

T.C.
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
İSTATİSTİK ANABİLİM DALI

**YAPAY SİNİR AĞLARININ MODERN SEZGİSEL-ÜSTÜ ALGORİTMALARI
YOLUYLA EĞİTİMİ**

DOKTORA TEZİ

HAZIRLAYAN: ERDAL EKER
DANIŞMAN: Prof. Dr. Murat KAYRI
İKİNCİ DANIŞMAN: Doç. Dr. Serdar EKİNCİ

VAN-2020

T.C.
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
İSTATİSTİK ANABİLİM DALI

**YAPAY SİNİR AĞLARININ MODERN SEZGİSEL-ÜSTÜ ALGORİTMALARI
YOLUYLA EĞİTİMİ**

DOKTORA TEZİ

HAZIRLAYAN: Erdal EKER

VAN-2020

KABUL VE ONAY SAYFASI

İstatistik Anabilim Dalı'nda Prof. Dr. Murat KAYRI ve Doç. Dr. Serdar EKİNCİ danışmanlığında, Erdal EKER tarafından sunulan “**Yapay Sinir Ağlarının Modern Sezgisel-Üstü Algoritmaları Yoluyla Eğitimi**” isimli bu çalışma Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili hükümleri gereğince .../...../2020 tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile başarılı bulunmuş ve doktora tezi olarak kabul edilmiştir.

Başkan: Prof. Dr. Sinan ÇALIK

İmza:

Üye: Prof. Dr. Murat KAYRI

İmza:

Üye: Prof. Dr. Fevzi ERDOĞAN

İmza:

Üye: Doç. Dr. Sıddık KESKİN

İmza:

Üye: Prof. Dr. Ömer Faruk ERDOĞAN

İmza:

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun/..../2020 tarih ve
..... sayılı kararı ile onaylanmıştır.

...../...../2020

Prof. Dr. Suat ŞENSOY

Enstitü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

Erdal EKER

ÖZET

YAPAY SİNİR AĞLARININ MODERN SEZGİSEL-ÜSTÜ ALGORİTMALAR YOLUYLA EĞİTİMİ

EKER, Erdal

Doktora Tezi, İstatistik Anabilim Dalı
Tez danışmanı: Prof. Dr. Murat KAYRI
İkinci danışman: Doç. Dr. Serdar EKİNCİ
Ağustos 2020, 127 sayfa

Bu tezin amacı, doğrusal olmayan problemlerin optimizasyonunda kullanılan ve iyi bir sınıflandırma aracı olan Yapay Sinir Ağlarının (YSA) eğitimini sağlamaktır. Tezin amacına uygun olarak YSA mimarilerinden Çok Katmanlı Algılayıcı (Multilayer Perceptron -MLP) kullanılmıştır. MLP eğitiminde olasılıksal algoritma çeşidi olan modern sezgisel-üstü algoritmalar eğitmen olarak seçilmiştir. Algoritmaların genel performansını ölçmek ve performans üstünlüğünü karşılaştırmak için Ackley, Easom, Egg Crate, Griewank, Quartic, Rosenbrock, Schwefel, Sphere ve Step kriter fonksiyonları kullanılmıştır. MLP eğitimi için UCI veri bankasından XOR, İris, Meme Kanseri, Kalp Hastalıkları ve Balloon veri setleri alınmıştır.

Tezde dört uygulama yapılmıştır. Uygulamalardan ilki Harris Şahinleri Optimizasyonu sezgisel-üstü algoritmasının (HHO) diğer algoritmalarla karşılaştırılması yapılarak üstünlükleri gösterilmiştir. İkinci uygulamada modern sezgisel-üstü algoritmaların global kısıtlı optimizasyon problemlerinin çözümü gösterilerek Atom Arama Optimizasyonu'nun (ASO) performans üstünlüğü istatistiki olarak ispat edilmiştir. Üçüncü uygulamada MLP eğitimi yapılarak Sinüs Kosinüs Algoritması (SCA) ve HHO algoritmaları karşılaştırılmıştır. Son uygulamada ise önerilen hibrid ASO-SA algoritması (hASO-SA) ile problem çözümündeki sezgisel-üstü algoritmaların dezavantajları ortadan kaldırılmıştır. Performans avantajı istatistiki olarak gösterilerek MLP eğitiminde üstün sınıflandırma becerisi, diğer sezgisel-üstü algoritmalarla karşılaştırılarak açıklanmıştır.

Anahtar kelimeler: Çok katmanlı algılayıcılar, optimizasyon, sezgisel-üstü algoritmalar, yapay sinir ağları.

ABSTRACT

TRAINING OF ARTIFICIAL NEURAL NETWORKS VIA MODERN META- HEURISTIC ALGORITHMS

EKER, Erdal

Ph.D. Thesis, Department of Statistic

1st Supervisor: Prof. Dr. Murat KAYRI

2nd Supervisor: Assoc. Prof. Dr. Serdar EKINCI

August 2020, 127 pages

The purpose of this thesis is to provide ANN training, which is a good classification tool used in optimization problems. In accordance with the purpose of the thesis, Multilayer Perceptron (MLP) is used from ANN architectures. Modern meta-heuristic algorithms which are probabilistic algorithm types in MLP training are selected as trainers. Ackley, Easom, Egg Crate, Griewank, Quartic, Rosenbrock, Schwefel, Sphere and Step benchmark functions are used to measure the overall performance of the algorithms and to compare performance advantage. XOR, Iris, Breast Cancer, Heart Diseases and Balloon datasets were taken from the UCI database to demonstrate the closeness of the classification rates to reality in MLP education.

Four applications were made in the thesis. In the first application, the advantages of Harris Hawks Optimization algorithm (HHO) were compared with other algorithms. In the second application, the performance advantages of the Atom Search Algorithm (ASO) has been statistically demonstrated by showing the solution of the global constrained optimization problems of the modern meta-heuristic algorithms. In the third application, the Sine Cosine Algorithm (SCA) and HHO algorithms were compared by making MLP training. In the fourth application, through proposed hybrid ASO-SA algorithm (hASO-SA) has been to eliminated by the disadvantages of the meta-heuristic algorithms in the problem solution. By showing the performance advantage statistically, high classification skill in MLP education is explained by comparing with other meta-heuristic algorithms.

Keywords: Meta-heuristic algorithms, Multilayer perceptron, Neural networks, Optimization.



ÖN SÖZ

Bu tez çalışmasında, her türlü ilgi ve yardımlarını esirgemeyen danışmanlarım Sayın Prof. Dr. Murat KAYRI ve Doç. Dr. Serdar EKİNCİ'ye teşekkür ederim. Ayrıca manevi destekleri ile anne ve babama ve fedakarlıklarıyla bana çalışma rahatlığı sağlayan sevgili eşime ve çocuklarıma teşekkürü bir borç bilirim.

2020

Erdal EKER



İÇİNDEKİLER

	Sayfa
ÖZET	i
ABSTRACT	iii
ÖN SÖZ	v
İÇİNDEKİLER	vii
ÇİZELGELER LİSTESİ	xi
ŞEKİLLER LİSTESİ	xiii
SİMGELER VE KISALTMALAR	xvii
1. GİRİŞ	1
1.1. Sezgisel-Üstü Algoritmalar ve Optimizasyonu	2
1.1.1. Deterministik algoritmalar	7
1.1.1.1. Tabu arama algoritması	7
1.1.2. Olasılıksal algoritmalar	10
1.1.2.1. Benzetilmiş tavlama algoritması	10
1.1.2.2. Evrimsel algoritmalar	13
1.1.2.3. Genetik algoritma	15
1.1.2.4. Evrimsel programlama ve evrimsel strateji algoritmaları	18
1.1.2.5. Diferansiyel evrim algoritması	21
1.1.3. Karınca kolonisi optimizasyon algoritması	23
1.1.4. Parçacık sürü optimizasyon algoritması	28
1.1.5. Yapay arı kolonisi algoritması	30
1.1.6. Rüzgâr güdümlü optimizasyon algoritması	33
1.1.7. Gri kurt optimizasyon algoritması	37
1.1.8. Sinüs kosinüs algoritması	41
1.1.9. Atom arama optimizasyon algoritması	45
1.1.10. Harris şahinleri optimizasyon algoritması	49
1.1.11. hASO-SA hibrit algoritması	54
1.2. Yapay Sinir Ağları	56
1.2.1. YSA mimarisi ve eğitimi	61
1.2.1.1. Tek katmanlı algılayıcılar	61

1.2.1.2. Çok katmanlı algılayıcılar.....	62
1.2.1.3. Yenilemeli ağlar.....	65
2. KAYNAK BİLDİRİŞLERİ.....	67
2.1. YSA'nın Tarihi Gelişimi	67
2.2. Sezgisel-Üstü Algoritmaların YSA Eğitiminde Kullanım Tarihçesi.....	67
3. MATERYAL VE YÖNTEM.....	71
3.1. Sezgisel-üstü Algoritmaların Optimizasyonunda Kullanılan Bazı Kriter Fonksiyonları	72
3.1.1. Ackley fonksiyonu.....	75
3.1.2. Easom fonksiyonu.....	76
3.1.3. Egg crate fonksiyonu	77
3.1.4. Griewank fonksiyonu.....	77
3.1.5. Quartic fonksiyonu	78
3.1.6. Rosenbrock fonksiyonu	79
3.1.7. Schwefel fonksiyonu.....	80
3.1.8. Sphere (De Jong's) fonksiyonları	81
3.1.9. Step fonksiyonu	82
3.2. YSA Eğitiminde Veri Setlerinin Kullanımı.....	83
3.2.1. XOR veri seti	84
3.2.2. İris veri seti	85
3.2.3. Meme kanseri (Breast Cancer) veri seti.....	87
3.2.4. Kalp hastalıkları (Heart Disease) veri seti	88
3.2.5. Balloon veri seti	89
4. BULGULAR VE TARTIŞMA.....	91
4.1. Modern Sezgisel-üstü Algoritmaların Performanslarının İstatistiki Olarak Karşılaştırılması ve Kutu Grafiği ile Performans Ölçüm Değerlendirmeleri	91
4.1.1. Algoritmaların sonuç analizleri	92
4.2. ASO, PSO, SCA, ABC Sezgisel-üstü Algoritmalarının Problem Çözme Performanslarının Karşılaştırılması ve Sonuçların Yakınsama Eğrileri Aracılığıyla Karşılaştırılması	96
4.3. Harris Şahinleri Optimizasyonunu Kullanarak Çok Katmanlı Algılayıcı Eğitimi	102
4.4. hASO-SA Hibrit Algoritması ile MLP Eğitimi	105
4.4.1. hASO-SA algoritmasının istatistiki sonuçlarının karşılaştırılması.....	105

4.4.2. hASO-SA algoritması tabanlı MLP eğitimi.....	106
5. SONUÇ.....	111
KAYNAKLAR.....	113
ÖZ GEÇMİŞ.....	127



ÇİZELGELER LİSTESİ

Çizelge	Sayfa
Çizelge 3.1. Diferransiyellenebilir kriter fonksiyonlarının mod ve ayrılabilirlik baz alınarak sınıflandırılması.....	74
Çizelge 3.2. XOR mantığını gösterir çizelge (F=0, T=1)	85
Çizelge 3.3. Veri setindeki Özellikler ve değer aralıkları	88
Çizelge 3.4. Kalp Hastalıkları veri tabanı özellikleri	88
Çizelge 3.5. Balon veri tabanının anlamsal özellikleri	89
Çizelge 4.1. Algoritmaların istatistiki performanslarının karşılaştırılması.....	92
Çizelge 4.2. Çeşitli test fonksiyonları için sonuçların karşılaştırılması	97
Çizelge 4.3. Sınıflandırma veri kümeleri ve özellikleri	104
Çizelge 4.4. Sınıflandırma veri kümeleri için sonuçlar.....	104
Çizelge 4.5. Sezgisel-üstü algoritmaların istatistiki sonuçlarının karşılaştırılması...	106
Çizelge 4.6. Sınıflandırma veri setleri.....	107
Çizelge 4.7. XOR sınıflandırma probleminin deneysel sonuçları.....	108
Çizelge 4.8. Balloon sınıflandırma probleminin deneysel sonuçları	108
Çizelge 4.9. Iris sınıflandırma probleminin deneysel sonuçları.....	108
Çizelge 4.10. Meme Kanseri sınıflandırma probleminin deneysel sonuçları	109
Çizelge 4.11. Kalp hastalıkları sınıflandırma probleminin deneysel sonuçları	109



ŞEKİLLER LİSTESİ

Şekil	Sayfa
Şekil 1.1. Sezgisel-üstü algoritmaların sınıflandırılması.....	7
Şekil 1.2. Tabu arama algoritması akış şeması.....	9
Şekil 1.3. Metropolis kuralı ile yerel optimadan kaçış.	11
Şekil 1.4. SA akış şeması.	13
Şekil 1.5. İleri beslemeli bir YSA' nın kodlama örneği.	14
Şekil 1.6. EA akış şeması.	15
Şekil 1.7. GA genel bakış.	17
Şekil 1.8. GA Akış diyagramı.	17
Şekil 1.9. (a) EP akış şeması, (b) ES akış şeması.	21
Şekil 1.10. DE algoritma şeması.	23
Şekil 1.11. Yuva ile yiyecek arasında üç yol.....	24
Şekil 1.12. Çift köprü deneyi.....	24
Şekil 1.13. ACO akış şeması.	27
Şekil 1.14. PSO çözüm güncelleme.	29
Şekil 1.15. PSO akış şeması.	30
Şekil 1.16. Nektar için balarısı yiyecek arama davranışı.	31
Şekil 1.17. ABC algoritması akış şeması.	33
Şekil 1.18. WDO algoritması akış şeması.	36
Şekil 1.19. Gri Kurt hiyerarşisi.....	37
Şekil 1.20. Gri kurtların pozisyon güncelleme mekanizması.	38

Şekil	Sayfa
Şekil 1.21. GWO algoritması.	40
Şekil 1.22. Sinüs ve kosinüsün bir sonraki pozisyona etkileri.	42
Şekil 1.23. [-2,2] aralığında sinüs ve kosinüs fonksiyonu.....	42
Şekil 1.24. [-2, 2] aralığındaki sinüs ve kosinüs fonksiyonlarının etkilerinin kavramsal modeli.	42
Şekil 1.25. $a=3$ için sinüs ve kosinüs aralığı için azalmanın gösterimi.	43
Şekil 1.26. SCA akış şeması.....	45
Şekil 1.27. Bir atom sisteminin kuvvetleri.	48
Şekil 1.28. ASO algoritma şeması.....	48
Şekil 1.29. HHO'nun tüm aşamaları.	49
Şekil 1.30. İlerlemeci hızlı dalışlar ile yumuşak kuşatma hali.	50
Şekil 1.31. İlerlemeci hızlı dalışlar ile sert kuşatma hali.....	51
Şekil 1.32. HHO akış şeması.	53
Şekil 1.33. Tepe Tırmanma hareketi	54
Şekil 1.34. hASO -SA akış şeması.	56
Şekil 1.35. (a) Normal sinir hücresi, (b) lineer olmayan sinir hücresi modeli.	58
Şekil 1.36. Aktivasyon fonksiyonları.	59
Şekil 1.37. YSA mimarileri sınıflandırılması.....	61
Şekil 1.38. Tek katmanlı sinir ağı.....	62
Şekil 1.39. İleri beslemeli sinir ağı.	63
Şekil 1.40. (a) yinelemeli ağ, (b) Gizli katmanlı yinelemeli ağ.	65
Şekil 3.1. Ackley fonksiyonu.	75
Şekil 3.2. Easom fonksiyonu.	76

Şekil	Sayfa
Şekil 3.3. Egg crate fonksiyonu.....	77
Şekil 3.4. Griewangk fonksiyonu.	78
Şekil 3.5. Quartic foksiyonu.....	79
Şekil 3.6. Rosenbrock fonksiyonu.....	80
Şekil 3.7. İki boyutlu Schwefel fonksiyonu.	81
Şekil 3.8. İki boyutlu yüzeyde Sphere fonksiyonu.	82
Şekil 3.9. Step fonksiyonu.....	83
Şekil 3.10. Sınıflandırma ağı.	84
Şekil 3.11. XOR problemini çözmek için ağın mimari grafiği.	85
Şekil 4.1. ABC, WDO, ASO ve HHO için optimizasyon süreci.....	91
Şekil 4.2. Sphere fonksiyonu için kutu grafiği.	92
Şekil 4.3. Rosenbrock fonksiyonu için kutu grafiği.	93
Şekil 4.4. Schwefel fonksiyonu için kutu grafiği.	93
Şekil 4.5. Ackley fonksiyonu için kutu grafiği.....	94
Şekil 4.6. Egg Crate fonksiyonu için kutu grafiği.	94
Şekil 4.7. Easom fonksiyonu için kutu grafiği.	95
Şekil 4.8. Optimizasyon aşaması.	96
Şekil 4.9. Sphere fonksiyonunun optimizasyon süreci.	98
Şekil 4.10. Rosenbrock fonksiyonunun optimizasyon süreci.	99
Şekil 4.11. Step fonksiyonunun optimizasyon süreci.	99
Şekil 4.12. Schwefel fonksiyonunun optimizasyon süreci.	100
Şekil 4.13. Rastrigin fonksiyonunun optimizasyon süreci.	100

Şekil	Sayfa
Şekil 4.14. Ackley fonksiyonunun optimizasyon süreci.	101
Şekil 4.15. Griewank fonksiyonunun optimizasyon süreci.	101
Şekil 4.16. Egg Crate fonksiyonunun optimizasyon süreci.....	102
Şekil 4.17. HHO tabanlı MLP eğitimi.....	103
Şekil 4.18. hASO-SA algoritması tabanlı MLP eğitimi.	107



SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamalarıyla birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklama
ABC	Yapay arı algoritması
ACO	Karınca kolonisi optimizasyonu
ASO	Atom arama optimizasyonu
DE	Diferansiyel evrim algoritması
EP	Evrimsel programlama algoritması
ES	Evrimsel strateji algoritması
GA	Genetik algoritma
GWO	Gri kurt algoritması
hASO-SA	Hibrit atom arama-benzetilmiş tavlama algoritması
HHO	Harris şahinleri optimizasyonu
MLP	Çok katmanlı algılayıcı
PSO	Parçacık sürü optimizasyonu
SA	Benzetilmiş tavlama algoritması
SCA	Sinüs kosinüs algoritması
SLP	Tek katmanlı algılayıcı
TS	Tabu arama algoritması
WDO	Rüzgâr güdümlü optimizasyon
YSA	Yapay sinir ağları



1. GİRİŞ

Yapay Sinir Ağları (YSA), yakın zamanda ortaya çıkmış ve birçok gerçek dünya problemini çözmek için farklı disiplinlerce çokça kabul görmüştür. YSA'lar son derece karmaşık fonksiyonları modelleyebilen doğrusal olmayan hesaplama araçlarıdır (Marini ve ark., 2008; Basheer ve Hajmer, 2000). YSA'ların kullanım üstünlükleri doğrusal olmamak, yüksek paralellik, sağlamlık, hata ve başarısızlık toleransı, öğrenme, kesin olmayan ve bulanık bilgileri işleme yeteneği gibi önemli bilgi işleme özelliklerinden ve bunların genelleme kabiliyetinden kaynaklanmaktadır (Jain ve ark., 1996).

Tahminleme, sınıflandırma veya kontrol problemleri olan tüm alanlarda YSA kullanılmaktadır (Marini ve ark., 2008). Bu nedenle finans, tıp, mühendislik, jeoloji, fizik ve kimya gibi farklı alanlarda başarıyla uygulanmaktadır.

Çok Katmanlı Algılayıcı (Multilayer Perceptron-MLP), ileri beslemeli geri yayılma eğitim algoritmasını kullanan YSA uygulamalarında en çok kullanılan modeldir. MLP'nin giriş ve çıkış katmanları arasında gizli bir katmanı vardır; sinirler katmanlar halinde düzenlenir, bağlantılar her zaman alt katmanlardan üst katmanlara yönlendirilir, aynı katmandaki sinirler birbirine bağlı değildir (Lins ve Ludermer, 2005). MLP'ler karmaşık ve doğrusal olmayan modelleme, sınıflandırma, tahmin ve örüntü tanıma gibi çok çeşitli problemlere başarıyla uygulanmıştır (Chaudhuri ve Bhattacharya, 2000; Faris ve ark., 2014; Hamed ve ark., 2014; Lee ve Choeh, 2014; Faris ve ark., 2015).

MLP'ler sınıflandırıcılar olarak yaygın kullanılmaktadır (Xao ve Bilmes, 2006) ve bu tür sinir ağlarının test verilerini sınıflandırmak için kullanmadan önce bir eğitim aşamasından geçilmesi gerekir. MLP eğitim süreci, ara bağlantıların ağırlıklarını ayarlamaktır. Bu işlem hata fonksiyonunun en aza indirilmesi olarak tanımlanabilir (Vishwakarma, 2012; Kawwam ve Mansur, 2012). Bir MLP'yi eğitmek, danışmanlı öğrenme için önemli olan karmaşık bir görevdir ve eğitim sürecinin etkinliğini arttırmak için sezgisel-üstü algoritmalar optimizasyoncu olarak kullanılmıştır (Çelik ve ark., 2019). Sezgisel-üstü arama algoritmaları, rastgeleliğe dayanan olasılıksal algoritmalar (Faris ve ark., 2016). Sezgisel-üstü algoritmalar, çok boyutlu problemlerde etkili arama metotlarıyla çözüm uzayındaki en iyi çözüme daha hızlı

şekilde ulaşmaktadır (Çelik, 2013; Faris ve ark., 2016). Sezgisel-üstü optimizasyon algoritmalarının en önemli avantajlarından bir diğeri yerel optimum noktalara takılmadan global sonuca ulaşabilme yeteneğidir, bu yönüyle Gradyan tabanlı yaklaşımlardan daha üstündürler (Faris ve ark., 2016). Dahası, problem hakkında önceden bilgi gerektirmezler ve genel olarak uygulanması daha kolaydır. Bununla birlikte temel dezavantajları hesaplama süresidir (Laporte ve ark., 2006; Kowalski ve Lukasik, 2015). Birçok evrimsel algoritma ve sürü temelli algoritma MLP'lerin eğitimi için çok çözümlü sezgisel-üstü arama yöntemleri olarak önerilmiştir (Behesti ve ark., 2014). Literatürde MLP ağlarının eğitiminde sezgisel-üstü algoritmalar, ağırlık ve parametre optimizasyonunda kullanılırlar (Azzini ve Tettamanzi, 2011).

Sezgisel-üstü algoritmalar popülerliğine rağmen bazı dezavantajlara da sahiptir. Sezgisel-üstü algoritmaların performansı kendi sınırlamalarından etkilendiği için sahip oldukları zayıflıkların üstesinden gelmek için çeşitli stratejiler önerilmiştir. Hibridizasyon, daha etkili sonuçlar verdiği için önerilen yaklaşımlar arasında mükemmel bir yöntemdir, bu nedenle son zamanlarda bu yöne eğilim artmıştır.

Tezin son kısmında MLP eğitimi için sezgisel-üstü algoritmalar ve sezgisel-üstü algoritmaların hibridizasyon çalışmaları önerilmektedir. Önerilen algoritmaların performansını değerlendirmek için en önemli standart sınıflandırma veri kümelerinden Balon, XOR, İris, Kalp Hastalıkları ve Meme Kanseri veri kümeleri UCI Makine Öğrenme Deposu'ndan seçilmiştir.

Bu çalışmanın amacı sezgisel-üstü algoritmaların üstün performanslarını ve eğitici (supervised) olarak YSA eğitiminde etkinliğini göstermektir. YSA eğitiminde sezgisel-üstü algoritmaların yüksek sınıflandırma becerisi göstermesinin ileride araştırmacıların bu yöntemi daha etkili kullanmalarına neden olacağı düşünülmektedir.

1.1. Sezgisel-Üstü Algoritmalar ve Optimizasyonu

Optimizasyon, en basit anlamda bir minimizasyon veya maksimizasyon problemi olarak düşünülebilir. Optimizasyonda amaç belirli bir sistemin daha düşük maliyetle optimal parametrelere ulaşılması süreci oluşturmaktır. Optimizasyon algoritmalarının geliştirilmesi mühendislik, bilim, ekonomi ve iş gibi çeşitli alanlarda optimizasyon problemleriyle karşılaşılabilindiğinden oldukça dikkat çekmiştir (Rao ve

Desai,1980; Antoniou ve Lu, 2007; Ursayev ve Pardalos, 2013). Gerçek dünyadaki bir optimizasyon problemi, matematiksel olarak formüle edilebilirse çözülebilir. Bu tür problemleri çözmek için çeşitli deterministik algoritmalar mevcuttur. Bununla birlikte bu problemlerin önemli bir kısmı sürekli olmayan ve ayırt edilemeyen doğa, çok fazla karar değişkeni ve amaç fonksiyonlar gibi spesifik özelliklere sahiptir; bu nedenle geleneksel matematiksel programlama yaklaşımları kullanılarak etkin bir şekilde çözülemediğinden (Heidari ve ark., 2019) bu tür problemlerde alternatif yöntemlere başvurulmuştur. Optimizasyon teknikleri, optimum noktanın hesaplanması için amaç fonksiyonun türevlerinin gerekip gerekmediğine bağlı olarak genellikle türevsel ve türevsel olmayan yöntemlere ayrılabilir (Du ve Swamy, 2016). Türev yöntemler, gradyan arama yöntemleri (Salomon, 1998) veya ikinci dereceden yöntemler (Bernard, 1992) olabilen hesap bazlı yöntemlerdir. Bu yöntemler yerel optimize edicilerdir. Örneğin Gradyan İnişi (Amari, 1993) olarak bilinen yöntemde fonksiyon gradyanının negatif yönü boyunca adımlar atarak yerel bir minimum arar. İkinci dereceden yöntemlere örnek olarak Newton (Dembo ve ark, 1982), Gauss-Newton (Wedderburn, 1974) ve Levenberg-Marquardt yöntemleri (Yamashita ve Fukushima, 2001) verilebilir. Türev yöntemler ayrıca model tabanlı ve metrik tabanlı yöntemler şeklinde de sınıflandırılabilir. Model tabanlı yöntemler, yerel bir yaklaşım modeliyle mevcut noktayı geliştirir. Newton yöntemi model tabanlı bir yöntemdir. Metrik tabanlı yöntemler, değişkenlerin dönüşümünü gerçekleştirir ve ardından noktayı iyileştirmek için indirgemeci arama yöntemi uygular. Bu yöntemde En Dik İniş (Fliege ve Svaiter, 2000) ve Yarı Newton (Schraudolph ve ark., 2007) yöntemleri örnek verilebilir.

Bir arama gerçekleştirmek ve sırayla çözüm alanını keşfetmek için gradyan bilgisi gerektirmeyen yöntemlere doğrudan arama yöntemleri denir. Bunlar, bir grup noktayı korurlar. Alanı aramak için bir çeşit deterministik keşif yöntemi kullanırlar ve neredeyse her zaman korunan noktaları güncellemek için açgözlü bir yöntemle başvururlar. Tek yönlü (basit) arama (Fan ve Zahara, 2007) ve desen arama (Vaz ve Vicente, 2007), etkili doğrudan arama yöntemlerine iki örnektir. Çok değişkenli fonksiyonlar için tipik türev gerektirmeyen yöntemler, rastgele yeniden başlama ile tepeye tırmanma (O'neil ve Burtscher, 2015), rastgele arama (Andradóttir,1999) ile birçok sezgisel ve sezgisel-üstü yöntemler ve bunların hibritleridir. Diğer türev gerektirmeyen arama yöntemleri arasında Sıralı Simpleks (Üstün ve ark., 2005), Hooke-

Jeeves (Hooke ve Jeeves, 1961) ve Rosenbrock yöntemi (Shang ve Qui., 2006) gibi doğrudan aramada hızlandırma yöntemleri yer alır. Tüm bu yöntemlerin içerisinde sezgisel-üstü algoritmalar, makul bir sürede tatmin edici çözümler sunarak büyük boyutlu problemleri çözmeyi sağlamalarının (Talbi, 2009) yanı sıra esnek ve basit yapıları, rastgele arama ve yerel optimadan kaçınma yetenekleri sayesinde de alternatif teknikler arasında inanılmaz bir ilgi görmüştür. Bu nedenle sezgisel-üstü algoritmalar daha önce bahsedilen problemleri ele almak için güçlü araçlar oldukları için bu tür problemlerle başa çıkmanın alternatif ve etkili bir yolu olarak kapsamlı bir şekilde incelenmiştir (Das, 2019; Zhang ve ark., 2019; Singh ve ark., 2020; Mohhammed ve Rashid, 2020). Sezgisel-üstü algoritmaların çalışma yöntemleri, etkin bir şekilde yüksek kalitede çözümler üretmek için arama uzayında zeki arama stratejileri uygulamalarıdır. Geçmişte, genellikle sezgisel algoritma olarak adlandırılan stokastik (olasılıksal) mekanizmalara sahip yöntemler, günümüzde yapılan son çalışmalarda, meta ve sezgisel sözcüklerinin birleşimi olan meta-sezgisel olarak adlandırılırlar. Meta üst seviye anlamına gelir, sezgisel yöntem ise yeni stratejiler keşfetme sanatını ifade eder, yani bir meta-sezgisel tarama genellikle “sezgisel-üstü tarama seviyesi” anlamına gelir (Glover, 1986; Talbi, 2009). Genellikle sezgisel-üstü algoritmalar, diğer sezgisel teknikleri, yerel optimum aramada üretilen çözümlerin ötesinde çözümler üretmek için yönlendiren ve değiştiren ana strateji olarak temsil edilir (Glover ve Lagna, 1998).

Optimizasyon problemlerinde sezgisel-üstü algoritmalar çok sayıda uygulanabilir çözüm üzerinde arama yaparak optimizasyon algoritmalarından, iteratif yöntemlerden veya basit sezgisel taramalardan daha az hesaplama çabasıyla iyi çözümler bulabilir. Bu nedenle optimizasyon problemleri için etkili aramalardır. Bu tür algoritmalar arama alanındaki umut verici ve güvenilir alanları doğru zamanda keşfedip tespit edebilir, daha büyük boyutlu problemleri çözebilir ve hesaplama verimliliğini arttırabilir. Ayrıca bu yöntemler optimizasyon işlemine çok sayıda başlangıç çözümleriyle başlar, türev bilgisine bağlı olmaz ve ne amaç fonksiyonları ne de değişkenler için süreklilik şartı gerektirmezler. Bu nedenle bu tür yöntemlerde çözülecek problem, bilinmeyen bir arama alanıyla gerçek problemleri çözmek için çok faydalı bir yol olan kara kutu olarak kabul edilir (Al mufti, 2019).

Sezgisel-üstü algoritmalar, çözüm uzayında küresel ve yerel aramayı dengeli biçimde yaparak yerel optimum noktalara takılmaksızın problemin en iyi çözümüne

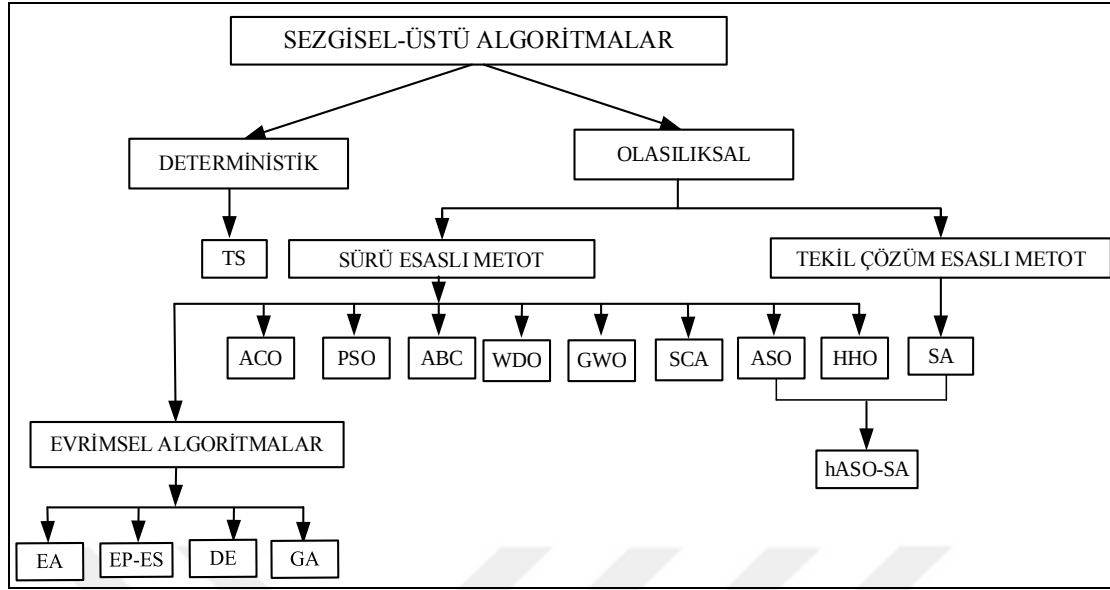
yakınsamak üzere iteratif bir süreç işletmektedirler. Sözü edilen iteratif sürecin gerçekleştirilmesi bir takım yerel ve küresel arama mekanizmaları ile sağlanmaktadır (Cihan ve Fiğlalı, 2018). Sezgisel-üstü algoritmalar en karmaşık problemlerde bile problemin boyutu ne olursa olsun en iyi çözümü üretebilme yeteneğine sahiptirler. Kesikli ya da sürekli, tek amaçlı ya da çok amaçlı, sınırlı (kısıtlı) ya da sınırsız özelliklere sahip geniş bir aralıktaki optimizasyon problemlerin çözümünde rahatlıkla uygulanabilirler (Dokeroğlu ve ark., 2019).

Tek amaçlı optimizasyon problemi, bir adet amaç fonksiyonuna sahipken çok amaçlı bir optimizasyon problemi, her biri bağımsız olarak optimal karara sahip olabilen birden fazla amaç fonksiyonu içerir. Problemin karmaşıklığı amaç fonksiyonunun sayısı ve arama alanının büyüklüğüne bağlıdır (Jain ve ark., 1999).

Sezgisel-üstü algoritmaların çok büyük kısmı sürü zekâsına dayalı sürü temelli bir metot uygular. Sürü terimi, örneklerini yapay alandaki uygulamalar için geliştiren ve sürü zekâsının beş temel ilkesini dile getiren Millonas'ın makalesine uygun kullanılmaktadır (Millonas, 1994). Bu ilkelerden birincisi yakınlık ilkesidir. Bu ilkeye göre sürü, basit alan ve zaman hesaplamaları yapabilmelidir. İkincisi sürünün çevresindeki kalite ilkesidir, bu ilkeye göre sürü çevresindeki kalite faktörlerine cevap verebilmelidir. Üçüncüsü çeşitliliğe cevap verebilme ilkesidir, buna göre sürü davranışlarını geniş bir alan üzerinde yürütebilmelidir. Dördüncüsü istikrar ilkesidir, çevre her değiştiğinde sürü davranış biçimini değiştirmemelidir. Beşincisi uyarlanabilirlik ilkesidir. Bu ilkeye göre sürü, hesaplama maliyetine uygun olarak davranış modunu değiştirebilmelidir (Kenedy ve Eberhart, 1995). Sürü zekası, tek tek bileşenlerin birbirleriyle ve çevreleriyle yerel etkileşimlerinden kaynaklanan, merkezi olmayan ve kendi kendini organize eden sürülerin kolektif davranışları ile ilgilendiği için (Matussta, 1998) sürü temelli optimizasyon teknikleri optimizasyon sürecine bir dizi rastgele çözümle başlar. Bu rastgele küme bir amaç fonksiyonu tarafından iteratif olarak değerlendirilir ve bir optimizasyon tekniğinin çekirdeği olan bir dizi kuralla geliştirilir. Sürü temelli optimizasyon teknikleri, optimizasyon problemlerini olasılıksal olarak optime etmeye çalıştığından tek bir çalıştırma ile çözüm bulma garantisi yoktur. Bununla birlikte yeterli sayıda rastgele çözüm ve optimizasyon adımı (iterasyon) ile küresel optimumu bulma olasılığı artar.

Sezgisel-üstü algoritmaların en önemli yönü, arama ve çeşitlendirme özelliklerini barındıran keşif (exploration/diversification) ve yoğunlaşmaya sahip sömürü (exploitation/intensification) aşamalarından oluşmasıdır (Mirjalili, 2016). Algoritmanın başarısı keşif ile sömürü arasında dinamik dengeyi oluşturması ile orantılıdır (Yang ve ark., 2014). Sömürü, mevcut sorunlardan daha iyi yeni çözümler üretilmesine yardımcı olmak için ilgili problemde elde edilen bilgileri kullanır. Bununla birlikte bu işlem tipik olarak yereldir. Bu nedenle yerel arama içindir. Sömürünün avantajı, genellikle çok yüksek yakınsama oranlarına yol açmasıdır ancak dezavantajı, yerel çözümde sıkışabilmesidir, çünkü nihai çözüm noktası büyük ölçüde başlangıç noktasına bağlıdır. Öte yandan keşif, arama alanını daha verimli bir şekilde taramayı mümkün kılar ve mevcut çözümlerden uzak, yeterli çeşitlilikte çözümler üretebilir. Bu nedenle arama genellikle küresel ölçekte yapılır. Keşfin avantajı, yerel optimumda takılma olasılığının düşük ve küresel tercihliliğin daha erişilebilir olmasıdır. Bununla birlikte dezavantajları yavaş yakınsama ve yerel alanda hesaplama işleminin yarattığı zaman kaybıdır. Bir algoritmanın iyi performans göstermesi için keşif ile sömürü arasında dengenin var olması şarttır. Sömürünün keşiften fazla olması sistemin daha çabuk yakınsayarak gerçek küresel optimayı bulma olasılığını düşürebilir. Optimal denge bir algoritmanın optimal performansı yakalamasını sağlayacak şekilde keşif ile sömürü arasında yeterli düzeyde dengeye sahip olması olarak izah edilebilir (Mirjalili ve ark., 2017).

Yukarıda öne çıkarılan tüm bu özelliklerden dolayı sezgisel-üstü optimizasyon algoritmaları son yirmi yılda en popüler araştırma alanı haline gelmiştir (Talbi, 2009). Sezgisel-üstü algoritmaları, Ghosh ve arkadaşlarının şemasından esinlenerek Şekil 1.1. deki gibi güncellenebilir (Ghosh ve ark., 2011; Khalilpourazari ve Khalilpourazari, 2019).



Şekil 1.1. Sezgisel-üstü algoritmaların sınıflandırılması.

1.1.1. Deterministik algoritmalar

1.1.1.1. Tabu arama algoritması

Tabu Arama (Tabu Search, TS) algoritması, belirli hareketleri cezalandıran bir strateji ile yerel optimumu aşmayı sağlar. Yasak bir hamleyi sınıflandırma amacı, arama bölgesinde bir kez incelenen nokta tabu olarak tanımlandığında algoritma tarafından tekrar incelenmeden devam edilmesi böylece aynı alanda kısır döngüyü engellemek içindir (Glover, 1986). TS'nin bu karakteristik özelliği, yerel en uygun çözümde sıkışmayı önleme ve arama geçmişinin esnek belleğini kullanarak çevrimi önleme özelliğidir. TS, deterministik kurallar yerine karar vermek için olasılıksal yöntemin ön gördüğü kuralları kullanır. Bu nedenle TS, küresel optimumu bulmak için karmaşık ve belirsiz bir alanı araştırabilen aynı zamanda bir tür olasılıksal optimizasyon algoritmasıdır. Bu, TS'yi geleneksel yöntemlerden daha esnek ve sağlam hale getirir. TS, problem alanındaki aramaya rehberlik etmek için amaç fonksiyonu bilgilerini kullanır. Bu nedenle gerçek hayattaki optimizasyon problemleri olan diferansiyellenemeyen, sürekli olmayan ve ayırt edilemeyen amaç fonksiyonlarıyla kolayca başa çıkabilir. Ek olarak bu özellik, TS'yi birçok pratik optimizasyon problemi için geleneksel optimizasyon yöntemlerinin gerektirdiği varsayım ve tahminlerden

kurtarır (Abido, 2002). Bu açıklamalar altında TS, x bir parametre vektörü olmak üzere, $G(x)$ fonksiyonunu optimize etmeye çalışan, x parametre uzayını en uygun çözüm için iteratif olarak tarayan bir optimizasyon tekniğidir (Hageman ve ark., 2002).

TS algoritmasının temel aşamaları (Abido, 2002) aşağıda ifade edilmiştir:

Mevcut çözüm (x_{mev}): Herhangi bir iterasyonda optimize edilmiş parametre değerleri kümesidir. Komşu deneme çözümlerinin üretilmesinde merkezi bir rol oynamaktadır.

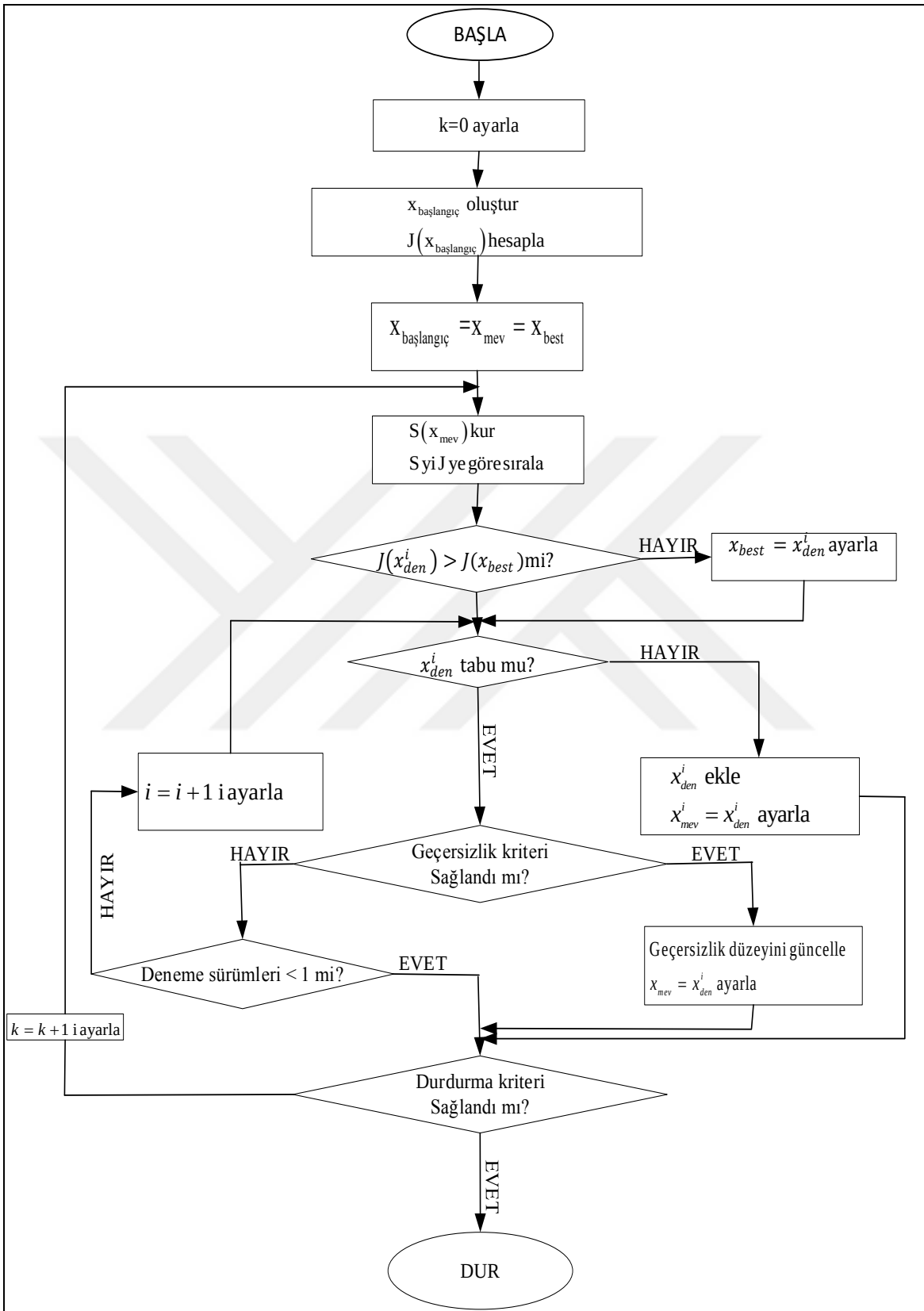
Hamleler: Deneme çözümleri (x_{den}) üretme sürecini karakterize ederler.

Aday hamle kümesi ($N(x_{mev})$): x_{mev} komşuluğundaki tüm olası aday çözümleri, x_{den} 's kümesinde yer alır. Sürekli değişkene sahip optimizasyon problemleri olması durumunda, bu çok büyük hatta sonsuz bir kümedir. Bu nedenle, bu kümenin sınırlı sayıda deneme çözümü (nt), $S \subset N$ ve $x_{mev} \in S(x_{mev})$ alt kümesi ile çalışabilir.

Tabu kısıtlamaları: Bunlar, belli koşullarda bazı hareketleri yasaklayıcı hamlelerdir. Bu yasak hareketler belirli bir boyuta listelenir ve tabu olarak bilinir. Bu listeye tabu listesi denir. Belli bir hareketi yasak olarak sınıflandırmanın ardındaki neden, temel olarak çevrimi önlemek ve henüz ziyaret edilen yerel optimuma geri dönmekten kaçınmaktır. Tabu liste boyutu, yüksek kaliteli çözümlerin aranmasında büyük rol oynar. İyi bir tabu listesi boyutunu tanımlamanın yolu, boyut çok küçük olduğunda çevrimin oluşumunu ve boyut çok büyük olduğunda çözüm kalitesindeki bozulmayı, çok fazla hareketin yasaklanması nedeniyle izlemektir.

Geçersizlik Kriteri (Seviye): Eğer tabu kısıtlaması ile belirli bir hareket yasaklanmışsa geçersizlik kriteri tatmin edildiğinde bu harekete izin verilebilir. Literatürde farklı geçersizlik kriterleri kullanılmaktadır. Burada düşünülen, eğer bu hamle, aynı hamleyle daha önce elde edilene göre daha iyi amaç fonksiyonuna (J) sahip bir çözüm üretirse, bir hamlenin tabu durumunu geçersiz kılmaktır. Geçersizlik kriterini kullanmanın önemi, TS'yi cazip hareketlere yönlendirerek esneklik katmaktır.

Durdurma Kriterleri: Bunlar, arama işleminin sona ereceği koşullardır. En iyi çözümün son değişikliğinden dolayı iterasyon sayısı önceden belirlenmiş bir sayıdan daha fazla olduğunda ya da iterasyon sayısı izin verilen maksimum sayıya ulaştığında algoritma durdurulur. Şekil 1.2'de Tabu arama algoritmasının akış şeması yer almaktadır (Abido, 2002):

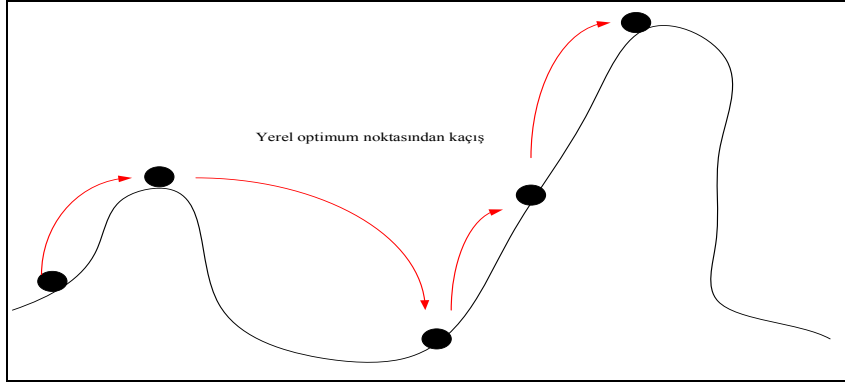


Şekil 1.2. Tabu arama algoritması akış şeması.

1.1.2. Olasılıksal algoritmalar

1.1.2.1. Benzetilmiş tavlama algoritması

Benzetilmiş Tavlama (Simulated Annealing, SA) algoritması, küresel optimizasyon problemlerinde kullanılan güncel özgün bir olasılıksal algoritmadır (Kirkpatrick ve ark., 1983). Adı ve ilham kaynağı metalurjide tavlamadan kaynaklanır. Tavlama, bir malzemenin ısıtılmasını ve kontrollü soğutulmasını içeren bir işlemdir. Atomlar, ısıtılarak başlangıç konumlarından çıkarılır ve yüksek enerjili hallerde rastgele dolaşırlar. Daha düşük iç enerjili konfigürasyonları başlangıçta olduğundan daha fazla bulma şansı yavaş soğutma ile sağlanır. Bu fiziksel süreçle benzer şekilde, SA’da her uygulanabilir çözüm, bir fiziksel sistemin durumuna benzerdir ve en aza indirilmesi gereken uygunluk fonksiyonu, bu durumda sistemin iç enerjisine benzer. Nihai hedef, sistemi rastgele bir başlangıç durumundan sistemin enerjisinin minimum düzeyde olduğu bir duruma getirmektir. Her aşamada SA, mevcut çözümü hem ilgili uygunluk değerleri arasındaki farklılığa hem de sıcaklık olarak adlandırılan bir parametreye bağlı olarak olasılıkla yakın rastgele bir çözümle değiştirir. SA’nın ana ayrıcalığı, metropol süreci denilen yerel optimumdan kaçabilmesidir. SA, Metropolis kuralına dayanan mevcut çözümünden daha kötü bir çözümü kabul ettiği anlamına gelen rastgele bir arama algoritmasıdır, bu nedenle küresel optimum çözümün elde edilmesini kolaylaştıracak yerel optimum çözümünden vazgeçmesi mümkündür (Xie ve ark., 2020). Şekil 1.3’te gösterildiği gibi Metropolis sürecini kabul etme olasılığını kullanarak en uygun çözüm olan en düşük enerji durumunu bulmada tavlama mekanizmasını simüle edebilir:



Şekil 1.3. Metropolis kuralı ile yerel optimadan kaçış.

SA'nın ana parametreleri başlangıç koşulu, metropol kabul mekanizması ve soğutma çizelgeleme yöntemidir (Askerzadeh ve ark., 2016).

Başlangıç Koşulu: SA algoritmasının arama işleminin başlangıcındaki başlangıç koşulu, performans üzerinde önemli bir etkisi olan başlangıç sıcaklığının (T_0) başlatılması ile start alır. Daha yüksek T_0 değeri optimizasyon işleminin hesaplama süresini artıracaktır. Aksine, eğer T_0 çok düşükse SA'nın arama alanını etkin bir şekilde araştırmamasına neden olacaktır. Bu nedenle en uygun çözümü elde etmek için dikkatlice seçilmesi gerekir. Aşağıdaki eşitlikte T_0 başlangıç sıcaklığı hesaplanmıştır:

$$T_0 = -\Delta T_0 / \ln X_0 \quad (1.1)$$

Yukarıda verilen ifadede İlk sıcaklık T_0 , yukarı ve aşağı geçiş olasılıklarını sağlayacak kadar büyük olmalıdır. Bunun için rastgele bir durum ve komşuları için bir ΔS farkı tahmininde bulunulmalıdır. Başlangıç değişimi, ΔS_0 minimizasyon için en kötü sonuç ile optimal çözüm arasındaki fark ve X_0 ise en kötü çözüm için başlangıç kabul oranıdır (Xinchao, 2011).

Metropolis onay mekanizması: Metropolis onay mekanizması hızlı bir şekilde en uygun çözümün elde edilmesini etkileyen anahtarlardan biridir. Metropol kuralları, uygunluk değeri düşük olan yakındaki çözümün mevcut çözüm olarak kabul edilip edilmediğini göstermek için kullanılır. Bu mekanizma, SA'nın yerel optimum koşullardan kaçma kabiliyetini etkiler. X'_n , X_n den daha iyi olduğunda SA, komşu çözümü mevcut çözüm olarak düşünmeyi ya da düşünmeyeceğini belirtmek için kabul

olasılık mekanizmasını kullanacaktır. Onay mekanizmasının olasılığı aşğıdaki eşitlikte verilmiştir (Askerzadeh ve ark., 2016):

$$X_{n+1} = \begin{cases} X'_n, \exp(\Delta S/T_k) > \rho \\ X_n, \text{diğer durumlar} \end{cases} \quad (1.2)$$

$\rho, (0,1)$ aralığında rastgele bir sayıdır ve yeni çözüm için olasılıksal bir karar vermek için kullanılır.

