞEKİLLER DİZİNİ

Şekil 3.1.	HS algoritmasının akış diyagramı	17
Şekil 3.2.	IHS algoritmasında PAR ve bw değerlerinin değişimi	20
Şekil 3.3.	PAHS algoritması $HMCR$ ve PAR değerleri	28
Şekil 3.4.	Sinüzoidal $HMCR$ değeri gösterimi.....	38
Şekil 3.5.	TRI-HS algoritmasının akış diyagramı	41
Şekil 4.1.	Deneylerde kullanılan parametrelerle elde edilen $HMCR$ değerleri	49
Şekil 4.2.	Sphere fonksiyonu için HS varyantlarının yakınsama grafiği ($D = 30$).....	51
Şekil 4.3.	Rosenbrock fonksiyonu için HS varyantlarının yakınsama grafiği	51
Şekil 4.4.	Rastrigin fonksiyonu için HS varyantlarının yakınsama grafiği.....	53
Şekil 4.5.	Ackley fonksiyonu için HS varyantlarının yakınsama grafiği.....	55
Şekil 4.6.	Griewank fonksiyonu için HS varyantlarının yakınsama grafiği	55
Şekil 4.7.	Schwefel 2.22 fonksiyonu için HS varyantlarının yakınsama grafiği	56
Şekil 4.8.	Generalized Schwefel 2.26 fonksiyonu için HS varyantlarının yakınsama grafiği	58
Şekil 4.9.	Step fonksiyonu için HS varyantlarının yakınsama grafiği	59
Şekil 4.10.	Rotated Hyper Ellipsoid fonksiyonu için HS varyantlarının yakınsama grafiği	61
Şekil 4.11.	Shifted Sphere fonksiyonu için HS varyantlarının yakınsama grafiği	61
Şekil 4.12.	Gürültüsüz 4 Kanallı EEG problemi için yakınsama grafiği (HMS: 5)	64
Şekil 4.13.	Gürültüsüz 4 Kanallı EEG problemi için kutu grafiği (HMS: 5)	64
Şekil 4.14.	Gürültüsüz 12 Kanallı EEG problemi için yakınsama grafiği (HMS: 5).....	65
Şekil 4.15.	Gürültüsüz 12 Kanallı EEG problemi için kutu grafiği (HMS: 5).....	65
Şekil 4.16.	Gürültüsüz 19 Kanallı EEG problemi için yakınsama grafiği (HMS: 5).....	66
Şekil 4.17.	Gürültüsüz 19 Kanallı EEG problemi için kutu grafiği (HMS: 5).....	66
Şekil 4.18.	Gürültüsüz 4 kanallı EEG problemi için yakınsama grafiği (HMS: 10)	67
Şekil 4.19.	Gürültüsüz 4 kanallı EEG problemi için kutu grafiği (HMS: 10)	68
Şekil 4.20.	Gürültüsüz 12 kanallı EEG problemi için yakınsama grafiği (HMS: 10).....	68
Şekil 4.21.	Gürültüsüz 12 kanallı EEG problemi için kutu grafiği (HMS: 10).....	68
Şekil 4.22.	Gürültüsüz 19 kanallı EEG problemi için yakınsama grafiği (HMS: 10).....	69
Şekil 4.23.	Gürültüsüz 19 kanallı EEG problemi için kutu grafiği (HMS: 10).....	69
Şekil 4.24.	Gürültü eklenmiş 4 Kanallı EEG problemi için yakınsama grafiği (HMS: 5) .	71
Şekil 4.25.	Gürültü eklenmiş 4 Kanallı EEG problemi için kutu grafiği (HMS: 5)	71
Şekil 4.26.	Gürültü eklenmiş 12 Kanallı EEG problemi için yakınsama grafiği (HMS: 5)	71
Şekil 4.27.	Gürültü eklenmiş 12 Kanallı EEG problemi için kutu grafiği (HMS: 5)	72
Şekil 4.28.	Gürültü eklenmiş 19 Kanallı EEG problemi için yakınsama grafiği (HMS: 5)	72

Şekil 4.29.	Gürültü eklenmiş 19 Kanallı EEG problemi için kutu grafiği (HMS: 5)	72
Şekil 4.30.	Gürültü eklenmiş 4 kanallı EEG problemi için yakınsama grafiği (HMS: 10)	74
Şekil 4.31.	Gürültü eklenmiş 4 kanallı EEG problemi için kutu grafiği (HMS: 10)	74
Şekil 4.32.	Gürültü eklenmiş 12 kanallı EEG problemi için yakınsama grafiği (HMS: 10)	75
Şekil 4.33.	Gürültü eklenmiş 12 kanallı EEG problemi için kutu grafiği (HMS: 10).....	75
Şekil 4.34.	Gürültü eklenmiş 19 kanallı EEG problemi için yakınsama grafiği (HMS: 10)	75
Şekil 4.35.	Gürültü eklenmiş 19 kanallı EEG problemi için kutu grafiği (HMS: 10).....	76



TABLÖLAR DİZİNİ

Tablo 4.1. HS Parametreleri	45
Tablo 4.2. IHS Parametreleri	46
Tablo 4.3. GHS Parametreleri	46
Tablo 4.4. MHS Parametreleri	46
Tablo 4.5. SAHS Parametreleri	47
Tablo 4.6. NGHS Parametreleri	47
Tablo 4.7. PAHS Parametreleri	47
Tablo 4.8. LHS Parametreleri	47
Tablo 4.9. BWMHS Parametreleri	48
Tablo 4.10. TRI-HS Parametreleri	48
Tablo 4.11. Sphere fonksiyonu HS varyant sonuçları	50
Tablo 4.12. Rosenbrock fonksiyonu HS varyant sonuçları	52
Tablo 4.13. Rastrigin fonksiyonu HS varyant sonuçları	53
Tablo 4.14. Ackley fonksiyonu HS varyant sonuçları	54
Tablo 4.15. Griewank fonksiyonu HS varyant sonuçları	56
Tablo 4.16. Schwefel 2.22 fonksiyonu HS varyant sonuçları	57
Tablo 4.17. Generalized Schwefel 2.26 fonksiyonu HS varyant sonuçları	58
Tablo 4.18. Step fonksiyonu HS varyant sonuçları	59
Tablo 4.19. Rotated Hyper Ellipsoid fonksiyonu HS varyant sonuçları	60
Tablo 4.20. Shifted Sphere fonksiyonu HS varyant sonuçları	62
Tablo 4.21. Gürültüsüz EEG Problemi HS varyant sonuçları (HMS: 5)	63
Tablo 4.22. Gürültüsüz EEG problemi HS varyant sonuçları (HMS: 10)	67
Tablo 4.23. Gürültü eklenmiş EEG Problemi sonuçları (HMS: 5)	70
Tablo 4.24. Gürültü eklenmiş EEG Problemi sonuçları (HMS: 10)	73

1. GİRİŞ

Bilgi, insanlık için tarih boyunca hayatta kalmak ve gelişim elde etmek için kilit rolü oynamıştır. Bilgi edinebilen, farklı kaynaklardan edindiği bilgileri bir araya getirerek değerlendirme ve çözümleme yapabilen, elde edilen sonuca göre karar vererek hareket eden birey ve topluluklar başarıya ulaşma ve tehlikeleri bertaraf edebilme hususunda çok daha başarılı olabilmişlerdir. İnsanların edindiği bilgileri unutmamak ve gelecek nesillere aktarmak için kaydetmeye başlamasının ilk örneklerini, antik çağlarda yaşadıkları mağaraların duvarlarına veya avladıkları hayvanların derilerine çizdikleri resim ve şekillerde görebiliyoruz. Daha sonraki süreçte ise hiyeroglifler, çivi yazıları ve alfabeler kullanarak edinilen deneyim ve bilgi birikimini kaydetmek için sistematik yöntemler geliştirilmiştir.

Bilişim çağı olarak adlandırılan günümüzde ise neredeyse her şey kaydedilmesi ve işlenmesi gereken veri olarak değerlendirilmektedir. 20. yüzyılın ikinci yarısında kısıtlı sayıda nokta arasında veri iletimiyle temelleri atılan (Kleinrock, 2010) internet ağı günümüzde Nesnelerin İnterneti konseptinin de yaygınlaşmasıyla milyarlarca cihazın bağlandığı devasa bir ağ haline gelmiştir. Çok ciddi miktarlarda altyapı yatırımı ve AR-GE çalışmasının yürütüldüğü bu alanda 2023 yılı itibariyle sadece nesnelerin interneti için harcanması beklenen miktarın 1 trilyon Amerikan Doları eşliğini aşması beklenmektedir (Wójcicki vd., 2022).

Veri üretimi ve kaydının durmaksızın ve çok fazla kaynaktan olmaya başlaması elde edilen verilerin işleme ve saklanma yöntemlerinin yetersiz kalmaya başlamasına ve süreçle birlikte değişmesine neden oldu. Son yıllarda büyük veri olarak adlandırılan ve beraberinde birçok zorluğu da getiren bu konu birçok yeni araştırma alanını da literatüre kazandırdı. Çeşitli problemlere en iyi çözümleri üretmek için kullanılan optimizasyon yöntemleri de bu süreçte gelişim ve çeşitlilik açısından ilerleme kat etmeye devam etmiştir. Büyük verinin daha kolay işlenebilmesi için optimize edilmesine ihtiyaç duyulduğunda ise kullanılan yöntemin gerektirdiği hesaplama maliyeti verinin büyüklüğü ile birlikte arttığı için klasik yöntemler günümüz ihtiyaçlarını karşılamakta yetersiz kalmaktadır.

Makine öğrenimi ve yapay zeka yöntemlerinde çoğunlukla doğadaki süreçler ve doğal fenomenler taklit edilmektedir. İnsan beyninin çalışmasını modelleyen nöral ağlar ve sürü halinde yaşayan canlıların davranışlarını modelleyen sürü zekası yöntemleri gibi teknikler diğer birçok optimizasyon probleminin yanı sıra büyük veri üzerinde de başarımlar elde etme potansiyeline sahiptir. İnsan hayatı ve sağlıkla ilgili konulara doğrudan temas edebilen büyük veri optimizasyonu problemlerinde başarılı optimizasyon yöntemlerinin tespit edilmesi önemli bir kazanım olacaktır.

Optimizasyon işlemlerinde başarılı sonuçlar üretilebilmesindeki anahtar rolü oynayan etkenlerden bir tanesi de problemin doğasını anlamak ve doğru bir yaklaşım yöntemi seçebilmektir. Doğru değer aralıklarıyla çalışmak, kullanılan yöntemin parametrelerini probleme uygun şekilde belirleyebilmek ve verilen girdilerden etkin şekilde yararlanabilmek daha başarılı çözümlere ulaşmayı kolaylaştırır. Optimizasyon amacıyla kullanılan algoritmaları tanımak, içerdiği yöntemlerin çözüm üretimindeki etkisine hakim olmanın yanı sıra üzerinde çalışılan probleme uygun şekilde kaliteli çözüme yaklaşmayı kolaylaştıran yöntemleri seçerek bir araya getirmek başarımlar ve yakınsama hızına önemli ölçüde etki edebilir.

1.1. Çalışmanın Amacı

Beyin-bilgisayar arayüzü veya beyin-makine arayüzü, canlıların beyni ile dijital cihazlar arasındaki bağlantıyı sağlayan arayüz olarak nitelendirilir ve genellikle medikal cihazlar tarafından kullanılır (Meselhi vd., 2021). Hesaplamalı nörobilim çalışmalarında, beyin-bilgisayar arayüzü kullanılarak insan beyninin dijital olarak görüntülenmesi ve analizi ile beyindeki rahatsızlıkların tespit ve tedavisi amaçlanmaktadır. EEG taraması insan beynindeki elektriksel aktivitenin, kişinin kafatasına yerleştirilen çok sayıda elektrod vasıtasıyla ölçümlemesine dayanan bir tıbbi görüntüleme yöntemidir (Sanei & Chambers, 2013). Beyinde gerçekleşen aktiviteler sırasında nöronlar arası meydana gelen elektriksel aktivite beyin sağlığı açısından önemli bilgiler edinmeyi sağlar.

EEG taraması, genellikle hastane ortamında uygun koşullarda ve bir uzman tarafından yönlendirilmesiyle gerçekleşse de ölçümlenen sinyaller dış etkenlerden veya ölçümü yapılan kişinin vücudundaki diğer aktivitelerden kaynaklanan gürültülere maruz kalır. Artifakt adı verilen ve tarama sinyallerinde meydana gelen bu

gürültüler EEG sinyalinin analizini zorlaştırmaktadır. Sağlıklı bir tıbbi değerlendirme yapılabilmesi için EEG tarama sinyallerinin mümkün olduğunca artifakt içermemesi gerekir. Bu amaçla kaydedilen sinyallerin artifakt adlı gürültülerden ayrıştırılması gerekir.

Ortalama olarak 60 dakika süren EEG taramasının yanı sıra bazı taramalar için hastanın uyur haldeyken tarama yapılması gerekebilir, bu durumda yapılan taramalar daha da uzun olabilmektedir. Yapılan taramanın saniyede alınan örnek sayısı ve sinyal alınan farklı kanal sayısı göz önünde bulundurulduğunda tarama süresinin uzunluğuna göre elde edilen sinyal verisi de büyümektedir. Bu nedenle EEG sinyallerinin gürültüden arındırılması bir büyük veri problemi olarak ele alınabilir (*BigOpt*, 2015).

Harmoni Arama (HS) Algoritması (Geem vd., 2001) ve varyantlarının EEG sinyallerinin gürültüden temizlenmesine dayalı büyük veri optimizasyon problemi üzerindeki başarımı incelenecektir.

1.2. Problem Tanımı

Büyük veri optimizasyonu (*BigOpt*, 2015) yarışması kapsamında örnek EEG sinyalleri ve gürültü eklenmiş halleri optimize edilmek istenilen amaç fonksiyonuyla birlikte araştırmacıların kullanımına sunulmuştur. Bu problem kapsamında 1024, 3072 ve 4864 parametreye sahip ham ve gürültü eklenmiş EEG sinyallerinin artifaktlardan arındırılması amaçlanmaktadır. Bu işlem ile kaynak EEG sinyalinin işlenmesi ve analizi kolaylaştırılmakta, bu sırada EEG taramasının teşhis yapmaya olanak sağlayan anlamlı bilgileri de korunmaktadır.

1.3. Katkılar

Literatürde bulunan HS varyantları incelenerek orijinal algoritma üzerinde yapılan iyileştirmeler ve işleyiş değişiklikleri incelenmiştir. Yapılan incelemeler ve elde edilen deneysel sonuçlar göz önünde bulundurularak çok parametrelili problemler üzerinde daha başarılı sonuçlar üretebilecek yeni bir varyant geliştirme çalışması yürütülmüştür. Bu çalışmalar kapsamında Trigonometrik Harmoni Arama (TRI-HS) adı verilen yeni bir HS varyantı bu tez çalışması kapsamında literatüre sunulmaktadır.

1.4. Tezin Organizasyonu

Bu tez 5 bölümden oluşmaktadır. İkinci bölüm, giriş bölümünü takiben “Literatür Taraması” olup bu bölümde optimizasyon algoritmalarının çeşitleri, yapıları ve kullanım alanlarından detaylı şekilde bahsedilmektedir. Deterministik ve stokastik olarak iki sınıfta toplanan optimizasyon algoritmalarının bu kapsamda değerlendirilen türleri açıklanmaktadır. Ardından optimizasyon amaç fonksiyonlarından ve türlerinden bahsedilerek, bu tez çalışması kapsamında optimize edilmek istenilen Büyük Veri Optimizasyonu başlığı altındaki EEG sinyallerinin gürültüden temizlenmesi problemi ele alınacaktır.

Üçüncü bölüm “Materyaller ve Yöntemler” olup, Harmoni arama algoritmasını ve zaman içerisinde bu algoritma temel alınarak geliştirilen varyantlarının detaylı incelemesi, açıklaması, güçlü ve zayıf yönlerinin araştırılmasını içermektedir. Ayrıca tez çalışmaları kapsamında geliştirilen yeni bir yöntem olan TRI-HS Algoritması açıklanmaktadır.

Dördüncü bölüm HS algoritması, varyantları ve yeni geliştirilen TRI-HS’nin yaygın ölçümleme fonksiyonları ve EEG sinyallerinin gürültüden temizlenmesi problemi üzerinde elde ettiği sonuçları içermektedir.

Son bölüm, tez çalışmaları kapsamında elde edilen bulguları ve değerlendirmeleri sunmakta, ilerleyen süreçte yapılabilecek çalışmalara yönelik öneriler getirmektedir.

2. GENEL BİLGİLER

Bu bölümde optimizasyon algoritmalarından bahsedilecek, optimizasyon problemlerinde kullanılan amaç fonksiyonlarının altyapısı verilecek ve bu tez çalışması kapsamında optimize edilmek istenilen EEG sinyallerinin gürültüden temizlenmesi problemi açıklanacaktır.

2.1. Optimizasyon Algoritmaları

Optimizasyon, belirli bir problem için uygun sonuçlar arasından en iyisinin bulunması anlamına gelmektedir (Arora, 2015). Belirli komut dizilerini ve karar verme mekanizmalarını istenilen sırayla çalıştırarak girdi olarak verilen problemler için en iyi çözümü mümkün olan en kısa sürede, en az deneme yaparak bulmayı amaçlayan algoritmalar ise optimizasyon algoritması olarak adlandırılmaktadır. Optimizasyon problemleri çok farklı yapıda, zorlukta ve karmaşıklıkta karşımıza çıkmaktadır. Bazı problemler tek bir parametrenin belirlenmesini amaçlamakta iken bazı problemler çok yüksek sayılarda parametrenin bir arada doğru şekilde belirlenmesini gerektirmektedir. Bunun yanında dinamik parametre sayısına sahip problemler de bulunmakta olup parametre sayısının artmasıyla birlikte optimize edilmesi de o denli zorlaşmaktadır.

Optimizasyon problemlerinden bazıları, özellikle düşük parametre sayısına sahip ya da formülize edilebilenler, yeterli teknik ve teorik bilgiye sahip uzmanlar tarafından kolaylıkla veya bir miktar üzerinde durularak en optimize sonuca ulaştırılabilmektedir. Ancak birçok gerçek hayat problemi, uzmanlardan oluşan bir komisyon tarafından çözülmek için bile fazlasıyla karmaşık olabilmektedir. Söz konusu problemler maliyetin düşürülmesi, işlem süresinin kısaltılması, doğru karar alınması gibi kritik öneme sahip olabilen, kısıtlı zaman aralığında çözümlenmesi ihtiyacı olan durumlar olarak karşımıza çıkabilir. Elde edilecek çözümün kalitesinin yüksek olması birincil amaç olsa da kabul edilebilir kalitedeki çözüme istenilen vakitte ulaşılamaması geri dönüşü olmayan olumsuz durumların oluşmasına neden olabilir. Özellikle doğa şartları ve insan başta olmak üzere herhangi bir canlının sağlığı gibi kritik öneme haiz hususlarda bu gecikmeler gerçek anlamda hayati

sonular doęurabilir. Bu nedenle gnmzde olduka yksek iřlem gcne sahip olan ve geliřmeye de devam eden bilgisayar teknolojilerinden yararlanma seeneęi tercih edilmektedir.

Yarı iletken teknolojisinin geliřmesiyle birlikte optimizasyon algoritmaları zerindeki ilgi ve alıřmalar da yoęunlařmıřtır. Bilgisayarların iřlemcileri geliřtike karmařık ve ok tekrarlı iřlemleri yapmak iin gerekli olan srenin kısılması, farklı operatrleri ierisinde barındıran birok yeni optimizasyon algoritmasının boy gstermesini saęlamıřtır. Ayrıca bilgisayarların geirdięi bu dnřmle birlikte algoritmaların daha yksek iterasyonda ve daha yksek parametrelili problemlerdeki performanslarını lmek de kolaylařmıřtır. Bu imkanlar dahilinde, mevcut problemler zerinde farklı optimizasyon problemleri adil kořullarda iřletilerek bir anlamda sonu gelmeyecek bir optimizasyon algoritmaları yarıřması ortaya ıktı diyebiliriz.

Genellikle yeni bir optimizasyon algoritmasının tanıtıldıęı yayınlar incelendięinde, literatre sunulan bu yeni algoritmanın dięer tm algoritmalarından daha iyi olduęu gibi bir izlenim edinilebilir. Arařtırmacılar ortaya koydukları algoritmaların yeteneklerini gsterebilmek amacıyla genellikle algoritmalarının yaygın ama problemleri zerinde elde ettięi sonulara da yer verirler. Bu ama problemleri algoritmanın yeteneklerini bir miktar gzlemlemeye imkan tanısa da aslında sadece bu sonulara bakarak bir algoritma iin bařarılı veya bařarısız demek uygun bir ynelim deęildir. Literatrde ve gerek hayatta karřılařılabilecek sonsuz problemler uzayının bir alanında alternatiflerine gre nispeten bařarılı olması o algoritma iin genel geer bir bařarılılık durumunu doęurmaz. Aynı olgu olumsuz durumlar iin de geerlidir. rneęin belirli bir alanda dięer optimizasyon algoritmalarından ok daha iyi sonu veren algoritmalar yaygın performans test fonksiyonlarında ok daha dřk bir performans ortaya koyabilir. Bu noktada bilinmesi gereken temel husus hem optimizasyon problemlerinin hem de optimizasyon algoritmalarının ok farklı karakteristiklere sahip olabileceęidir.

Optimizasyon algoritmalarını birok farklı zellikleri ynnden sınıflandırmaya tabi tutabiliriz. Ancak bu konuda literatrde en yaygın olarak kullanılan sınıflandırmalardan biri algoritmaları deterministik ve stokastik olarak ikiye ayırmaktır.

2.1.1. Deterministik Optimizasyon Algoritmaları

Deterministik optimizasyon, bir diğer adıyla matematiksel programlama optimizasyon algoritmalarının klasik dalıdır (Cavazzuti, 2013). Bu türdeki optimizasyon algoritmaları doğrusal cebir yöntemlerini kullanmaktadır. Deterministik yöntemler aynı koşullar için aynı sonuçları ürettiğinden dolayı bu türdeki optimizasyon algoritmaları, kaç defa çalıştırılırsa çalıştırılsın, aynı problemin aynı koşullardaki optimizasyonunun tüm adımlarında önceki koşullarla birebir aynı sonucu üretirler.

2.1.2. Stokastik Optimizasyon Algoritmaları

Stokastik optimizasyon algoritmaları, deterministik algoritmalarından farklı olarak içerisinde olasılıksal durumları da barındırır. Bilgisayarların deterministik makineler olduğu ve herhangi bir rastsallığa imkan vermediği için bilgisayar bilimi tarafından “sözde rastgele” kavramı kullanılmaktadır. “Sözde rastgele sayı üreticiler” olarak adlandırılan ve yazılımsal olarak rastgeleliği gerçek anlamı olmasa da sağlamayı amaçlayan fonksiyonlar geliştirilmiştir. Bu fonksiyonlar üzerinde koştukları cihazın sıklıkla değişen, tahmini kolaylıkla mümkün olmayan ve dış etkenler nedeniyle cihazın deterministik yapısının bir ürünü olmayan ortam sıcaklığı, imlecin ekrandaki konumu gibi bağımsız birçok parametreyi bir arada kullanarak “sözde rastgele” olarak nitelendirilen sayılar üretmektedir.

2.1.2.1. Sezgisel Algoritmalar

Sezgisellik terimi, problem çözümleri, öğrenme ve keşif süreçlerinde deneyime dayalı teknikleri ifade etmek için kullanılır (Kokash, 2005). Bilgisayar bilimleri, yapay zeka ve matematiksel optimizasyon konularında klasik optimizasyon yöntemlerinin çok yavaş kaldığı veya kesin sonucu bulamadığı durumlar için yaklaşık çözüm bulmayı hedefleyen, kesin çözüm bulma garantisi vermeyen yöntemler olarak açıklanabilir (Desale vd., 2015). Sezgisel yöntemler genellikle hızlı çalışmaları ve sonuca hızla yakınsamaları ile bilinirler.

2.1.2.2. Yarı-Sezgisel Algoritmalar

Bilgisayar bilimlerinde yarı-sezgisel, bir problemi iteratif olarak çözümler üreterek verilen kalite ölçütleri çerçevesinde optimize etmeyi amaçlayan tekniktir (Desale vd., 2015). Sezgisel algoritmalarda olduğu gibi kesin çözümü bulma garantisi vermemekle birlikte verilen problemler için çok geniş çözüm uzaylarında çözüm arayabilir. Doğrudan bir çözüm yöntemini içerisinde bulundurmeyen yarı-sezgisel algoritmalar, arama sürecinde yürütülecek olan stratejiyi belirlemektedir. Temel amaç çözüm uzayını keşfederek optimuma yakın çözümleri bulmaktır. Yarı-sezgisel algoritmalar çeşitlerine göre içerisinde basit yerel arama operatörlerini barındıracağı gibi bazı yarı-sezgiseller ise karmaşık öğrenme süreçlerinden de faydalanabilir. Yarı-sezgisellerin en önemli ve yaygın olarak kullanılmasını sağlayan özelliği ise herhangi bir probleme özel olmayışlarıdır. Bu nedenle bu türde algoritmalar için “kara kutu” tabiri de kullanılmaktadır. Benzetimli tavlama (Simulated Annealing, SA), Tabu arama (Tabu Search, TS), Yapay Arı Kolonisi (Artificial Bee Colony, ABC), Havai Fişek Algoritması (Fireworks Algorithm, FWA), Parçacık Sürü Optimizasyonu (Particle Swarm Optimization, PSO), İmmün Plazma Algoritması (Immune Plasma Algorithm, IPA), HS gibi yöntemler bilinen yarı sezgisel algoritmalarından bazılarıdır.

2.1.2.3. Üst-Sezgisel Algoritmalar

Üst-sezgisel algoritmalar, kısaca “sezgiselleri seçmek için kullanılan sezgisel yöntem” olarak tanımlanır (Burke vd., 2009) . Kendisine verilen sezgisel yöntemler havuzundan faydalanarak farklı problemleri çözmek için yöntem tasarlama sürecini otomatize şekilde yapabilen yöntemler üst-sezgisel olarak adlandırılır. “Üst-sezgisel” ifadesi 2000’li yıllardan itibaren kullanılan nispeten yeni bir terim olsa da esasında otomatik şekilde yöntem tasarlama fikri 1960’lı yıllara dayanmaktadır (Burke vd., 2013).

2.1.3. Optimizasyon Amaç Fonksiyonları

Belirli bir problemi optimize etmenin ilk aşaması problemi anlamak ve ardından formülize etmektir. Bir optimizasyon problemi doğru şekilde formülize edildikten sonra temel amaç doğru matematiksel teknikleri kullanarak üretilen uygun çözümler

arasından optimum olanları bulmaktır (Yang & He, 2016). Optimizasyon problemleri, optimizasyon algoritmasının üzerinde çalışabileceği ve ürettiği muhtemel çözümlerin değerlendirmesini yapabileceği yazılımsal fonksiyonlar haline getirildiğinde “amaç fonksiyonu” olarak isimlendirilir. Genellikle değerlendirme sonucu minimize veya maksimize edilmek istenilen amaç fonksiyonları farklı yapılarda olabilmektedir. Değişken sayılarına göre tek değişkenli ve çok değişkenli, sayısal değişkenlerin türüne göre ikili, tamsayı ve sürekli olarak sınıflandırılabilen amaç fonksiyonlarını yapılarına göre “Tek Amaçlı” ve “Çok Amaçlı” olmak üzere ikiye ayırabiliriz.

2.1.3.1. Tek Amaçlı Optimizasyon

Tek amaçlı optimizasyon fonksiyonları, nispeten formülizasyonu ve çözümü daha kolay olan optimizasyon problemleridir. Bu tip amaç fonksiyonları bir kara kutu olarak düşünülürse optimizasyon algoritmalarının tek yapması istenilen bu fonksiyonun ürettiği değeri en istenilen seviyeye getirmektir. Tek amaçlı optimizasyon problemlerinin çözülebilmesi için bilinmesi gerekenler girdi vektörünün boyutu, vektör elemanlarının alabileceği minimum ve maksimum değerler ve problemin minimize, maksimize veya ulaşması istenilen değerden hangisi olduğudur. Bu bilgiler girildikten sonra optimizasyon algoritmaları için bütün tek amaçlı fonksiyonlar üzerinde deneme yanılma yapılabilecek, verilen girdilere çıktı üreten bir kara kutudan ibarettir.

2.1.3.2. Çok Amaçlı Optimizasyon

Özellikle birçok gerçek hayat probleminde ulaşılmak istenilen amaç tek bir durumu minimize veya maksimize etmekten ziyade belirli kısıtlamaları ve birden fazla durumu göz önünde bulundurarak en uygun çözümü bulmaktır. Çok amaçlı optimizasyon yöntemlerinde amaçlar ve kısıtlar ayrı şekilde ele alınır. (Coello, 2000) Bu tür problemlere örnek olarak herhangi bir şekilde çok sınıfta aynı anda eğitim öğretim yapılan bir kurumda yer almış herkesi ilgilendiren ders çizelgeleme problemlerini gösterebiliriz (Altunay & Tamer, 2017). Ders çizelgeleme problemlerinde her bir sınıfın haftalık olarak alması gereken derslerin çeşidi ve haftalık ders saat sayıları, farklı branşlara sahip öğretmen veya akademisyenlerin hangi derslere girebileceği, peşpeşe gelmemesi gereken dersler, öğretmen veya

akademisyenlerin ders verebileceği gün sayısı gibi göz önünde bulundurulması gereken çok sayıda faktör bir araya gelmektedir. Bu tarz problemlerde en iyi çözümü üretmek bir kenara, tüm şartları ve kısıtlamaları sağlayan uygun çözümler üretmek bile oldukça zorlayıcı bir görevdir. Örnekteki gibi çok sayıda kısıtlaması olan amaçların çözümü için bu kabiliyetlere sahip olacak şekilde özelleştirilmiş “çok amaçlı optimizasyon algoritmaları” kullanılmaktadır.

2.1.4. EEG Sinyallerinin Gürültüden Temizlenmesi Problemi

Bu tezde EEG Sinyallerinin gürültüden temizlenmesi probleminin optimizasyonu üzerine çalışmalar yürütülmüştür. Büyük veri optimizasyon problemi olarak ele alınan problem, IEEE tarafından düzenlenen Evrimsel Hesaplama Kongresinin 2015 yılında gerçekleştirdiği Büyük Optimizasyon özel oturumu (*CEC BigOpt*, 2015) için gerçekleştirilen Büyük Veri Optimizasyon yarışma konusu olarak ortaya çıkmıştır (*BigOpt*, 2015).

Büyük veri denildiğinde genellikle insanların aklına veritabanlarında tutulan milyonlarca veya milyarlarca satır girdi, büyük depolama çözümlerinde saklanan çok fazla sayıda fotoğraf veya video dosyaları gelmektedir. Büyük veri, sadece mevcut verinin ne kadar büyük veya fazla sayıda olduğu değil; depolama, işleme ve yönetim açısından klasik yöntemlerle başa çıkılamayacak veri topluluklarını da ifade etmektedir. Yarışmanın internet sayfasında (*BigOpt*, 2015) da belirtilen örnekte olduğu gibi, eğer ki 12 MB boyutunda bir veriyi işlemekten bahsedecek olursak bu günümüz modern bilgisayar ve sunucularının işlem gücünü de göz önünde bulundurduğumuzda önemsenmeyecek sürelerde işlenecek bir problem olarak algılanmaktadır. Ancak bu işlemi her saniye yapmamız gerekirse, saniyede 12 MB işlememiz gereken veri, bir dakikalık bir periyotta 720 MB olarak karşımıza çıkmaktadır. Aynı veri akışının devam etmesi durumunda bir saatlik bir zaman aralığında işlememiz gereken verinin miktarı 42 GB olacaktır. Bu veri akışı bir gün boyunca devam ederse toplamda 1 TB boyutunda veriyi işlemiş olmak gerekecektir. Bu örnek aslında tek bir kesitine bakıldığında küçük olarak algılanan verinin süreklilik kazandığında bir büyük veri problemine dönüştüğünü görmemiz için güzel bir örnektir. Bu problemde de ikili formatta tutulduğunda saniyede 20 KB, düz metin formatında tutulduğunda saniyede 0.5 MB verinin işlenmesi amaçlanmaktadır.

Yarışma konusu olarak katılımcılara hem tek amaçlı hem de çok amaçlı optimizasyon problemleri sunulmuştur. Zaman serisi uzunluğunun 256 olarak belirlendiği problemde değişken sayısı zaman serisinin uzunluğu ile kanal sayısının çarpımı kadardır. 4, 12 ve 19 kanallı zaman serilerinin kullanıldığı problemlerde amaç fonksiyonunun parametre sayısı sırayla 1024, 3072 ve 4864 olarak hesaplanır.

