



T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

YAPAY SİNİR AĞLARININ GİRDAP ARAMA ALGORİTMASIYLA EĞİTİLMESİ

Zainab Abdullah Jalil JALIL

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Mayıs-2018
KONYA
Her Hakkı Saklıdır

ÖZET

YÜKSEK LİSANS TEZİ

YAPAY SİNİR AĞLARININ GİRDAP ARAMA ALGORİTMASIYLA EĞİTİLMESİ

Zainab Abdullah Jalil JALIL

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Tahir SAĞ

2018, 96 Sayfa

Jüri
Prof. Dr. Harun UĞUZ
Dr. Öğr. Üyesi Onur İNAN
Dr. Öğr. Üyesi Tahir SAĞ

Yapay Sinir Ağları (YSA) biyolojik beyinin öğrenme ve bilgi saklama yeteneklerinin benzetilmeye çalışıldığı yapay zekâ tekniğidir. YSA'da bilgileri depolayan bağlantı ağırlıkları, nöronları paralel ve ardışık bir biçimde birleştirirler. Oldukça zor bir optimizasyon problemi olan yapay sinir ağlarının öğrenmesi süreci ağ yapısındaki ağırlıklara uygun değerlerin atanması işlemidir. YSA, karmaşık gerçek dünya problemlerini modellemek için birçok alanda yaygın olarak kullanılan bir hesaplama aracıdır. Literatürde birçok çalışmanın yapıldığı YSA alanında ilk çalışmalar matematiksel hata indirgeme tekniklerine dayalı olmakla birlikte son zamanlarda meta-sezgisel optimizasyon algoritmaları yapay sinir ağlarının eğitiminde yaygın olarak kullanılmaya başlanmıştır.

Bu tez çalışmasında, yakın zamanda tek çözüm tabanlı optimizasyon algoritması olarak geliştirilen Girdap Arama (Vortex Search) Algoritması üzerinde durulmakta ve literatürde ilk kez YSA eğitimi için önerilmektedir. Bu algoritmada, karıştırılan sıvılarda meydana gelen girdap deseninden esinlenilmektedir. Girdap Arama Algoritmasının en belirgin özellikleri, yüksek hesaplama hassasiyeti ve hızlı yakınsama kabiliyetidir. Bu avantajları sayesinde diğer optimizasyon algoritmaları ile karşılaştırıldığında oldukça başarılı bir performansa sahip olduğu görülür. En yaygın kullanılan yapay sinir ağı mimarisi ileri beslemeli çok katmanlı algılayıcı (ÇKA) modeli yapay sinir ağlarıdır. Yapılan çalışmada, VS algoritması ÇKA ağlarının eğitimi için çalıştırıldı ve performansı Yapay Arı Kolonisi optimizasyon algoritması (ABC) ve Parçacık Sürü Optimizasyonu (PSO) algoritması ile karşılaştırılarak analiz edildi.

Önerilen yaklaşım, UCI (*machine learning repository of University of California at Irvine*) veri tabanından alınan beş veri seti (Iris Plants Data Set, Wine Recognition Data Set, Breast Cancer Data Set, Thyroid Disease ve Pima Indians Diabetes) üzerinde uygulandı. VS algoritması ileri beslemeli YSA eğitimindeki sınıflandırma performansı değerlendirildi. Elde edilen deneysel sonuçlar PSO ve ABC optimizasyon algoritmaları ile karşılaştırmalı olarak ele alındı. Sonuç olarak, VS algoritmasının performansının en az diğer algoritmalar kadar başarılı olduğu görüldü. VS ile eğitilen YSA modelinin sınıflandırma problemlerinin çözümü için yetenekli alternatif bir araç olarak kullanılabileceği ispatlandı.

Anahtar Kelimeler: Girdap Arama Algoritması, Yapay Sinir Ağı Eğitimi, Sınıflandırma, Çok Katmanlı Algılayıcı Ağları, Optimizasyon.

ABSTRACT

MSc. THESIS

TRAINING ARTIFICIAL NEURAL NETWORKS WITH VORTEX SEARCH ALGORITHM

Zainab Abdullah Jalil JALIL

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF SELÇUK UNIVERSITY
THE DEGREE OF MASTER OF PHILOSOPHY
IN COMPUTER ENGINEERING

Advisor: Assist. Prof. Dr. Tahir SAĞ

2018, 96 Pages

Jury
Prof. Dr. Harun UĞUZ
Assist. Prof. Dr. Onur İNAN
Assist. Prof. Dr. Tahir SAĞ

Artificial Neural Networks (ANN) are artificial intelligence techniques in which the biological brain is attempting to simulate learning and information storage capabilities. In ANN, the connection weights storing information are combined in a parallel and sequential manner. The learning of artificial neural networks is the process of finding optimal values that are appropriate for weights of network topology. ANN is a computational modeling tool that is widely used in many areas to model complex real-world problems. The first studies in the field of ANN where many studies have been done in the literature are based on mathematical error reduction techniques, but recently meta-heuristic optimization algorithms has also been thoroughly used in the training of artificial neural networks.

In this thesis study, Vortex search (VS) algorithm, which has recently developed as a single-solution based optimization algorithm, is proposed for ANN training. Vortex search algorithm is an algorithm that was recommended for the first time in ANN training in the literature. VS algorithm is inspired by the vortex pattern generated by the vortex flow of mixed liquids. The most special features of the Vortex Search Algorithm are high computational accuracy and fast convergence speed. These advantages of VS algorithm make it to be a good performance when compared to other optimization algorithms. The mostly used ANN architecture is the model of multilayer feedforward perceptron (MLP). In the study, VS algorithm was run for the training of multilayer perceptron networks and its performance of presented training approach was analyzed by comparing with artificial bee colony algorithm (ABC) and particle swarm optimization (PSO) algorithms.

The proposed approach was applied on five data sets (Iris Plants Data Set, Wine Recognition Data Set, Breast Cancer Data Set, Thyroid Disease and Pima Indians Diabetes) obtained from the UCI (*machine learning repository of University of California at Irvine*) database. VS algorithm was evaluated by classification performance in the forward-feed ANN training. Experimental results obtained were compared with PSO and ABC optimization algorithms. As a result, it is seen that VS algorithm's performance is at least as good as other algorithms. It has been demonstrated that the VS algorithm trained ANN model can be used as an alternative tool capable of solving classification problems. In comparison of the results, the proposed training strategy based on VS algorithm has achieved the best outcomes for ANN.