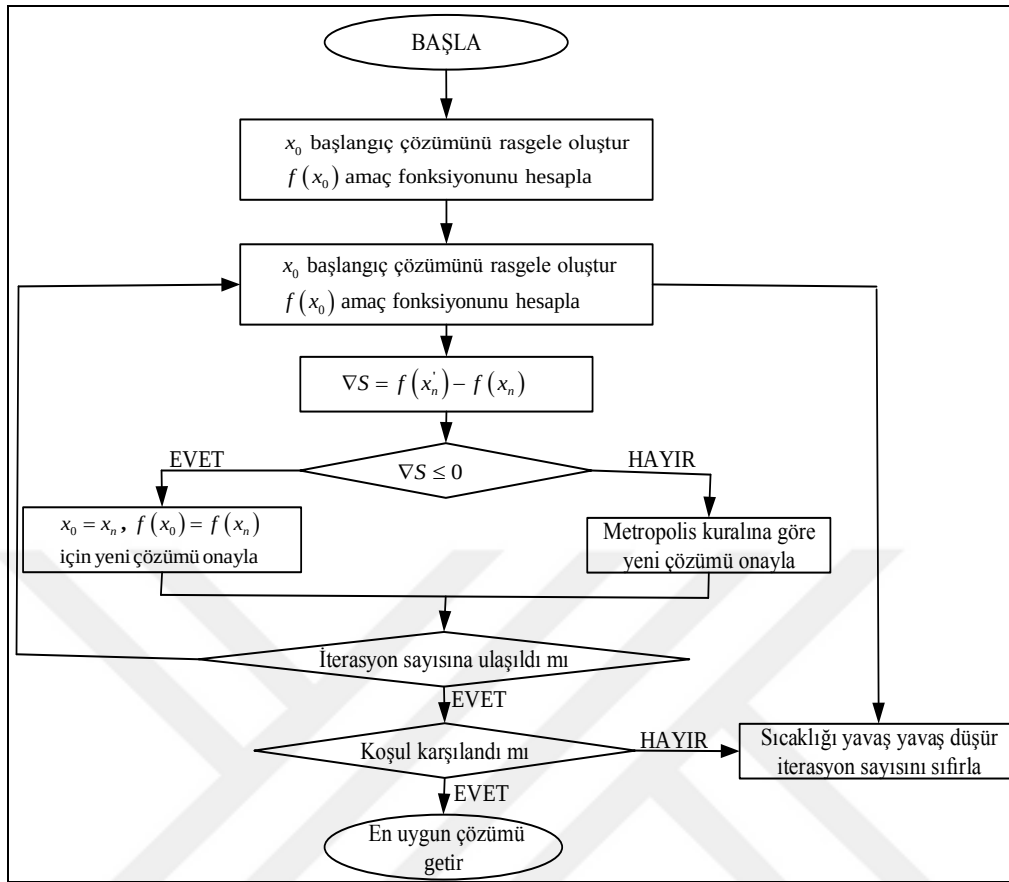
$$\nabla S = f(X'_n) - f(X_n) \quad (1.3)$$

ve $f(.)$ objektif (amaç) fonksiyonudur.

Soğutma çizelgeleme yöntemi: Fiziksel tavlama işlemi sırasında, sıcaklık azaldıkça, soğutma hızı da sabit zemin durumuna ulaşmak için azalır. Bunun için sistemin başlangıçta sıcaklığın yavaş yavaş azalmasıyla daha hızlı ve daha sonra yavaş bir şekilde soğutulması gerekir. Soğutma programı parametresi (σ), aşğıdaki eşitlikte ifade edilmiştir (Askerzadeh ve ark., 2016)

$$\sigma: T_{n+1} = \alpha T_n \quad (1.4)$$

verilen eşitlikte T_n , n . iterasyonda, bu optimizasyon işleminde belirli bir soğutma fonksiyonu ile soğutulan sıcaklık seviyesidir. α , 0.9 ile 0.99 aralığında bulunan sabit bir değerdir. Sıcaklık αT_n azaldıkça, çözünürlük kalitesinde büyük bir düşüşü kabul etme olasılığı Boltzman dağılımına (Probst ve Rothlauf, 2015) göre üssel olarak sıfıra düşer. Bu nedenle sıcaklık sıfıra yaklaştığında, nihai çözünürlük optimum olur (Xinchao, 2011). Şekil 1.4'te Benzetilmiş tavlama algoritmasının akış şeması yer almaktadır (Zhou ve ark., 2018).



Şekil 1.4. SA akış şeması.

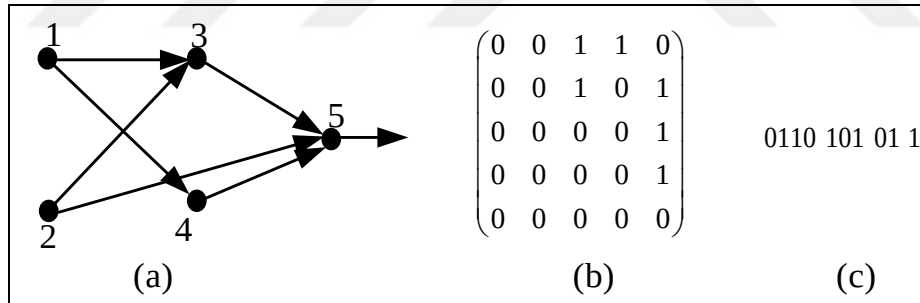
1.1.2.2. Evrimsel algoritmalar

Evrimsel Algoritmalar (Evolutionary Algorithm, EA)'ın dayandığı temel fikir, rekabete dayalı en güçlü bireyin hayatta kalmasını sağlayan doğal seçimdir. EA, evrim (Darwin, 1929) fikrinden geliştirilen sürü tabanlı olasılıksal arama algoritmasıdır. Evrimsel Programlama (Evolutionary Programming, EP) (Fogel, 1995), Evrim Stratejileri (Evolutionary Strategies, ES) (Rechenberg, 1973; Schwefel, 1995), Diferansiyel Evrim (Differential Evolution, DE) (Storn ve price, 1997) ve Genetik Algoritma (Genetic Algorithm, GA) (Goldberg, 1989) gibi algoritmaları içerir. GA kromozomal operatörleri vurgularken, ES birey düzeyindeki davranış değişikliklerini, EP ise tür düzeyinde davranış değişikliğini vurgular (Fogel, 1994).

EA için rastgele bir dizi aday çözüm oluşturulur, kalite/uygunluk (fitness) fonksiyonu belirtilir, çaprazlama (recombination/crossover) ya da değişim (mutation) uygulanarak gelecek neslin üretilmesi için iyi adayların bazıları seçilir. Çaprazlama, bir

veya daha fazla yeni aday üreten, iki veya daha fazla seçilmiş adaya uygulanan bir işlemdir. Değişim işlemi bir adaya uygulanır ve yeni bir adayla sonuçlanır. Bu nedenle ebeveynler üzerinde Çaprazlama veya değişim işlemlerinin yürütülmesi, bir dizi yeni adayın oluşturulmasına yol açar. Bunların uygunlukları değerlendirilir ve daha sonra uygunluklarına dayanarak gelecek nesilde yer bulmak için eskileriyle rekabet eder. Bu süreç, yeterli kalitede bir aday bulunana veya önceden belirlenmiş bir hesaplama sınırına ulaşılan kadar iteratif olarak devam eder. Çaprazlama ve değişim işlemleri sürüde çeşitliliği yaratarak yeniliği kolaylaştırırken seçim de sürüde ortalama çözüm kalitesini arttıran bir güç görevi görür (Eiben ve Smith, 2015).

EA, fuzzy kontrol, yol planlama, modelleme, sınıflandırma, parametre sayısının fazla olduğu ve analitik çözümlerin elde edilmesinin zor olduğu karmaşık optimizasyon problemlerinin çözümlerinde kullanılır (Azzini ve Tettamanzi, 2011). EA keşif yönü güçlü sezgisel-üstü bir algoritma olarak her bir iterasyonda olası çözümlerin kümesini korur ve güncellemesini sağlar. Şekil 1.5'te bir EA yoluyla YSA eğitiminin nasıl yapıldığına dair kısa bir çalışma verilmiştir (Yao, 1999).



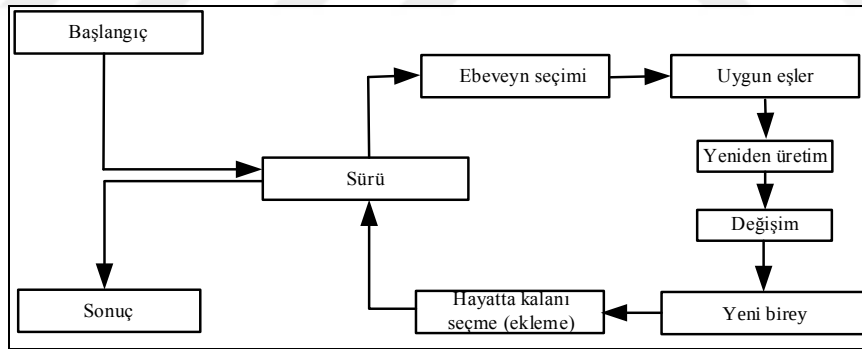
Şekil 1.5. İleri beslemeli bir YSA'nın kodlama örneği.

Şekil 1.5'te (a) mimari yapısı, iki giriş ve bir çıkış ile ileri beslemeli bir YSA'yı göstermektedir. (b) bağlantı matrisini verir, burada girdi düğümünden düğüme bir bağlantının varlığını veya yokluğunu gösterir. Örneğin ilk satır düğüm 1'den diğer tüm düğümlere bağlantıları gösterir. İlk iki sütun 0'dır, çünkü düğüm 1'den kendisine bağlantı yoktur ve düğüm 2'ye bağlantı bulunmamaktadır. Ancak düğüm 1 düğüm 3 ve 4'e bağlanır. Dolayısıyla sütun 3 ve 4'te 1'ler bulunur. Bu bağlantı matrisini bir kromozoma dönüştürmek kolaydır. Tüm satırları (veya sütunları) birleştirebilir ve 00110 00101 00001 00001 00000 elde edilebilir. YSA ileri beslendiğinden kromozom

uzunluğunu azaltmak için sadece bağlantı matrisinin üst üçgenini temsil etmemiz gerekir. (c) ikili dizi gösterimini verir. Burada ileri beslemeli mimariler dikkate alındığından ikili dizi gösterimi için kare matrisin sağ üst üçgeninin dikkate alınması gereklidir. Anlam olarak ikili dizi gösterimi indirgenmiş kromozomu gösterir. Daha sonra bu tür kromozomların bir sürüsünü geliştirmek için bir EA kullanılabilir. Her kromozomun uygunluğunu değerlendirmek için, bir kromozomu YSA'ya dönüştürmemiz, rastgele ağırlıklarla başlatmamız ve eğitmemiz gerekir. Ortaya çıkan hata miktarı, uygunluğu ölçmek için kullanılacaktır (Yao, 1999). Bir öğrenme kuralı yerel değişkenlerin ve sonuçlarının lineer bir fonksiyonu olarak kabul edilir. Aşağıdaki eşitlikte ANN için bir öğrenme kuralı tanımlanmıştır (Yao, 1993).

$$\Delta w(t) = \sum_{k=1}^n \sum_{i_1, i_2, \dots, i_k=1}^n (\theta_{i_1, i_2, \dots, i_k} \prod_{j=1}^k x_{ij}(t-1)) \quad (1.5)$$

Yukarıdaki eşitlikte t zaman, Δw ağırlıktaki değişimi, $x_1, x_2, \dots, x_n$ yerel değişkenleri ve θ evrim tarafından belirlenebilen gerçek değerli katsayıyı gösterir. Şekil 1.6'da bir EA'nın işlem aşamaları verilmiştir (Eiben ve Smith, 2015).



Şekil 1.6. EA akış şeması.

1.1.2.3. Genetik algoritma

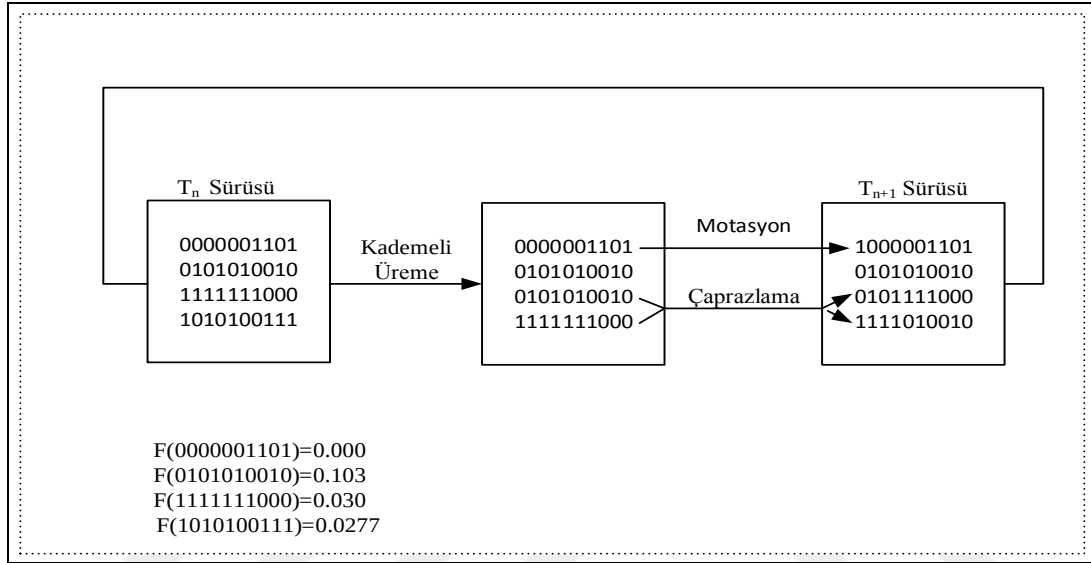
Genetik Algoritma (Genetic Algorithm, GA), tarihte ilk ortaya atılan sürü zekâsına dayalı olasılıksal algoritmalarından biridir. GA, Darwin'in evrim teorisinde "Güçlü bireyler hayatlarını devam ettirirler (doğal seleksiyon)" savından hareketle hayatını devam ettiren daha güçlü bireyin genlerinin taklit edildiği sayısal bir modeldir. (Holland, 1992a; Holland, 1992b).

GA'da her çözüm bir kromozoma denk gelir ve her bir parametre bir geni temsil eder. GA, seçim, çaprazlama ve değişim adlı üç ana operatörden oluşur. GA, bir uygunluk (objektif) fonksiyonu kullanarak sürüdeki her bir bireyin uygunluğunu değerlendirir. Zayıf çözümleri iyileştirmek için, en iyi çözümler rulet tekerleği (He ve ark., 2013) gibi rastgele bir seçim mekanizmasıyla seçilmiştir. Bu operatör uygunluk (objektif) değeri ile orantılı olduğu için en iyi çözümleri seçme olasılığı daha yüksektir. Yerel optimadan kaçınmayı arttıran neden aynı zamanda kötü çözümlerin seçilip onlardan kaçınılmasıdır ki bu yerelde en iyi çözümleri bulmak ve diğer çözümlerden kaçınmak anlamına gelir. Bu algoritmanın güvenilirliği de global olarak en iyi değeri tahmin etmesi ve her iterasyonda en iyi çözümleri sağlama ve diğer çözümleri geliştirmek için kullanma sürecidir (Mirjalili, 2019). İlk olarak, bir grup birey rastgele oluşturulur. En basit durumda, her birey bir byte dizisidir ve bazı problemler için aday bir çözüm olarak düşünülebilir. Sürüdeki bireyler arasındaki farklılıklar, bazı bireylerin diğerlerinden daha uygun olmalarına neden olur. Bu farklılıklar, bir sonraki adımda, seçim olarak adlandırılan yeni bir aday çözüm kümesinin seçimine ağırlık vermek için kullanılır. Seçim sırasında, daha başarılı kişilerin kopyalarını alarak ve daha az başarılı olanları silerek yeni bir sürü yaratılır. Kopyalama işlemi sırasında değişim, çaprazlama gibi olasılıksal süreçler vardır. Önceki iyi birey kümesini yenisine dönüştürerek değişim ve çaprazlama işlemleri, ideal olarak ortalama iyi olma şansından daha iyi olan yeni bir grup birey üretir. Bu değerlendirme döngüsü, seçim ve genetik işlemler döngüsü iteratif olarak devam ettiğinde, sürü genel olarak en iyiye yaklaşır ve sürüdeki bireyler “çözümler” i temsil eder. Aşağıda bu konuyla ilgili en yaygın örnek Holland tarafından oluşturuldu (Holland, 1975; Goldberg, 1989; Forrest, 1993). Şekil 1.7’de, $F(x, y)$ fonksiyonunu optimize etmek için kullanılan bir GA örneği gösterilmektedir. Dört kişilik bir sürüye, aşağıdaki eşitlikte belirtilen fonksiyona göre uygun bir değer atanır

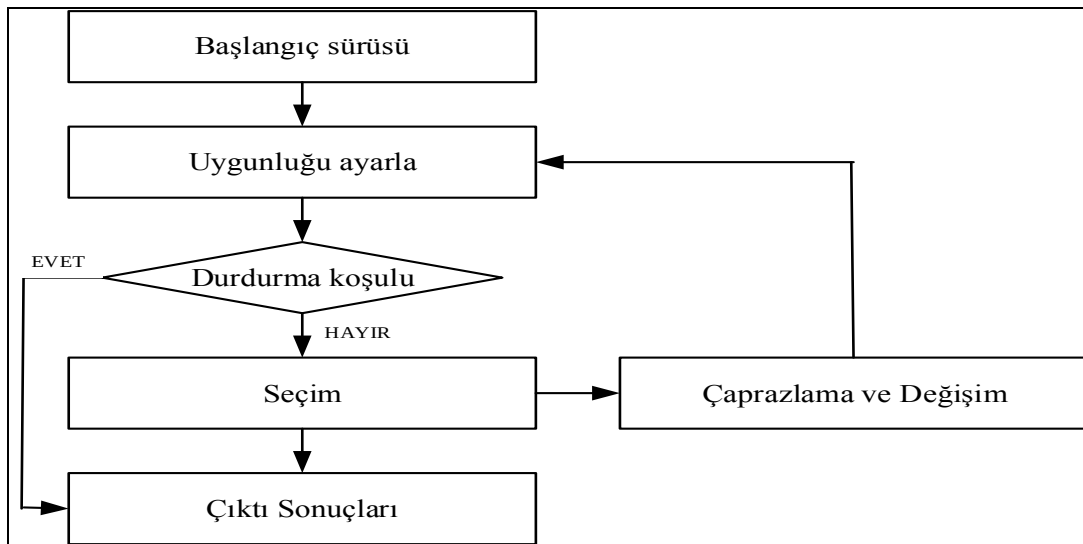
$$F(x, y) = yx^2 - x^4 \quad (1.6)$$

Şekil 1.7’de GA algoritmasının tüm aşamaları gösterilmiştir. T_n sürüsü için en uygun değer her bir birey için yer almaktadır. İlk beş byte x 'i ve sonraki beş byte y 'yi temsil eden $[0,1)$ aralığında normalize edilmiş değerlerdir. Bu uygun değerler temelinde seçim aşaması ilk bireyi (000001101) bir , ikinciyi (0101010010) iki, üçüncüyü

(1111111000) bir ve dördüncüyü (1010100111) sıfır kopya olarak atar. Seçimden sonra genetik operatörler olasılıksal olarak uygulanır. İlk birey ilk bytei 0'dan 1'e değişime uğratar ve son ikiyi, iki yenisiyle çapraz birleştirir. Sürü sonucu şekilde T_{n+1} etiketli kutuda gösterilmiştir (Forrest, 1996). Ayrıca Şekil 1.8'de de GA'nın akış şeması gösterilmiştir (Abdeslam ve ark., 2014).



Şekil 1.7. GA genel bakış.



Şekil 1.8. GA Akış diyagramı.

1.1.2.4. Evrimsel programlama ve evrimsel strateji algoritmaları

Evrimsel Programlama (Evolution Programming, EP) algoritması, Fogel tarafından (Fogel ve ark., 1966; Fogel ve ark., 1994) ilk kez sonlu-durum makinalarının (Chow, 1978) evrimi için önerilmiştir. EP, gerçek değerli nesne değişkenlerini veya diğer herhangi bir uygun veri yapısını optimize etmek için genişletilmiştir. EA'lar doğal seleksiyon, varyasyon ve genetiğin simüle edilmiş evrimsel sürecine dayanan arama algoritmalarıdır. EA'lar klasik hesaplama yöntemlerine göre daha esnektir ve sağlamdır. EP, geleneksel GA'lardan iki açıdan farklıdır: EP, kontrol parametrelerini (gerçek değerler) kullanır, ancak kodları geleneksel GA'larda olduğu gibi kullanmaz ve EP, geleneksel GA'larda olduğu gibi öncelikle değişime ve seçime dayanır, ancak çaprazlamaya dayanmaz. Bu nedenle EP'de hatırı sayılır derecede uzun bir hesaplama süresi kaydedilebilir (Sinha ve ark., 2003). Her iterasyon sırasında, bir değişim operatörü kullanılarak mevcut bir sürüden ikinci bir yeni sürü oluşturulur. Bu operatör, mevcut bir çözümün her bir bileşenini rastgele bir miktarda tutarak yeni bir çözüm üretir. Aday çözümlerin her birinin en uygunluk (optimum) derecesi, problemin amaç fonksiyonu olarak tanımlanabilecek fonksiyonun uygun sonuçları ile ölçülür. Bir rekabet şemasının kullanılmasıyla her sürüdeki bireyler birbirleriyle rekabet eder. Kazanan bireyler, gelecek nesil olarak kabul edilen yeni bir sürü oluşturur. Optimizasyonun gerçekleşmesi için rekabet şeması, en uygun çözümlerin daha zayıf çözümlerden daha fazla hayatta kalma şansına sahip olacağı şekilde olmalıdır. Bu sayede sürü küresel optimal noktaya doğru evrilir (Yuryevich ve Wong, 1999). EP algoritmasının optimizasyon süreci aşağıdaki aşamalardan oluşur (Lee ve Yang, 1998).

Başlangıç: Kontrol değişkenlerinin ilk sürüsü, üst ve alt sınırları arasında değişen eşit dağılımlı kontrol değişkenleri kümesinden rastgele seçilerek amaç fonksiyonu aracılığıyla kontrol puanı (f) elde edilir.

İstatistikler: Mevcut neslin maximum uygunluğu (f_{max}), minimum uygunluğu (f_{min}), uygunluklarının toplamı ($\sum f$) ve ortalama uygunluğu (f_{avg}) hesaplanır.

Değişme (Mutasyon): Her seçilen ebeveyn (P_i), değiştirilir ve aşağıda verilen eşitlikte ifade edilen kurala göre sürüye eklenir.

$$P_{i+m,j} = P_{i,j} + N\left(0, \beta(\bar{x}_j - \underline{x}_j) \frac{f_i}{f_{max}} \mid j = 1, 2, \dots, n\right) \quad (1.7)$$

Bu eşitlikte n bir bireydeki karar değişkenlerinin sayısı, $P_{i,j}$ i . bireyin j . değişkenini, $N(\pi, \sigma^2)$ ortalaması (π) ve varyansı (σ^2) olan bir gauss değişkenini temsil eder. f_{max} istatistiklerde elde edilen eski neslin maksimum uygunluğu, $\bar{x}_j$ ve x_j sırasıyla j . değişkenin alt ve üst sınırlarını, β , nesiller boyunca uyarlanabilir bir şekilde azaltılabilen değişim aralığını ifade eder ve $\beta \in [0,1]$ dir. Herhangi bir değişime uğramış değer, sınırını aşarsa yeni bir sınır değeri verilir. Değişim süreci, daha büyük uygunluğa sahip bir bireyin gelecek nesil için daha fazla yavru üretmesine izin verir.

Rekabet: En iyi uygunluğa sahip birkaç birey (k), gelecek nesil için ebeveyn olarak tutulmaktadır. Seçimin yapıldığı sürüdeki ($2m-k$) diğer bireyler, gelecek nesil de yer alma şanslarını elde etmek için birbirleriyle rekabet etmek zorundadır. Aşağıda verilen eşitlikte i . bireyin ağırlık değeri

$$W_i = \sum_{t=1}^N W_{i,t} \quad (1.8)$$

ile hesaplanır. Burada N , rastgele üretilen yarışma numarası, $W_{i,t}$, sürü için kayıp 0, kazanma için 1'dir, çünkü i . birey, seçimin yapıldığı sürüde rastgele seçilmiş r . bir bireyle rekabet eder. Aşağıdaki eşitlikte $W_{i,t}$ değeri verilmiştir.

$$W_{i,j} = \begin{cases} 1 & U_1 < \frac{f_r}{f_r + f_i} \\ 0, & \text{diğer durumlarda} \end{cases} \quad (1.9)$$

burada rastgele seçilen r . bireyin uygunluğu f_r , i . bireyin uygunluğu ise f_i 'dir. U_1 düzensiz dağılım $U(0,1)$ kümesinden rastgele seçilen bir değerdir. Tüm $2m$ bireyler, rekabet ağırlıklarını aldıklarında, karşılık gelen W_i değerlerine göre azalan bir sıralamada sıralanırlar. İlk m bireyler, gelecek kuşakların temelleri olacak şekilde f_i uygunlukları ile seçilir. Mevcut neslin maksimum, minimum ve ortalama uygunluk değerleri ve uygunluk değerleri toplamı istatistikler aşamasında hesaplanır.

Yakınsama testi: Yakınsama koşulu yerine getirilmezse değişim ve rekabet süreçleri tekrar çalışır. Maksimum nesil sayısı yakınsama koşulu için kullanılabilir. Sürünün maximum uygunluğu ve ortalama uygunluğu gibi diğer kriterler aşağıda ifade edilen eşitlikte olduğu gibi hesaplanır.

$$\frac{f_{avg}}{f_{max}} \geq \delta \quad (1.10)$$

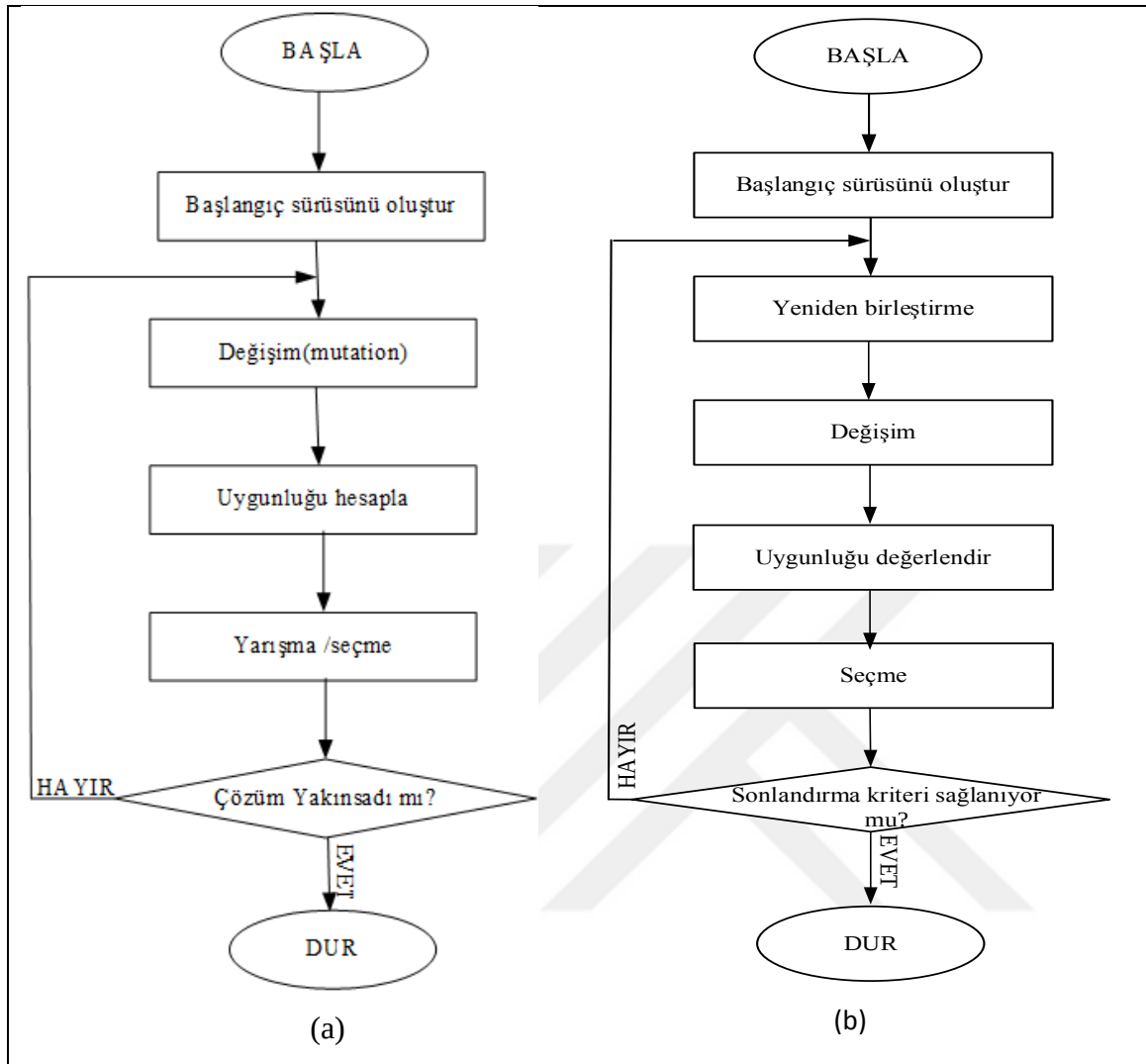
Burada memnuniyet derecesini ifade eden δ değeri 1'e çok yakın olmalıdır. Yakınsama belirli bir hassasiyete ulaşmışsa bir optimizasyon problemi için en uygun çözüm bulunmuştur.

Evrin Stratejileri (Evolution Strategies, ES) algoritması, EP'de olduğu gibi, ES'ler de gerçek değerli nesne değişkenlerine ve sifıra en yakın noktaya yaklaşma beklentisiyle normal olarak dağıtılmış rastgele değişikliklere dayanır (Fogel, 1994).

ES algoritması EP ile aynı işlem aşamalarına sahip olmakla beraber ES algoritmasını EP' den ayıran tek faktör değişim (mutation) sürecinde her seçilen ebeveynin, örneğin P_i olarak ifade edilirse, değişime uğramış ve aşağıda verilen eşitlikte belirtilen kuralın ardından sürüye eklenmiş olmasıdır (Lee ve Yang, 1998).

$$P_{i+m,j} = P_{i,j} + N(0, \beta \nabla_{dev}), j = 1, 2, \dots, n \quad (1.11)$$

Verilen eşitlikte, n karar değişkenlerinin sayısı, $P_{i,j}$ i . bireyin j . değişkenini, $N(0, \beta \nabla_{dev})$ bir Gaus değişkeninin ortalama (0) ve varyansını ($\beta \nabla_{dev}$) temsil eder. β , nesiller boyunca uyarlanabilir bir şekilde azaltılabilen değişim aralığını ifade eder ve $\beta \in [0,1]$ dir. ∇_{dev} bir sabit ve değeri karar değişkenlerinin değer büyüklüklerine bağlıdır. Şekil 1.9.a ve 1.9.b'de Evrimsel Programlama (Musirin ve ark., 2008) ve Evrimsel Strateji algoritmaları (Khan, Ghalban, 2008) akış şeması gösterilmiştir.



Şekil 1.9. (a) EP akış şeması, (b) ES akış şeması.

1.1.2.5. Diferansiyel evrim algoritması

Diferansiyel Evrim (Differential Evolution, DE) algoritması Price ve Storn tarafından önerilmiştir (Storn ve Price, 1997). DE sürü temelli sezgisel-üstü algoritmadır. Tıpkı genetik algortmada olduğu gibi değişim (mutation) ve crossover (çaprazlama) işlemlerine sahiptir. Bu operatörler sırasıyla DE sürecinde de kullanılır. Tüm bu işlemler sürüdeki tüm kromozomlara uygulanır ve yeni bir kromozom elde edilir. Değişim ve çaprazlama işlemleri kullanıldıktan sonra, ilgili kromozom ve yeni kromozom belirlenmiş bir uygunluk değeri ile karşılaştırılır. Daha iyi uygunluk

değerine sahip kromozom, amaç fonksiyonuna göre yeni bir nesle aktarılır. En iyi kromozom, DE işleminin sonunda optimal çözüm olarak alınır (Bas, 2016).

Tüm evrim algoritmaları değişim, çaprazlama ve seçim tekniklerini kullanarak çözüm geliştirirler. $f(x_1, x_2, \dots, x_n)$, $x_j^L \leq x_j \leq x_j^U, j = 1, 2, \dots, n$ olmak üzere f amaç fonksiyonunun minimize edilmesi için aşağıdaki adımlar takip edilmelidir (Mingguang ve Gaoyang, 2009). Aşağıdaki eşitlikte ifade edilen her bir vektör optimizasyon problemi için bir çözümü temsil eder.

$$X_i(t) = (X_{i1}(t), X_{i2}(t), \dots, X_{in}(t)) \quad (1.12)$$

eşitlikte verilen n , toplam birey sayısını ifade eden kromozomun uzunluğunu, NP sürü büyüklüğünü ve t_{max} maksimum iterasyon sayısını ifade eder.

Başlangıç sürüsü oluşturmak: DE'nin ilk adımı başlangıç sürüsü oluşturmaktır. x_{ij}^U, x_{ij}^L sürüde j . değişkenin sırasıyla üst ve alt sınırı olsun. $rand_{ij}(0,1)$, $[0,1]$ kapalı aralığında rastgele bir değeri ifade etmek üzere n boyutlu arama uzayında aşağıdaki eşitlikte ifade edildiği gibi rastgele bir başlangıç sürüsü oluşturulur.

$$x_{ij}(0) = rand_{ij}(0,1)(x_{ij}^U - x_{ij}^L) + x_{ij}^L \quad (1.13)$$

Değişim işlemi: DE, x_{p_2}, x_{p_3} ($p_2 \neq p_3$) şeklinde iki sürüyü rastgele seçer, diferansiyelini kullanır daha sonra eşitlik olarak x_{p_1} 'i değiştirmek için η ölçeklendirme faktörünü kullanır.

$$h_{ij}(t+1) = x_{p_1j}(t) + \eta (x_{p_2j}(t) - x_{p_3j}(t)) \quad (1.14)$$

eşitliğinde η , $(0,2)$ aralığında ölçeklendirme faktörü ve $x_{p_2j}(t) - x_{p_3j}(t)$ diferansiyel vektör olarak yer alır.

Çaprazlama işlemi: Nüfus çeşitliliğini artırmak için ikinci adım olarak bir yeniden üretim aşaması olan çaprazlama işlemi yapılır. Aşağıdaki eşitlikle çaprazlama işlemi başlatılır.

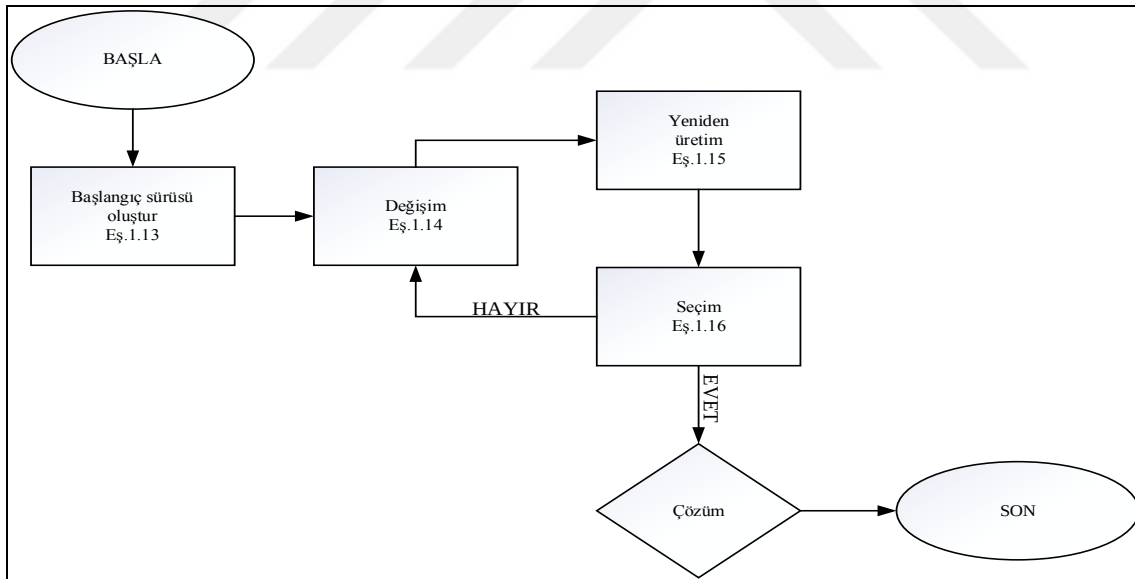
$$v_{ij}(t+1) = \begin{cases} h_{ij}(t+1) & \text{eğer } (rand1_{ij} \leq CR) \text{ ya da } (j = rand(i)) \\ x_{ij}(t) & \text{eğer } (rand1_{ij} > CR) \text{ ve } (j \neq rand(i)) \end{cases} \quad (1.15)$$

eşitlikte verilen $rand1_{ij}$, $[0,1]$ kapalı aralığında herhangi bir rastgele sayıdır. CR , $[0,1]$ aralığına ait sonuç ile çaprazlama sabitini belirtir. $rand(i)$ rastgele bir tam sayıdır. Bu adımda $x_i(t+1)$ ' in $h_i(t+1)$ 'den karşılık gelen en az bir parametreyi almasını sağlar.

Seçim İşlemi: $x_i(t)$ 'nin yeni neslin bir üyesi olup olmayacağına karar vermek için $v_i(t+1)$ vektörü, objektif (amaç) fonksiyonu kullanılarak hedef vektör $x_i(t)$ ile karşılaştırılır.

$$x_i(t+1) = \begin{cases} v_i(t+1) & \text{eğer } f(v_{i1}(t+1), \dots, v_{in}(t+1)) < f(x_{i1}(t), \dots, x_{in}(t)) \\ x_i(t) & \text{eğer } f(v_{i1}(t+1), \dots, v_{in}(t+1)) \geq f(x_{i1}(t), \dots, x_{in}(t)) \end{cases} \quad (1.16)$$

İterasyon maximum sayıya ulaşıncaya kadar sırasıyla değişim, çaprazlama ve seçim süreçleri sürekli tekrar eder. Algoritma durdurulduğunda, optimal ağırlık ve bias değerleri en iyi uygunluk değerine sahip kromozomdan alınır. Şekil (1.10)'da DE algoritmasının akış şeması verilmiştir (Chaparala ve Sajja, 2018).

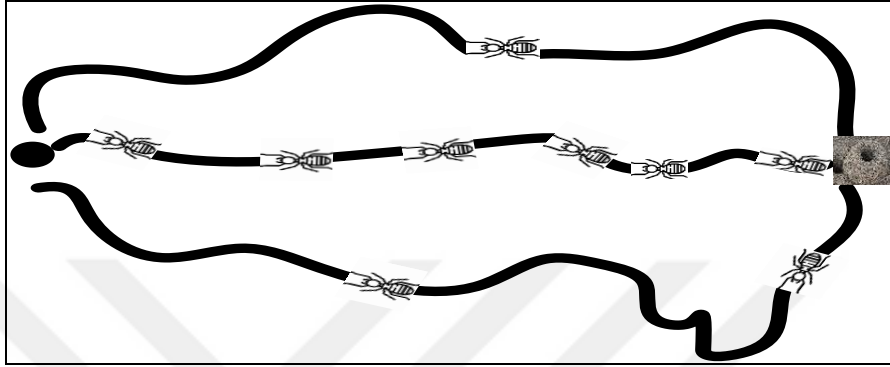


Şekil 1.10. DE algoritma şeması.

1.1.3. Karınca kolonisi optimizasyon algoritması

Karınca kolonisi optimizasyon (Ant colony optimization, ACO) algoritması, doğadan esinlenen ve sürü zekâsına dayanan bir sezgisel-üstü algoritmadır (Dorigo ve

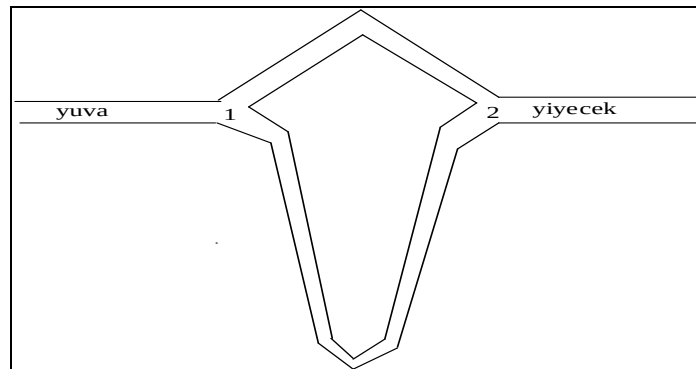
ark., 1999; Dorigo ve ark., 2006). ACO, bazı karınca türlerinin beslenme davranışından ilham alır. Bu karıncalar, koloninin diğer üyeleri tarafından izlenmesi gereken bazı uygun yolları işaretlemek için feromon denen bir madde salgırlarlar. Şekil 1.11’de karıncaların en kısa yolu tercih ettiği görülmektedir. (Mirjalili ve ark., 2020).



Şekil 1.11. Yuva ile yiyecek arasında üç yol.

Bir yuvadan yemek kaynağına uzanan üç yol örneğinin gözlemlendiği Şekil 1.11’de bir rotada biriken feromon miktarı en yakın yol üzerinde en yüksektir. Feromon, yolun uzunluğuna göre dönel olarak azalır. Karıncalar besin kaynağına giden yollara feromon eklerken, buharlaşma meydana gelir. Bir karıncanın buharlaşmadan önce feromonu doldurduğu süre, yolun uzunluğu ile ters orantılıdır. Bu, en yakın rotadaki feromonun daha fazla karınca en güçlü feromona çekildikçe daha konsantre hale geldiği anlamına gelir (Mirjalili ve ark., 2020).

Karınca kolonisi optimizasyonu, optimizasyon problemlerini çözmek için benzer bir mekanizmadan yararlanır (Dorigo ve ark., 2006).



Şekil 1.12. Çift köprü deneyi.

Şekil 1.12’de görüldüğü gibi Goss ve arkadaşları bir köprünün diğerinden önemli ölçüde daha uzun olduğu çift köprü deneyinin bir varyantını ele almıştır (Goss ve ark., 1989). Bu durumda, ilk köprü seçimindeki olasılıksal değişme çok azalır ve ikinci bir mekanizma önemli bir rol oynar. Karıncaların bir kısmı kısa köprüyü seçerek yuvaya ilk ulaşanlardır. Bu nedenle kısa köprüye uzun olandan daha önce feromon bırakılır ve bu gerçek, uzun yolu kullanan karıncalar yerine karıncaların kısa yolu seçme olasılığını artırır. Goss ve arkadaşları bu gözlemlenen davranışın bir modelini geliştirdiler, belirli bir anda m_1 karıncalarının birinci köprüyü, m_2 karıncaları da ikinci köprüyü kullandıkları varsayılarak p_1 olasılığı aşağıdaki eşitlikte gösterildiği gibi hesaplınsın.

$$p_1 = \frac{(m_1 + k)^h}{(m_1 + k)^h + (m_2 + k)^h} \quad (1.17)$$

bu ifadede k ve h parametreleri deneysel verilere eklenen rastgele değerlerdir. Monte Carlo simülasyonları en iyi değerlerin $k \approx 20$ ve $h \approx 20$ olarak göstermiştir (Pasteels ve ark., 1987).

ACO optimizasyon problemlerinin çözümünde sezgisel-üstü bir algoritma olarak formülize edildi (Dorigo ve ark., 1999). Orijinal karınca kolonisi optimizasyon algoritması karınca sistemi olarak bilinir (Dorigo ve ark., 1989; Dorigo ve ark., 1991). Parametreleri ve feromon izlerini başlattıktan sonra, ana döngü üç temel adımdan oluşur. İlk olarak karıncalar, söz konusu problem örneğine feromon bilgisine ve muhtemelen mevcut içgüdüsel bilgilere dayanarak çözümler oluştururlar. Karıncalar çözümlerini tamamladıktan sonra, bunlar isteğe bağlı bir yerel arama aşamasında geliştirilebilir. Son olarak bir sonraki iterasyonun başlamasından önce feromon izleri, karıncaların arama tecrübelerini yansıtacak şekilde uyarlanmıştır. ACO sezgisel-üstü algoritmasının ana adımları aşağıda ifade edilmiştir (Dorigo ve Stützle, 2019).

Başlangıç: Algoritmanın başlangıcında parametreler ayarlanır ve tüm feromon değişkenleri, algoritmanın bir parametresi olan τ_0 değerine başlatılır.

Karınca çözümleri oluşturmak: Bir dizi karınca (m), ele alınan probleme çözümler oluşturmaya başlar. Bunu yapmak için her karınca başlangıçta boş bir çözüm olan $s_p = \phi$ ile başlar. Her bir çözüm aşamasında bir karınca, bir uygulanabilir çözüm

bileşeni $c_i^j \in N(S_p) \subseteq C$ seçerek ve mevcut kısmi çözümüne ekleyerek mevcut kısmi çözümü olan S_p 'yi genişletir.

$N(S_p)$, uygulanabilir bileşenlerin bir kümesidir. Eğer uygulanabilirlik korunurken kısmi bir çözüme ulaşılmazsa, problem kısıtlarının ihlali derecesine bağlı olarak, mümkün olmayan çözümler cezalandırılabilir. Eklenecek çözüm bileşeninin seçimi, her aşamada belli bir olasılık dağılımıyla gerçekleştirilir (Dorigo ve Stützle, 2019).

$$p\left(\frac{c_i^j}{S_p}\right) = \frac{\tau_{ij}^\alpha \cdot [\eta(c_i^j)]^\beta}{\sum_{c_i^l \in (S_p)} \tau_{il}^\alpha \cdot [\eta(c_i^l)]^\beta}, \forall c_i^j \in N(S_p) \quad (1.18)$$

bu ifadede $\eta(\cdot)$ her $c_i^j \in N(S_p)$ için bir c_i^j değeri atan bir fonksiyondur. α ve β parametreleri feromon izlerinin göreceli etkisini ve sezgisel bilgileri belirler ve algoritma davranışı üzerinde aşağıdaki etkiye sahiptir. Eğer $\alpha = 0$ ise seçim olasılıkları η_{ij}, β ile orantılıdır ve yüksek sezgisel değere sahip bir çözüm bileşeninin seçilme olasılığı yüksektir ki bu durum stokastik bir açgözlü algoritmaya karşılık gelir. $\beta = 0$ ise sadece feromon artımı söz konusudur.

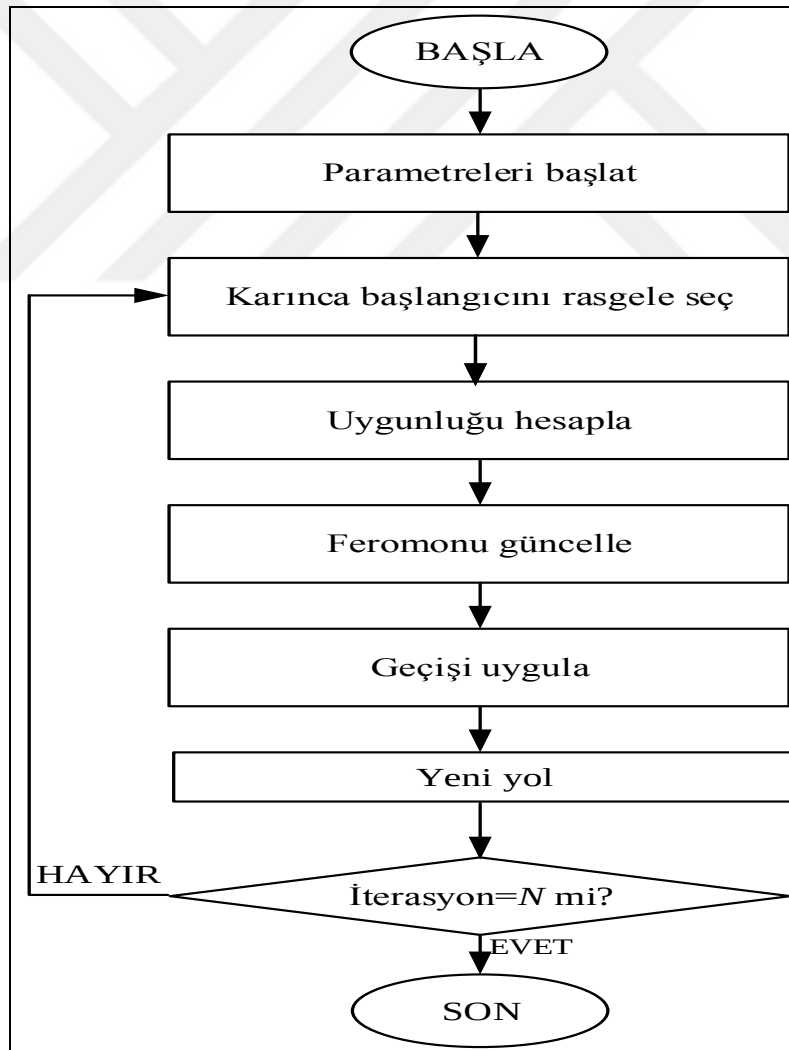
Yerel arama uygulamaları: Tam aday çözümler elde edildikten sonra, bunlar yerel arama algoritmaları uygulanarak daha da geliştirilebilir. Aslında, çok çeşitli kısıtlı optimizasyonu problemleri için ACO algoritmaları yerel arama algoritmalarıyla birleştiğinde en iyi performansa ulaşır. Daha genel olarak yerel arama, feromon değerlerini güncelleyerek yerel arama çözümleri üreten daemon eylemleri olarak ifade edilen süreçlerin bir örneğidir (Mirjalili ve ark., 2020).

Küresel feromon güncelleme: Feromon güncellemesi, iyi çözümlere ait çözüm bileşenlerini veya aşağıdaki iterasyonları daha cazip hale getirmeyi amaçlamaktadır. Bu hedefe ulaşmak için esasen kullanılan iki mekanizma vardır. Birincisi, seçilmiş iyi bir dizi S_{upd} ile ilişkili olan çözüm bileşenlerinin feromon seviyesini arttıran feromon birikintisidir. Amaç, bu çözüm bileşenlerini aşağıdaki yinelemelerde karıncalar için daha çekici hale getirmektir. İkincisi, önceki karıncalar tarafından biriken feromonun zamanla azaldığı mekanizma olan feromon izi buharlaşmasıdır. Feromon buharlaşması algoritmanın optimal olmayan bir bölgede çok hızlı yakınsamasını önlemek açısından

önemlidir. Buharlaşıma sayesinde algoritmanın keşif gücü artar. Feromon güncellemesi aşağıdaki eşitlikte ifade edildiği şekilde uygulanır (Dorigo ve Stützle, 2019).

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_{s \in Supd | c_i^j \in s} g(s) \quad (1.19)$$

Bu eşitlikte S_{ubd} feromon biriktirmek için kullanılan bir dizi çözüm, $\rho \in (0,1]$ parametresi buharlaşma hızı, $g(.): S \rightarrow S'$ ve $(s) < f(s') \Rightarrow g(s) \geq g(s')$ olmak üzere $g(.)$ fonksiyonuna değerlendirme fonksiyonu denir ve amacı çözümün kalitesini belirlemektir (Dorigo ve Stützle, 2019). Şekil 1.13'te ACO algoritmasının akış şeması verilmiştir (Okonta ve ark., 2016).



Şekil 1.13. ACO akış şeması.