Problemde N EEG sinyalinin kullanılan kanal sayısı ve M zaman serisinin uzunluğu olmak üzere EEG taramasından elde edilen işlenmemiş sinyal verisinin $N \times M$ boyutlu kaynak sinyal matrisi olan S matrisinde tutulduğu varsayılmaktadır. Amaç, S matrisinin içerdiği gürültü verilerini boyutu $N \times N$ olan A dönüşüm matrisi yardımıyla kaynak sinyalden temizleyerek daha kaliteli ve tıbbi olarak yorumlamaya daha elverişli sinyali içeren $N \times M$ boyutlu X matrisini oluşturmaktır. S , A ve X matrislerinin arasındaki ilişki Eşitlik 2.1 ile ifade edilmektedir.

$$X = A \times S \quad (2.1)$$

Problem S matrisini $S1$ ve $S2$ adında $N \times M$ boyutlu iki ayrı matrise ayrıştırmaktır. S , A , X ile $S1$ ve $S2$ matrisleri arasındaki ilişki Eşitlik 2.2 ve Eşitlik 2.3 ile ifade edilmektedir.

$$S = S1 + S2 \quad (2.2)$$

$$X = A \times S1 + A \times S2 \quad (2.3)$$

Ayrıştırma işlemi için X ile $S1$ matrisleri arasındaki Pearson korelasyon katsayısını hesaplamak için Eşitlik 2.4 kullanılır. Burada temel amaç C matrisinin köşegen elemanlarının değerlerini maksimize ederken diğer elemanları sıfıra yaklaştırmaktır.

$$C = \frac{\text{covar}(X, A \times S1)}{\text{var}(X) \times \text{var}(A \times S1)} \quad (2.4)$$

Pearson korelasyon katsayısı matrisini uygun değerler aralığına getirirken S ve

$S1$ matrislerinin arasındaki uzaklığın da mümkün olduğunca az olması sağlanmalıdır. Bir diğer ifadeyle S ve $S1$ matrisleri birbirine mümkün olduğunca yakın değerler içermeli, yani matrisler benzemelidir. Bu amaçlar göz önünde bulundurularak problemin çok amaçlı optimizasyon fonksiyonları Eşitlik 2.5 minimizasyonu ve Eşitlik 2.6 minimizasyonu olarak belirlenir.

$$f_1 = \frac{1}{N \times M} \sum_{ij} (S_{ij} - S1_{ij})^2 \quad (2.5)$$

$$f_2 = \frac{1}{N^2 - N} \sum_{i,j \neq i} (C_{ij})^2 + \frac{1}{N} \sum_i (1 - C_{ij})^2 \quad (2.6)$$

Verilen S , A ve X matrisleri için Eşitlik 2.7 amaç fonksiyonunun alabileceği minimum değeri bulmaya çalışarak optimize edecek $S1$ matrisinin hesaplanması da problemin tek amaçlı optimizasyon fonksiyonu olarak kullanılabilir.

$$f_1 + f_2 \quad (2.7)$$

Bu problemde, tüm parametreler için alt sınır -8.0 ve üst sınır 8.0 olarak belirlenmiştir.

3. MATERYALLER VE YÖNTEMLER

3.1. Harmoni Arama (HS) Algoritması

Harmoni arama algoritması (HSA), Simulation dergisinde 2001 yılında yayınlanan bir makale ile literatüre kazandırılmıştır (Geem vd., 2001). Doğadaki farklı frekanslara sahip ses dalgalarının uygun şekillerde bir araya gelerek oluşturduğu, insan kulağına hoş gelerek müziksel bir doyum sağlayan harmonilerin tarihsel süreçte çeşitli bilim insanı ve düşünürlerce belirli bir sistematığı yakalanmaya çalışılmıştır. Geem tarafından önerilen yöntemde de bir bestekarın yeni bir beste oluştururken içgüdüsel olarak gerçekleştirdiği kulağa en güzel gelen melodiyi oluşturma amacının optimizasyon problemlerindeki en iyiyi elde etme arayışıyla benzer süreçleri barındırdığı, bu doğal fenomenin taklit edilerek başarılı sonuçlar elde edilebileceği göz önünde bulundurulmuştur.

Geliştirilen yöntemde müzik performanslarındaki en iyi durum "fantastik harmoni" arayışı ile optimizasyon algoritmalarındaki en iyi durum arayışı olan minimum maliyet ya da maksimum fayda ve verim arayışına karşılık gelmektedir. Yeni bir müziğin bestelenmesi sırasında aşama aşama olarak deneme-yanılma yoluyla üretilen aday bir beste, HSA algoritmasında harmoni olarak isimlendirilmektedir. Geliştirilen bestelerin nota kağıtlarına yazılarak en iyilerinin saklanıldığı bir sistematik için de algoritma tarafından o zamana kadar üretilmiş olan en iyi harmonilerden belirli bir miktarını içeren bir harmoni hafızası (HM) tutulmaktadır. HM'nin tutulmasının amacı sadece en iyi harmonileri tutmaktan ziyade onlardan faydalanarak daha da iyi bir harmoni üretebilmektir. Bestelenen çeşitli müziklerin kulağa en güzel gelen kısımlarının uygun şekilde bir araya getirilerek veya benzetilerek daha da güzel bir müzik eserinin elde edilebilmesini sağlamak üzere algoritmada da HM'de tutulan harmonilerden faydalanılmaktadır. Mevcut üretilmiş harmonilerden faydalanarak yeni harmoniler üretmek daha iyi sonuçlar elde edebilmeyi sağlasa da sadece bu harmonilere bağlı kalmak belirli bir noktadan sonra hep benzer harmoniler üretilmesine neden olacak ve optimizasyon sürecinin yerel minimuma takılma ihtimalini bir hayli artıracaktır. Bu durumun önüne geçilerek

algoritmanın keşif (exploration) yapabilmesi için Harmoni hafızasından yararlanma oranı (Harmony memory consideration rate) (*HMCR*) adında bir parametre kullanılmaktadır. *HMCR*, yeni bir harmoni bestelenmesi sırasında mevcut HM'deki harmonilerden faydalanılıp faydalanılmayacağını belirlemek için kullanılır. Harmoni bestelemeye başlarken üretilen rastgele bir değer eğer ki *HMCR* parametresinden daha küçükse yeni bestelenecek harmoni HM'deki harmonilerden faydalanılarak oluşturulur, rastgele değer *HMCR*'dan büyük ise yeni harmoni sıfırdan rastgele değerler kullanılarak üretilir. Bestelenen yeni harmoni eğer HM'den faydalanılarak üretiliyorsa, müzik besteleme sürecindeki enstrümanların ince akort ayarı yapılmasına benzer şekilde yeni harmonide kullanılacak değer de belirli bir oranda komşu değerlerle değiştirilebilir. Bu duruma karar vermek için Ton Ayarlama Oranı (Pitch Adjusting Rate) (*PAR*) adı verilen bir parametre kullanılır. Üretilen rastgele bir değer *PAR*'dan küçükse kullanılacak değerde ton ayarlaması yapılır. Bu ayarlama sırasında %50 ihtimalle alt komşuluk %50 ihtimalle üst komşuluk kullanılır.

HS algoritmasının sözde kodu Algoritma 1 ile gösterilmiştir. Sözde kodun işleyişini açıklayacak olursak 4 ile 9 numaralı satırlar arasında algoritmanın başlangıç çözümleri olan ve harmoni hafızasını oluşturan *harmoni* adı verilen çözümler oluşturulur. Bu çözümler tamamen rastgele şekilde ve *HMS* parametresi ile belirlenen sayıda oluşturulurlar. *D* problemin parametre sayısını belirtir. $x_{j,L}$ ve $x_{j,U}$ ifadeleri sırasıyla *j* indisli parametrenin alabileceği en düşük ve en yüksek değerleri belirtir. $rand(0, 1)$ ifadesi ise 0 ile 1 arasında her seferinde rastgele olarak elde edilen değeri belirtir. İlksel çözümler oluşturulduktan sonra harmoni hafızası çözümlerin kalitesine göre sıralanır.

Algoritma 1'de iteratif hesaplama 14 - 35 satırları arasında yapılır. *k* değişkeni iterasyon sayacını belirtir ve her iterasyonun sonunda 34 numaralı satırda gösterildiği üzere bir artırılır. Her bir iterasyonda yeni bir harmoni oluşturulur. Yeni harmoninin *j* değişkeniyle takip edilen parametrelerinden her birini oluşturmak için ilk olarak hafızadan faydalanıp faydalanılmayacağına karar verilir. Rastgele olarak 0 ile 1 değer aralığında seçilen bir sayı eğer ki 16. satırda gösterildiği gibi *HMCR* değerinden küçükse mevcut parametre belirlenilirken harmoni hafızasından faydalanılır, küçükse 23. satırdaki gibi parametrenin alt ve üst sınırları arasında rastgele oluşturulur. Hafızadan faydalanılması durumunda harmoni hafızasındaki rastgele seçilen bir

Algoritma 1 HS Algoritmasının Sözde Kodu

```
1: Parametre sayısını  $D$ , maksimum değerlendirme sayısını  $K$  olarak belirle
2:  $HMS$ ,  $HMCR$ ,  $PAR$  ve  $bw$  değerlerini ata
3:
4: for  $i \leftarrow 1 \dots HMS$  do
5:   for  $j \leftarrow 1 \dots D$  do
6:      $x_{i,j} = x_{j,L} + rand(0, 1) * (x_{j,U} - x_{j,L})$ 
7:      $f(x_i)$  değerini hesapla
8:   end for
9: end for
10:
11: HM'yi  $f(x)$  değerlerine göre sırala
12:
13:  $k \leftarrow 1$ 
14: while  $k < K$  do
15:   for  $j \leftarrow 1 \dots D$  do
16:     if  $rand() < HMCR$  then
17:        $x_{new,j} = x_{rand,j}$ 
18:
19:       if  $rand() < PAR$  then
20:          $x_{new,j} = x_{new,j} + rand(-1.0, 1.0) * bw$ 
21:       end if
22:     else
23:        $x_{new,j} = x_{j,L} + rand(0, 1) * (x_{j,U} - x_{j,L})$ 
24:     end if
25:   end for
26:
27:  $f(x_{new})$  değerini hesapla
28:
29: if  $f(x_{new}) < f(x_{worst})$  then
30:    $HM_{worst} = x_{new}$ 
31:   HM'yi  $f(x)$  değerlerine göre sırala
32: end if
33:
34:    $k = k + 1$ 
35: end while
```

harmoninin aynı parametresinin değeri kullanılır.

Hafızadan yararlanılarak oluşturulan parametre için bir sonraki aşama ise ton ayarlama işlemi yapıp yapılmayacağına karar vermektir. Rastgele olarak 0 ile 1 arasında üretilen seçilen bir sayı eğer ki 19. satırda gösterildiği gibi PAR parametresinden küçükse ton ayarlama işlemi yapılır, değilse parametrenin mevcut değeri korunur. Ton ayarlama işlemi 20. satırda da görülebileceği üzere mevcut değerin rastgele olarak -1 ile 1 arasında üretilen bir sayı ile bw parametresinin çarpımı toplanılarak yapılmaktadır. Buradaki rastgele olarak üretilen değer sıfırdan daha küçük ise ton ayarlama işlemi bw miktarına varan alt komşuluk değerlerine, sıfırdan büyük ise yine bw miktarına varan üst komşuluk değerlerine güncellenir. -1 ile 1 arasında rastgele olarak üretilen bir sayının sıfırdan büyük ve küçük olma ihtimali teorik olarak aynı olduğu için sıfır gelme durumu göz ardı edilerek %50 ihtimalle alt

komşuluk değerlere, %50 ihtimalle ise üst komşuluk değerlere güncellenmiş olur.

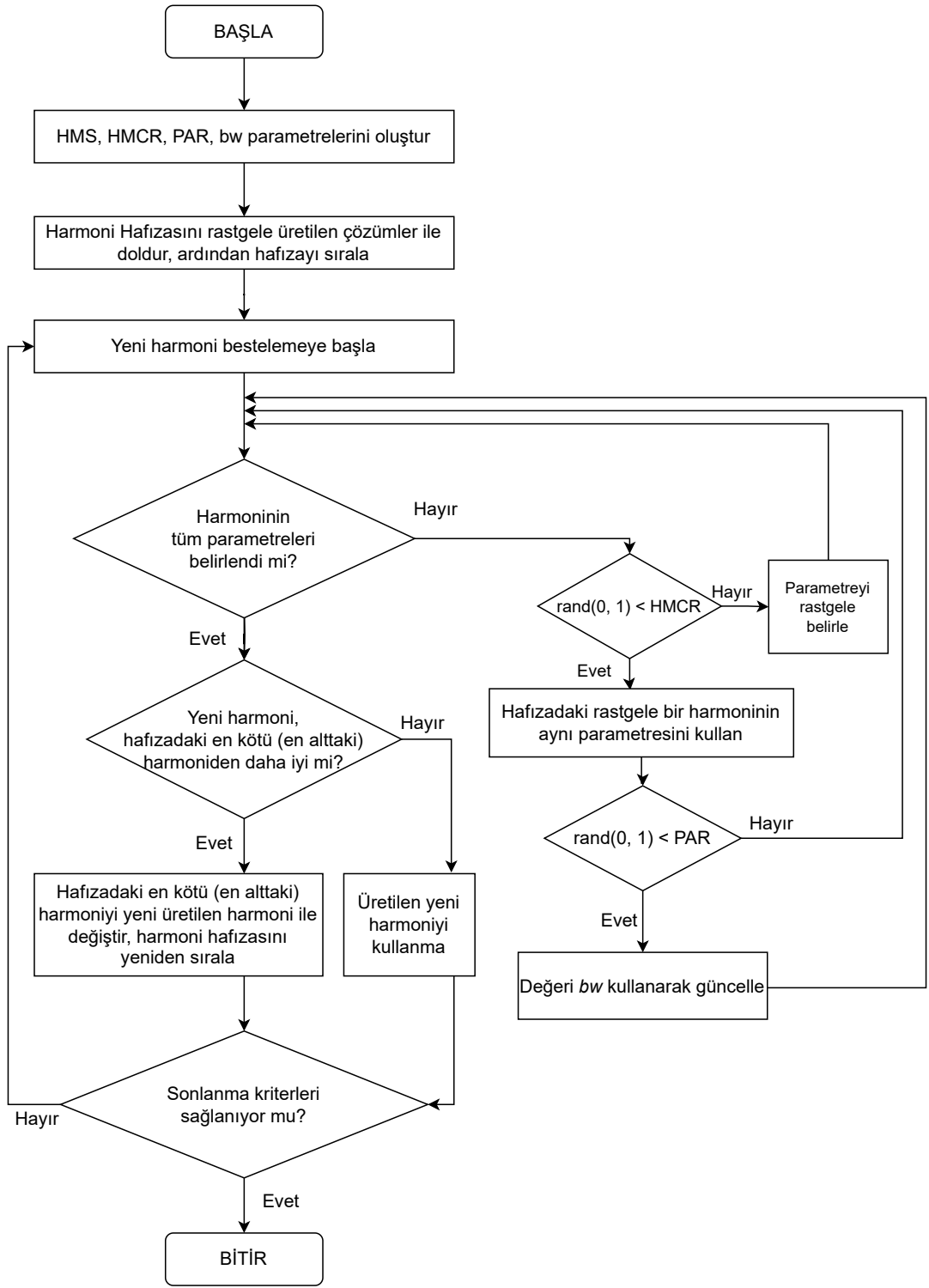
Algoritmanın iteratif sürecinin son aşaması ise harmoni hafızasının güncellenmesidir. Sözde kodda 29 - 32 satırları arasında gösterilen hafıza güncelleme aşamasında, eğer ki mevcut iterasyonda üretilen yeni çözümün amaç fonksiyon değeri harmoni hafızasındaki en başarısız çözümün amaç fonksiyon değerinden daha iyi ise hafızadaki en kötü harmoni çıkartılarak yeni üretilen harmoni hafızaya dahil edilir. Ardından hafızanın en iyiden en kötüye sıralandırılmış yapısını korumak için sıralama işlemi yapılır.

HS algoritmasının işleyişini görsel olarak ifade eden akış diyagramı Şekil 3.1 ile gösterilmiştir.

3.1.1. Harmoni Arama Algoritması Varyantları

HSA'nın çeşitli optimizasyon problemlerinde özellikle alternatif meta-sezgisel algoritmalarla benzer ve bazı problemlerde daha başarılı sonuçlar üretmesi nedeniyle algoritmaya olan ilgi ve uygulandığı problem sayısı da zamanla artmıştır. Algoritmanın problemin çözüm uzayı üzerindeki keşif (exploration) ve sömürü (exploitation) işlemlerinde kullandığı *HMCR*, *PAR* ve *bw* parametreleri sayesinde çözüme yaklaşma konusunda dengeli bir politika izlemesi araştırmacıların algoritmanın iç dinamiklerini merak etmesine ve buralarda yapılacak bir takım düzenleme ve ince ayarlarla daha iyi sonuçlar üretebilen bir hale getirme isteği uyandırmıştır. Zaman içerisinde birçok varyantı çıkan HS algoritmasında yapılan değişikliklere genel olarak bakıldığında; parametre seçimi ve atamasının dinamik hale getirilmesi yöntemi ve algoritmanın iç dinamiklerinin diğer meta-sezgisel algoritmalarda kullanılan yöntemlerle sentezlenerek hibrit algoritmalar geliştirildiğini söyleyebiliriz (Dubey vd., 2021; F. Qin vd., 2022).

Algoritmanın orijinal halinde araştırmacıların daha iyi sonuç elde edebilmek için yaptıkları ve algoritmanın performansına en çok etki eden faktör *HMCR*, *PAR* ve *bw* parametreleri için probleme uygun olacak şekilde doğru değerler seçilmesidir. Bu değerler sabit olduğu için algoritma çalıştırılmadan önce belirlenir ve tüm koşma süreci boyunca aynı değerler kullanılır. Parametre değerlerini seçmenin en yaygın yolu çeşitli değer kombinasyonlarının küçük veya az tekrarlı örnekler üzerinde



Şekil 3.1. HS algoritmasının akış diyagramı

koşturularak en iyi sonuç veren parametreleri kullanmaktır. Parametrelerin sabit olması özellikle değerlendirme sayısı ilerledikçe ve çözüm vektöründeki değerlerin daha küçük değişimler gerektirmeye başladığı aşamada bir dezavantaj olarak karşımıza çıkmaktadır.

Örneğin 0.75 olarak seçilecek olan *HMCR* değeri sayesinde üretilecek ortalama olarak 100 harmoniden 75 tanesi HM'den faydalanılarak önceki harmonilere çok benzer şekilde üretilirken 25 tane harmoni ise tamamen rastgele değerlerle üretilerek algoritmanın çözüm uzayında yeni alanlar keşfetmesini ve farklı parametrelerin yeni değerlerle kombinasyonunun çözüme etkisini gözlemlemeyi sağlar. Bu davranış algoritmanın ilk iterasyonlarında yerel ve küresel optimum değerlerin keşfini kolaylaştırır ve ilkleme sırasında tamamen rastgele seçilen değerlerle oluşturulan harmoni hafızasının boyutunun çok büyük olması gereğini ortadan kaldırır. Ancak belirli bir aşamadan sonra algoritma yerel ve küresel optimum bölgelerini büyük oranda keşfettikten sonra artık parametrelerin daha stabil olması ve çok küçük değişiklikler yapılması gerekirken, *HMCR* değerinin sabit olması nedeniyle 0.75 olarak belirlendiği varsayımından devam edersek ortalama olarak her 100 denemeden 25 denemede mevcut bulunmuş çözüm bölgelerinden çok daha uzak alanlarda rastgele harmoniler üreterek kaynakların ve zamanın boşa tüketilmesine neden olmaktadır. Benzer şekilde *PAR* değerinin sabit tutulması da sömürü sürecinin olumsuz etkilenmesine yol açmaktadır. Örneğin 0.2 olarak belirlenen *PAR* değeri, HM kullanılarak üretilen bir harmoninin içerisindeki parametrelerin %20 ihtimalle komşulardan faydalanılarak güncellenmesini, %80 ihtimalle ise alt ve üst sınır arasında rastgele bir değer ile deneme yapılmasını sağlar. Bu durum algoritmanın ilk iterasyonları sırasında çözüm vektöründeki diğer parametrelerin yeni değerlerle güzel sonuçlar üretebilme potansiyelinin ortaya çıkmasını sağlarken ilerleyen iterasyonlarda ise hali hazırda optimum değerine yaklaşmış olan değerlerin %80 ihtimalle tamamen rastgele değerler ile değiştirilmesine neden olarak büyük bir kaynak ve zaman kaybına neden olmaktadır. *HMCR* ve *PAR* göz önünde bulundurularak üretilen harmonilere son dokunuşu yapan *bw* parametresi de algoritmanın performansı üzerine önemli bir etkiye sahiptir. Özellikle parametrelerdeki çok küçük değişimlerin başarıma yüksek oranda etki ettiği problemlerde *bw*'nin yanlış seçimi adeta algoritmanın elini kolunu bağlayacaktır. *HMCR* ve *PAR* gibi *bw*'nin de dinamik olarak belirlenerek

araştırmanın ilerleyen süreçlerinde daha hassas değerler kullanılması yakınsama grafiklerinde logaritmik ölçek kullanılmasını gerektirecek hassasiyetin yakalanmasını sağlayabilmektedir.

3.1.1.1. Geliştirilmiş Harmoni Arama Algoritması (IHS)

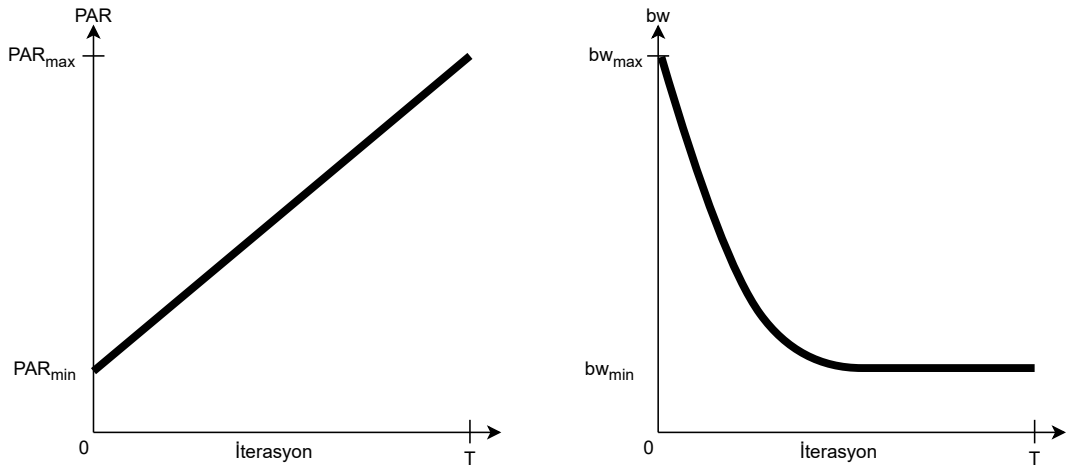
Parametrelerin HS algoritması üzerindeki etkisinden yola çıkarak parametreler ve parametrelerin kullanım stratejileri üzerinde değişiklik yapan HS varyantları ortaya çıkmıştır. En eski ve en çok bilinen HS varyantlarından biri olan, Mahdavi ve arkadaşları tarafından 2007 yılında literatüre sunulan Geliştirilmiş HS (Improved Harmoy Search, IHS) varyantında algoritmanın PAR ve bw parametrelerin dinamik hale getirilmiş, bu şekilde algoritmanın keşif ve sömürü yetenekleri daha dengeli bir hale gelmiştir (Mahdavi vd., 2007). Bu değişiklik sonucunda üretilen sonuçlarda da iyileşme elde edilmiştir. PAR ve bw değerlerinin iterasyon bazlı olarak güncellenmesi için Eşitlik 3.1 ve Eşitlik 3.2 kullanılmıştır.

$$PAR(t) = PAR_{min} + \frac{PAR_{max} - PAR_{min}}{T_{max}} \times t \quad (3.1)$$

$$bw(t) = bw_{max} \times \exp\left(\frac{\ln(bw_{min}/bw_{max})}{T_{max}} \times t\right) \quad (3.2)$$

t mevcut iterasyon sayısını ifade etmek üzere $PAR(t)$ mevcut iterasyonda kullanılmak üzere hesaplanan PAR değerini ifade eder. HS algoritmasında sabit olarak tanımlanan PAR parametresi yerine algoritma boyunca PAR 'ın alabileceği en yüksek değeri belirlemek için PAR_{max} ve en düşük değeri belirlemek için PAR_{min} değerlerinin tanımlanmasına ihtiyaç duyulmaktadır. Aynı durum bw parametresi için de geçerli olup bw_{max} ve bw_{min} parametreleri ile bw değerinin sırasıyla en büyük ve en küçük alabileceği değerler belirlenir. $bw(t)$ değeri de mevcut iterasyonda kullanılacak bw değerini verir. Bu düzenleme sonucunda PAR değeri iterasyon bazlı olarak doğrusal bir şekilde PAR_{min} değerinden PAR_{max} değerine kadar yükselir. Bu sayede başta komşu değerlerden yararlanma oranı daha az olup yeni ve daha uygun değerlerin keşfedilmesini kolaylaştırırken iterasyonlar ilerledikçe rastgeleliğin yerini komşulardan yararlanma alacağı için harmoni hafızasında koşmalar süresince elde

edilen deneyimden daha fazla yararlanılır. bw değeri ise başlangıçta bw_{max} seviyesinde olup üstel olarak bw_{min} seviyesine yaklaşır. Bu iki dinamik parametre sayesinde özellikle son iterasyonlara yaklaşıldığında PAR değeri büyümüş ve bw değeri küçülmüş olduğundan mevcut en iyi harmonilerde küçük dokunuşlar ile daha da iyi harmoniler üretilerek sürecin tamamlanmasına yardımcı olurlar. IHS algoritmasında önerilen hesaplamalar ile PAR ve bw parametrelerinin iterasyonlar boyunca değişimi Şekil 3.2'deki gibi olmaktadır.



Şekil 3.2. IHS algoritmasında PAR ve bw değerlerinin değişimi

3.1.1.2. Küresel En İyi Harmoni Arama Algoritması (GHS)

Omran ve Mahdavi tarafından 2008 yılında yayınlanan Küresel-En İyi HS (Global-Best Harmony Search, GHS) algoritmasında ise Parçacık Sürü Optimizasyonu (Particle Swarm Optimization, PSO) yöntemindeki sürü zekası konseptinden esinlenerek IHS varyantında değişiklikler yapılmıştır (Omran & Mahdavi, 2008). Birçok adımı IHS algoritması ile aynı olan GHS'de PAR ile yapılan ton ayarlama işleminde bw 'ye başvurmak yerine HM'deki mevcut en iyi harmoninin ilgili parametresi kullanılmıştır. Yeni parametre belirlenmesindeki en iyi harmoniden yararlanma davranışı nedeniyle algoritma "Küresel-En İyi" olarak nitelendirilmektedir. Ton ayarlama işleminde yapılan bu değişiklik nedeniyle HS'deki bw ve IHS'deki bw_{max} ve bw_{min} parametrelerine olan ihtiyaç ortadan kalkmıştır. Bu yönüyle algoritma IHS'deki bw_{min} ve bw_{max} değerlerinin seçiminde yaşanan zorluğu ortadan kaldırmayı da hedeflemiştir. GHS algoritması bu haliyle birçok kıyaslama

probleminde HS ve IHS den daha başarılı sonuçlar üretebilmiştir.

3.1.1.3. Değiştirilmiş Harmoni Arama Algoritması (MHS)

Cheng ve arkadaşları tarafından 2008 yılında geliştirilen Değiştirilmiş HS (Modified HS, MHS) algoritmasının, orijinal HS algoritmasının 25 parametreye kadar olan problemlerde güzel sonuçlar elde etse de daha fazla parametrelili problemler için verimli bir yöntem sunmadığı için başarısız olduğu yorumu yapılarak daha fazla kontrol parametresine sahip karmaşık problemler için geliştirildiği belirtilmiştir (Cheng vd., 2008). 0 ile 1 arasında değerler alabilen δ parametresinin eklendiği yeni yöntemde, harmoni hafızasındaki her bir çözüm için Eşitlik 3.3 kullanılarak bir seçilme olasılığı hesaplanmaktadır.

$$ps(t) = \delta \times 1 - \delta^{t-1} \quad (3.3)$$

Hesaplanan seçilme olasılıkları ST adındaki bir dizide kendisinden daha iyi harmonilerin seçilme olasılıklarının kümülatif toplamına eklenerek tutulur. ST dizisinin ilk elemanına 0.0 değeri atanır. Hafızadan yararlanılırken üretilen 0 ile 1 arasındaki rastgele değer, seçilme olasılıkları arasında olduğu iki harmoniden seçilme olasılığı büyük olandan faydalanılarak yeni harmoni üretilir. Algoritmadaki bir diğer değişiklik ise HM'deki her iterasyonda tek aday harmoni üretiminin yerine Nhm adında bir parametre ile ifade edilen, belirlenen sayıda yeni harmoninin üretilmesidir. Üretilen yeni harmonilerin, hafızadaki herhangi bir harmoniden daha iyi olanları hafızaya dahil edilerek daha kötü olanlar hafızadan çıkartılarak hafızanın boyutunun sabit kalması sağlanmaktadır.

3.1.1.4. Kendi-Uyarlamalı Küresel En İyi Harmoni Arama Algoritması (SGHS)

Pan ve arkadaşları tarafından 2010 yılında yayınlanan Kendi-Uyarlamalı Küresel-En İyi HS (Self-Adaptive Global-Best HS, SGHS) algoritması GHS'nin geliştirilmesiyle elde edilen bir yöntemdir (Pan vd., 2010). Önceki HS varyantlarından farklı olarak yeni bir harmoni besteleme yapısı ve uyarlamalı bir parametre ayarlama metodu kullanmaktadır. HS algoritmasında sabit olarak verilen $HMCR$ ve PAR

parametreleri kendiliğinden uyarlanabilir hale getirilmiş, probleme uygun $HMCR$ ve PAR seçim gereğini ortadan kaldırmayı hedeflemişlerdir. Değer aralığı $HMCR$ için 0.9 ile 1.0 arasında, PAR için 0.0 ile 1.0 arasında belirlenmiş olup standart sapmaları da sırasıyla 0.01 ve 0.05 olarak belirlenmiştir. $HMCR$ ortalamasını ve PAR ortalamasını tutan $HMCR_m$ ve PAR_m değerleri sırasıyla 0.98 ve 0.9 olarak ilklendirilmiştir. Yeni bir parametre olarak ise öğrenme periyodu (learning period, LP) eklenmiştir. İterasyon sayısı öğrenme periyoduna ulaştığında yeni $HMCR$ ve PAR değerleri hesaplanmakta ve $HMCR_m$ ve PAR_m değerleri de güncellenmektedir. Bu şekilde uygun $HMCR$ ve PAR değerlerinin öğrenme yoluyla kendiliğinden uyarlamalı olarak probleme göre uygun değerleri alması amaçlanmıştır. bw değerinin belirlenmesi için ise IHS'de olduğu gibi bw_{max} ve bw_{min} parametrelerinden faydalanılarak iterasyon sayısına göre uygun değer hesaplanmıştır. bw değeri, toplam iterasyon sayısının yarısına kadar doğrusal bir şekilde azalmakta, kalan iterasyonlarda ise bw_{min} değeri kullanılmaktadır. bw değerinin hesaplanması için Eşitlik 3.4 kullanılmaktadır.

$$bw(t) = \begin{cases} bw_{max} - \frac{bw_{max}-bw_{min}}{NI} \times 2t & \text{eğer } t < NI/2 \\ bw_{min} & \text{eğer } t \geq NI/2 \end{cases} \quad (3.4)$$

3.1.1.5. Kaşif Harmoni Arama Algoritması (EHS)

Das ve arkadaşları tarafından 2010 yılında literatüre sunulan Kaşif HS (Explorative HS, EHS) ise bw değerini dinamik olarak ayarlamaktadır. bw değeri mevcut iterasyon için yeni üretilcek çözüm vektör elemanının harmoni hafızasındaki standart sapması kullanılarak Eşitlik 3.5 ile hesaplanır (Das vd., 2010). Eşitlikteki k sabit bir katsayı olup çok büyük veya çok küçük seçilmesinin algoritmanın başarımlı performansını olumsuz etkilediği, yapılan ölçümlerde 1.2 civarındaki değerlerin kıyaslama fonksiyonlarında yüksek başarımlı sağladığı aktarılmıştır. bw değeri harmoni hafızasının standart sapması kullanılarak hesaplandığı için hafıza rastgele oluşturulmuş değerleri içeriyorsa bw de rastgeleliğin etkisini aktaracaktır.

$$bw = k \times \sqrt{Var(x)} \quad (3.5)$$

3.1.1.6. Yeni Bir Küresel Harmoni Arama Algoritması (NGHS)

Zou ve arkadaşları tarafından 2010 yılında literatüre kazandırılan Yeni Bir Küresel HS (A Novel Global HS, NGHS) algoritmasında PSO algoritmasındaki parçacık sürüsü sürü zekasından yola çıkılarak HS algoritmasının işleyişinde birtakım değişiklikler yapılmıştır (Zou vd., 2010). *HMCR* ve *PAR* parametrelerinin devre dışı kaldığı algoritmaya genetik mutasyon olasılığı (genetic mutation probability, p_m) parametresi eklenmiştir. Ayrıca yeni harmoni besteleme süreci de değiştirilerek pozisyon güncelleme ve genetik mutasyon operasyonları eklenmiştir. Ayrıca HS algoritmasındaki HM'nin güncellenme stratejisi de değiştirilerek yeni üretilen harmoni, hafızadaki en kötü harmoniden daha iyi olup olmadığına bakılmaksızın hafızadaki en kötü harmoniyle değiştirilmektedir.

3.1.1.7. Önerilen HS (PHS) ve İyileştirilen Önerilen HS Algoritması (IPHS)

Jaberipour ve Khorram tarafından 2010 yılında iki yeni yöntem birlikte önerilmiştir (Jaberipour & Khorram, 2010). Önerilen HS (Proposed HS, PHS) ve İyileştirilen Önerilen HS (Improving Proposed HS, IPHS) algoritmalarında *bw* değeri dinamik olarak türev kullanılarak hesaplanmaktadır. Türev hesaplama işlemi karmaşık bir hale gelebileceğinden ötürü Taylor teoremi kullanılarak yaklaşık türev değerlerinin hesaplanması önerilmiştir. IPHS algoritmasında PHS algoritmasına ek olarak *PAR* parametresinin iki farklı şekilde hesaplanmasını sağlayacak şekilde bir *A* parametresi eklenmiştir.