Keywords: Vortex Search algorithm, Artificial Neural Network Training, Classification, Multilayer Feedforward Networks, Optimization.

ÖNSÖZ

Yüksek lisans tezi olan bu çalışmada, “Yapay Sinir Ağlarının Girdap Arama Algoritmasıyla Eğitilmesi” araştırılmıştır. Tez çalışması sürecinde değerli katkılarını ve desteklerini esirgemeyen değerli Danışman Hocam Dr. Öğr. Üyesi Tahir SAĞ’a, her an benimle olan ve destekleyen aileme, değerli gurbetlik kardeşlerim İnes Cemil, Rüya İsam’a ve tüm sevdiklerime sonsuz teşekkürlerimi sunarım.

Zainab Abdullah Jalil JALIL
KONYA-2018

İÇİNDEKİLER

ÖZET ..... **iv**

ABSTRACT ..... **v**

ÖNSÖZ ..... **vi**

İÇİNDEKİLER ..... **vii**

ŞEKİL LİSTESİ ..... **ix**

TABLOLAR LİSTESİ ..... **xiii**

SİMGELER VE KISALTMALAR ..... **xv**

1. GİRİŞ **1**

1.1. Çalışmanın Amacı ve Yöntemi 3

1.2. Kaynak Araştırması 4

1.3. Tezin Organizasyonu 9

2. YAPAY SİNİR AĞLARI **10**

2.1. Yapay Sinir Hücresi 11

2.2. Yapay Sinir Ağlarının Eğitilmesi 13

2.3. Eğitim Algoritmaları 15

2.3.1. Klasik Algoritmalar 15

2.4. Parçacık Sürü Optimizasyonu 19

2.5. Yapay Arı Kolonisi Optimizasyonu 23

2.6. Bölüm Değerlendirmesi 26

3. GİRDAP ARAMA ALGORİTMASI (VS: Vortex Search) **27**

3.1. Algoritmanın Yapısı 27

3.2. Bölüm Değerlendirmesi 33

4. OPTİMİZASYON ALGORİTMALARI İLE YSA EĞİTİMİ **34**

4.1. Sınıflandırma Problemleri 35

4.1.1. Diyabet Problemi (Pima Indians Diabetes) 36

4.1.2. Kanser Problemi (Breast Cancer Data Set) 37

4.1.3.Tiroit Problemi (Thyroid Disease) 38
4.1.4. Şarap Problemi (Wine Recognition Data Set) 40
4.1.5. İris Problemi (Iris Plants Data Set) 41
4.2. ÇKA Ağlarının Eğitiminde Yapılan Ayarlamalar 41
4.2.1. Birinci Aşama Performans Değerlendirmesi 44
4.2.2. İkinci Aşama Performans Değerlendirmesi 53
4.2.3. Üçüncü Aşama Performans Değerlendirmesi 65
4.3. Bölüm Değerlendirmesi 77
5. ÇKA AĞLARININ VS İLE EĞİTİLMESİNİ ETKİLEYEN FAKTÖRLERİN İNCELENMESİ 78
5.1. Popülasyon Sayısı ve Parametre Aralık Değerlerinin Etkisi 79
5.2. ÇKA Ağlarının VS Algoritması ile Eğitilmesini Etkileyen Faktörler 80
5.3. Bölüm Değerlendirmesi 85
6. SONUÇLAR VE ÖNERİLER 86
KAYNAKLAR 88
ÖZGEÇMİŞ 96

ŞEKİL LİSTESİ

Şekil 2.1. Biyolojik nöron (Gupta ve ark., 2016) 10

Şekil 2.2. Yapay Bir Sinir Hücresinin Temel Bileşenleri 12

Şekil 2.3. YSA çalışma prensibi 12

Şekil 2.4. Geriye yayım algoritmasının sözde kodu. 16

Şekil 2.5. Levenberg Marquardt algoritmasının sözde kodu 19

Şekil 2.6. PSO algoritmasının sözde kodu 21

Şekil 2.7. (a) 2007 yılına kadar Canonical ve Standard PSO algoritmalarının geometrik yorumu. (b) Standart PSO 2011 (PSO-2011) algoritmasının geometrik yorumu (Zambrano-Bigiarini ve ark., 2013). 22

Şekil 2.8. PSO(2011) algoritmasının sözde kodu 23

Şekil 2.9. ABC algoritmasının sözde-kodu 25

Şekil 3.1. Girdap çemberinin başlangıç merkezi. 27

Şekil 3.2. VS algoritmasının arama sürecinin örnek bir çizimi (Dogan ve Olmez, 2015) .. 29

Şekil 3.3. VS Algoritmanın Arama İşleminden Sonra Girdap Benzeri Bir Yapı 29

Şekil 3.4. VS algoritmasının sözde-kodu 30

Şekil 3.5. (1/x) gammaincinv(x,a) burada x = 0.1 ve a Є [0,1]. 31

Şekil 3.6. (1/0.1) gammaincinv(0.1,a) for (a) MaxItr = 100 (b) MaxItr = 10,000. (Dogan ve Olmez, 2015) 31

Şekil 3.7. Aramanın çözünürlüğü adım boyutunda bir azalma ile artar (Dogan ve Olmez, 2015) 32

Şekil 3.8. (1/x) gammaincinv(x,a) farklı x değerleri için (basamak boyutu = 0.0001) (Dogan ve Olmez, 2015) 33

Şekil 4.1. Diyabet probleminin (a) ÇKA ağının tek çıkışlı topolojisi (b) ÇKA ağının çok çıkışlı topolojisi. 37

Şekil 4.2. Kanser probleminin (a) ÇKA ağının tek çıkışlı topolojisi (b) ÇKA ağının çok çıkışlı topolojisi 38

Şekil 4.3. Tiroit probleminin (a) ÇKA ağının tek çıkışlı topolojisi (b) ÇKA ağının çok çıkışlı topolojisi 39

Şekil 4.4. Şarap probleminin (a) ÇKA ağının tek çıkışlı topolojisi (b) ÇKA ağının çok çıkışlı topolojisi 40

Şekil 4.5. İris probleminin (a) ÇKA ağının tek çıkışlı topolojisi (b) ÇKA ağının çok çıkışlı topolojisi 41