1.1.4. Parçacık sürü optimizasyon algoritması

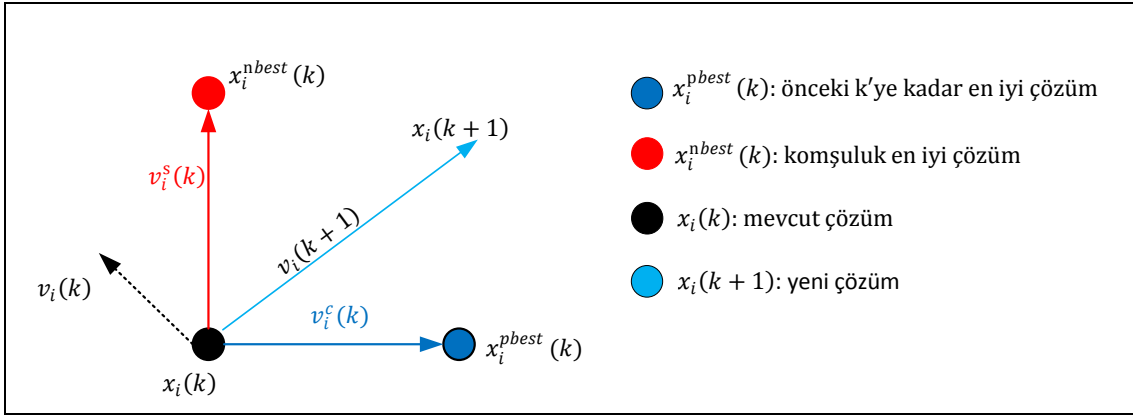
Parçacık Sürü Optimizasyon (Particle Swarm Optimization, PSO) algoritması, sürü temelli evrimsel bir hesaplama tekniğidir. Amaç, bir kuş sürüsünün zarif ama öngörülemeyen hareketlerini grafiksel olarak simüle etmektir (Kennedy ve Eberhart, 1995). PSO, başlangıç sürüsü ile rastgele çözümler üreterek sistemi başlattığından genetik algoritmaya benzemekle beraber her bir potansiyel çözüme rastgele bir hız atanması ve tarama alanında her bir parçacık kendi deneyimi ve diğer parçacıkların deneyimine göre hareketini ayarlaması yönleriyle de GA'dan farklıdır (Eberhart ve Shi, 1998). PSO algoritmasında her bir parçacık olası bir çözümü ifade eder. Algoritma en iyi çözümü bulmak için arama uzayında birtakım parçacıklar (aday çözümler) kullanır. Her bir parçacık elde ettiği en iyi çözümü (*pbest*) kaydeder, PSO'nun küresel versiyonu tarafından toplam en iyi değer sürüdeki herhangi bir parçacık tarafından şimdiye kadar bulunan en iyi konumdur (*gbest*). PSO algoritmasının her bir adımı, parçacıkların yerel ve küresel en iyi konumlarına doğru hızının değiştirilmesinden oluşur. Hızlanma rastgele değerlerle ağırlıklandırılır. İlk önce, başlangıç sürüsü belli bir aralıkta rastgele konum ve hız değerleri ile atanır. Finalde parçacıklar kendi uygun konumlarına geldiklerinde optimizasyon meydana gelir. Sıfır hız başlatması durumunda, eğer bir parçacık kendinden daha iyi bir komşuya sahipse ilk iterasyon sırasında arama alanından çıkabilir (Aydilek, 2018).

PSO algoritması aşağıdaki eşitliklerle ifade edilir (Abedifar ve ark., 2013).

$$v_i^{t+1} = w \cdot v_i^t + c_1 \cdot r_1 \cdot (pbest_i - x_i^t) + c_2 \cdot r_2 \cdot (gbest_i - x_i^t) \quad (1.20)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (1.21)$$

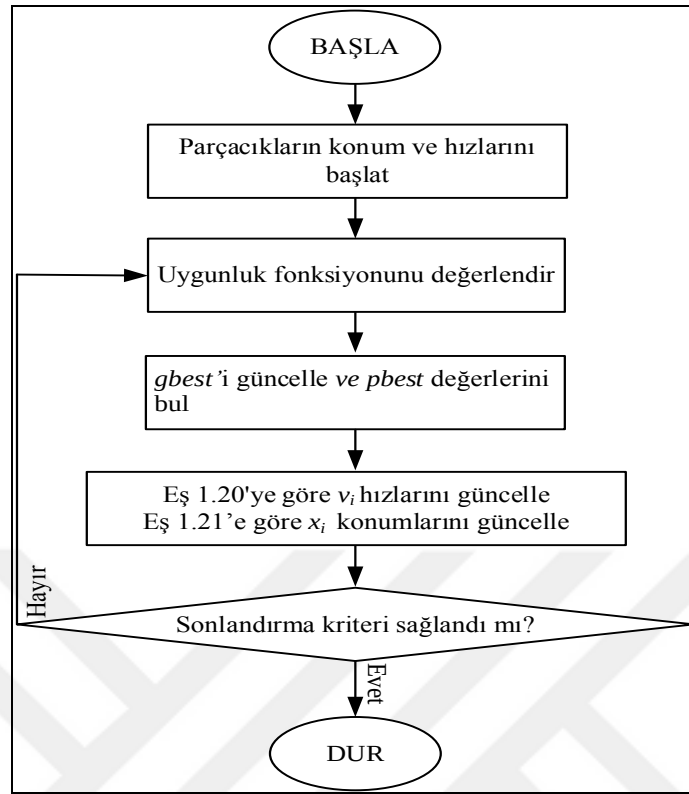
“Eş. 1.20” de v_i^t t . iterasyonda i . parçacığın hızı, w ağırlıklandırma fonksiyonu, c_1 ve c_2 ağırlıklandırma faktörleri, r_1 ve r_2 0 ile 1 aralığında rastgele sayılar, x_i^t t . iterasyonda i . parçacığın mevcut konumu, $pbest_i$ t . iterasyonda i . parçacığın en iyi konumu, $gbest_i$ şu ana kadar en iyi çözümü ifade eder. Şekil 1.14'te iki boyutlu resimde PSO konum güncellemesi verilmektedir (Zhou ve ark., 2003). Şekilde sosyal (s) ve bilişsel (c) bilgiler, uygun bir değer bulmaya çalışan algoritma ile parçacıklar arasında paylaşılmaktadır (Çavdar, 2016).



Şekil 1.14. PSO çözüm güncelleme.

PSO, parçacıkları rastgele bir arama ya da problem uzayına yerleştirmekle başlar. Parçacıkların hızları “Eş.1.20” kullanılarak her bir iterasyonda hesaplanır. Hızları tanımladıktan sonra, sürünün konumu “Eş.1.21” kullanılarak hesaplanabilir. Parçacıkların konumlarını değiştirme süreci bir son kriteri karşılayana kadar devam edecektir.

Kennedy ve Eberhart “PSO’nun, sosyal-psikolojik teoriden türetilen adaptasyon ile doğrusal olmayan ve ayırt edilemezlik, çoklu optima ve yüksek boyutsallık içeren problemleri çözmede sağlam olduğu bulunmuştur” (Kennedy, 1997) şeklinde tespit etmişlerdir (Kennedy ve Eberhart, 1997). Şekil 1.15’te PSO akış şeması çizilerek işlem aşamaları gösterilmiştir.



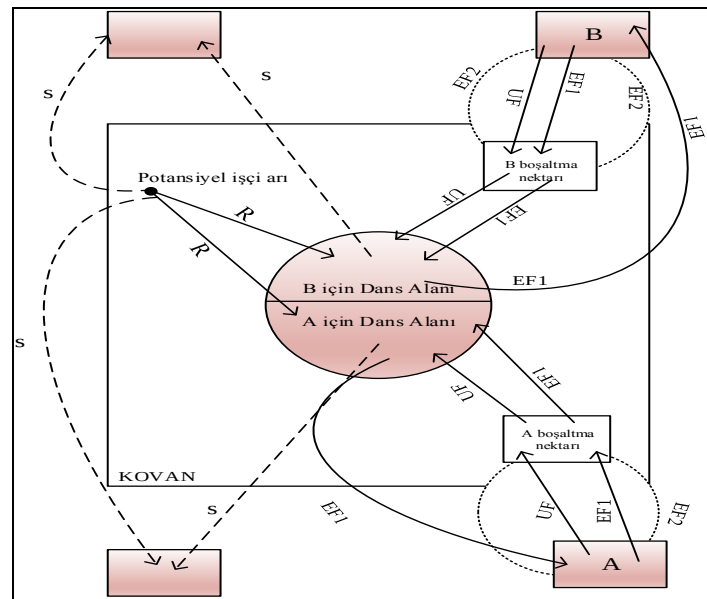
Şekil 1.15. PSO akış şeması.

1.1.5. Yapay arı kolonisi algoritması

Yapay Arı Kolonisi (Artificial Bee Colony, ABC) algoritması, koloni şeklinde yaşayan arıların besin kaynağı arama davranışları ilham alınarak yapılmış bir sezgisel-üstü algoritmadır (Karaboğa, 2005). ABC algoritmasında bir gıda kaynağı, arama alanındaki optimizasyon problemine olası bir çözümü temsil eder ve gıda kaynağının nektar miktarı bu çözümün uygunluğunu temsil eder. ABC algoritması üç tür arı tanımlar: işçi arılar, gözcü arılar ve kaşif arılar. İşçi bir arı, daha önce ziyaret ettiği gıda kaynağının çevresinde yeni gıda kaynakları arar. Gözcü bir arı dansları gözlemler ve ziyaret etmek için bir yiyecek kaynağı seçer. İşçi arılar tarafından bulunanlardan daha iyi gıda kaynakları seçme eğilimindedir. Kaşif bir arı rastgele yeni yiyecek kaynakları arar. ABC algoritma döngüsü şöyle ifade edilebilir: İşçi arılar çözüm olarak ifade edilen, önce gıda kaynaklarının yakınında belleklerinde bir alan araştırması yaparlar. Sonra kovana geri dönüp dans ederler. Danslar, gözcü arıları her çözümün uygunluğu hakkında bilgilendirir. Her gözcü arı, gıda kaynağı uygunluğuna orantılı bir olasılık temelinde başka bir alan araması yapmak için bir gıda kaynağını gözlemler ve seçer.

Gözcü arılar, işçi arılar tarafından bulunanlardan daha iyi gıda kaynakları seçme eğilimindedir. İşçi ve gözcü arılar, yeni bir çözüm üretmek için mevcut bir çözümü bozarak alan araması yaparlar. Yeni bozulan çözümün kabul edilip edilmeyeceğine karar vermek için açgözlü bir yaklaşım uygulanır. Önceden belirlenmiş sayı sınırından sonra bir çözüm geliştirilemezse terk edilir. İyileşmeyen bu çözüm ile ilişkili olarak kullanılan arı lokal optimum olarak ifade edilerek terk edilir ve bir kaşif arısı haline gelir. Kaşif arı arama alanını rastgele keşfeder ve yeni bir çözüm arar (Choong ve ark., 2019).

Aşağıda şekil 1.16’da bir arı kolonisindeki yapay arıların temel davranışlarını göstermektedir (Karaboğa ve Akay, 2009). A ve B gibi iki besin kaynağı olduğu varsayalım. Arı ilk besin aramaya başladığında bir kaşif arı olarak kendiliğinden besin arayabilir (S) ya da dansları izledikten sonra besin kaynağı arayabilir (R). Yiyecek kaynağını bulduktan sonra, arı işçi arı haline gelecektir. İşçi arı, kaynaktan bir nektar yükü alır ve kovana geri döner ve nektarı bir gıda deposuna boşaltır. Yiyecekleri boşalttıktan sonra arı şu seçeneklerden birini kullanabilir. Gıda kaynağını (UF) terk ettikten sonra gözcü olabilir veya aynı gıda kaynağına (EF1) dönmeden önce dans edebilir ve yeni işçi arılara danslarıyla rehberlik edebilir, son seçenek olarak gıda kaynağında nektar taşımaya devam edebilir (EF2) (Karaboğa ve Akay, 2009).



Şekil 1.16. Nektar için balarısı yiyecek arama davranışı.

ABC algoritmasında sadece üç adet kontrol parametresi vardır; gözcü arı sayısına eşit olan besin kaynak sayısı (NS), maximum döngü sayısı (MDS) ve sınır değerleri x^{min} , x^{max} dır. $i=1, \dots, NS$ ve $j=1, \dots, D$, burada D optimize edilecek problem sayısını göstermek üzere yiyecek kaynakları aşağıdaki eşitlikte ifade edilir (Babayiğit ve Özdemir, 2012).

$$x_{i,j} = x_j^{min} + rand[0,1](x_j^{max} - x_j^{min}) \quad (1.22)$$

Başlangıçta MDS sıfırlanır. Arılar yeni besin kaynağı bulduklarında eskisiyle karşılaştırarak iyi çözümü hafızada tutar diğerini silerler. Eski çözüm daha iyiye hali hazırdaki MDS bir artırılır. Kâşif arıların arama işlemi aşağıdaki eşitlikte ifade edilmiştir (Babayiğit ve Özdemir, 2012).

$$v_{i,j} = x_{i,j} + \varphi_{i,j}(x_{i,j} - x_{k,j}) \quad (1.23)$$

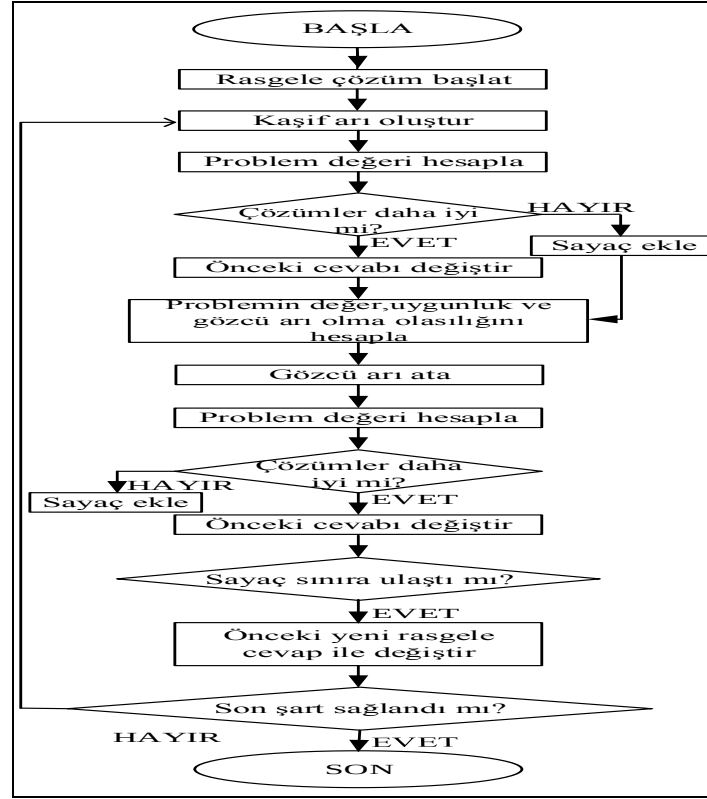
Eşitlikte x_i , mevcut çözümü ve v_i , yeni çözümü belirtsin, k , $[1, NS]$ aralığındaki rastgele bir tamsayı, x_k besin kaynakları içerisinde seçilmiş, x_i den farklı bir çözümü, $j, [1, D]$ kapalı aralığında yer alan, optimize edilecek problemin rastgele bir parametresini belirtirken, $[-1, 1]$ aralığında seçilmiş rastgele bir değer $\varphi_{i,j}$ ile gösterilir. Kâşif arılar besin kaynaklarına ulaştıktan sonra gözcü arıların bunlar içerisinde seçim yapma olasılıkları aşağıdaki eşitlikte ifade edilmiştir (Babayiğit ve Özdemir, 2012).

$$p_i = \frac{f_i t_i}{\sum_{i=1}^{NS} f_i t_i} \quad (1.24)$$

Bu eşitlik, normalize edilmiş maliyet fonksiyonu değeri olarak adlandırılır. Kaşif arı, p_i sonuçları ile rastgele belirlenmiş bir değeri karşılaştırır eğer p_i değeri büyükse gözcü arı bu besin kaynağına yönelir ve $v_{i,j}$ bağlantısını kullanarak yeni çözümler arar.

Arama işlemleri bittikten sonra her bir çözüm için MDS kontrol edilir. Eğer bir çözümün sayaç değeri sınır değerine ulaşmış ise o kaynağı kullanan görevli arı kâşif arı olur, $x_{i,j}$ eşitliği kullanılarak çözüm uzayında yeni bir noktaya yönelir ve araştırmasına buradan devam eder. Bütün işlemler maksimum çevrim sayısına ulaşıncaya kadar sürer,

en iyi çözüm kaydedilir (Karaboğa ve Akay, 2009). ABC algoritmasının işlem aşamaları Şekil 1.17’de verilmiştir (Ghaleini ve ark., 2019).



Şekil 1.17. ABC algoritması akış şeması.

1.1.6. Rüzgâr güdümlü optimizasyon algoritması

Rüzgâr Güdümlü Optimizasyon (Wind Driven Optimization, WDO) algoritması, Dünya atmosferindeki rüzgârın hareketinden ilham alan çok boyutlu ve çok modlu problemlerin çözümü için önerilebilen sürü tabanlı, sezgisel-üstü küresel optimizasyon algoritmasıdır (Bayraktar ve ark., 2010; Bayraktar ve ark., 2013). WDO, atmosferik hareketlere uygulandığında çok doğru sonuçlar veren Newton’un ikinci hareket kanununu başlangıç noktası kabul ederek geliştirilmiştir (Bayraktar ve ark., 2013). WDO, çok boyutlu optimizasyon problemlerini çözmek için sürü temelli sezgisel-üstü ve iteratif sürecin bulunabileceği doğadan ilham alan diğer optimizasyon algoritmalarına benzer. Benzer parçacık temelli algoritmalar ile karşılaştırıldığında, WDO algoritması yerçekimi ve coriolis (saptırıcı) kuvvetleri (Riehl, 1978) gibi hız

güncelleme denkleminde sağlamlık ve ince ayar serbestliği sağlamak için kullanılan ek terimlerden yararlanır (Bhandari ve ark., 2014). WDO' nun başlangıç adımı, Newton'un özellikle atmosferik hareketin analizi için doğru sonuçları belirlemekte kullanılan aşağıda matematiksel ifadesi verilmiş olan ikinci hareket yasası tarafından desteklenmektedir (Bayraktar ve ark., 2013).

$$\rho \vec{a} = \sum \vec{F}_i \quad (1.25)$$

Yukarıdaki ifadede, $\vec{a}$ hız vektörü, ρ son derece küçük hacimli bir element için hava yoğunluğudur ve F_i kütle üzerine etki eden kuvvetlerdir. İdeal gaz yasası ile hava basıncı, yoğunluğu ve sıcaklığı arasındaki ilişki aşağıda ifade edilmiştir (Bayraktar ve ark., 2013).

$$P = \rho RT \quad (1.26)$$

Yukarıdaki ifadede, P basınç, R küresel gaz sabiti ve T sıcaklığı ifade eder.

(1.27) eşitliğinde, rüzgârın belirli bir yönde hareket etmesine veya yolundan sapmasına neden olan dört ana kuvvet vardır. Havanın hareket etmesine neden olan en gözlenebilir kuvvet basınç eğim kuvveti F_{PG} iken, bu harekete zıt yönde sürtünme kuvveti F_F hareket eder. Yerçekimi kuvveti F_F fiziksel üç boyutlu atmosferimizde dikey bir kuvvet gibi görünse de, N -boyutlu uzaya eşlendiğinde, koordinat sisteminin orijinine doğru çeken cazip bir kuvvet haline gelir. Bu nedenle yerçekimi kuvveti algoritmaya dahil edilmiştir. Coriolis kuvvetine F_C dünyanın dönüşü neden olur ve rüzgârın yolunu bir boyuttan diğerine saptırdığı için WDO'da, bir boyutta diğerinde hızı etkileyen bir hareket olarak uygulanır (Bayraktar ve ark., 2010). Bu kuvvetlerin her birini yöneten fiziksel eşitlikler aşağıda verilmiştir, burada δV son derece küçük miktarda hava hacmini temsil eder. ∇P basınç gradyanıdır, Ω dünyanın dönüşünü temsil eder, g yerçekimi ivmesidir ve u rüzgârın hız vektörüdür. (Bayraktar ve ark., 2010).

$$\left. \begin{aligned} \vec{F}_{PG} &= -\nabla P \delta V \\ \vec{F}_G &= \rho P \delta \vec{g} \\ \vec{F}_C &= -2\Omega \times \vec{u} \\ \vec{F}_F &= -\rho \alpha \vec{u} \end{aligned} \right\} \quad (1.27)$$

Tüm bu kuvvetleri ifade eden eşitlikleri Newton'un ikinci yasasını ifade eden eşitlikte yerine yazılabilir (Bayraktar ve ark., 2010).

$$\rho \vec{u} \Delta t = (\rho \delta V \vec{g}) + (-\nabla P \delta V) + (-\rho \alpha \vec{u}) + (-2\Omega \times \vec{u}) \quad (1.28)$$

Rüzgârla hareket eden çok küçük bir hava parselini (hava miktarı) düşünürsek, basitleştirerek bir hız güncelleme (1.28) eşitliği türetilebilir. (1.29) eşitliğinden ideal gaz yasası eşitliğini kullanarak, aynı şekilde ρ 'yi basınç olarak yazılabilir ve basitlik için ilk adımı, $\Delta t = 1$ olduğu varsayılabilir.

(1.29) eşitliğindeki terimleri yeniden düzenledikten sonra, aşağıdaki hız güncelleme eşitliği türetilebilir (Bayraktar ve ark., 2010).

$$\vec{u}_{new} = ((1 - \alpha)\vec{u}_{old}) + g(-\vec{x}_{old}) + \left[\left| \frac{P_{max}}{P_{old}} - 1 \right| RT(x_{max} - x_{old}) \right] + \left[\frac{-c_{old}^{other \ dim}}{P_{old}} \right] \quad (1.29)$$

(1.28) eşitliğinde, bir sonraki iterasyon için güncellenmiş hız, yeni iterasyon, mevcut yineleme hızına u_{old} , arama alanındaki hava parselinin geçerli konumuna x_{old} , en yüksek basınç noktasından uzaklığa bağlı bulunan x_{max} maksimum basınç P_{max} mevcut konumdaki basınç P_{old} , sıcaklık T , yerçekimi ivmesi g , ve R, α ve c sabitleri bulunur. WDO'daki basınç terimi, GA terminolojisinde bir kromozomun uygunluğuna benzer. WDO, itici olarak en yüksek basınç noktasına doğru ilerleyen arama alanı üzerine rastgele dağıtılan çok küçük hava parselleri sürüsüne bağlıdır, bu hareket atmosferdeki rüzgâr hareketini yöneten fiziksel denklemler tarafından yönlendirilir. Hava parselinin konumu ancak hava parselinin hızı (1.29) ile güncellendikten sonra (1.30) ile güncellenebilir (Bayraktar ve ark., 2010).

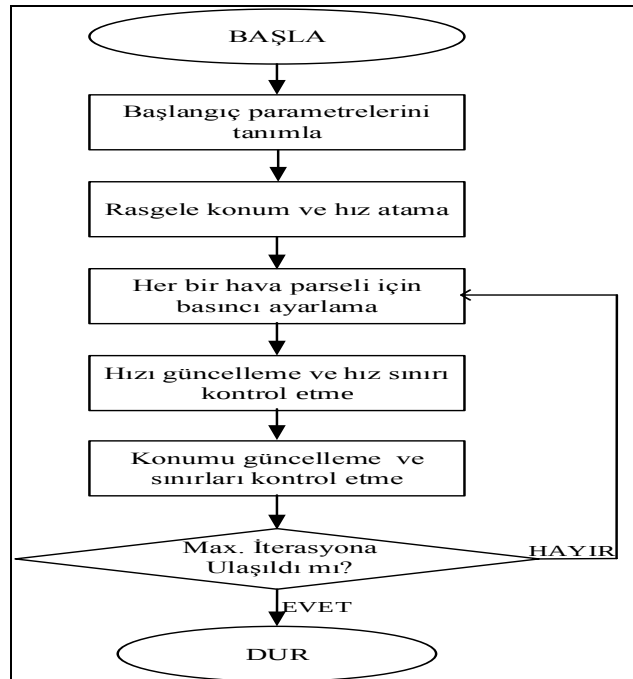
$$\vec{x}_{new} = \vec{x}_{old} + (\vec{u}_{new} \times \Delta t) \quad (1.30)$$

“Eş. 1.30”, $\vec{x}_{old}$, hava parselinin sürtünme nedeniyle oluşan bir muhalefetle önceki yolunda ilerlemeye devam edeceğini gösterir. $\vec{u}_{new}$, koordinat sisteminin merkezine doğru çeken cazip bir kuvvettir. Δt , optimizasyon problemi için küresel en iyi konum olarak kabul edilen maksimum basınç konumuna zıt kuvveti gösterir. $\vec{x}_{new}$, aslında, belirleyici bir kuvvet olan Coriolis kuvvetini takip eder. Bu şekilde, WDO karmaşık optimizasyon problemlerini çözmek için basit ama oldukça etkili bir yol sunar. Her

boyut için WDO hava parsellerinin sadece $[-1,1]$ sınırları içinde hareket etmesine izin verir. WDO algoritmasında, bir hava parselinin herhangi bir boyuttaki bu somut sınırların dışından geçmeye çalışması durumunda, konumu bu boyutta sınır değerine ayarlanır. Sonuç olarak yerçekimi kuvveti, sınırlarda tutulan her hava parselini arama alanına geri döndürür. Ayrıca, hava parsellerinin güncellenmiş hızlarının iterasyon başına maksimum bir değerle sınırlı olduğu da görülmelidir. Bunun nedeni, hava parsellerinin büyük adımlar atmasını ve arama alanındaki belirli bölgelere bakmasını önlemektir. Hız, herhangi bir boyutta maximum sınırı aştığında, bu boyuttaki hız denklemi aşağıda “Eş.1.31” de ifade edildiği gibi hareket yönü korunur ancak büyüklük herhangi bir boyutta $|u_{max}|$ ile sınırlıdır ve $|u_{max}^*|$, maximum hız ile sınırlandırıldıktan sonra ayarlanan hızı temsil eder (Bhandari ve ark., 2014).

$$U_{new}^* = \begin{cases} U_{max}, & U_{new} > U_{max} \\ -U_{max}, & U_{new} < -U_{max} \end{cases} \quad (1.31)$$

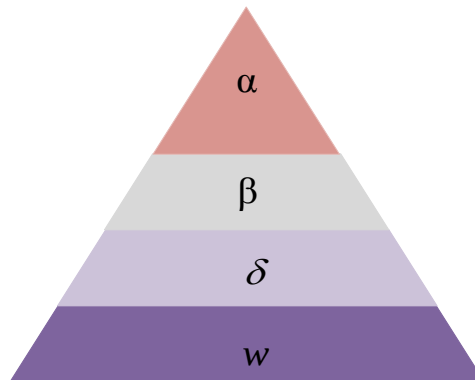
Şekil 1.18’de WDO algoritması akış şeması ile işlem sırası gösterilmiştir (Bhandari ve ark., 2014).



Şekil 1.18. WDO algoritması akış şeması.

1.1.7. Gri kurt optimizasyon algoritması

Gri Kurt Optimizasyon (Grey wolf Optimization, GWO) algoritması sürü bazlı sezgisel-üstü bir optimizasyon algoritmasıdır (Mirjalili ve ark., 2014). GWO sosyal bir hiyerarşi türüne sahip bir kurt sürüsünü baz alır. Algoritmada α, β, δ ve ω şeklinde modellenmiş kurt türleri bulunmaktadır. Sürüde α (alfa) en fazla sorumluluğu alan lider kurttur ve sürü α 'nın kararlarıyla hareket eder. β hiyerarşide ikinci sırada gelir ve görevi α 'ya karar alma ve diğer sürü faaliyetlerinde yardımcı olmak, sürüye emirleri iletme ve α 'ya geri bildirimde bulunarak bir nevi danışmanlık yapmaktır. α öldüğünde de lider olacak en güçlü adaydır. Tüm sürüdeki en düşük dereceli kurt w (omega)'dır. Küçük kurtlara bakmak gibi sıradan işlere bakan omegalar diğer baskın kurtlara boyun eğmek zorundadır. Bu sınıflandırmaların hiçbirine girmeyen alt ya da δ (delta) olarak anılan hiyerarşide üçüncü sırada gelen kurt α ve β 'ya boyun eğen ama w 'ya hâkimiyet kuran kurttur. İzçiler, nöbetçiler, yaşlılar, avcılar ve bakıcılar bu kategoriye girer. İzçiler, bölgenin sınırlarını izlemek ve herhangi bir tehlike durumunda sürüyü uyarmaktan sorumludur. Nöbetçiler, sürünün güvenliğini korur ve garanti eder. Yaşlılar, eskiden alfa veya beta olan deneyimli kurtlardır. Avcılar av avlarken ve sürü için yiyecek sağlarken alfa ve betalara yardım eder. Son olarak, bakıcılar sürüdeki zayıf, hasta ve yaralı kurtların bakımından sorumludur (Mirjalili ve ark., 2014; Dökeroğlu ve ark., 2019). Şekil 1.19 kurt sürüsünün hiyerarşik durumunu sembolize etmiştir (Mirjalili ve ark., 2014).



Şekil 1.19. Gri Kurt hiyerarşisi.

GWO'nun matematiksel modellenmesinde sosyal hiyerarşi, izleme, çevreleme ve ava saldırı ve av arama GWO algoritmasının kilit noktalarıdır. Bu modelde en iyi çözüm alfa (α) olarak kabul edilir. İkinci ve üçüncü en iyi çözümler sırasıyla beta (β) ve delta (δ) 'dır. Sürünün geri kalanına omega (w) denir (Dökeroğlu ve ark., 2019). Optimizasyon sırasında kurtlar α, β ya da δ etrafındaki konumlarını aşağıda eşitlikte ifade edildiği gibi güncellerler (Mirjalili, 2015).

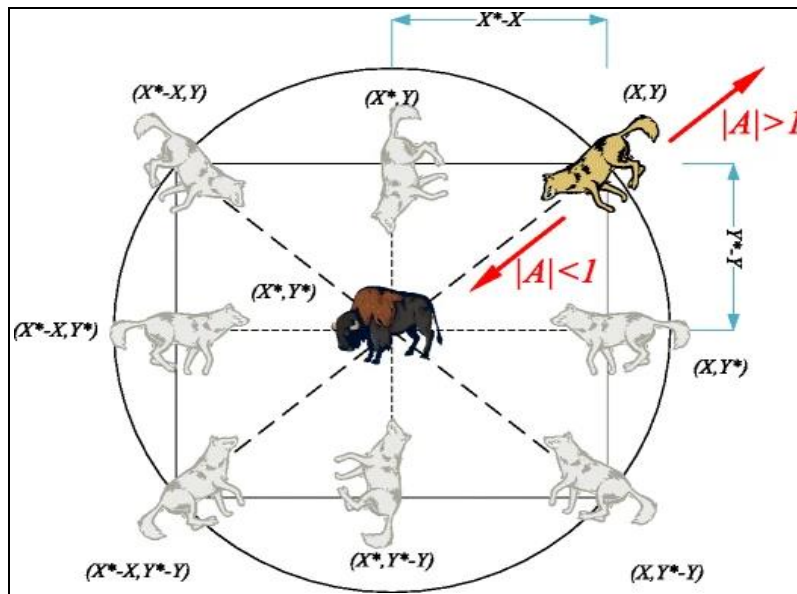
$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \quad (1.32)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (1.33)$$

Yukarıdaki eşitliklerde verilen t mevcut iterasyonu, aşağıdaki eşitlikler ise konum vektörlerini formülize eder.

$$\left. \begin{aligned} \vec{A} &= 2a \cdot \vec{r}_1 - a \\ \vec{C} &= 2 \cdot \vec{r}_2 \cdot \vec{X}_p \end{aligned} \right\} \quad (1.34)$$

$\vec{A}$ ve $\vec{C}$ avnı konum vektörlerini, $\vec{X}$ bir gri kurdun konum vektörünü, a , 2'den 0'a doğrusal olarak düşürülür, r_1 ve r_2 , $[0,1]$ kapalı aralığında rastgele bir vektördür. Şekil 1.20'de farklı pozisyonlardaki gri kurdun pozisyon güncellemesi görsel olarak anlatılmıştır (Mirjalili ve ark., 2014).



Şekil 1.20. Gri kurtların pozisyon güncelleme mekanizması.

“Eş.1.32” ve “Eş. 1.33” ile pozisyon güncelleme Şekil 1.20’de gösterilmiştir. Bu şekilde, (X, Y) konumundaki bir kurtun önerilen eşitliklerle kendini avın etrafında yeniden konumlandığı görülebilir. Şekilde olası yerlerin yedisi gösterilmiş olmasına rağmen, rastgele parametreler A ve C , kurtların avın çevresindeki sürekli boşluktaki herhangi bir konuma taşınmasına izin verir. Optimizasyon sırasında şimdiye kadar elde edilen ilk üç en iyi çözüm sırasıyla α, β ve δ olarak kabul edilir. Daha sonra, diğer kurtlar ω olarak kabul edilir ve α, β ve δ ’ya göre yeniden konumlandırılabilir. Kurtların konumunu yeniden ayarlamak için önerilen eşitlik aşağıdaki gibidir (Mirjalili ve ark., 2014).

$$\left. \begin{aligned} \vec{D}_\alpha &= |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}| \\ \vec{D}_\beta &= |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \\ \vec{D}_\delta &= |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \end{aligned} \right\} \quad (1.35)$$

Bu ifadelerde $\vec{X}_\alpha, \vec{X}_\beta, \vec{X}_\delta$ sırasıyla α, β, δ ’nın konumlarını, $\vec{C}_1, \vec{C}_2, \vec{C}_3$ rastgele vektörleri ve $\vec{X}$ mevcut çözümün konumunu gösterir. Yukarıdaki ifadelerde sırasıyla mevcut çözüm ile α, β, δ arasındaki mesafe belirlenmiş oldu. Mevcut çözümün son konumunun çözümü aşağıdaki gibi ifade edilebilir.

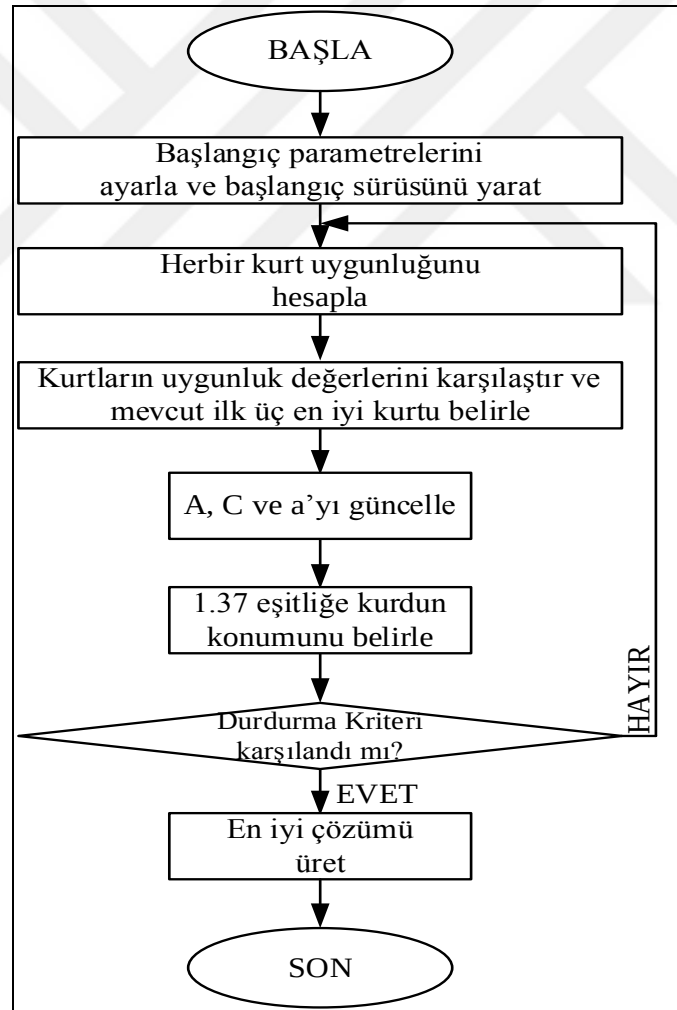
$$\left. \begin{aligned} \vec{X}_1 &= \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha \\ \vec{X}_2 &= \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta \\ \vec{X}_3 &= \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta \end{aligned} \right\} \quad (1.36)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (1.37)$$

Bu eşitliklerde görülebileceği gibi, Eş.1.35 eşitliğinde yer alan ifadeler sırasıyla ω ‘nın α, β ve δ ’ya doğru adım boyutunu tanımlar. Eş.1.36 ve Eş.1.37 ise ω ’nin son konumunu tanımlar. $\vec{A}$ ve $\vec{C}$ vektörleri ise GWO algoritması için keşif ve sömürü sağlayan rastgele ve uyarlanabilir vektörlerdir. Şekilde görüldüğü gibi $|A| > 1$ ve ayrıca $C > 1$ olduğunda keşif aşaması, $|A| < 1$ ve $C < 1$ olduğunda ise sömürü aşaması gerçekleşir. Burada, yineleme sayacı arttıkça sömürüyü vurgulamak için A ’nın optimizasyon sırasında doğrusal olarak azaldığı belirtilmelidir. Bununla birlikte, C

optimizasyon boyunca herhangi bir aşamada keşif / sömürüyü vurgulamak için rastgele üretilir, bu da yerel optimum tuzağından kurtulmak için çok yararlı bir mekanizmadır.

GWO algoritmasının keşif gücü çok yüksektir ve bu sayede yerel optimumdan kaçınmasını sağlar. Ayrıca, keşif ve sömürü dengesi, zorlu problemlerin çözümünde gerçek problemlerin sonuçlarına göre çok basit ve etkilidir (Mirjalili ve ark., 2014). YSA'ları eğitme sorunu, YSA için sağlanabilecek çeşitli veri kümeleri ve girdilere bağlı olarak bilinmeyen bir arama alanı ile zorlu bir sorun olarak kabul edilir. Bu nedenle, GWO algoritmasının yüksek oranda keşif gücüne sahip olması, bir MLP eğitmeni olarak verimli olmasını gerektirir (Mirjalili, 2015). Şekil 1.21 GWO algoritmasının işlem sırasını göstermektedir (Gai ve ark., 2018).



Şekil 1.21. GWO algoritması.

1.1.8. Sinüs kosinüs algoritması

Sinüs Kosinüs Algoritması (Sine Cosine Algorithm, SCA), sürü temelli matematiksel fonksiyonlara dayalı sezgisel-üstü bir algoritmadır (Mirjalili, 2016). SCA, başlangıçta birçok rastgele aday çözüm üretmekle beraber bir amaç fonksiyonu tarafından iteratif bir şekilde bir dizi kural uygulayarak en iyi çözümü bulmaya çalışır. Bu aşamayla beraber, araştırma uzayının keşfi ve sömürülmesi için birkaç rastgele ve uyarlanabilir değişkeni algoritmaya ekler (Bairathi ve Gopalani, 2017; Eker ve ark., 2019). SCA algoritmasında keşif ve sömürü aşamalarının konum güncelleme eşitlikleri aşağıda ifade edildiği şekilde verilmiştir (Mirjalili, 2016).

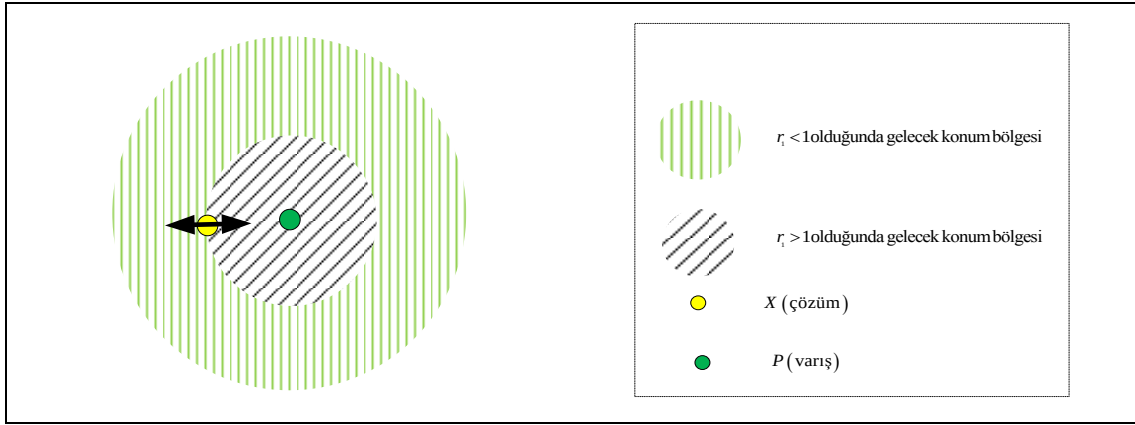
$$X_i^{t+1} = X_i^t + r_1 \times \sin(r_2) \times |r_3 P_i^t - X_i^t| \quad (1.38)$$

$$X_i^{t+1} = X_i^t + r_1 \times \cos(r_2) \times |r_3 P_i^t - X_i^t| \quad (1.39)$$

Bu eşitlikte X_i^t , t . İterasyonda i . boyutta mevcut çözümün konumu r_1, r_2, r_3 rastgele sayılar, P, i . boyuttaki varış noktasının konumu ve $|\cdot|$ mutlak değeri gösterir.

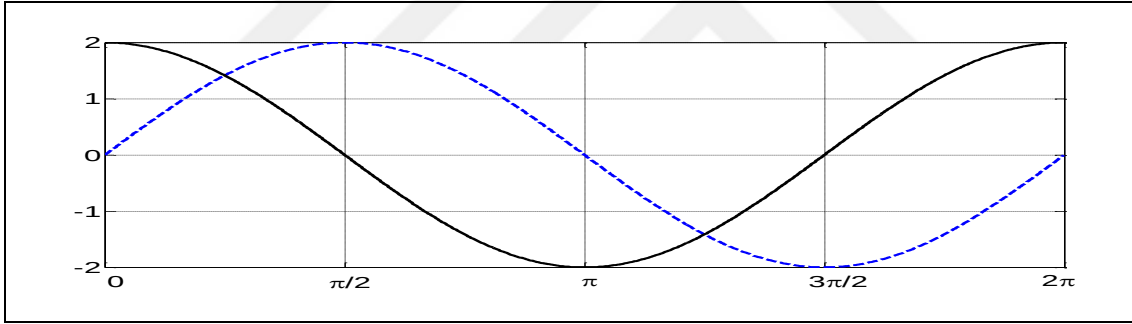
$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \times \sin(r_2) \times |r_3 P_i^t - X_i^t|, & r_4 < 0.5 \\ X_i^t + r_1 \times \cos(r_2) \times |r_3 P_i^t - X_i^t|, & r_4 \geq 0.5 \end{cases} \quad (1.40)$$

Bu algoritmanın ana parametreleri r_1, r_2, r_3 ve r_4 tür. r_1 parametresi, çözüm ile hedef arasındaki alanda veya dışında olabilecek bir sonraki konumun hareket yönünü belirler. r_2 tanım parametresi, hareketin varış yerini korumak veya dışarıya doğru olması gerektiğini gösterir. r_3 parametresi, ($r_3 > 1$) ise hedefin mesafeyi tanımlamadaki etkisini olasılıksal olarak arttırmak ($r_3 < 1$) ise arttırmamak biçiminde hedef için rastgele bir ağırlık getirir. Son olarak $[0,1]$ kapalı aralığında bir r_4 parametresi birinci eşitlikteki iki değeri birleştirir ve $r_4 = 0.5$ gibi belli bir değer üzerinden de geçişi sağlar (Mirjalili, 2016; Eker ve ark., 2019).

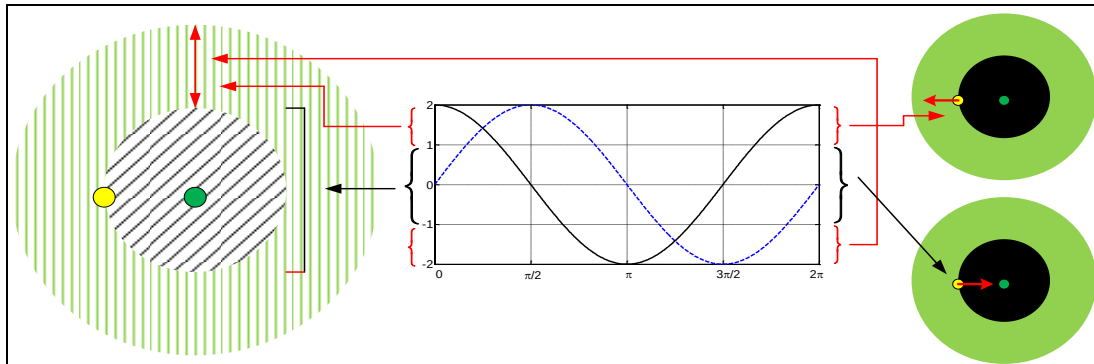


Şekil 1.22. Sinüs ve kosinüsün bir sonraki pozisyona etkileri.

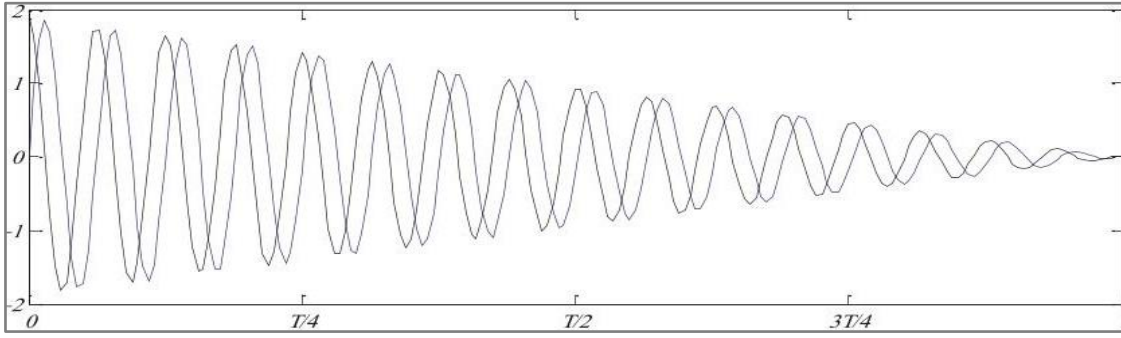
Şekil 1.23, verilen eşitliklerin arama alanındaki iki çözüm arasındaki alanı nasıl tanımladığını göstermektedir. Bu eşitlikler şekilde iki boyutlu gösterilmesine rağmen daha yüksek boyutlara genişletilebilir (Mirjalili, 2016).



Şekil 1.23. [-2,2] aralığında sinüs ve kosinüs fonksiyonu.



Şekil 1.24. [-2, 2] aralığındaki sinüs ve kosinüs fonksiyonlarının etkilerinin kavramsal modeli.



Şekil 1.25. $a=3$ için sinüs ve kosinüs aralığı için azalmanın gösterimi.

Sinüs ve kosinüs fonksiyonunun döngüsel modeli, bir çözümünün başka bir çözüm etrafında yeniden konumlandırılmasına izin verir. Bu, iki çözüm arasında tanımlanan alanın sömürülmesini sağlayabilir. Arama alanını keşfetmek için çözümlere karşılık gelen hedefleri arasındaki alanın dışında da arama yapabilmesi gerekir. Bu, Şekil 1.23'te gösterildiği gibi sinüs ve kosinüs fonksiyonlarının aralığını değiştirerek gerçekleştirilebilir (Mirjalili, 2016). Sinüs ve kosinüs fonksiyonlarının $[-2,2]$ aralığıyla etkilerinin kavramsal bir modeli, Şekil 1.24'te gösterilmiştir (Mirjalili, 2016).

Bu şekil, sinüs ve kosinüs fonksiyonlarının aralığının değiştirilmesinin kendi ile başka bir çözüm arasındaki alan dışındaki veya içindeki konumunu güncellemek için nasıl bir çözüm gerektirdiğini gösterir. İçeride veya dışarıda rastgele bir konum, “Eş. 1.38” de r_2 için $[0,2\pi]$ aralığında rastgele bir sayı tanımlayarak elde edilir. Bu nedenle bu sistem sırasıyla arama alanının keşfini ve sömürülmesini sağlar.

Bir algoritma, arama alanının optime edilebilecek bölgelerini bulmak için keşif ve sömürüyü dengeleyebilmeli ve nihayetinde küresel optimuma yaklaşabilmelidir. Keşif ve sömürüyü dengelemek için “Eş.1.38-1.40” eşitliklerinde sinüs ve kosinüs aralığı aşağıdaki eşitlik kullanılarak ayarlanabilir (Mirjalili, 2016).

$$r_1 = a - t \frac{a}{T} \quad (1.41)$$

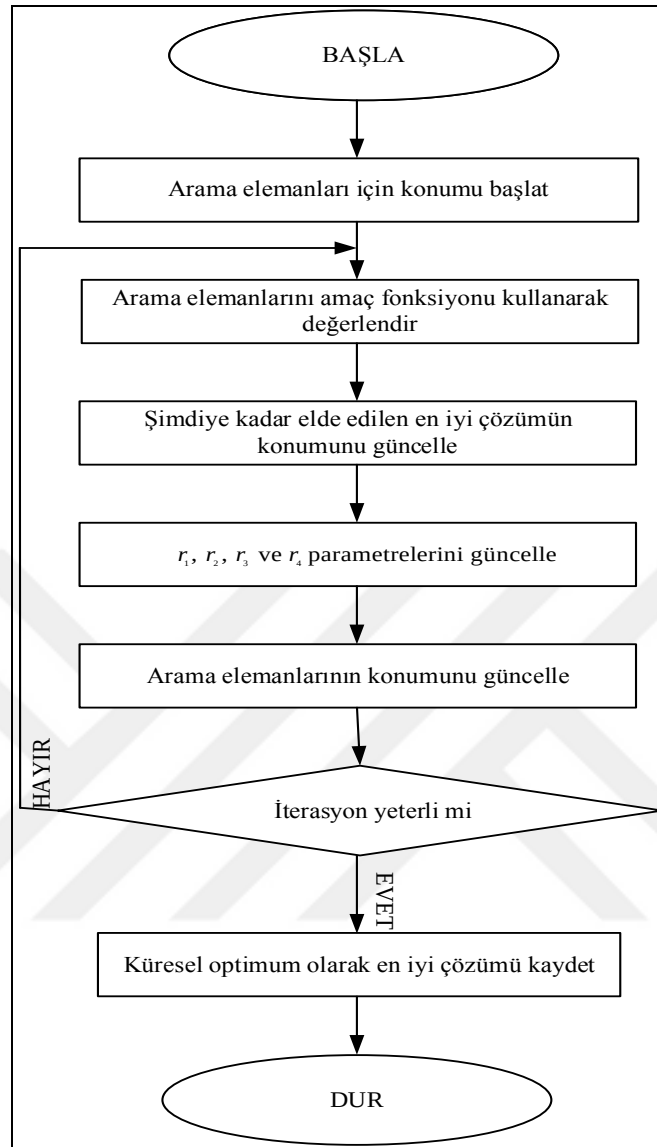
Bu eşitlikte t mevcut iterasyon, T en yüksek iterasyon ve a bir sabittir.

Şekil 1.25, bu eşitliğin, tekrarlar boyunca sinüs ve kosinüs fonksiyonları aralığını nasıl azalttığını göstermektedir (Mirjalili, 2016). Şekil 1.23 ve Şekil 1.24'de sinüs ve kosinüs fonksiyonlarının aralıkları $(1,2]$ ve $[-2, -1)$ olduğunda SCA

algoritmasının arama alanını keşfetmektedir. Ancak bu algoritma $[-1,1]$ aralığında olduğunda arama alanını sömürür.

SCA algoritması belirli bir problem için bir dizi rastgele çözüm oluşturur ve geliştirir, bu nedenle bireysel tabanlı algoritmalara oranla yüksek keşif ve yerel optimaya takılmama özelliğinden yararlanır.