3.1.1.8. Zeki Ayarlı Harmoni Arama Algoritması (ITHS)

Yadav ve arkadaşları tarafından 2012 yılında yayınlanan Zeki Ayarlı HS (Intelligent Tuned HS, ITHS) algoritmasında, harmoni hafızasındaki mevcut harmoniler, hafızanın ortalamasından daha iyi olanlar A grubu, daha kötü olanlar B grubu şeklinde iki gruba ayrılır (Yadav vd., 2012). A grubu hem keşif hem de sömürü işleminden sorumlu iken B grubu sadece keşif işleminden sorumludur. Eşitlik 3.6 ile dinamik olarak hesaplanan *PAR* değeri, yeni bestelenen harmoninin elemanı A grubundan faydalanılarak oluşturuluyorsa olasılıksal olarak üç farklı yöntemden bir tanesi uygulanarak ton ayarlamasında kullanılır. Eğer ki bestelenen yeni harmoninin

elemanı B grubundan faydalanılarak oluşturuluyorsa hafızadaki en iyi harmoniden faydalanılır.

$$PAR_t = PAR_{max} - (PAR_{max} - PAR_{min}) \times \frac{t}{T_{max}} \quad (3.6)$$

3.1.1.9. Dinamik Kontrol Parametrelili Harmoni Arama Algoritması (NDHS)

Chen ve arkadaşları tarafından 2012 yılında yayınlanan Dinamik Kontrol Parametrelili HS (HS Algorithm with Dynamic Parameters, NDHS) algoritmasında yeni bir hafızadan yararlanma kuralı ile PAR ve bw parametrelerini dinamik olarak güncelleyebilen yeni bir dinamik parametre ayarlama şeması sunulmuştur (Chen vd., 2012). Geliştirilen hafızadan yararlanma kuralında turnuva seçim yöntemi kullanılmıştır. Belirli bir sayıda seçilen harmoniler arasındaki en iyi harmoniden yararlanılarak yeni harmoni oluşturulur. Ton ayarlama işleminde ise PAR parametresi Eşitlik 3.7 kullanılarak dinamik olarak hesaplanır.

$$PAR_t = PAR_{min} + (PAR_{max} - PAR_{min}) \times \exp(-kt/T_{max}) \quad (3.7)$$

k değerinin toplam iterasyon sayısının 10.000'de 1'i olarak kullanılması önerilmiştir. Ton ayarlama işlemi koşmanın ilk yarısındaki iterasyonlarda farklı bw hesaplama yöntemleri ile yapılmaktadır. Eşitlik 3.8'e göre hesaplanan bw değeri için fonksiyon parametresi ilk yarıdaki koşmalarda doğrudan üretilen değer olarak kullanılırken son yarıdaki koşmalarda yeni üretilen değerden hafızadaki rastgele seçilen başka bir harmoninin aynı sütunundaki değer çıkarılması ile elde edilen değer kullanılır.

$$bw(x) = \begin{cases} (UB_j - LB_j)/1000.0 & \text{eğer } x = 0 \\ x & \text{eğer } 0 < x \leq 1 \\ \max((UB_j - LB_j)/50, 1) & \text{eğer } x > 1 \end{cases} \quad (3.8)$$

3.1.1.10. Kendinden-Uyarlamalı Harmoni Arama Algoritması (NSHS)

Luo tarafından 2013 yılında yayınlanan Yeni Kendinden-Uyarlamalı HS (Novel Self-Adaptive HS, NSHS) algoritmasında $HMCR$ ve bw parametreleri dinamik hale getirilmiş, PAR parametresi ise ihtiyaç kalmadığı için kaldırılmıştır (Luo, 2013). $HMCR$ değeri 0.5 ile 1.0 arasında değişecek şekilde Eşitlik 3.9 kullanılarak hesaplanmaktadır. n değeri problemin parametre sayısını belirtmektedir. bw değeri de Eşitlik 3.10 kullanılarak hesaplanmakta, harmoni hafızasındaki amaç fonksiyon değerlerinin standart sapması 0.0001 seviyesi ve altındaysa bw değeri de 0.0001 olarak kullanılmaktadır. Eşitlikteki T_{max} toplam iterasyon sayısını, t değeri mevcut iterasyon sayısını ifade eder.

$$HMCR = 1 - \frac{1}{n + 1} \quad (3.9)$$

$$bw_i(t) = \begin{cases} \frac{UB_i - LB_i}{100} \times \left(1 - \frac{t}{T_{max}}\right) & \text{eğer } f_{std} > 0.0001 \\ 0.0001 & \text{aksi takdirde} \end{cases} \quad (3.10)$$

3.1.1.11. Dinamik Kendinden-Uyarlamalı Harmoni Arama Algoritması (DSHS)

Kattan ve Abdullah tarafından 2013 yılında önerilen Dinamik Kendinden-Uyarlamalı HS (Dynamic Self-Adaptive HS, DSHS) yönteminde PAR değerini hesaplamak için yapay sinir ağlarında da kullanılan En İyinin En Kötüye (best-to-worst, BtW) oranından faydalanılmıştır (Kattan & Abdullah, 2013). Harmoni hafızasındaki en kaliteli çözümün amaç fonksiyon değerinin hafızadaki en başarısız çözümün amaç fonksiyon değerine oranını belirten bu parametre ne kadar büyükse hafızada bulunan harmonilerin o kadar birbirine yakın kalitede çözümler olduğunu belirtir. BtW oranı Eşitlik 3.11 ile hesaplanmaktadır. $f(x_{best})$ en iyi harmoninin değerini, $f(x_{worst})$ ise en kötü harmoninin değerini ifade eder.

$$BtW = \frac{f(x_{best})}{f(x_{worst})} \quad (3.11)$$

PAR değerinin hesaplanması için Eşitlik 3.12 kullanılır.

$$PAR = (PAR_{min} - PAR_{max}) \times BtW + PAR_{max} \quad (3.12)$$

bw değerinin hesaplanması için ise aktif bant genişliği (bw_A) ve bu değerden faydalanılarak rastsal şekilde üretilen dinamik bant genişliği (Dynamic Bandwidth Value, bw_D) değeri kullanılır. bw_A 'nın hesaplanması için Eşitlik 3.13 kullanılır.

$$bw_A(i) = C \times \text{std}(x_i) \quad (3.13)$$

Burada bw değeri uygulanacak mevcut parametre için harmoni hafızasındaki değerler arasındaki standart sapması, yani her bir harmoni bir satır olarak düşünülürse, harmoni hafızasındaki tüm satırların sadece işlem yapılacak olan sütunundaki değerlerinin standart sapması ve C değeri kullanılır. C değeri EHS'deki k değerine benzer şekilde bir sabit olarak düşünülebilir ancak 1.0'dan büyük seçilmesi ve çok büyük değerleri de almaması gereken C değerinin seçimi için Eşitlik 3.14 önerilmiştir.

$$C = \begin{cases} 2 & \text{eğer } AccRate \geq AC_{th} \\ 2 \times \frac{AccRate}{AC_{th}} & \text{eğer } AC_{th} > AccRate > 1 \\ 0.1 & \text{eğer } AccRate \leq 1 \end{cases} \quad (3.14)$$

Buradaki $AccRate$ değeri yeni üretilen harmoniler için hesaplanan kabul edilme oranı olup son 100 iterasyonda harmoni hafızasına dahil edilen, yani hafızadaki en kötü harmoniden daha iyi sonuç üreten harmoni sayısının oranını ifade eder. AC_{th} ise kabul oranı için belirlenmiş olan eşik değeri olup %20 oranının çok iyi performans sağladığı çalışmada belirtilmiştir.

bw_D 'nin hesaplanması için ise Eşitlik 3.15 kullanılır.

$$bw_D(i) = \text{rand}(-bw_A(i), bw_A(i)) \quad (3.15)$$

3.1.1.12. Parametre Uyarlamalı Harmoni Arama Algoritması (PAHS)

Kumar ve arkadaşları tarafından 2014 yılında önerilen Parametre Uyarlamalı HS (Parameter Adaptive HS, PAHS) algoritmasında ise $HMCR$ ve PAR parametrelerinin her birisi için bir adet doğrusal ve bir adet üstel hesaplama modeli önerilmiştir (Kumar vd., 2014). $HMCR$ ve PAR için doğrusal hesaplama sırasıyla Eşitlik 3.16 ve Eşitlik 3.17 ile, üstel hesaplama ise sırasıyla Eşitlik 3.18 ve Eşitlik 3.19 ile yapılmaktadır.

$$HMCR(t) = HMCR_{min} + \frac{HMCR_{max} - HMCR_{min}}{T_{max}} \times t \quad (3.16)$$

$$PAR(t) = PAR_{min} + \frac{PAR_{max} - PAR_{min}}{T_{max}} \times (T_{max} - t) \quad (3.17)$$

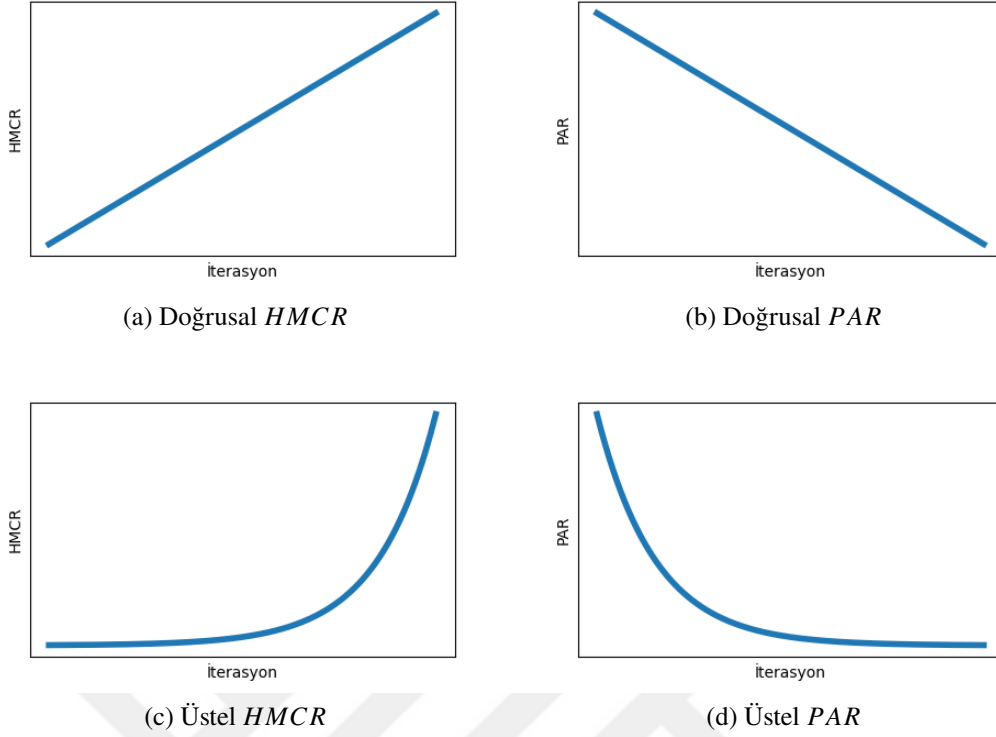
$$HMCR(t) = HMCR_{min} * \exp\left(\frac{\ln(HMCR_{min}/HMCR_{max})}{T_{max}} \times t\right) \quad (3.18)$$

$$PAR(t) = PAR_{max} * \exp\left(\frac{\ln(PAR_{min}/PAR_{max})}{T_{max}} \times t\right) \quad (3.19)$$

Bu iki parametre için önerilen doğrusal ve üstel hesaplama yöntemleri istenilen kombinasyonlarda kullanılabilir. Çalışmada bu kombinasyonlardan (Doğrusal $HMCR$, Doğrusal PAR) için PAHS-1, (Üstel $HMCR$, Doğrusal PAR) için PAHS-2, (Doğrusal $HMCR$, Üstel PAR) için PAHS-3, (Üstel $HMCR$, Üstel PAR) için PAHS-4 ifadesi kullanılmıştır. $HMCR$ ve PAR değerlerinin grafiksel değişimi Şekil 3.3 ile gösterilmiştir.

3.1.1.13. Küresel Dinamik Harmoni Arama Algoritması (GDHS)

Khalili ve arkadaşları tarafından geliştirilen Küresel Dinamik HS (Global Dynamic HS, GDHS) algoritmasında HS algoritmasında kullanılan $HMCR$, PAR ve bw parametrelerinin tamamı dinamik hale getirilmiştir (Khalili vd., 2014). Algoritmanın koşturulması için sadece harmoni hafıza boyutunun verilmesi yeterli olmaktadır. Önceki birçok yöntemde $HMCR$ ve PAR parametreleri dinamik hale



Şekil 3.3. PAHS algoritması $HMCR$ ve PAR değerleri

getirilerek doğrusal veya üstel şekilde devamlı artış veya devamlı azalış seyri gösterirken, önerilen bu yöntemde $HMCR$ değeri 0.9 ile 1.0 değer aralığında başlangıçtan toplam iterasyon sayısının yarısına kadar parabolik bir artış gösterirken bu noktadan sonra ise yine 0.9 değerine doğru parabolik şekilde azalmaktadır. PAR parametresi de benzer şekilde 0.85 ile 1.0 arasında toplam iterasyon seviyesinin yarısına kadar artarken bu noktadan sonra azalmaktadır. Uygulanan bu yöntemle başlangıçta $HMCR$ değerinin bir miktar düşük olması uygulamanın keşif özelliğine katkıda bulunurken koşmanın ortalarına doğru 1.0 seviyesine yaklaşıldığı için artık çözüm uzayında yeni alanların keşfinden ziyade harmoni hafızasından büyük oranda faydalanılmakta, bu sayede o zamana kadar elde edilen birikimden daha yüksek oranda faydalanma imkanı olmaktadır. Sonrasında da tekrardan 0.9 seviyesine gerilemesi sayesinde de olası yerel optimum çözümlere takılmış olma durumu mevcut ise bu yerel optimum noktasından kurtulabilmek için yeniden algoritmaya fırsat sunmaktadır. bw değeri ise problemin üst ve alt limitleri kullanılarak hesaplanan bw_{max} ve bw_{min} sabit değerleri yardımıyla hesaplanır. PAR değerinde de aynı durum sömürü işlemi için benzer etkiler oluşturmaktadır. GDHS algoritmasında $HMCR$

değeri Eşitlik 3.20 ile, PAR değeri ise Eşitlik 3.21 ile hesaplanmaktadır.

$$HMCR(t) = 0.9 + 0.2 \times \sqrt{\frac{t-1}{T_{max}-1} \times \left(1 - \frac{t-1}{T_{max}-1}\right)} \quad (3.20)$$

$$PAR(t) = 0.85 + 0.3 \times \sqrt{\frac{t-1}{T_{max}-1} \times \left(1 - \frac{t-1}{T_{max}-1}\right)} \quad (3.21)$$

3.1.1.14. Kendinden Uyarlamalı Geliştirilmiş HS Algoritması (SIHS)

Kalivarapu ve arkadaşları tarafından 2016 yılında yayınlanan Kendinden Uyarlamalı Geliştirilmiş HS (Self-Adaptive Improved HS, SIHS) algoritmasında ise bw değeri dinamik hale getirilmiştir (Kalivarapu vd., 2016). Eşitlik 3.22 kullanılarak hesaplanan k değerindeki a sabiti için 50, b sabiti için 100 değerinin kullanılması önerilmektedir. bw değeri ise Eşitlik 3.23 kullanılarak hesaplanmaktadır. Buradaki c ve m sabit değerler olup 1'den daha büyük değerler olması gerekmektedir.

$$k = a \times \ln \left[\frac{bw_{max}}{b \times bw_{min}} \right] \quad (3.22)$$

$$bw(t) = \frac{c}{1 + k(t/T_{max})} \quad (3.23)$$

3.1.1.15. Bulanık Harmoni Arama Algoritması (FHS)

Peraza ve arkadaşları tarafından 2016 yılında literatüre sunulan Bulanık HS (Fuzzy HS, FHS) algoritmasında bulanık mantıktan faydalanılmıştır. HS algoritmasında sabit değerler alan $HMCR$ ve PAR , bulanık parametrelere dönüştürülmüştür. $HMCR$ ve PAR değerlerini dinamik olarak ayarlayacak olan bulanık sistemi geliştirmek için iterasyon tamamlanma yüzdesi girdi olarak kullanılmıştır. Bu türdeki bulanık kontrollerde ve durulaştırma işlemlerinde ağırlıklı olarak yaygın olduğu için Mamdani yöntemi kullanılmıştır.

3.1.1.16. Yerel Karşıt Tabanlı Kendinden Uyarlamalı Küresel HS (LHS)

Ouyang ve arkadaşları tarafından 2017 yılında önerilen Yerel Karşıt Tabanlı Kendinden-Uyarlamalı Küresel HS (Local Opposition-Based Self-Learning Global HS, LHS) algoritmasında ise HS algoritmasının önceki çalışmalarda daha kısıtlı şekilde karşıt-taban metodu uygulanarak elde edilen başarılar göz önünde bulundurularak yeni bir yöntem sunulmuştur (Ouyang vd., 2017). Yeni harmoni bestelenme işlemi sırasında üretilen bir harmoninin karşıtının da değerlendirmeye tabi tutarak çözüm uzayının keşfedilmesini kolaylaştırmıştır. Ton ayarlama işleminde de harmoni hafızasındaki en iyi ve en kötü çözümlerden faydalanarak kendinden-uyarlamalı bir yöntem kullanılmıştır. Hafızanın güncellenme stratejisi de değiştirilerek üretilen harmoni ve karşıt harmoniden hangisi daha iyi ise hafızadaki en kötü harmoniden daha iyi olup olmadığına bakılmaksızın hafızadaki en kötü harmoni çıkartılarak yerine üretilen iki harmoniden biri eklenmektedir. Orijinal HS ile karşılaştırıldığında LHS varyantı üç temel işleyiş değişikliğine sahiptir.

Karşıt tabanlı öğrenme konsepti bir makine zekası yöntemi olup, bir tahmin geliştirilirken onun karşıtı bir tahmin de aynı zamanda üretilerek mevcut aday çözüm için daha iyi bir yakınsama imkanı sunulmaya çalışılır (Tizhoosh, 2005). Rahnamayan ve arkadaşları tarafından geliştirilen Karşıt Tabanlı Diferansiyel Gelişim Algoritmasında (OBE) da bu yöntem uygulanmıştır (Rahnamayan vd., 2008). Konsept PSO, Karınca Koloni Optimizasyonu (ACO) ve Grup Arama Optimizasyonu (GSO) gibi algoritmaların başlangıç popülasyonu oluşturma mekanizmalarını iyileştirmek için de kullanılmıştır (H. Wang vd., 2007; Malisia & Tizhoosh, 2007; S.-W. Wang vd., 2012).

HS algoritmasını iyileştirmek için de karşıt tabanlı öğrenme konsepti kullanılmıştır. Hibrit bir algoritma olarak geliştirilen HS-OBL algoritmasında HS algoritmasının mutasyon operatörü karşıt tabanlı öğrenme ile iyileştirilmiştir (Gao vd., 2010). HS algoritması üzerinde karşıt tabanlı konsepti uygulayan çeşitli algoritmalarda HM oluşturulurken bu yöntemden faydalanılmıştır (A. K. Qin & Forbes, 2011; Singh vd., 2013). Ancak sadece ilk çözümlerin üretim süreci olan HM'nin ilklenmesi sürecinde karşıt tabanlı yaklaşımın kullanılması aslında HS algoritmasının bu yöntemden yeterince faydalanmasını sağlayamıyordu. LHS

algoritmasında ise yerel karşıt tabanlı yaklaşım aynı zamanda HS algoritmasının harmoni hafızasından faydalanma aşamasına dahil edilerek bu yöntemden sadece algoritmanın başında değil, optimizasyon sürecinin tamamında faydalanılmasını sağlamıştır.

Karşıt tabanlı öğrenme konseptini anlamak için $x \in [a, b]$ olmak üzere x 'in karşıt değeri Eşitlik 3.24 olarak hesaplanır.

$$\tilde{x} = a + b - x \quad (3.24)$$

Buradan yola çıkarak, çok boyutlu bir sayı dizisinin karşıt değeri de Eşitlik 3.25 olarak hesaplanır.

$$\tilde{x}_i = a_i + b_i - x_i \quad (3.25)$$

Bu şekilde üretilen x ve $\tilde{x}$ çözüm vektörleri birbirinin karşıtı çözümler olarak adlandırılır. Yeni bir çözüm üretilirken x ile birlikte $\tilde{x}$ çözümü de üretilir ve her bir iterasyonda bu iki çözüm de amaç fonksiyonunda değerlendirmeye tabi tutularak $f(x)$ ve $f(\tilde{x})$ hesaplanır. Bu değerlerden $f(x)$ daha kaliteli bir değer sunuyor ise x çözümü, $f(\tilde{x})$ daha kaliteli bir değer sunuyor ise $\tilde{x}$ çözümü kullanılır. Yani bir minimizasyon problemi için Eşitlik 3.26 kullanılarak aday çözüm belirlenir.

$$x_{yeni} = \begin{cases} x & \text{eğer } f(x) \leq f(\tilde{x}) \\ \tilde{x} & \text{aksi takdirde} \end{cases} \quad (3.26)$$

HS algoritmasında harmoni hafızasından yararlanma aşamasında *HMCR* değeri o iterasyonda üretilen rastgele bir değerle karşılaştırılarak hafızadaki rastgele seçilen bir çözümden yararlanıp yararlanmayacağı belirlenir. Yerel karşıt tabanlı hafızadan yararlanma tekniğinde ise bu aşamada karşıt iki çözüm üretilir. İlk çözüm olan x HS algoritmasının işlem adımları takip edilerek uygulanırken yerel karşıt tabanlı ikinci çözüm ise Eşitlik 3.27 ile üretilir.

$$\tilde{x}_j = \begin{cases} x_{j,U} + x_{j,L} - x_j & HMCR \text{ olasılıkla} \\ x_{j,L} + rand \times (x_{j,U} - x_{j,L}) & 1 - HMCR \text{ olasılıkla} \end{cases} \quad (3.27)$$

Bu şekilde üretilen iki yeni harmoninin ikisi de harmoni hafızasından ve *HMCR* değerinden faydalanılarak ve belirlenen çözüm aralığında birbirinin karşısı iki çözüm olduğu göz önünde bulundurulduğunda algoritmanın artan keşif yeteneği sayesinde yerel optimum çözümlere takılması engellenirken aynı zamanda çözüm uzayının biririnden uzak iki farklı bölgesinde de sömürü yapma olanağı doğmaktadır.

HS algoritmasında "ton ayarlama" işlemi olarak isimlendirilen operatör, *HMCR* parametresiyle harmoni hafızası kullanılarak üretilen çözümler üzerinde *PAR* parametresi yardımıyla küçük değişiklikler yaparak algoritmanın sömürü işlemi yapmasına yardımcı olmaktadır. LHS varyantında ise bu işlem kendinden uyarlamalı olacak şekilde harmoni hafızasındaki en iyi ve en kötü çözümden faydalanılarak yapılmaktadır.

Kendinden uyarlamalı küresel ton ayarlama yönteminde η parametresi yardımıyla küresel uyarlamalı ve en kötü uyarlamalı işlem adımlarından birinin seçimi yapılmaktadır. Eşitlik 3.28 ile hesaplanan η parametresi, algoritmanın ilk iterasyonlarında hafızadaki tüm harmoniler birbirinden uzak çözümler olduğu için büyük değerler olması sağlanmakta, algoritmanın ilerleyen süreçlerinde 0.5 değeri kullanılmakta, son iterasyonlarda ise daha küçük değerler almaktadır.

$$\eta = \begin{cases} \sqrt{1 - 1.5t/T_{max}} & \text{eğer } t < T_{max}/2 \\ 0.5 & \text{eğer } T_{max}/2 \leq t < 3T_{max}/4 \\ \exp(-(\sqrt{14.53 - 14.04t/T_{max}})) & \text{aksi takdirde} \end{cases} \quad (3.28)$$

Evrimsel algoritmaların (EA) en önemli aşamalarından birisi olan seçim operatörü HS algoritmasında üretilen aday çözüm eğer ki harmoni hafızasındaki en başarısız çözümden daha kaliteli ise hafızadaki en başarısız çözümün çıkartılarak yeni çözümü dahil edilerek yapılmaktadır. Karşıt tabanlı bir varyant olan LHS

algoritmasında ise her iterasyonda iki adet aday çözüm üretilmektedir.

Rekabet-seçim şeması olarak adlandırılan yöntemde, LHS algoritmasında üretilen aday çözümlerin amaç fonksiyon değerleri hesaplanarak karşıt tabanlı olarak hesaplanan çözümlerden daha başarılı olan, harmoni hafızasındaki en başarısız çözümün kalitesinin daha iyi olup olmadığına bakılmaksızın hafızadaki en başarısız çözüm çıkartılarak hafızaya dahil edilmektedir. Bu sayede harmoni hafızasının daha dinamik olması sağlanmakta, ve küresel olarak hafızadan faydalanan operatörlerin keşif ve sömürü yeteneklerini de daha dinamik hale getirmektedir.

LHS varyantının sözde kodu Algoritma 2 ile gösterilmektedir.

Algoritma 2 LHS Algoritmasının Sözde Kodu

```
1: Parametre sayısını  $D$ , maksimum değerlendirme sayısını  $T_{max}$  olarak belirle
2:  $HMS$  ve  $HMCR$  değerlerini ata
3:
4: for  $i \leftarrow 1 \dots HMS$  do // HM'yi ilkle
5:   for  $j \leftarrow 1 \dots D$  do
6:      $x_{i,j} = x_{j,L} + rand * (x_{j,U} - x_{j,L})$ 
7:      $f(x)$  değerini hesapla
8:   end for
9: end for
10:
11: HM'yi  $f(x)$  değerlerine göre sırala
12:
13: while  $t < T_{max}$  do
14:   for  $j \leftarrow 1 \dots D$  do
15:     if  $rand() < HMCR$  then
16:        $x_{new,j} = x_{r,j}$ 
17:        $\tilde{x}_{new,j} = x_{j,U} + x_{j,L} - x_{new,j}$ 
18:        $\eta$  değerini hesapla
19:
20:       if  $rand() < \eta$  then
21:          $x_{new,j} = x_{new,j} + rand * (x_{best,j} - x_{new,j})$ 
22:       else
23:          $x_{new,j} = x_{new,j} - rand * (x_{worst,j} - x_{new,j})$ 
24:       end if
25:     else
26:        $x_{new,j} = x_{j,L} + rand * (x_{j,U} - x_{j,L})$ 
27:        $\tilde{x}_{new,j} = x_{j,L} + rand * (x_{j,U} - x_{j,L})$ 
28:     end if
29:   end for
30:
31:    $f(x_{new})$  ve  $f(\tilde{x}_{new})$  değerlerini hesapla
32:
33:   if  $f(x_{new}) < f(\tilde{x}_{new})$  then
34:      $HM_{worst} = f(x)_{new}$ 
35:   else
36:      $HM_{worst} = f(\tilde{x}_{new})$ 
37:   end if
38:
39:   HM'yi  $f(x)$  değerlerine göre sırala
40:    $t = t + 2$ 
41: end while
```

3.1.1.17. Yeni Bir Geliştirilmiş Harmoni Arama Algoritması (NEHS)

Li ve arkadaşları tarafından 2019 yılında iletim kaybını göz önünde bulundurarak dinamik ekonomik emisyon dağıtım problemlerinin çözümünde kullanılmak üzere önerilen Yeni Bir Geliştirilmiş HS (A New Enhanced HS, NEHS) algoritması önerilmiştir (Z. Li vd., 2019). Harmoni hafızasından faydalanma yöntemini değiştirerek koşmanın ilk yarısında harmoni hafızasındaki rastgele çözümlerden yararlanılırken ikinci yarısında ise sadece koşma boyunca elde edilen Küresel En İyi ve Küresel En Kötü çözümlerden faydalanılarak yeni harmoni ilklendirilmektedir, bu da yarı yarıya bir oranda gerçekleştirilmektedir. Ayrıca bw parametresi de 3 farklı değer olarak kullanılmaktadır. Keyfi olarak belirlenebilecek bu üç bw değerinden en büyüğü koşmanın ilk çeyreğinde, ikinci büyük değer koşmanın ilk çeyreği ile yarısı arasında, koşmanın geri kalan yarısında ise en küçük değer kullanılmaktadır.

3.1.1.18. Bilgi Geri Beslemeli Kendinden-Uyarlamalı HS (IFSHS)

Li ve arkadaşları tarafından 2020 yılında medikal olarak kemiklerin kesilmesi ve delinmesi işleminde titreşim destekli delme işlemini optimize etmek amacıyla geliştirilen Bilgi Geri Beslemeli Kendinden-Uyarlamalı HS (Information Feedback Self-Adaptive HS, IFSHS) algoritmasında $HMCR$, PAR ve bw değerleri geri beslemeli olarak kendiliğinden uyarlamalı şekilde dinamik olarak belirlenmektedir (S. Li vd., 2020). $HMCR$, PAR ve bw değerleri için birer üst ve alt değer belirlenerek bu üç parametre de geri besleme faktörüne bağlı olarak iteratif olarak hesaplanmaktadır.

3.1.1.19. En İyi - En Kötü - Ortalama HS Algoritması (BWMHS)

Talaei ve arkadaşları tarafından 2020 yılında önerilen En İyi-En Kötü-Ortalama HS (Best-Worst-Mean HS, BWMHS) algoritmasında $HMCR$ parametresi ile yapılan harmoni besteleme işlemi yerine en iyi, en kötü ve ortalama adında üç yeni kural kullanarak harmoni hafızasından daha fazla yararlanan bir sistem geliştirilmiştir (Talaei vd., 2020). Ortalama kuralı, harmoni hafızasındaki tüm harmonilerin ortalamasını olarak kullanır. En iyi ve en kötü kuralı ise hafızadaki en iyi ve en kötü harmonileri

kullanır. Bu üç kural kullanılarak her iterasyonda biri ortalama harmoni diğeri de olasılıksal olarak en iyi veya en kötüyü kullanan 2 adet yeni harmoni bestelenir. En iyi ve en kötü arasında seçim yapmak için Etha adı verilen ve 3.29 ile hesaplanan bir parametre kullanılır. 0.9 değerinden başlayarak gittikçe azalan Etha değeriyle ilk iterasyonlarda en iyi kuralına ağırlık verilirken ilerleyen koşmalarda ise en kötü kuralının olasılığı artmaktadır. *PAR* ve *bw* değerleri için ise IHS’de kullanılan yöntem kullanılmaktadır.

$$Etha = 0.9 \times \exp(\log(1/9) \times (t/T_{max})) \quad (3.29)$$

3.1.1.20. Doğrusal Dinamik Alan ile İyileştirilmiş Diferansiyel Tabanlı HS

Zhu ve arkadaşlarının 2020 yılında literatüre sunduğu Doğrusal Dinamik Alan ile İyileştirilmiş Diferansiyel Tabanlı HS (Improved Differential-Based HS Algorithm with Linear Dynamic Domain, ID-HS-LDD) algoritmasında iyileştirilmiş diferansiyel tabanlı hesaplama modu (ID) kullanılarak *bw* parametresi iyileştirilmiş, optimal değerlerin arama alanının doğrusal dinamik değişimi (LDD) ile arama alanının aralık parametreleri ayarlanmaktadır. *ID* modu kapsamında *bw* değeri harmoni hafızasından dinamik olarak rastgele seçilen 4 adet harmoni kullanılarak Eşitlik 3.30 ile hesaplanmaktadır (Zhu vd., 2020). Bu hesaplama yöntemiyle hafızadaki harmoniler birbirine yaklaştıkça *bw* değeri de sıfıra yaklaşmaktadır.

$$bw(t, j) = x_{r1,j} - x_{r2,j} + x_{r3,j} - x_{r4,j} \quad (3.30)$$

Hafızadan yararlanarak harmoni bestelenmesi sürecinde yakınsama hızını artırmak için yeni bir yakınsama parametresi oluşturularak zorunlu yakınsama metodu uygulanmıştır. Yakınsama parametresi olan ψ değeri 3.31 ile hesaplanmaktadır.

$$\psi = \min(1, 3 \times (1 - t/T_{max})) \quad (3.31)$$

Harmoni hafızası kullanılarak üretilen harmonilerde hafızadaki en iyi harmoniden faydalanılır. Ton ayarlaması işleminde ise Eşitlik 3.32 kullanılır.

$$x_j = \psi \text{ times}(x_{best,j} + rand \times bw(j)) \quad (3.32)$$

Parametrelerin alabileceği en küçük ve en büyük değerleri belirleyen alt ve üst değer aralığı da doğrusal dinamik arama aralığı yöntemi ile dinamik olarak güncellenmektedir. Aynı şekilde $HMCR$ ve PAR değerleri de sırasıyla $[HMCR_{min}, HMCR_{max}]$ ve $[PAR_{min}, PAR_{max}]$ aralığında iterasyon bazlı olarak güncellenmektedir.

3.1.2. Trigonometrik Harmoni Arama Algoritması (TRI-HS)

HS algoritması, 2001 yılından itibaren birçok problem üzerinde performansı ölçülmüş, ve araştırmacılar tarafından geliştirilmeye ve yeni varyantları üretilmeye devam edilen bir algoritmadır. Bu tez çalışmasında HS algoritmasının varyantlarından bazıları 3.1.1. bölümünde kısa kısa açıklanmıştır. Bahsedilen varyantların her biri HS algoritmasının temel işleyişine mümkün olduğunca bağlı kalarak kullanılan parametreler, operatörler ve çözüm üretme yöntemleri değiştirilerek belirli bir optimizasyon problemi veya yaygın performans ölçüm problemleri üzerinde öncekilerden daha başarılı sonuçlar üretmeye çalışmıştır. Bu tez çalışması kapsamında da literatürdeki varyantların incelenmesi ve belirlenen problem üzerindeki performanslarının ölçülmesi amacının yanında incelenen varyantların orijinal HS algoritması ve diğer varyantlara göre çeşitli koşullarda daha iyi veya daha kötü sonuçlar üretmesine neden olan unsurları anlamaya çalışmak ve buradan edinilen deneyimle mevcut yöntemleri geliştirmek de amaçlanmıştır.