Şekil 4.6. Diyabet Veri Seti için Çok-Çıkışlı ÇKA Ağlarının Birinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri 46

Şekil 4.7. Kanser Veri Seti için Çok-Çıkışlı ÇKA Ağlarının Birinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri 46

Şekil 4.8. Tiroit Veri Seti için Çok-Çıkışlı ÇKA Ağlarının Birinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri 47

Şekil 4.9. Şarap Veri Seti için Çok-Çıkışlı ÇKA Ağlarının Birinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri 48

Şekil 4.10. İris Veri Seti için Çok-Çıkışlı ÇKA Ağlarının Birinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri 48

Şekil 4.11. Diyabet Veri Seti için Tek-Çıkışlı ÇKA Ağlarının Birinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri 50

Şekil 4.12. Kanser Veri Seti için Tek-Çıkışlı ÇKA Ağlarının Birinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri 50

Şekil 4.13. Tiroit Veri Seti için Tek-Çıkışlı ÇKA Ağlarının Birinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri 51

Şekil 4.14. Şarap Veri Seti için Tek-Çıkışlı ÇKA Ağlarının Birinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri 51

Şekil 4.15. İris Veri Seti için Tek-Çıkışlı ÇKA Ağlarının Birinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri 52

Şekil 4.16. Diyabet Veri Seti için Çok-Çıkışlı ÇKA Ağlarının İkinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri 56

Şekil 4.17. Kanser Veri Seti için Çok-Çıkışlı ÇKA Ağlarının İkinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri 57

Şekil 4.18. Tiroit Veri Seti için Çok-Çıkışlı ÇKA Ağlarının İkinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri..... 58

Şekil 4.19. Şarap Veri Seti için Çok-Çıkışlı ÇKA Ağlarının İkinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri..... 58

Şekil 4.20. İris Veri Seti için Çok çıkışlı ÇKA Ağlarının İkinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri..... 60

Şekil 4.21. Diyabet Veri Seti için Tek-Çıkışlı ÇKA Ağlarının İkinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri..... 61

Şekil 4.22. Kanser Veri Seti için Tek-Çıkışlı ÇKA Ağlarının İkinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri..... 62

Şekil 4.23. Tiroit Veri Seti için Tek-Çıkışlı ÇKA Ağlarının İkinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri..... 63

Şekil 4.24. Şarap Veri Seti için Tek-Çıkışlı ÇKA Ağlarının İkinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri..... 64

Şekil 4.25. İris Veri Seti için Tek-Çıkışlı ÇKA Ağlarının İkinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri..... 64

Şekil 4.26. K katlamalı çapraz doğrulamasının akış şeması (Jiang ve Chen, 2016) 66

Şekil 4.27. Diyabet Veri Seti için Çok-Çıkışlı ÇKA Ağlarının Üçüncü Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri..... 70

Şekil 4.28. Kanser Veri Seti için Çok-Çıkışlı ÇKA Ağlarının Üçüncü Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri..... 70

Şekil 4.29. Tiroit Veri Seti için Çok-Çıkışlı ÇKA Ağlarının Üçüncü Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri..... 70

Şekil 4.30. Şarap Veri seti için Çok-Çıkışlı ÇKA ağlarının Üçüncü Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri..... 70

Şekil 4.31. İris Veri Seti için Çok-Çıkışlı ÇKA Ağlarının Üçüncü Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri..... 71

Şekil 4.32. Diyabet Veri Seti için Tek-Çıkışlı ÇKA Ağlarının Üçüncü Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri..... 74

Şekil 4.33. Kanser Veri Seti için Tek-Çıkışlı ÇKA Ağlarının Üçüncü Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri 74

Şekil 4.34. Tiroit Veri Seti için Tek-Çıkışlı ÇKA Ağlarının Üçüncü Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri 75

Şekil 4.35. Şarap Veri Seti için Tek-Çıkışlı ÇKA Ağlarının Üçüncü Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri 75

Şekil 4.36. İris Veri Seti için Tek-Çıkışlı ÇKA Ağlarının Üçüncü Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri 76

Şekil 5.1. Diyabet Veri Seti için Çok-Çıkışlı ÇKA ağlarının eğitilmesinde VS algoritmasının Ortalama Eğitim Hata yakınsama grafiği 81

Şekil 5.2. Kanser Veri Seti için Çok-Çıkışlı ÇKA ağlarının eğitilmesinde VS algoritmasının Ortalama Eğitim Hata yakınsama grafiği 81

Şekil 5.3. Tiroit Veri Seti için Çok-Çıkışlı ÇKA ağlarının eğitilmesinde VS algoritmasının Ortalama Eğitim Hata yakınsama grafiği 81

Şekil 5.4. Şarap Veri Seti için Çok-Çıkışlı ÇKA ağlarının eğitilmesinde VS algoritmasının Ortalama Eğitim Hata yakınsama grafiği 82

Şekil 5.5. İris Veri Seti Çok çıkışlı ÇKA ağlarının eğitilmesinde VS algoritmasının Ortalama Eğitim Hata yakınsama grafiği 82

Şekil 5.6. Diyabet Veri Seti Tek çıkışlı ÇKA ağlarının eğitilmesinde VS algoritmasının Ortalama Eğitim Hata yakınsama grafiği 82

Şekil 5.7. Kanser Veri Seti Tek çıkışlı ÇKA ağlarının eğitilmesinde VS algoritmasının Ortalama Eğitim Hata yakınsama grafiği 83

Şekil 5.8. Tiroit Veri Seti Tek çıkışlı ÇKA ağlarının eğitilmesinde VS algoritmasının Ortalama Eğitim Hata yakınsama grafiği 83

Şekil 5.9. Şarap Veri Seti Tek çıkışlı ÇKA ağlarının eğitilmesinde VS algoritmasının Ortalama Eğitim Hata yakınsama grafiği 83

Şekil 5.10. İris Veri Seti Tek çıkışlı ÇKA ağlarının eğitilmesinde VS algoritmasının Ortalama Eğitim Hata yakınsama grafiği 84

Şekil 5.11. Programın MATLAB ara yüzü 84

TABLOLAR LİSTESİ

Tablo 4.1. Sınıflandırma Problemleri için Çok-Çıkışlı ÇKA Ağında Kullanılan Parametreler 42