Sinüs ve kosinüs fonksiyonları 1'den büyük veya -1'den küçük bir değer aldığı anda arama alanının farklı bölgelerini keşfeder. SCA algoritması, sinüs ve kosinüs fonksiyonlarında uyarlanabilir aralık kullanarak keşiften sömürüye sorunsuz bir şekilde geçer. Küresel optimumun en iyi değeri, hedef noktası olarak bir değışkende saklanır ve optimizasyon sırasında asla kaybolmaz. Çözümler her zaman şimdiye kadar elde edilen en iyi çözüm etrafındaki konumlarını güncellediğinden, optimizasyon sırasında arama alanlarının en iyi bölgelerine doğru bir eğilim vardır. Önerilen algoritma optimizasyon problemini kara kutu olarak gördüğünden uygun problem formülasyonuna tabi olan farklı alanlardaki problemlere kolayca dahil edilebilir (Mirjalili, 2016). SCA'da, her iterasyon aday çözüm kümesini rastgele günceller ve en iyi aday çözümü ezberler. Yeni güncellenen aday çözümün kaydedilmesi mevcut durumuna ve en iyi aday çözümün sağladığı yöneme bağlıdır. Bu arama mekanizması SCA'da yeterli keşif yeteneğini göstermekle beraber SCA'da düşük sömürü, gerçek çözümleri yakalayamamak, çözüm değerlerinde yüksek çeşitlilik ve sömürü ile keşif arasındaki yeterli olmayan denge gözlenmiştir (Gupta ve Deep, 2020). Şekil 1.26'da SCA algoritmasının işlem sırasını gösteren akış şeması gösterilmiştir (El-Shorbagy ve ark., 2019).



Şekil 1.26. SCA akış şeması.

1.1.9. Atom arama optimizasyon algoritması

Atom Arama Optimizasyon (Atom Search Optimization, ASO) algoritması, Newton kuralları tarafından belirlenen ve maddelerdeki atom hareketlerini taklit ederek modellenen fizik temelli bir algoritmadır (Zhao at All, 2019). “Eş.1.42” de diğer atomlarla bağlantılı bir şekilde i . atoma etki eden d boyutlu rastgele ağırlıklara sahip bileşenlerin toplamı, toplam kuvvet olarak ifade edilebilir ve $rand_j$ $[0,1]$ aralığında bir sayıyı temsil eder.

$$F_i^d(t) = \sum_{j \in K_{best}} rand_j F_{ij}^d(t) \quad (1.42)$$

ASO algoritmasında, her bir atom en iyi atomun belli bir sınırlandırıcı kuvveti tarafından kontrol edilir. Dolayısıyla herhangi bir i . atomun kısıtı $\theta_i(t)$ aşağıdaki eşitlikte ifade edilmiştir.

$$\theta_i(t) = [|x_i(t) - x_{best}(t)|^2 - b_{i,best}^2] \quad (1.43)$$

Bu eşitlikte $x_{best}(t)$ ifadesi, t . iterasyonda en iyi atomun konumu, $b_{i,best}$ ise i . atom ile en iyi atom arasındaki sabit bağ uzunluğudur. Dolayısıyla sınırlayıcı kuvvet aşağıda verildiği gibi ifade edilir.

$$G_i^d(t) = -\lambda(t) \nabla \theta_i^d(t) = -2\lambda(t) (x_i^d(t) - x_{best}^d(t)) \quad (1.44)$$

burada $\lambda(t)$ bir lagrange çarpanı, β çarpan ağırlığı olmak üzere;

$$\lambda(t) = \beta \exp\left(\frac{-20t}{T}\right) \quad (1.45)$$

dir. t mevcut iterasyon ve T en yüksek iterasyon sayısıdır. t . iterasyonda i . atomun kütlesi aşağıdaki eşitliklerde ifade edilmiştir.

$$M_i(t) = \exp\left(-\frac{Fit_i(t) - Fit_{best}(t)}{Fit_{worst}(t) - Fit_{best}(t)}\right) \quad (1.46)$$

$$m_i(t) = \frac{M_i(t)}{\sum_{j=1}^N M_j(t)} \quad (1.47)$$

burada $Fit_{best}(t)$ ve $Fit_{worst}(t)$ ifadeleri i . iterasyonda atomların en uygun değerleridir. $Fit_i(t)$ fonksiyonu t . iterasyonda i . atomun en uygun değeridir. $Fit_{best}(t)$ ve $Fit_{worst}(t)$ fonksiyonları ise aşağıda ifade edildiği gibi hesaplanır.

$$Fit_{best}(t) = \min_{i \in \{1,2,\dots,N\}} Fit_i(t) \quad (1.48)$$

$$Fit_{worst}(t) = \max_{i \in \{1,2,\dots,N\}} Fit_i(t) \quad (1.49)$$

Algoritmayı indirgemek için, $(t + 1)$. iterasyonda i . atomun hız vektörü v_i ve konum vektörü x_i aşağıda ifade edildiği gibi hesaplanır.

$$v_i^d(t + 1) = rand_i^d v_i^d(t) + a_i^d(t) \quad (1.50)$$

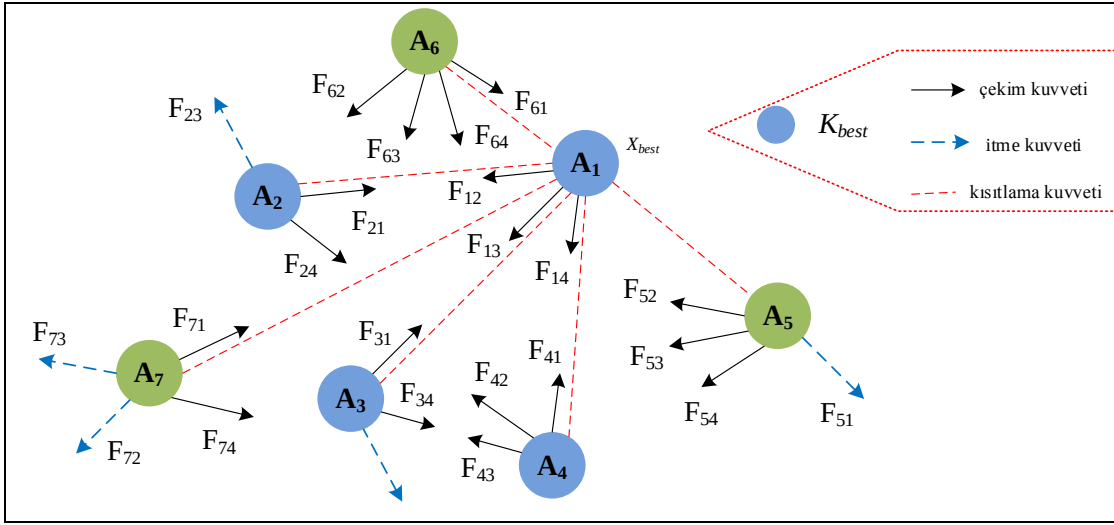
$$x_i^d(t + 1) = x_i^d(t) + v_i^d(t + 1) \quad (1.51)$$

bu ifadelerde $a_i^d(t)$ i . atomun t zamandaki eğimidir. Aşağıdaki eşitlikte ifade edildiği şekilde hesaplanır.

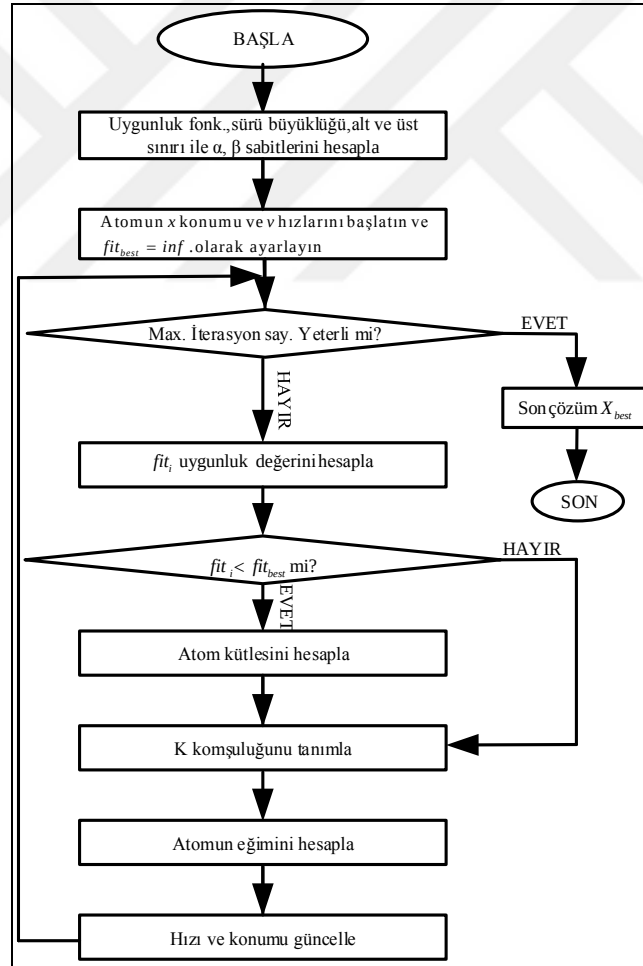
$$a_i^d(t) = \frac{F_i^d(t) + G_i^d(t)}{m_i^d(t)} = -\alpha \left(1 - \frac{t-1}{T}\right)^3 \exp\left(-\frac{20t}{T}\right). \\ \sum_{j \in Kbest} \frac{rand_j \left[2 \times (h_{ij}(t))^{13} - (h_{ij})^7\right] (x_i^d(t) - x_j^d(t))}{m_i(t) \|x_i(t), x_j(t)\|_2} + \beta \exp\left(\frac{-20t}{T}\right) \frac{(x_{best}^d(t) - x_i^d(t))}{m_i(t)} \quad (1.52)$$

ASO algoritmasında, iterasyona başlarken keşif aşamasını geliştirmek için, sürüdeki her atom, K komşuları kadar mümkün olan en iyi uygunluk değerine sahip birkaç atomla birbirlerini etkilemek zorundadır. İterasyonların son aşamasındaki sömürüyü optimize etmek için atomlar, K komşuları kadar mümkün olan en iyi uygunluk değerine sahip birkaç atomla birbirlerini etkilemelidir. Böylece, bir zaman fonksiyonu olarak iterasyonlar arttıkça K belli oranda azalır ve aşağıdaki eşitlikte belirtildiği gibi ifade edilir. Şekil 1.27 bir atom sisteminin kuvvetlerini gösterirken (Zhao ve ark., 2019) şekil 1.28 ise ASO algoritmasının işlem sırasını bir akış şeması halinde sunmuştur (Almagboul ve ark., 2019).

$$K(t) = N - (N - 2) \times \sqrt{\frac{t}{T}} \quad (1.53)$$



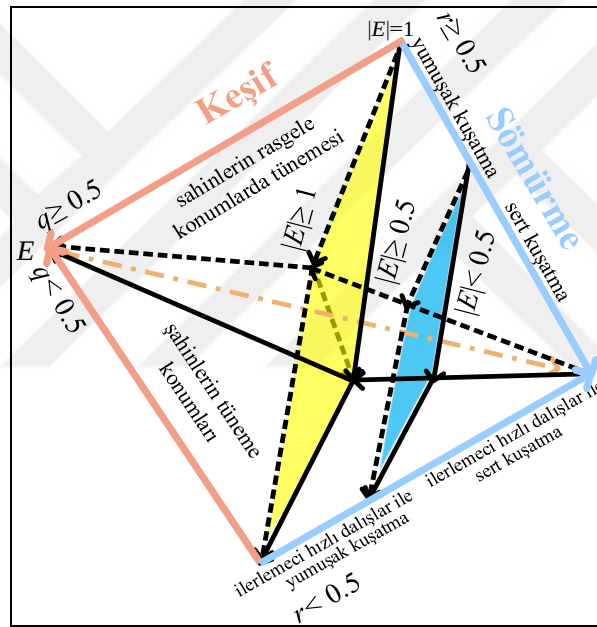
Şekil 1.27. Bir atom sisteminin kuvvetleri.



Şekil 1.28. ASO algoritma şeması.

1.1.10. Harris şahinleri optimizasyon algoritması

Harris Şahinleri Optimizasyon (Harris Hawks Optimization, HHO) algoritması, Amerika'nın bir bölgesinde topluluk halinde yaşayan şahinlerin avlanmak için gösterdikleri ve sürü davranışını baz alan sezgisel-üstü algoritmadır (Heidari ve ark., 2019). Şahinler tarafından gösterilen avın keşfi, sürpriz dalışlar ve farklı saldırı stratejileri HHO algoritmasında keşif ve sömürü aşamaları olarak modellenenecektir. HHO sürü bazlı bir algoritma olduğu için bu problemlere uygulanabilir. Aşağıdaki Şekil 1.29'da avlanma süreçlerinin tüm aşamaları gösterilmiştir (Heidari ve ark., 2019).



Şekil 1.29. HHO'nun tüm aşamaları.

Keşif aşamasında, Harris Şahinleri avı görmek için bulundukları bölgeyi bazen saatler süren bekleme ve gözlemlleme durumunda kalırlar. Bu nedenle her bir şahin HHO algoritmasında aday çözüm olarak tanımlanır ve her bir adımda amaçlanan av için en iyi aday çözüm, neredeyse optimuma yakındır. Keşif ya da inceleme aşaması olarak tanımladığımız bu aşamada şahinlerin avı saptamak için farklı konumlarda rastgele tüneme stratejileri q olasılığına eşitlenirse grubun belli alanda uzun ağaçların üzerinde tünemeleri $q > 0.5$, grubun ava (tavşana) karşı saldırıya geçebilecek uzaklıkta

tünemeleri ise $q < 0.5$ olarak modellenir. Keşif aşamasındaki bu strateji (1.54) eşitliğinde ifade edilmiştir (Heidari ve ark., 2019).

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1 |X_{rand}(t) - 2r_2 X(t)| & q \geq 0.5 \\ \{X_{av}(t) - X_m(t)\} - r_3(LB + r_4(UB - LB)) & q < 0.5 \end{cases} \quad (1.54)$$

(1.55) eşitliğinde $X_{av}(t)$ avın (tavşanın) konumunu, $X(t)$ harris şahinlerinin mevcut konum vektörünü, $X(t+1)$ bir sonraki iterasyondaki şahinlerin konum vektörünü, r_1, r_2, r_3, r_4 ve q ise her iterasyonda yenilenen (0,1) aralığındaki rastgele sayıları, LB ve UB değişkenlerin en alt ve en üst sınırlarını gösterir. $X_{rand}(t)$ mevcut popülasyondan rastgele seçilen şahini ve $X_m(t)$ ise şahinlerin mevcut popülasyonlarının ortalama konumlarını ifade eder ve (1.55) eşitliğinde ifade edilmiştir (Heidari ve ark., 2019).

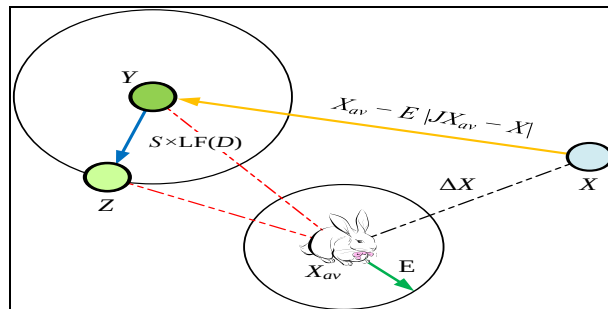
$$X_m(t) = \frac{1}{N} \sum_{i=1}^N X_i(t) \quad (1.55)$$

Bu eşitlikte $X_i(t)$ her şahinin t . iterasyondaki konumunu ve N ise şahin sürüsünün büyüklüğünü gösterir.

Keşiften sömürüye geçiş aşamasında, bir avın kaçış enerjisi aşağıdaki gibi modellenir (Heidari ve ark., 2019):

$$E = 2E_0 \left(1 - \frac{t}{T}\right) \quad (1.56)$$

Burada E keşiften sömürüye geçiş halinde avın (tavşanın) azalan enerjisini, avın E_0 enerjisinin ilk halini, T ise maksimum iterasyon sayısını ifade etmektedir.



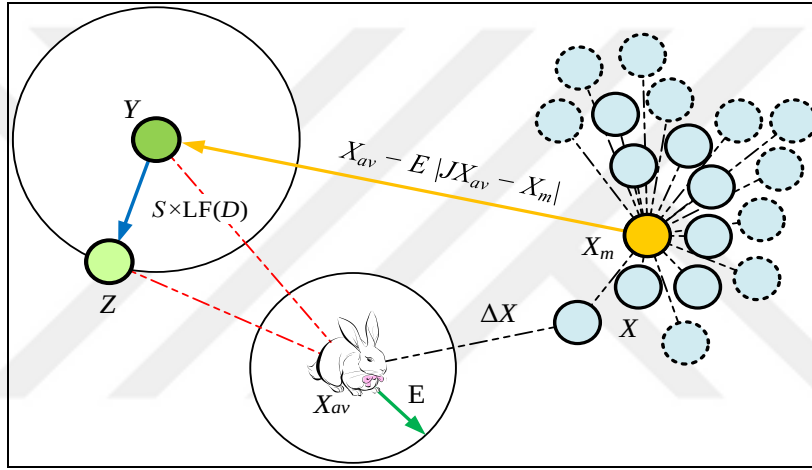
Şekil 1.30. İlerlemeci hızlı dalışlar ile yumuşak kuşatma hali.

Sömürü aşamasında, Şekil 1.30’da gösterilen sömürü halinde yumuşak kuşatma aşamasındaki şahinlerin davranışı aşağıdaki gibidir (Heidari ve ark., 2019):

$$X(t + 1) = \Delta X(t) - E|J \cdot X_{av}(t) - X(t)| \quad (1.57)$$

$$\Delta X(t) = X_{av}(t) - X(t) \quad (1.58)$$

Burada $\Delta X(t)$ t . iterasyonundaki avın konumu ile konum vektörü arasındaki farkı verir. r_5 , (0,1) aralığında rastgele bir değeri, $J = 2(1 - r_5)$ eşitliği ise avın enerjisi düştüğü anda rastgele zıplama gücünü gösterir.



Şekil 1.31. İlerlemeci hızlı dalışlar ile sert kuşatma hali.

Şekil 1.31’de gösterilen sömürü halinde sert kuşatma aşamasında mevcut pozisyonlar aşağıdaki gibi güncellenir (Heidari ve ark., 2019).

$$X(t + 1) = X_{tavşan}(t) - E|\Delta X(t)| \quad (1.59)$$

Şahinlerin yumuşak kuşatma için hızlı dalışlar yaparak bir sonraki hareket için karar vermeleri aşağıdaki eşitlik ile modellenir (Heidari ve ark., 2019).

$$Y = X_{tavşan}(t) - E|J \cdot X_{tavşan}(t) - X(t)| \quad (1.60)$$

Bu aşamadaki şahin dalışları Levy uçuşu (LF) modeline göre yapılır (Heidari ve ark., 2019).

$$Z = Y + S \times LF(D) \quad (1.61)$$

burada D problemin boyutunu ve S , $1 \times D$ boyutunda rastgele bir vektörü ifade eder. Lévy uçuşları, ya uzun süre arayla av arayan ya da farklı türlerin yiyecek aramalarında kullandıkları hareket modellerine iyi bir yaklaşımdır ve doğada Lévy uçuş süreçleri, birçok hayvan türünün geniş alanlar üzerinde hareket etmesini içermiştir (Rios ve ark., 2019). Levy uçuşu algoritmasını sömürü aşamasına ekleyerek, yerel optimum noktalarına takılmadan, yerel arama işlemine devam edilmesi sağlanmaktadır. Kullanılan E, r parametreleri ve LF sayesinde HHO keşif ve sömürü arasında oldukça başarılı bir denge elde etmiştir (Çelik ve ark., 2019).

Levy uçuşu fonksiyonunun tanımı aşağıdaki ifade edilmiştir (Heidari ve ark., 2019):

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \quad \sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}} \quad (1.62)$$

burada u ve v $(0,1)$ aralığında rastgele değerlerdir, β sabit bir sayıdır ve genellikle 1.5 alınır. Yumuşak kuşatmadaki şahinler durumlarını aşağıdaki eşitliğe uygun güncellerler (Heidari ve ark., 2019).

$$X(t+1) = \begin{cases} Y & , F(Y) < F(X(t)) \text{ ise} \\ Z & , F(Z) < F(X(t)) \text{ ise} \end{cases} \quad (1.63)$$

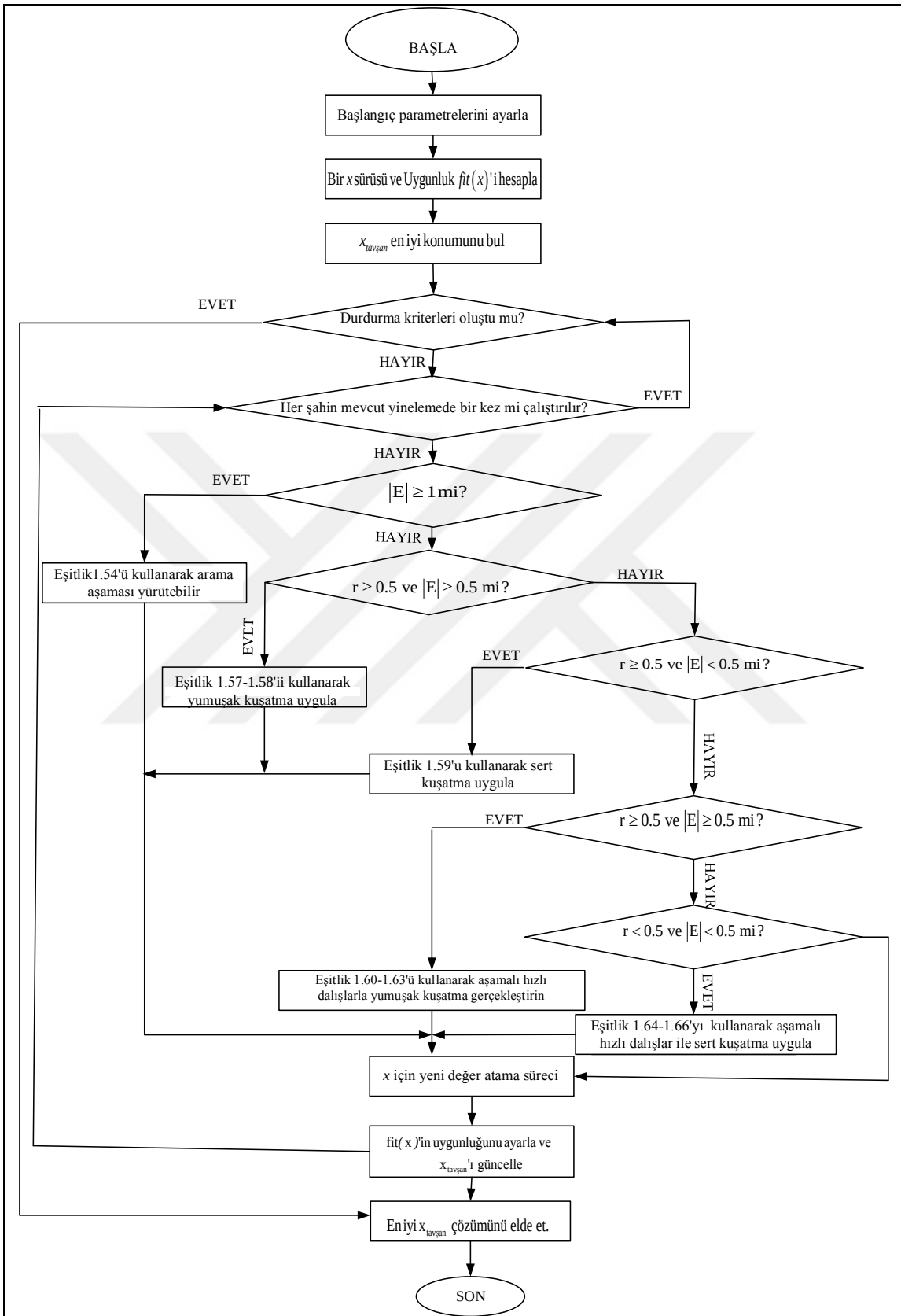
Şekil 1.30'da bu aşamasının basit gösterimi yapılmıştır. İlerlemeci hızlı dalışlar ile sert kuşatma aşamasında şahinler konumlarını aşağıda ifade edilen eşitlik ile günceller (Heidari ve ark., 2019).

$$X(t+1) = \begin{cases} Y & , F(Y) < F(X(t)) \text{ ise} \\ Z & , F(Z) < F(X(t)) \text{ ise} \end{cases} \quad (1.64)$$

Son eşitlikteki Y ve Z aşağıdaki yeni kurallara göre bulunur (Heidari ve ark., 2019).

$$Y = X_{tavşan}(t) - E|J \cdot X_{tavşan}(t) - X_m(t)| \quad (1.65)$$

$$Z = Y + S \times LF(D) \quad (1.66)$$

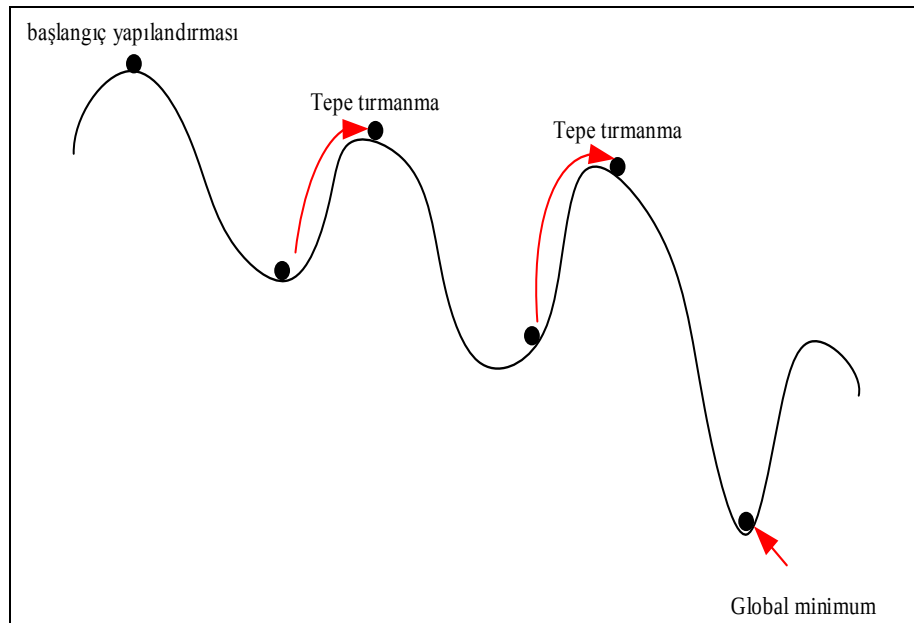


Şekil 1.32. HHO akış şeması.

1.1.11. hASO-SA hibrit algoritması

Bu çalışma, ASO'nun keşif ve sömürü aşamaları ile SA algoritmasının mükemmel yerel arama yeteneği arasındaki denge eksikliğini dikkate alarak yeni bir hibrit ASO ve SA (hASO-SA) algoritması önermektedir. Geliştirilen karma algoritma SA algoritmasını, her iki algoritmayı tek tek çalıştırmak yerine ASO algoritmasının gömülü bir parçası olarak benimser. Bu, SA'nın daha kötü çözümler için çalışmasına yardımcı olur, böylece komşuluk çözümlerinin potansiyeli ihmal edilmez. Önerilen hibrit algoritmanın kriter fonksiyonları için etkili bir optimizasyon algoritması ve çok katmanlı algılayıcı (MLP) eğitimi için yararlı olduğu gösterilmiştir.

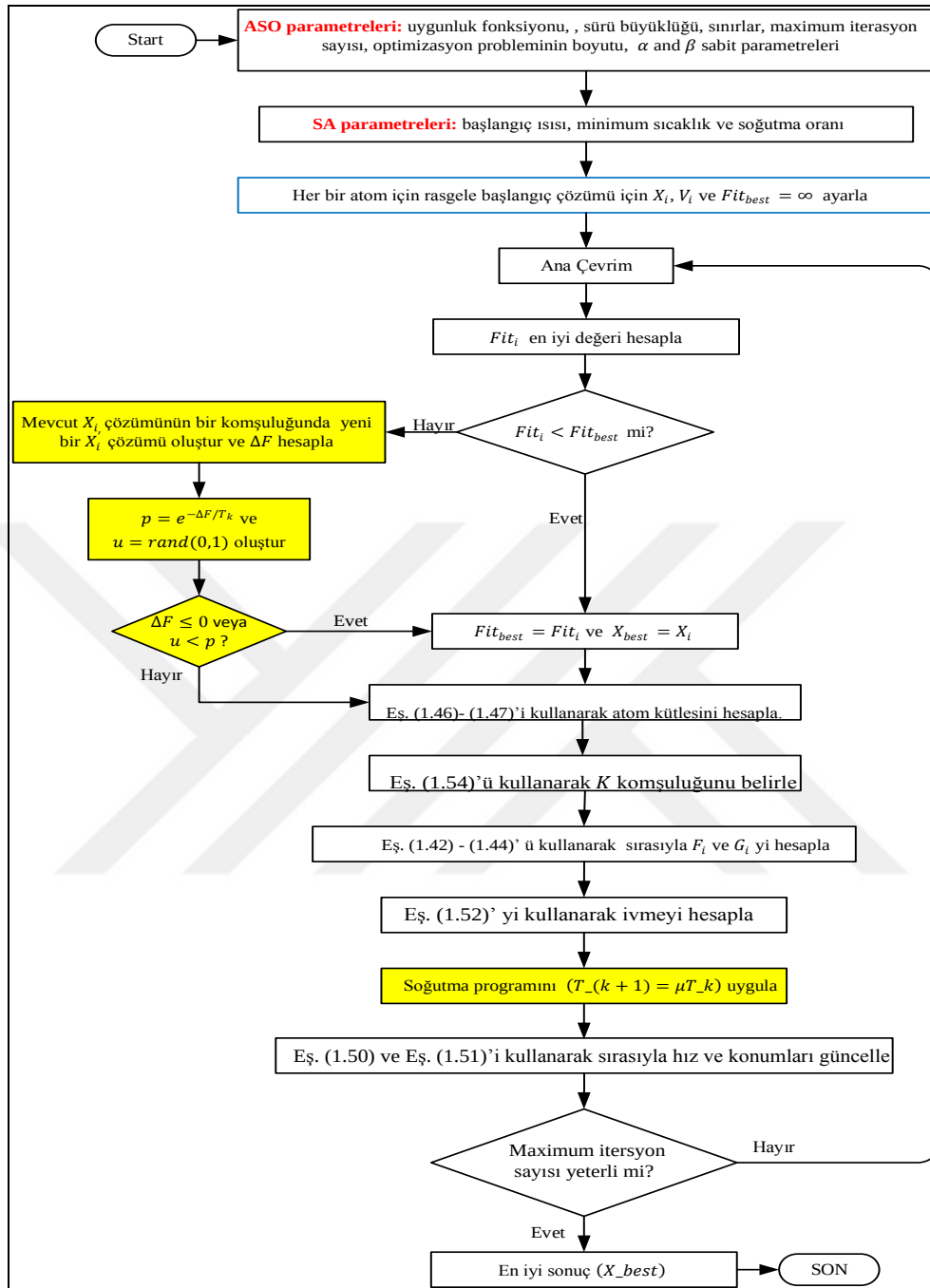
SA sürekli ve kesikli optimizasyon problemlerinin çözümünde yaygın olarak kullanılan bir yerel arama sezgisel-üstü algoritmadır. Tepe tırmanma hareketleri yoluyla yerel minimumlardan kaçma yeteneği, küresel bir çözüm arama açısından yararlı olan SA'nın ana avantajlarından biridir (Capparini 2019). SA bu özelliği ile ASO'ya yerel minimumlardan kaçınma konusunda yardımcı olabileceğinden bu hibrit yaklaşımın önerilmesine vesile olmuştur. Bunun yanı sıra, arama alanında optimum çözüm ararken keşfi güçlendirmeye de yardımcı olur. Şekil 1.33'te tepe tırmanma hareketini bir similasyonu mevcuttur.



Şekil 1.33. Tepe Tırmanma hareketi

Modern bir çalışma olan hibrid hASO-SA algoritması, sırasıyla, ASO ve SA algoritmalarının hızlı arama yeteneğinden ve tepeye tırmanma özelliğinden yararlanır ve farklı optimizasyon problemleri çözmeyi önerir.

Akış şemasında görülebileceği gibi hASO-SA algoritması, ASO ve SA algoritmalarının parametrelerini tanımakla beraber rastgele bir atom kümesi ile hızlarını ve sonsuzluğa ayarlanmış bir uygunluk değerini başlatır. Bu aşama gerçekleştiğinde, her bir atom için uygunluk değeri iteratif olarak hesaplanmaya başlanır, sonra bulunan uygun değer en iyi değerle karşılaştırılır. Daha iyi değerler olması durumunda, algoritma en iyi çözümü ve uygunluk değerini günceller ve akış şemasındaki adımların geri kalanı yürütülür. Bununla birlikte, önerilen hibrit algoritması, mevcut uygunluk değeri en iyi olandan daha iyi olmasa bile mevcut çözüme bir şans verir. Böyle bir durumda algoritma mevcut çözümün bir komşuluğunda yeni bir çözüm üretir ve yeni üretilen çözümü olasılığın gerekçesine göre değerlendirir. Bu, SA'nın ASO'nun yerleşik bir parçası olarak davrandığı ve daha iyi uygunluk değerleri olmadan mevcut çözüm durumunda komşuluk çözümünü kabul etmek için çalıştığı anlamına gelir. Gerekçeye dayanarak en iyi çözüm algoritma tarafından güncellenebilir veya güncellenmeyebilir. Böyle bir durumda SA, ASO'nun bir parçası olarak sadece en iyisinden daha kötü olan uygunluk değerleri için iyi çalışır, böylece herhangi bir potansiyel komşuluk sadece uygunluk değerine bakılarak doğrudan geçilmez. Şekil 1.34'te hASO-SA algoritmasının akış diyagramı verilmektedir. Akış diyagramında belirtilen değişiklikler orijinal ASO algoritmasından uyarlanarak hASO-SA algoritması önerilmiştir (Zhao, 2020).



Şekil 1.34. hASO -SA akış şeması.

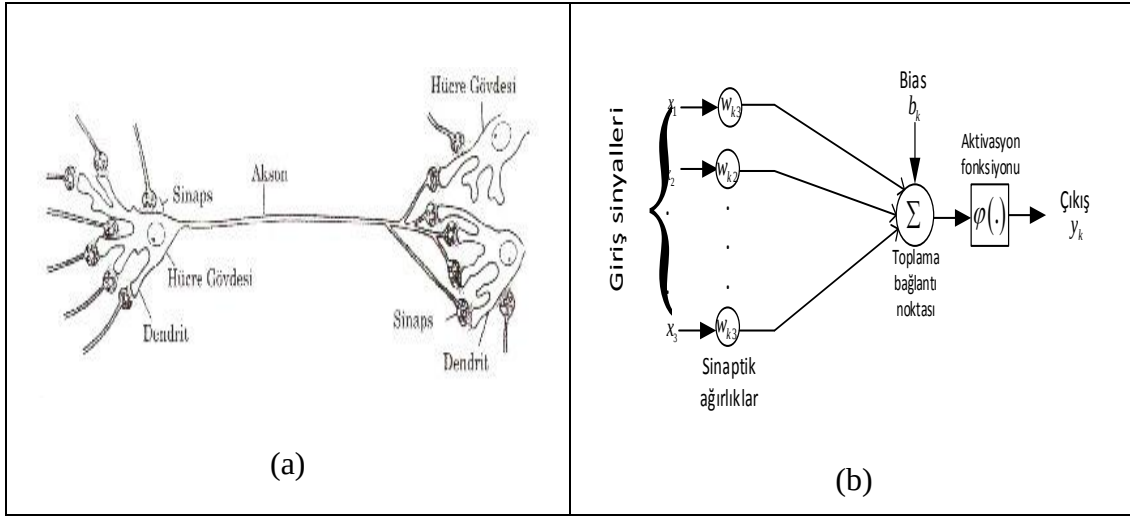
1.2. Yapay Sinir Ağları

İnsan beyni bireyin bilişsel, algısal ve kontrol görevlerini başarıyla gerçekleştirmesini sağlayan muazzam bir sinir ağına sahiptir. Beynin en büyük özelliği bir problemin çözümünde birden çok hesaplama kaynağını aynı anda

kullanması, sonsuz bilgi işleme yeteneğine sahip olması ve tüm özelliklerini etkin bir biçimde kullanmasıdır. İnsan beyni birbirine bağlı on milyardan fazla sinirin organizasyonundan meydana gelir. Her bir sinir (nöron) bilgi almak, işlemek ve iletmek için biyokimyasal reaksiyonları kullanan bir hücredir. Dentrit adı verilen ağaç dallarına benzeyen sinir lifleri, soma adı verilen hücrenin ana çıktısına bağlanır. Somadan uzanan akson olarak adlandırılan uzun lifler sinaps adı verilen küçük açıklıklardan diğer sinirlere bağlanır. Sinapslar bir sinirden diğerine sinyallerin aktarımına izin vererek belli iletiler maddelerin kesişme noktasının sonuna gönderilip hücrenin gövdesinde elektrik enerjisini artırma-azaltma etkisi yaratan karmaşık kimyasal bir süreç yürütür. Eğer bu elektrik potansiyeli bir eşiğe ulaşırsa aksondan bir sinyal gönderilir ve hücre “ateşlenir”. Sinir ağları vücuttan gelen sinyalleri beyne yollayarak beynin aldığı kararları vücuda eylem olarak iletir (Abraham, 2005).

Yapay sinir ağları (YSA) bu biyolojik sinir sisteminin matematiksel modeli geliştirilerek beyindeki mekanizmaya benzer şekilde bilgi işleme sistemine sahip bir yapıdır. Belirli problemleri çözmek için çok sayıda birbirine bağlı düğümden (sinirden) oluşan bir mimarisi vardır. YSA’ da bilgi, ağ aracılığıyla çevreden elde edilir ve sinirler arası bağlantı güçleri (ağırlıkları) ile birleştirilerek geliştirilip karara bağlanır (Haykin, 2005).

YSA yapay sinirlerden müteşekkil olup sinir hücresinde tanımladığımız her bir elemanın bir karşılığı bulunmaktadır. Sinir sisteminde toplama fonksiyonu (dentrid) girdi sinyallerini alır, aktivasyon fonksiyonu (soma) alınan sinyalleri işler, yapay sinir çıkışı (akson) girdi sinyallerini çıktı sinyallerine dönüştürür en son ağırlıklar (sinapslar) sinirler arası elektrokimyasal aktarımı sağlar. Üretilen çıktı, başka bir hücre ya da ağı kendi için tekrar girdi olabileceği gibi tatmin edici bir sonuçta olabilir.



Şekil 1.35. (a) Normal sinir hücresi, (b) lineer olmayan sinir hücresi modeli.

Şekil 1.35 (b)'de dışarıdan bir biasın (b_k) uygulandığı sinir modeli görülmektedir (Haykin, 2005). Bu modele aynı zamanda bir algılayıcı (percpetron) denir (Basheer ve Hajmer, 2000). Bias b_k , aktivasyon fonksiyonunun net girişinin pozitif ya da negatif oluşuna bağlı olarak sırasıyla azaltma ya da yükseltme etkisine sahiptir. Bir sinirin normalize edilmiş genişliği $[0,1]$ ya da $[-1,1]$ kapalı aralığıdır. Aşağıdaki eşitliklerde bir k sinirinin matematiksel modeli ifade edilmiştir.

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (1.67)$$

$$y_k = \phi(u_k + b_k) \quad (1.68)$$

verilen eşitliklerde $x_1, x_2, \dots, x_m$ giriş sinyalleri, $w_{k1}, w_{k2}, \dots, w_{km}$ k sinirinin sinaptik ağırlıkları, u_k giriş sinyallerine bağlı olarak liner birleştirici çıkış, b_k bias, $\phi(\cdot)$ aktivasyon fonksiyonu ve y_k sinirin çıkış sinyalidir (Haykin, 2005).

Aktivasyon fonksiyonu ($\phi(v)$)'nu üç tipte tanımlanabilir (Haykin, 2005). Şekil 1.36'da her bir fonksiyonun gösterimi mevcuttur.

a. Başlangıç değeri (threshold) fonksiyonu

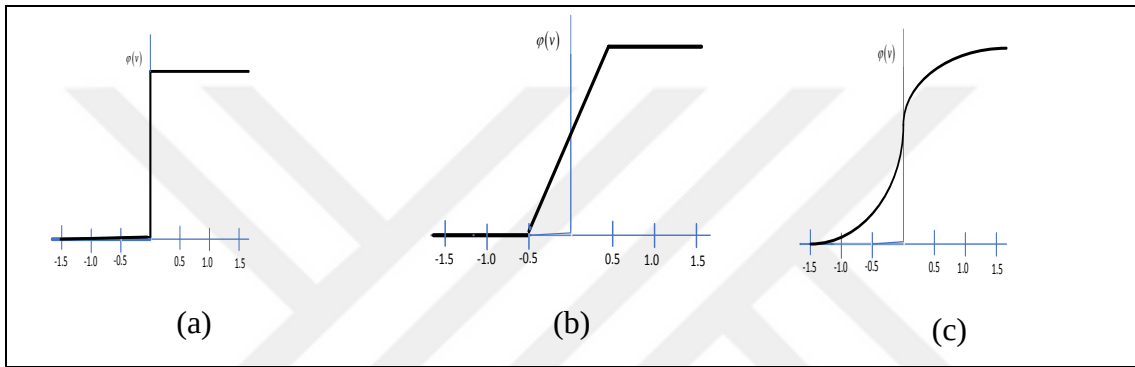
$$\phi(v) = \begin{cases} 1 & v \geq 0 \\ 0 & v < 0 \end{cases} \quad (1.69)$$

b. Parçalı doğrusal lineer (piecewise linear) fonksiyon

$$\varphi(v) = \begin{cases} 1, & v \geq \frac{1}{2} \\ v, & \frac{1}{2} > v > -\frac{1}{2} \\ 0, & v \leq -\frac{1}{2} \end{cases} \quad (1.70)$$

c. Sigmoid fonksiyonu

$$\varphi(v) = \frac{1}{1+\exp(-v)} \quad (1.71)$$



Şekil 1.36. Aktivasyon fonksiyonları.

Yukarıda sıralanan “Eş.1.69-1.71” eşitlikleri $[0,1]$ kapalı aralığında yer alır. Ancak aktivasyon fonksiyonunun bazen $[-1,1]$ aralığında olması istenir. Bu durum başlangıç noktasına göre simetrik olmadığından “Eş.1.69”, “Eş.1.72” ifadesine dönüşürken Signum fonksiyonu olarak ifade edilir. Sigmoid tipinde bir aktivasyon fonksiyonunun negatif değerler almasına izin verilmesi için bu form “Eş.1.72” de olduğu gibi hiperbolik tanjant fonksiyonu ile ifade edilebilir (Haykin, 2005).

$$\varphi(v) = \tanh(v) \quad (1.72)$$

$$\varphi(v) = \begin{cases} 1, & v > 0 \\ 0, & v = 0 \\ -1, & v < 0 \end{cases} \quad (1.73)$$

Şimdiye kadar elemanlarını tanıttığımız YSA yakın zamanda ortaya çıkan ve birçok disiplinde geniş kabul gören bilgisayarlı modelleme araçlarıdır. YSA’lar veri işleme ve bilgi gösterimi için büyük ölçüde paralel hesaplamalar yapabilen yoğun şekilde birbirine bağlı uyarlanabilir basit işleme elemanlarından oluşan yapılar olarak

tanımlanabilir (Hecht ve Nielsen, 1992). Yapay sinir ağlarının özellikleri aşağıda maddeler halinde sıralanmıştır.

a. Geleneksel model tabanlı yöntemlerin aksine, YSA'lar veri odaklı kendiliğinden uyarlanabilir yöntemlerdir, çünkü incelenen problemlere ilişkin modeller hakkında önceden var olan bir varsayım yoktur. Örneklerden öğreniyorlar ve temeldeki ilişkilerin bilinmemesi veya tanımlanması zor olsa bile veriler arasındaki ince fonksiyonel ilişkileri yakalıyorlar. Bu nedenle YSA'lar, çözümleri zor ve karmaşık bilgi gerektiren, ancak yeterli veri veya gözlem bulunan problemler için çok uygundur. Bu anlamda çok değişkenli, doğrusal olmayan ve parametrik olmayan istatistiki yöntemlerden biri olarak ele alınabilirler (White, 1989; Tan ve ark., 2018). Deneyimden öğrenme yeteneğine sahip bu modelleme yaklaşımı, birçok pratik problem için çok yararlıdır, çünkü veriye sahip olmak, verilerin üretildiği sistemleri düzenleyen temel yasalar hakkında iyi teorik tahminlere sahip olmaktan daha kolaydır.

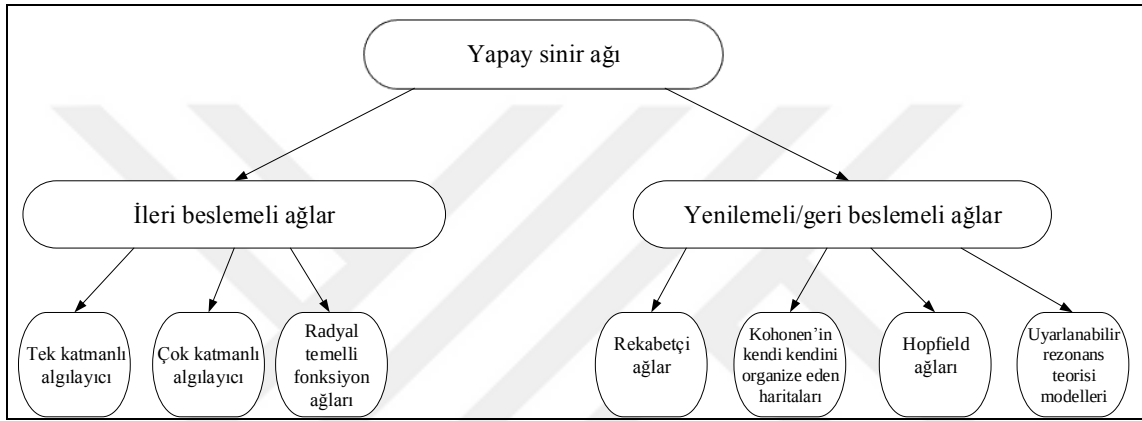
b. YSA genelleme yapabilir. Kendilerine sunulan verileri öğrendikten sonra, YSA'lar, örnek veriler gürültülü bilgi içeriyor olsa bile, genellikle bir sürünün görünmeyen kısmını doğru bir şekilde çıkarabilir. Öngörme, geçmişteki davranış örneklerinden gelecekteki davranışın (görülmeyen kısım) tahmini yoluyla gerçekleştirildiğinden en azından prensipte sinir ağları için ideal bir uygulama alanıdır (Tan ve ark., 2018).

c. YSA'lar evrensel fonksiyonel yaklaşımlardır. Bir ağın herhangi bir sürekli fonksiyonu istenen herhangi bir doğrulukta yaklaşık olarak tahmin edebildiği gösterilmiştir (Hornik, 1991; Chen, 2019). YSA'ların geleneksel istatistiki yöntemlerin etkili bir şekilde başa çıkabileceğinden daha genel ve esnek fonksiyonel formları vardır. Herhangi bir öngörme modeli, girdiler (zaman serisinin geçmiş değerleri ile diğer ilgili değişkenler) ve çıktılar (gelecekteki değerler) arasında temelde bilinen veya bilinmeyen bir ilişki olduğunu varsayar. Geleneksel istatistik tahmin modellerinin sıklıkla, gerçek sistemin karmaşıklığı nedeniyle bu temel işlevi tahmin etmede sınırlamaları vardır. YSA'lar bu işlevi tanımlamak için iyi bir alternatif yöntem olabilir.

d. YSA'lar doğrusal değildir. (Barron, 1989; Zhang ve ark., 1998).

1.2.1. YSA mimarisi ve eğitimi

Bir sinir ağının nöronlarının yapılandırılma biçimi, ağı eğitmek için kullanılan öğrenme algoritması ile yakından bağlantılıdır. Farklı ağ mimarileri uygun öğrenme algoritmaları gerektirir. Bu nedenle sinir ağlarının yapısı hangi öğrenme algoritmasının kullanılacağını da belirler. YSA mimarisi şekil 1.37’de yapay sinir ağlarının sınıflandırılması gösterilmiştir (Jain ve ark., 1996).

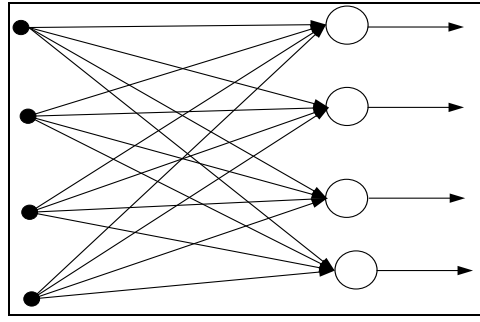


Şekil 1.37. YSA mimarileri sınıflandırılması.

YSA mimarileri en temelde ileri beslemeli ağlar ve yenilemeli geri beslemeli ağlar şeklinde ikiye ayrılır. İleri beslemeli ağları da tek katmanlı (SLP) ve çok katmanlı ağlar (MLP) ve radyal temelli fonksiyon ağları diye üç alt gruba ayrılabilir.

1.2.1.1. Tek katmanlı algılayıcılar

Bir sinir ağında düğümler katmanlar halinde dizilir. En basit ağda sinirleri çıkış katmanına yansıtılan ancak bunun tersi olmayan bir giriş katmanı vardır. Bu ağ türü ileri beslemeli bir ağıdır ve döngüsel değildir. Bu ağa çıkış katmanına atıfta bulunularak tek katmanlı ağ denmiştir. Giriş katmanında herhangi bir hesaplama yapılmadığı için dikkate alınmamıştır, şekil 1.38’de bir Tek Katmanlı Algılayıcı (Single Layer Perceptron, SLP) örneği gösterilmiştir (Haykin, 2005).



Şekil 1.38. Tek katmanlı sinir ağı.

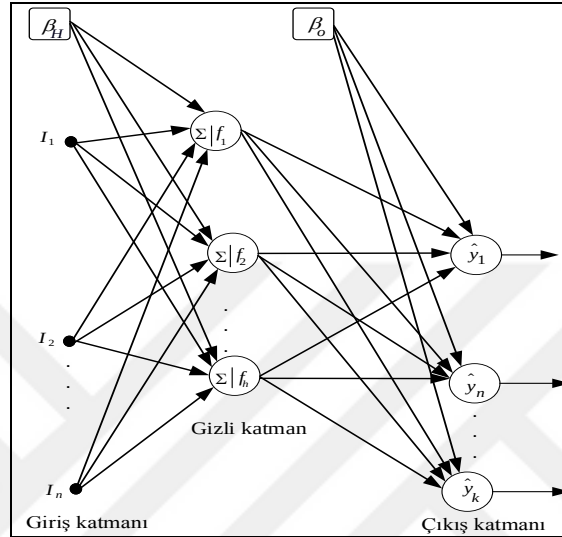
1.2.1.2. Çok katmanlı algılayıcılar

Birçok Katmanlı algılayıcı (Multilayer Perceptron, MLP), kaynak düğümleri de söylenen veri toplama düğümlerinin yer aldığı giriş katmanı, hesaplama düğümlerinin yer aldığı gizli katman ya da katmanları ve aynı şekilde hesaplama düğümlerinin yer aldığı çıkış katmanından oluşur.