HS varyantlarının Bölüm 4.'te yer alan yaygın performans ölçüm problemleri üzerinde yapılan incelemelerde, problemin türü ve parametre sayısına bağlı olarak varyantların başarı sıralamasının farklılık gösterdiği gözlemlenmiştir. Ancak genel olarak LHS varyantının yapılan denemelerde başarılı sonuçlar üretmek konusunda diğer varyantlardan çoğunlukla daha fazla ön plana çıktığı görülmüştür. Bu nedenle 3.1.1.16.'da anlatılan LHS varyantı üzerinde yoğunlaşmaya karar verilmiştir.

3.1.2.1. Dinamik Sinüzoidal HMCR

Trigonometrik HS (TRI-HS) adını verdiğimiz yeni HS varyantında, LHS varyantında sabit değer olarak kullanılan $HMCR$ değerini trigonometrik bir fonksiyon olan sinüs fonksiyonu ile dinamik olarak hesaplanacak şekilde güncellenmiştir. LHS algoritmasında tek ve sabit olarak kullanılan $HMCR$ parametresinin yerine $HMCR_{min}$ ve $HMCR_{max}$ parametreleri başlangıçta verilmektedir. Bu parametreler sırasıyla iterasyon bazlı hesaplanan $HMCR_i$ değerinin alabileceği minimum ve maksimum değerleri belirtir.

$HMCR$ parametresi birçok HS varyantında dinamik hale getirilmiştir. Bu varyantların çoğu iterasyon bazlı olarak $HMCR$ değerini doğrusal olarak artan veya parabolik olarak önce artıp sonra azalan şekilde güncellemektedir. $HMCR$ değeri, HS algoritmasının keşif gücünü etkileyen birinci faktör olması nedeniyle algoritmanın ilk iterasyonlarında düşük olması keşfi hızlandıracak, son iterasyonlara doğru ise yüksek olması algoritmanın keşiften çok sömürüye ağırlık vermesini sağlamaktadır. Ancak bazı durumlarda bu yaklaşım algoritmanın özellikle ilerleyen iterasyonlarda yerel minimuma takılı kalmasına neden olabilir. Bu nedenle bazı varyantlar doğrusal sürekli artış yerine $HMCR$ değerinin parabolik olarak değişmesini tercih etmişlerdir.

TRI-HS algoritmasında bu iki yaklaşımdan daha farklı olarak $HMCR$ değerinin bir sinüs dalgası şeklinde belirli bir periyotta artıp azalan ve bunu tekrarlayan bir yapıda olması önerilmiştir. $HMCR$ değerinin sinüs dalgasının batan kısmındaki değerleri daha düşük değerler olup algoritmanın keşif yeteneğini daha iyi kullanmasını sağlarken sinüs dalgasının yükseldiği kısımlardaki değerleri ise daha büyük değerler olup algoritmanın daha az keşif yaparak sömürüyü daha etkili şekilde kullanabilmesini sağlamaktadır. Bu iki durum birbirini takip edecek şekilde tekrarlandığı için algoritmanın farklı zaman aralıklarında keşif ve sömürü işlemlerine ağırlık vererek dinamik bir ilerleme süreci yürütmesini sağlamaktadır.

Örnek olarak, algoritmanın belirli bir aşamada yerel minimum sonuçları keşfettiği ve bu noktaya yoğunlaştığı bir durumda, $HMCR$ değeri sinüs dalgasının çukur kısmına geldiğinde algoritma keşif faaliyetine ağırlık vereceği için yeni ve optimum çözüme yaklaşan çözümler üretme olasılığı artacaktır. Bu sürecin devamında keşif ile elde edilen yeni çözüm, $HMCR$ değeri sinüs dalgasının tepe

kısmına geldiğinde daha büyük olacağı için algoritma keşiften ziyade sömürüye ağırlık verecek, bu şekilde de yeni keşfedilmiş çözümün komşu çözümlerinin sömürülmesine imkan tanınacaktır. Bu şekilde elde edilen yeni çözüm de bir yerel minimum olsa bile sinüs dalgasının sonraki çukur ve tepeleri yardımıyla yeni çözümler bulunma ve sömürü ile iyileştirilme imkanı doğacaktır.

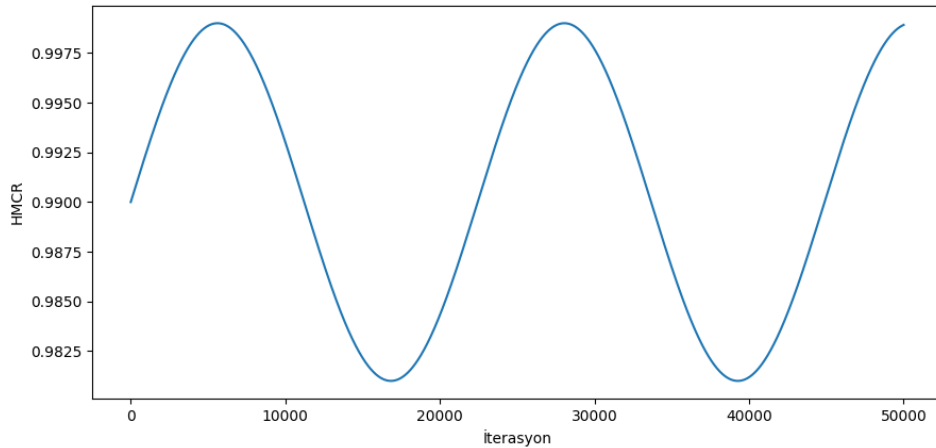
$HMCR$ değeri, Eşitlik 3.33 ve Eşitlik 3.34 ile elde edilen ortalama değer ve yükseklik değeri yardımıyla Eşitlik 3.35 kullanılarak hesaplanmaktadır. c değeri varyantın ihtiyaç duyduğu bir parametre olup sinüs dalgalarının sayısının hesaplanmasında kullanılmaktadır.

$$HMCR_{mean} = \frac{HMCR_{max} + HMCR_{min}}{2} \quad (3.33)$$

$$h = \frac{HMCR_{max} - HMCR_{min}}{2} \quad (3.34)$$

$$HMCR_t = HMCR_{mean} + \sin(t/(T_{max}/c)) * h \quad (3.35)$$

Örnek bir koşma senaryosu için 50.000 iterasyonlu bir koşmada $HMCR_{min}$ değeri 0.981 , $HMCR_{max}$ değeri 0.999 ve c değeri 15 olarak belirlenirse koşma sırasında kullanılacak $HMCR$ değerleri iterasyon bazlı olarak Şekil 3.4 ile gösterildiği gibi hesaplanacaktır.



Şekil 3.4. Sinüzoidal $HMCR$ değeri gösterimi

3.1.2.2. Küresel Ton Ayarlama

TRI-HS algoritmasında LHS varyantında dinamik olarak iterasyon bazlı hesaplanan η parametresi kaldırılarak ton ayarlama işlemi herhangi bir koşula bağlı olmaksızın karşıt tabanlı çözümler hariç üretilen tüm yeni çözümlere uygulanmaktadır. LHS varyantında yeni üretilen çözüm dinamik olarak η ihtimalle harmoni hafızasındaki en iyi çözüm veya $(1 - \eta)$ ihtimalle ise harmoni hafızasındaki en kötü çözüm kullanılarak ton ayarlama işlemine tabi tutulmakta. TRI-HS varyantında ise tüm çözümler harmoni hafızasından rastgele seçilen komşu parametre değerleri kullanılarak güncellenmektedir. TRI-HS varyantının ton ayarlama işlemi Eşitlik 3.36 ile yapılmaktadır.

$$x_{new,j} = x_{new,j} + (U(0, 1) * (x_{rand,j} - x_{new,j})) \quad (3.36)$$

3.1.2.3. Rekabet Seçim Şeması

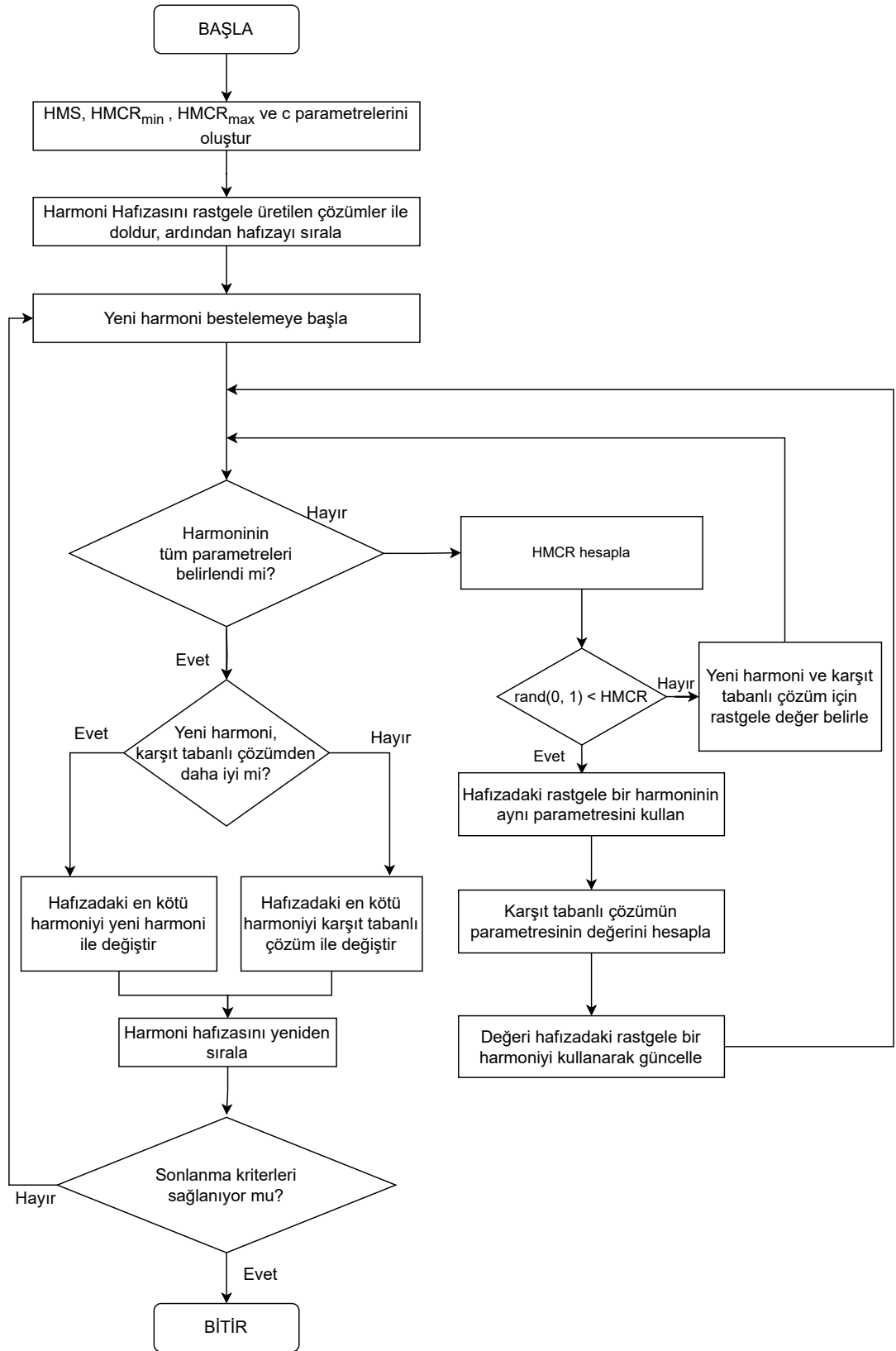
TRI-HS varyantı da LHS varyantı gibi karşıt tabanlı stratejiyi kullanarak her iterasyonda iki adet yeni çözüm ürettiği için bu çözümleri doğru şekilde kullanmak büyük önem taşımaktadır. LHS varyantında uygulanan rekabet-seçim şeması, karşıt tabanlı çözüm üretildiği için ve aynı zamanda küresel ton ayarlaması yapıldığı için harmoni hafızasının daha dinamik olması durumunda daha iyi verim alınmasını sağlamaktadır. Bu nedenle TRI-HS varyantında da rekabet seçim şeması kullanılarak, her iterasyon sonunda üretilen yeni çözüm ve onun karşıt tabanlı olarak üretilen çözümünden daha kaliteli olan, harmoni hafızasındaki en kötü çözümden daha iyi olup olmadığına bakılmaksızın hafızaya dahil edilerek hafızadaki en kötü çözüm hafızadan çıkartılmaktadır.

TRI-HS varyantının genel işleyişi Algoritma 3 ile gösterilmiştir.

TRI-HS algoritmasının işleyişini görsel olarak ifade eden akış diyagramı Şekil 3.5 ile gösterilmiştir.

Algoritma 3 TRI-HS Algoritmasının Sözde Kodu

```
1: Parametre sayısını  $D$ , maksimum değerlendirme sayısını  $T_{max}$  olarak belirle
2:  $HMS$ ,  $HMCR_{min}$ ,  $HMCR_{max}$  ve  $c$  değerlerini ata
3:
4: for  $i \leftarrow 1 \dots HMS$  do // HM'yi ilkle
5:   for  $j \leftarrow 1 \dots D$  do
6:      $x_{i,j} = x_{j,L} + rand * (x_{j,U} - x_{j,L})$ 
7:      $f(x)$  değerini hesapla
8:   end for
9: end for
10:
11: HM'yi  $f(x)$  değerlerine göre sırala
12:
13: while  $t < T_{max}$  do
14:    $HMCR_i$  değerini hesapla
15:
16:   for  $j \leftarrow 1 \dots D$  do
17:     if  $rand() < HMCR_i$  then
18:        $x_{new,j} = x_{r,j}$ 
19:        $\tilde{x}_{new,j} = x_{j,U} + x_{j,L} - x_{new,j}$ 
20:
21:        $x_{new,j} = x_{new,j} + rand * (x_{rand,j} - x_{new,j})$ 
22:     else
23:        $x_{new,j} = x_{j,L} + rand * (x_{j,U} - x_{j,L})$ 
24:        $\tilde{x}_{new,j} = x_{j,L} + rand * (x_{j,U} - x_{j,L})$ 
25:     end if
26:   end for
27:
28:    $f(x_{new})$  ve  $f(\tilde{x}_{new})$  değerlerini hesapla
29:
30:   if  $f(x_{new}) < f(\tilde{x}_{new})$  then
31:      $HM_{worst} = f(x)_{new}$ 
32:   else
33:      $HM_{worst} = f(\tilde{x}_{new})$ 
34:   end if
35:
36:   HM'yi  $f(x)$  değerlerine göre sırala
37:    $t = t + 2$ 
38: end while
```



Şekil 3.5. TRI-HS algoritmasının akış diyagramı

4. BULGULAR VE DEĞERLENDİRME

Literatürdeki çok sayıda HS varyantı incelenmiş, aralarından en bilinenler ve güncel olanlardan 20 tanesi seçilerek 3. bölümde önerdikleri değişiklikler ve işleyişleri anlatılmıştır. Bahsedilen varyantların tamamı için C programlama dilinde implementasyon çalışmaları yapılmış, yayınlardaki ile aynı parametrelerle yapılan koşmalarda elde edilen sonuçlar yayındaki sonuçlarla karşılaştırılarak birbiriyle tutarlı olan varyantların implementasyonlarının doğruluğu teyit edilmiştir.

Yapılan testler sonucunda, orijinal yayını ve / veya yaygın amaç fonksiyonları ile yapılan koşmaların sonuçlarının bulunduğu yayınlarla eşleşen sonuçlar vermiş olan HS, IHS, GHS, MHS, SAHS, PAHS1, PAHS2, PAHS3, PAHS4, LHS, NGHS ve BWMHS varyantları ile bu tez çalışmaları kapsamında geliştirilen TRI-HS varyantı eşit ve adil şartlar altında karşılaştırmaya tabi tutulmuştur.

Ele alınan HS varyantları, öncelikle yaygın olarak çeşitli optimizasyon algoritmalarının yakınsama ve optimizasyon performanslarını ölçmek ve dinamik parametre sayısı ile algoritmaların aynı formülasyona sahip olan farklı boyuttaki problemler üzerindeki performans değişimlerini ölçümlemeyi sağlayan *kıyaslama amaç fonksiyonları* üzerinde koşturulmuştur.

4.1. Kıyaslama Amaç Fonksiyonları

Kıyaslama amaç fonksiyonları Evrimsel Hesaplama Kongresi özel oturumu için yayınlanan problemlerden ve diğer yaygın kıyaslama fonksiyonları arasından seçilmiştir (Zhong vd., 2004; Suganthan vd., 2005). Tez çalışmaları kapsamında kullanılan kıyaslama fonksiyonları aşağıda açıklanmıştır.

Sphere fonksiyonu dinamik parametrelili bir fonksiyon olup, girdi vektörünün tüm elemanlarının karelerinin toplamını sonuç olarak üretmekte ve Eşitlik 4.1 ile hesaplanmaktadır. Sphere fonksiyonunun tüm parametreleri $[-100, 100]$ aralığında değerler alabilir ve girdi vektörünün tüm elemanları 0 olduğunda küresel minimum noktası olan 0 değerini üretmektedir.

$$f(x) = \sum_{i=1}^D x_i^2 \quad (4.1)$$

Rosenbrock fonksiyonu dinamik parametrelili bir fonksiyon olup Eşitlik 4.2 ile hesaplanmaktadır. Rosenbrock fonksiyonunun tüm parametreleri $[-30, 30]$ aralığında değerler alabilir ve girdi vektörünün tüm elemanları 0 değerini aldığında küresel minimum değeri olan 0 sonucunu üretmektedir.

$$f(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2) \quad (4.2)$$

Rastrigin fonksiyonu dinamik parametrelili bir fonksiyon olup Eşitlik 4.3 ile hesaplanmaktadır. Rastrigin fonksiyonunun tüm parametreleri $[-5.12, 5.12]$ aralığında değerler alabilir ve girdi vektörünün tüm elemanları 0 değerini aldığında küresel minimum değeri olan 0 sonucunu üretmektedir.

$$f(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad (4.3)$$

Ackley fonksiyonu dinamik parametrelili bir fonksiyon olup Eşitlik 4.4 ile hesaplanmaktadır. Ackley fonksiyonunun tüm parametreleri $[-32, 32]$ aralığında değerler alabilir ve girdi vektörünün tüm elemanları 0 değerini aldığında küresel minimum değeri olan 0 sonucunu üretmektedir.

$$f(x) = -20 \exp(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}) - \exp(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)) + 20 + e \quad (4.4)$$

Griewank fonksiyonu dinamik parametrelili bir fonksiyon olup Eşitlik 4.5 ile hesaplanmaktadır. Griewank fonksiyonunun tüm parametreleri $[-600, 600]$ aralığında değerler alabilir ve girdi vektörünün tüm elemanları 0 değerini aldığında küresel minimum değeri olan 0 sonucunu üretmektedir.

$$f(x) = \sum_{i=1}^D (\frac{x_i^2}{4000}) - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}}) + 1 \quad (4.5)$$

Step fonksiyonu dinamik parametrelili bir fonksiyon olup Eşitlik 4.6 ile hesaplanmaktadır. Step fonksiyonunun tüm parametreleri $[-100, 100]$ aralığında değerler alabilir ve girdi vektörünün tüm elemanları 0 değerini aldığında küresel minimum değeri olan 0 sonucunu üretmektedir.

$$f(x) = \sum_{i=1}^D (\lfloor x_i + 0.5 \rfloor)^2 \quad (4.6)$$

Schwefel 2.22 fonksiyonu dinamik parametrelili bir fonksiyon olup Eşitlik 4.7 ile hesaplanmaktadır. Schwefel 2.22 fonksiyonunun tüm parametreleri $[-10, 10]$ aralığında değerler alabilir ve girdi vektörünün tüm elemanları 0 değerini aldığında küresel minimum değeri olan 0 sonucunu üretmektedir.

$$f(x) = \sum_{i=1}^D |x_i| + \prod_{i=1}^D |x_i| \quad (4.7)$$

Generalized Schwefel 2.26 fonksiyonu dinamik parametrelili bir fonksiyon olup Eşitlik 4.8 ile hesaplanmaktadır. Generalized Schwefel 2.26 fonksiyonunun tüm parametreleri $[-500, 500]$ aralığında değerler alabilir ve girdi vektörünün tüm elemanları 420.9687 değerini aldığında küresel minimum değeri olan -12569.5 sonucunu üretmektedir.

$$f(x) = - \sum_{i=1}^D (x_i \sin(\sqrt{|x_i|})) \quad (4.8)$$

Rotated hyper ellipsoid fonksiyonu dinamik parametrelili bir fonksiyon olup Eşitlik 4.9 ile hesaplanmaktadır. Rotated hyper ellipsoid fonksiyonunun tüm parametreleri $[-100, 100]$ aralığında değerler alabilir ve girdi vektörünün tüm elemanları 0 değerini aldığında küresel minimum değeri olan 0 sonucunu üretmektedir.

$$f(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right) \quad (4.9)$$

Shifted sphere fonksiyonu, Sphere fonksiyonunun belirli bir değerde kaydırılmış dinamik parametrelili bir sürümü olup, girdi vektörünün tüm elemanlarının karelerinin toplamının kaydırılan değer ile toplamını vermektedir. Eşitlik 4.10 ile hesaplanan bu

fonksiyonun kaydırılma miktarı -450 olarak belirlenmiştir. Shifted sphere fonksiyonunun tüm parametreleri $[-100, 100]$ aralığında değerler alabilir ve girdi vektörünün tüm elemanları 0 değerini aldığında küresel minimum değeri olan kaydırılma miktarına eşit olarak -450 sonucunu üretmektedir.

$$f(x) = \left(\sum_{i=1}^D x_i^2 \right) - 450 \quad (4.10)$$

4.2. Deney Parametreleri

Her bir deney için 30 bağımsız koşma gerçekleştirilerek sonuçları kaydedilmiştir. Tüm varyantlar için ortak parametreler olan **HMS** değeri kıyaslama fonksiyonları için 5, EEG problemi için 5 ve 10, maksimum değerlendirme sayısı (T_{max}) ise 50.000 olarak belirlenmiştir.

HS varyantlarının hem kıyaslama fonksiyonları hem de EEG sinyali optimizasyonu problemi için varyantların yayınlandığı çalışmalarda önerilen parametreler kullanılarak koşma işlemleri gerçekleştirilmiştir. Farklı HS varyantları, parametreler ve işleyişe olan etkileri üzerinde de değişiklik yapma yoluna gittiği için her bir varyant için farklı parametre setleri belirlenmesi gerekmektedir. Varyantların yayınlandığı çalışmalarda yer alan ve bu tez çalışmaları kapsamında da kullanılan parametreler aşağıda açıklanmıştır.

Orijinal HS algoritması, *HMCR*, *PAR* ve *bw* parametreleri ile koşturulmaktadır. Koşmalarda HS algoritması için Tablo 4.1 ile gösterilen parametreler kullanılmıştır.

Tablo 4.1. HS Parametreleri

Parametre	Değer
<i>HMCR</i>	0.90
<i>PAR</i>	0.30
<i>bw</i>	0.01

IHS algoritması, *PAR* ve *bw* parametrelerini belirlenen aralıkta dinamik olarak hesaplamaktadır. Koşmalarda IHS algoritması için Tablo 4.2 ile gösterilen parametreler kullanılmıştır.

Tablo 4.2. IHS Parametreleri

Parametre	Değer
$HMCR$	0.95
PAR_{min}	0.35
PAR_{max}	0.99
bw_{min}	0.000001
bw_{max}	$\frac{1}{20}(x_{i,U} - x_{i,L})$

GHS algoritması, PAR değerini belirlenen aralıkta dinamik olarak hesaplamakta ve bw parametresini kullanmamaktadır. Koşmalarda GHS algoritması için Tablo 4.3 ile gösterilen parametreler kullanılmıştır.

Tablo 4.3. GHS Parametreleri

Parametre	Değer
$HMCR$	0.90
PAR_{min}	0.01
PAR_{max}	0.99

MHS algoritması, HS parametrelerine ek olarak her iterasyonda kaç yeni harmoni üretileceğini belirleyen N_{hm} ve yeni sabitleri de parametre olarak kullanmaktadır. Koşmalarda MHS algoritması için Tablo 4.4 ile gösterilen parametreler kullanılmıştır.

Tablo 4.4. MHS Parametreleri

Parametre	Değer
$HMCR$	0.98
PAR	0.10
N_{hm}	$[0.1 * HMS]$
C_1	0.10
C_2	0.50
C_3	0.80
NM_1	500
NM_2	200

SAHS algoritması, PAR değerini belirlenen aralıkta dinamik olarak hesaplamakta ve bw parametresini kullanmamaktadır. Koşmalarda SAHS algoritması için Tablo 4.5 ile gösterilen parametreler kullanılmıştır.

Tablo 4.5. SAHS Parametreleri

Parametre	Değer
$HMCR$	0.99
PAR_{min}	0.0
PAR_{max}	1.0

NGHS algoritması, HS parametrelerinin hiç birini kullanmamakta, Tablo 4.6 ile de gösterildiği üzere sadece mutasyon olasılığı parametresini kullanmaktadır.

Tablo 4.6. NGHS Parametreleri

Parametre	Değer
p_m	0.005

PAHS algoritması $HMCR$, PAR ve bw parametrelerinin hepsini belirlenen aralıkta dinamik olarak hesaplamaktadır. PAHS1, PAHS2, PAHS3 ve PAHS4 algoritmaları için Tablo 4.7 ile gösterilen parametreler kullanılmıştır.

Tablo 4.7. PAHS Parametreleri

Parametre	Değer
$HMCR_{min}$	0.70
$HMCR_{max}$	0.99
PAR_{min}	0.01
PAR_{max}	0.99
bw_{min}	0.001
bw_{max}	$\frac{1}{20}(x_{i,U} - x_{i,L})$

LHS algoritması, Tablo 4.8 ile gösterildiği üzere $HMCR$ parametresine ihtiyaç duymakta, η parametresini ise dinamik olarak hesaplamaktadır.

Tablo 4.8. LHS Parametreleri

Parametre	Değer
$HMCR$	0.99

BWMHS algoritması, PAR ve bw parametrelerini belirlenen aralıkta dinamik olarak hesaplamaktadır. BWMHS algoritması için Tablo 4.9 ile gösterilen parametreler

kullanılmıştır.

Tablo 4.9. BWMHS Parametreleri

Parametre	Değer
$HMCR$	0.90
PAR_{min}	0.01
PAR_{max}	0.99
bw_{min}	0.0001
bw_{max}	$\frac{x_{i,U}-x_{i,L}}{20}$

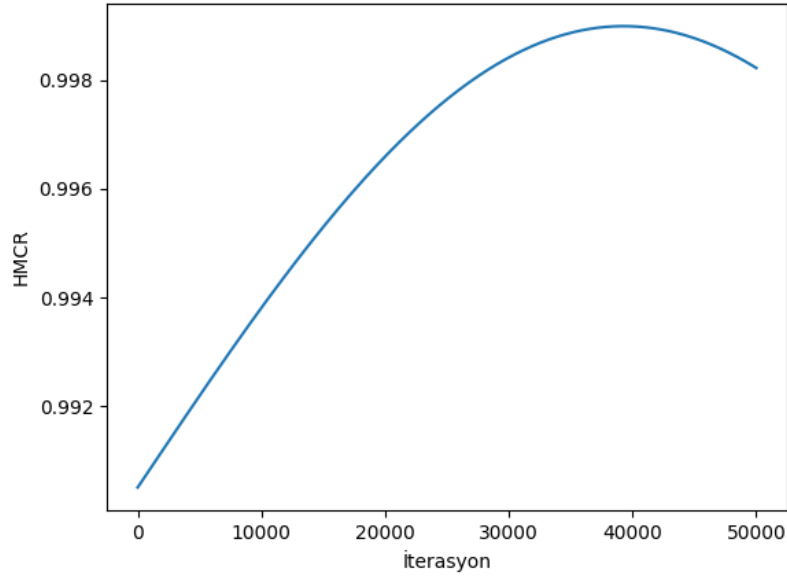
TRI-HS algoritması, LHS algoritması geliştirilerek oluşturulduğu için HS algoritmasındaki PAR ve bw parametrelerini kullanmamaktadır. Ancak LHS algoritmasından farklı olarak sabit bir $HMCR$ değeri yerine sinüzoidal şekilde iterasyon bazlı olarak $HMCR_{min}$ ve $HMCR_{max}$ değer aralığında dinamik olarak hesaplanan $HMCR_i$ parametresini kullanmaktadır. TRI-HS için bu çalışma kapsamında yapılan deneyler sonucu genel olarak başarılı sonuç verdiği gözlemlenen ve kullanılması önerilen parametre seti Tablo 4.10 ile gösterilmiştir.

Tablo 4.10. TRI-HS Parametreleri

Parametre	Değer
$HMCR_{min}$	0.981
$HMCR_{max}$	0.999
C	2

TRI-HS algoritması için kullanılan $HMCR_{min}$ ve $HMCR_{max}$ parametreleri, $HMCR$ değerinin dinamik olarak hesaplanması sırasında takip edilen sinüs dalgasının tepe ve çukur noktalarının derinliğini belirlemektedir. Çukur nokta olan 0.981 değeri kullanıldığında rastgele çözüm üretilme ihtimali %1.9 olurken, sinüs dalgasının tepe noktası olan 0.999 değeri kullanıldığında ise bu oran %0.1 seviyesine kadar düşmektedir. Her ne kadar sayısal olarak bakıldığında 0.981 ve 0.999 değerleri birbirine çok yakın olarak algılansa da çukur değerde iken rastgele çözüm üretme ihtimali tepe değerde iken üretme ihtimalinin 19 katıdır. C değeri 2 olarak seçildiği için tam bir sinüs dalgası tamamlanmayarak orta nokta olan 0.99 seviyesinden başlayan $HMCR$ değeri en yüksek noktası olan 0.999 seviyesine ulaştıktan sonra bir miktar düşerek koşma tamamlanmaktadır. Verilen parametrelerle elde edilen $HMCR$

değerler eğrisi Şekil 4.1 ile gösterilmiştir.



Şekil 4.1. Deneylerde kullanılan parametrelerle elde edilen *HMCR* değerleri

4.3. Deneysel Çıktılar

HS varyantlarının kıyaslama fonksiyonları üzerindeki performansının incelenmesi için tüm HS varyantları, 30, 50, 100, 250 ve 500 parametrelili kıyaslama fonksiyonları üzerinde tekrarlı bağımsız koşmalar gerçekleştirmiştir. Aşağıda farklı problemler için oluşturulan tablolarda koyu yazı tipiyle yazılan değerler minimum yani en iyi değeri göstermektedir.

4.3.1. Kıyaslama Fonksiyonları Deneysel Çıktıları

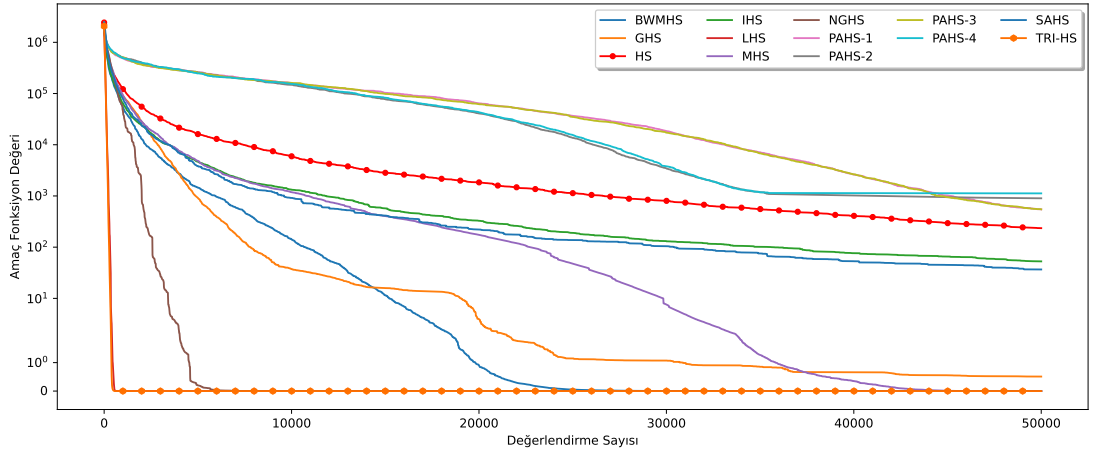
Sphere fonksiyonu için HS varyantlarının 30, 50, 100, 250 ve 500 parametrelili problem için 50.000 değerlendirme sayısı sonucunda 30 bağımsız koşma ile elde ettiği sonuçların ortalaması ve standart sapması Tablo 4.11 ile gösterilmiştir. Yine Sphere fonksiyonu için 30 parametrelili problemde HS varyantlarının sergilediği yakınsama grafiği Şekil 4.2 ile görselleştirilmiştir. Tablodaki sonuçlara genel olarak bakıldığında, tüm algoritmaların daha düşük sayıda parametre ile daha iyi sonuçlar elde ettiği görülmektedir. Parametre sayısı arttıkça elde edilen en iyi değerlerin de katlanarak büyüdüğü kolaylıkla görülmektedir. Koyu yazı tipiyle öne çıkartılan en iyi değerlerin

algoritmalarla göre dağılımı incelendiğinde LHS ve TRI-HS varyantlarının bu kıyaslamada en başarılı varyantlar olduğu görülmektedir. PAHS varyantları haricindeki diğer HS varyantlarının ise orijinal HS algoritmasına yakın ya da bir miktar daha iyi sonuçlar ürettiği söylenebilir.