Tablo 4.2. Sınıflandırma Problemleri için Tek-Çıkışlı ÇKA Ağında Kullanılan Parametreler 42

Tablo 4.3. Sınıflandırma Problemlerine Ait Deneysel Başarılar 44

Tablo 4.4. Birinci Aşama Eğitim Süreci Sonunda 30 Bağımsız Çalıştırma için Çok-Çıkışlı ÇKA Ağlarında Algoritmaların Elde Ettiği MSE Değerleri 45

Tablo 4.5. Birinci Aşama Eğitim Süreci Sonunda 30 Bağımsız Çalıştırma için Tek-Çıkışlı ÇKA Ağlarında Algoritmaların Elde Ettiği MSE Değerleri 49

Tablo 4.6. İkinci Aşama Eğitim Süreci Sonunda 30 Bağımsız Çalıştırma için Çok-Çıkışlı ÇKA Ağlarında Algoritmaların Elde Ettiği MSE Değerleri 54

Tablo 4.7. İkinci Aşama Test İşlemi Sonunda 30 Bağımsız Çalıştırma için Çok-Çıkışlı ÇKA Ağlarında Algoritmaların Elde Ettiği Yüzde Olarak Sınıflandırma Doğrulukları 55

Tablo 4.8. İkinci Aşama Eğitim Süreci Sonunda 30 Bağımsız Çalıştırma için Tek çıkışlı ÇKA Ağlarında Algoritmaların Elde Ettiği MSE Değerleri 60

Tablo 4.9. İkinci Aşama Test İşlemi Sonunda 30 Bağımsız Çalıştırma için Tek-Çıkışlı ÇKA Ağlarında Algoritmaların Elde Ettiği Yüzde Olarak Sınıflandırma Doğrulukları 61

Tablo 4.10. K-Katlamalı-Çapraz-Doğrulama Yöntemi ile Eğitilen Çok-Çıkışlı ÇKA Ağlarının Sınıflandırma Doğruluk Değerleri 67

Tablo 4.11. Üçüncü Aşama Eğitim Süreci Sonunda Çapraz doğrulama yöntemi Kullanılarak yapılan 30 Bağımsız Çalıştırma için Çok-Çıkışlı ÇKA Ağlarında Algoritmaların Elde Ettiği MSE Değerleri 68

Tablo 4.12. K-Katlamalı-Çapraz-Doğrulama Yöntemi ile Eğitilen Tek-Çıkışlı ÇKA Ağlarının Sınıflandırma Doğruluk Değerleri 72

Tablo 4.13. Üçüncü Aşama Eğitim Süreci Sonunda Çapraz doğrulama yöntemi Kullanılarak yapılan 30 Bağımsız Çalıştırma için Tek-Çıkışlı ÇKA Ağlarında Algoritmaların Elde Ettiği MSE Değerleri 73

Tablo 5.1. Iterasyon Sayısının ÇKA Ağlarının VS Algoritması İle Eğitilmesinin Etkisi ... 78

Tablo 5.2. Parametre Aralığı [-10,10], Popülasyon Büyüklüğü 20 Seçilmesi Sonucu MSE Değerleri 80

Tablo 5.3. Parametre Aralığı [-50,50], Popülasyon Sayısı 60 Seçilmesi Sonucu MSE Değerleri 80

SİMGELER VE KISALTMALAR

Simgeler

b	:	Bias (önyargı)
∂	:	Yeni çözümünün bir parametresi için hesaplanan değişim miktarı (Delta)
$\mathcal{J}$	:	Jacobian matrisi
$pbest$	:	PSO algoritmasının bir parçacık için bulunan en iyi çözüm
$gbest$	:	PSO algoritmasının tüm parçacıklar için bulunan en iyi çözüm
c_1 ve c_2	:	PSO algoritmasında öğrenme katsayıları
v_i ve x_i	:	Yeni üretilen aday çözümleri
x_j^{min}	:	j. karar değişkenin alt sınır değeri
x_j^{max}	:	j. karar değişkenin üst sınır değeri
SN	:	ABC algoritmasında besin kaynaklarının sayısı
D	:	Karar değişkenlerinin sayısı
fit_i	:	ABC algoritmasında kolonideki i.çözümün uygunluk değeri
p_i	:	i.çözümün uygunluk değerine göre hesaplanan seçilme olasılığı
μ	:	VS algoritmasında örnek ortalama merkezi
σ	:	Standart sapma dağılım varyansı
$\sum$	:	Kovaryans matrisi
$rand$	:	Düzgün dağıtılan rasgele sayıdır
Γ	:	Gama fonksiyonu
$MaxItr$	:	Yinelemelerin maksimum sayısı
d_k	:	İstenen çıktı değeri
o_k	:	Üretilen çıktı değeri
$\vec{\omega}(t)$	:	t. İterasyondaki bağlantı ağırlıkları
P	:	Çıktı nöron sayısı
N	:	Örnek sayı

Kısaltmalar

VS	:	Girdap Arama
MVS	:	Modifiyeli Girdap Arama
MVSA	:	Çoklu Girdap Arama Algoritması
MOVS	:	Çok Amaçlı Girdap Arama Algoritması
ABC	:	Yapay Arı Kolonisi
PSO	:	Parçacık Sürü Optimizasyonu
YZ	:	Yapay Zekâ
SA	:	Sinir Ağı
YSA	:	Yapay Sinir Ağı
PS	:	Desen Arama
ÇKA	:	Çok Katmanlı İleri Beslemeli Algılayıcı
BP	:	Geri Yayılım
BPNN	:	Geri Yayılım Sinir Ağı
GA	:	Genetik Algoritması
LM	:	Levenberg-Marquardt
FA	:	Ateş Böceği Algoritması
BBO	:	Biyocoğrafya Tabanlı Optimizasyon
GWO	:	Gri Kurt Optimizasyonu
ALO	:	Karınca Aslan Optimizasyonu
BOA	:	Kelebek Optimizasyonu
WOA	:	Balina Optimizasyon Algoritması
UCI	:	Machine Learning Repository Of University Of California At Irvine
FNN	:	İleri Beslemeli Sinir Ağı
CG	:	Eşlenik Gardiyan