Rosenblatt'ın perceptron kavramını (Rosenblatt, 1962) geliştirmesine rağmen 1970'ler boyunca YSA çalışmaları iki problemten dolayı kesintiye uğramıştır. Bunlardan birincisi gerçek dünya problemlerinin çözümünde yaşanan zorluklar ikincisi de Minsky ve Papert tek katmanlı bir ağı yalnızca doğrusal olarak ayrılabilen veri kümelerini sınıflandırabildiğini ve dolayısıyla XOR örneği gibi doğrusal olmayan problemleri çözemediğini göstermiş olmalarıdır (Minsky ve papert, 1969). XOR problemini çözmek için ağırlık ve eşik değerleri ayarlanmış bir MLP modeli oluşturulmuş ve farklı çözümler üretilmiştir (Murtagh, 1991; Singh, 2016, Ojha ve ark. 2017). MLP'nin doğrusal olmama (non-linearity), yüksek paralellik, gürültüye karşı sağlamlık, hataya dayanıklılık, öğrenme ve yüksek genelleştirme yetenekleri gibi üstünlükleri vardır ve MLP'ler, karmaşık ve doğrusal olmayan modelleme, sınıflandırma, tahmin ve örüntü tanıma gibi çok çeşitli problemlere başarıyla uygulanmıştır. (Faris ve ark., 2016).

MLP'yi eğitmek için danışmanlı ya da danışmansız eğitim stratejileri kullanılabilir (Aljarah ve ark. 2016). Danışmanlı eğitmenler de gradyan ve olasılık temelli yaklaşımlar olarak iki kategoride tanımlanabilir (Ojha ve ark. 2017). Danışmansız öğrenme (Murphy, 2012) daha çok Hopfield ağı (Kohonen, 1993) ve kümelemede kullanılmaktadır. MLP'ler çok iyi sınıflandırıcılar olarak kriter

fonksiyonlarının olasılık temelli algoritmalar yoluyla optimize edilmesini sağlayan araçlardır (Hemeida ve ark., 2020). Aşağıda Şekil 1.39’da bir MLP örneği olarak ileri beslemeli sinir ağı (Feedforward Neural Network, FNN) çizilmiştir (Faris ve ark., 2016).



Şekil 1.39. İleri beslemeli sinir ağı.

MLP ağlarında, katmanlardaki sinirler sayısal ağırlıklar ile tamamen birbirine bağlıdır, her sinir toplama ve aktivasyon fonksiyonları içerir. Toplama fonksiyonu, “Eş. 1.74” de gösterildiği gibi her bir girdi ve ağırlıkların çarpımı ile biasın toplamını özetler. Eşitlikte w_{ij} , I_i ’yi j ’ye bağlayan bağlantı ağırlığı, β_j bias, n giriş sinirlerinin toplam sayısıdır. Toplama fonksiyonunun çıktısı bir aktivasyon fonksiyonuna (transfer fonksiyonuna) bir giriş olacaktır. j sinirinin çıkışı “Eş. 1.76” da ifade edildiği gibidir.

$$S_j = \sum_{i=1}^n w_{ij} I_i + \beta_j \quad (1.74)$$

$$f(x) = \frac{1}{1+e^{-x}} \quad (1.75)$$

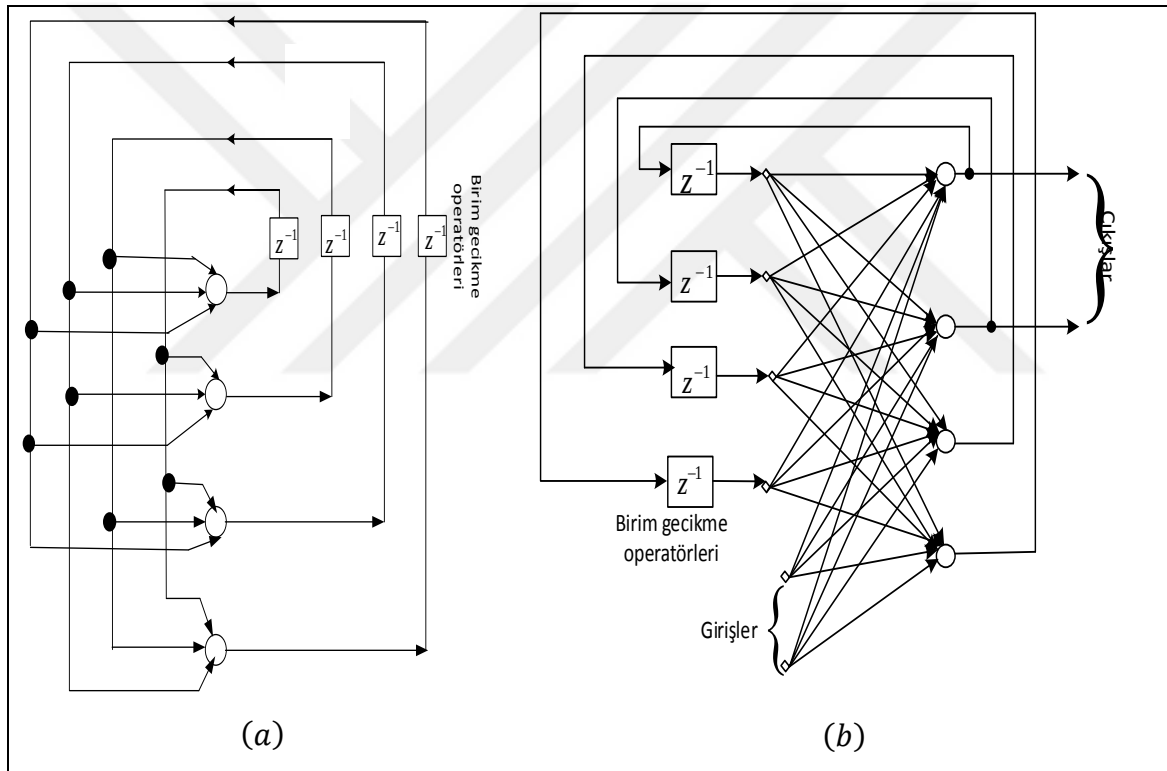
$$y_j = f_j(\sum_{i=1}^n w_{ij} I_i + \beta_j) \quad (1.76)$$

YSA’nın yapısı oluşturulduğunda, ağın parametrelerini (ağırlık seti) ayarlamak için bu modele öğrenme süreci uygulanır. YSA bağlamında bir öğrenme süreci, bir ağın belirli bir görevi etkili bir şekilde yerine getirebilmesi için ağ mimarisini ve bağlantı ağırlıklarını güncelleme problemi olarak görülebilir. Ağ genellikle bağlantı ağırlıklarını

mevcut eğitim modellerinden öğrenmelidir. Ağdaki ağırlıkları tekrar tekrar güncelleyerek performans zamanla artırılır. Ağın eğitim süreci, MLP ağının farklı türdeki problemleri çözmek için gücünü yansıtan zorlu bir görevdir. YSA eğitimi için farklı yaklaşımlar mevcuttur. Başlıca MLP eğitim yaklaşımlarından biri danışmanlı öğrenmedir. Danışmanlı öğrenmenin amacı, istenen ve hesaplanan çıktı değerleri arasındaki hatayı en aza indirmektir. Geri yayılım olarak adlandırılan Gradient tabanlı bir yöntem, ortak danışmanlı öğrenme algoritmaları olarak kabul edilir. MLP, geri yayılma algoritması kullanılarak örüntü tanıma ve sınıflandırma problemlerine uygulanır. Bu algoritmanın temel amacı, ağdaki tüm ağırlıkları hesaplayarak tahmin hatasını en aza indirmektir. Bu amacı gerçekleştirmek için iki adım uygulanır. Bunlar yayılma ve ağırlık güncellemesidir (Kayri, 2015). Geri yayılım algoritması en popüler Gradyan temelli eğitimcidir ve MLP ağını eğitmek için geleneksel bir metot olarak dikkate alınmıştır. Araştırmacılar arasında popüler olmasına karşın Gradyan temelli eğitimcinin, yerel optimizasyona takılıp kalma, yavaş yakınsama ve parametrelere güçlü güven gibi dezavantajları (Faris ve ark., 2016; Mirjalili, 2015) olduğundan dolayı genellikle yerel arama algoritmaları olarak kullanılabilir (Rumelhart ve ark., 1986; Faris ve ark., 2016). Bununla beraber küresel optimum seviyeye ulaşmak için, herhangi bir optimize edicinin keşif ve sömürü gibi iki temel kapasite arasında ince bir denge kurulabilmesi gerekir. Keşif, uygunluk durumundaki yeni ve bilinmeyen bölgeleri keşfetmek için hayati öneme sahiptir ve daha önce keşfedilen konumların avantajlarından yararlanmak için sömürü gereklidir (Heidari ve ark. 2017; Faris ve ark., 2018). Bu bağlamda, MLP ağlarının eğitimi için çeşitli olasılıksal optimizasyon yöntemlerinin gerekliliği literatürde tanınmıştır (Ojha ve ark., 2017). Bu arama yöntemlerinin avantajı, genellikle yerel bir optimadan ziyade küresel bir optima bulmaları yönüyle gradyan temelli yaklaşımlardan daha iyi olmalarıdır. Geleneksel yöntemlerin aksine sezgisel-üstü algoritma tabanlı eğitimler yerel optimizasyondan kaçınmada yüksek verimlilik gösterir (Aljarah ve ark., 2016; Faris ve ark., 2016). Tüm bu özelliklerinin yanı sıra, problem hakkında önceden bilgi gerektirmezler ve genel olarak uygulanması daha kolay olmakla beraber hesaplama zamanı bazen dezavantaj oluşturmaktadır (Kowalski ve Lukasik, 2016).

1.2.1.3. Yenilemeli ağlar

En az bir geri besleme döngüsüne sahip ağa yenilemeli (recurrent) sinir ağı denir. Geri besleme döngüleri ağın performansı üzerinde oldukça büyük etkiye sahiptir. Ayrıca, geri besleme döngüleri, sinir ağının doğrusal olmayan birimler içerdiği varsayılarak doğrusal olmayan bir dinamizm davranışı ile sonuçlanan z^{-1} ile gösterilen birim geciktirme elemanlarından oluşan belirli dalların kullanımını içerir. (Haykin, 2005). Kohonen ağları en güzel örneklerden birini oluşturur (Kohonen, 1993). Aşağıda Şekil 1.40'ta yenilemeli ağ örnekleri gösterilmiştir (Haykin, 2005).



Şekil 1.40. (a) yinelemeli ağ, (b) Gizli katmanlı yinelemeli ağ.



2. KAYNAK BİLDİRİŞLERİ

2.1. YSA'nın Tarihi Gelişimi

Nelson ve Illingworth (Nelson ve Illingworth, 1990), William James'in beyin aktiviteleri üzerindeki çalışmasını YSA ile ilgili çalışmaların başlangıcı olarak kabul ederler (James, 1890). Genel kanaate göre ise YSA araştırmalarının aktif olarak yapıldığı dönemleri üçe ayırmak mümkündür. Birincisi, ön açıcı çalışmaları nedeniyle 1940'larda McCulloch ve Pitts elektrik devrelerinden faydalanarak ilk yapay sinir ağı hücrelerini oluşturdular (McCulloch ve Pitts, 1943). İkincisi 1960'larda Rosenblatt'ın algılayıcı (perceptron) yakınsama teoremi (Rosenblatt, 1962) ile bir ya da daha fazla çıkışı olan her bir giriş için bir ağırlık ve bias kümesi olan üç katmanlı bir yapı oluşturdu, Minsky ve Papert'in basit bir perceptronun sadece çok sınırlı lineer olarak ayrılabilir problemleri çözebileceğini gösteren çalışması bu alandaki etkinliğin azalmasına neden oldu (Minsky ve Papert, 1969). Üçüncü aşamada Hopfield'in 1982'de enerji yaklaşımı (Hopfield, 1982) ve Werbos'un ilk kez 1974'de dile getirdiği MLP için geri yayılım algoritması (Werbos, 1974) Rumelhart ve arkadaşları tarafından 1986'da yeniden tanımlanmasıyla (Rumelhart ve McClelland, 1986) YSA çalışmaları yeni bir ivme kazanmıştır (Jain ve Mao, 1996). 1988'de radial temelli Fonksiyon modelleri (Broomhead ve Lowe, 1988) geliştirilmişken, 1991'de stokastik (olasılıksal) sinir ağıları geliştirilmiştir.

2.2. Sezgisel-Üstü Algoritmaların YSA Eğitiminde Kullanım Tarihçesi

Tarihi arka planı incelendiğinde, 1940'lardan 1960'lara kadar sezgisel algoritmalar farklı uygulamalarda kullanıldıysa da temel başarı 1960'ların başından itibaren I. Rechenberg ve H. Schwefel tarafından evrimsel algoritmaya girişle sağlandı (Rechenberg, 1965; Schwefel, 1965). 1975'te J. Holland Genetik Algoritma (GA)'yı geliştirdi (Holland, 1975; 1992a). 1983'te S. Kirkpatrick ve arkadaşları tarafından Benzetilmiş Tavlama (SA) algoritması (Kirkpatrick ve ark., 1983) ve Farmer ve arkadaşları tarafından yapay bağışıklık sistemi geliştirilerek bu alanda en büyük adım atılmış oldu. 1986'da F. Glover ve M. Laguna tarafından Tabu Arama (TS) geliştirilerek

ilk kez sezgisel-üstü algoritmada bellek kullanıldı (Glover ve Laguna, 1986). 1995’de L.M. Gambardella ve M. Dorigo tarafından başlatılan çalışmalar (Gambardella ve Dorigo, 1995) ile beraber 1999’ da Karınca Kolonisi Optimizasyon (ACO) algoritması (ACO) (Dorigo, 1999) ve 1995 yılında J. Kennedy ve R.C. Erberhart tarafından Parçacık Sürü Optimizasyon (PSO) algoritması (Kenedy ve Errberhart, 1995) ve 1997’de de R. Storn ve K. Price tarafından Diferansiyel Evrim (DE) algoritması geliştirildi. 2000’lerden günümüze kadar ise bu çalışmalar pik yaparak yüzellinin üzerinde sezgisel-üstü algoritma geliştirildi (Almufti, 2019; Khalilpourazari ve Khalilpourazari, 2019). Klonal Seçim Algoritması (CSA) (De Castro ve Von zuben, 2000), Uyum Arama (HS) algoritması (Geem ve ark., 2001), Bakteri Arama (BO) algoritması (Passino, 2002), Biyocoğrafyaya Dayalı Optimizasyon (BFO) algoritması (Simon, 2008), Guguk Kuşu Arama (CS) algoritması (Yang ve Deb, 2009), Yerçekimi Arama Algoritması (GSA) (Rashedi ve ark., 2009), Ateş Böceği (FA) algoritması (Yang, 2009), Rüzgâr Güdümlü Optimizasyon (WDO) algoritması (Bayraktar ve ark., 2010), Göçmen Kuşlar Optimizasyon (MBO) algoritması (Duman ve ark., 2011), Krill Sürüsü (KH) algoritması (Gandomi ve Alavi, 2012), Sosyal Örümcek Algoritması (SSA) (Cuevas ve ark., 2013), Karınca Aslanı Optimizasyon (ALO) algoritması (Mirjalili, 2015), Balina Optimizasyon Algoritması (WOA) (Mirjalili, 2016), Salp Sürüsü Optimizasyon (SSA) algoritması (Mirjalili ve ark., 2017), Tarım Arazisi Verimlilik Algoritması (FFA) (Shayanfar ve Gharehchopogh, 2018) ve Denge Optimizasyon (EO) (Faramarzi ve ark., 2020) algoritmaları bu çalışmalara örnek verilebilir.

Sezgisel-üstü optimizasyon algoritmalarının performanslarının kapsamlı bir şekilde karşılaştırılması ile ilgili de son yıllarda birçok çalışma mevcuttur (Wang, 2018; Eker ve ark., 2019a; Cheng ve ark., 2019; Jaddi ve Abdullah, 2020). Son yıllarda özellikle mühendislik çalışmalarında optimizasyon problemlerini çözmek için sezgisel-üstü algoritmalar hatırı sayılır bir yer tutmaya başladı (Farmer ve ark., 1986; Ekinci ve ark., 2019; Blackstock ve ark., 2018).

YSA’da sezgisel-üstü algoritmaların yaygın olarak kullanılmasının en temel nedenlerinden biri doğrusal olmayan problemlerde kullanılabilmesi (Bansal ve ark., 2020), MLP’nin gizli katmanlı ileri beslemeli sinir ağlarının bir parçası olması (Haykin, 1999) ve aynı zamanda yaygın olarak benimsenen bir sinir ağı olması ve belli

uygulamalar için kullanılmasında eğitime ihtiyaç duymasıdır (Gupta ve Keep, 2020). Sezgisel-üstü algoritmalar'ın YSA eğitiminde başarılı bir şekilde kullanımı ile ilgili örnekler de literatürde ciddi bir yere sahiptir. GA (Kolhe ve Maunuksela, 2011), DE (Bas, 2016), ACO (Eroğlu ve Seçkiner, 2019), ABC (Ghaleini ve ark., 2019), SA (Bahrami ve Ardejani, 2016), PSO (Wang ve ark., 2016), SCA (Bairathi ve Gopalani, 2017), PSO (Ashraf ve ark., 2020), HHO (Bui ve ark., 2019), GWO (Emary ve ark., 2017), WDO (Bayraktar, 2019), ASO (Agwa ve ark., 2019) birer eğitim algoritması olarak örnek verilebilir. YSA eğitim süreci, hata miktarını minimal düzeye indirmek için bir dizi ağırlık bulma amaçlı optimizasyon görevidir. Sinir ağı eğitimi için kullanılan algoritmalar açısından literatürde Geri Yayılımlı Öğrenme Algoritması (Back Propagation Algorithm, BP) (Rumelhart ve ark., 1986) ve deterministik yaklaşımlar da kullanılmaktadır (Suratgar ve ark., 2005). Ancak çalışmalar bu algoritmaların yavaş yakınsama hızı ve yerel optimada takılma ve başarı koşullarına aşırı duyarlılık gibi dezavantajlara sahip olduğunu vurgulamışlardır (Azzini ve Tettamanzi, 2011). Minggiang ve Gaoyang DE sezgisel-üstü algoritması ile BP algoritmasını karşılaştıran bir çalışma ile bu dezavantajları göstermişlerdir (Mingguang ve Gaoyang, 2009). Bu nedenle, bu tür bir yapının eğitimi, sayılan sorunların üstesinden gelmek için daha iyi bir algoritma gerektirir. Bunu yapmak için şimdiye kadar birkaç sezgisel-üstü algoritma önerilmiştir. Bu algoritmalarından bazıları Gri Kurt Optimizasyon (GWO) (Mirjalili, 2015; Faris ve ark., 2019) algoritması ve Geliştirilmiş Gri Kurt Optimizasyon (IGWO) algoritması (Zhang ve ark., 2020), Karınca Aslanı Optimizasyon (ALO) algoritması (Heidari ve ark., 2020), Şempanze Optimizasyon Algoritması (ChOA) (Khishe ve Mosavi, 2020), Çekirge Optimizasyon Algoritması (GOA) (Heidari ve ark., 2019), Salp Sürüsü Algoritması (SSA) (Khishe ve Mohammadi, 2019) ve Çoklu Lider Salp Sürüsü Optimizasyon Algoritması (MLSSA) (Bairathi ve Gopalani, 2019) olarak sıralanabilir. Bu algoritmaların yanısıra hibrit algoritmalar da hybrid Nelder-Mead ve Yusufçuk Algoritması (INMDA) (Xu ve Yan, 2019), hibrit Parçacık Sürüsü Optimizasyonu ve Yerçekimi Araması Algoritması (PSOGSA) (Mirjalili ve ark., 2012), Hibrit Monarch Kelebek ve Yapay Arı Kolonisi (HAM) algoritması (Ghanem ve Jantan, 2020) örnekleri verilebilir.



3. MATERYAL VE YÖNTEM

Bu bölümde sezgisel-üstü algoritmalar yoluyla yapay sinir ağlarının eğitimi ile ilgili olarak kriter fonksiyonları ve veri setleri verilecektir. Yöntem olarak bir taraftan Sezgisel-üstü algoritmaların kriter fonksiyonları aracılığıyla performans başarıları MATLAB programı vasıtasıyla yapılacak, diğer taraftan YSA' nın sezgisel-üstü algoritmalar vasıtasıyla eğitimi gerçekleştirilecektir. Bu eğitimde tanıtılacak olan veri setleri aracılığıyla modern sezgisel-üstü algoritmaların sınıflandırma becerilerine bakılacaktır.

Tezde ilk çalışma, HHO algoritması ile sürü tabanlı ABC (Karaboğa ve ark., 2005) ile fizik tabanlı WDO (Bayraktar ve ark., 2010) ve ASO (Zhao ve ark., 2019) algoritmalarının Rosenbrock, Schwefel, Easom, Egg Crate, Ackley, Sphere tkriter fonksiyonları kullanılarak sınıflandırma becerilerindeki performansları karşılaştırılmıştır (Eker ve ark., 2019b). Ayrıca açıklayıcı veri analizi (Mc Gill ve ark., 1978) yöntemiyle kutu grafikleri kullanılmıştır.

İkinci çalışma ASO, ABC, PSO ve SCA sezgisel-üstü algoritmaları aracılığıyla Ackley, Egg Crate, Griewank, Rastrigin, Rosenbrock, Schwefel, Sphere, Step kriter fonksiyonları optimize edilerek ortalama, standart sapma ve optimum değere en yakın değer gibi istatistiki sonuçlara ulaşılmıştır. PSO için (Yarpiz Team, 2019), ABC için (Karaboga, 2019), SCA için (Mirjalili, 2019) ve ASO için (Zhao, 2019) çalışmalarında kullandıkları MATLAB tabanlı kodlardan değişiklikler yapılmak suretiyle yararlanılmıştır. Bu çalışmanın amacı seçilen bu algoritmaların kısıtlı optimizasyon problemleri (Joines, Houck, 1994) çözmedeki başarılarının karşılaştırılması ve yakınsama eğrileri göz önüne alınarak bu kriter fonksiyonlarının optimizasyon süreçlerini değerlendirmektir (Eker ve ark., 2019a).

Üçüncü çalışmada MLP eğitimi için eğitimci olarak HHO ve SCA sezgisel-üstü algoritmaları seçilerek eğitimcilerin performansı UCI veri kümesinden alınan XOR, Heart, Iris, Balloon ve Breast Cancer veri setleri kullanılarak test edildi. Bu çalışmanın amacı HHO'nun bir MLP sınıflandırıcısı olarak kullanılması ve başarısının SCA algoritması ile karşılaştırılmasıdır. Sınıflandırma veri kümeleri ve özelliklerini gösteren çizelge daha önce literatürde kullanılmıştır (Mirjalili, 2015a).

Dördüncü çalışmada önerilen hASO-SA hibrit sezgisel-üstü algoritması ile iki uygulama yapılmıştır. Birinci uygulamada hASO-SA hibrit algoritması PSO, GA, SA, GSA, WDO ve ASO algoritmaları ile istatistiki yönden karşılaştırılmıştır. Mukayese için Sphere, Rosenbrock, Step, Quartic, Schwefel, Rastrigin, Ackley ve Griewank kriter fonksiyonları kullanılmıştır. İkinci uygulamada ise önerilen hASO-SA algoritması aracılığıyla MLP eğitimi sağlanmıştır. Çalışmada MLP eğitiminde ağırlık ve biasların optimal bağlantılarını bulmak amaçlanmıştır. hASO-SA MLP hatasını en aza indirmek için bir öğrenme algoritması olarak değerlendirilmiştir. Ağırlık ve biaslar vektör metodu ile ifade edilirken amaç fonksiyonu olarak MSE kullanılmıştır. Algoritmaların sınıflandırma performanslarının karşılaştırılması ise UCI'den alınan beş sınıflandırma veri seti (XOR, Ballon, İris, Heart, Meme Kanseri) kullanılarak yapılmıştır (Blake ve Merz, 1998). Önerilen hASO-SA'nın performansı, bu sınıflandırma problemlerini çözmeye çalışan klasik ASO, GWO, PSO, ACO, ES ve PBIL algoritmaları gibi çeşitli algoritmalar ile karşılaştırılarak değerlendirilmiştir (Mirjalili, 2015). hASO-SA algoritması kullanılarak bir MLP eğitimi için karşılaştırılan algoritmalar (Gupta ve Deep, 2020) den alınmıştır.

3.1. Sezgisel-üstü Algoritmaların Optimizasyonunda Kullanılan Bazı Kriter Fonksiyonları

Sezgisel-üstü herhangi bir algoritmanın performansını ölçmek için kriter ,kaliteya da test fonksiyonları adıyla isimlendirilen fonksiyonlar uygulanır. Fonksiyonlar ayrıca algoritmaların güvenilirlik, verimlilik ve geçerlilik ile duyarlılık, yakınsama hızı, direnç (sağlamlık) ve genel performansı gibi kalitesini ortaya koyan özelliklerini test eder. Kriter fonksiyonu optimizasyonunu içeren birçok araştırma, büyük bir test kümesindeki farklı algoritmaları karşılaştırmaktadır. Bir algoritmayı değerlendirmek için, diğer algoritmalarla oranla daha iyi performans gösterdiği fonksiyonları tanımlamak gerekir. Kullanılan kriter fonksiyonlarını, tek modlu, çok modlu, ayrılabilirlik ve çok boyutluluk gibi özelliklerini baz alacak şekilde tasnif edilebilir (Jamil ve ark, 2013; Alatas ve ark., 2009; Zhang ve ark., 2016).

Problem çözme tekniğinin güvenilirliği ve doğrulanması, kıkriter fonksiyonları üzerinde test edilerek gerçekleştirilir (Innocente, 2010). Fonksiyonların özellikleri

irdelenirse birden fazla yerel optimum bulunan fonksiyona multimodal (çok modlu) denir. Bu fonksiyonlar, bir algoritmanın herhangi bir yerel minimumdan kaçma yeteneğini test etmek için kullanılır. Algoritmalar bir arama işlemi sırasında bu yerel optimum noktalarıyla karşılaşır, algoritmanın bu noktalardan birinde sıkışabilme eğilimi vardır. Bir algoritmanın yerel minimumda takılma nedeni keşif sürecinin kötü tasarlanmış olmasıdır çünkü keşfin amacı arama alanının etkili bir şekilde taranmasını sağlamaktır. Tek modlu kriter fonksiyonlarının bir küresel optimum özelliği vardır, bu nedenle algoritmaların yerel kullanım kabiliyetini karşılaştırmak için uygundurlar. Bu tür fonksiyonlar, nihai sonuçlardan ziyade test edilen algoritmaların yakınsama oranlarına daha fazla odaklanmaya izin verecektir. Çok boyutlu kriter fonksiyonlarının birçok yerel minimumu vardır ve yerel optima sayısı problem boyutu ile genellikle katlanarak artar, bu nedenle algoritmaların küresel keşif kabiliyetini karşılaştırmak için uygun olmakla beraber, çoğu algoritma için en zor problem sınıfları arasındadır (Zhang ve ark., 2016). Ayrıca düz yüzeyli fonksiyonlar da algoritmalar için bir zorluk oluşturur, çünkü fonksiyonun düzlüğü algoritmaya arama sürecini minimuma yönlendirecek herhangi bir bilgi vermez. Başka bir test problem grubu, ayrılabilir ve ayrılmaz fonksiyonlarla formüle edilir. Ayrılabilirlik, farklı karşılaştırma fonksiyonlarının zorluğunun bir ölçüsüdür. Genel olarak, ayrılabilir fonksiyonların ayrılmaz fonksiyonlara oranla çözülmesi nispeten kolaydır, çünkü bir fonksiyonun her değişkeni diğer değişkenlerden bağımsızdır. Tüm parametreler veya değişkenler bağımsızsa, bir dizi n bağımsız optimizasyon işlemi gerçekleştirilebilir. Sonuç olarak her değişken ya da parametre bağımsız olarak optimize edilebilir. Arama alanının boyutsallığı problemle ilgili önemli bir konudur. Bazı fonksiyonlarda, küresel minimum nokta(ları) içeren alan, tüm arama alanına oranla çok küçüktür. Bazı fonksiyonlarda küresel minimum yerel minimuma çok yakındır. Algoritma, dar kavisli bir vadiye sahip bu fonksiyonlardaki yön değişikliklerini sürdüremezse veya birden çok küresel minimuma sahip bir fonksiyon olması durumunda keşif aşamasını etkili bir şekilde değerlendirmezse bu tür problemler için başarısız olacaktır. Algoritmaların yaşayabileceği bir diğer sorun, fonksiyonun etki alanı ile taranması gereken tüm alan arasındaki büyüklük farklılıklarının ölçeklendirme sorunudur. Bir problemin zorluğu genellikle boyutları arttıkça artar. Parametre veya boyut sayısı arttıkça, arama alanı da katlanarak artar. Doğrusal olmayan problemler için, çok boyutluluk hemen hemen tüm optimizasyon

algoritmaları için önemli bir engel olabilir (Yau ve Liu, 1996; Molga ve Smutnicki, 2005; Alatas ve ark., 2009; Yang, 2010; Jamil ve ark., 2013). Çizelge 3.1’de kriter fonksiyonları özellikleriyle birlikte verilmiştir. Sonraki alt bölümlerde bazı kriter fonksiyonlarına detaylıca yer verilmiştir (Jamil ve ark., 2013; Zhang ve ark., 2016).

Çizelge 3.1. Kriter fonksiyonlarının mod ve ayrılabilirlik baz alınarak sınıflandırılması

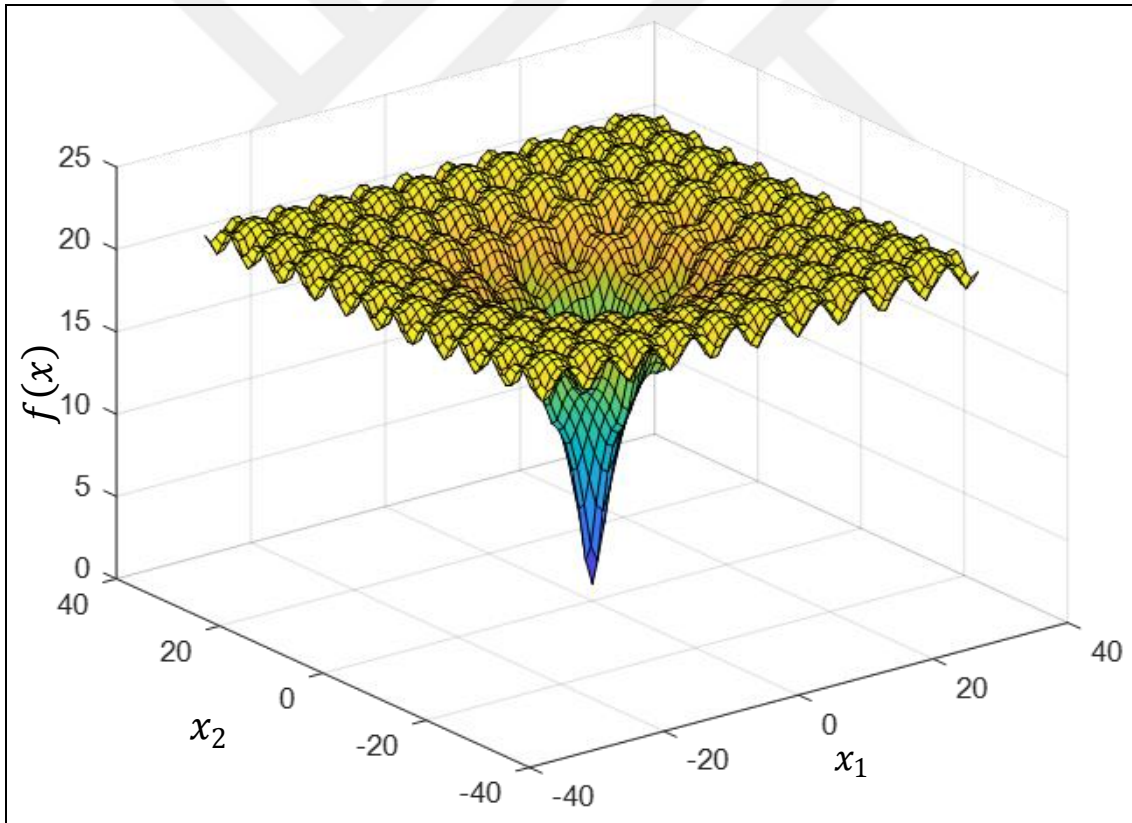
Fonksiyon İsmi	Tanımı	Değişken Değer Aralığı	Özellikler	Küresel minimum f_{min}
Ackley	$f(x) = -a \cdot \exp\left(-b \cdot \sqrt{\frac{1}{n} \sum_{i=1}^n X_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(cx_i)\right) + a + \exp(1)$	$-32.77 \leq x_i \leq 32.77$	Çok modlu, ayrılamaz	0
Bird	$f(x) = \sin(x_1) e^{(1-\cos(x_2))^2} + \cos(x_2) e^{(1-\sin(x_1))^2} + (x_1 - x_2)^2$	$-2\pi \leq x_i \leq 2\pi$	Çok modlu, ayrılamaz	-106.765
Camel-three hump	$f(x) = 2x_1^2 - 1.05x_1^4 + (x_1^6/6) + x_1x_2 + x_2^2$	$-5 \leq x_i \leq 5$	Çok modlu, ayrılamaz	0
Camel-six hump	$f(x) = (4 - 2.1x_1^2 + (x_1^4/3))x_1^2 + x_1x_2 + (4x_2^2 - 4)x_2^2$	$-5 \leq x_i \leq 5$	Çok modlu, ayrılamaz	-1.0316
Exponential	$f(x) = -\exp\left(-0.5 \sum_{i=1}^D x_i^2\right)$	$-1 \leq x \leq 1$	Çok modlu, ayrılamaz	1
Griewank	$f(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$-100 \leq x_i \leq 100$	Çok modlu, ayrılamaz	0
Rastrigin	$f(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i + 10)]$	$-5.12 \leq x_i \leq 5.12$	Çok modlu, ayrılabilir	0
Periyodik	$f(x) = 1 + \sin^2(x_1) + \sin^2(x_2) - 0.1e^{-(x_1^2 - x_2^2)}$	$-10 \leq x_i \leq 10$	Çok modlu, ayrılabilir	0.9
Weierstrass	$f(x) = \sum_{i=1}^n \left[\sum_{k=0}^{K_{max}} \cos(2\pi b^k(x_i + 0.5)) a^k - n \sum_{k=0}^{K_{max}} \cos(\pi b^k) a^k \right]$	$-0.5 \leq x_i \leq 0.5$	Çok modlu, ayrılabilir	0
Easom	$f(x) = -\cos(x_1) \cos(x_2) \exp[-(x_1 - \pi)^2 - (x_2 - \pi)^2]$	$-100 \leq x_i \leq 100$	Tek modlu, ayrılabilir	-1
Egg crate	$f(x) = x_1^2 + x_2^2 + 25(\sin^2(x_1) + \sin^2(x_2))$	$-5 \leq x \leq 5$	Çok modlu, ayrılabilir	0
Sphere	$f(x) = \sum_{i=1}^D x_i^2$	$0 \leq x \leq 10$	Tek modlu, ayrılabilir	0
Beale	$f(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$	$-4.5 \leq x_i \leq 4.5$	Tek modlu, ayrılamaz	0
Rosenbrock	$f(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$-30 \leq x_i \leq 30$	Tek modlu, ayrılamaz	0
Schwefel	$f(x) = 418,9829d - \sum_{i=1}^n [-x_i \sin(\sqrt{ x_i })]$	$-500 \leq x \leq 500$	Tek modlu Kısmen ayrılabilir	0
Sum Squares	$f(x) = \sum_{i=1}^D ix_i^2$	$-10 \leq x \leq 10$	Tek modlu, ayrılabilir	0
Step	$f(x) = \sum_{i=1}^D (x_i + 0.5)^2$	$-100 \leq x \leq 100$	Tek modlu, süreksiz, ayrılabilir	0
Quartic	$f(x) = \sum_{i=1}^n ix_i^4 + rand[0,1]$	$-1.28 \leq x \leq 1.28$	Tek modlu, ayrılabilir	0

3.1.1. Ackley fonksiyonu

Bu çok modlu fonksiyon sayısız yerel (local) minimuma sahipken ortasında bulunan derin dar havzasında bir tane küresel çözüme sahiptir (hussain ve ark., 2017). Eşitlik aşağıda ifade edilmiştir.

$$f(x_i) = -a \cdot \exp\left(-b \cdot \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(cx_i)\right) + a + \exp(1) \quad (3.1)$$

Eşitlikte $a = 20, b = 0.2, c = 2\pi$ olarak ayarlanması önerilir. Test bölgesi genellikle $-32.768 \leq x_i \leq 32.768, (i = 1, 2, \dots, n)$ hiperküp ile sınırlıdır. $x_i = 0$ için elde edilen global minimum, $f(x) = 0$ değeridir (Molga ve Smutnicki, 2005).



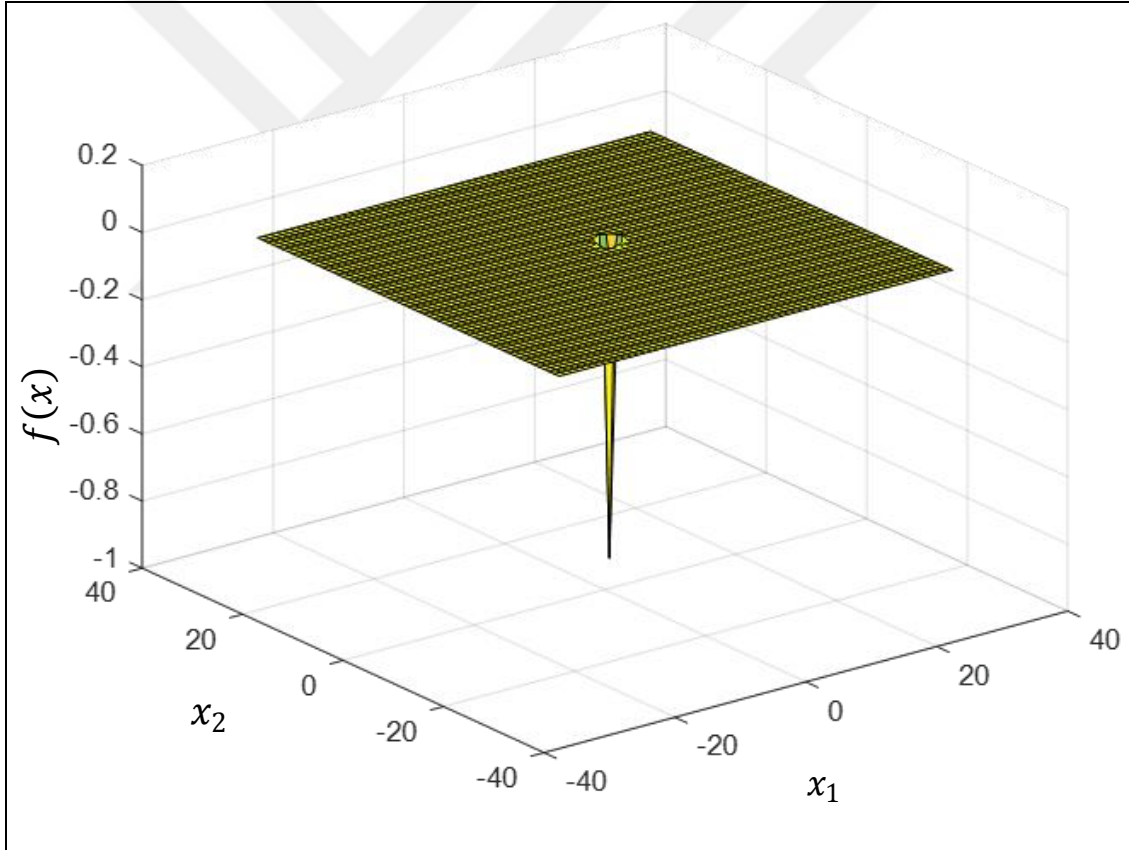
Şekil 3.1. Ackley fonksiyonu.

3.1.2. Easom fonksiyonu

Easom fonksiyonu tek modlu, iki boyutlu, birkaç yerel minimumu olan bir kriter fonksiyonudur ve genel minimum arama alanına göre küçük bir alana sahiptir. Eşitlik aşağıda verilmiştir (Molga, Smutnicki, 2005).

$$f(x_1, x_2) = -\cos(x_1) \cos(x_2) \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2) \quad (3.2)$$

Test bölgesi genellikle $-100 \leq x_1 \leq 100, -100 \leq x_2 \leq 100$ ile boyutlandırılan kare ile sınırlıdır. $(x_1, x_2) = (\pi, \pi)$ için küresel minimum $f(x) = -1$ 'dir.



Şekil 3.2. Easom fonksiyonu.

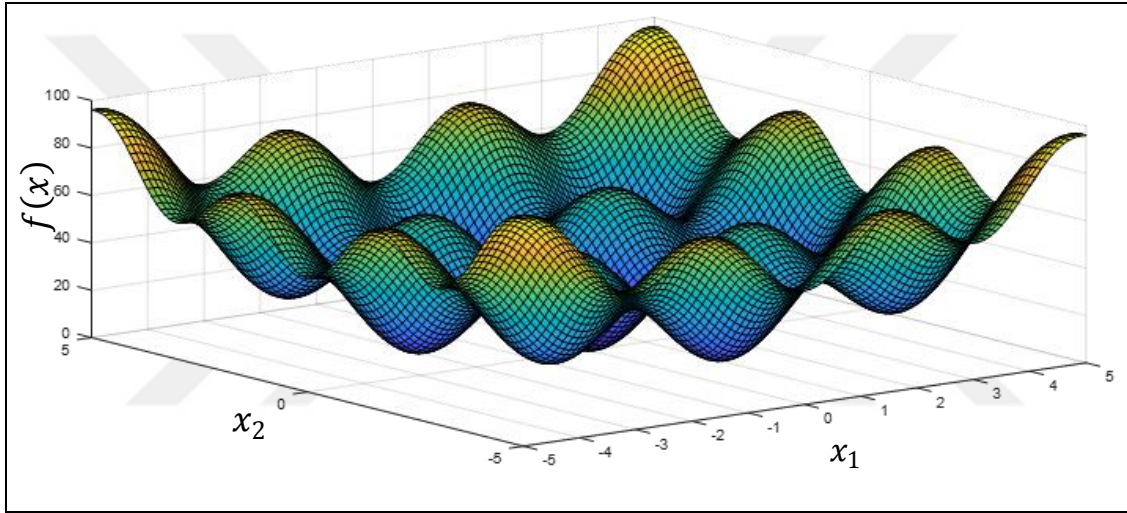
3.1.3. Egg crate fonksiyonu

Egg crate fonksiyonu çok modlu bir kriter fonksiyonudur Eşitlik aşağıdaki gibi ifade edilir (Jamil, yang, 2013).

$$f(x) = x_1^2 + x_2^2 + 25(\sin^2(x_1) + \sin^2(x_2)) \quad (3.3)$$

Test alanı genellikle $-5 \leq x_i \leq 5$ hiperküpü ile sınırlıdır.

Küresel minimum $x = 0$ için $f(x) = 0$ 'dır.



Şekil 3.3. Egg crate fonksiyonu.

3.1.4. Griewank fonksiyonu

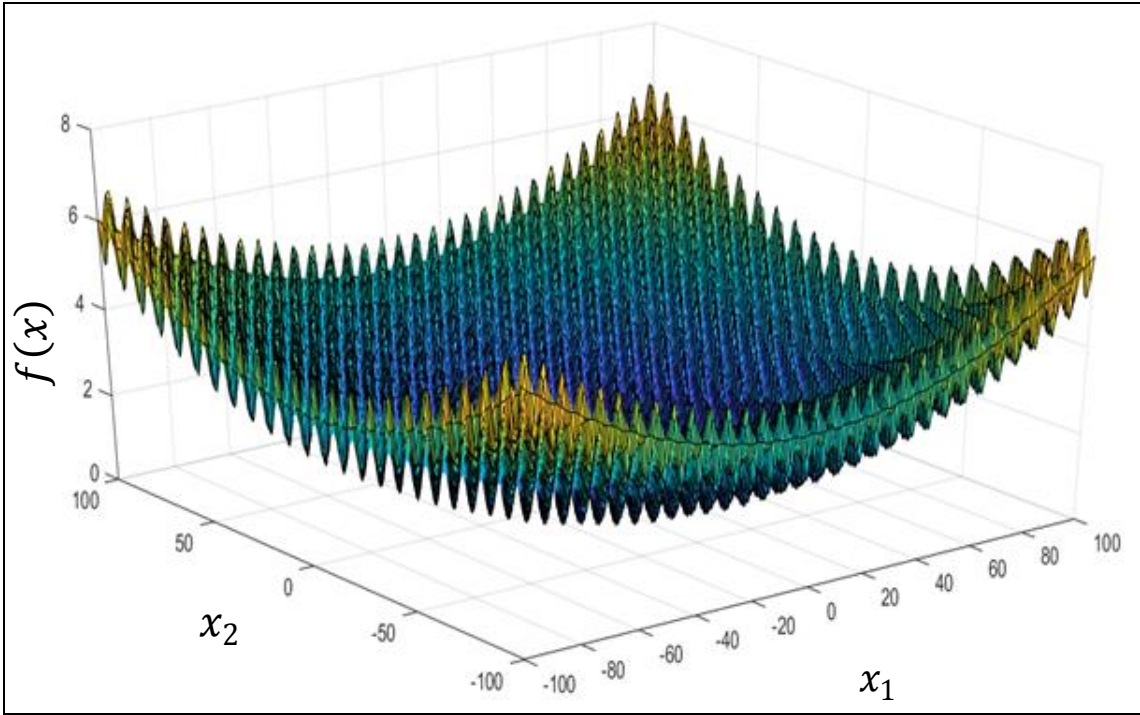
Fonksiyon düzenli olarak dağılan birçok yerel minimaya sahiptir. Eşitlik aşağıda verildiği gibi ifade edilir (Molga ve Smutnicki, 2005).

$$f(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (3.4)$$

Test alanı genellikle $-600 \leq x_i \leq 600$ hiperküpü ile sınırlıdır. Global minimum $x_i = 0, i = 1, 2, \dots, n$ olmak üzere $f(x) = 0$ 'dır.

Aşağıdaki Şekil 3.4 $f(x, y)$ fonksiyonu için çizilmiştir.

$$f(x, y) = \frac{x^2 + y^2}{4000} - \cos(x) \cos\left(\frac{y}{\sqrt{2}}\right) + 1 \quad (3.5)$$



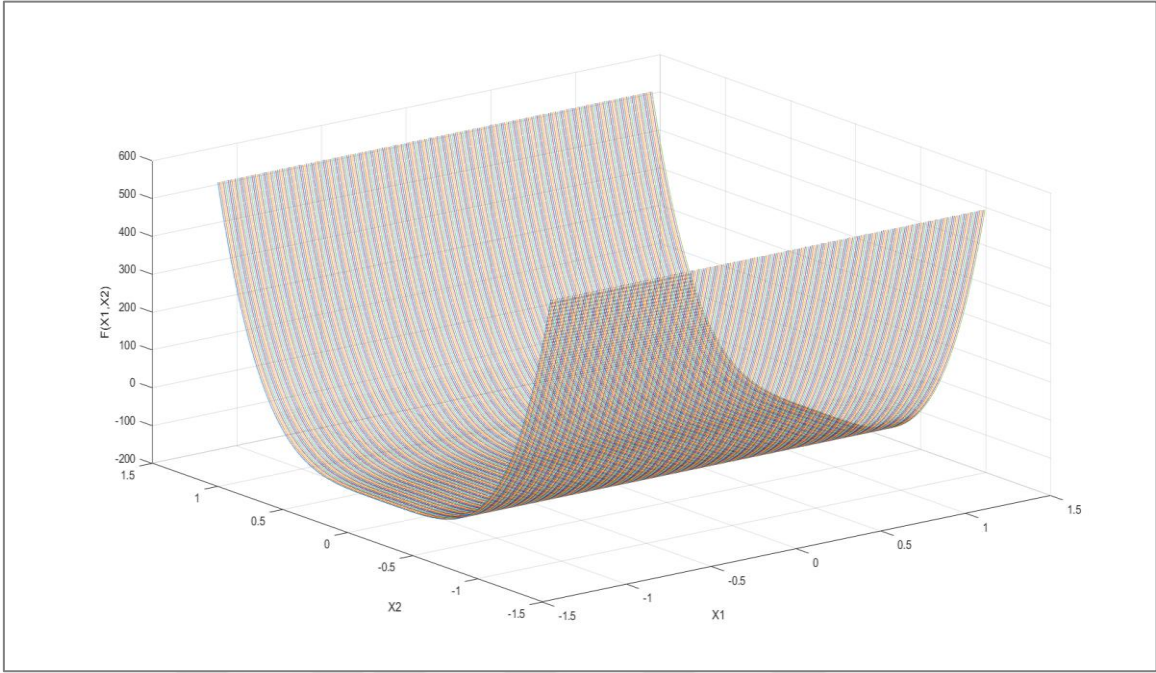
Şekil 3.4. Griewangk fonksiyonu.

3.1.5. Quartic fonksiyonu

Quartic sürekli ve unimodal bir fonksiyon olduğundan kolay bir çözüme sahiptir, sphere fonksiyonlarının özelliklerini taşır (Hussain ve ark., 2017).

Test alanı genellikle $-1.28 \leq x_i \leq 1.28$ hiperküpü ile sınırlıdır. Global minimumu $x_i = 0, i = 1, 2, \dots, n$ olmak üzere $f(x) = 0$ 'dır. Aşağıda “Eş. 3.6” ile ifade edilmiştir.

$$f(x) = \sum_{i=1}^n ix_i^4 + rand[0,1] \quad (3.6)$$



Şekil 3.5. Quartic fonsiyonu.

3.1.6. Rosenbrock fonsiyonu

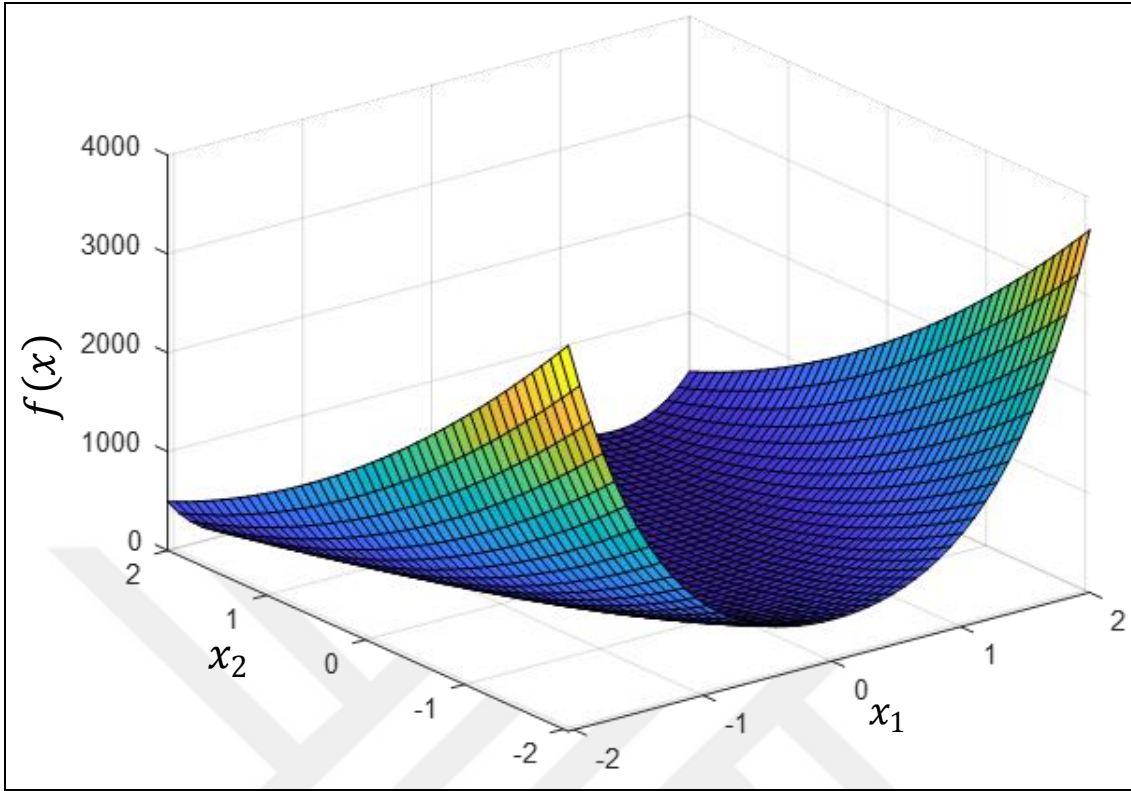
Bu fonksiyon banana ya da DeJong'un ikinci fonksiyonu (the second function of De Jong) olarak da bilinir (Molga ve Smutnicki, 2005). Fonksiyon unimodal ve küresel minimum dar, parabolik bir vadide uzanmakla birlikte minimuma yakınsama güçlükleri vardır (Picheny ve ark., 2012). Eşitlik aşağıda ifade edilmiştir (Molga ve Smutnicki, 2005).

$$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2] \quad (3.7)$$

Test bölgesi $-2.048 \leq x_i \leq 2.048, i = 1, 2, \dots, n$ ve küresel minimum fonksiyonu, $f(x) = 0$ 'dır.