Tablo 4.11. Sphere fonksiyonu HS varyant sonuçları

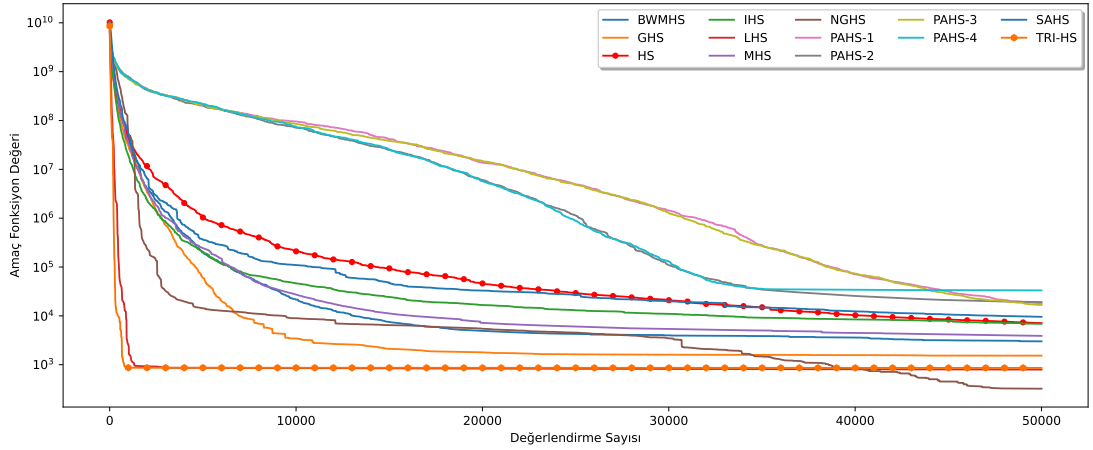
Algoritma	Fn.	D = 30	D = 50	D = 100	D = 250	D = 500
HS	Ort.	7.8328e+00	4.9862e+02	1.9669e+04	1.6855e+05	5.7096e+05
	Std.	(4.41e+00)	(7.97e+01)	(1.96e+03)	(1.21e+04)	(1.45e+04)
IHS	Ort.	1.7600e+00	2.3327e+01	2.4714e+03	8.3489e+04	3.3851e+05
	Std.	(7.83e-01)	(5.69e+00)	(3.78e+02)	(5.62e+03)	(1.49e+04)
GHS	Ort.	1.7001e-02	2.8881e+00	5.8677e+03	9.0866e+04	3.6774e+05
	Std.	(3.22e-02)	(3.94e+00)	(1.47e+03)	(9.22e+03)	(2.23e+04)
MHS	Ort.	2.8727e-06	2.4736e-01	9.4761e+01	1.7036e+04	1.5546e+05
	Std.	(2.49e-06)	(3.68e-01)	(2.02e+01)	(1.18e+03)	(8.73e+03)
NGHS	Ort.	5.155e-15	6.4093e-08	6.0818e-02	8.6294e+02	4.1469e+04
	Std.	(1.41e-14)	(4.53e-08)	(2.28e-02)	(1.19e+02)	(1.73e+03)
PAHS-1	Ort.	1.7807e+01	1.8011e+02	3.8334e+03	6.6376e+04	2.7951e+05
	Std.	(5.35e+00)	(4.35e+01)	(5.08e+02)	(4.75e+03)	(1.23e+04)
PAHS-2	Ort.	3.0006e+01	3.6176e+02	5.2369e+03	6.4774e+04	2.5151e+05
	Std.	(8.7e+00)	(1.08e+02)	(6.75e+02)	(3.77e+03)	(1.33e+04)
PAHS-3	Ort.	1.8252e+01	1.8279e+02	3.9155e+03	6.5897e+04	2.7648e+05
	Std.	(6.86e+00)	(3.74e+01)	(6.39e+02)	(5.57e+03)	(1.05e+04)
PAHS-4	Ort.	3.7419e+01	3.6326e+02	5.0600e+03	6.2459e+04	2.5835e+05
	Std.	(1.55e+01)	(1.02e+02)	(7.46e+02)	(4.83e+03)	(1.25e+04)
LHS	Ort.	3.8106e-317	9.4799e-223	3.6999e-106	2.0166e-11	3.6338e+02
	Std.	(2.05e-316)	(2.97e-222)	(1.34e-105)	(3.53e-11)	(1.69e+02)
BWMHS	Ort.	5.4807e-10	2.6571e+01	1.5347e+04	1.831e+05	6.3071e+05
	Std.	(3.58e-10)	(1.69e+01)	(2.40e+03)	(9.04e+03)	(1.80e+04)
SAHS	Ort.	1.2230e+00	4.6537e+00	2.6281e+01	1.8851e+03	7.2290e+04
	Std.	(6.97e-01)	(2.83e+00)	(9.29e+00)	(2.39e+02)	(4.90e+03)
TRI-HS	Ort.	0	0	0	0	1.234e-291
	Std.	(0)	(0)	(0)	(0)	(6.39e-291)

Sphere fonksiyonu için varyantların yakınsama grafiğini içeren Şekil 4.2 incelendiğinde, birçok varyantın optimum değer olan 0 değerine yaklaşmayı başardığı görülmektedir. Grafiğin Y eksenini için simetrik logaritmik gösterim tercih edilmesine rağmen başlangıç çözümleri çok yüksek değerler olduğu için birçok varyant belirli bir aşama sonrasında eşitlenmiş gibi görülmektedir. Ayrıca LHS varyantı TRI-HS ile çok benzer bir yakınsama eğrisi oluşturduğu için grafikte TRI-HS'nin yakınsama eğrisi ile ayırt edilememektedir. Tablo 4.11 ile birlikte değerlendirildiğinde sıfıra çok yakınsanan yerlerdeki fark daha iyi anlaşılmaktadır.



Şekil 4.2. Sphere fonksiyonu için HS varyantlarının yakınsama grafiği (D = 30)

Rosenbrock fonksiyonu için 30, 50, 100, 250 ve 500 parametrelili problemler için HS varyantlarıyla gerçekleştirilen koşmalardan elde edilen ortalama sonuçlar ve standart sapma değerleri Tablo 4.12 ile gösterilmiştir. 30 parametre için NGHS, 50 ve 100 parametre için LHS, 250 ve 500 parametrelili problemde ise TRI-HS algoritması en iyi sonucu üretmiştir. HS varyantlarının koşmalar sırasında elde ettikleri ortalama yakınsama grafiği Şekil 4.3 ile gösterilmiştir.



Şekil 4.3. Rosenbrock fonksiyonu için HS varyantlarının yakınsama grafiği

Rastrigin fonksiyonu için 30, 50, 100, 250 ve 500 parametre için HS varyantlarının koşturulmasıyla elde edilen sonuçların ortalama ve standart sapma değerleri Tablo 4.13 ile gösterilmiştir. Tabloda koyu yazı tipi ile gösterilen en iyi sonuçlar incelendiğinde LHS ve TRI-HS algoritmalarının bu problem için en başarılı sonuçları ürettiği söylenebilir.

Tablo 4.12. Rosenbrock fonksiyonu HS varyant sonuçları

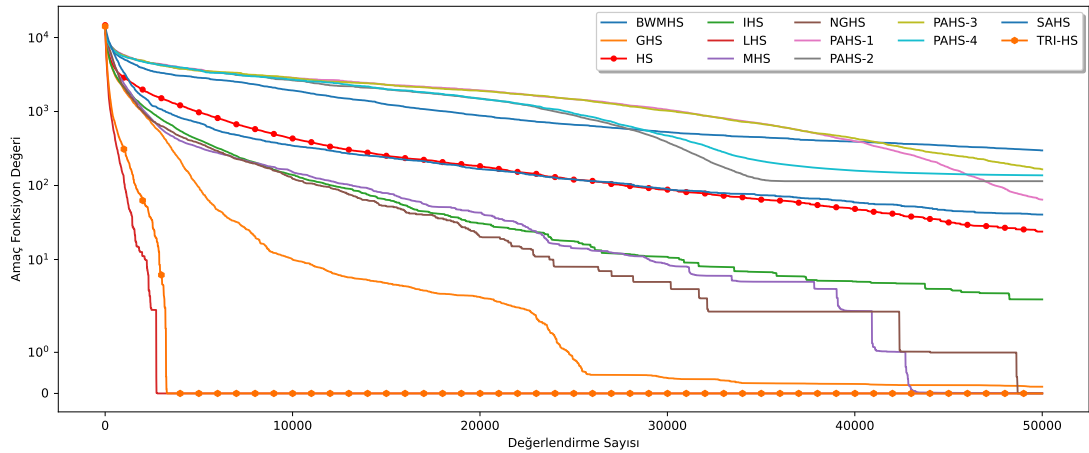
Algoritma	Fn.	D = 30	D = 50	D = 100	D = 250	D = 500
HS	Ort.	2.3722e+02	2.8002e+04	1.4138e+07	3.8558e+08	1.6686e+09
	Std.	(1.34e+02)	(1.18e+04)	(2.31e+06)	(3.93e+07)	(9.11e+07)
IHS	Ort.	2.2881e+02	9.3486e+02	2.4141e+05	1.2131e+08	8.0261e+08
	Std.	(1.41e+02)	(5.27e+02)	(7.87e+04)	(1.44e+07)	(5.67e+07)
GHS	Ort.	5.1121e+01	2.9049e+02	3.6585e+06	2.2808e+08	1.1087e+09
	Std.	(6.04e+01)	(4.8e+02)	(1.22e+06)	(3.13e+07)	(7.68e+07)
MHS	Ort.	1.2988e+02	2.8715e+02	2.2558e+03	4.9512e+06	2.1493e+08
	Std.	(3.41e+02)	(6.01e+02)	(6.98e+02)	(6.02e+05)	(1.58e+07)
NGHS	Ort.	1.0761e+01	1.3295e+02	6.1903e+02	2.4894e+04	1.0310e+07
	Std.	(1.66e+01)	(2.86e+02)	(6.29e+02)	(4.33e+03)	(9.88e+05)
PAHS-1	Ort.	6.0273e+02	7.1067e+03	6.8057e+05	7.2730e+07	5.9046e+08
	Std.	(3.35e+02)	(3.74e+03)	(2.22e+05)	(8.41e+06)	(3.36e+07)
PAHS-2	Ort.	6.4004e+02	1.0567e+04	1.1400e+06	6.9883e+07	4.8792e+08
	Std.	(6.86e+02)	(5.86e+03)	(3.85e+05)	(8.23e+06)	(3.88e+07)
PAHS-3	Ort.	5.5824e+02	7.1422e+03	7.275e+05	7.5101e+07	5.9362e+08
	Std.	(2.29e+02)	(2.75e+03)	(1.91e+05)	(1.08e+07)	(4.52e+07)
PAHS-4	Ort.	1.1092e+03	1.7525e+04	1.2544e+06	6.9763e+07	5.0494e+08
	Std.	(7.12e+02)	(9.40e+03)	(3.62e+05)	(7.97e+06)	(4.77e+07)
LHS	Ort.	2.6792e+01	4.7619e+01	9.849e+01	2.4886e+02	2.1664e+04
	Std.	(7.84e-01)	(6.08e-01)	(2.73e-01)	(1.12e-01)	(1.64e+04)
BWMHS	Ort.	8.3186e+01	1.5537e+04	4.2695e+07	8.8190e+08	3.0184e+09
	Std.	(1.33e+02)	(1.3e+04)	(1.17e+07)	(1.08e+08)	(3.69e+08)
SAHS	Ort.	3.1842e+02	5.9479e+02	2.2528e+03	2.1429e+05	7.6798e+07
	Std.	(2.01e+02)	(1.99e+02)	(8.67e+02)	(5.54e+04)	(9.76e+06)
TRI-HS	Ort.	2.8666e+01	4.8705e+01	9.8720e+01	2.4874e+02	4.9880e+02
	Std.	(1.33e-01)	(1.64e-01)	(1.06e-01)	(8.12e-02)	(5.17e-02)

Rastrigin fonksiyonu için 30 parametrelili problemde HS varyantlarının oluşturduğu yakınsama grafiği ise Şekil 4.4 ile gösterilmiştir. Birçok varyantın orijinal HS'den daha iyi sonuç ürettiğinin görülebileceği grafikte, LHS ve TRI-HS varyantlarının diğer varyantlara göre çok daha erken bir yakınsama performansı gösterdiği, ardından GHS, MHS ve NGHS'nin ise kademeli olarak yakınsama performansı gösterdiği, IHS'nin ise belirli bir noktadan itibaren daha yavaş bir şekilde rakiplerine yaklaştırmaya başladığı görülmektedir. PAHS varyantları ise orijinal HS'den daha başarısız veya benzer düzeyde sonuçlar elde edebilmiştir.

Ackley fonksiyonu için HS varyantları ile yapılan koşmalarda elde edilen sonuçlar Tablo 4.14 ile gösterilmiştir. LHS ve TRI-HS varyantlarının bu koşmalarda ortalama en iyi sonuçları elde ettiği görülmektedir. MHS, NGHS ve BWMHS varyantlarının da genel olarak orijinal HS algoritmasından daha başarılı sonuçlar elde ettiği görülmektedir. Bu problem için PAHS varyantlarının ise HS algoritmasına yakın

Tablo 4.13. Rastrigin fonksiyonu HS varyant sonuçları

Algoritma	Fn.	D = 30	D = 50	D = 100	D = 250	D = 500
HS	Ort.	7.9438e-01	4.0825e+01	3.1892e+02	1.6796e+03	4.7987e+03
	Std.	(7.94e-01)	(5.07e+00)	(2.76e+01)	(5.87e+01)	(8.27e+01)
IHS	Ort.	9.6283e-02	8.0152e+00	1.3649e+02	1.0453e+03	3.4016e+03
	Std.	(1.50e-01)	(2.11e+00)	(1.05e+01)	(3.43e+01)	(8.91e+01)
GHS	Ort.	5.4469e-03	3.6293e-01	8.8964e+01	7.5678e+02	2.8901e+03
	Std.	(1.20e-02)	(6.21e-01)	(2.40e+01)	(7.88e+01)	(1.78e+02)
MHS	Ort.	2.4414e-04	2.253e-01	1.3321e+01	5.0896e+02	2.001e+03
	Std.	(1.79e-04)	(4.27e-01)	(3.28e+00)	(2.82e+01)	(6.75e+01)
NGHS	Ort.	1.7892e-06	3.3175e-02	1.2739e+01	4.2859e+02	2.4347e+03
	Std.	(9.64e-06)	(1.79e-01)	(3.41e+00)	(3.09e+01)	(7.11e+01)
PAHS-1	Ort.	2.1473e+00	2.3755e+01	1.5626e+02	9.6548e+02	3.0465e+03
	Std.	(1.27e+00)	(3.08e+00)	(1.15e+01)	(4.45e+01)	(1.03e+02)
PAHS-2	Ort.	3.8140e+00	1.9369e+01	8.9049e+01	5.5545e+02	2.2059e+03
	Std.	(1.74e+00)	(3.60e+00)	(1.03e+01)	(3.44e+01)	(8.87e+01)
PAHS-3	Ort.	5.5117e+00	2.8017e+01	1.5473e+02	9.6777e+02	3.0477e+03
	Std.	(1.62e+00)	(4.71e+00)	(1.55e+01)	(4.10e+01)	(9.45e+01)
PAHS-4	Ort.	4.5763e+00	2.7590e+01	1.5328e+02	8.9921e+02	2.8749e+03
	Std.	(1.94e+00)	(4.45e+00)	(1.18e+01)	(4.91e+01)	(9.44e+01)
LHS	Ort.	0	1.0955e-76	0	2.7303e-01	8.6893e+00
	Std.	(0)	(3.65e-77)	(0)	(1.47e+00)	(4.68e+01)
BWMHS	Ort.	2.6054e-01	1.4493e+02	2.6415e+01	9.2423e+01	8.599e+02
	Std.	(1.40e+00)	(2.02e+01)	(1.42e+02)	(4.98e+02)	(2.19e+03)
SAHS	Ort.	1.3489e+00	3.5246e+00	1.9365e+01	3.0099e+02	1.7143e+03
	Std.	(7.34e-01)	(1.35e+00)	(3.08e+00)	(1.41e+01)	(6.42e+01)
TRI-HS	Ort.	0	0	0	0	2.4263e+00
	Std.	(0)	(0)	(0)	(0)	(1.31e+01)



Şekil 4.4. Rastrigin fonksiyonu için HS varyantlarının yakınsama grafiği

ve zaman zaman daha başarısız sonuçlar elde ettiği söylenebilir.

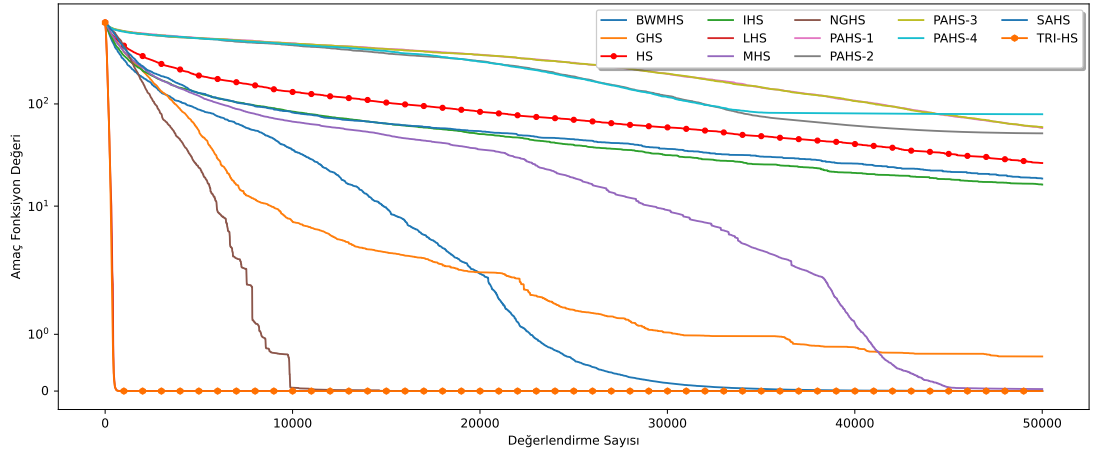
Ackley fonksiyonu için HS varyantlarının 30 parametrelili koşmada elde ettikleri yakınsama grafiği Şekil 4.5 ile gösterilmiştir. LHS varyantı TRI-HS ile çok benzer bir

yakınsama eğrisi oluşturduğu için çizimde fark edilmesi zordur. Hemen ardından NGHS ve GHS varyantları hızlı bir yakınsama eğrisi oluştururken GHS bu yakınsama hızını koruyamamış ve 20.000 değerlendirmenin ardından BMWHS ve MHS'nin gerisinde kalmıştır. PAHS varyantları ise HS algoritmasından daha kötü bir yakınama performansı göstermişlerdir.

Tablo 4.14. Ackley fonksiyonu HS varyant sonuçları

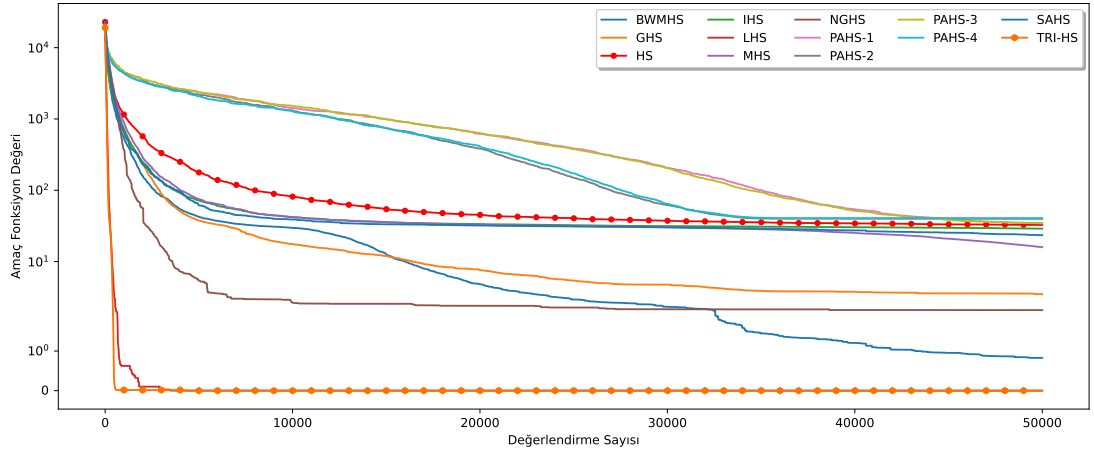
Algoritma	Fn.	D = 30	D = 50	D = 100	D = 250	D = 500
HS	Ort.	8.799e-01	5.359e+00	1.3511e+01	1.7828e+01	1.9313e+01
	Std.	(4.26e-01)	(4.65e-01)	(4.11e-01)	(1.59e-01)	(7.38e-02)
IHS	Ort.	5.4166e-01	1.9786e+00	7.0839e+00	1.5364e+01	1.7778e+01
	Std.	(1.37e-01)	(1.90e-01)	(3.09e-01)	(2.68e-01)	(1.25e-01)
GHS	Ort.	2.0316e-02	3.1906e-01	8.9696e+00	1.5419e+01	1.7834e+01
	Std.	(2.08e-02)	(3.80e-01)	(9.95e-01)	(4.16e-01)	(2.23e-01)
MHS	Ort.	1.1814e-03	1.6678e-01	2.1276e+00	9.8169e+00	1.5135e+01
	Std.	(5.37e-04)	(2.9e-01)	(1.64e-01)	(2.54e-01)	(1.85e-01)
NGHS	Ort.	1.6065e-08	5.6799e-05	3.7465e-02	4.0846e+00	1.0746e+01
	Std.	(1.22e-08)	(2.16e-05)	(7.99e-03)	(1.49e-01)	(1.41e-01)
PAHS-1	Ort.	1.9372e+00	3.8341e+00	8.1672e+00	1.4445e+01	1.7225e+01
	Std.	(3.44e-01)	(2.68e-01)	(4.66e-01)	(3.16e-01)	(1.71e-01)
PAHS-2	Ort.	1.7186e+00	3.9481e+00	8.4798e+00	1.4106e+01	1.679e+01
	Std.	(4.50e-01)	(4.09e-01)	(5.17e-01)	(2.44e-01)	(2.e-01)
PAHS-3	Ort.	1.9781e+00	3.8365e+00	8.0552e+00	1.4562e+01	1.7206e+01
	Std.	(3.43e-01)	(3.41e-01)	(4.30e-01)	(2.75e-01)	(1.80e-01)
PAHS-4	Ort.	2.6436e+00	4.7184e+00	8.6725e+00	1.4438e+01	1.6920e+01
	Std.	(4.39e-01)	(4.08e-01)	(4.68e-01)	(3.07e-01)	(1.98e-01)
LHS	Ort.	3.8784e-15	3.9968e-15	4.5889e-15	1.2107e-07	1.302e+00
	Std.	(6.38e-16)	(0)	(1.32e-15)	(1.47e-07)	(3.25e-01)
BMWHS	Ort.	2.2563e-05	5.6796e-02	4.3417e-01	6.0286e-01	6.5397e-01
	Std.	(7.56e-06)	(3.06e-01)	(2.34e+00)	(3.25e+00)	(3.52e+00)
SAHS	Ort.	6.2004e-01	9.0963e-01	2.1526e+00	7.2153e+00	1.485e+01
	Std.	(2.68e-01)	(3.26e-01)	(4.52e-01)	(1.33e+00)	(1.20e+00)
TRI-HS	Ort.	3.9968e-15	3.9968e-15	3.9968e-15	3.9968e-15	3.9968e-15
	Std.	(0)	(0)	(0)	(0)	(0)

Griewank fonksiyonu için HS varyantlarının bağımsız koşmalar ile elde ettiği sonuçların ortalama ve standart sapma değerleri Tablo 4.15 ile gösterilmiştir. LHS ve TRI-HS varyantlarının en iyi sonuçları elde ettiği, hatta TRI-HS varyantının tüm parametreler için küresel optimum olan 0 değerini tüm bağımsız koşmalarda elde edebildiği görülmekte olup neredeyse tüm varyantların orijinal HS algoritmasından daha başarılı sonuçlar elde edebildiği söylenebilir.



Şekil 4.5. Ackley fonksiyonu için HS varyantlarının yakınsama grafiği

Griewank fonksiyonu için HS varyantlarının 30 parametrelilikte iterasyon bazlı yakınsama grafiği Şekil 4.6 ile görselleştirilmiştir. LHS ve TRI-HS varyantlarının yine çok benzer bir yakınsama eğrisi oluşturduğu, NGHS, GHS ve BWMHS varyantlarının sonradan logaritmik olarak bu varyantlara yaklaştığı görülmektedir. MHS varyantı ise daha da ilerleyen aşamalarda daha iyi sonuçlar vermeye başlamış ancak 0'a yakınsayan diğer varyantlara ulaşamamıştır.



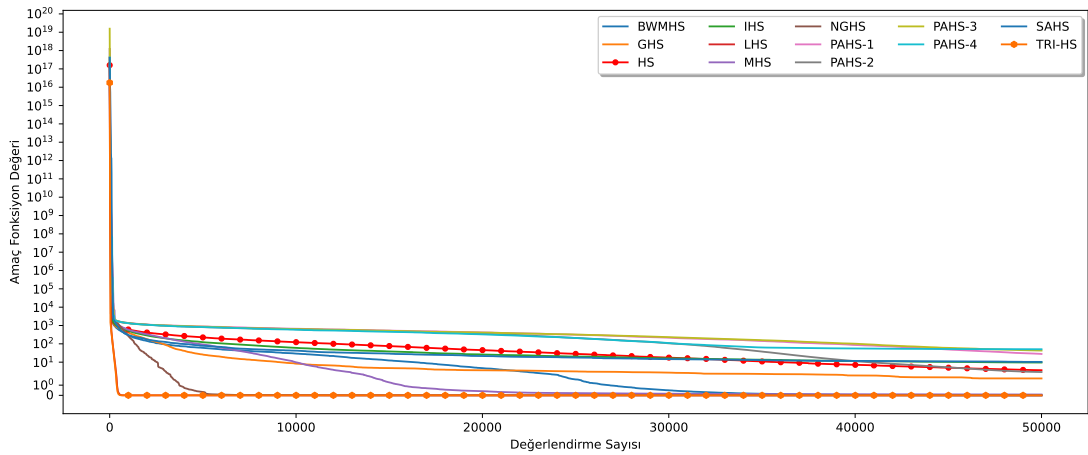
Şekil 4.6. Griewank fonksiyonu için HS varyantlarının yakınsama grafiği

Schwefel 2.22 fonksiyonu için HS varyantlarının elde ettiği sonuçlar Tablo 4.16 ile gösterilmiştir. LHS ve TRI-HS varyantları en iyi sonuçları elde etmiştir. Schwefel 2.22 fonksiyonu için 30 parametrelilikte koşmada HS varyantlarının yakınsama grafikleri Şekil 4.7 ile gösterilmiştir. İlk elde edilen değerlerin çok yüksek olmasından kaynaklı olarak Y eksenini simetrik logaritmik ölçüde gösterilmesine rağmen birçok varyantın

Tablo 4.15. Griewank fonksiyonu HS varyant sonuçları

Algoritma	Fn.	D = 30	D = 50	D = 100	D = 250	D = 500
HS	Ort.	1.0876e+00	5.7497e+00	1.7419e+02	1.4814e+03	5.0692e+03
	Std.	(3.32e-02)	(1.1e+00)	(2.18e+01)	(8.32e+01)	(1.72e+02)
IHS	Ort.	9.6566e-01	1.2141e+00	2.3783e+01	7.4665e+02	2.9832e+03
	Std.	(9.09e-02)	(6.7e-02)	(3.99e+00)	(4.96e+01)	(1.24e+02)
GHS	Ort.	1.1641e-01	7.1626e-01	4.4938e+01	8.147e+02	3.2085e+03
	Std.	(2.15e-01)	(3.95e-01)	(1.27e+01)	(9.24e+01)	(1.55e+02)
MHS	Ort.	5.3009e-01	1.0450e+00	1.9595e+00	1.5659e+02	1.3955e+03
	Std.	(2.14e-01)	(5.47e-02)	(1.66e-01)	(1.41e+01)	(7.00e+01)
NGHS	Ort.	6.9561e-02	4.1439e-02	5.1536e-02	8.7857e+00	3.7208e+02
	Std.	(5.17e-02)	(3.74e-02)	(2.45e-02)	(7.02e-01)	(1.96e+01)
PAHS-1	Ort.	1.143e+00	2.7225e+00	3.3198e+01	5.881e+02	2.4860e+03
	Std.	(5.47e-02)	(4.41e-01)	(4.48e+00)	(5.4e+01)	(1.17e+02)
PAHS-2	Ort.	1.3254e+00	4.5228e+00	4.7933e+01	5.8199e+02	2.3060e+03
	Std.	(1.34e-01)	(1.03e+00)	(6.52e+00)	(4.75e+01)	(1.07e+02)
PAHS-3	Ort.	1.1548e+00	2.6518e+00	3.5888e+01	5.9756e+02	2.5207e+03
	Std.	(6.69e-02)	(4.53e-01)	(4.3e+00)	(4.99e+01)	(1.32e+02)
PAHS-4	Ort.	1.3563e+00	4.6135e+00	4.6712e+01	5.8018e+02	2.3575e+03
	Std.	(1.06e-01)	(8.85e-01)	(7.07e+00)	(5.15e+01)	(1.42e+02)
LHS	Ort.	0	0	0	9.5929e-03	4.1412e+00
	Std.	(0)	(0)	(0)	(4.13e-02)	(1.68e+00)
BWMHS	Ort.	2.7537e-02	3.894e-02	4.5926e+00	5.5064e+01	1.7998e+02
	Std.	(4.09e-02)	(2.1e-01)	(2.47e+01)	(2.97e+02)	(9.69e+02)
SAHS	Ort.	7.8249e-01	1.0178e+00	1.2170e+00	1.8521e+01	6.6289e+02
	Std.	(1.93e-01)	(3.58e-02)	(5.68e-02)	(2.51e+00)	(3.76e+01)
TRI-HS	Ort.	0	0	0	0	0
	Std.	(0)	(0)	(0)	(0)	(0)

değerleri ilerleyen aşamalarda birbirine çok yakın gösterilmiştir. Ancak LHS ve TRI-HS varyantları Tablo 4.16 ile de görüleceği üzere diğer varyantlardan çok daha küçük değerler üretmeyi başarmıştır.



Şekil 4.7. Schwefel 2.22 fonksiyonu için HS varyantlarının yakınsama grafiği

Tablo 4.16. Schwefel 2.22 fonksiyonu HS varyant sonuçları

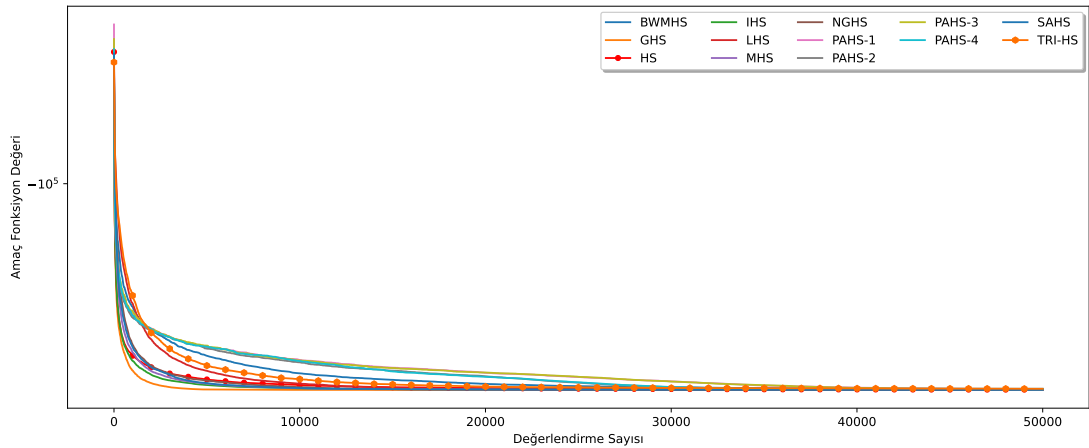
Algoritma	Fn.	D = 30	D = 50	D = 100	D = 250	D = 500
HS	Ort.	1.2129e-01	9.8340e+00	7.4309e+01	4.0410e+02	1.2746e+03
	Std.	(4.80e-02)	(1.25e+00)	(5.44e+00)	(1.39e+01)	(4.37e+01)
IHS	Ort.	3.1896e-01	2.2424e+00	2.9090e+01	2.5166e+02	8.1283e+02
	Std.	(7.26e-02)	(2.21e-01)	(3.09e+00)	(1.19e+01)	(1.91e+01)
GHS	Ort.	5.5291e-02	4.7795e-01	1.8467e+01	1.9185e+02	7.2491e+02
	Std.	(5.21e-02)	(4.40e-01)	(4.71e+00)	(1.69e+01)	(4.28e+01)
MHS	Ort.	2.6884e-03	2.4408e-02	1.5164e+00	1.1692e+02	4.6611e+02
	Std.	(5.63e-04)	(3.17e-03)	(3.45e-01)	(6.51e+00)	(1.8e+01)
NGHS	Ort.	6.9145e-09	5.0949e-05	1.1819e-01	3.1047e+01	3.1966e+02
	Std.	(1.25e-08)	(1.96e-05)	(1.82e-02)	(1.6e+00)	(9.28e+00)
PAHS-1	Ort.	9.3179e-01	5.9432e+00	3.6475e+01	2.3387e+02	7.2994e+02
	Std.	(2.07e-01)	(7.53e-01)	(3.12e+00)	(1.44e+01)	(2.76e+01)
PAHS-2	Ort.	9.6345e-02	3.3371e+00	3.2444e+01	2.1481e+02	6.7947e+02
	Std.	(8.76e-02)	(9.45e-01)	(3.22e+00)	(8.79e+00)	(2.46e+01)
PAHS-3	Ort.	1.4312e+00	6.0585e+00	3.6195e+01	2.2824e+02	7.265e+02
	Std.	(3.08e-01)	(7.28e-01)	(2.62e+00)	(1.07e+01)	(2.63e+01)
PAHS-4	Ort.	1.6867e+00	7.7196e+00	3.9634e+01	2.2602e+02	6.9905e+02
	Std.	(3.68e-01)	(1.08e+00)	(4.18e+00)	(9.92e+00)	(2.32e+01)
LHS	Ort.	6.4621e-177	1.3303e-124	1.2693e-61	7.9683e-08	6.5227e+00
	Std.	(3.48e-176)	(6.5e-124)	(3.69e-61)	(1.23e-07)	(1.93e+00)
BWMHS	Ort.	1.2608e-04	8.4398e-01	5.8606e+01	4.2915e+02	1.4327e+49
	Std.	(2.48e-05)	(2.69e-01)	(6.53e+00)	(1.52e+01)	(7.71e+49)
SAHS	Ort.	3.4406e-01	6.8649e-01	2.5272e+00	3.4164e+01	3.0874e+02
	Std.	(9.92e-02)	(1.46e-01)	(3.06e-01)	(2.27e+00)	(1.20e+01)
TRI-HS	Ort.	0	0	0	1.0712e-263	3.3161e-156
	Std.	(0)	(0)	(0)	(4.07e-263)	(9.27e-156)

Generalized Schwefel 2.26 fonksiyonu için HS varyantları ile farklı parametre sayılarında elde edilen sonuçların ortalama ve standart sapma değerleri Tablo 4.17 ile gösterilmiştir. Generalized Schwefel 2.26 fonksiyonu için HS varyantlarının 30 parametrelili koşmada elde ettiği yakınsama grafiği Şekil 4.8 ile gösterilmiştir. En iyi sonuçlar ise NGHS, LHS, MHS ve SAHS varyantları tarafından elde edilmiştir.