RBF	:	Radiyal Temel Fonksiyon Ağı
RVFL	:	Rasgele Vektör Fonksiyonel Bağlantılı Sinir Ağı
CNN	:	Konvolüsyonel Sinir Ağı
JNN	:	Jacobian Sinir Ağı
FWD	:	Havai Fişek Algoritması
MBN	:	Çok Katmanlı Önyükleme Ağı
CDOA	:	Toplu Karar Optimizasyon Algoritması
APNR	:	Otomatik Plaka Numara Tanıyıcı
EBP	:	Hata Geri Yayılım
Gama	:	gama fonksiyonu
ILS	:	Yenilenen Yerel Arama Algoritması
MSE	:	Ortalama Kare Hata Oranı
IRQ	:	Çeyrekler Arası Aralık
SD	:	Standart Sapma

1. GİRİŞ

Yapay zekâ (YZ); insan zihnine özgü akıl yürütme, anlam çıkartma, genelleme ve geçmiş deneyimlerden öğrenme gibi zihinsel süreçlere ilişkin yetenekleri bir bilgisayar ya da bilgisayar denetimli makineye sezgisel ve meta-sezgisel programlama temelinde geliştirilen yazılımlar aracılığıyla kazandırılmasını hedefleyen çok geniş kapsamlı bir çalışma alanıdır. Yapay zekâ teknikleri; akıllı araçlarda, uzman sistem olarak otomatik kontrol sistemlerinde, doğal dil işleme tekniklerinin kullanıldığı ses ve yazı tanıma uygulamalarında, örüntü tanıma, optimizasyon, bulanık mantık tabanlı akıllı ev cihazları gibi pek çok alanda başarıyla uygulanmaktadır. Bu teknikler günlük yaşantının verimliliğini, emniyetini veya sağlığını geliştirilmeyi hedefler (Mata ve ark., 2018). Bununla birlikte; yapay zekâ algoritmaları, karşılaşılan sorun deterministtik yöntemlerle çözülemediğinde ve/veya sorunun çözümü için matematiksel bir model oluşturulabilmesine rağmen makul bir süre içinde bu model ile çözülemediğinde tercih edilebilecek tekniklerdir. Diğer bir ifadeyle nondeterministik ve NP-Zor biçiminde tanımlanan problemlerin çözümü için önerilen alternatif araçlardır.

Yapay zekânın bir dalı olan yapay sinir ağlarının (YSA) çalışma mekanizması tasarlanırken; hızlı bir şekilde karar vermeyi kolaylaştırmak için paralel çalışan çok sayıda sinir hücresinden oluşan canlıların merkezi sinir sisteminden esinlenilmiştir. Bu sistem, bilim ve mühendislik alanlarında çeşitli sorunları çözmek için heyecan verici alternatif bir yöntem sağlamıştır (Vaferi ve ark., 2015). YSA, insan nöron yapısının giriş / çıkış birlikteliğini taklit etmek için tasarlanmış analitik modellerdir. YSA, güçlü örüntü tanıyıcılar ve sınıflandırıcılar ile matematiksel ve geleneksel yöntemlerle modellenmesi çok karmaşık olan problemler için özellikle uygundur. Yapay sinir ağları, matematiksel olarak açık formüllerin bulunmadığı durumlarda bile giriş/ çıkış fonksiyonellikleri bilinen ilişkileri modelleme yeteneğine sahiptir; Ağın girdi / çıktı alanını uygun bir şekilde temsil edene kadar bilinen giriş / çıkış verileri üzerindeki ağları eğiterek elde edilir (Sitton ve ark., 2017).

Optimizasyon işlemi; mühendislik problemlerinde hesaplamanın önemli bir parçasıdır. Akıllı şebekeler (Rios-Mercado ve Borraz-Sanchez, 2015), lojistik (Wang ve ark., 2015), sensör ağları (Lanza-Gutierrez ve Gomez-Pulido, 2015) gibi farklı alanlarda uygulanmıştır. Bu tür mühendislik problemlerinin, çözümü için doğrusal optimizasyon teknikleri bulunmaktadır (Back ve Schwefel, 1993). Diğer türdeki problemler de doğrusal

olmayan optimizasyon problemleridir ve çözümleri çok daha zordur. Doğrusal kısıtlamalar ve objektif fonksiyonlarla çalışırken problemlerin çözülmesi nispeten kolaydır. Çünkü en iyi çözümü elde etmek için kesin sonuç veren matematiksel yöntemler vardır. Bununla birlikte, doğrusal olmayan kısıtlamalar veya objektif fonksiyonlar söz konusu olduğunda, optimal bir çözüm elde etmek için sezgisel yöntemler kullanmak gerekebilir. Bu sebeplerle, bilim insanları doğrusal olmayan kısıtlamalar ile çalışırken veya model uydururken güçlü alternatif optimizasyon tekniklerine ihtiyaç duymuşlardır (Villarrubia ve ark., 2018).

Son on yılda hem tek çözüm tabanlı hem de popülasyon tabanlı meta-sezgisel optimizasyon algoritması, birçok gerçek dünya problemine başarıyla uygulanmıştır. Bu algoritmalar, arama alanını etkili bir şekilde keşfederken arama etkinliğini azaltarak çözüm üretir. Böylece, belirli bir optimizasyon problemi üzerinde meta-sezgisel metodun başarısı, keşif ve sömürü arasında iyi bir denge kurma becerisi ile tanımlanır. Meta-sezgisel optimizasyon algoritmaları bilimsel topluluklarda özellikle birçok karmaşık optimizasyon probleminin çözümü için önemli gelişmelerle dikkatleri üzerine çekmiştir. Optimizasyon problemleri için birçok algoritma önerilmiştir. Genel amaçlı sezgisel optimizasyon algoritmaları, biyoloji tabanlı, sürü tabanlı, fizik tabanlı, müzik tabanlı, sosyal tabanlı ve kimya tabanlı olmak üzere farklı gruplarda değerlendirilmektedir. Bu algoritmalar basitlik, esneklik, sağlamlık ve verimlilik nedeniyle çeşitli alanlara uygulanabilir (Kaur ve Arora, 2018). Şimdiye kadar geliştirilen ve doğadan ilham alan bazı meta-sezgisel algoritmalar, Parçacık Sürü Optimizasyonu (PSO) (Kennedy ve Eberhart, 1995), Yapay Arı Kolonisi (ABC) (Karaboga, 2005), Ateşböceği Algoritması (FA) (Yang, 2009), Biyocoğrafya Tabanlı Optimizasyon algoritması (BBO) (Simon, 2008), Yarasa Algoritması (Yang, 2010), Kedi Sürüsü Optimizasyonu (Chu ve ark., 2006), Gri Kurt Optimizasyonu (GWO) (Mirjalili ve ark., 2014), Karınca aslanı Optimizasyonu (ALO) (Mirjalili, 2015), Kelebek Optimizasyon Algoritması (BOA) (Arora ve Singh, 2015) ve Balina Optimizasyon Algoritması (WOA) (Mirjalili ve Lewis, 2016) şeklinde sıralanabilir.