$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2 \quad (3.8)$$

için Şekil 3.6'da çizilmiştir.



Şekil 3.6. Rosenbrock fonksiyonu.

3.1.7. Schwefel fonksiyonu

Fonksiyon aşağıdaki eşitlikte ifade edildiği gibi tanımlanır (Molga ve Smutnicki, 2005).

$$f(x) = 418,9829d - \sum_{i=1}^n [-x_i \sin(\sqrt{|x_i|})] \quad (3.9)$$

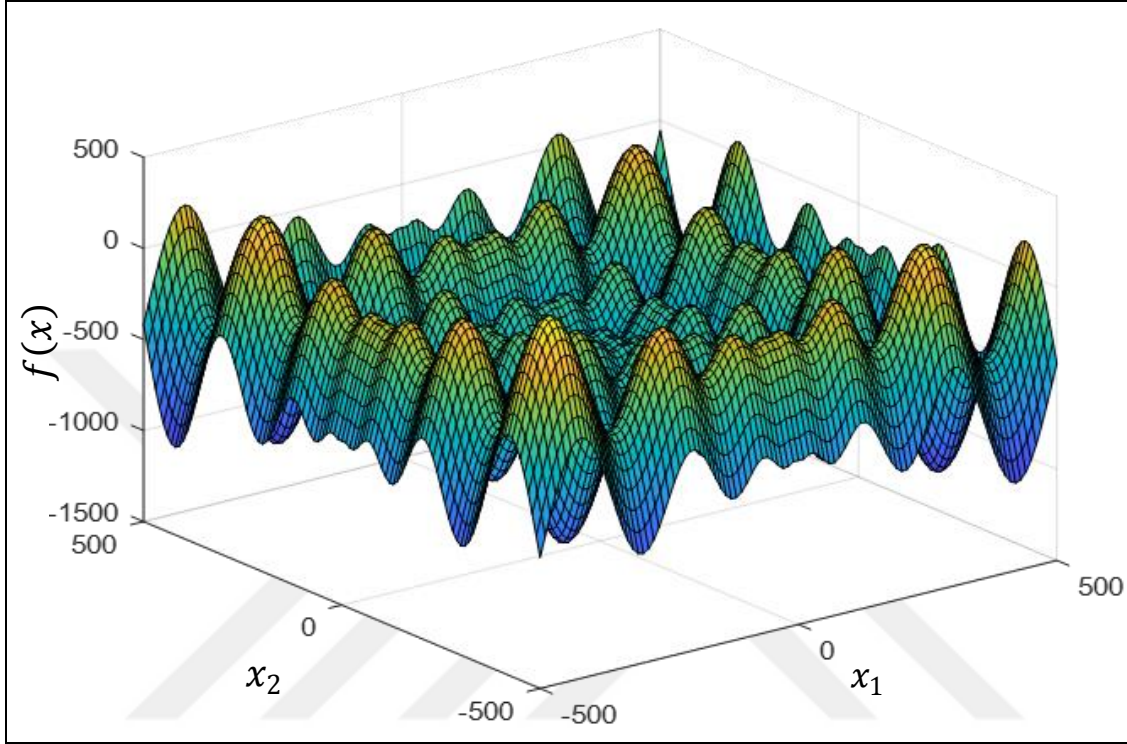
Test alanı genellikle $-500 \leq x_i \leq 500$ hiperküpü ile sınırlıdır. Global minimum

$$x_i = -420.9687, i = 1, 2, \dots, n \text{ olmak üzere } f(x) = 0 \text{ 'dır.}$$

Schwefel fonksiyonu, küresel minimum değer bir sonraki en iyi yerel minimumdan parametre alanı üzerinde geometrik olarak uzak olması nedeniyle yanıltıcıdır. Bu nedenle, arama algoritmaları potansiyel olarak yanlış yönde yakınsama eğilimi gösterir (Molga ve Smutnicki, 2005).

$$f(x, y) = -x \sin(\sqrt{|x|}) - y \sin(\sqrt{|y|}) \quad (3.10)$$

için Şekil 3.7’de fonksiyon ifadesini bulmuştur.i



Şekil 3.7. İki boyutlu Schwefel fonksiyonu.

3.1.8. Sphere (De Jong’s) fonksiyonları

a. Sphere tek modlu ve sürekli bir fonksiyondur. Bu fonksiyon konveks ve tek modludur. Test alanı genellikle hiperküp $-5.12 \leq x_i \leq 5.12, i = 1, 2, \dots, n$ ile sınırlıdır. Aşağıda eşitlik verilmiştir (Molga ve Smutnicki, 2005).

$$f(x) = \sum_{i=1}^n x_i^2 \quad (3.11)$$

Küresel minimum $x_i = 0$ için $f(x) = 0$ elde edilebilir (Molga ve Smutnicki, 2005).

b. Sphere fonksiyonu ile ilişkili bir diğer fonksiyon, konveks ve unimodal olan ve aşağıdaki eşitlikte ifade edilen, ağırlıklı küre fonksiyonu veya hiper-elipsoid fonksiyondur (Yang, 2010; Molga ve Smutnicki, 2005).

$$f(x) = \sum_{i=1}^n i x_i^2 \quad (3.12)$$

$-5.12 \leq x_i \leq 5.12$ ve $x_i = 0$ için $f(x) = 0$ 'dır.

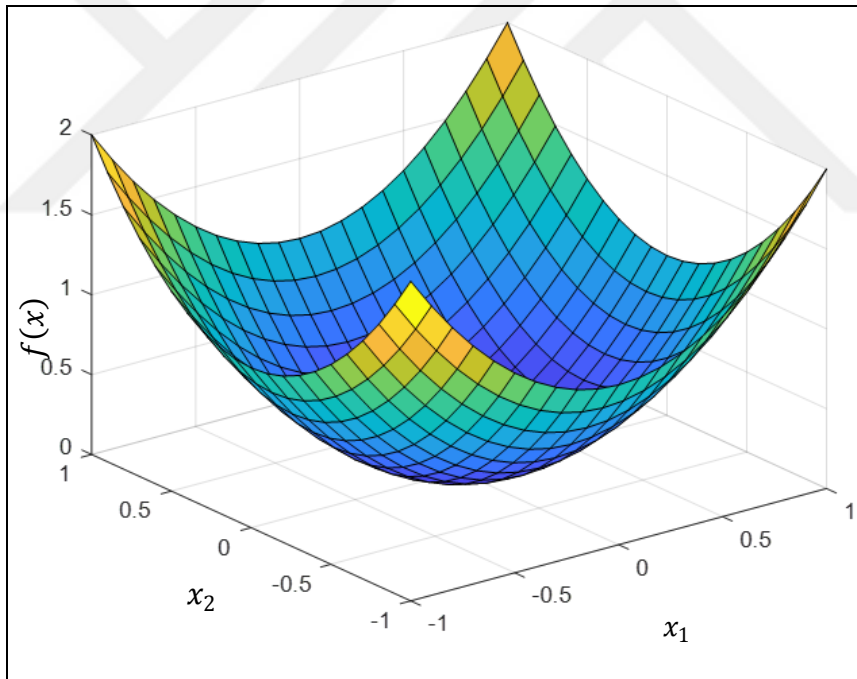
c. Diğer bir ilişkili fonksiyon ise aşağıdaki eşitlikte ifade edilen farklı güç fonksiyonlarının toplamı fonksiyonudur (Yang, 2010).

$$f(x) = \sum_{i=1}^n |x_i|^{i+1} \quad (3.13)$$

$-1 \leq x_i \leq 1$ ve $x_i = 0$ için $f(x) = 0$ 'dır.

$$f(x, y) = x^2 + y^2 \quad (3.14)$$

için aşağıda Şekil 3.8'de çizilmiştir.



Şekil 3.8. İki boyutlu yüzeyde Sphere fonksiyonu.

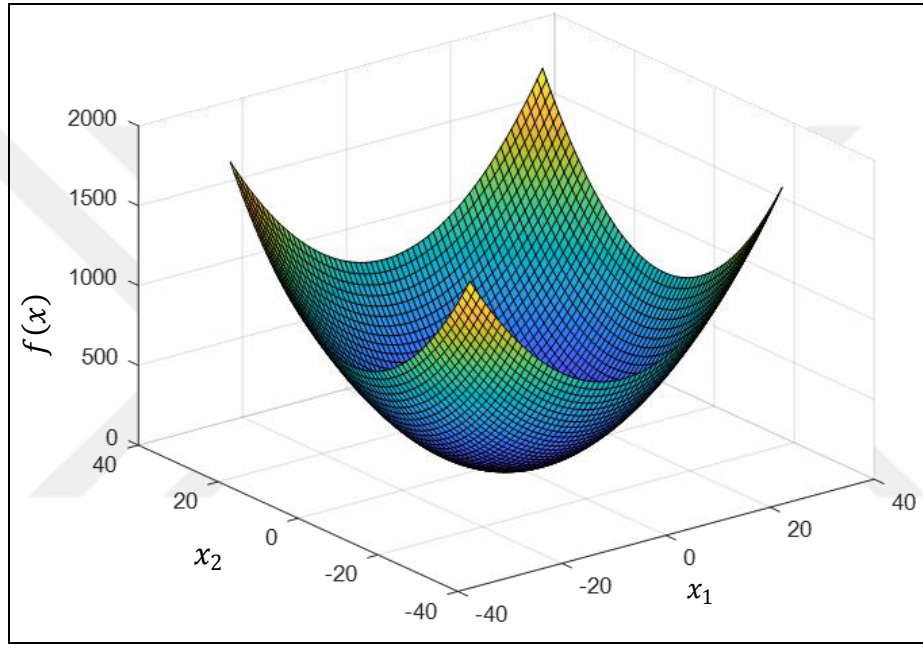
3.1.9. Step fonksiyonu

Bu fonksiyon düz bir yüzeyi temsil eder ki bu yüzeyin özelliği, küresel olarak optimum konuma doğru hiçbir uygun yön bulunamamasıdır. Tek modlu, kesikli ve differansiyellenemez bir fonksiyondur. Matematiksel ifadesi aşağıdaki eşitlikte verilen

bu fonksiyon $-100 \leq x_i \leq 100$ alanıyla sınırlandırılmış ve küresel minimumu $x = 0$ için $f(x) = 0$ 'dır (Jamil ve Yang, 2013; Zhang ve ark., 2016; Hussain ve ark., 2017).

$$f(x) = \sum_{i=1}^n (x_i + 0.5)^2 \quad (3.15)$$

(3.15) eşitliği için aşağıdaki şekil çizilmiştir (Al Roomi, 2019).



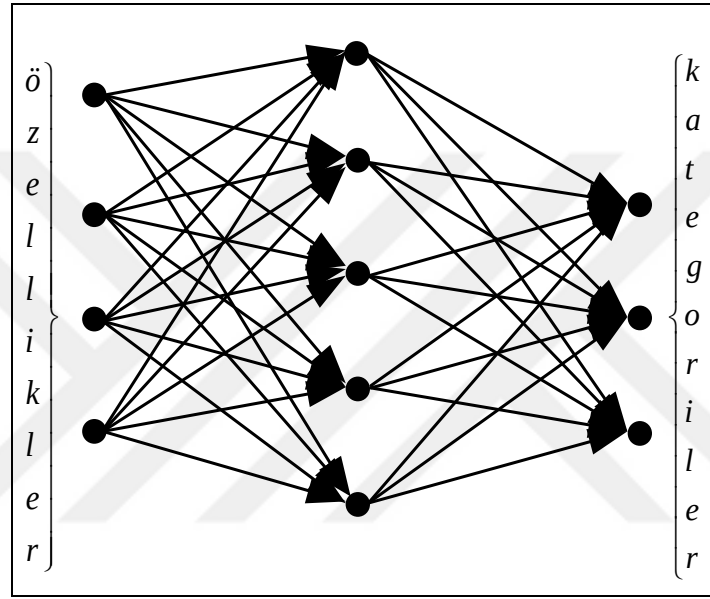
Şekil 3.9. Step fonksiyonu.

3.2. YSA Eğitiminde Veri Setlerinin Kullanımı

YSA biyolojik sinir sistemlerinin gözlenen bazı özelliklerini taklit eden ve uyarlanabilir biyolojik öğrenmenin birbirine benzeşmelerinden yararlanan matematiksel model kümeleridir. YSA modelinin kilit unsuru, bilgi işleme sisteminin yeni yapısıdır. Sinapslara benzer çok sayıda birbirine bağlı işleme elemanlarından oluşur (Haykin, 2005). YSA'da sınıflandırma tahminlerinin yüksekliği problem çözümünün kalitesini de güçlendirir. Ele alınan veri kümeleri problemlerinin çözümü de istenen sınıflandırma oranına yakınlığı ifade eder. Sağlık alanında Breast Cancer (Meme Kanseri) ve Heart Disease (Kalp Hastalıkları) veri setlerikullanılacaktır, kalp hastalıkları veri seti kısaca Heart olarak adlandırılmıştır. Hayatta kalma tahminleri günümüzde klinik

çalışmalarında sağ kalım analizinin istatistiki teknikleri kullanılarak yapılmaktadır. Bu şekilde, yapay sinir ağlarının, girdi verileri ile tahmin edilecek bilgiler arasında karmaşık doğrusal olmayan etkileşimlerin olduğu veri kümelerini analiz etmek için güçlü bir araç olduğu gösterilmiştir (Chunekar ve Ambulgekar, 2009).

Aşağıdaki şekilde ifade edildiği gibi eğitmek üzere yapay sinir ağına girilen veri seti belli kategorilere sınıflandırılarak çıkar.



Şekil 3.10. Sınıflandırma ağı.

3.2.1. XOR veri seti

$(n + 1)$ reel sayısı ve $(w_1, w_2, w_3, \dots, w_n)$ ağırlıklar olmak üzere, $n -$ boyutlu uzayda her bir $(x_1, x_2, x_3, \dots, x_n) \in A$ ve $(x_1, x_2, x_3, \dots, x_n) \in B$ olmak üzere

$$\sum_{i=1}^n w_i x_i \geq w_{n+1} \quad (3.16)$$

$$\sum_{i=1}^n w_i x_i \leq w_{n+1} \quad (3.17)$$

Eşitlikleri şeklinde ise A ve B iki nokta kümesi lineer olarak ayrılabilir. Tek katmanlı algılayıcılar (perceptron) kullanılarak AND, OR, NAND, NOR, NOT mantık kalıplarının inşası, problemlerde doğrusal ayrılabilirliğin varlığı nedeniyle mümkündür. Ancak XOR (exclusive OR) problemi, lineer olarak ayrılamayan bir problem. XOR mantığının sergilendiği problemde doğrusal olmama(non-linearity) durumu nedeniyle

XOR problemini tek katmanlı perspetron kullanarak çözmek mümkün değildir (Singh, 2016). T (doğru)'yi F (yanlış)'den ayıran hiper düzlem XOR durumunda eşitlik

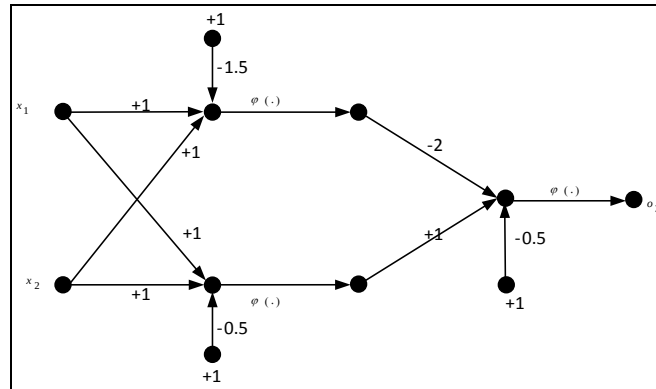
$$\sum_{i=1}^n w_i x_i = \theta \quad (3.18)$$

şeklinde ifade edilir (Murtagh, 1991). XOR Çizelge 3.2'de gösterildiği gibi AND ve OR yapılarından farklı doğrusal olmayan özel bir mantığa sahiptir (Haykin, 2005).

Çizelge 3.2. XOR mantığını gösterir çizelge (F=0, T=1)

Giriş vektörü		İstenilen Çıkış	AND			OR			XOR		
0	0	0	0	0	F	0	0	F	0	0	F
0	1	1	0	1	F	0	1	T	0	1	T
1	0	1	1	0	F	1	0	T	1	0	T
1	1	0	1	1	T	1	1	T	1	1	F

Çizelge 3.2' de istenilen çıkışın sadece XOR yapısıyla sağlandığı görülmektedir. Şekil 3.11'de ise XOR yapısını yansıtan YSA mimarisi çizilmiştir.



Şekil 3.11. XOR problemini çözmek için ağın mimari grafiği.

3.2.2. İris veri seti

Biyolojik bilginin bilgisayar yardımı ile incelenmesi ve işlenmesi 21. yüzyılda gelecek vaat eden yeni bir araştırma alanıdır. Bu alan veriye dayalıdır ve biyolojiden veri alınarak işlenmesini amaçlar. Biyolojik verilerde kodlanan bu bilgiyi elde etmek için ileri hesaplama teknolojileri, algoritmalar ve araçların kullanılması gerekmektedir.

Protein yapı tahmini, dizilerin çoklu hizalanması, çeşitli organizma toplulukları arasındaki evrimsel ilişkiler ile ilgili çıkarımlar vb. gibi bu alandaki temel problemleri çözmek için yapay zekâ yöntemleri güçlü ve verimli bir yaklaşım sunar. Yapay Sinir Ağları, hızlı çözüm ve temsil etme yeteneği nedeniyle yaygın olarak kullanılan tekniklerinden biridir (Singh ve Hachimi, 2018).

İris çiçeği bu alanda en çok kullanılan veri setlerinden biridir (Fisher, 1936). İris çiçeği sınıflandırma probleminin çok katmanlı algılayıcı tarafından hem eğitim hem de test sırasında daha iyi performans göstererek çözümü ispatlanmıştır (Borovinskiy ve Potočník, 2009). İris veri seti <http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data> adresinden elde edilmiştir.

İris veri kümesi desen sınıflandırması yaparak üç farklı iris sınıfını sınıflandırır (Hoey, 2004). Veri kümesi, her biri 50 türden oluşan 3 sınıf içerir, burada her sınıf bir tür iris bitkisine karşılık gelir ve veri kümesinin tüm özelliklerinde değerler eksiksizdir. Üç sınıf arasında eşit olarak ayrılmış 150 örnek, dört sayısal özelliği içerir. Bunlar şu şekildedir:

1. Çanak yaprağı (sepal) uzunluk
2. Sepal genişlik
3. Taç yaprağı (petal) uzunluğu
4. Petal genişliği

Yukarıdaki dört özellik incelenerek aşağıdaki üç sınıf oluşur:

- İris Setosa
- İris Versicolour
- İris Virginica

İris veri setinin uygulanmasındaki amaç iris bitkisinin petal ve sepal büyüklüğünü inceleyerek iris çiçeğinin sınıflandırma tahmininin boyutunu görmektir. İris veri kümesi kullanılarak çıkarılan ilişkinin türü bir sınıflandırma modelidir. Bu, petal ve sepal boyutlarını inceleyerek iris bitkisinin tipini sınıflandırabilir. Sepal genişliğin Sepal uzunluk ile pozitif ilişkisi ve petal genişliğinin petal uzunluğu ile pozitif ilişkisi vardır. (Swain ve ark., 2012; A. Kadir ve ark., 2017).

3.2.3. Meme kanseri (Breast Cancer) veri seti

Meme Kanseri Wisconsin Tanı (WDBC) veri tabanından edinilen ilk veri seti, Wisconsin Üniversitesi Madison Hastanelerinde Dr. William H. Wolberg tarafından toplanmış ve veri seti UCI veri deposundan elde edilmiştir (<http://archive.ics.uci.edu/ml/machine-learning-databases/breastcancer/breastcancer.data>). Meme kanseri teşhisi hakkındaki birçok çalışmada Wisconsin'e ait meme kanseri veritabanında bulunan veriler kullanılmaktadır (El-Rashidy ve ark., 2011). Meme kanseri veri kümesinde toplam 11 özellik vardır. 9 adet giriş, 2 adet çıkış (sınıf) bilgisi içeren 699 veriden oluşan bir veri kümesidir. Veriler üzerinde özellikle sınıflandırma ve özellik seçimi uygulamaları yapılmaktadır (Elmore ve ark., 1994; İlkuçar ve ark., 2014). Veri tabanındaki her kayıt, bir meme kütlesinin ince iğne ile alınan doku materyalinin sayısallaştırılmış görüntüsünden hesaplanan dokuz özelliğe sahiptir. Her hücre çekirdeği için hesaplanan bu özelliklerin meme kanseri teşhisi için önemli özellikler olduğu düşünülmektedir; bu özellikler arasında Yığın kalınlığı (Clump), hücre boyutunun homojenliği (Ucellsize), hücre şeklinin homojenliği (Ucellshap), marjinal yapışma (Mgadhesis), tekil epitel hücre boyutu (Sepics), zarla çevrili çekirdeği olmayan (Bnuclei), yumuşak kromatin (Bchromatin), normal nükleoller (Normnuct) ve Mitozlar vardır. Bu veri tabanında 241 (% 34.5) kayıt kötü huylu, 458 (% 65.5) kayıt ise iyi huyludur. Veri tabanında bulunan 699 örneğe ait veri ve bunların ait oldukları (çıkış) 2 sınıf mevcuttur. Sınıflar benign (iyi huylu) ve malignant (kötü huylu) şeklindedir. Örneklerin 458'i benign sınıfına aitken, 241'i malignant sınıfına aittir.

Çizelge 3.3'te görüldüğü gibi tahminde kullanılacak nitelikler 2-10 arasındaki özelliklerdir. 1. Özellik şahsi kodu, 11. Özellik ise sınıf bilgisini içermektedir (Şentürk ve Şentürk, 2016).

Çizelge 3.3. Meme kanseri veri setindeki özellikler ve değer aralıkları

	Veri Seti	Değer Aralığı
1	Veri seti kod numarası	ID Numarası
2	Clump Thickness (Clump)	1 - 10
3	Uniformity of Cell Size (Ucellsize)	1 - 10
4	Uniformity of Cell Shape (Ucellshap)	1 - 10
5	Marginal Adhesion (Mgadheseion)	1 - 10
6	Single Epithelial Cell Size (Sepics)	1 - 10
7	Bare Nuclei (Bnuclei)	1 - 10
8	Bland Chromatin (Bchromatin)	1 - 10
9	Normal Nucleoli (Normnuct)	1 - 10
10	Mitoses	1 - 10
11	Class	2 - 4

3.2.4. Kalp hastalıkları (Heart Disease) veri seti

Kalp Hastalığı veri setinin amacı, bir hasta üzerinde yapılan çeşitli tıbbi testlerin sonuçları göz önüne alındığında, kalp hastalığının varlığını veya yokluğunu tahmin etmektir. Bu tahmin seti hastalar üzerinde yapılan çeşitli tıbbi testlerin sonuçlarından oluşmuştur. Kalp hastalığı veritabanı UCI makine öğrenimi havuzundan alınmıştır (<ftp://ftp.ics.uci.edu/pub/ma>, 2008). Cleveland kalp hastalığı verileri V.A. Robert Detrano Tıp Merkezi, Long Beach ve Cleveland Klinik Vakfı'na aittir. Veri tabanı, 297'si tam örnek, altısı eksik sınıf değerlerine sahip 303 örnek içermektedir. Örneklerden 27'si anlaşmazlık halinde kullanılmak için saklandı ve toplam 270 örnek üzerinden işlem yapıldı. Bu veritabanı, daha geniş bir 75 setten çıkarılmış 13 özellik içermektedir, çıktı kalp hastalığı vardır ya da yoktur şeklinde iki sınıftan oluşur (Polat ve ark., 2005; Das, 2009).

Çizelge 3.4. Kalp Hastalıkları veri tabanı özellikleri

(1) yaş,
(2) cinsiyet,
(3) meme ağrısı tipi (dört değer),
(4) istirahat kan basıncı,
(5) mg / dl cinsinden serum kolesterol,
(6) açlık kan şekeri > 120 mg / dl,
(7) istirahat elektrokardiyografik sonuçları (0, 1 ve 2 değerleri),
(8) elde edilen maksimum kalp atış hızı,
(9) egzersize bağlı meme ağrısı,
(10) eski tepe = istirahata göre egzersize bağlı ST depresyonu,
(11) egzersize anında ST segmentinin pik eğimi,
(12) floroskopi ile renklendirilmiş büyük damarların (0-3) sayısı,
(13) talyum testi: normal =3; sabit kusur= 6; tersinir kusur= 7.

3.2.5. Balloon veri seti

Daha önce bilişsel psikoloji deneyinde kullanılan verilerden oluşmuştur. Bir balon deneyini baz alarak hazırlanan bu veri setinde (Pazzani, 1991), ikinci denek grubunun birinci grubun yaptığı şişirme eyleminden tahminler çıkararak gerçek çıktıyla yaklaşık sonucu bulmayı hedefler. Veri seti <http://www.ics.uci.edu/~mlearn/databases> adresinden elde edilmiştir. Balon veri kümesi dört özellikten oluşmuştur. Bunlar renk, boyut, hareket ve yaş özellikleridir. Renkler sarı ve mor, boyutlar büyük ve küçük, hareketler gergin ve dalma, yaş ise çocuk ve olgun olmak üzere her özellik iki ayrı sınıfa ayrılmıştır. Balonun pozisyonu ise eğer şişmiş ise E, şişmemiş ise H sembolüyle gösterilir. E ya da H durumunda her biri bir sınıfa yerleştirilir.

Balon veri kümesinde, aynı özelliklere sahip bir deneyin farklı koşullarını temsil eden dört veri kümesi vardır. Yani ağ her özellik için bilgileri temsil etmek için 4 giriş ve şişirilmiş ya da değil göstermek için 1 çıkışa sahip olacaktır. Balonlar veri kümesi toplam 16 örnek olmak üzere 12 eğitim ve 4 test için ayrılmıştır (Shafie ve ark., 2012). Çizelge 3.5'te balon veri setinin özellikleri verilmiştir (Özdemir, 2007).

Çizelge 3.5. Balon veri tabanının anlamsal özellikleri

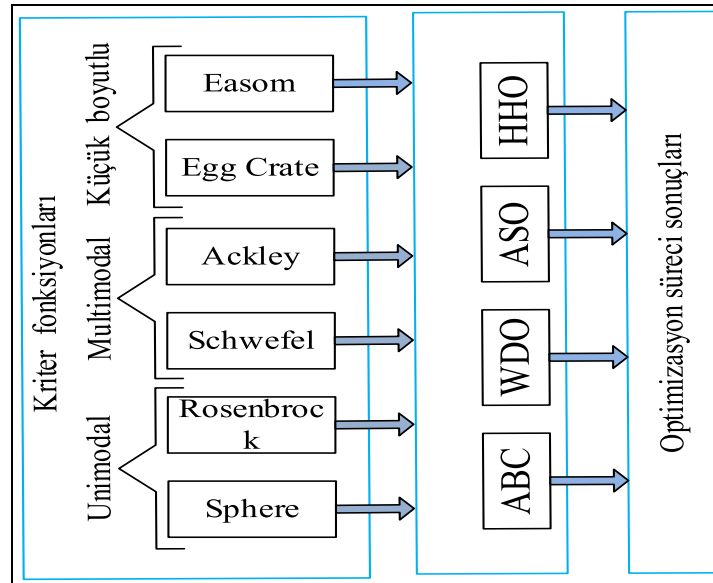
Sıra No.	Renk	Boyut	Hareket	Yaş	Şişmiş
1	Sarı	Küçük	Gergin	Yetişkin	E
2	Sarı	Küçük	Gergin	Çocuk	E
3	Sarı	Küçük	Dalma	Yetişkin	E
4	Sarı	Küçük	Dalma	Çocuk	H
5	Sarı	Geniş	Gergin	Yetişkin	E
6	Sarı	Geniş	Gergin	Çocuk	E
7	Sarı	Geniş	Dalma	Yetişkin	E
8	Sarı	Geniş	Dalma	Çocuk	H
9	Mor	Küçük	Gergin	Yetişkin	E
10	Mor	Küçük	Gergin	Çocuk	E
11	Mor	Küçük	Dalma	Yetişkin	E
12	Mor	Küçük	Dalma	Çocuk	H
13	Mor	Geniş	Gergin	Yetişkin	E
14	Mor	Geniş	Gergin	Çocuk	E
15	Mor	Geniş	Dalma	Yetişkin	E
16	Mor	Geniş	Dalma	Çocuk	H



4. BULGULAR VE TARTIŞMA

4.1. Modern Sezgisel-üstü Algoritmaların Performanslarının İstatistiki Olarak Karşılaştırılması ve Kutu Grafiği ile Performans Ölçüm Değerlendirmeleri

Bu uygulamada HHO algoritmasının global optimizasyon problemlerini çözmedeki etkinliğini ve avantajlarını değerlendirmek amacıyla sürü zekasına dayalı modern sezgisel-üstü ABC, WDO ve ASO algoritmalarıyla karşılaştırmalar yapılmıştır. Bu karşılaştırmalarda altı adet kriter fonksiyonu kullanılmıştır. Bunlar Sphere, Rosenbrock, Easom, Schwefel, Ackley ve Egg Crate fonksiyonlarıdır. İyi bir performans ölçümü için farklı özelliklere sahip fonksiyonlar seçilmiştir. Karşılaştırmada istatistiki ölçüt olarak ortalama, standart sapma, en iyi ve en kötü değer parametreleri değerlendirilmiştir. Ayrıca açıklayıcı veri analizi yöntemiyle kutu grafiği çizdirilerek 5 verinin özeti (1. ve 3. kartil, medyan, en alt ve en üst değer) grafiksel olarak sağlanarak performans ölçüm değerlendirmeleri yapılmıştır. Algoritmaların optimize edilme aşamasında 30 farklı sürüş yapılarak her çalıştırmada sürü boyutu 50 ve 500 iterasyon olarak belirlenmiştir. Şekil 4.1, optimizasyon sürecini göstermektedir. Analizler MATLAB R2014a programı kullanılarak yürütülmüştür.

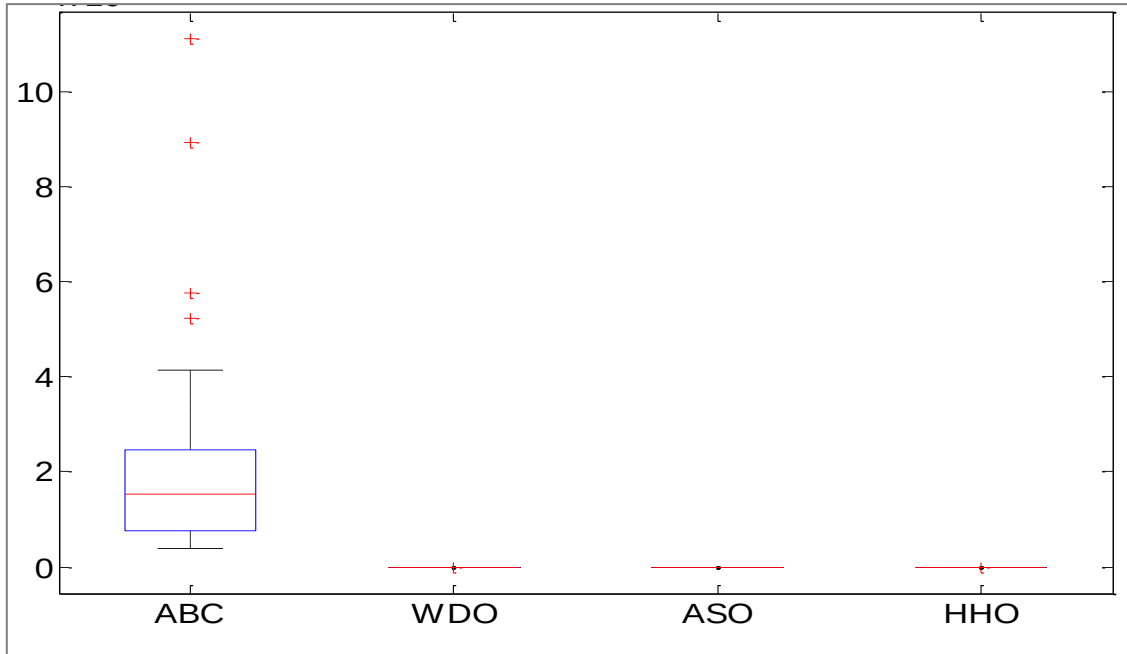


Şekil 4.1. ABC, WDO, ASO ve HHO için optimizasyon süreci.

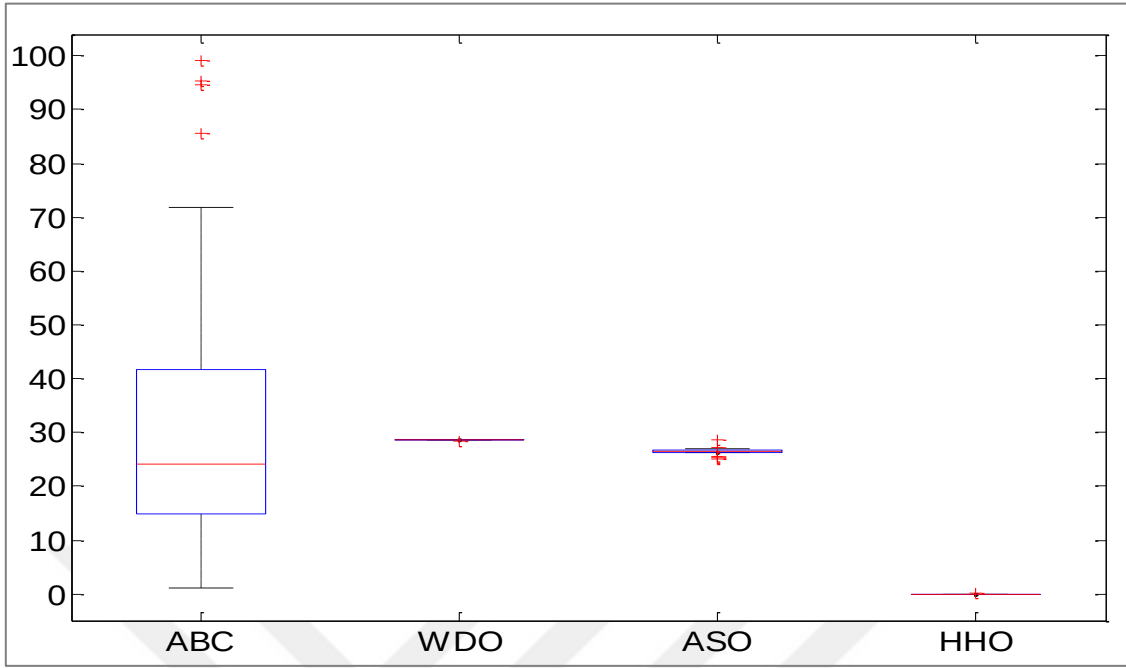
4.1.1. Algoritmaların sonuç analizleri

Çizelge 4.1. HHO algoritmasının diğer algoritmalarla performans karşılaştırması

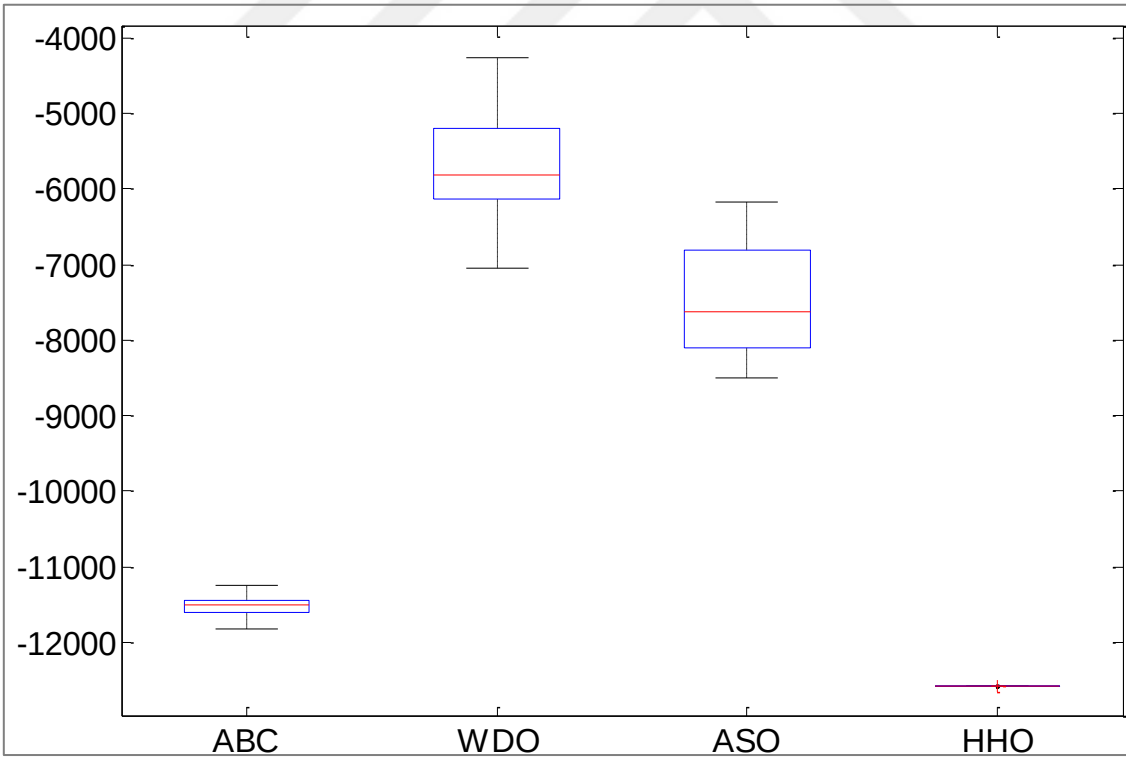
Algoritma	İstatistiki ölçütler	Kriter fonksiyonları					
		Sphere	Rosenbrock	Schwefel	Ackley	Egg Crate	Easom
ABC	En iyi	3.99E-06	1.1928	-11825.43	0.0151	3.19E-21	-1
	En kötü	1.11E-04	99.0436	-11248.52	0.3610	5.05E-18	-1
	Ortalama	2.29E-05	33.3025	-11513.48	0.0868	1.13E-18	-1
	Standart sapma	2.49E-05	28.0139	139.09	0.0818	1.21E-18	0
WDO	En iyi	0	28.3497	-7055.08	8.88E-16	0	-0.9999
	En kötü	2.93E-24	28.6934	-4270.52	1.82E-13	0	-0.9999
	Ortalama	2.30E-25	28.6166	-5674.58	2.75E-14	0	-0.9999
	Standart sapma	6.55E-25	0.0881	747.29	4.27E-14	0	7.57E-06
ASO	En iyi	1.20E-20	25.0683	-8496.42	3.45E-11	0	-3.88E-05
	En kötü	8.45E-19	28.5206	-6173.09	1.59E-09	1.31E-23	-9.87E-142
	Ortalama	3.49E-19	26.4486	-7542.25	5.32E-10	3.65E-24	2.08E-06
	Standart sapma	2.60E-19	0.6349	715.54	3.80E-10	4.08E-24	7.57E-06
HHO	En iyi	3.39E-114	1.39E-05	-12569.49	8.88E-16	6.49E-137	-1.0000
	En kötü	5.91E-100	0.0367	-12567.83	8.88E-16	1.96E-112	-0.9999
	Ortalama	2.10E-101	0.0058	-12569.28	8.88E-16	6.54E-114	-1.0000
	Standart sapma	1.08E-100	0.0081	0.3412	4.01E-31	3.58E-113	4.34E-06



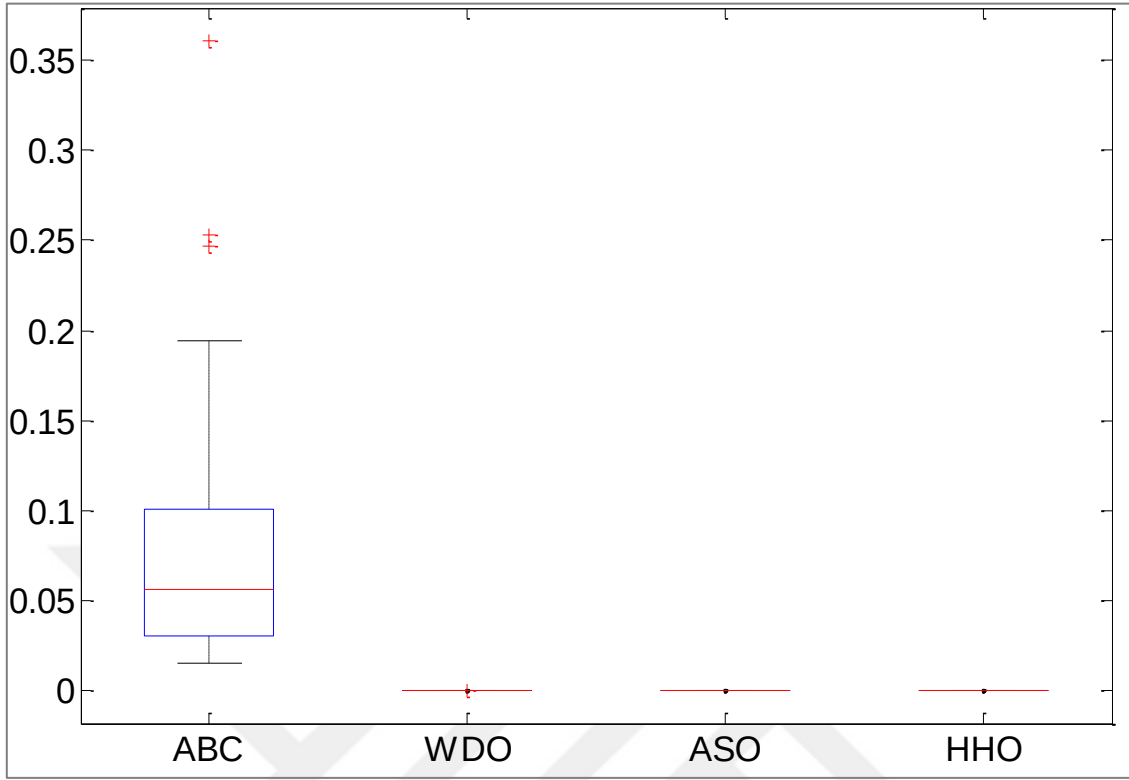
Şekil 4.2. Sphere fonksiyonu için kutu grafiği.



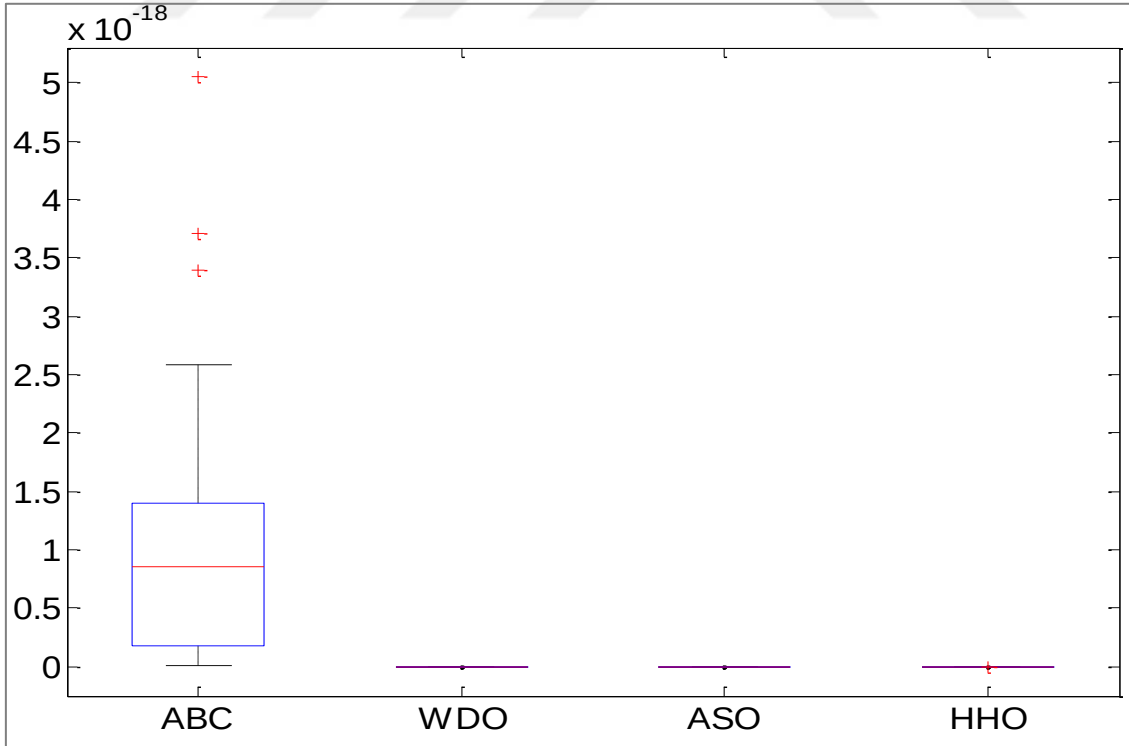
Şekil 4.3. Rosenbrock fonksiyonu için kutu grafiği.



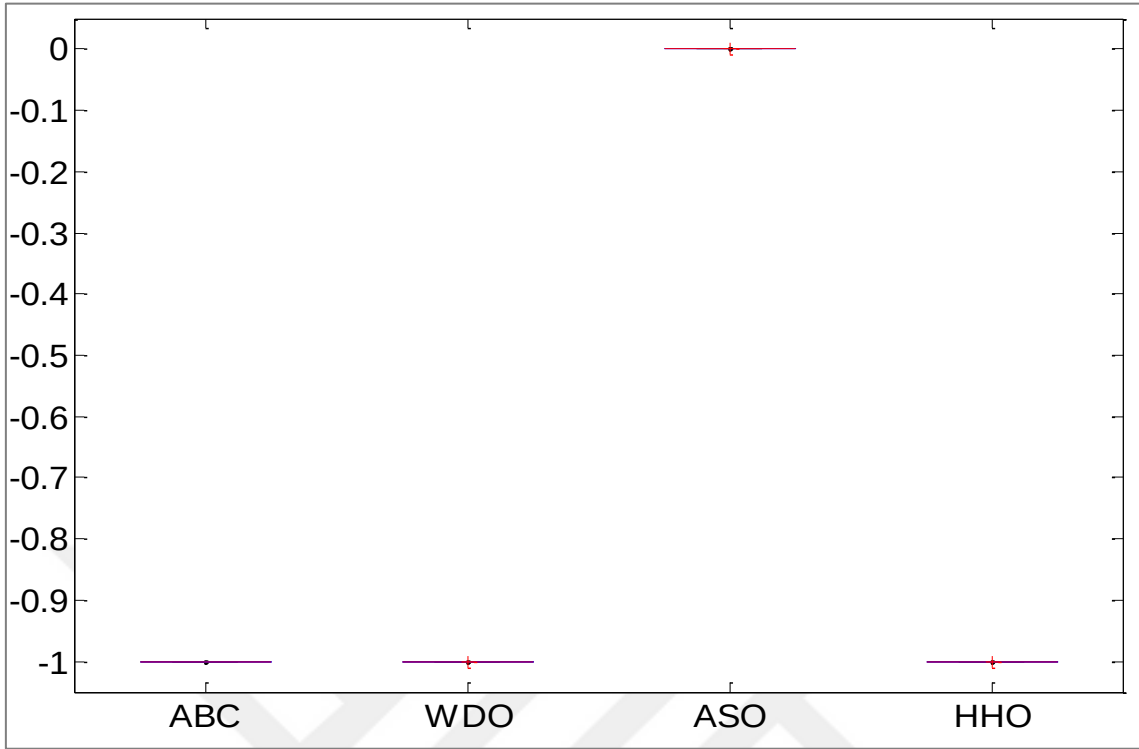
Şekil 4.4. Schwefel fonksiyonu için kutu grafiği.



Şekil 4.5. Ackley fonksiyonu için kutu grafiği.



Şekil 4.6. Egg Crate fonksiyonu için kutu grafiği.



Şekil 4.7. Easom fonksiyonu için kutu grafiği.

Çizelge 4.1’de tüm fonksiyonlar için sonuçlar gösterilmiştir. İstatistiki ölçütler olarak en iyi, en kötü, ortalama ve standart sapma değerleri belirlenmiştir. HHO algoritmasının üstünlüğü tabloda net olarak anlaşılabilmektedir.

Kutu grafikleri bir seri kümesinde medyanı, birinci ve üçüncü kartilleri, en küçük veri ile en büyük veriyi ve varsa uç değerleri gösteren açıklayıcı istatiki bir yöntemdir. Tüm algoritmaların sergiledikleri başarı kutu grafikleri aracılığıyla anlaşılabilir. ABC, WDO, ASO ve HHO algoritmalarının 30 kez çalıştırılması ile optimize edilen altı fonksiyona ilişkin kutu grafikleri Şekil 4.2-4.7’de gösterilmiştir.

Şekil 4.2-4.7’deki kutu grafiklerinden görüleceği gibi HHO algoritması ABC, WDO ve ASO modern sezgisel-üstü algoritmalara oranla çok etkin ve daha kararlı bir yapıya sahiptir. Çünkü HHO algoritması diğer algoritmalar gibi yerel optimum noktasına takılıp kalmamakta ve keşif ile sömürü arasında çok iyi bir denge sağlamaktadır.

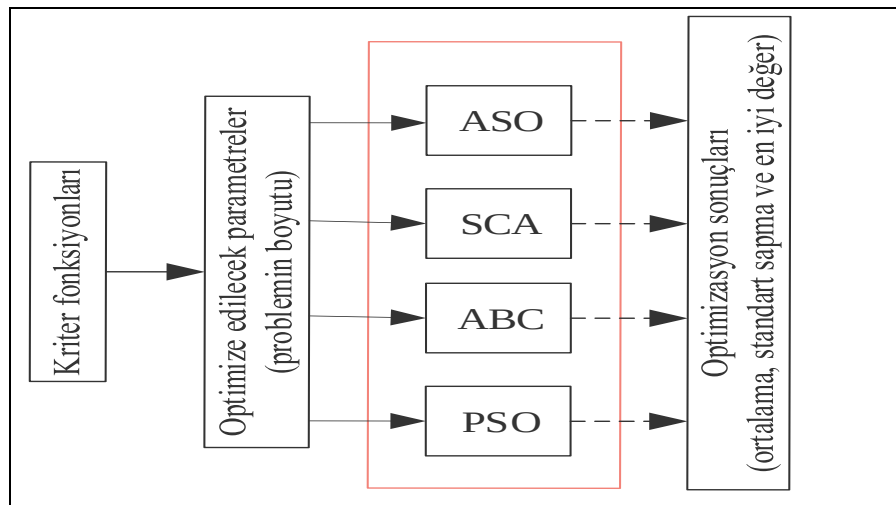
Çalışmanın sonuçların irdelendiğinde, altı kriter fonksiyonu aracılığıyla oldukça güncel ve efektif olan HHO sezgisel-üstü algoritmasının performansı; ABC, WDO ve ASO algoritmaları ile karşılaştırılmıştır. Kriter fonksiyonu olarak Sphere, Rosenbrock, Schwefel, Ackley, Egg Crate ve Easom değerlendirilmiştir. İlgili fonksiyonların

optimizasyonunda yürütülen istatistiki analizlerden, HHO algoritmasının daha üstün performansa sahip olduğu anlaşılmıştır. Seçilen kriter fonksiyonları aracılığıyla yapılan analizler, HHO sezgisel-üstü algoritmasının istatistiki olarak daha anlamlı ve performans üstünlüğüne sahip olduğunu göstermiştir. Bununla beraber kutu grafiğindeki medyan ve kartil değerleri arasındaki aralığın birbirine çok yakın olması ve uç değerlerin olmaması HHO algoritmasının etkinliğini doğrulamakta ve avantajlı olduğunu göstermektedir.