Step fonksiyonu için HS varyantları tarafından elde edilen sayısal sonuçlar Tablo 4.18 ile gösterilmiştir. 30 parametrelili problemde GHS, NGHS, BWMHS ve TRI-HS varyantları, 50 parametrelili problem için GHS, NGHS ve TRI-HS varyantları, 100 parametrelili problem için NGHS ve TRI-HS varyantları, 250 ve 500 parametre için ise sadece TRI-HS varyantı küresel optimum olan 0 çözümünü istisnasız tüm bağımsız koşmalarda elde edebilmiştir. LHS varyantı geri kalan varyantlardan çok daha küçük değerler elde edebilmeyi başarmış olsa da ortalama olarak optimum değere ulaşamamıştır. TRI-HS varyantı ise tüm koşmalarda optimum değere ulaşmayı

Tablo 4.17. Generalized Schwefel 2.26 fonksiyonu HS varyant sonuçları

Algoritma	Fn.	D = 30	D = 50	D = 100	D = 250	D = 500
HS	Ort.	-1.2546e+04	-2.013e+04	-3.4396e+04	-6.2235e+04	-9.0446e+04
	Std.	(6.29e+00)	(2.03e+02)	(7.68e+02)	(1.48e+03)	(1.88e+03)
IHS	Ort.	-1.2564e+04	-2.0885e+04	-3.8940e+04	-7.9388e+04	-1.2328e+05
	Std.	(2.14e+00)	(2.47e+01)	(3.24e+02)	(1.32e+03)	(2.44e+03)
GHS	Ort.	-1.2569e+04	-2.0943e+04	-4.0739e+04	-8.9699e+04	-1.4742e+05
	Std.	(9.33e-02)	(9.18e+00)	(3.42e+02)	(1.80e+03)	(3.17e+03)
MHS	Ort.	-1.2568e+04	-2.0933e+04	-4.1603e+04	-9.2002e+04	-1.5980e+05
	Std.	(1.11e+00)	(5.36e+00)	(7.00e+01)	(8.12e+02)	(2.47e+03)
NGHS	Ort.	-1.2569e+04	-2.0945e+04	-4.1746e+04	-4.5497e+04	-1.3715e+05
	Std.	(0)	(2.13e+01)	(1.32e+02)	(2.10e+03)	(1.65e+03)
PAHS-1	Ort.	-1.2524e+04	-2.0563e+04	-3.8280e+04	-8.0607e+04	-1.2956e+05
	Std.	(1.75e+01)	(8.26e+01)	(3.31e+02)	(1.13e+03)	(2.54e+03)
PAHS-2	Ort.	-1.2483e+04	-2.0368e+04	-3.7867e+04	-7.9727e+04	-1.3080e+05
	Std.	(2.80e+01)	(1.42e+02)	(4.62e+02)	(1.43e+03)	(2.12e+03)
PAHS-3	Ort.	-1.2524e+04	-2.0536e+04	-3.8344e+04	-7.9809e+04	-1.3058e+05
	Std.	(1.64e+01)	(7.86e+01)	(4.42e+02)	(1.28e+03)	(1.99e+03)
PAHS-4	Ort.	-1.2483e+04	-2.0329e+04	-3.7813e+04	-7.9401e+04	-1.3157e+05
	Std.	(2.74e+01)	(1.02e+02)	(4.47e+02)	(1.19e+03)	(2.37e+03)
LHS	Ort.	-1.2569e+04	-2.0940e+04	-4.1086e+04	-7.5555e+04	-1.0358e+05
	Std.	(1.92e-03)	(2.95e+01)	(2.37e+02)	(1.93e+03)	(2.94e+03)
BWMHS	Ort.	-1.2518e+04	-1.6857e+04	-2.1931e+04	-3.2946e+04	-4.6064e+04
	Std.	(2.4e+01)	(7.33e+02)	(1.19e+03)	(1.47e+03)	(1.99e+03)
SAHS	Ort.	-1.2564e+04	-2.0935e+04	-4.1784e+04	-9.7847e+04	-1.5958e+05
	Std.	(3.80e+00)	(5.59e+00)	(3.24e+01)	(7.e+02)	(1.92e+03)
TRI-HS	Ort.	-1.2455e+04	-2.0689e+04	-4.0474e+04	-9.2135e+04	-1.5419e+05
	Std.	(8.43e+01)	(9.49e+01)	(3.32e+02)	(1.03e+03)	(2.04e+03)



Şekil 4.8. Generalized Schwefel 2.26 fonksiyonu için HS varyantlarının yakınsama grafiği

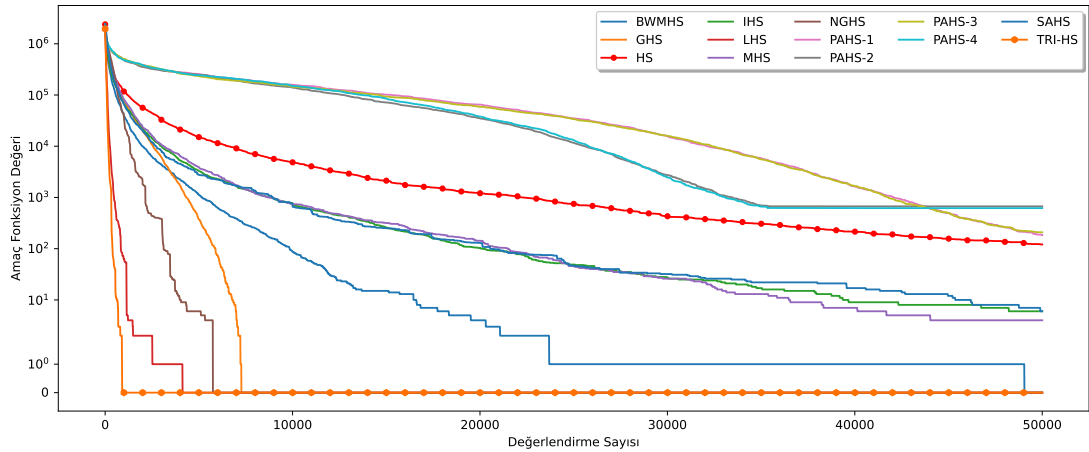
başarmıştır.

Step fonksiyonu için HS varyantlarının 30 parametrelili problemdeki yakınsama grafiği Şekil 4.9 ile gösterilmiştir. İlk olarak LHS ve TRI-HS varyantları optimum

Tablo 4.18. Step fonksiyonu HS varyant sonuçları

Algoritma	Fn.	D = 30	D = 50	D = 100	D = 250	D = 500
HS	Ort.	4.0333e+00	4.4273e+02	1.8337e+04	1.5866e+05	5.4824e+05
	Std.	(2.58e+00)	(8.43e+01)	(2.13e+03)	(9.27e+03)	(1.76e+04)
IHS	Ort.	2.e-01	1.0333e+01	2.0015e+03	8.1586e+04	3.2478e+05
	Std.	(4.e-01)	(5.14e+00)	(3.04e+02)	(5.22e+03)	(1.27e+04)
GHS	Ort.	0	0	5.2845e+03	9.0079e+04	3.4836e+05
	Std.	(0)	(0)	(1.72e+03)	(1.02e+04)	(2.52e+04)
MHS	Ort.	1.3333e-01	1.2667e+00	5.6333e+01	1.5468e+04	1.4699e+05
	Std.	(3.4e-01)	(1.48e+00)	(1.59e+01)	(1.17e+03)	(7.98e+03)
NGHS	Ort.	0	0	0	7.2967e+01	3.7533e+04
	Std.	(0)	(0)	(0)	(1.21e+01)	(1.87e+03)
PAHS-1	Ort.	6.1333e+00	1.203e+02	3.2424e+03	6.4038e+04	2.7059e+05
	Std.	(3.69e+00)	(3.25e+01)	(4.81e+02)	(4.45e+03)	(1.08e+04)
PAHS-2	Ort.	2.2367e+01	3.1227e+02	4.7913e+03	6.1391e+04	2.4738e+05
	Std.	(1.01e+01)	(1.11e+02)	(7.79e+02)	(4.37e+03)	(1.26e+04)
PAHS-3	Ort.	6.9667e+00	1.4523e+02	3.2972e+03	6.4621e+04	2.7193e+05
	Std.	(4.35e+00)	(5.28e+01)	(4.99e+02)	(5.1e+03)	(1.26e+04)
PAHS-4	Ort.	2.0633e+01	2.8863e+02	4.5691e+03	6.0198e+04	2.5252e+05
	Std.	(1.02e+01)	(5.64e+01)	(1.01e+03)	(4.61e+03)	(1.04e+04)
LHS	Ort.	7.2383e-43	1.0955e-76	6.1688e-43	1.6667e-01	3.533e+02
	Std.	(3.62e-43)	(3.65e-77)	(3.08e-43)	(5.22e-01)	(1.85e+02)
BWMHS	Ort.	0	2.1833e+01	4.955e+02	6.1622e+03	2.0652e+04
	Std.	(0)	(1.79e+01)	(2.67e+03)	(3.32e+04)	(1.11e+05)
SAHS	Ort.	2.e-01	1.1333e+00	1.1633e+01	1.5230e+03	7.0272e+04
	Std.	(4.76e-01)	(1.23e+00)	(6.36e+00)	(1.56e+02)	(4.45e+03)
TRI-HS	Ort.	0	0	0	0	0
	Std.	(0)	(0)	(0)	(0)	(0)

çözümüne çok hızlı bir şekilde yakınsarken, bu varyantları sırasıyla NGHS, GHS, BWMHS, MHS ve son olarak da SAHS takip etmiştir. PAHS varyantları ise orijinal HS algoritmasıyla birlikte daha az başarılı bir grafik sergilemiştir.



Şekil 4.9. Step fonksiyonu için HS varyantlarının yakınsama grafiği

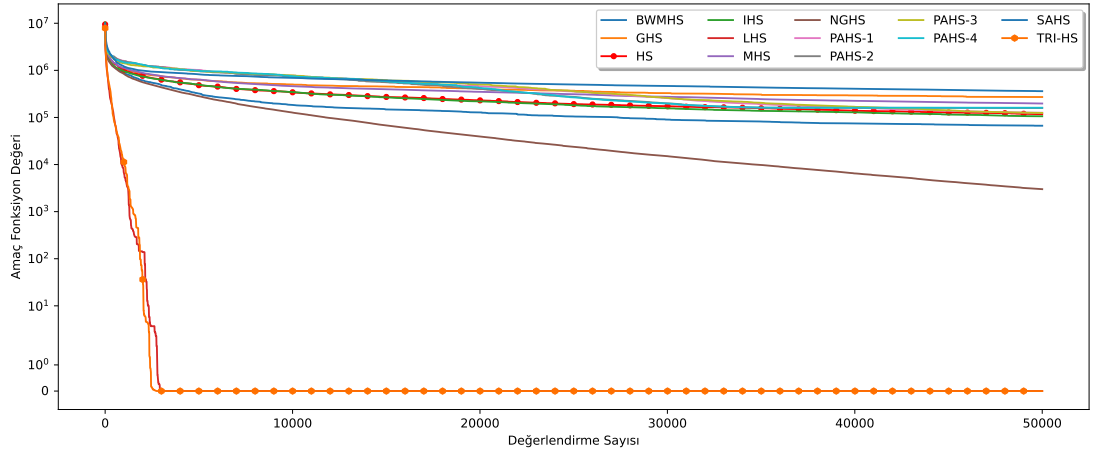
Rotated Hyper Ellipsoid fonksiyonu için HS varyantlarıyla yapılan koşmalardan elde edilen ortalama ve standart sapma değerleri Tablo 4.19 ile gösterilmiştir. LHS ve TRI-HS varyantları en kaliteli sonuçları elde eden algoritmalar olmuştur.

Tablo 4.19. Rotated Hyper Ellipsoid fonksiyonu HS varyant sonuçları

Algoritma	Fn.	D = 30	D = 50	D = 100	D = 250	D = 500
HS	Ort.	3.9050e+03	2.6058e+04	2.0158e+05	1.5701e+06	6.5537e+06
	Std.	(1.15e+03)	(6.09e+03)	(3.16e+04)	(1.64e+05)	(5.66e+05)
IHS	Ort.	3.5221e+03	1.7136e+04	1.179e+05	1.1095e+06	4.6779e+06
	Std.	(9.37e+02)	(3.21e+03)	(1.6e+04)	(1.19e+05)	(5.18e+05)
GHS	Ort.	9.0464e+03	6.2367e+04	3.4339e+05	2.2734e+06	9.0049e+06
	Std.	(8.88e+03)	(1.66e+04)	(4.90e+04)	(2.6e+05)	(9.67e+05)
MHS	Ort.	6.5859e+03	1.8149e+04	9.5366e+04	9.2793e+05	4.0838e+06
	Std.	(2.04e+03)	(3.59e+03)	(1.79e+04)	(1.37e+05)	(4.47e+05)
NGHS	Ort.	9.9998e+01	1.9829e+03	2.9344e+04	3.2865e+05	1.582e+06
	Std.	(4.90e+01)	(5.15e+02)	(4.91e+03)	(3.42e+04)	(1.40e+05)
PAHS-1	Ort.	4.1889e+03	2.4668e+04	1.6001e+05	1.3852e+06	6.1921e+06
	Std.	(1.44e+03)	(5.84e+03)	(4.97e+04)	(3.35e+05)	(1.02e+06)
PAHS-2	Ort.	5.2827e+03	2.9887e+04	1.7312e+05	1.1049e+06	4.0879e+06
	Std.	(1.38e+03)	(5.83e+03)	(3.46e+04)	(2.11e+05)	(6.18e+05)
PAHS-3	Ort.	4.1608e+03	2.4009e+04	1.5047e+05	1.2607e+06	6.4713e+06
	Std.	(1.14e+03)	(3.58e+03)	(3.01e+04)	(3.24e+05)	(1.28e+06)
PAHS-4	Ort.	5.3562e+03	2.9820e+04	1.7392e+05	1.1215e+06	4.3962e+06
	Std.	(1.73e+03)	(7.11e+03)	(4.14e+04)	(2.03e+05)	(6.24e+05)
LHS	Ort.	3.8838e-57	9.7665e-10	1.658e+03	2.1106e+05	1.1395e+06
	Std.	(2.08e-56)	(5.26e-09)	(6.57e+03)	(1.62e+05)	(4.62e+05)
BWMHS	Ort.	1.4246e+03	1.0196e+03	7.8021e+03	6.7711e+04	2.0105e+05
	Std.	(1.66e+03)	(5.49e+03)	(4.20e+04)	(3.65e+05)	(1.08e+06)
SAHS	Ort.	1.2041e+04	3.3216e+04	1.2452e+05	7.3127e+05	3.6405e+06
	Std.	(4.8e+03)	(5.15e+03)	(1.45e+04)	(7.74e+04)	(4.16e+05)
TRI-HS	Ort.	0	7.7125e-248	2.8318e-141	1.5313e-44	1.6211e-08
	Std.	(0)	(4.05e-247)	(1.52e-140)	(7.95e-44)	(8.73e-08)

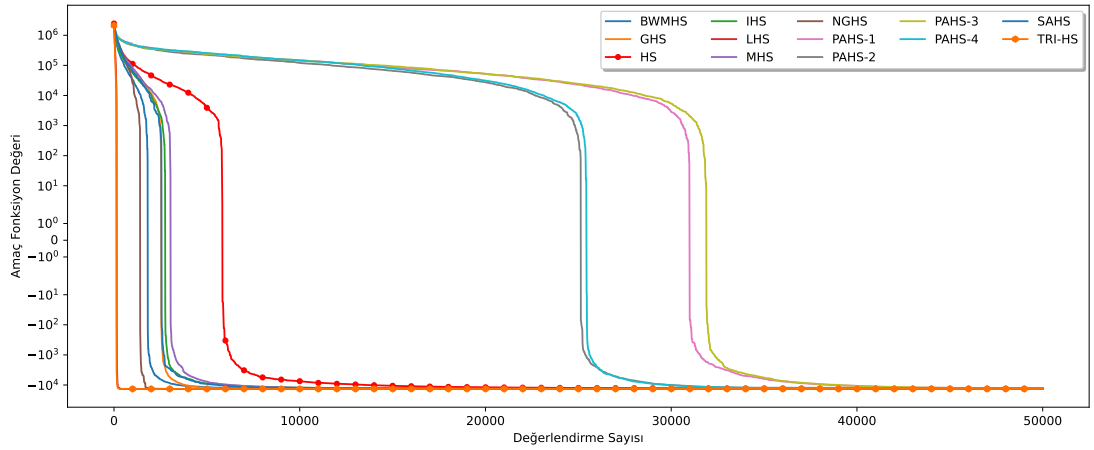
Rotated Hyper Ellipsoid fonksiyonu için varyantların koşma sonuçlarıyla elde edilen yakınsama grafiği Şekil 4.10 ile gösterilmiştir. LHS ve TRI-HS varyantları çok hızlı şekilde optimum sonuca yakınsarken, NGHS ve GHS varyantları da ilerleyen aşamalarda diğer varyantlardan ayrılarak nispeten daha optimum sonuçlar üretmeye başlamıştır. LHS ve TRI-HS varyantları hem yakınsama hızı hem de sonuçta diğer varyantlarla olan farktan dolayı diğer varyantlardan gözle görülür düzeyde üstün bir başarı göstermiştir denilebilir.

Shifted Sphere fonksiyonu için HS varyantlarıyla elde edilen sayısal sonuçlar Tablo 4.20 ile gösterilmiştir. Sonuçlar incelendiğinde NGHS, LHS ve TRI-HS



Şekil 4.10. Rotated Hyper Ellipsoid fonksiyonu için HS varyantlarının yakınsama grafiği

varyantlarının farklı parametrelerde en iyi sonuçları elde eden varyantlar olduğu görülmektedir. Shifted Sphere fonksiyonu 30 parametrelili problemi için HS varyantları tarafından elde edilen sonuçların yakınsama grafiği Şekil 4.11 ile gösterilmiştir. Tüm varyantların yüksek sonuçlardan başlayarak optimum değere yakınsamayı başardığı görülmektedir. PAHS varyantları haricindeki diğer varyantlar 10.000 iterasyon civarında optimum değer olan -450 seviyesine yakınsamayı başarmıştır. En hızlı yakınsama gösteren varyantlar ise LHS ve TRI-HS olmuştur.



Şekil 4.11. Shifted Sphere fonksiyonu için HS varyantlarının yakınsama grafiği

4.3.2. EEG Optimizasyonu Deneysel Çıktıları

EEG sinyallerinden artifaktların temizlenmesi için mevcut HS varyantları ve bu tez çalışmaları kapsamında geliştirilen TRI-HS varyantının performanslarını kıyaslamak amacıyla her bir varyant ile belirlenen parametreler kullanılarak 30

Tablo 4.20. Shifted Sphere fonksiyonu HS varyant sonuçları

Algoritma	Fn.	D = 30	D = 50	D = 100	D = 250	D = 500
HS	Ort.	-4.431e+02	1.2906e+02	1.8588e+04	1.6366e+05	5.6879e+05
	Std.	(2.96e+00)	(1.36e+02)	(2.43e+03)	(1.15e+04)	(1.55e+04)
IHS	Ort.	-4.4821e+02	-4.2833e+02	1.9891e+03	8.251e+04	3.3714e+05
	Std.	(8.14e-01)	(5.39e+00)	(3.95e+02)	(5.78e+03)	(1.4e+04)
GHS	Ort.	-4.4998e+02	-4.4811e+02	4.6773e+03	9.1659e+04	3.6489e+05
	Std.	(2.64e-02)	(2.72e+00)	(1.28e+03)	(9.17e+03)	(1.52e+04)
MHS	Ort.	-4.5e+02	-4.4981e+02	-3.5487e+02	1.6652e+04	1.5327e+05
	Std.	(3.e-06)	(2.17e-01)	(1.81e+01)	(1.32e+03)	(9.01e+03)
NGHS	Ort.	-4.5e+02	-4.5e+02	-4.4995e+02	4.152e+02	4.1076e+04
	Std.	(0)	(0)	(1.26e-02)	(8.58e+01)	(2.59e+03)
PAHS-1	Ort.	-4.3310e+02	-2.7207e+02	3.3539e+03	6.5811e+04	2.7753e+05
	Std.	(5.96e+00)	(4.75e+01)	(6.67e+02)	(5.39e+03)	(1.38e+04)
PAHS-2	Ort.	-4.1994e+02	-9.6472e+01	4.5678e+03	6.3516e+04	2.598e+05
	Std.	(1.51e+01)	(9.75e+01)	(6.33e+02)	(5.41e+03)	(1.68e+04)
PAHS-3	Ort.	-4.3272e+02	-2.5711e+02	3.2926e+03	6.6626e+04	2.7447e+05
	Std.	(7.3e+00)	(4.4e+01)	(5.16e+02)	(6.37e+03)	(1.10e+04)
PAHS-4	Ort.	-4.1871e+02	-7.0668e+01	4.6051e+03	6.4451e+04	2.5952e+05
	Std.	(1.12e+01)	(1.00e+02)	(7.85e+02)	(6.4e+03)	(1.43e+04)
LHS	Ort.	-4.5e+02	-4.5e+02	-4.5e+02	-4.5e+02	-9.8295e+01
	Std.	(0)	(0)	(0)	(0)	(1.53e+02)
BWMHS	Ort.	-6.e+01	-1.4789e+01	5.6007e+02	6.4314e+03	2.1610e+04
	Std.	(1.53e+02)	(7.96e+01)	(3.02e+03)	(3.46e+04)	(1.16e+05)
SAHS	Ort.	-4.485e+02	-4.467e+02	-4.2482e+02	1.3593e+03	7.3255e+04
	Std.	(8.25e-01)	(1.52e+00)	(7.67e+00)	(2.00e+02)	(5.14e+03)
TRI-HS	Ort.	-4.5e+02	-4.5e+02	-4.5e+02	-4.5e+02	-4.5e+02
	Std.	(0)	(0)	(0)	(0)	(0)

bağımsız koşma gerçekleştirilmiştir. Gürültüsüz ve gürültü eklenmiş 4, 12 ve 19 kanallı EEG sinyalleri üzerinde HS varyantları önerilen parametreler kullanılarak harmoni hafıza boyutu için 5 ve 10 değerleri verilerek koşmalar gerçekleştirilmiştir.

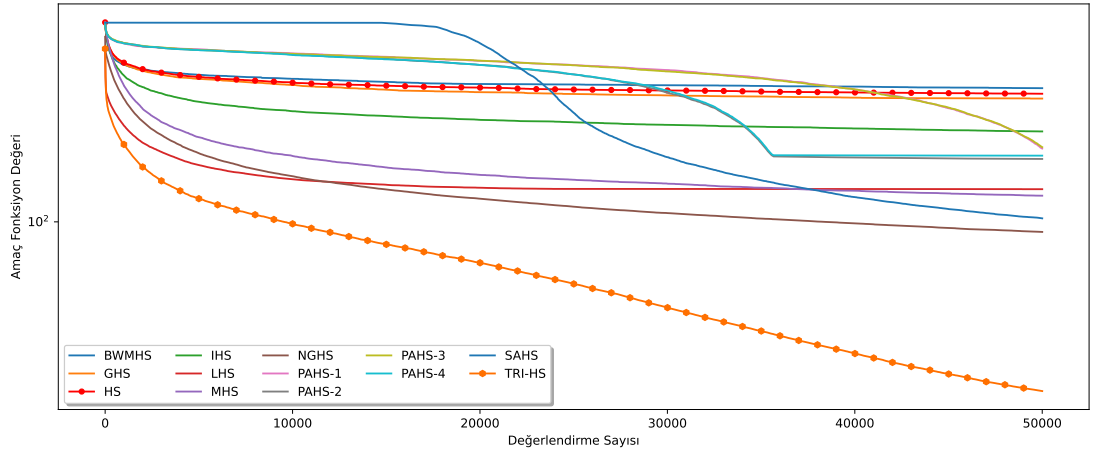
Gürültü eklenmemiş EEG sinyalleri üzerinde HS varyantları ile 4, 12 ve 19 kanallı optimizasyon işlemleri ile elde edilen ortalama sonuçlar ve standart sapma değerleri Tablo 4.21 ile gösterilmiştir. 4 kanallı problemde BWMHS hariç diğer varyantlar, 12 ve 19 kanal sayısına sahip problemde ise tüm varyantlar orijinal HS algoritmasından daha başarılı sonuç elde etmiştir. TRI-HS algoritması üç farklı kanal sayısına sahip problem için de en başarılı sonuçları üretmiştir. 4 kanallı yani 1024 parametrelili problem için TRI-HS algoritması orijinal HS algoritmasından **17.3** kat, 12 kanallı yani 3072 parametrelili problem için **6.4** kat, 19 kanallı yani 4864 parametrelili problem için ise **5.1** kat daha hassas sonuç üretmiştir.

Tablo 4.21. Gürültüsüz EEG Problemi HS varyant sonuçları (HMS: 5)

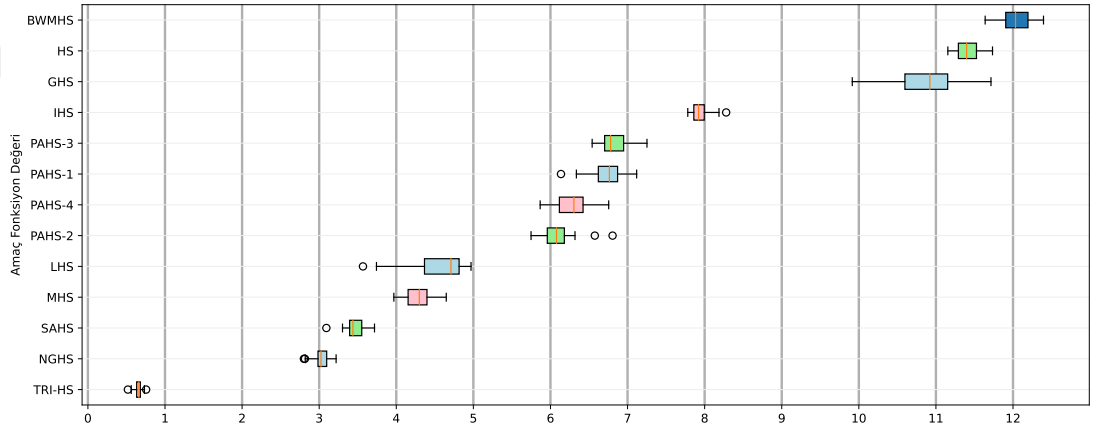
Algoritma	Fn.	D = 4	D = 12	D = 19
HS	Ort.	1.1407e+01	1.5669e+01	1.7076e+01
	Std.	(1.49e-01)	(1.58e-01)	(1.09e-01)
IHS	Ort.	7.9435e+00	1.2945e+01	1.4789e+01
	Std.	(1.15e-01)	(1.28e-01)	(1.13e-01)
GHS	Ort.	1.0892e+01	1.5184e+01	1.6614e+01
	Std.	(4.12e-01)	(3.93e-01)	(4.69e-01)
MHS	Ort.	4.2853e+00	8.7648e+00	1.092e+01
	Std.	(1.52e-01)	(1.27e-01)	(1.00e-01)
NGHS	Ort.	3.0254e+00	8.1261e+00	1.0099e+01
	Std.	(1.01e-01)	(1.01e-01)	(7.25e-02)
PAHS-1	Ort.	6.7331e+00	1.1884e+01	1.3846e+01
	Std.	(2.10e-01)	(1.94e-01)	(1.49e-01)
PAHS-2	Ort.	6.0982e+00	1.1036e+01	1.3055e+01
	Std.	(2.18e-01)	(1.79e-01)	(1.46e-01)
PAHS-3	Ort.	6.813e+00	1.1861e+01	1.3832e+01
	Std.	(1.85e-01)	(1.6e-01)	(1.52e-01)
PAHS-4	Ort.	6.2951e+00	1.1136e+01	1.3145e+01
	Std.	(2.27e-01)	(1.70e-01)	(1.11e-01)
LHS	Ort.	4.5613e+00	8.1599e+00	9.2683e+00
	Std.	(3.47e-01)	(1.39e+00)	(1.90e+00)
BWMHS	Ort.	1.2035e+01	1.5169e+01	1.6160e+01
	Std.	(1.96e-01)	(1.73e-01)	(1.14e-01)
SAHS	Ort.	3.4472e+00	8.9477e+00	1.1532e+01
	Std.	(1.24e-01)	(1.7e-01)	(1.60e-01)
TRI-HS	Ort.	6.5624e-01	2.4312e+00	3.3169e+00
	Std.	(4.57e-02)	(1.17e-01)	(1.8e-01)

Tablo 4.21 ile ortalama ve standart sapması gösterilen 4 kanallı koşma sonuçları için HS varyantlarının elde ettikleri yakınsama grafiği Şekil 4.12 ile, sonuçların dağılımını görebilmeyi sağlayan kutu grafiği Şekil 4.13 ile gösterilmiştir. Yakınsama grafiğinden kolaylıkla görülebileceği üzere TRI-HS varyantı koşmanın ilk iterasyonlarından itibaren diğer varyantlardan ayrılarak çok daha küçük sonuçlar üretmeye başlamıştır. Kutu grafiği incelendiğinde ise TRI-HS varyantının hem ortalama değer hem de değer aralığı bakımından diğer tüm varyantlardan başarılı durumda olduğu, BWMHS algoritmasının bu koşmalardaki en az başarılı sonuçları ürettiği görülmektedir.

Gürültü eklemesi yapılmamış EEG sinyalleri üzerindeki 12 kanallı koşma sonuçları için HS varyantlarının ortalama ve standart sapma değerleri Tablo 4.21 ile gösterilmiştir. Varyantların bu deneyde oluşturdukları yakınsama grafiği Şekil 4.14 ile, sonuçların dağılımını görebilmeyi sağlayan kutu grafiği Şekil 4.15 ile



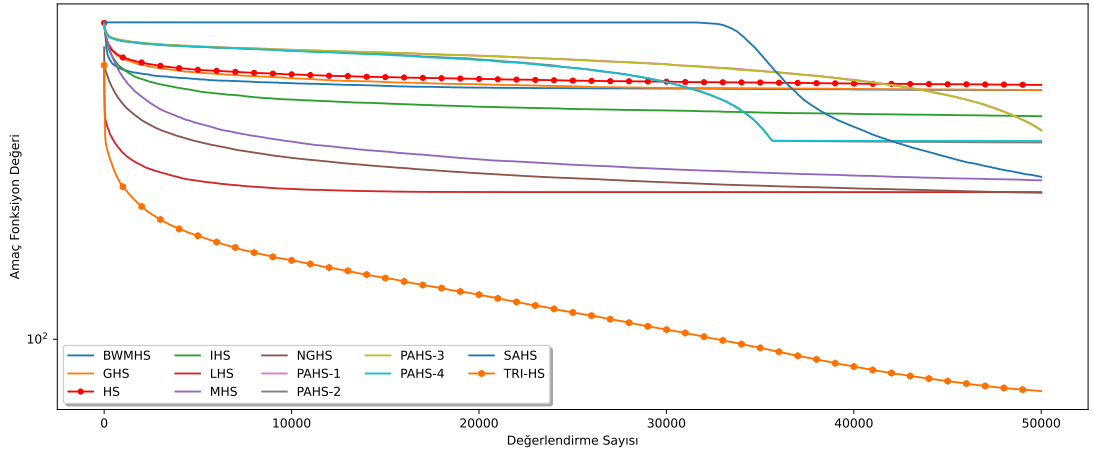
Şekil 4.12. Gürültüsüz 4 Kanallı EEG problemi için yakınsama grafiği (HMS: 5)



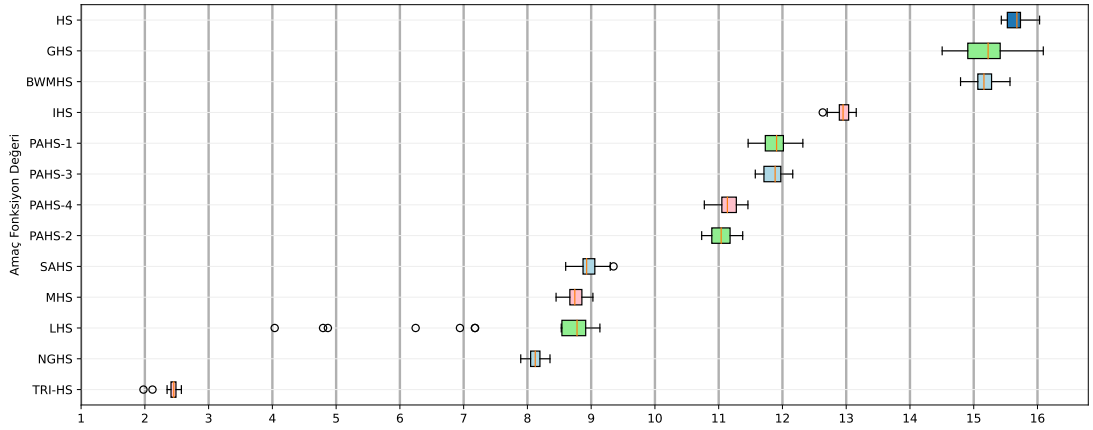
Şekil 4.13. Gürültüsüz 4 Kanallı EEG problemi için kutu grafiği (HMS: 5)

gösterilmiştir. Yakınsama grafiğinde TRI-HS algoritmasının yine ilk iterasyonlardan itibaren diğer varyantlardan çok daha küçük değerler üretmeye başladığı, benzer bir ivmeyle başlayan LHS algoritmasının ise ilerleyen aşamalarda yakınsama hızının yavaşlaması üzerine diğer varyantlarla arasının kapanmaya başladığı, NGHS algoritmasının ise son iterasyonlarda LHS'den daha iyi sonuçlar üretebildiği görülebilmektedir. Kutu grafikten anlaşılabileceği üzere tüm varyantlar orijinal HS algoritmasından ortalama olarak daha iyi sonuçlar üretmiştir. TRI-HS algoritması hem ortalama hem de değer aralığı bakımından diğer tüm varyantlardan daha iyi bir konumda yer alırken LHS algoritmasının bazı sonuçlarının diğer koşmalardan ayrılarak NGHS'den daha iyi sonuçlar elde etmiş olsa da ortalama olarak NGHS'den daha az başarılı olduğu görülmektedir.