Yapay sinir ağlarında optimum ağırlık setinin bulunması algoritmanın performansını etkileyen en önemli faktördür. Literatürde başarılı yaklaşımların bulunmasına rağmen, topolojideki ağırlıkların güncellemesi işlemi halen iyileştirmelerin yapılabileceği bir araştırma alanıdır. Bu tez kapsamında yapay sinir ağlarının doğrusal olmayan karakterdeki ağırlık setinin en uygun şekilde hesaplanmasında girdap arama (VS) algoritmasının kullanılması ve algoritma performansının araştırılması hedeflenmiştir.

Berat Doğan ve Tamer Ölmez tarafından 2015 yılında önerilen VS algoritması; birçok gerçek dünya problemi gibi zor optimizasyon problemlerinin çözümü için geliştirilmiştir. Girdap Arama (VS) algoritması tek çözüm tabanlı meta-sezgisel bir algoritmadır. Bu bağlamda; rasgele arama (RS) ve desen arama (PS) algoritmalarını içeren arama algoritmaları ailesinde incelenebilir. Belirli bir optimizasyon problemi üzerinde meta-sezgisel bir metodun başarısı, keşif ve faydalanma arasında iyi bir denge kurma becerisi ile tanımlanır. VS algoritmasının arama davranışı girdap modelinden esinlendiğinden, yeni önerilen algoritmayı "Vorteks Search" algoritması olarak adlandırıldı. VS algoritmasının en önemli avantajı kullanıcı-tanımlı parametre içermeyen bir algoritma olmasıdır.

VS algoritması 50 adet matematiksel karşılaştırma fonksiyonu üzerinde test edilmiş ve sonuçlar hem tek çözüm tabanlı (Benzetimli tavlama, SA ve Desen Arama, PS) hem de popülasyon tabanlı (Parçacık sürü Optimizasyonu, PSO2011 ve Yapay Arı Kolonisi, ABC) algoritmalarla karşılaştırılmıştır. Sonuç olarak, VS algoritmasının diğer algoritmaların performansıyla karşılaştırıldığında oldukça rekabetçi olduğu ortaya koyulmuştur. VS algoritması oldukça basit olduğu için popülasyona dayalı meta-sezgisellere kıyasla, hesaplama açısından da verimli bir yöntemdir. VS algoritmasının bu avantajları, onu gerçek hayatta optimizasyon problemlerinin çözümü için alternatif iyi bir araç yapmaktadır (Dogan ve Olmez, 2015).

1.1.Çalışmanın Amacı ve Yöntemi

Tez çalışmasının temel amacı, daha önce literatürde yapay sinir ağlarının eğitimi için önerilmemiş bir algoritma olan Girdap Arama (*Vortex Search*) optimizasyon algoritmasını yapay sinir ağlarının eğitiminde kullanılması ve algoritma performansının araştırılması hedeflenmiştir. Girdap Arama Algoritması, 2015 yılında tek-çözüm temelli geliştirilmiş sezgisel bir optimizasyon tekniğidir. VS Algoritması; ayarlanması gereken kullanıcı tanımlı parametre sayısının az olması, hesaplama karmaşıklığının düşük olması ve Gauss dağılımına dayanan basit ama etkili bir arama yöntemi içermesi bakımından dikkat çekici bir özelliğe sahiptir. Literatürde kullanılan test fonksiyonları üzerinde aldığı başarılı sonuçlar da bu özelliğini doğrulamaktadır.

Bu tez çalışmasında; girdap arama algoritması, çok-katmanlı-algılayıcı ağlarının eğitilmesi için adapte edilmiştir. Alternatif bir eğitim yöntemi olarak geliştirilmiş ve

performansı Yapay Arı Kolonisi (ABC) ve Parçacık Sürü Optimizasyonu (PSO) gibi diğer meta-sezgisel tekniklerin performanslarıyla karşılaştırılmıştır. Bu noktada; test amacıyla kullanılacak veri kümeleri UCI veri tabanından (*Machine Learning Repository Of University Of California At Irvine*) alınmıştır. En yaygın kullanılan yapay sinir ağ mimarisi ileri beslemeli çok katmanlı algılayıcı (ÇKA) yapay sinir ağı modelleridir. Girdap arama algoritmasının etkinliğinin analiz edilebilmesi için UCI makine öğrenme deposundan alınan beş farklı test sınıflandırma problemi üzerinde deneysel çalışmalar ve analizler gerçekleştirilmiştir.

1.2. Kaynak Araştırması

Yapay sinir ağları örneklerden öğrenebilme ve çevreden gelen olaylara karşı tepkiler üretebilme yetenekleri ile esnek ve güçlü yapıları sayesinde makina öğrenme, sınıflandırma, genelleme ve optimizasyon gibi alanlarda başarıyla kullanılmaktadırlar. Bununla birlikte, çok çeşitli yapay sinir ağı mimarileri geliştirilmiş ve bu modeller çeşitli alanlarda uygulanmıştır. Bunların dışında genetik algoritmalar ve evrimsel hesaplamalar; içinde karmaşık optimizasyon problemleri barındıran yapay sinir ağı eğitimi için önerilmiş efektif araçlardır. Bunun yanı sıra son çalışmalar göstermektedir ki; sürü zekâsına dayanan Yapay Arı Kolonisi Optimizasyonu (ABC) ve Parçacık Sürü Optimizasyonu (PSO) gibi algoritmalar, yapay sinir ağı probleminin çözümünde kullanılabilen paralel işlem yeteneğine sahip faydalı tekniklerdir. Bu çalışmaların da birçoğu yapay sinir ağı eğitimi üzerinde denenmiştir. Tezin literatüre katkısı bakımından dikkat çekici olan bir diğer nokta ise girdap arama algoritması ile yapılmış çok az sayıda çalışma bulunmasıdır.