4.2. ASO, PSO, SCA, ABC Sezgisel-üstü Algoritmalarının Problem Çözme Performanslarının Karşılaştırılması ve Sonuçların Yakınsama Eğrileri Aracılığıyla Karşılaştırılması

Bu uygulamada amaç belli kısıtlara sahip optimizasyon problemlerinin çözümünde modern sezgisel-üstü algoritmaların üstün başarılarını göstermektir. Uygulamada, ASO, ABC, PSO ve SCA ile farklı özelliklere sahip kriter fonksiyonlarından Ackley, Egg Crate, Griewank, Rastrigin, Rosenbrock, Schwefel, Sphere, Step fonksiyonları uygulanarak en iyi değere en yakın değer, ortalama ve standart sapma gibi istatistiki sonuçlar bulunmuştur.

ASO, ABC, PSO ve SCA sezgisel-üstü algoritmaları kriter fonksiyonlarına uygulandı. Uygulamada 500 iterasyon, 30 sürüş ve sürü büyüklüğü 50 olarak belirlenmiştir. Şekil 4.8, optimizasyon algoritmalarının kriter fonksiyonlarına uygulanmasını göstermektedir.



Şekil 4.8. Optimizasyon aşaması.

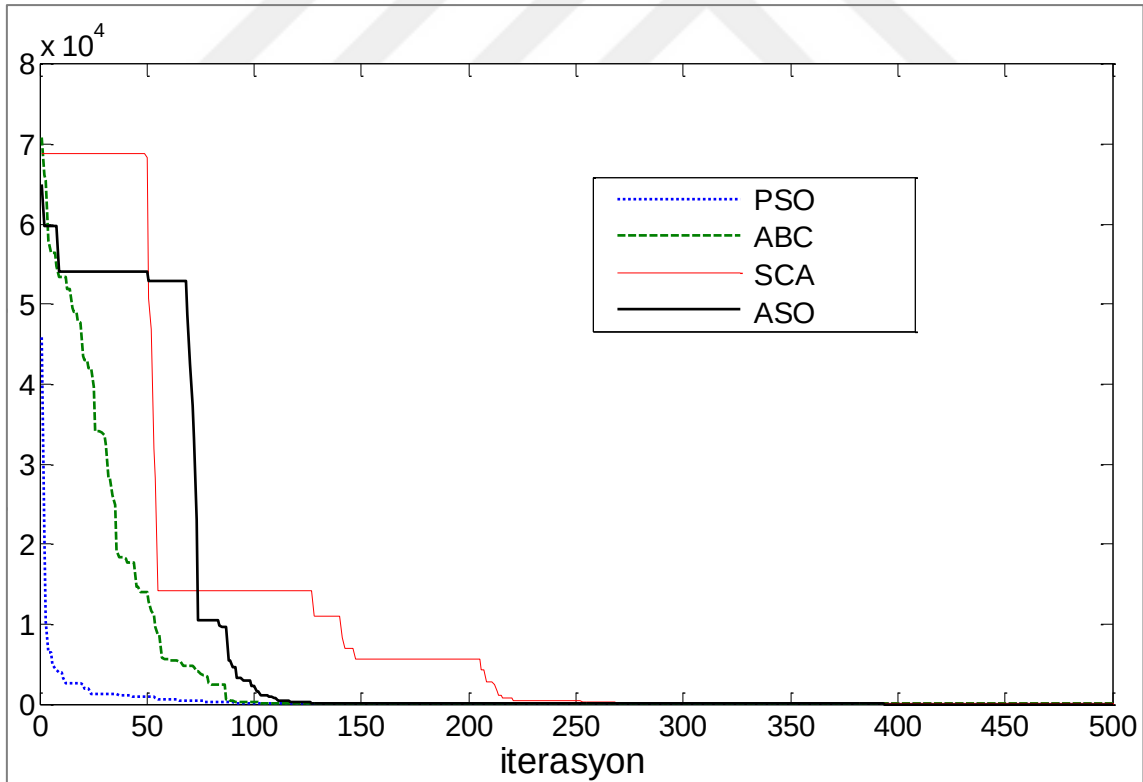
Kriter fonksiyonları için bulunan istatistiki sonuçlar Çizelge 4.2’de sunulmuştur. En iyi sonuçlar bold olarak vurgulanmıştır. Her bir tkriter fonksiyonu optimize edilirken, PSO, ABC, SCA ve ASO algoritmalarının çalıştırılmasıyla geçen süreler yaklaşık olarak sırasıyla 2.42 s, 1.13 s, 0.24 s ve 2.890 s’dir.

Çizelge 4.2. Çeşitli kriter fonksiyonları için sonuçların karşılaştırılması

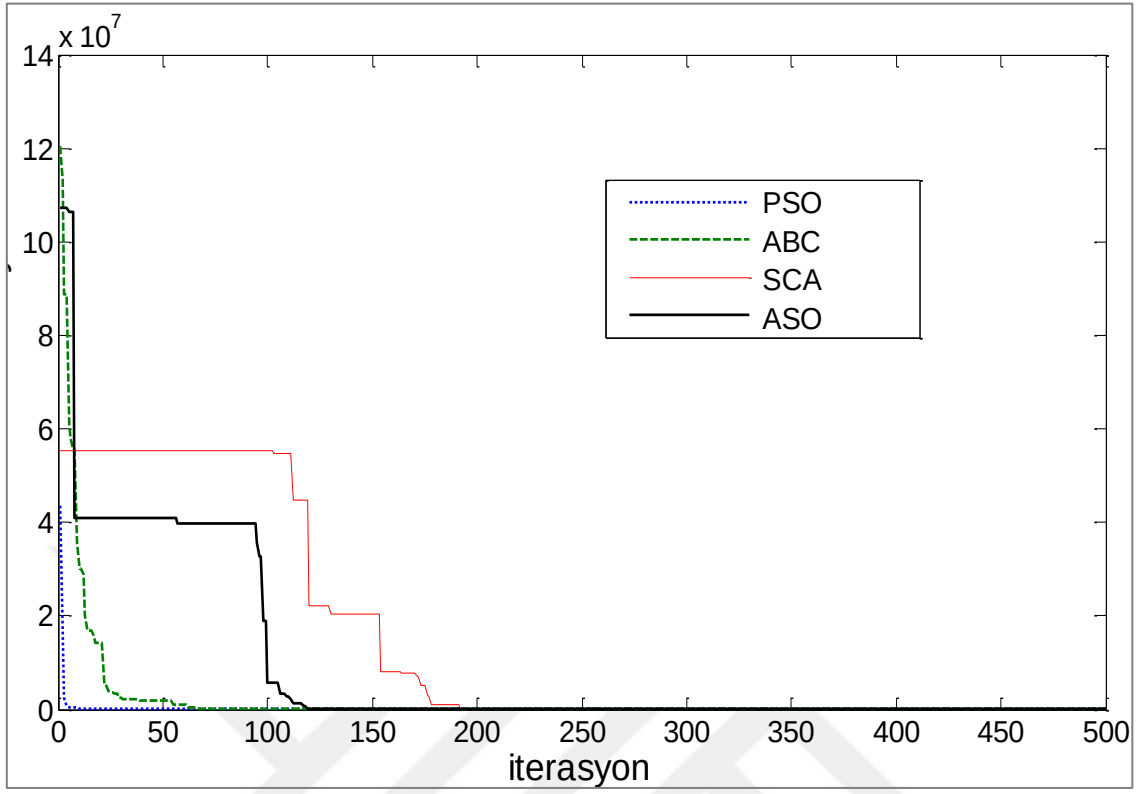
Fonksiyon Adı	İndeks	Sezgisel-üstü optimizasyon teknikleri			
		PSO	ABC	SCA	ASO
Sphere	Ortalama	4.16E-09	2.88E-05	0.259636	4.55E-16
	Standart sapma	8.68E-09	1.74E-05	0.283591	1.75E-15
	En iyi değer	8.30E-12	1.08E-05	0.004759	8.00E-20
Rosenbrock	Ortalama	25.65893	7.458535	17.49040	15.54956
	Standart sapma	21.80477	4.648221	0.283592	7.548646
	En iyi değer	6.323248	0.578422	17.60098	10.20560
Step	Ortalama	4.83E-08	0.003640	5.428303	1.12E-18
	Standart sapma	2.71E-07	0.016399	1.288786	2.20E-18
	En iyi değer	9.50E-13	6.69E-06	3.666500	2.87E-20
Schwefel	Ortalama	-1484.964	-1675.9315	-1440.078	-1615.626
	Standart sapma	88.42275	0	97.58904	68.76877
	En iyi değer	-1675.9315	-1675.9315	-1644.343	-1675.9315
Rastrigin	Ortalama	6.400986	6.46E-15	0.000724	5.107576
	Standart sapma	2.702742	7.42E-15	0.003799	2.857608
	En iyi değer	1.989976	0	0	0.995053
Ackley	Ortalama	0.000274	0.101194	7.171143	5.58E-10
	Standart sapma	0.001066	0.094629	9.253197	3.25E-10
	En iyi değer	2.88E-07	0.009319	0.018818	5.34E-11
Griewank	Ortalama	0.091754	0.008473	0.061780	0
	Standart sapma	0.042514	0.007385	0.097008	0
	En iyi değer	0.009901	1.86E-10	6.27E-15	0
Egg Crate	Ortalama	2.12E-212	3.08E-19	3.53E-77	0
	Standart sapma	6.38E-213	5.59E-19	5.86E-78	0
	En iyi değer	1.42E-215	4.54E-20	3.24E-81	0

Çizelge 4.2'ye göre, altı adet (Ackley, Egg Crate, Griewank, Sphere, Step) kriter fonksiyonu için ortalama, standart sapma ve en iyi değer istatistiki sonuçları açısından ASO sezgisel-üstü algoritması ile sağlanmışken Rosenbrock ve Rastrigin kriter fonksiyonları için en iyi istatistiki sonuçlar ABC sezgisel-üstü algoritması ile sağlanmıştır. Schwefel kriter fonksiyonu için ise ASO ve ABC sezgisel-üstü algoritmasının sonuçları diğerlerinden üstün ve başarıları birbirine yakındır.

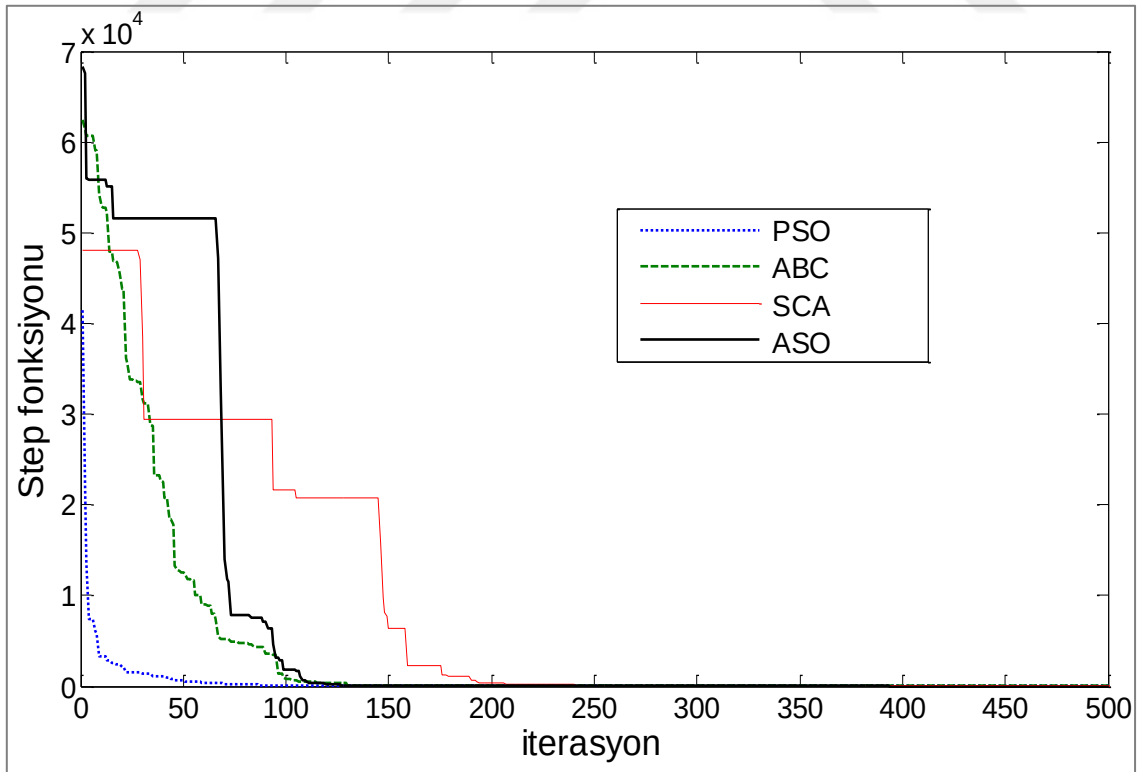
Şekil 4.9–4.16'da ABC, ASO, PSO ve SCA sezgisel-üstü algoritmalarının yakınsama eğrileri gösterilmiştir. Yakınsama eğrilerini gösteren şekiller herhangi bir sürüşte elde edilen optimizasyon ilerlemelerinin iteratif olarak gösteriminden oluşmuştur. Çizelge 4.2'deki istatistiki sonuçlara ve Şekil 4.11'deki yakınsama eğrilerine bakıldığında, ASO sezgisel-üstü algoritmasının modern ve kullanışlı bir algoritma olan SCA'ya ve ABC ile PSO sezgisel-üstü algoritmalarına oranla daha başarılı bir performans gösterdiği görülebilmektedir.



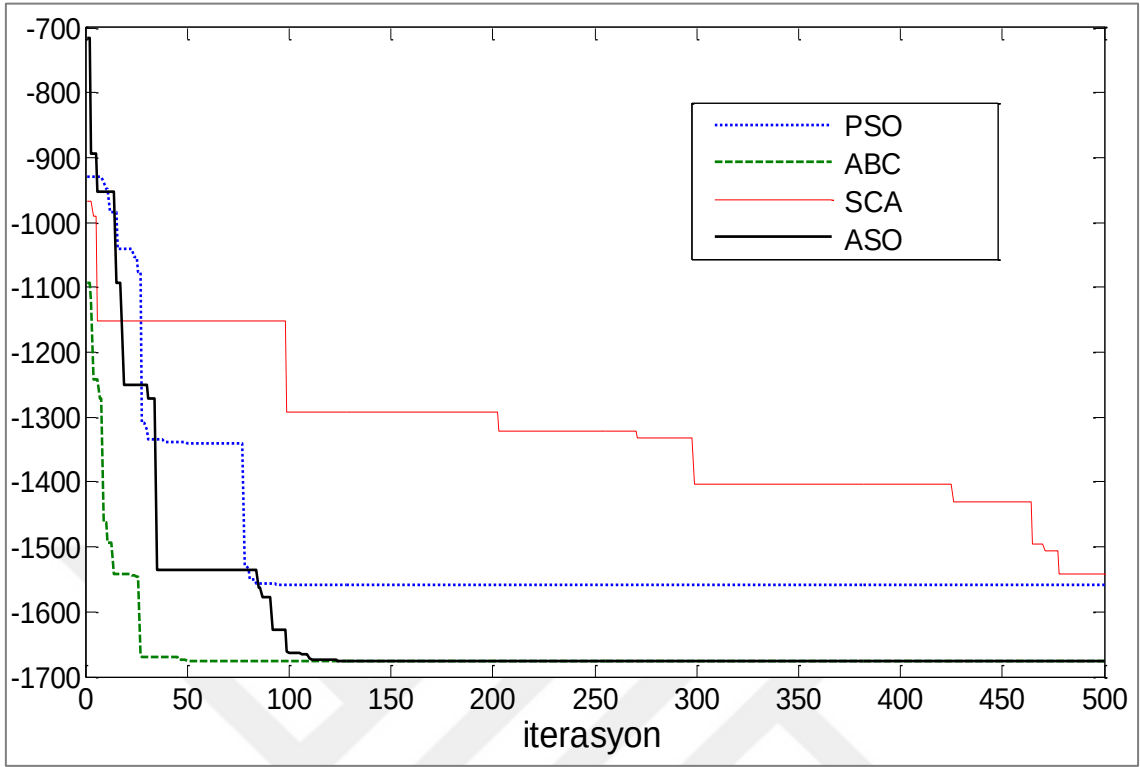
Şekil 4.9. Sphere fonksiyonunun optimizasyon süreci.



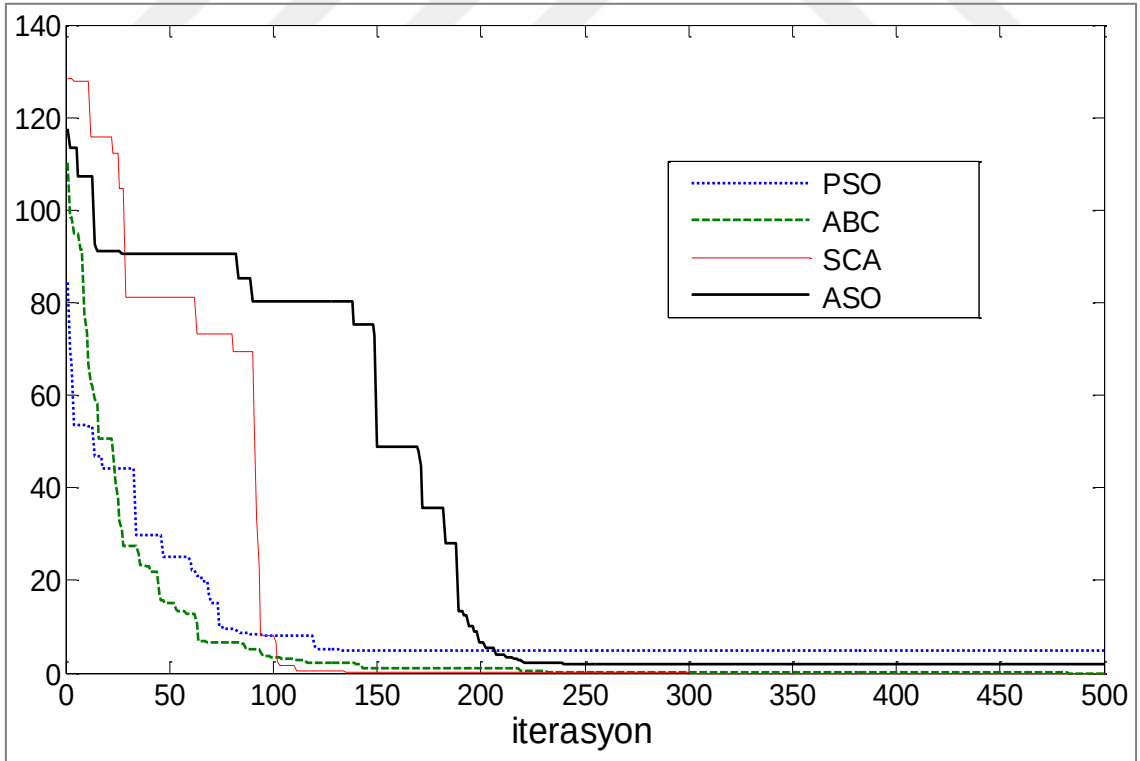
Şekil 4.10. Rosenbrock fonksiyonunun optimizasyon süreci.



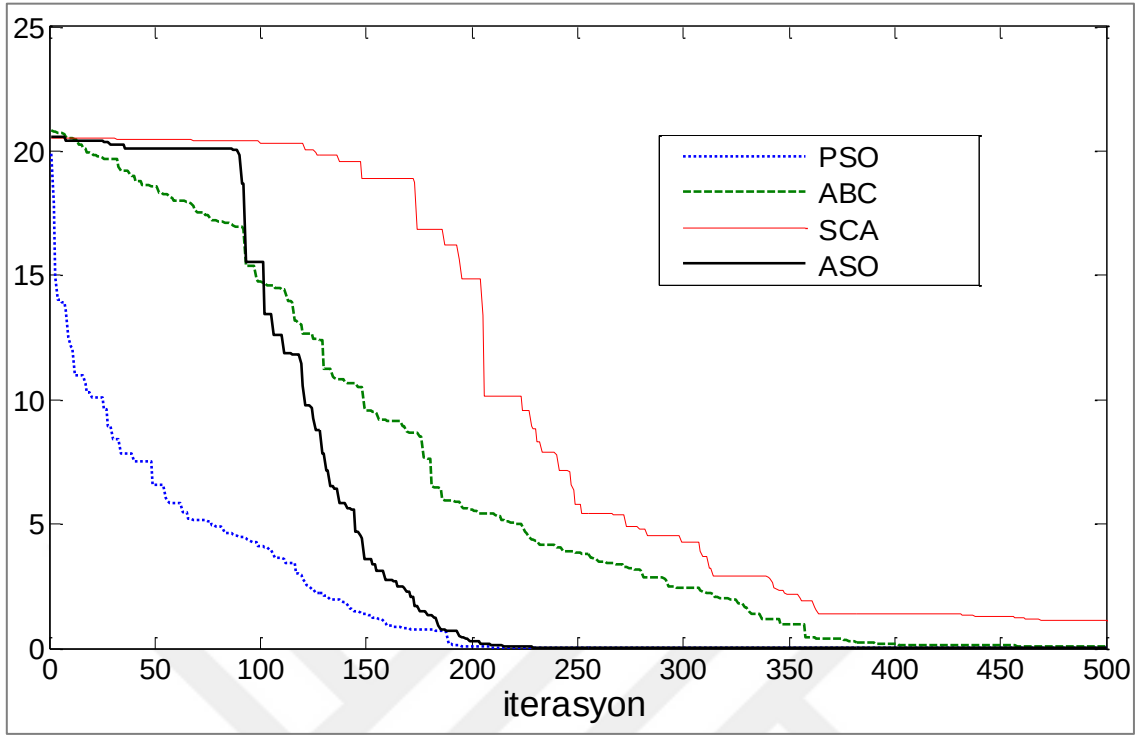
Şekil 4.11. Step fonksiyonunun optimizasyon süreci.



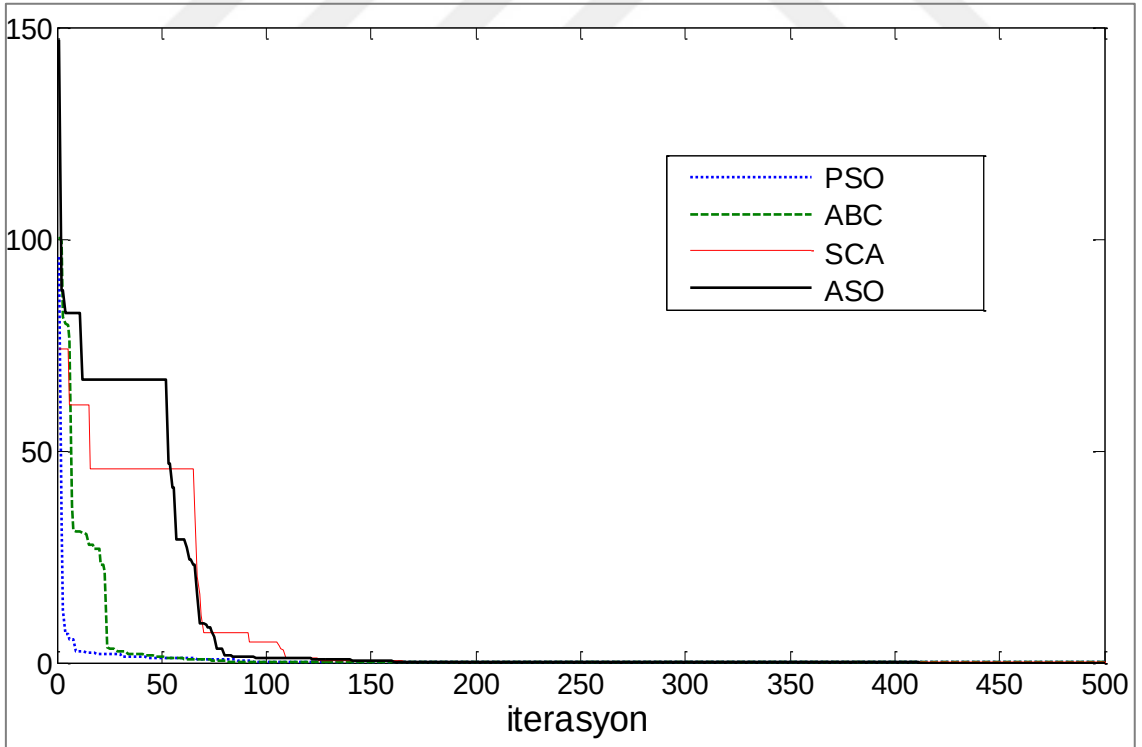
Şekil 4.12. Schwefel fonksiyonunun optimizasyon süreci.



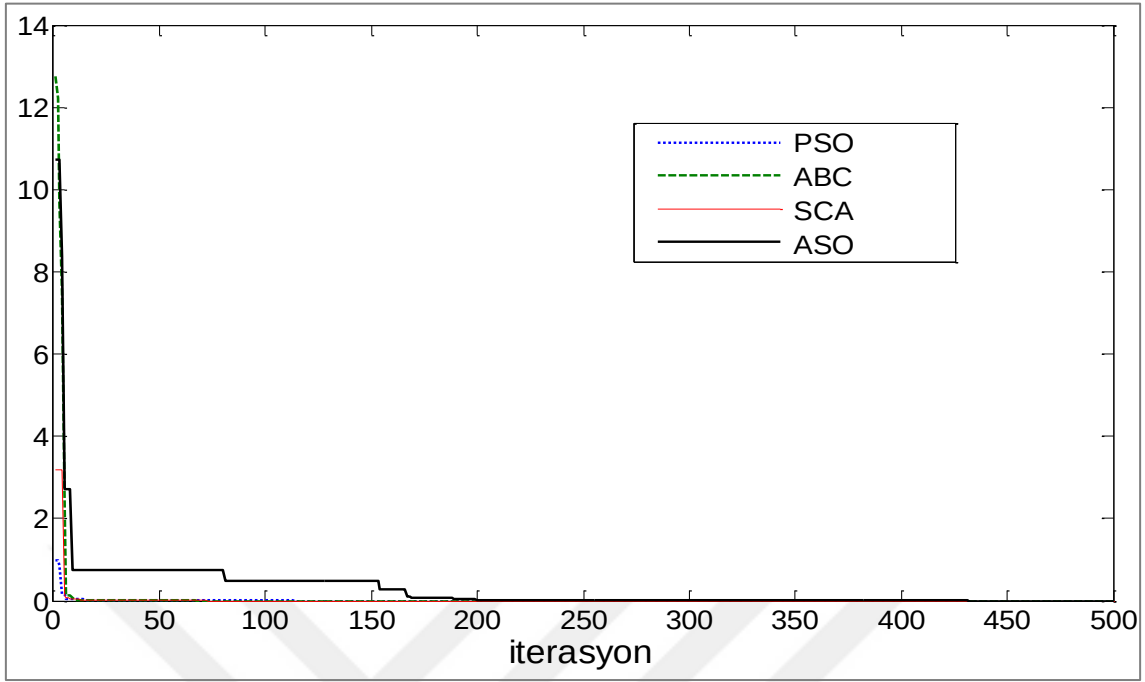
Şekil 4.13. Rastrigin fonksiyonunun optimizasyon süreci.



Şekil 4.14. Ackley fonksiyonunun optimizasyon süreci.



Şekil 4.15. Griewank fonksiyonunun optimizasyon süreci.



Şekil 4.16. Egg Crate fonksiyonunun optimizasyon süreci.

Bu çalışmada sürü zekâsının farklı varyantlarına sahip modern sezgisel-üstü algoritmaların ortalama, standart sapma ve en iyi değer gibi istatistiki sonuçlarının karşılaştırılması yapıldı. Yakınsama eğrileri grafiksel olarak değerlendirildi. Karşılaştırmada sonuçlar belli kısıtlara sahip kriter fonksiyonları aracılığıyla değerlendirildi ve tüm değerlerden elde edilen sonuçlara göre ASO modern sezgisel-üstü algoritmasının büyük oranda en iyi sonucu verdiği gözlemlenmiştir.

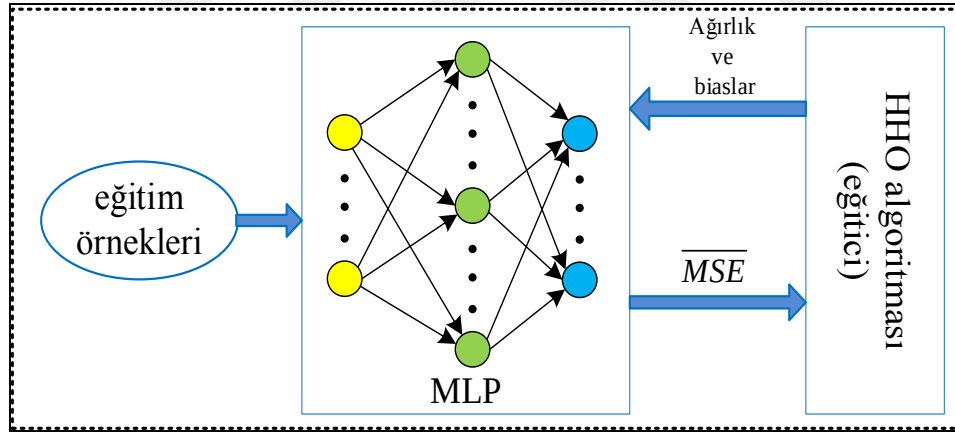
4.3. Harris Şahinleri Optimizasyonunu Kullanarak Çok Katmanlı Algılayıcı Eğitimi

Bu uygulamada bir sınıflandırıcı olarak kullanılan MLP eğitimi için eğitimci olarak modern sezgisel-üstü algoritmalarından HHO seçilmiş ve belli kriterlerle test edilmiştir. Aynı test aşamalarından SCA sezgisel-üstü algoritması da geçirilerek performans başarıları karşılaştırılmıştır. HHO ve SCA sezgisel-üstü algoritmaları yaygın kullanılan veri setlerinden XOR, Balloon, Iris, Breast Cancer ve Heart kullanılarak test edildi. Sonuçlar karşılaştırıldığında HHO algoritmasının sınıflama becerisinin de daha yüksek olduğu görüldü.

Bu uygulamada ortalama kare hatadır (mean square error, MSE) uygunluk fonksiyonu kullanıldı. Her iterasyonda ortalama MSE değeri aşağıdaki eşitlik kullanılarak hesaplanmıştır.

$$\overline{MSE} = \sum_{k=1}^q \sum_{i=1}^m (o_i^k - d_i^k)^2 q \quad (4.1)$$

Bu ifadede q eğitim örneklerinin sayısını, m çıkışların sayısını, d_i^k ve o_i^k sırasıyla k . eğitim örneği kullanıldığında i . giriş biriminin istenilen çıkışını ve gerçek çıkışını temsil etmektedir. Şekil 4.17, HHO algoritması kullanılarak bir MLP eğitim sürecini göstermektedir. Şekilden anlaşılabacağı üzere, tüm eğitim örnekleri için eğitmen (dasnışman) tarafından bulunan en iyi ağırlık ve bias değerleri MLP'ye verilir ve ortalama MSE değeri alınır. HHO sezgisel-üstü algoritması, her iterasyonda tüm eğitim örneklerinin ortalama MSE değerini minimize ederek ağırlık ve bias değerlerini günceller.



Şekil 4.17. HHO tabanlı MLP eğitimi.

XOR, Balloon, Iris, Breast Cancer ve Heart çalışmada kullanılan beş veri kümesidir. Çizelge 4.3, veri kümeleri hakkında bazı bilgileri özetlemektedir.

Çizelge 4.3. Sınıflandırma veri kümeleri ve özellikleri

Veri kümesi →	XOR	Balloon	Iris	Breast Cancer	Heart
Özelliklerin sayısı	3	4	4	9	22
Eğitim örneklerinin sayısı	8	16	150	599	80
Test örneklerinin sayısı	8	16	150	100	187
Sınıf sayısı	2	2	3	2	2
MLP yapısı	3-7-1	4-9-1	4-9-3	9-19-1	22-45-1

D, hem problemin boyutunu hem de değişken sayısını göstermek üzere, aşağıdaki eşitlikte gösterildiği şekilde hesaplanır.

$$D = (g \times h) + (h \times o) + h + o \quad (4.2)$$

Burada g , giriş katmanındaki düğüm sayısı, h , gizli katman (hidden layer) daki düğümlerin sayısı, o ise çıkış katmanındaki düğümlerin sayısıdır. Bu nedenle, XOR, Balloon, Iris, Breast Cancer ve Heart veri kümeleri için değişken sayısı sırasıyla 36, 55, 75, 210 ve 1081'dir.

MLP yapısının eğitiminde HHO ve SCA algoritmaları çalıştırılırken XOR ve Balloon için sürü büyüklüğü 50; Iris, Breast Cancer ve Heart için ise sürü büyüklüğü 200 alınmıştır. Veri kümelerinin tümü için maksimum iterasyon sayısı 250 seçilmiştir. Tüm veri kümeleri, eğitimci algoritmalarla 10 kez çalıştırılarak çözüldü ve elde edilen istatistiki sonuçlar Çizelge 4.4'te listelendi.

Çizelge 4.4. Sınıflandırma veri kümeleri için sonuçlar

Veri Kümesi ↓	HHO		SCA	
	MSE (Ortalama ± Standart sapma)	En iyi sınıflandırma oranı	MSE (Ortalama ± Standart sapma)	En iyi sınıflandırma oranı
XOR	1.29e-03±2.08e-03	%100	3.78e-02 ± 2.83e-02	%87.50
Balloon	3.24e-10±8.82e-10	%100	8.33e-06 ± 1.28e-05	%100
Iris	1.73e-02±9.49e-05	%93.33	1.27e-01 ± 3.17e-02	%88.67
Breast Cancer	1.44e-03±8.52e-05	%99	1.21e-02 ± 6.38e-03	%92
Heart	9.96e-02±9.49e-03	%82.50	1.63e-01 ± 2.06e-02	%78.75

Çizelge 4.4'te görüleceği gibi sayısal sonuçlardan anlaşılabileceği üzere, en düşük MSE ve en yüksek sınıflandırma oranı değerleri HHO sezgisel-üstü algoritmasıyla sağlanmıştır. Bu sonuçlar, HHO algoritmasının sömürü-keşif dengesini iyi sağladığını, yerel minimuma takılıp kalmadığını ve keşif düzeyinin yüksek olduğunu göstermiştir.

Bu çalışmanın sonucunda, modern sezgisel-üstü algoritmalarından HHO, MLP mimarisindeki bağlantı ağırlık ve bias değişkenlerinin en iyi değerlerinin bulunmasında

kullanılmıştır. Ortalama MSE uygunluk fonksiyonu olarak seçilmiştir. HHO tabanlı MLP eğiticisi, tüm veri setlerine başarıyla uygulandı ve elde edilen ortalama MSE ve sınıflandırma oranları aynı zamanda SCA tabanlı MLP eğiticisi ile karşılaştırıldı. Çizelge 4.4'teki sonuçlar, HHO tabanlı MLP eğiticisinin ortalama MSE ve sınıflandırma oranı açısından daha iyi performans sergilediğini göstermiştir.

4.4. hASO-SA Hibrit Algoritması ile MLP Eğitimi

4.4.1. hASO-SA algoritmasının istatistiki sonuçlarının karşılaştırılması

Karşılaştırmalar Sphere, Rosenbrock, Step, Quartic, Schwefel, Rastrigin, Ackley ve Griewank kriter fonksiyonları ve PSO, GA, SA, GSA, WDO ve ASO olasılıksal algoritmaları kullanılarak gerçekleştirilmiştir. Karşılaştırma için kullanılan algoritmalar arasında PSO, GA ve SA üç modern ve sık kullanılan sezgisel-üstü algoritma ve GSA, WDO ve o ASO ise yakın zamanda önerilen üç modern sezgisel-üstü algoritma bulunmaktadır.

Bu çalışmada SA algoritması için parametrelerden başlangıç sıcaklığı 0.1, sıcaklığı düşürme oranı 0.98 ve değişim oranı 0.5 olarak ayarlandı. GA algoritması, eşleşme için bir kromozom seçiminde Rulet Tekerleği tekniği, çaprazlama ve seçim parametrelerinin katsayıları olarak sırasıyla 0.8 ve 0.4 seçilmiştir. PSO algoritmasının parametrelerinden bilişsel ve sosyal sabitlerin her ikisi için 2, Durağanlık sabiti ise doğrusal olarak 0,8'den 0,2'ye düşecek şekilde ayarlanmıştır. GSA algoritmasının parametreleri, başlangıçtaki yerçekimi sabiti ve azalan katsayısı için sırasıyla 100 ve 20'ye ayarlanmıştır.

WDO algoritması için parametre değerleri RT katsayısı 3, yerçekimi sabiti 0.2 ve Coriolis etkisi 0.4'e ayarlanmıştır. İzin verilen maksimum hız 0,3 olarak ayarlanırken güncelleme eşitliklerindeki sabit 0,4'tür.

Önerilen hASO-SA algoritması, SA, GA, PSO, GSA, WDO ve ASO algoritmaları ile adil bir karşılaştırma elde etmek için seçilen sürü büyüklüğü 50 ve maksimum 1000 tekrar sayısı ile 50 kez çalıştırılmıştır. Listelenen algoritmalar kullanılarak kabul edilen kriter fonksiyonları için elde edilen istatistiki sonuçlar, en iyi sonuçlar Çizelge 4.5'te sunulmaktadır. Diğer algoritmalar ile karşılaştırıldığında, en iyi

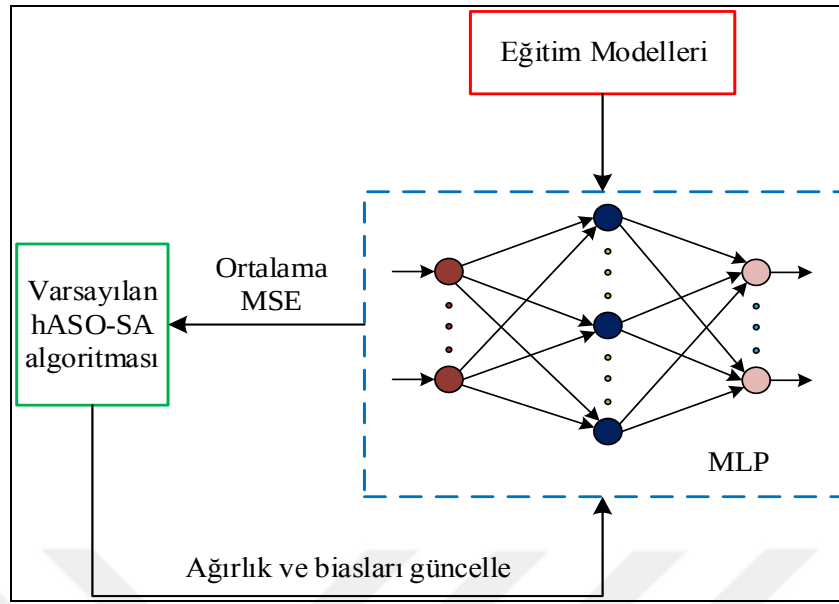
sonuçlar önerilen hASO-SA algoritması ile elde edilmiştir. Çizelge 4.5'te hASO-SA algoritmasının diğer algoritmalarla belirlenen kriter fonksiyonları aracılığıyla istatistiki sonuçları gösterilmiştir.

Çizelge 4.5. Sezgisel-üstü algoritmaların istatistiki sonuçlarının karşılaştırılması

$F(x)$	istatistikler	SA	GA	PSO	GSA	WDO	ASO	hASO-SA (önerilen)
$F_1(x)$	En iyi D.	7.76E-14	2.98E-03	9.56E-06	1.06E-17	0.00E+00	3.52E-22	8.36E-99
	ortalama	2.04E-13	1.02E-02	1.46E-04	2.11E-17	0.00E+00	2.68E-21	5.57E-96
	Std. sapma	6.14E-14	5.07E-03	1.19E-04	6.67E-18	0.00E+00	3.65E-21	3.02E-95
$F_2(x)$	En iyi D.	2.32E+01	9.69E+00	2.62E+01	2.57E+01	2.80E+01	1.66E+01	4.12E-05
	ortalama	1.06E+03	9.71E+01	1.34E+02	2.81E+01	2.82E+01	2.48E+01	2.03E-02
	Std. sapma	2.05E+03	1.28E+02	1.28E+02	1.13E+01	1.69E-01	5.16E-01	2.84E-02
$F_3(x)$	En iyi D.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	ortalama	5.67E-01	0.00E+00	1.33E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std. sapma	7.28E-01	0.00E+00	3.46E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
$F_4(x)$	En iyi D.	6.25E-02	1.32E-02	2.98E-02	7.00E-03	1.43E-06	3.61E-02	2.77E-12
	ortalama	1.24E-01	5.05E-02	6.63E-02	2.08E-02	6.10E-05	3.56E-02	1.19E-10
	Std. sapma	3.97E-02	2.18E-02	1.95E-02	7.72E-03	4.49E-05	1.95E-02	7.93E-11
$F_5(x)$	En iyi D.	-1.01E+04	-8.15E+03	-6.79E+03	-3.49E+03	-7.59E+03	-5.56E+03	-1.25E+04
	ortalama	-9.29E+03	-6.80E+03	-5.20E+03	-2.65E+03	-5.84E+03	-7.43E+03	-1.24E+04
	Std. sapma	4.02E+02	6.33E+02	5.29E+02	3.42E+02	8.57E+02	4.22E+02	1.53E+02
$F_6(x)$	En iyi D.	2.69E+01	6.09E+00	1.73E+01	7.96E+00	1.56E+01	0.00E+00	0.00E+00
	ortalama	5.44E+01	1.25E+01	2.93E+01	1.51E+01	5.77E+01	0.00E+00	0.00E+00
	Std. sapma	1.38E+01	2.92E+00	6.88E+00	4.44E+00	2.12E+01	0.00E+00	0.00E+00
$F_7(x)$	En iyi D.	8.29E-08	1.07E-02	5.49E-04	2.96E-09	8.88E-16	1.13E-11	8.88E-16
	ortalama	3.44E-01	2.12E-02	7.43E-03	3.69E-09	8.88E-16	3.00E-11	8.88E-16
	Std. sapma	4.47E-01	4.55E-03	1.42E-02	3.96E-10	0.00E+00	2.15E-11	3.68E-31
$F_8(x)$	En iyi D.	3.16E-06	6.12E-03	5.88E-05	1.94E+00	0.00E+00	0.00E+00	0.00E+00
	ortalama	1.24E-02	1.84E-02	2.28E-02	4.47E+00	9.89E-03	0.00E+00	0.00E+00
	Std. sapma	1.03E-02	9.77E-03	2.74E-02	2.05E+00	2.19E-02	0.00E+00	0.00E+00

4.4.2. hASO-SA algoritması tabanlı MLP eğitimi

Şekil 4.18, önerilen hibrid hASO-SA algoritması kullanılarak MLP eğitiminin genel sürecini temsil eder. Görülebileceği gibi, hASO-SA algoritması MLP'ye ağırlık / sapma sağlar ve tüm eğitim örnekleri için ortalama MSE alır. HASO-SA algoritması, tüm eğitim örneklerinin ortalama MSE'sini en aza indirmek için ağırlıkları ve sapmaları tekrar tekrar değiştirir.



Şekil 4.18. hASO-SA algoritması tabanlı MLP eğitimi.

Önerilen hASO-SA algoritmasını karşılaştırmak için beş sınıflandırma veri seti (XOR, Balon, İris, Kalp Hastalıkları ve Meme Kanseri) kullanıldı. Her aday çözüm, eğitim algoritmasında rastgele bir dizi $[-10,10]^d$ kapalı aralığından seçildi. Aday çözümlerin popülasyon büyüklüğü İris, Kalp ve Meme Kanseri için 200, XOR ve balon sınıflandırma problemleri için 50 olarak seçilmiştir. Veri kümeleri Çizelge 4.6’da sunulduğu gibi sınıflandırıldı.

Çizelge 4.6. Sınıflandırma veri setleri

Veri Setleri	Özellik Sayısı	Eğitim Örnekleri Sayısı	Test Örnekleri Sayısı	Sınıf Sayısı
XOR	3	8	8	2
Balloon	4	16	16	2
Iris	4	150	150	3
Breast cancer	9	599	100	2
Heart	22	80	187	2

Algoritma veri setlerine 10 kez uygulandı. Bu veri kümelerinden elde edilen sonuçlar, algoritmanın son yinelemesinde elde edilen en iyi ortalama kare hatasının (MSE) ortalaması (ORT.) ve standart sapmasını (STD) sağlayan Çizelge 4.7’den Çizelge 4.11’e kadar gösterilmektedir. Açıkçası, MSE’nin son iterasyonda daha düşük ortalama ve standart sapması daha iyi performansın bir göstergesidir. Dikkate alınan veri kümelerinin sonuçları aşağıda çizelgeler eşliğinde verilmiştir.

XOR için sayısal sonuçlar, önerilen hASO-SA algoritmasının performansının, optimal olmayan çözümlerden kaçınarak bu problemi çözmede daha iyi olabileceğini açıkça gösteren Çizelge 4.7’de sunulmaktadır.

Çizelge 4.7. XOR sınıflandırma probleminin deneysel sonuçları

Eğitim algoritması	MSE (ORT $\pm$ STD)	Sınıflandırma oranı (%)
hASO-SA (önerilen)	2.31E-04 $\pm$ 1.63E-04	100.00
ASO	5.72E-03 $\pm$ 2.37E-02	100.00
GWO	9.41E-03 $\pm$ 2.95E-02	100.00
PSO	8.40E-02 $\pm$ 3.59E-02	37.50
ACO	1.80E-01 $\pm$ 2.53E-02	62.50
PBIL	3.02E-02 $\pm$ 3.97E-02	62.50
ES	1.19E-01 $\pm$ 1.16E-02	62.50

Ballon veri kümesi için elde edilen sonuçlar, hASO-SA’nın diğer yöntemlere oranla minimum hatayı sağladığını gösteren Çizelge 4.8’de sunulmaktadır. Tüm algoritmaların sınıflandırma oranları aynıdır ve %100’dür.

Çizelge 4.8. Balloon sınıflandırma probleminin deneysel sonuçları

Eğitim algoritması	MSE (ORT $\pm$ STD)	Sınıflandırma oranı (%)
hASO-SA (önerilen)	4.66E-16 $\pm$ 3.71E-15	100.00
ASO	3.47E-08 $\pm$ 1.53E-07	100.00
GWO	9.38E-15 $\pm$ 2.81E-14	100.00
PSO	5.85E-04 $\pm$ 7.49E-04	100.00
ACO	4.85E-03 $\pm$ 7.76E-03	100.00
PBIL	2.49E-05 $\pm$ 5.27E-05	100.00
ES	1.91E-02 $\pm$ 1.70E-01	100.00

Iris için elde edilen sonuçlar Çizelge 4.9’da verilmektedir. hASO-SA, kare hata ve sınıflandırma oranının ortalama değerine dayanarak MLP’yi eğitmek için daha iyi performans sağlar. Karşılaştırmalı sonuçlar, diğer algoritmalara oranla hASO-SA’nın üstün performansını gösterdi.

Çizelge 4.9. Iris sınıflandırma probleminin deneysel sonuçları

Eğitim algoritması	MSE (ORT $\pm$ STD)	Sınıflandırma oranı (%)
hASO-SA (önerilen)	4.66E-16 $\pm$ 3.71E-15	100.00
ASO	3.47E-08 $\pm$ 1.53E-07	100.00
GWO	9.38E-15 $\pm$ 2.81E-14	100.00
PSO	5.85E-04 $\pm$ 7.49E-04	100.00
ACO	4.85E-03 $\pm$ 7.76E-03	100.00
PBIL	2.49E-05 $\pm$ 5.27E-05	100.00
ES	1.91E-02 $\pm$ 1.70E-01	100.00

Meme kanseri için elde edilen sonuçlar sonuçları Çizelge 4.10’da verilmiştir.

Görülebileceği gibi, hASO-SA, ortalama değer ve %100 sınıflandırma oranı açısından en iyi ortalama kare hatasını sağlar; bu, önerilen algoritmanın yerel optima’dan kaçma açısından daha iyi arama yeteneğinin açık bir göstergesidir.

Çizelge 4.10. Meme Kanseri sınıflandırma probleminin deneysel sonuçları

Eğitim algoritması	MSE (ORT $\pm$ STD)	Sınıflandırma oranı (%)
hASO-SA (önerilen)	1.04E-03 $\pm$ 4.68E-05	100.00
ASO	3.47E-03 $\pm$ 1.64E-03	99.00
GWO	1.20E-03 $\pm$ 7.45E-05	99.00
PSO	3.49E-02 $\pm$ 2.47E-03	11.00
ACO	1.35E-02 $\pm$ 2.14E-03	40.00
PBIL	3.20E-02 $\pm$ 3.07E-03	7.00
ES	4.03E-02 $\pm$ 2.47E-03	6.00

Bir kalp hastasının durumu, veri kümesinin ikili formu kullanılarak normal veya normal değil olarak ifade edilebilir. Sonuçlar çizelge 4.11’de gösterilmektedir. Önerilen hASO-SA’nın diğer algoritmalarından daha iyi sonuçlar ve sınıflandırma oranı sağlayabildiğini açıkça göstermektedir.

Çizelge 4.11. Kalp hastalıkları sınıflandırma probleminin deneysel sonuçları

Eğitim algoritması	MSE (ORT $\pm$ STD)	Sınıflandırma oranı (%)
hASO-SA (önerilen)	8.57E-02 $\pm$ 9.46E-03	75.25
ASO	9.52E-02 $\pm$ 1.38E-02	72.75
GWO	1.23E-01 $\pm$ 7.70E-03	75.00
PSO	1.86E-01 $\pm$ 8.94E-03	68.75
ACO	2.28E-01 $\pm$ 4.98E-03	0.00
PBIL	1.54E-01 $\pm$ 1.82E-02	45.00
ES	1.92E-01 $\pm$ 1.52E-02	71.25

ASO ve SA algoritmalarına dayanan modern bir hibrid sezgisel-üstü algoritma olan hASO-SA, SA algoritmasının ASO algoritmasına gömülmesi ile bu çalışmada önerilmiştir. Önerilen bu özel yapılandırma ASO’nun arama yeteneğinin geliştirilmesine yardımcı oldu. Önerilen hibrit hASO-SA algoritmasının performansını gözlemlemek için farklı yapıdaki çeşitli optimizasyon problemleri kullanılmıştır. Algoritmanın ilk değerlendirmesi için Sphere, Rosenbrock, Step, Quartic, Schwefel, Rastrigin, Ackley ve Griewank’in kriter fonksiyonları benimsenmiştir. Ayrıca, önerilen algoritmanın üstünlüğünü göstermek için PSO, GA, GSA ve WDO ile SA ve orijinal ASO ile performans karşılaştırmaları yapılmıştır. Teklif edilen algoritmanın farklı

nitelikte bir optimizasyon problemi için yeteneğini gözlemlemek amacıyla MLP eğitimi için önerilen algoritma kullanılarak daha fazla değerlendirme yapılmıştır. Bunu yapmak için XOR, Balon, Iris, Meme kanseri ve Kalp hastalıkları veri setleri kullanıldı ve önceki duruma benzer şekilde, algoritmanın performansı ASO, GWO, PSO, ACO, ES ve PBIL algoritmalarının diğer sezgisel-üstü algoritmalar ile karşılaştırıldı. Önerilen hASO-SA algoritmasının farklı nitelikteki problemlere uygulanması, bu algoritmanın çeşitli optimizasyon problemleri için güçlü olduğunu göstermiştir.