Gürültüsüz EEG sinyalleri üzerindeki 19 kanallı problem için yapılan koşmalardaki ortalama ve standart sapma değerleri 4.21 ile gösterilmiştir. HS



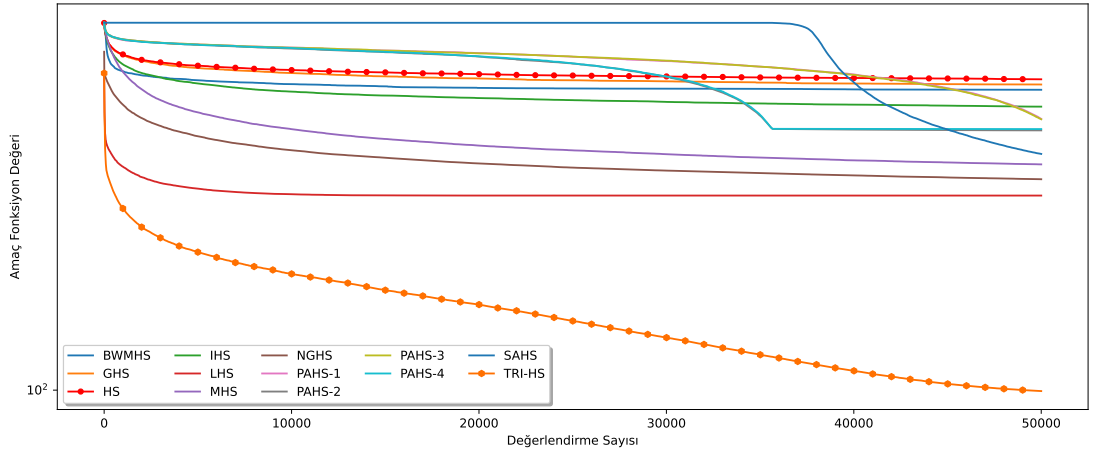
Şekil 4.14. Gürültüsüz 12 Kanallı EEG problemi için yakınsama grafiği (HMS: 5)



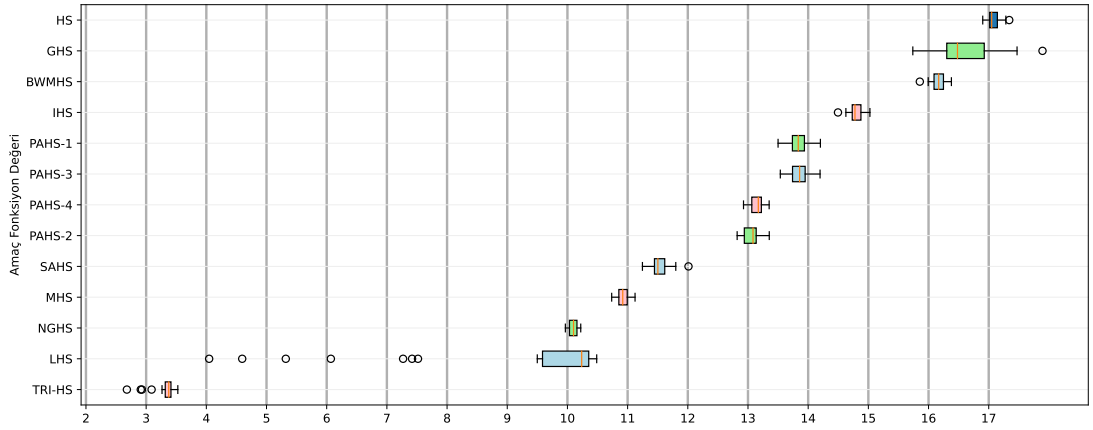
Şekil 4.15. Gürültüsüz 12 Kanallı EEG problemi için kutu grafiği (HMS: 5)

varyantlarının bu deneylerde koşmaların başından sonuna kadar oluşturdukları yakınsama grafiği Şekil 4.16 ile, koşmalar sonucunda elde ettikleri değerlere ait kutu grafiği Şekil 4.17 ile gösterilmiştir. Yakınsama grafiği 12 kanallı koşma ile çok büyük oranda benzerlik gösterirken LHS varyantının yine benzer şekilde yakınsama hızı düşmüş olsa da bu deneylerde NGHS varyantından daha iyi sonuçlar ile koşmaları tamamlamıştır. Kutu grafiğinde görülebileceği üzere TRI-HS varyantı hem ortalama, hem de değer aralığı olarak en iyi sonucu vermiş, LHS varyantının bazı sonuçları da TRI-HS'ye yaklaşmasına rağmen ortalama olarak yine de aradaki fark kapanmamıştır.

Harmoni hafıza boyutu, HS algoritmasının temel parametrelerinden bir tanesi olup bazı problemlerde ve varyantlarda sonuca yüksek oranda etki edebilmektedir. Bu nedenle yukarıda 5 olarak verilen HMS değeri iki katına çıkartılarak 10 adet harmoniyi bünyesinde barındıran hafıza ile de EEG sinyallerinin artifaktlardan arındırılması problemi üzerinde deneyler gerçekleştirilmiştir. Gürültü eklenmemiş 4,



Şekil 4.16. Gürültüsüz 19 Kanallı EEG problemi için yakınsama grafiği (HMS: 5)



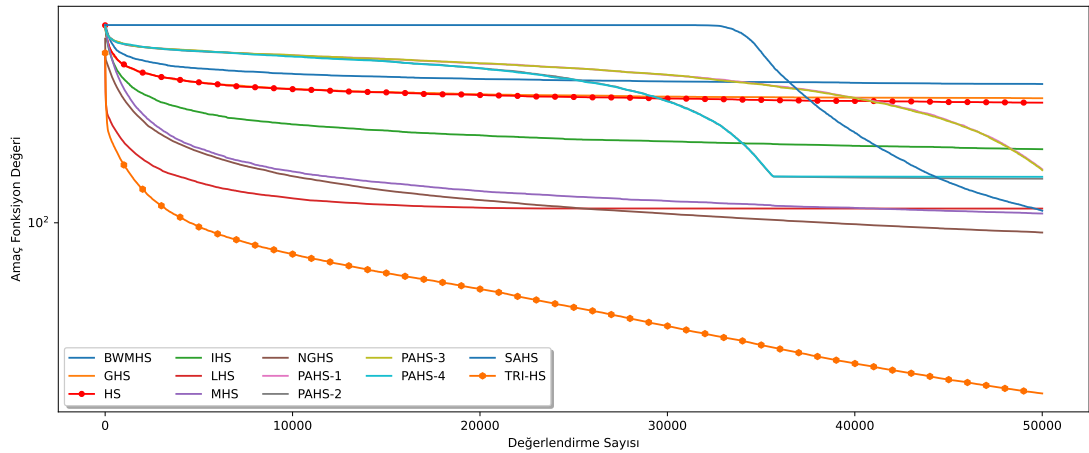
Şekil 4.17. Gürültüsüz 19 Kanallı EEG problemi için kutu grafiği (HMS: 5)

12 ve 19 kanallı EEG sinyalleri üzerinde HS varyantlarının $HMS = 10$ değeri ile elde ettikleri ortalama değerler ve bu değerlerin standart sapmaları Tablo 4.22 ile gösterilmiştir. Üç farklı kanal sayısının hepsinde TRI-HS algoritması en başarılı sonucu elde eden algoritma olmuştur. Elde edilen sonuçlara göre TRI-HS algoritması, orijinal HS varyantından 4 kanallı problem için **16.3** kat, 12 kanallı problem için **8.1** kat, 19 kanallı problem için ise **6.8** kat daha hassas sonuçlar elde etmiştir.

Gürültüsüz 4 kanallı EEG sinyalleri için $HMS = 10$ değeri ile yapılan koşullarda HS varyantlarının yakınsama grafikleri Şekil 4.18 ile, kutu grafiği Şekil 4.19 ile gösterilmiştir. TRI-HS varyantı ilk iterasyonlardan itibaren diğer varyantlardan daha iyi bir yakınsama hızı sergilemiştir. Kutu grafiğinden görülebildiği üzere BWMHS ve GHS haricindeki varyantlar orijinal HS algoritmasından daha iyi sonuçlar üretmiş, TRI-HS varyantı hem ortalama değer hem de değer aralığı olarak tüm varyantlardan daha iyi bir sonuç elde etmiştir.

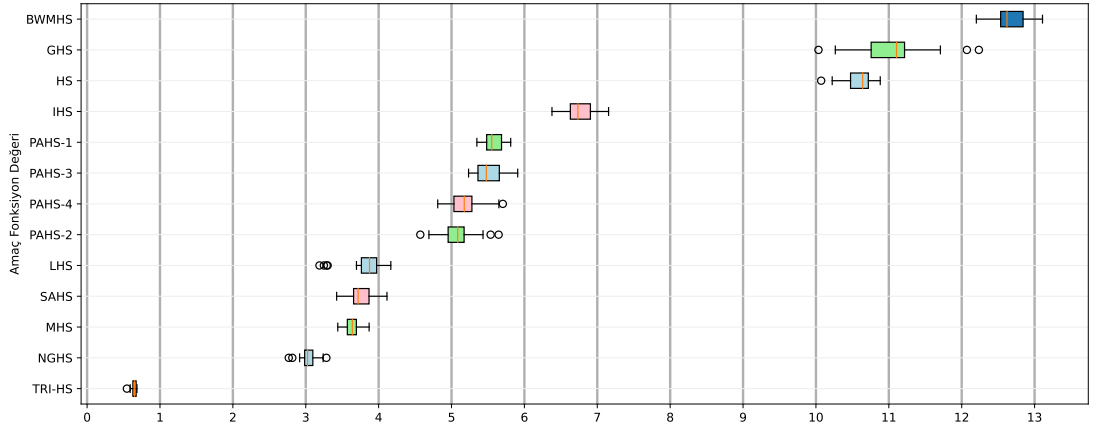
Tablo 4.22. Gürültüsüz EEG problemi HS varyant sonuçları (HMS: 10)

Algoritma	Fn.	D = 4	D = 12	D = 19
HS	Ort.	1.0578e+01	1.5049e+01	1.6551e+01
	Std.	(2.03e-01)	(1.05e-01)	(7.87e-02)
IHS	Ort.	6.7626e+00	1.1776e+01	1.3681e+01
	Std.	(1.68e-01)	(1.33e-01)	(9.76e-02)
GHS	Ort.	1.1039e+01	1.5157e+01	1.6573e+01
	Std.	(4.74e-01)	(4.12e-01)	(3.09e-01)
MHS	Ort.	3.6439e+00	7.5972e+00	9.6371e+00
	Std.	(1.08e-01)	(1.04e-01)	(7.72e-02)
NGHS	Ort.	3.0348e+00	8.1049e+00	1.0077e+01
	Std.	(1.12e-01)	(9.83e-02)	(1.05e-01)
PAHS-1	Ort.	5.5726e+00	1.0486e+01	1.2603e+01
	Std.	(1.45e-01)	(1.65e-01)	(1.22e-01)
PAHS-2	Ort.	5.0923e+00	9.8329e+00	1.1912e+01
	Std.	(2.24e-01)	(1.41e-01)	(1.2e-01)
PAHS-3	Ort.	5.5211e+00	1.0532e+01	1.2567e+01
	Std.	(2.05e-01)	(1.54e-01)	(1.43e-01)
PAHS-4	Ort.	5.1865e+00	9.8792e+00	1.1980e+01
	Std.	(2.1e-01)	(1.61e-01)	(1.8e-01)
LHS	Ort.	3.8212e+00	6.7704e+00	8.2694e+00
	Std.	(2.48e-01)	(1.05e+00)	(9.31e-01)
BWMHS	Ort.	1.2655e+01	1.6363e+01	1.7469e+01
	Std.	(2.22e-01)	(1.28e-01)	(1.05e-01)
SAHS	Ort.	3.742e+00	9.3201e+00	1.2091e+01
	Std.	(1.50e-01)	(1.61e-01)	(1.4e-01)
TRI-HS	Ort.	6.4634e-01	1.8439e+00	2.4112e+00
	Std.	(3.10e-02)	(8.94e-02)	(5.48e-02)



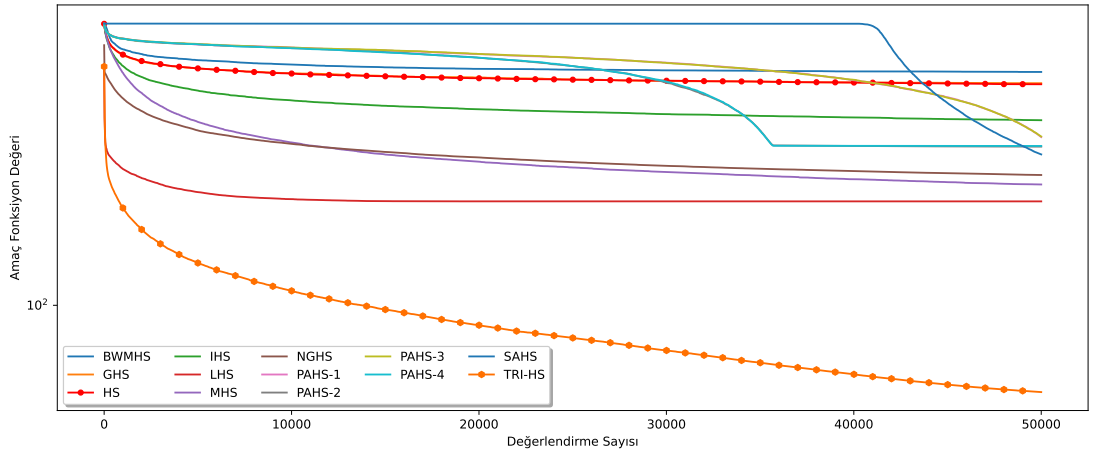
Şekil 4.18. Gürültüsüz 4 kanallı EEG problemi için yakınsama grafiği (HMS: 10)

Gürültüsüz 12 kanallı EEG sinyalleri için $HMS = 10$ değeri ile yapılan koşmalarda elde edilen sonuçların yakınsama grafiği Şekil 4.20 ile, kutu grafiği 4.21 ile gösterilmiştir. $HMS = 5$ ile benzer grafikler ve sonuçlar elde edilirken LHS'nin

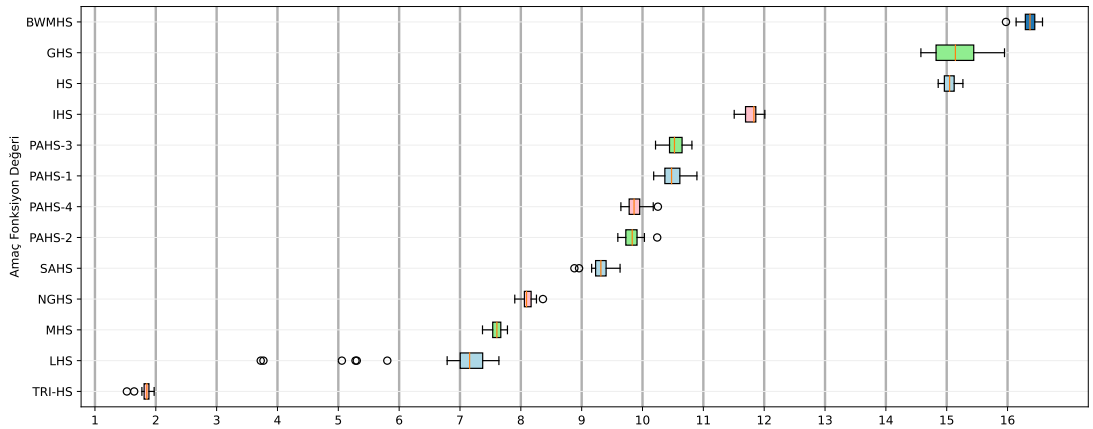


Şekil 4.19. Gürültüsüz 4 kanallı EEG problemi için kutu grafiği (HMS: 10)

NGHS'den daha iyi sonuçlar elde ettiği dikkat çekmektedir. TRI-HS algoritması diğer deneylerde olduğu gibi bu deneyde de en iyi sonuçları vermiştir.

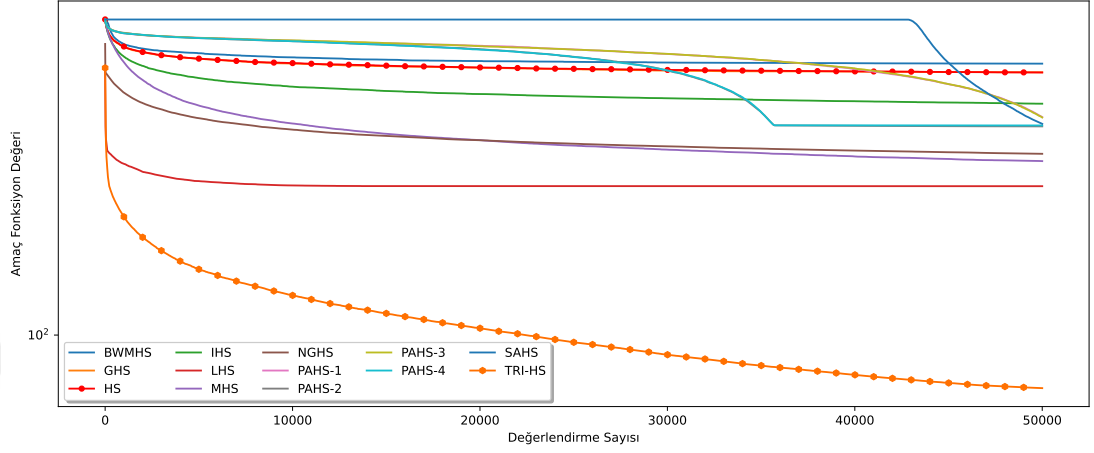


Şekil 4.20. Gürültüsüz 12 kanallı EEG problemi için yakınsama grafiği (HMS: 10)

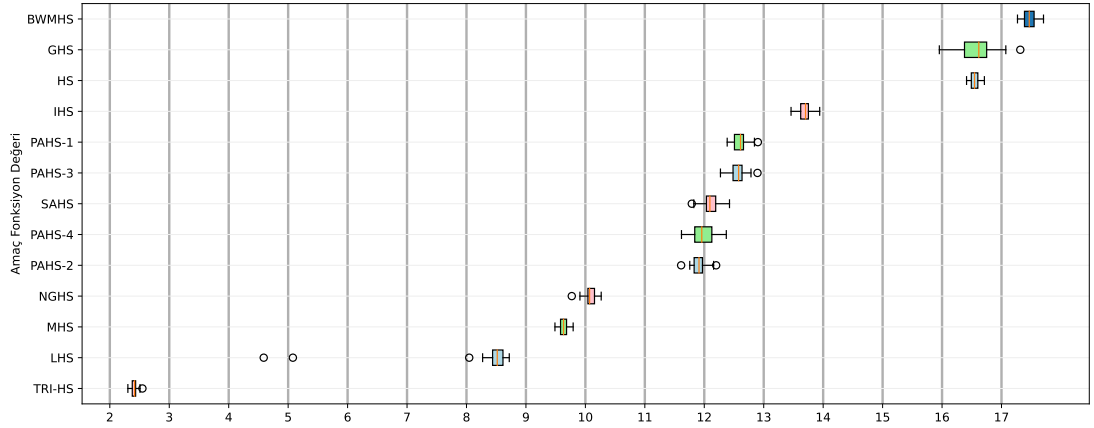


Şekil 4.21. Gürültüsüz 12 kanallı EEG problemi için kutu grafiği (HMS: 10)

Gürültüsüz 19 kanallı EEG sinyalleri için $HMS = 10$ değeri ile elde edilen yakınsama grafiği Şekil 4.22 ile, kutu grafiği 4.23 ile gösterilmiştir. $HMS = 5$ ile benzer grafikler ve sonuçlar elde edilmiş olup BWMHS ve GHS orijinal HS algoritmasından daha az başarılı sonuçlar elde etmiş, TRI-HS algoritması ise en iyi sonuçları vermiştir.



Şekil 4.22. Gürültüsüz 19 kanallı EEG problemi için yakınsama grafiği (HMS: 10)



Şekil 4.23. Gürültüsüz 19 kanallı EEG problemi için kutu grafiği (HMS: 10)

Gürültü eklenmiş EEG sinyalleri üzerinde HS varyantları ile $HMS = 5$ parametresini kullanarak 4, 12 ve 19 kanallı optimizasyon işlemleri ile elde edilen ortalama sonuçlar ve standart sapma değerleri Tablo 4.23 ile gösterilmiştir. TRI-HS algoritması üç farklı kanal sayısına sahip problem için de en başarılı sonuçları üretmiştir. 4 kanallı problem için TRI-HS algoritması orijinal HS algoritmasından **17.4** kat, 12 kanallı problem için **6.4** kat, 19 kanallı problem için ise **5.1** kat daha hassas sonuç üretmiştir.

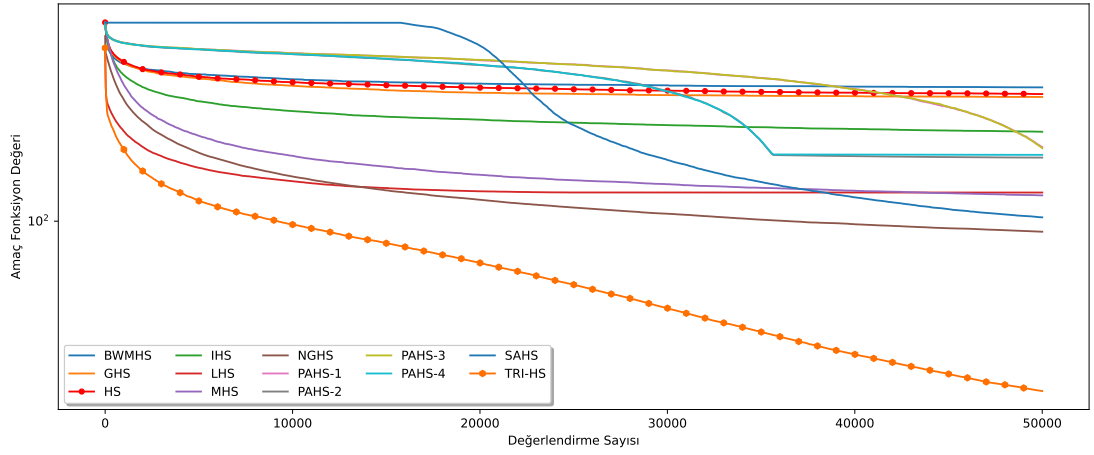
Tablo 4.23. Gürültü eklenmiş EEG Problemi sonuçları (HMS: 5)

Algoritma	Fn.	D = 4	D = 12	D = 19
HS	Ort.	1.1327e+01	1.567e+01	1.7102e+01
	Std.	(1.92e-01)	(1.18e-01)	(1.39e-01)
IHS	Ort.	7.8784e+00	1.297e+01	1.4793e+01
	Std.	(1.84e-01)	(1.48e-01)	(9.85e-02)
GHS	Ort.	1.1014e+01	1.5096e+01	1.645e+01
	Std.	(4.78e-01)	(3.89e-01)	(3.59e-01)
MHS	Ort.	4.2730e+00	8.7996e+00	1.0922e+01
	Std.	(1.43e-01)	(1.06e-01)	(9.73e-02)
NGHS	Ort.	3.0113e+00	8.1164e+00	1.0075e+01
	Std.	(1.06e-01)	(9.45e-02)	(9.78e-02)
PAHS-1	Ort.	6.7963e+00	1.187e+01	1.3830e+01
	Std.	(1.94e-01)	(1.51e-01)	(1.14e-01)
PAHS-2	Ort.	6.1455e+00	1.1036e+01	1.3062e+01
	Std.	(1.92e-01)	(1.72e-01)	(1.33e-01)
PAHS-3	Ort.	6.7166e+00	1.187e+01	1.3783e+01
	Std.	(1.64e-01)	(1.78e-01)	(1.54e-01)
PAHS-4	Ort.	6.3102e+00	1.1222e+01	1.3186e+01
	Std.	(2.14e-01)	(2.16e-01)	(1.73e-01)
LHS	Ort.	4.3901e+00	8.1296e+00	8.9082e+00
	Std.	(4.96e-01)	(1.54e+00)	(2.16e+00)
BWMHS	Ort.	1.2076e+01	1.5176e+01	1.6127e+01
	Std.	(2.44e-01)	(1.72e-01)	(1.01e-01)
SAHS	Ort.	3.4588e+00	8.9193e+00	1.1655e+01
	Std.	(1.42e-01)	(1.95e-01)	(1.56e-01)
TRI-HS	Ort.	6.4993e-01	2.4123e+00	3.3333e+00
	Std.	(3.17e-02)	(1.56e-01)	(1.95e-01)

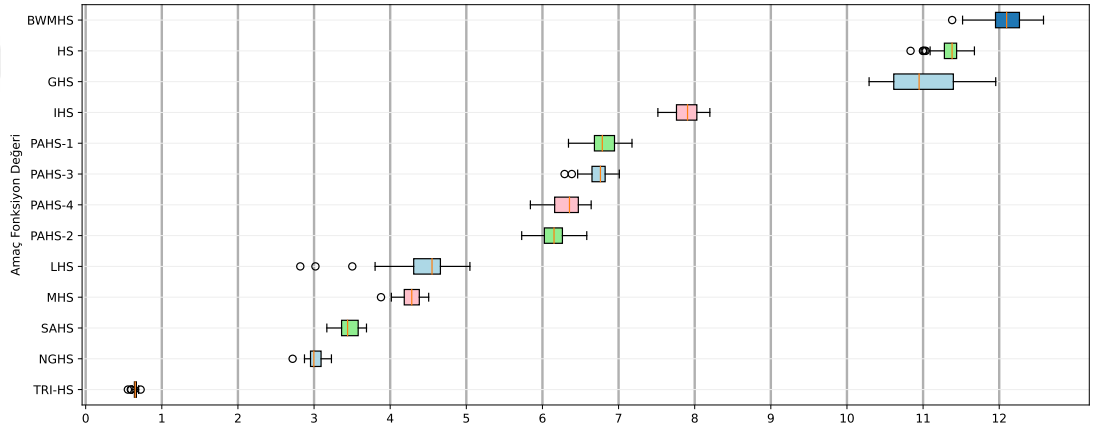
Gürültü eklenmiş 4 kanallı EEG sinyalleri üzerinde $HMS = 5$ parametresi ile elde edilen yakınsama grafiği Şekil 4.24 ile, kutu grafiği Şekil 4.25 ile gösterilmiştir. TRI-HS algoritmasının en başarılı sonuçları verdiği görülürken ilk iterasyonlardan itibaren diğer varyantlarla olan yakınsama hız farklılığı da dikkat çekmektedir. BWMHS algoritması ise orijinal HS algoritmasından daha başarısız sonuçlar üretmiştir.

Gürültü eklenmemiş 12 kanallı EEG sinyallerinin $HMS = 5$ parametresi ile optimize edilmeye çalışıldığı koşmaların sonuçlarının yakınsama grafiği Şekil 4.26 ile, kutu grafiği Şekil 4.27 ile gösterilmiştir. Gürültüsüz koşma ile çok benzer yakınsama ve kutu grafiği oluşmuş olup TRI-HS varyantı en başarılı sonucu vermiştir.

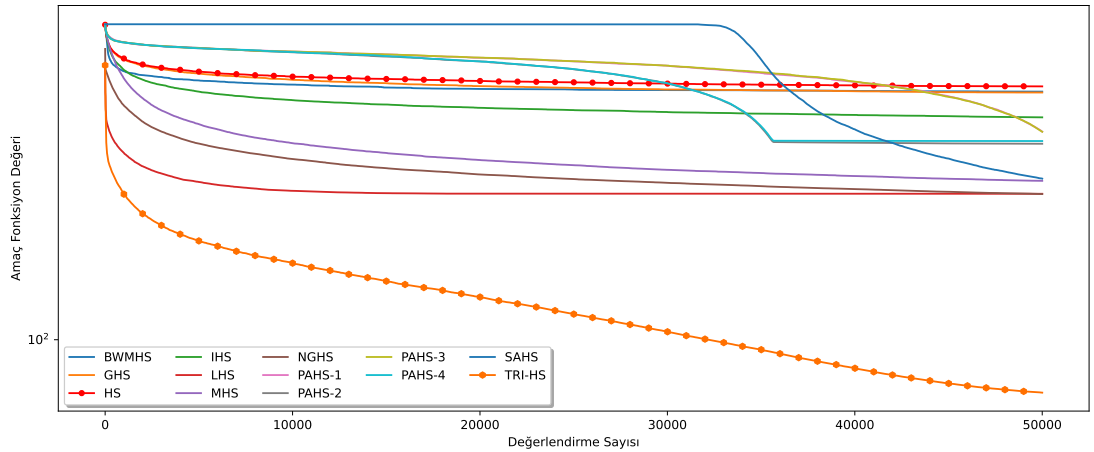
Gürültü eklenmiş EEG sinyalleri üzerindeki 19 kanallı problem üzerinde yapılan koşmalar için ortalama ve standart sapma değerleri 4.23 ile gösterilen deneylerde HS varyantlarının koşmaların başından sonuna kadar oluşturdukları yakınsama grafiği Şekil



Şekil 4.24. Gürültü eklenmiş 4 Kanallı EEG problemi için yakınsama grafiği (HMS: 5)

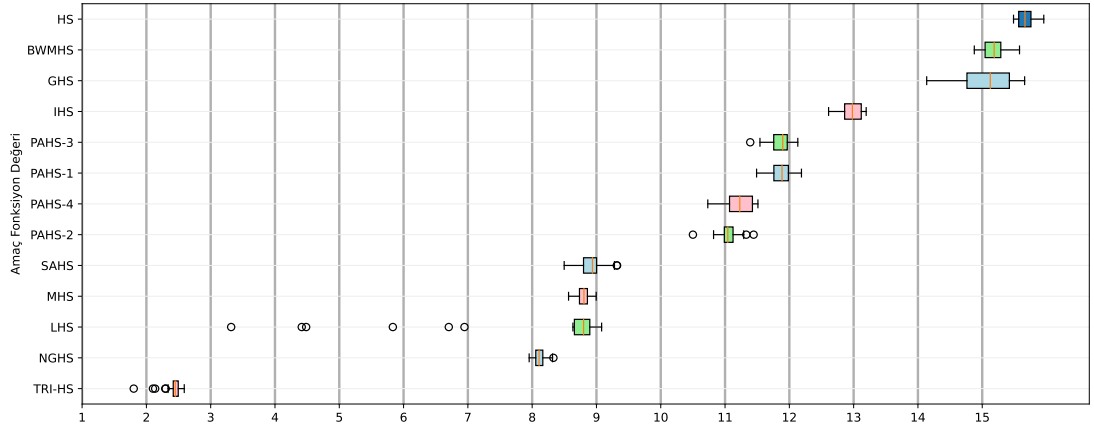


Şekil 4.25. Gürültü eklenmiş 4 Kanallı EEG problemi için kutu grafiği (HMS: 5)

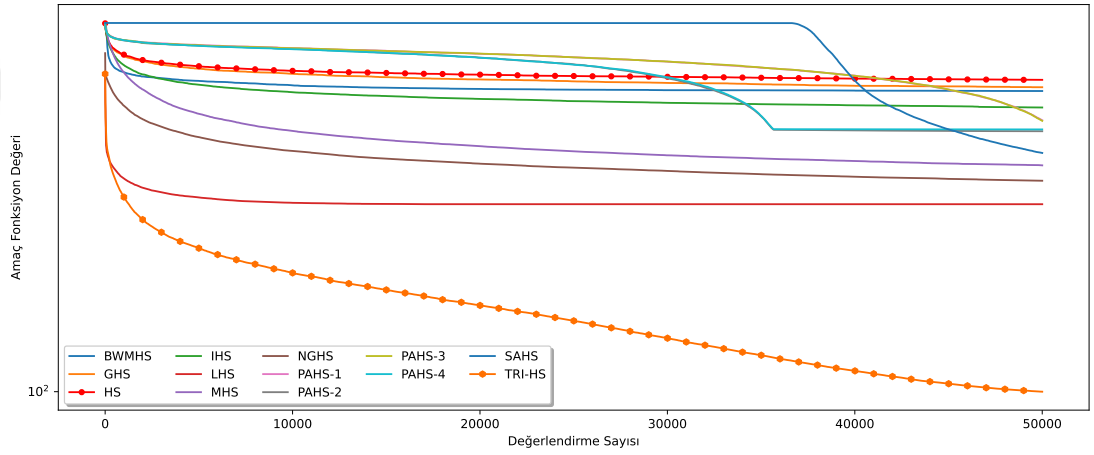


Şekil 4.26. Gürültü eklenmiş 12 Kanallı EEG problemi için yakınsama grafiği (HMS: 5)

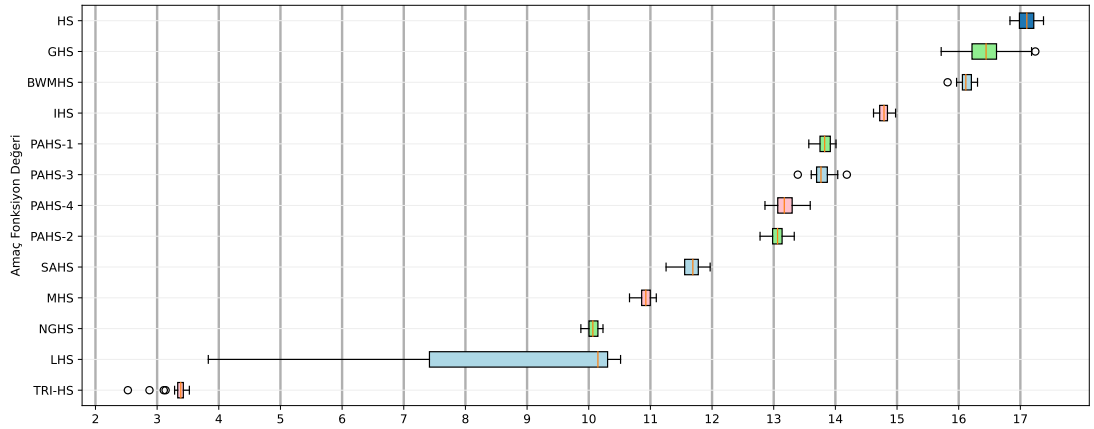
4.28 ile, koşmalar sonucunda elde ettikleri değerlere ait kutu grafiği Şekil 4.29 ile gösterilmiştir. Yakınsama grafiği gürültüsüz koşma ile çok benzer olup kutu grafiğinde ise LHS algoritmasının değer aralığının genişliği dikkat çekmektedir.