(Rumelhart ve ark., 1986), Geri yayılım (BP), ileri beslemeli sinir ağı (FNN) optimizasyonu için birinci dereceden gardiyan-iniş algoritmasını önermişlerdir. Öğrenme oranı çok düşükse, öğrenme yavaşlayacaktır. Eğer öğrenme oranı çok büyükse, öğrenme zikzak olacak ve algoritma istenilen derecede başarılı olamayacaktır. Buna ek olarak, yüksek bir momentum faktörü minimuma aşmak için yüksek bir risk oluşturur ve düşük bir momentum faktörü yerel minimumdan kaçınabilir, ancak öğrenme yavaş olacaktır. Klasik BP algoritması yavaştır ve yerel minimale düşme eğilimindedir.

(Charalambous, 1992), BP'nin aksine, eşlenik gardiyan (CG) adı verilen ikinci dereceden minimizasyon yöntemi, ağırlık optimizasyonu için kullanılmıştır. CG gardiyan ile ilerlemez. Bunun yerine önceki adımla aynı yönde hareket eder. Başka bir deyişle,

geçerli adıma karşılık gelen gardiyan, daha önceki tüm adımların yönüne dik kalır. Her adım en azından bir önceki adıma benzer şekilde iyi olur. Bu tür adımlara müdahale edilmez. Bu nedenle, bir adımda gerçekleştirilen küçültme sonraki adımlar tarafından geri alınmaz.

(Poggio ve Girosi, 1990), sinir ağı topluluğuna dikkat çeken radyal temel fonksiyon ağını (RBF) önermişlerdir. RBF, verimliliği ve eğitim kolaylığı nedeniyle popüler olmuştur. RBF ağı, standart çok katmanlı ileri beslemeli sinir ağı gibi benzer bir mimariyi paylaşmaktadır. Farkı ise gizli katmandaki nöronların girdilerine dayanır. YSA'de olduğu gibi girdinin ağırlıklı toplamı yerine, giriş paterni ile temel fonksiyonun merkezi arasındaki mesafenin toplamı kullanılır.

(Pao ve Takefuji, 1992), Rasgele Vektör Fonksiyonel Bağlantılı Sinir Ağı (RVFL) modelini önerdiler. Orijinal özniteliklerdeki nonlineerliği kullanan birtakım dönüşümlerle elde edilmiş zenginleştirilmiş öznitelikler ile ağın genelleştirme yeteneği iyileştirilmiştir. Temel mimarinin gösterildiği Fonksiyonel Bağlantı sinir ağlarının yarı-rastgele gerçekleştirilmesi olarak kabul edilir.

(Lecun ve ark., 1998) rakam, nesne ve konuşma tanıma gibi pek çok görevi başarıyla uygulayan Konvolüsyonel Sinir Ağı (CNN) sinir ağı modelini önermişlerdir. CNN, bazı derecelerden kayma, ölçek ve bozulma değişmezliğini sağlamak için üç mimari fikir birleştiriyor: paylaşılmış ağırlık, alt örnekleme ve yerel alıcı alanlarıdır.

(Elisseeff ve Paugam-Moisy, 1999), optimal bir ağa sahip olma olasılığı olan bir polinom zamanlı randomize algoritma olan "Jacobian Sinir Ağı" (JNN)'nı önermişlerdir. JNN'de, gizli nöronların sayısı da öğrenilebilir. Bunlardan biri rastgele üretilmiştir. Diğeri ise analitik olarak ilk rasgele sinir ağından elde edilebilen iki ağın doğrusal bir birleşimini sağlamışlardır.

(Wan ve ark., 2013) 'te, DropConnect ağını devamsızlıkların genelleştirilmesi olarak görülebilen sinir ağlarında geniş tam bağlantılı katmanların düzenlenmesi için önermişlerdir. DropConnect, bırakılma durumundan çok derin sinir ağı topluluğu olarak kabul edilebilmektedir. Test etme sürecinde, DropConnect ağı bırakma işleminin kullandığı ortalama çıkarımdan daha iyi olduğu gösterilen bir örnekleme tabanlı çıkarım kullanır. DropConnect'in Rademacher karmaşıklığının standart modellerden daha az olduğunu kanıtlamıştır.

(Gopalakrishnan, 2010), Sinir ağları yardımlı kaldırım teşhisine eğitim algoritmalarının etkisini göstermiştir. Doğrusal olmayan kaldırım tabakası modülünün FWD saptırma verilerinden tahmin etmek için, karma Sınır Elemanı-Sinir Ağları (FE-NN) destekli ters analiz başarılı bir alternatif olarak ortaya koymuştur. Bayes Regülerizasyonu (BR) prosedürü ile Lavenberg-Marquardt algoritmasının, hibrid FENN ters çevirme modelleri kullanılarak doğrusal olmayan kaldırım tabakası geri hesaplaması için en başarılı algoritma olduğu bulunmuştur.

(Ince ve ark., 2010), tıbbi tanı problemlerini çözmek için sinir ağı sınıflandırıcılarının eğitimine uygulanan küresel ve yerel teknikler arasındaki performansı araştırmaktadır. Bu çalışmada, iki popüler sinir ağı eğitim yöntemi olan geleneksel geri yayılım (BP) ve parçacık sürüsü optimizasyonu (PSO) araştırıldığı ileri-beslemeli tam-bağlı yapay sinir ağı (YSA) üzerinde uygulanmıştır. Proben1 test verilerinden üç tıbbi teşhis problemi (göğüs kanseri, kalp hastalığı ve şeker hastalığı) üzerinde performans analizi ve kapsamlı bir değerlendirme yapılmıştır. Son olarak, beklendiği gibi, BP'nin PSO'dan hesaplama açısından daha verimli olduğu bulunmuştur.

(Bhargavi ve Jyothi, 2010), Naive Bayes veri madenciliği tekniklerini ve genetik algoritmaları kullanarak tarla verilerinde toprakları hızla sınıflandırmak için uygulanmıştır. Bu çalışmada, makine öğrenme teknikleri özel ekipmanı kullanarak yapılan testlerden elde edilen alan verilerinin bir veri kümesi üzerinde çalıştırılmıştır.