5. SONUÇ

Bu tezde modern sezgisel-üstü algoritmalar yoluyla yapay sinir ağlarının eğitimi sağlandı. Eğitim için bir YSA mimari biçimi olan MLP seçildi. MLP'ler doğrusal olmayan problemlerin çözümünde iyi bir sınıflandırıcı olarak kullanıldı. MLP eğitiminde UCI'den alınan veri setleri ağına tanıtıldı. Eğitim metodu olarak danışmanlı eğitim metodu uygulandı. Ağı eğitimi için eğitmen olarak olasılıksal çözüm üreten sezgisel-üstü algoritmalar kullanıldı. Olasılıksal yöntemlerin üstünlüğü yerel optimadan ziyade küresel optimaya yönelmeleri, iterasyona dayalı hareket ederek en iyi çözümü bulana kadar eğitime devam etmeleri ve gradyan yöntemin aksine çözümde türeve ihtiyaç duymayan basit hesaplama tekniklerine sahip olmalarıdır. Sezgisel-üstü algoritmalar sürü zekâsına dayalı olmaları, birden fazla adayla çözüme başlamaları, keşif ve sömürü aşamalarını kullanarak küresel çözümü bulma kabiliyetleri ve rekabetçi yapılarından dolayı en iyi çözümü bulmada etkin araçlardır.

MLP'ler sınıflandırma problemlerine başarıyla uygulanmışlardır. Bu bağlamda sezgisel-üstü algoritmaların sınıflandırma gücü ve uygulanacak problemin özelliklerinin uygunluğu, MLP mimarisinin özellikleri gibi hassasiyetler çalışmanın başarısını ve gerçek dünya problemlerine uygulanabilirliğini belirleyen önemli kriterlerdir.

Tezde dört farklı uygulama yapılarak algoritmanın öncelikle sınıflandırma başarısı ve problemin optimizasyonunu sağlaması ile YSA eğitimindeki başarısı değerlendirilmiştir. Ek olarak bir hibrit algoritma olan hASO-SA önerilmiştir. hASO-SA algoritması ile sezgisel-üstü algoritmaların arama gücü artırılarak daha iyi performans gösterdiği bulgularla gösterilmiştir.

Tezde ortaya çıkartılan sonuçlar sıralandığında, birinci uygulamada farklı özelliklere sahip global optimizasyon problemlerinin çözümünde modern bir sezgisel-üstü algoritma olan HHO algoritması ile yine modern sezgisel-üstü farklı algoritmalar karşılaştırılarak bir performans değerlendirilmesi yapılmıştır. HHO algoritmasının keşif ile sömürü arasındaki dengeyi korumadaki üstün yeteneği çalışmada daha etkin sonuçlara gidilmesini sağlamıştır.

İkinci uygulamada seçilen modern sezgisel-üstü algoritmaların kısıtlı optimizasyon problemlerine uygulanması ve optimizasyon süreçleri değerlendirilmiştir.

ASO sezgisel-üstü algoritmasının istatistiki bulgular ve yakınsama eğrilerinde görülen yakınsama hızı daha üstün bir performans sergilediğini göstermiştir. Bununla birlikte, kaotik haritalar yaklaşımı veya karşıt tabanlı öğrenme stratejisi uygulanarak ASO algoritmasının verimliliği ve etkinliği daha da artırılabilir.

Üçüncü uygulamada daha önce üstün başarısı görülen HHO algoritmasının beş veri kümesi kullanılarak eğitilen MLP’de bir sınıflandırıcı olarak daha etkin olduğu veri kümelerinin sınıflandırılmasında kullanılan diğer sezgisel-üstü algoritmalara göre daha üstün bir sınıflandırıcı olduğu ispatlanmıştır.

Dördüncü uygulamada ASO’nun arama özelliklerini güçlendirmek için SA’nın yerel çözümden kaçınma özelliğini kullanarak bulunan en iyi olmayan sonuçlar için mevcut çözümün komşuluğunda bir çözüm öneren yeni bir hibrit uygulaması olan hASO-SA algoritması ile performans karşılaştırılması yapıldı ve MLP eğitiminde kullanıldı. Tüm kriter fonksiyonlarında en iyi performansı hASO-SA algoritması gösterdi. Kendi bileşenleriyle karşılaştırıldığında da güçlendirilmiş melez yapısıyla daha iyi sonuçlara ulaşıldı. MLP eğitimi için seçilen veri setlerinde en iyi sınıflandırma performansı da hASO-SA tarafından sağlandı. Hibrit çalışmaları sezgisel-üstü güncel algoritmaların güçlendirilerek bir anlamda modernizasyonunu sağlamakta ve zaman kaybı gibi birçok maliyetin de düşürülmesine katkı sunmaktadır.

Tüm uygulamalardan çıkan ortak sonuç modern sezgisel-üstü algoritmalar ve önerilen hASO-SA hibrit algoritması hem optimizasyon problemlerinde optimum noktaya en yakın çözümü üretebilme yeteneği hem de MLP eğitiminde çok yüksek sınıflandırma kabiliyetine sahip olduklarını göstermiştir. Bu göstergeler sezgisel-üstü algoritmaların birçok alandaki mühendislik çalışmaları ve gerçek dünya problemlerinin çözümünde ve diğer birçok alanda optimizasyon algoritması olarak başarılı bir şekilde kullanılabileceğini göstermiştir.

KAYNAKLAR

- Abdeslam, A., El Bouanani, F., Ben-Azza, H., 2014. Four parallel decoding schemas of product block codes. *Transactions on Networks and Communications*, 2: 49-69.
- Abdulkadir, R. A., Imam, K. A., ve Jibril, M. B., 2017. Simulation of back propagation neural network for iris flower classification. *American Journal of Engineering Research*, 6 (1): 200-205.
- Abedifar, V., Eshghi, M., Mirjalili, S., Mirjalili, S. M., 2013. An optimized virtual network mapping using PSO in cloud computing. *21st Iranian Conference on Electrical Engineering (ICEE)*. 1-6. IEEE.
- Abido, M. A., 2002. Optimal power flow using tabu search algorithm. *Electric Power Components and Systems*, 30 (5): 469-483.
- Abraham, A., 2005. AI Signal Processing Techniques, Chap.3. *Handbook of measuring system design* (Editor: Sydenham P.H., Thorn R., ISBN: 0-470-02143-8). John Wiley & Sons, London, 1358.
- Agwa, A. M., El-Fergany, A. A., ve Sarhan, G. M., 2019. Steady-state modeling of fuel cells based on atom search optimizer. *Energies*, 12(10): 1884.
- Alam, M. N., 2016. Particle swarm optimization: Algorithm and its codes in matlab, *ResearchGate*, 1-10.
- Alatas, B., Akin, E., Ozer, A. B., 2009. Chaos embedded particle swarm optimization algorithms. *Chaos, Solitons & Fractals*, 40 (4):1715-1734.
- Almagboul, M. A., Shu, F., Qian, Y., Zhou, X., Wang, J., & Hu, J., 2019. Atom search optimization algorithm based hybrid antenna array receive beamforming to control sidelobe level and steering the null. *AEU-International Journal of Electronics and Communications*, 111.
- Almufti, S. M., 2019. Historical survey on metaheuristics algorithms, *International Journal of Scientific World*, 7(1): 1-12.
- Al-Roomi, Ali R., 2015. Unconstrained Single-Objective Benchmark Functions Repository. *Electrical and Computer Engineering*, Dalhousie University, Halifax, Nova Scotia, Canada.
- Amari, S. I. (1993). Backpropagation and stochastic gradient descent method. *Neuro-computing*, 5(4-5): 185-196.
- Andradóttir, S., 1999. Accelerating the convergence of random search methods for discrete stochastic optimization. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 9(4), 349-380.
- Antoniou, A., & Lu, W. S., 2007. *Practical Optimization: Algorithms and Engineering Applications*. Springer Science & Business Media.
- Ashraf, A., Bangyal, W. H., Rauf, H. T., Pervaiz, S., Ahmad, J., 2020. Training of artificial neural network using new initialization approach of particle swarm optimization for data classification. *International Conference on Emerging Trends in Smart Technologies (ICETST)*, IEEE. 1-6.
- Askarzadeh, A., Dos Santos Coelho, L., Klein, C. E., Mariani, V. C., 2016. A population-based simulated annealing algorithm for global optimization. *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 004626-004633.

- Aydilek, I. B., 2018. A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems. *Applied Soft Computing*, **66**, 232-249.
- Azzini, A., Tettamanzi, A. G., 2011. Evolutionary ANNs: A state of the Art Survey, *Intelligenza Artificiale*, **5**(1): 19-35.
- Babayiğit, B., ve Özdemir, R., 2012. Modifiye yapay arı koloni algoritması ile nümerik fonksiyon optimizasyonu, *ELECO '2012 Elektrik –Elektronik ve Bilgisayar Mühendisliği Sempozyumu*, 618-622.
- Bahrami, S., ve Ardejani, F. D., 2016. Prediction of pyrite oxidation in a coal washing waste pile using a hybrid method, coupling artificial neural networks and simulated annealing (ANN/SA). *Journal of cleaner production*, **137**, 1129-1137.
- Bairathi, D., Gopalani, D., 2017. "Opposition-based sine cosine algorithm (OSCA) for training feed-forward neural networks". *13th International Conference on Signal-Image Technology ve Internet-Based Systems (SITIS)*. 438-444.
- Bairathi, D., Gopalani, D., 2019. Numerical optimization and feed-forward neural networks training using an improved optimization algorithm: multiple leader salp swarm algorithm. *Evolutionary Intelligence*, 1-17.
- Bansal, P., Kumar, S., Pasrija, S., Singh, S., 2020. A hybrid grasshopper and new cat swarm optimization algorithm for feature selection and optimization of multi-layer perceptron. *Soft Computing*, 1-27.
- Barnard, E. (1992). Optimization for training neural nets. *IEEE transactions on neural networks*, **3**(2): 232-240.
- Barron, A. R., 1989. Statistical properties of artificial neural networks. *Proceedings of the 28th IEEE Conference on Decision and Control*, 280-285.
- Bas E., 2016. The training of multiplicative neuron model based artificial neural networks with differential evolution algorithm for forecasting. *Journal of Artificial Intelligence and Soft Computing Research*, **6**(1): 5-11.
- Bas, E. 2016. The training of multiplicative neuron model based artificial neural networks with differential evolution algorithm for forecasting. *Journal of Artificial Intelligence and Soft Computing Research*, **6**(1): 5-11.
- Basheer, I. A., ve Hajmeer, M. (2000). Journal of microbiological methods. *Artificial neural networks: fundamentals, computing, design, and application*, **43**: 3-31.
- Bayraktar, Z., 2019. Adaptive Wind Driven Optimization Trained Artificial Neural Networks. *arXiv preprint arXiv:1911.08942*.
- Bayraktar, Z., Komurcu, M., Werner, D. H., 2010. Wind Driven Optimization (WDO): A novel nature-inspired optimization algorithm and its application to electromagnetics. *IEEE antennas and propagation society international symposium*, 1-4.
- Beheshti, Z., Shamsuddin, S. M. H., Beheshti, E., Yuhani, S. S., 2014. Enhancement of artificial neural network learning using centripetal accelerated particle swarm optimization for medical diseases diagnosis. *Soft Computing*, **18**(11): 2253-2270.
- Bhesdadiya, R., Jangir, P., Jangir, N., Trivedi, I. N., Ladumor, D., 2016. Training multilayer perceptron in neural network using whale optimization algorithm. *Indian Journal of Science and Technology*, **9**(19): 28–36.

- Blackstock, F. C., Lareau, S. C., Nici, L., ZuWallack, R., Bourbeau, J., Buckley, M., Kelly, W., (2018). Chronic obstructive pulmonary disease education in pulmonary rehabilitation. An official american thoracic society/thoracic society of Australia and New Zealand/canadian thoracic society/british thoracic society workshop report. *Annals of the American Thoracic Society*, **15**(7): 769-784.
- Blake, C., Merz, C.J., UCI “Repository of Machine Learning Databases”, <http://www.ics.uci.edu/~mllearn/MLRepository.html>, 1998.
- Borovinskiy, V., Ol, P., 2009. Classification of Iris data set. *University of Ljubljana*.
- Bui, D. T., Moayedi, H., Kalantar, B., Osouli, A., Pradhan, B., Nguyen, H., ve Rashid, A. S. A., 2019. A novel swarm intelligence—Harris hawks optimization for spatial assessment of landslide susceptibility. *Sensors*, **19**(16): 3590.
- Capparini, F. S., 2019. Simulated Annealing in NetLogo. <http://www.cs.us.es/~fsancho/?e=206>
- Chaudhuri, B. B., & Bhattacharya, U., 2000. Efficient training and improved performance of multilayer perceptron in pattern classification. *Neurocomputing*, **34**(1-4):11-27.
- Chen, Y., Yu, G., Long, Y., Teng, J., You, X., Liao, B. Q., ve Lin, H., 2019. Application of radial basis function artificial neural network to quantify interfacial energies related to membrane fouling in a membrane bioreactor. *Bioresource technology*, **(293)**: 122103.
- Choong, S. S., Wong, L. P., Lim, C. P., 2019. An artificial bee colony algorithm with a modified choice function for the Traveling Salesman Problem. *Swarm and evolutionary computation*, **(44)**: 622-635.
- Chow, T. S., 1978. Testing software design modeled by finite-state machines. *IEEE transactions on software engineering*, **(3)**: 178-187.
- Chunekar, V. N., ve Ambulgekar, H. P. 2009. Approach of neural network to diagnose breast cancer on three different data set. In *2009 International Conference on Advances in Recent Technologies in Communication and Computing*. IEEE. 893-895.
- ÇİHAN, A., FIĞLALI, A., 2018. Metasezgisel Yöntemlerde Parametre Optimizasyonu. *Mühendislikte Yapay Zeka Uygulamaları*, **2**, Sakarya Üniversitesi Kütüphanesi Yayınevi. 11-18.
- Cuevas, E., Cienfuegos, M., Zaldívar, D., Pérez-Cisneros, M., 2013. A swarm algorithm inspired in the behavior of the social-spider. *Expert Systems with Applications*, **40**(16): 6374-6384.
- Çavdar, T., 2016. PSO tuned ANFIS equalizer based on fuzzy C-means clustering algorithm. *AEU-International Journal of Electronics and Communications*, **70**(6): 799-807.
- Çelik, Y., Yıldız, İ., Karadeniz, A. T., 2019. Son üç yılda geliştirilen metasezgisel algoritmalar hakkında kısa bir inceleme. *Avrupa Bilim ve Teknoloji Dergisi*, 463-477.
- Darwin, C., 1929. The Origin of Species by Means of Natural Selection or The Preservation of Favored Races in the Struggle for Life, *The Book League of America*.
- Das, P. K. (2019). Hybridization of Kidney-Inspired and sine–cosine algorithm for multi-robot path planning. *Arabian Journal for Science and Engineering*, 1-18.

- Das, R., Turkoglu, I., Sengur, A., 2009. Effective diagnosis of heart disease through neural networks ensembles. *Expert systems with applications*, 36(4), 7675-7680.
- De Castro, L. N., Von Zuben, F. J., 2000. The clonal selection algorithm with engineering applications. *Proceedings of GECCO*, (2000): 36-39.
- Dembo, R. S., Eisenstat, S. C., Steihaug, T., 1982. Inexact newton methods. *SIAM Journal on Numerical analysis*, 19(2): 400-408.
- Dokeroglu, T., Sevinc, E., Kucukyilmaz, T., ve Cosar, A., 2019. *A survey on new generation metaheuristic algorithms*. *Computers ve Industrial Engineering*, 137, 106040.
- Dorigo, M., Stützle, T., 2019. Ant colony optimization: overview and recent advances. *Handbook of metaheuristics*, Springer, Cham. 311-351.
- Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE computational intelligence magazine*, 1(4): 28-39.
- Dorigo, M., Di Caro, G., 1999. Ant colony optimization: a new meta-heuristic. *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*, (2): 1470-1477, IEEE.
- Du, K. L., Swamy, M. N. S., 2016. Search and optimization by metaheuristics. *Techniques and Algorithms Inspired by Nature*; Birkhauser: Basel, Switzerland.
- Duman, E., Uysal, M., Alkaya, A. F., 2011. Migrating birds optimization: a new meta-heuristic approach and its application to the quadratic assignment problem. *European Conference on the Applications of Evolutionary Computation*, Springer, Berlin, Heidelberg. 254-263.
- Eberhart, R., ve Kennedy, J., 1995. Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks* Vol. 4 Citeseer. 1942-1948.
- Eberhart, R., ve Kennedy, J., 1999. A new optimizer using particle swarm optimization. *Proceedings of the 1995 Sixth International Symposium on Micro Machine and Human Science*. 39-43.
- Eiben, A. E., & Smith, J. E., 2015. What is an evolutionary algorithm? *Introduction to Evolutionary Computing*. Springer, Berlin, Heidelberg. 25-48.
- Eker, E., Kayri, M., & Ekinci, S., 2019a. Kısıtlı optimizasyon problemlerinin çözümü için atom arama optimizasyon algoritması. *DÜMF Mühendislik Dergisi*. 10(3): 841-851.
- Eker, E., Kayri, M., Ekinci, S. 2019b. Investigation of performance of hho algorithm in solving global optimization problems. *2019 International Artificial Intelligence and Data Processing Symposium (IDAP)*. IEEE. 1-6.
- Ekinci, S., 2016. Application and comparative performance analysis of pso and abc algorithms for optimal design of multi-machine power system stabilizers. *Gazi University Journal of Science*, 29 (2): 323-334.
- Ekinci, S., Hekimoğlu, B., Demirören, A., Kaya, S., 2019. Harris Hawks Optimization Approach for Tuning of FOPID Controller in DC-DC Buck Converter. In *2019 International Artificial Intelligence and Data Processing Symposium. IDAP*. 1-9. IEEE.
- Elmore, J. G., Wells, C. K., Lee, C. H., Howard, D. H., & Feinstein, A. R. (1994). Variability in radiologists' interpretations of mammograms. *New England Journal of Medicine*, 331(22): 1493-1499.

- El-Rashidy, M., Taha, T., Ayad, N., Sroor, H., 2011. An Effective Intelligent Model for Medical Diagnosis. *International Journal of Computer Science Issues (IJCSI)*, 8(5): 130.
- El-Shorbagy, M. A., Farag, M. A., Mousa, A. A., El-Desoky, I. M., 2019. A hybridization of sine cosine algorithm with steady state genetic algorithm for engineering design problems. *International Conference on Advanced Machine Learning Technologies and Applications*. Springer, Cham. 143-155.
- Emary, E., Zawbaa, H. M., ve Grosan, C., 2017. Experienced gray wolf optimization through reinforcement learning and neural networks. *IEEE transactions on neural networks and learning systems*, 29(3): 681-694.
- Eroğlu, Y., ve Seçkiner, S. U., 2019. Early fault prediction of a wind turbine using a novel ANN training algorithm based on ant colony optimization. *Journal of Energy Systems*, 3(4):139-147.
- Fan, S. K. S., Zahara, E., 2007. A hybrid simplex search and particle swarm optimization for unconstrained optimization. *European Journal of Operational Research*, 181(2): 527-548.
- Faramarzi, A., Heidarinejad, M., Stephens, B., Mirjalili, S., 2020. Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Systems*, 105190. 191.
- Faris, H., Aljarah, I., Alqatawna, J. F., 2015. Optimizing feedforward neural networks using krill herd algorithm for e-mail spam detection. *2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*. IEEE. 1-5.
- Faris, H., Aljarah, I., Al-Madi, N., ve Mirjalili, S., 2016. Optimizing the learning process of feedforward neural networks using lightning search algorithm. *International Journal on Artificial Intelligence Tools*, 25(06): 1650033.
- Faris, H., Alkasassbeh, M., Rodan, A., 2014. Artificial neural networks for surface ozone prediction: models and analysis. *Polish Journal of Environmental Studies*, 23(2).
- Faris, H., Mafarja, M. M., Heidari, A. A., Aljarah, I., Ala'M, A. Z., Mirjalili, S., Fujita, H., 2018. An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. *Knowledge-Based Systems*, 154, 43-67.
- Faris, H., Mirjalili, S., Aljarah, I., 2019. Automatic selection of hidden neurons and weights in neural networks using grey wolf optimizer based on a hybrid encoding scheme. *International Journal of Machine Learning and Cybernetics*, 10(10): 2901-2920.
- Farmer, J. D., Packard, N. H., & Perelson, A. S., 1986. The immune system, adaptation, and machine learning. *Physica D: Nonlinear Phenomena*, 22(1-3):187-204.
- Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2): 179-188.
- Fliege, J., Svaiter, B. F., 2000. Steepest descent methods for multicriteria optimization *Mathematical Methods of Operations Research*, 51(3): 479-494.
- Fogel L. J., Fogel D. B., and Angeline P. J., 1994. "A preliminary investigation on extending evolutionary programming to include self-adaptation on finite state machines," *Informatica*, vol. 18, 387–398.

- Fogel L. J., Owens A. J., and Walsh M. J., 1966. **Artificial Intelligence Through Simulated Evolution**. New York: Wiley.
- Fogel, D. B., 1994. An introduction to simulated evolutionary optimization. **IEEE transactions on neural networks**, 5(1): 3-14.
- Fogel, D. B., 1995. **Evolutionary Computation: Toward a New Philosophy of Machine Intelligence**. New York: IEEE Press.
- Forrest, S. 1993. Genetic algorithms: Principles of adaptation applied to computation. **Science**, 261, 872–878.
- Forrest, S., 1996. Genetic algorithms, **ACM Computing Surveys (CSUR)**, 28(1): 77-80.
- Gai, W., Qu, C., Liu, J., Zhang, J., 2018. A novel hybrid meta-heuristic algorithm for optimization problems. **Systems Science & Control Engineering**, 6(3): 64-73.
- Gambardella, L. M., Dorigo, M., 1995. Ant-Q: A reinforcement learning approach to the traveling salesman problem. **Machine Learning Proceedings 1995**, 252-260.
- Gandomi, A. H., Alavi, A. H., 2012. Krill herd: a new bio-inspired optimization algorithm, **Communications in nonlinear science and numerical simulation**, 17(12): 4831-4845.
- Geem, Z. W., Kim, J. H., Loganathan, G. V., 2001. A new heuristic optimization algorithm: harmony search. **Simulation**, 76(2): 60-68.
- Ghaleini, E. N., Koopialipour, M., Momenzadeh, M., Sarafraz, M. E., Mohamad, E. T., Gordan, B., 2019. A combination of artificial bee colony and neural network for approximating the safety factor of retaining walls. **Engineering with Computers**, 35(2): 647-658.
- Ghanem, W. A., Jantan, A., 2020. Training a Neural Network for Cyberattack Classification Applications Using Hybridization of an Artificial Bee Colony and Monarch Butterfly Optimization. **Neural Processing Letters**, 51(1): 905-946.
- Ghosh, T., Sengupta, S., Chattopadhyay, M ve Dan, P., 2011. Meta-heuristics in cellular manufacturing: A state-of-the-art review. **International Journal of Industrial Engineering Computations**, 2(1): 87-122.
- Glover, F., 1986. Future paths for integer programming and links to artificial intelligence **Computers operations research**, 13(5): 533-549.
- Glover, F., Laguna, M., 1998. Tabu search. **Handbook of combinatorial Optimization**, Springer, Boston, MA. 2093-2229.
- Goldberg, D. E., 1989., **Genetic Algorithms in Search, Optimization, and Machine Learning**. Addison-Wesley, Reading, MA.
- Gordan, B., 2019. A combination of artificial bee colony and neural network for approximating the safety factor of retaining walls. **Engineering with Computers**, 35(2): 647-658.
- Goss, S., Aron, S., Deneubourg, J. L., Pasteels, J. M., 1989. Self-organized shortcuts in the Argentine ant. **Naturwissenschaften**, 76(12): 579-581.
- Gupta, S., Deep, K., 2019. A novel hybrid sine cosine algorithm for global optimization and its application to train multilayer perceptrons. **Applied Intelligence**, 1-34.
- Gülpınar, V., 2013. **Yapay Sinir Ağları İşleyişinin Sosyal Ağ Analizi Yardımı ile Çözümlemesi** (Doktora tezi), MÜ, Ekonometri Ana Bilim Dalı, İstanbul.
- Hageman, J. A., Streppel, M., Wehrens, R., Buydens, L. M. C., 2003. Wavelength selection with Tabu search. **Journal of Chemometrics: A Journal of the Chemometrics Society**, 17(8-9): 427-437.

- Hamed, M., Salleh, S. H., Astaraki, M., Noor, A. M., Harris, A. R. A., 2014. Comparison of multilayer perceptron and radial basis function neural networks for EMG-based facial gesture recognition. **The 8th International Conference on Robotic, Vision, Signal Processing & Power Applications** Springer, Singapore. 285-294.
- Hamzaçebi, C., 2011. **Yapay sinir ağı**, Ekin Yayınları, Bursa.
- Hao, P., Wang, Z., Wu, G., Boriboonsomsin, K., Barth, M., 2017. Intra-platoon vehicle sequence optimization for eco-cooperative adaptive cruise control. **2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)** IEEE. 1-6.
- Haykin, S., 1999. Self-organizing maps. **Neural Networks-A Comprehensive Foundation, 2nd edition**, Prentice-Hall.
- Haykin, S., 2005. **Neural Networks: A Comprehensive Foundation**, ninth edition Prentice-Hall, Upper Saddle River, NJ., 30-52.
- He, P., Tan, C., Huang, H., 2013. Research on receiver autonomous integrity monitoring algorithm using genetic algorithm resampling particle filter. **China Satellite Navigation Conference (CSNC) 2013 Proceedings**. Springer, Berlin, Heidelberg. 221-231.
- Hecht-Nielsen, R., 1992. Theory of the backpropagation neural network. **Neural networks for perception**, Academic Press. 65-93.
- Heidari, A. A., Abbaspour, R. A., & Jordehi, A. R., 2017. An efficient chaotic water cycle algorithm for optimization tasks. **Neural Computing and Applications**, 28(1), 57-85.
- Heidari, A. A., Faris, H., Aljarah, I., Mirjalili, S., 2019. An efficient hybrid multilayer perceptron neural network with grasshopper optimization. **Soft Computing**, 23(17), 7941-7958.
- Heidari, A. A., Faris, H., Mirjalili, S., Aljarah, I., Mafarja, M., 2020. Ant lion optimizer: theory, literature review, and application in multi-layer perceptron neural networks. In **Nature-Inspired Optimizers**, 23-46. Springer, Cham.
- Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., ve Chen, H., 2019. Harris hawks optimization: Algorithm and applications. **Future generation computer systems**. 97, 849-872.
- Hemeida, A. M., Hassan, S. A., Mohamed, A. A. A., Alkhalaf, S., Mahmoud, M. M., Senjyu, T., El-Din, A. B., 2020. Nature-inspired algorithms for feed-forward neural network classifiers: A survey of one decade of research. **Ain Shams Engineering Journal**.
- Hoey, P. S., 2004. Statistical Analysis of the Iris Flower Dataset. **University of Massachusetts At Lowell, Massachusetts**.
- Holland, J. H., 1975. **Adaptation in Natural and Artificial Systems**. The University of Michigan Press, Ann Arbor, MI.
- Holland, J. H., 1992a. Genetic algorithms. **Scientific american**, 267(1): 66-73.
- Holland, J. H., 1992b. **Adaptation In Natural and Artificial Systems**, MIT Press.
- Hooke R. Jeeves T.A., 1961 "Direct search solution of numerical and statistical problems," **J. Assoc. Comput. Mach.**, vol.8, 212-229.
- Hopfield, J., 1982. **Neural networks and physical systems with emergent collective computational abilities**, 79(8): 2554-2558. Proc. Natl. Acad. Sci. U. S. A.

- Hornik, K., 1991. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2): 251-257.
- Huang, H., Jin, Y., Huang, B., Qiu, H. G., 2014. Mixed replenishment policy for a to supply chain based on hybrid genetic simulated annealing algorithm. *Mathematical Problems in Engineering*.
- Hussain, K., Salleh, M. N. M., Cheng, S., Naseem, R., 2017. Common benchmark functions for metaheuristic evaluation: A review. *JOIV: International Journal on Informatics Visualization*, 1(4-2): 218-223.
- Ilonen, J., Kamarainen, J. K., & Lampinen, J., 2003. Differential evolution training algorithm for feed-forward neural networks. *Neural Processing Letters*, 17(1):93-105.
- Innocente, M. S., 2010. *Development and testing of a Particle Swarm Optimizer to handle hard unconstrained and constrained problems*. Swansea University.
- İlkuçar, M., Işık, A. H., & Çifci, A., 2014. Classification of breast cancer data with harmony search and back propagation based artificial neural network. *2014 22nd Signal Processing and Communications Applications Conference (SIU)*. IEEE.62-765.
- Jaddi, N. S., Abdullah, S., 2020. Global search in single-solution-based metaheuristics. *Data Technologies and Applications*.
- Jain, A. K., Mao, J., Mohiuddin, K. M., 1996. Artificial neural networks: A tutorial. *Computer*, 29(3): 31-44.
- Jain, A. K., Murty, M. N., ve Flynn, P. J., 1999. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3): 264-323.
- James, W., 1890. *The principles of psychology*, vol. 2. ny, us: Henry holt and company.
- Jamil, M., Yang, X. S., 2013. A literature survey of benchmark functions for global optimization problems. *arXiv preprint arXiv:1308.4008*.
- Joines, J. A., Houck, C. R., 1994. On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA's. *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*. IEEE. 579-584.
- Karaboga, D., 2005. An Idea Based on Honey Bee Swarm for Numerical Optimization, *Erciyes University, Engineering Faculty, Computer Engineering Department, Technical report-tr06*.
- Karaboga, D., Akay, B., 2009. A comparative study of artificial bee colony algorithm. *Applied mathematics and computation*, 214(1): 108-132.
- Karaboga, D., Artificial Bee Colony (ABC) Algorithm, <https://abc.erciyes.edu.tr>, Erişim tarihi Şubat 16, 2019.
- Karaboga, D., Basturk, B., 2008. On the performance of artificial bee colony (ABC) algorithm. *Applied soft computing*, 8(1): 687-697.
- Kawam, A. A., Mansour, N., 2012. Metaheuristic optimization algorithms for training artificial neural networks. *Int. J. Comput. Inf. Technol*, 1(2): 156-161.
- Kayri, M., 2015. An intelligent approach to educational data: performance comparison of the multilayer perceptron and the radial basis function artificial neural networks. *Educational Sciences: Theory & Practice*, 15(5).

- Kennedy, J., Eberhart, R. C., 1997. A discrete binary version of the particle swarm algorithm. **1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation**, Vol. 5, IEEE. 4104-4108.
- Kennedy, J., 1997. The particle swarm: Social adaptation of knowledge. **Proc. IEEE Intl. Conf on Neural Networks** (Indianapolis, Indiana), IEEE Service Center, Piscataway, NJ, 303-308.
- Kennedy, J., Eberhart, R. 1995. Particle swarm optimization. **Proceedings of ICNN'95-International Conference on Neural Networks**, Vol. 4, IEEE. 1942-1948.
- Keumarsi V, Simab M, Shahgholian G., 2014. An integrated approach for optimal placement and tuning of power system stabilizer in multi-machine systems. **Int J of Electric Power Energy Syst**, **63**:132-139.
- Khalilpourazari, S., Khalilpourazary, S., 2019. An efficient hybrid algorithm based on Water Cycle and Moth-Flame Optimization algorithms for solving numerical and constrained engineering optimization problems. **Soft Computing**, **23**(5): 1699-1722.
- Khishe, M., & Mosavi, M. R., 2020. Classification of underwater acoustical dataset using neural network trained by Chimp Optimization Algorithm. **Applied Acoustics**, **157**, 107005.
- Khishe, M., Mohammadi, H., 2019. Passive sonar target classification using multi-layer perceptron trained by salp swarm algorithm. **Ocean Engineering**, **181**, 98-108.
- Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., 1983. Optimization by simulated annealing. **science**, **220**(4598): 671-680.
- Kohonen, T. (1993). Physiological interpretation of the self-organizing map algorithm. **Neural Networks**, **6**(7): 895-905.
- Kolhe, M., Lin, T. C., ve Maunuksela, J., 2011. GA-ANN for short-term wind energy prediction. **2011 Asia-Pacific Power and Energy Engineering Conference**, 1-6. IEEE.
- Kowalski, P. A., ve Łukasik, S., 2016. Training neural networks with krill herd algorithm. **Neural Processing Letters**, **44**(1): 5-17.
- Laporte, G., Gendreau, M., Potvin, J. Y., & Semet, F., 2000. Classical and modern heuristics for the vehicle routing problem. **International transactions in operational research**, **7**(4-5): 285-300.
- Lee, K. Y., Yang, F. F. (1998). Optimal reactive power planning using evolutionary algorithms: A comparative study for evolutionary programming, evolutionary strategy, genetic algorithm, and linear programming. **IEEE Transactions on power systems**, **13**(1): 101-108.
- Lee, S., Choeh, J. Y., 2014. Predicting the helpfulness of online reviews using multilayer perceptron neural networks. **Expert Systems with Applications**, **41**(6): 3041-3046.
- Liu, Y., Yao, X., 1996. Evolutionary design of artificial neural networks with different nodes. **Proceedings of IEEE International Conference on Evolutionary Computation**, IEEE. 670-675.
- Marini, F., Bucci, R., Magri, A. L., Magri, A. D., 2008. Artificial neural networks in chemometrics: History, examples and perspectives. **Microchemical journal**, **88**(2): 78-185.

- Matsuda, S.,1998. "Optimal" Hopfield network for combinatorial optimization with linear cost function. *IEEE transactions on neural networks*, **9**(6), 1319-1330.
- McCulloch, W.S., Pitts, W., 1943. A Logical Calculus of Ideas Immanent in Activity, *Bull. Mathematical Bio-physics*, **5**: 115-133.
- McGill, R., Tukey, J. W., Larsen, W. A.,1978. Variations of box plots. *The American Statistician*, **32**(1): 12-16.
- Millonas, M. M.,1994. Swarms, phase transitions, and collective intelligence. *C. G. Langton*, (Ed., Artificial Life III. Addison Wesley), Reading, MA.
- Mingguang, L., Gaoyang, L., 2009. Artificial neural network co-optimization algorithm based on differential evolution. *2009 Second International Symposium on Computational Intelligence and Design* (Vol. 1), IEEE. 256-259.
- Minsky, M., Papert, S., 1969. *Perceptrons: An Introduction to Computational Geometry*, MIT Press, Cambridge, Mass.
- Mirjalili, S., Lewis, A., 2016. The whale optimization algorithm. *Advances in engineering software*, **95**, 51-67.
- Mirjalili, S., 2015a. How effective is the Grey Wolf optimizer in training multi-layer perceptrons. *Applied Intelligence*, **43**(1):150-161.
- Mirjalili, S., 2015b. The ant lion optimizer. *Advances in engineering software*, **83**, 80-98.
- Mirjalili, S., 2016. SCA: a sine cosine algorithm for solving optimization problems. *Knowledge-based systems*, **96**, 120-133.
- Mirjalili, S., 2019. Genetic algorithm, In *Evolutionary Algorithms and Neural Networks*, 43-55, Springer, Cham.
- Mirjalili, S., 2019. Sine Cosine Algorithm (SCA). <http://www.alimirjalili.com/SCA.html>, Erişim tarihi Şubat 18, 2019.
- Mirjalili, S., Dong, J. S.ve Lewis, A., 2020. Ant Colony optimizer: theory, literature review, and application in AUV path planning. *Nature-Inspired Optimizers*, 7-21. Springer, Cham.
- Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., Mirjalili, S. M., 2017. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, **114**, 163-191.
- Mirjalili, S., Hashim, S. Z. M., Sardroudi, H. M., 2012. Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm. *Applied Mathematics and Computation*, **218**(22): 11125-11137.
- Mirjalili, S., Mirjalili, S. M., Lewis, A., 2014. Grey wolf optimizer. *Advances in engineering software*, **69**, 46-61.
- Mohammed, H., Rashid, T., 2020. A novel hybrid GWO with WOA for global numerical optimization and solving pressure vessel design. *Neural Computing and Applications*, 1-18.
- Molga, M., Smutnicki, C., 2005. Test functions for optimization needs. *Test functions for optimization needs*, 101.
- Murphy, K. P., 2012. Machine learning: a probabilistic perspective, *MIT press*.
- Murtagh, F., 1991. Multilayer perceptrons for classification and regression. *Neurocomputing*, **2**:(5-6), 183-197.
- Musirin, I., Radzi, N. D. M., Othman, M. M., Idris, M. K., Rahman, T. K. A., 2008. Voltage profile improvement using unified power flow controller via artificial immune system. *WSEAS Transaction on Power Systems*, **3**(4).

- Nelson, M. M., Illingworth, W. T., 1991. *A practical guide to neural nets*.
- Nordin, P., Keller, R. E., Francone, F. D., 1998. *Genetic programming*. (Editor: W. Banzhaf). Springer.
- O'Neil, M. A., Burtscher, M., 2015. Rethinking the parallelization of random-restart hill climbing: a case study in optimizing a 2-opt TSP solver for GPU execution. *Proceedings of the 8th workshop on general purpose processing using GPUs* 99-108.
- Ojha, VK, Abraham, A, Snášel, V., 2017. Metaheuristic design of feedforward neural networks: a review of two decades of research. *Eng Appl Artif Intell*, **60**: 97–116.
- Okonta, C. I., Kemp, A. H., Edopkia, R. O., Monyei, G. C., Okelue, E. D., 2016. A heuristic based ant colony optimisation algorithm for energy efficient smart homes. *IAEMM*.
- Özdemir, A., 2007. *Melez tek bir işlem birimli yapay sinir ağı hücresinin otomatik öğrenme veri tabanlarına uygulanması* (Master's thesis, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü).
- Passino, K. M., 2002. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE control systems magazine*, **22**(3): 52-67.
- Pasteels, J. M., Deneubourg, J. L., Goss, S., 1987. Self-organization mechanisms in ant societies. I: Trail recruitment to newly discovered food sources. *Experientia. Supplementum*, (54): 155-175.
- Pazzani, M. J., 1991. Influence of prior knowledge on concept acquisition: Experimental and computational results. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, **17**(3): 416–432.
- Picheny, V., Wagner, T., Ginsbourger, D., 2013. A benchmark of kriging-based infill criteria for noisy optimization. *Structural and Multi disciplinary Optimization*, **48**(3): 607-626.
- Polat, K., Sahan, S., Kodaz, H., Günes, S., 2005. A new classification method to diagnosis heart disease: Supervised artificial immune system (AIRS). *Proceedings of the Turkish Symposium on Artificial Intelligence and Neural Networks* (TAINN).
- Porto, V. W., Saravanan, N., Waagen, D., & Eiben, A. E., 1998. Evolutionary Programming VII 7th International Conference, EP98 San Diego, California, USA, March 25–27, 1998 Proceedings. *Conference proceedings EP*. 939.
- Probst, M., Rothlauf, F., 2015. Deep boltzmann machines in estimation of distribution algorithms for combinatorial optimization. *arXiv preprint arXiv:1509.06535*.
- Rao, S. S., Desai, R. C., 1980. Optimization theory and applications. *IEEE Transactions on Systems, Man, and Cybernetics*, **10**(5): 280-280.
- Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S., 2009. GSA: a gravitational search algorithm. *Information sciences*, **179**(13): 2232-2248.
- Rechenberg, I., 1965. Cybernetic solution path of an experimental problem. *Royal Aircraft Establishment Library Translation* 1122.
- Rechenberg, I., 1973. Evolutionsstrategie—*Optimierung technischer Systeme nach Prinzipien der biologischen Information*.
- Resende, M. G., Ribeiro, C. C., 1998. Greedy randomized adaptive search procedures (GRASP). *AT&T Labs Research Technical Report*, **98**(1): 1-11.
- Riehl H., *Introduction to the Atmosphere*, McGraw Hill, 1978.

- Ríos-Uzeda, B., Brigatti, E., & Vieira, M. V., 2019. Lévy like patterns in the small-scale movements of marsupials in an unfamiliar and risky environment. **Scientific reports**, 9(1): 1-12.
- Rosenblatt, R., 1962. **Principles of Neurodynamic.**, Spartan Books, New York.
- Rumelhart DE, Hinton GE, Williams RJ., 1986. Learning internal representation by error propagation. Rumelhart DE, McClelland JL and the PDP Research Group (eds) **Parallel distributed processing**, vol 1, chap 8. MIT Press, Cambridge, Mass.
- Rumelhart, D.E., McClelland J.L., 1986. Parallel Distributed Processing: Exploration in the **Microstructure of Cognition**, MIT Press, Cambridge, Mass.
- Salomon R., 1998. Evolutionary algorithms and gradient search: similarities and differences. **IEEE Transactions on Evolutionary Computation**, 2(2): 45-55.
- Schraudolph, N. N., Yu, J., & Günter, S., 2007. A stochastic quasi-Newton method for online convex optimization. **Artificial intelligence and statistics**. 436-443.
- Schwefel, H.P., 1995. **Evolution and Optimum Seeking**. New York: Wiley.
- Shafie, A. S., Mohtar, I. A., Masrom, S., & Ahmad, N., 2012. Backpropagation neural network with new improved error function and activation function for classification problem. **2012 IEEE Symposium on Humanities, Science and Engineering Research**, 1359-1364. IEEE.
- Shang, Y. W., Qiu, Y. H., 2006. A note on the extended Rosenbrock function. **Evolutionary Computation**, 14(1): 119-126.
- Shayanfar, H., Gharehchopogh, F. S., 2018. Farmland fertility: A new metaheuristic algorithm for solving continuous optimization problems. **Applied Soft Computing**, 71: 728-746.
- Shi, Y., & Eberhart, R. C., 1998. Parameter selection in particle swarm optimization. **International Conference on Evolutionary Programming**, Springer, Berlin. 591-600.
- Simon, D., 2008. Biogeography-based optimization. **IEEE transactions on evolutionary computation**, 12(6): 702-713.
- Singh, N., Chiclana, F., Magnot, J. P., 2020. A new fusion of salp swarm with sine cosine for optimization of non-linear functions. **Engineering with Computers**, 36(1): 185-212.
- Singh, N., ve Hachimi, H., 2018. A new hybrid whale optimizer algorithm with mean strategy of wolf optimizer for global optimization. **Mathematical and Computational Applications**, 23(1): 14.
- Singh, V. K., 2016. Proposing Solution to XOR problem using minimum configuration MLP. **Procedia Computer Science**, 85, 263-270.
- Singh, V. K., ve Pandey, S., 2016. Minimum configuration MLP for solving XOR problem. **In 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)**, IEEE. 174-179.
- Sinha, N., Chakrabarti, R., Chattopadhyay P. K., 2003. "Evolutionary programming techniques for economic load dispatch, **IEEE Transactions on Evolutionary Computation**, vol. 7, no. 1, 83-94.
- Sinha, N., Chakrabarti, R., Chattopadhyay, P. K., 2003. Evolutionary programming techniques for economic load dispatch. **IEEE Transactions on evolutionary computation**, 7(1): 83-94.
- Storn, R., and Price, K.V., 1997. "Differential evolution-A simple and efficient heuristic for global Optimization over continuous Spaces", **J.Global Optim**, 11341-359.

- Suratgar, A. A., Tavakoli, M. B., Hoseinabadi, A., 2005. Modified Levenberg-Marquardt method for neural networks training. *World Acad Sci Eng Technol*, **6**(1): 46-48.
- Swain, M., Dash, S. K., Dash, S., ve Mohapatra, A., 2012. An approach for iris plant classification using neural network. *International Journal on Soft Computing*, **3** (1): 79.
- Şentürk, A., Şentürk, Z. K., 2016. Yapay Sinir Ağları İle Göğüs Kanseri Tahmini. *El-Cezeri Journal of Science and Engineering*, **3**(2).
- Talbi, E. G., 2009. *Metaheuristics: from design to implementation* (Vol. 74). John Wiley & Sons.
- Tan, S. C., Wang, S., ve Watada, J., 2018. A self-adaptive class-imbalance TSK neural network with applications to semiconductor defects detection. *Information sciences*, **427**, 1-17.
- Uryasev, S., Pardalos, P. M., 2013. *Stochastic optimization: algorithms and applications* Springer Science & Business Media. 54.
- Üstün, B., Melssen, W. J., Oudenhuijzen, M., Buydens, L. M. C., 2005. Determination of optimal support vector regression parameters by genetic algorithms and simplex optimization. *Analytica Chimica Acta*, **544**(1-2): 292-305.
- Vaz, A. I. F., Vicente, L. N., 2007. A particle swarm pattern search method for bound constrained global optimization. *Journal of Global Optimization*, **39**(2): 197-219.
- Vishwakarma, M. D. D., 2012. Genetic algorithm based weights optimization of artificial neural network. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, **1**(3): 206-211.
- Wang, C., Wu, F., Shi, Z., ve Zhang, D., 2016. Indoor positioning technique by combining RFID and particle swarm optimization-based back propagation neural network. *Optik*, **127**(17): 6839-6849.
- Wang, D., Tan, D., ve Liu, L. (2018). Particle swarm optimization algorithm: an overview. *Soft Computing*, **22**(2): 387-408.
- Wang, G. G., 2018. Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems. *Memetic Computing*, **10**(2): 151-164.
- Wedderburn, R. W. (1974). Quasi-likelihood functions, generalized linear models, and the Gauss-Newton method. *Biometrika*, **61**(3): 439-447.
- Werbos, P., 1974. "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences" (PhD thesis), Dept. of Applied Mathematics, Harvard University, Cambridge, Mass.
- White, H., 1989. Learning in artificial neural networks: A Statistical perspective. *Neural Computation* **1**, 425-464.
- Xiao L, and Bilmes J., 2006. "Regularized Adaptation of Discriminative Classifiers," *IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, Toulouse, I-I.
- Xie, Y., Liu, M., Feng, T., Xu, Y., 2020. Compact disordered magnetic resonators designed by simulated annealing algorithm. *Nanophotonics*,
- Xinchao, Z., 2011. Simulated annealing algorithm with adaptive neighborhood. *Applied Soft Computing*, **11**(2): 1827-1836.
- Xu, J., Yan, F., 2019. Hybrid Nelder-Mead algorithm and dragonfly algorithm for function optimization and the training of a multilayer perceptron. *Arabian Journal for Science and Engineering*, **44**(4): 3473-3487.

- Yamashita, N., Fukushima, M., 2001. On the rate of convergence of the Levenberg-Marquardt method. In *Topics in numerical analysis*, **Springer**, Vienna. 239-249.
- Yang, X. S. (2009). Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms*. 169-178. Springer, Berlin, Heidelberg.
- Yang, X. S., 2010. "Test Problems in Optimization," **Engineering Optimization: An Introduction with Metaheuristic Applications John Wiley & Sons**.
- Yang, X. S., Deb, S., 2009. Cuckoo search via Lévy flights. *2009 World congress on nature & biologically inspired computing (NaBIC)*. IEEE. 210-214.
- Yao, X., 1993. Evolutionary artificial neural networks. *International journal of neural systems*, **4**(03): 203-222.
- Yao, X. (1999). Evolving artificial neural networks. *Proceedings of the IEEE*, **87**(9): 1423-1447.
- Yarpiz Team, 2019. Particle Swarm Optimization (PSO), <https://www.mathworks.com/matlabcentral/fileexchange/52857-particle-swarm-optimization-psy>, Erişim tarihi Şubat 17, 2019.
- Yuryevich, J., Wong, K. P., 1999. Evolutionary programming based optimal power flow algorithm. *IEEE transactions on Power Systems*, **14**(4): 1245-1250.
- Zhang, G., Patuwo, B.E., Hu, Y.M. 1998. Forecasting with artificial neural networks: *The state of the art. International Journal of Forecasting*, **14**, 35-62.
- Zhang, L., Liu, L., Yang, X. S., Dai, Y., 2016. A novel hybrid firefly algorithm for global optimization. *PloS one*, **11**(9).
- Zhang, X., Wang, X., Chen, H., Wang, D., Fu, Z., 2020. Improved GWO for large-scale function optimization and MLP optimization in cancer identification. *Neural Computing and Applications*, 1-21.
- Zhang, Z., Ding, S., Jia, W., 2019. A hybrid optimization algorithm based on cuckoo search and differential evolution for solving constrained engineering problems. *Engineering Applications of Artificial Intelligence*, **85**, 254-268.
- Zhao, W., 2019. Atom Search Optimization (ASO) Algorithm, <https://www.mathworks.com/matlabcentral/fileexchange/67011-atom-search-optimization-aso-algorithm>, Erişim tarihi Şubat 21, 2019.
- Zhao, W., Wang, L., Zhang, Z., 2019. Atom search optimization and its application to solve a hydrogeologic parameter estimation problem, *Knowledge-Based Systems*, **163**, 283-304.
- Zhou, A. H., Zhu, L. P., Hu, B., Deng, S., Song, Y., Qiu, H., Pan, S., 2019. Traveling-salesman-problem algorithm based on simulated annealing and gene-expression programming. *Information*, **10**(1): 7.
- Zhou, C., Gao, H. B., Gao, L., Zhang, W. G., 2003. Particle Swarm Optimization (PSO) Algorithm [J]. *Application Research of Computers*, **12**, 7-11.
- W. Zhao, 2020. Atom Search Optimization (ASO) Algorithm, <https://www.mathworks.com/matlabcentral/fileexchange/67011-atom-search-optimization-aso-algorithm>, MATLAB Central File Exchange. Retrieved August 5, 2020.

ÖZ GEÇMİŞ

Adı soyadı : Erdal EKER

e-mail (1) : e.eker@alparslan.edu.tr

e-mail (2) : erdaleker4965@gmail.com

1973 yılında Van-Başkale doğumludur. Evli ve 4 çocuk babasıdır. Muş Alparslan Üniversitesi Sosyal Bilimler Meslek Yüksekokulu'nda öğretim görevlisi olarak görev yapmaktadır.

Eğitim

1989-1994: Yüzüncü Yıl Üniversitesi Fen-Edb fak. Matematik bölümü lisans eğitimi

2011-2014: Atatürk üniversitesi Fen fak. Uygulamalı matematik alanında yüksek lisans eğitimi

2016-2020: Yüzüncü Yıl Üniversitesi İstatistik Anabilim Dalı doktora eğitimi

T.C
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
LİSANSÜSTÜ TEZ ORJİNALLİK RAPORU

Tarih: 06/08/2020

Tez Başlığı / Konusu: YAPAY SİNİR AĞLARININ MODERN SEZGİSEL-ÜSTÜALGORİTMALARI
YOLUYLA EĞİTİMİ

Yukarıda başlığı/konusu belirlenen tez çalışmamın Kapak sayfası, Giriş, Ana bölümler ve Sonuç bölümlerinden oluşan toplam 127 sayfalık kısmına ilişkin, 06/08/2020 tarihinde şahsım/tez danışmanım tarafından TURNİTİN intihal tespit programından aşağıda belirtilen filtreleme uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı %8 (Sekiz) dir.

Uygulanan filtreler aşağıda verilmiştir:

- Kabul ve onay sayfası hariç,
- Teşekkür hariç,
- İçindekiler hariç,
- Simge ve kısaltmalar hariç,
- Gereç ve yöntemler hariç,
- Kaynakça hariç,
- Alıntılar hariç,
- Tezden çıkan yayınlar hariç,
- 7 kelimeden daha az örtüşme içeren metin kısımları hariç (Limit inatch size to 7 words)

Van Yüzüncü Yıl Üniversitesi Lisansüstü Tez Orijinallik Raporu Alınması ve Kullanılmasına İlişkin Yönergeyi inceledim ve bu yönergede belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini bilgilerinize arz ederim.

06.08.2020

Adı Soyadı: Erdal EKER

Öğrenci No: 169102056

Anabilim Dalı: İstatistik

Programı:

Statüsü: Y. Lisans ☐

Doktora ☒

DANIŞMAN ONAYI
UYGUNDUR

Prof. Dr. Murat KAYRI

(Unvan, Ad Soyad, İmza)

ENSTİTÜ ONAYI
UYGUNDUR

Prof. Dr. Suat ŞENSOY

(Unvan, Ad Soyad, İmza)