Şekil 4.27. Gürültü eklenmiş 12 Kanallı EEG problemi için kutu grafiği (HMS: 5)



Şekil 4.28. Gürültü eklenmiş 19 Kanallı EEG problemi için yakınsama grafiği (HMS: 5)



Şekil 4.29. Gürültü eklenmiş 19 Kanallı EEG problemi için kutu grafiği (HMS: 5)

Gürültü eklenmiş EEG sinyalleri üzerinde HS varyantları ile $HMS = 10$ parametresi ile elde edilen ortalama sonuçlar ve standart sapma değerleri Tablo 4.23 ile gösterilmiştir. 4, 12 ve 19 kanallı koşmaların tümünde TRI-HS varyantı en iyi sonucu vermiştir. Elde edilen sonuçlar incelendiğinde, bu tez çalışması kapsamında

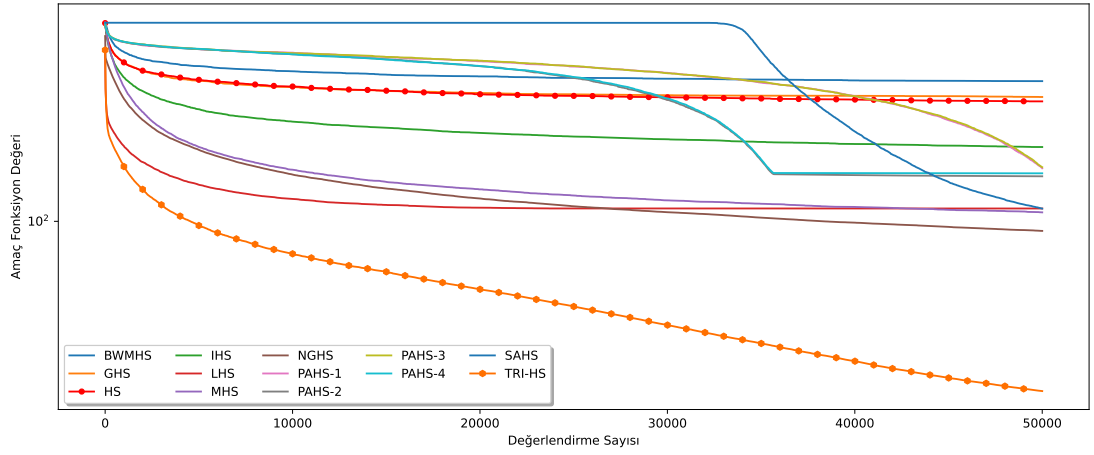
geliştirilen TRI-HS algoritmasıyla elde edilen değerler, orijinal HS algoritmasından 4 kanallı yani 1024 parametrelili problem için **16** kat, 12 kanallı yani 3072 parametrelili problem için **8.1** kat, 19 kanallı yani 4864 parametrelili problem için ise **6.7** kat daha hassas sonuçlar elde etmiştir.

Tablo 4.24. Gürültü eklenmiş EEG Problemi sonuçları (HMS: 10)

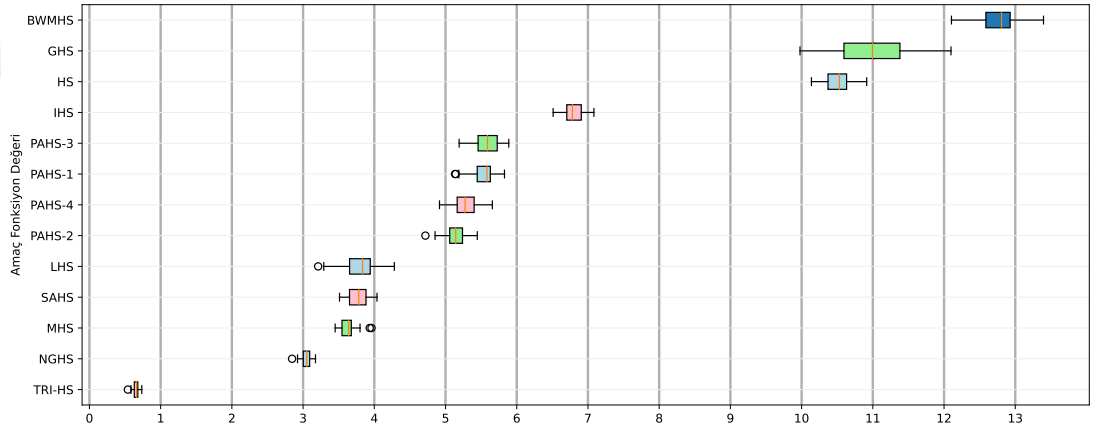
Algoritma	Fn.	D = 4	D = 12	D = 19
HS	Ort.	1.0513e+01	1.507e+01	1.654e+01
	Std.	(1.75e-01)	(9.97e-02)	(9.98e-02)
IHS	Ort.	6.7910e+00	1.1776e+01	1.3692e+01
	Std.	(1.43e-01)	(1.35e-01)	(1.13e-01)
GHS	Ort.	1.0965e+01	1.5213e+01	1.6605e+01
	Std.	(5.40e-01)	(3.67e-01)	(3.68e-01)
MHS	Ort.	3.6323e+00	7.5697e+00	9.6749e+00
	Std.	(1.24e-01)	(1.28e-01)	(7.65e-02)
NGHS	Ort.	3.0413e+00	8.1533e+00	1.0043e+01
	Std.	(6.66e-02)	(1.25e-01)	(9.87e-02)
PAHS-1	Ort.	5.5377e+00	1.0519e+01	1.2589e+01
	Std.	(1.85e-01)	(1.71e-01)	(1.45e-01)
PAHS-2	Ort.	5.131e+00	9.7695e+00	1.1851e+01
	Std.	(1.64e-01)	(1.69e-01)	(1.69e-01)
PAHS-3	Ort.	5.5877e+00	1.0564e+01	1.2588e+01
	Std.	(1.73e-01)	(1.56e-01)	(1.55e-01)
PAHS-4	Ort.	5.2788e+00	9.8994e+00	1.2003e+01
	Std.	(1.84e-01)	(1.59e-01)	(1.66e-01)
LHS	Ort.	3.7693e+00	7.0138e+00	7.9922e+00
	Std.	(2.46e-01)	(6.9e-01)	(1.26e+00)
BWMHS	Ort.	1.2761e+01	1.6364e+01	1.7467e+01
	Std.	(2.56e-01)	(1.43e-01)	(1.07e-01)
SAHS	Ort.	3.7646e+00	9.3202e+00	1.206e+01
	Std.	(1.49e-01)	(1.56e-01)	(1.70e-01)
TRI-HS	Ort.	6.5591e-01	1.8534e+00	2.4401e+00
	Std.	(4.37e-02)	(6.42e-02)	(5.56e-02)

Gürültü eklenmiş 4 kanallı EEG sinyalleri için $HMS = 10$ değeri ile yapılan koşmalarda HS varyantlarının yakınsama grafikleri Şekil 4.30 ile, kutu grafiği Şekil 4.31 ile gösterilmiştir. TRI-HS varyantının diğer varyantlardan çok daha hızlı bir yakınsama sergilediği, BWMHS ve GHS haricindeki varyantların orijinal HS algoritmasından daha iyi sonuçlar verdiği görülmektedir.

Gürültü eklenmiş 12 kanallı EEG sinyallerinin gürültüden temizlenmesi problemi için HS varyantlarının ortak parametresi olan $HMS = 10$ değeri ile koşmaların başlangıç iterasyonundan son iterasyona kadar elde ettiği sonuçların



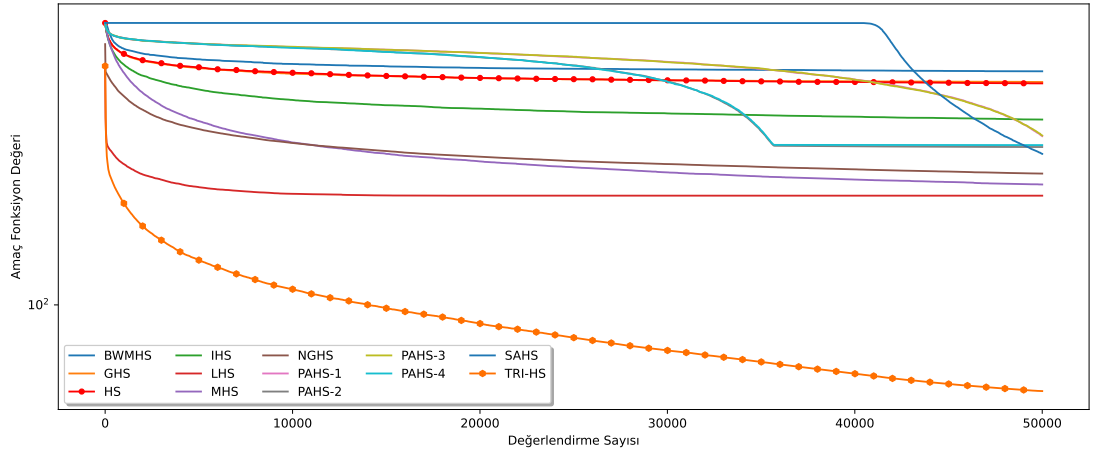
Şekil 4.30. Gürültü eklenmiş 4 kanallı EEG problemi için yakınsama grafiği (HMS: 10)



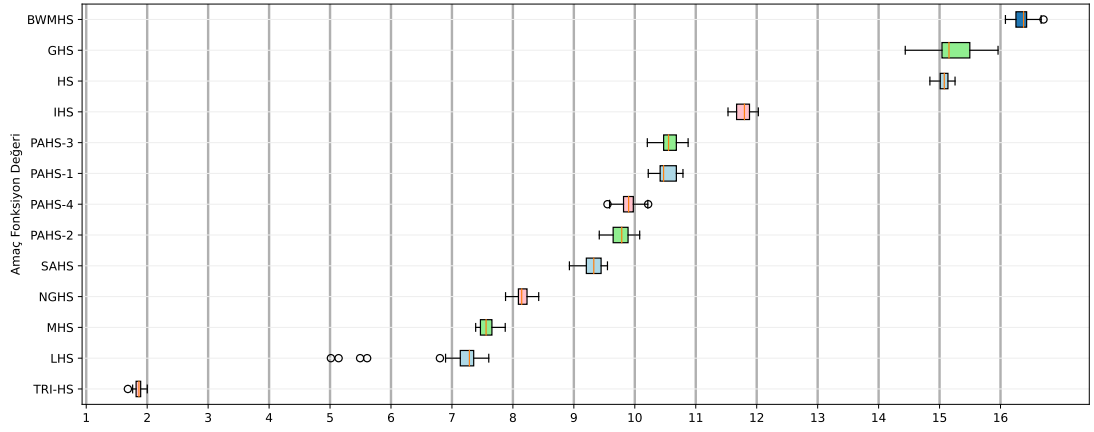
Şekil 4.31. Gürültü eklenmiş 4 kanallı EEG problemi için kutu grafiği (HMS: 10)

görselleştirilmesiyle elde edilen yakınsama grafiği Şekil 4.32 ile, ağırsız koşmaların sonuçlarının ortalamaların alınarak elde edilen kutu grafiği 4.33 ile gösterilmiştir. TRI-HS varyantının hızlı bir yakınsama eğilimini burada da sürdürdüğü ve en iyi sonucu elde ettiği görülmektedir. BMWHS ve GHS varyantları orijinal HS algoritmasından daha az başarılı sonuçlar elde etmiştir.

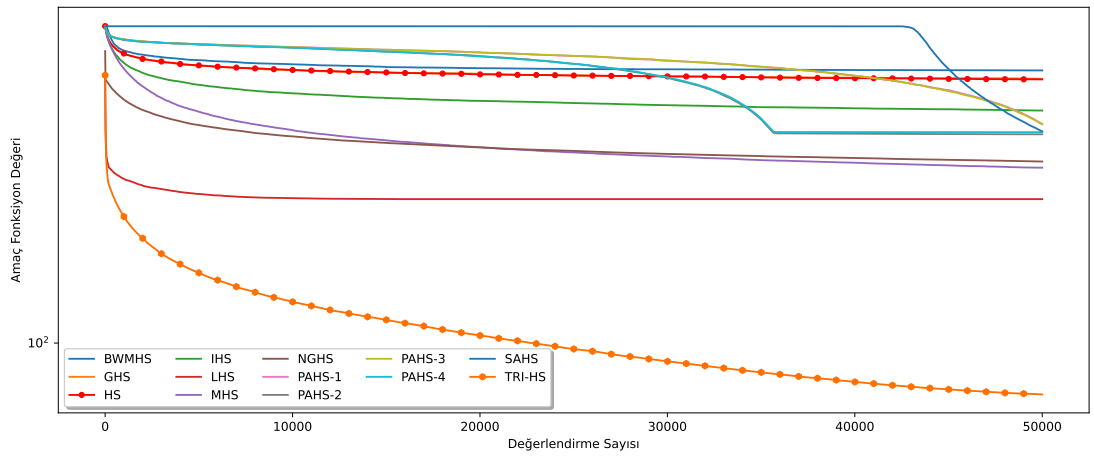
Gürültü eklenmiş 19 kanallı EEG sinyalleri için $HMS = 10$ değeri ile elde edilen yakınsama grafiği Şekil 4.34 ile, kutu grafiği 4.35 ile gösterilmiştir. Gürültü eklenmiş 19 kanallı problem için HS varyantlarının aynı HMS değeri verilerek yapılan koşmalarda oluşturdukları yakınsama grafiği 12 kanallı koşma ile büyük oranda benzerlik gösterirken LHS varyantı ile MHS ve NGHS varyantları arasındaki fark oransal olarak bir miktar artmıştır. Kutu grafiğinden görülebileceği üzere TRI-HS varyantı ortalama değer ve değer aralığı olarak diğer varyantlardan çok daha iyi sonuçlar elde edebilmiştir.



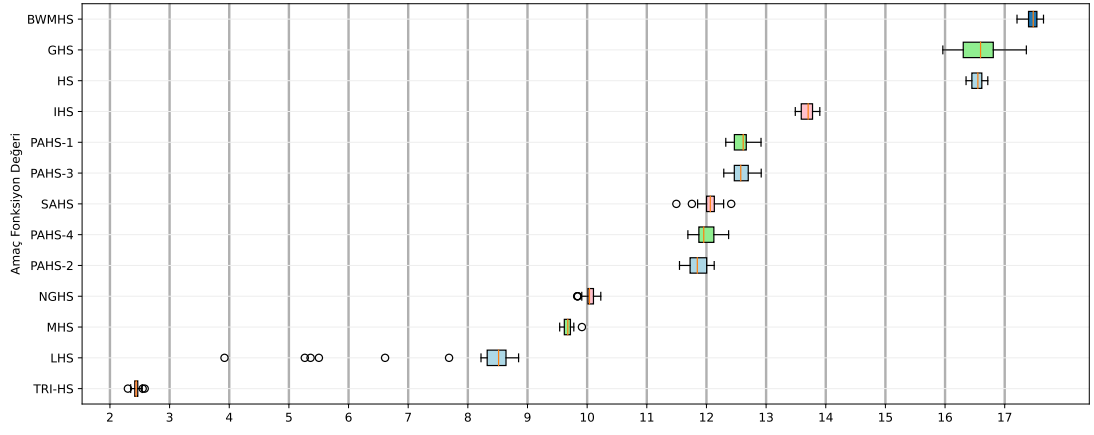
Şekil 4.32. Gürültü eklenmiş 12 kanallı EEG problemi için yakınsama grafiği (HMS: 10)



Şekil 4.33. Gürültü eklenmiş 12 kanallı EEG problemi için kutu grafiği (HMS: 10)



Şekil 4.34. Gürültü eklenmiş 19 kanallı EEG problemi için yakınsama grafiği (HMS: 10)



Şekil 4.35. Gürültü eklenmiş 19 kanallı EEG problemi için kutu grafiği (HMS: 10)

5. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasının ilk bölümünde araştırma konusu olan EEG sinyallerinden gürültülerin temizlenmesi problemine değinilmiş, ikinci bölümde optimizasyon algoritmalarının çeşitleri ve optimizasyon problemlerinin yapısı ile birlikte bir büyük veri optimizasyon problemi olan EEG sinyallerinin gürültüden temizlenmesi probleminin yapısı incelenmiştir. Üçüncü bölümde ise HS algoritması detaylı şekilde açıklanmış ve HS varyantlarından IHS, GHS, MHS, SGHS, EHS, NGHS, PHS, IPHS, ITHS, NDHS, NSHS, DSHS, PAHS, GDHS, SIHS, FHS, LHS, NEHS, IFSHS, BWMHS ve ID-HS-LDD incelenerek HS algoritması üzerinde yaptıkları değişiklikler ve uyguladıkları stratejiler değerlendirilmiştir.

Tez çalışmaları kapsamında, incelenen HS varyantlarından elde edilen bilgi birikimi ve deneyimler kullanılarak TRI-HS adı verilen yeni bir HS varyantı da literatüre sunulmaktadır. Tezde, büyük veri optimizasyon problemi olan EEG sinyallerinin tarama sırasında dış etkenlerden kaynaklı oluşan gürültülerden temizlenmesi problemi HS algoritması ve varyantlarını kullanarak optimize edilmeye çalışılmış, bu problem için en başarılı HS varyantı bulunmaya çalışılmıştır. Halihazırda literatürdeki varyantlardan birçoğu implemente edilerek yaygın olarak kullanılan kıyaslama fonksiyonları üzerinde ve EEG sinyallerinin gürültüden temizlenmesi problemi üzerindeki performansları ve yakınsama grafikleri incelenmiştir. Elde edilecek çözümlerin kalitesini artırmak adına mevcut varyantlar incelenerek LHS varyantının birçok problem üzerinde diğer varyantlardan daha başarılı olduğu ve işleyişinde yapılacak değişikliklerle daha iyi sonuçlar elde edebileceği değerlendirilerek TRI-HS varyantı oluşturulmuştur.

HS algoritmasının en dikkat edilmesi gereken ve doğru şekilde ayarlanması için deneyime ihtiyaç duyulan hususu olan algoritma parametrelerini belirleme işlemi için birçok HS varyantının parametreleri dinamikleştirilmesinin getirdiği avantaj ile birlikte, bazı varyantların mevcut parametreleri dinamik hale getirmek için parametre sayısını daha da artırması da bir dezavantaj olarak karşımıza çıkmaktadır. HS algoritmasının *HMCR* , *PAR* ve *bw* parametrelerini dinamik hale getiren veya çeşitli evrimsel operatörler yardımıyla çözüm üreten varyantların çoğunlukla HS algoritmasının

orijinal halinden daha başarılı sonuçlar üretebildiği gözlemlenmiştir. EEG sinyalinin gürültüden temizlenmesi problemi üzerinde de bu tezde önerilen TRI-HS varyantının 4, 12 ve 19 kanallı sinyallerin orijinal ve gürültü eklenmiş hallerinin hepsinin üzerinde orijinal HS ve diğer varyantlardan daha başarılı sonuçlar elde ettiği ve çok daha erken aşamalarda daha yüksek yakınsama hızları ortaya koyduğu belirlenmiştir. Kıyaslama fonksiyonlarının birçoğu üzerinde de 30, 50, 100, 250 ve 500 parametrelili problemlerin büyük çoğunluğunda sonradan geliştirilen varyantların orijinal HS algoritmasından çoğunlukla daha iyi sonuçlar üretebildiği, TRI-HS varyantının ise birçok problemde diğer varyantlardan da daha başarılı sonuçlar üretebildiği gözlemlenmiştir.

Literatürdeki HS varyantları ve TRI-HS varyantının daha farklı yapıda ve günlük hayat problemleri üzerindeki başarımı, diğer sezgisel ve yarı-sezgisel algoritmalar ile adil şartlarda ve farklı iterasyon sayılarıyla yapılacak koşmalarda, kullanılan kaynaklar ve çalışma zamanı da dikkate alınarak kapsamlı değerlendirmeler yapılması gelecekte yürütülmesi planlanan çalışmalar arasındadır.

HS algoritması, basit yapısı ve keşif - sömürü dengesini sağlayabilmesi açısından birçok problem için diğer algoritmalarından daha başarılı sonuç üretme potansiyeline sahip olup üzerinde yapılacak problem bağımlı veya problemden bağımsız geliştirmelerle daha da etkili bir algoritma haline gelebilmektedir. Literatürdeki mevcut varyantların ve diğer evrimsel algoritmaların güçlü yönleri bir araya getirilerek hibrit algoritmalar oluşturularak farklı karaktere sahip problemler üzerinde daha etkili varyantlar elde edilebilir. Bu hususlar değerlendirildiğinde HS algoritması ile birlikte literatürdeki varyantlarının da güncel problemlere uygulanması önerilmektedir.

KAYNAKLAR

- Altunay, H., & Tamer, E. (2017). Ders programı çizelgeleme problemi için bir literatür taraması. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 23(1), 55–70.
- Arora, R. K. (2015). *Optimization: algorithms and applications*. CRC press.
- BigOpt. (2015). <http://www.husseinabbass.net/BigOpt.html>. ([Erişim Tarihi: 06 Haziran 2023])
- Burke, E. K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., & Qu, R. (2013). Hyper-heuristics: A survey of the state of the art. *Journal of the Operational Research Society*, 64, 1695–1724.
- Burke, E. K., Hyde, M., Kendall, G., Ochoa, G., Ozcan, E., & Qu, R. (2009). A survey of hyper-heuristics. *Computer Science Technical Report No. NOTTCS-TR-SUB-0906241418-2747, School of Computer Science and Information Technology, University of Nottingham*.
- Cavazzuti, M. (2013). Deterministic optimization. *Optimization Methods: From Theory to Design Scientific and Technological Aspects in Mechanics*, 77–102.
- Cec BigOpt. (2015). <http://www.husseinabbass.net/BigOptSessionCec2015.html>. ([Erişim Tarihi: 04 Haziran 2023])
- Chen, J., Pan, Q.-k., & Li, J.-q. (2012). Harmony search algorithm with dynamic control parameters. *Applied Mathematics and Computation*, 219(2), 592–604.
- Cheng, Y. M., Li, L., Lansivaara, T., Chi, S., & Sun, Y. (2008). An improved harmony search minimization algorithm using different slip surface generation methods for slope stability analysis. *Engineering Optimization*, 40(2), 95–115.
- Coello, C. A. C. (2000). Treating constraints as objectives for single-objective evolutionary optimization. *Engineering Optimization+ A35*, 32(3), 275–308.
- Das, S., Mukhopadhyay, A., Roy, A., Abraham, A., & Panigrahi, B. K. (2010). Exploratory power of the harmony search algorithm: analysis and improvements for global numerical optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(1), 89–106.
- Desale, S., Rasool, A., Andhale, S., & Rane, P. (2015). Heuristic and meta-heuristic algorithms and their relevance to the real world: a survey. *Int. J. Comput. Eng. Res. Trends*, 351(5), 2349–7084.
- Dubey, M., Kumar, V., Kaur, M., & Dao, T.-P. (2021). A systematic review on harmony search algorithm: theory, literature, and applications. *Mathematical Problems in Engineering*, 2021, 1–22.
- Gao, X. Z., Wang, X., & Ovaska, S. J. (2010). A hybrid harmony search method based on obl. In *2010 13th IEEE international conference on computational science and engineering* (pp. 140–145).
- Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: harmony search. *Simulation*, 76(2), 60–68.
- Jaberipour, M., & Khorram, E. (2010). Two improved harmony search algorithms for solving engineering optimization problems. *Communications in Nonlinear Science and Numerical Simulation*, 15(11), 3316–3331.

- Kalivarapu, J., Jain, S., & Bag, S. (2016). An improved harmony search algorithm with dynamically varying bandwidth. *Engineering Optimization*, 48(7), 1091–1108.
- Kattan, A., & Abdullah, R. (2013). A dynamic self-adaptive harmony search algorithm for continuous optimization problems. *Applied Mathematics and Computation*, 219(16), 8542–8567.
- Khalili, M., Kharrat, R., Salahshoor, K., & Sefat, M. H. (2014). Global dynamic harmony search algorithm: Gdhs. *Applied Mathematics and Computation*, 228, 195–219.
- Kleinrock, L. (2010). An early history of the internet [history of communications]. *IEEE Communications Magazine*, 48(8), 26–36.
- Kokash, N. (2005). An introduction to heuristic algorithms. *Department of Informatics and Telecommunications*, 1–8.
- Kumar, V., Chhabra, J. K., & Kumar, D. (2014). Parameter adaptive harmony search algorithm for unimodal and multimodal optimization problems. *Journal of Computational Science*, 5(2), 144–155.
- Li, S., Zhang, D., Shao, Z., & Tang, H. (2020). Information feedback self-adaptive harmony search algorithm for the bovine cortical bone vibration-assisted drilling optimization. *Measurement*, 149, 107020.
- Li, Z., Zou, D., & Kong, Z. (2019). A harmony search variant and a useful constraint handling method for the dynamic economic emission dispatch problems considering transmission loss. *Engineering Applications of Artificial Intelligence*, 84, 18–40.
- Luo, K. (2013). A novel self-adaptive harmony search algorithm. *Journal of Applied Mathematics*, 2013.
- Mahdavi, M., Fesanghary, M., & Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied mathematics and computation*, 188(2), 1567–1579.
- Malisia, A. R., & Tizhoosh, H. R. (2007). Applying opposition-based ideas to the ant colony system. In *2007 IEEE Swarm Intelligence Symposium* (pp. 182–189).
- Meselhi, M. A., Elsayed, S. M., Sarker, R. A., & Essam, D. L. (2021). Parallel evolutionary algorithm for eeg optimization problems. In *2021 IEEE Congress on Evolutionary Computation (CEC)* (pp. 2577–2584).
- Omran, M. G., & Mahdavi, M. (2008). Global-best harmony search. *Applied mathematics and computation*, 198(2), 643–656.
- Ouyang, H.-b., Gao, L.-q., Li, S., Kong, X.-y., Wang, Q., & Zou, D.-x. (2017). Improved harmony search algorithm: Lhs. *Applied Soft Computing*, 53, 133–167.
- Pan, Q.-K., Suganthan, P. N., Tasgetiren, M. F., & Liang, J. J. (2010). A self-adaptive global best harmony search algorithm for continuous optimization problems. *Applied Mathematics and Computation*, 216(3), 830–848.
- Qin, A. K., & Forbes, F. (2011). Dynamic regional harmony search with opposition and local learning. In *Proceedings of the 13th annual conference companion on genetic and evolutionary computation* (pp. 53–54).

- Qin, F., Zain, A. M., & Zhou, K.-Q. (2022). Harmony search algorithm and related variants: A systematic review. *Swarm and Evolutionary Computation*, 101126.
- Rahnamayan, S., Tizhoosh, H. R., & Salama, M. M. (2008). Opposition-based differential evolution. *IEEE Transactions on Evolutionary computation*, 12(1), 64–79.
- Sanei, S., & Chambers, J. A. (2013). *Eeg signal processing*. John Wiley & Sons.
- Singh, R., Mukherjee, V., & Ghoshal, S. (2013). The opposition-based harmony search algorithm. *Journal of the Institution of Engineers (India): Series B*, 94, 247–256.
- Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K., Chen, Y.-P., Auger, A., & Tiwari, S. (2005). Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. *KanGAL report, 2005005*(2005), 2005.
- Talaei, K., Rahati, A., & Idoumghar, L. (2020). A novel harmony search algorithm and its application to data clustering. *Applied Soft Computing*, 92, 106273.
- Tizhoosh, H. R. (2005). Opposition-based learning: a new scheme for machine intelligence. In *International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (cimca-iawtic'06)* (Vol. 1, pp. 695–701).
- Wang, H., Li, H., Liu, Y., Li, C., & Zeng, S. (2007). Opposition-based particle swarm algorithm with cauchy mutation. In *2007 IEEE congress on evolutionary computation* (pp. 4750–4756).
- Wang, S.-W., Ding, L.-X., Xie, D.-T., Shu, W.-N., Xie, C.-W., & Yang, H. (2012). Group search optimizer applying opposition-based learning. *Computer Science*, 39(9), 183–187.
- Wójcicki, K., Biegańska, M., Paliwoda, B., & Górna, J. (2022). Internet of things in industry: Research profiling, application, challenges and opportunities—a review. *Energies*, 15(5), 1806.
- Yadav, P., Kumar, R., Panda, S. K., & Chang, C. (2012). An intelligent tuned harmony search algorithm for optimisation. *Information Sciences*, 196, 47–72.
- Yang, X.-S., & He, X. (2016). Nature-inspired optimization algorithms in engineering: overview and applications. *Nature-inspired computation in engineering*, 1–20.
- Zhong, W., Liu, J., Xue, M., & Jiao, L. (2004). A multiagent genetic algorithm for global numerical optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(2), 1128–1141.
- Zhu, Q., Tang, X., Li, Y., & Yeboah, M. O. (2020). An improved differential-based harmony search algorithm with linear dynamic domain. *Knowledge-Based Systems*, 187, 104809.
- Zou, D., Gao, L., Wu, J., Li, S., & Li, Y. (2010). A novel global harmony search algorithm for reliability problems. *Computers & Industrial Engineering*, 58(2), 307–316.

ÖZGEÇMİŞ

Serhat Celil İLERİ, Ondokuz Mayıs Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği lisans bölümünden 2019 yılında derece ile mezun oldu. Yüksek lisans öğrenimine Ondokuz Mayıs Üniversitesi Lisansüstü Eğitim Enstitüsünde 2019 yılında başladı (04/07/2023).

İletişim Bilgileri

ORCID ID : 0000-0002-0259-0791

Yayınlar :

1. İleri, S. C. , Aslan, S. & Demirci, S. (2022). Büyük Veri Optimizasyonu için Kaynak-Bağılantılı Harmoni Arama Algoritmasının Performans Analizi . Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi , 15 (2) , 151-160 . DOI: 10.54525/tbbmd.1090787
2. Demirci, S., İleri, S. C., & Duraki, S. (2022). OMNeT++ Framework for Simulation of Centralized and Distributed Algorithms in Multi-Hop Networks. In Handbook of Research on Advances in Data Analytics and Complex Communication Networks (pp. 1-33). IGI Global.
3. İleri, S.C., S., Aslan, S., & Demirci, S. (2021). A Novel Harmony Search Based Method for Noise Minimization on EEG Signals. In 2021 6th International Conference on Computer Science and Engineering (UBMK) (pp. 747-750). IEEE.
4. İleri, S. C., Aslan, S., & Demirci, S. (2021). Mobil Hava Baz İstasyonu İçin En İyi Konumun Bulunmasında Optimizasyon Algoritmalarının Karşılaştırılması. Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 14(1), 68-76.
5. İleri, S. C., Aslan, S., & Demirci, S. (2020). Finding Optimum Location for a Mobile Aerial Base Station with Harmony Search Algorithm. In 2020 5th International Conference on Computer Science and Engineering (UBMK) (pp. 224-227). IEEE.
6. Demirci, S., Aslan, S., & İleri, S. C. (2019), Performance Analysis of Parallel Differential Evolution Algorithm with Cooperative Emigrant Creation Strategy. 2019 1st International Informatics and Software Engineering Conference (UBMYK), Ankara, Turkey, 2019, pp. 1-6, doi: 10.1109/UBMYK48245.2019.8965600.
7. Aslan, S., & İleri, S. C. (2019). Performance analysis of ARM big. LITTLE architecture based mobile processor with multi-thread face detection. In 2019 4th International Conference on Computer Science and Engineering (UBMK) (pp. 336-339). IEEE.
8. İleri, S. C., Aslan, S., & Demirci, S., (2019). Konferans, Kongre ve Sempozyum Yönetim Sistemleri için Bir Mobil İstemci Uygulama . 13. Uusal Yazılım Mühendisliği Sempozyumu (UYMS-2019) (pp.15). İzmir, Turkey

9. İleri, S. C., Kiroğlu, T. Y., Yazbahar, H., & Aslan, S. (2019). Çok Çekirdek Kullanımının Mobil İşlemcilerin Frekansları Üzerindeki Etkisi.

10. İleri, S. C., Karabina, A., & Kılıç, E. (2018). Comparison of Different Normalization Techniques on Speakers' Gender Detection. Mehmet Akif Ersoy Üniversitesi Uygulamalı Bilimler Dergisi, 2(2), 1-12.