(Kalderstam ve ark., 2013), Yapay sinir ağlarını, doğrudan genetik algoritmasını kullanılarak pronostik bir model oluşturulmuştur. YSA modeli topluluklarına dayalıdır ve bir genetik algoritma kullanılarak eğitilmiştir. Model, beş farklı çalışmadan ve veri setinden kaynaklanan göğüs kanseri veri setinde test edilmiştir. Karşılaştırma için Cox modeli kullanılmış ve kanser veri seti için sonuçlar YSA ve Cox modelleri arasında benzer bir performans göstermiştir. Bununla birlikte YSA, Cox modeli rasgele sonuçlardan daha iyi performans gösteremediğinde sentetik doğrusal olmayan verileri doğru bir şekilde modellemeyi başarmıştır.

(Zhang, 2014), çok katmanlı önyükleme ağı (MBN) önermiştir. Önerilen ağın her katmanının karşılıklı olarak bağımsız k-kümeler grubunun merkezleri rastgele örneklenmiş veri noktalarından oluşan bir derin sinir ağı modelidir.

(Suksri ve Kimpan, 2016), yapay sinir ağlarını yakın zamanda geliştirilmiş ve sürü zekası temelli bir algoritma olan Havai Fişek (Fireworks) Algoritması ile eğitmişlerdir. Bu

araştırma Havai Fişek Algoritması tarafından eğitilmiş Yapay Sinir Ağlarını kullanan bir hava tahmin yöntemini içermektedir. Yöntemin temel amacı, Bangkok'ta bulunan Meteoroloji İstasyonu'ndan elde edilen çeşitli ölçülmüş parametrelere dayanan günlük ortalama sıcaklığı tahmin etmektir. Deneysel sonuçlar, önerilen yöntemin hava tahmini için başarılı olduğunu göstermektedir.

(Zhang ve ark., 2017), yapay sinir ağlarının eğitimi için yeni bir meta sezgisel teknik olan toplu karar optimizasyon algoritmasını (CDOA) önermişlerdir. İnsanın karar verme davranışı, toplu karar optimizasyon algoritması (CDOA) olarak adlandırılan yeni bir stokastik popülasyon temelli yöntem olarak modellenmiştir. CDOA'nın performansı değerlendirilmiş ve yapay sinir ağları eğitimini analiz edilmiştir.

(JosephTarigan ve ark., 2017), Otomatik park kapısı veya otomatik bilet sistemi için kullanılan Plaka tanıma için Genetik Algoritma (GA) kullanarak YSA ile eğitilmiştir. Genetik algoritma gizli nöron, öğrenme hızı ve momentum oranının optimum sayısını bulmak için kullanılmıştır. Otomatik Plaka Numarası Tanıyıcıya (APNR) uygulanan Geriye yayılma Sinir Ağı (BPNN) üzerindeki GA'nın etkililiğini belirlemiştir. GA'nın optimize edilmiş geri yayılım nöron ağları, optimize edilmemiş geri yayılım nöron ağına göre daha hızlıdır.

Girdap arama yeni bir algoritma olmasından dolayı literatürde çok az kullanılmıştır. Arama motorlarında ve dergi yayın sitelerinde yapılan incelemelerde sadece beş yayın ve bir bildiri bulunduğu saptanmıştır. Aşağıda bu yayınların içeriklerine kısaca değinilmektedir.

(Dogan ve Olmez, 2015), Bu çalışmada, analog filtrelerin grup gecikmelerini minimize etmek amacıyla Girdap Arama algoritmasına dayalı bir yöntem öne sürülmüştür. Yapılan denemeler sonucunda, grup gecikmesi 2.593 sn olan 5. dereceden alçak-geçiren filtreye bağlanan 4. dereceden optimum tüm-geçiren filtre ile grup gecikmesinin 1.547 sn'ye kadar düşürülebileceği gösterilmiştir.

(Doğan, 2016), mevcut VS algoritmasının bazı zayıflıklarının üstesinden gelmek için modifiyeli Girdap Arama algoritması (MVS) önerilmiştir. MVS algoritmasında, aday çözümler her yineleme geçişinde bir takım puan etrafında üretilir. Hesaplamalı sonuçlar, bu değişikliğin yardımıyla, mevcut VS algoritmasının küresel arama becerisinin geliştirildiğini ve MVS algoritmasının, benchmark sayısal fonksiyon seti için mevcut VS

algoritması, PSO2011 ve ABC algoritmalarından daha iyi performans sergilediğini göstermektedir.

(Sajedi ve Razavi, 2016), bildirilerinde çoklu girdap arama algoritmasını (MVSA) önermişlerdir. MVSA algoritmasında, VS algoritmasının bir girdap yerine, bir dizi girdabın bir ana vorteks ve birçok çocuk vorteksi olduğu düşünülmektedir. Ayrıca her iterasyonda en iyi çocuk vorteksi ana vortekse yönelmektedir. Böylece mekanizma yerel optimum yakınsamayı önler.

(Ozkis ve Babalik, 2017), çok amaçlı optimizasyon problemleri için yeni bir meta-sezgisel girdap arama algoritması (MOVS) önermişlerdir. MOVS algoritmasında, NSGAII'nin temel stratejilerini kullanan girdap arama algoritmasını (VS) tek amaçlıdan çok amaçlıya çevrilmiştir. MOVS algoritması 36 farklı kıyaslama fonksiyon problemi üzerinde ve NSGAII, MOCeII, IBEA, MOEA test edilmiş ve başarılı sonuçlar elde edilmiştir.

(Ali ve ark., 2018), tekli karışık soğutucu (SMR-LNG) doğal gaz sıvılaştırma işleminin enerji verimliliğini arttırmak için meta-sezgisel girdap arama algoritması incelenmesi ve optimizasyonunu önermişlerdir. Girdap arama algoritması tarafından optimize edilmiş bu yöntem temel duruma kıyasla gerekli sıkıştırma gücünü 41.5% oranında azaltmış ve temel durumla karşılaştırıldığında performans katsayısını 32.8% oranında artırmıştır. Girdap arama algoritması ayrıca, genetik ve parçacık sürü optimizasyonu algoritmaları karşılaştırılmış ve mevcut yaklaşımlar üzerinde üstün bir performans sergilediğini göstermektedir.

(Li ve ark., 2018), kaos ve Lévy uçuş stratejisine dayanan girdap arama algoritması ile yeni bir I-VS algoritması önerilmiştir. I-VS algoritmasının optimizasyon performansı 13 benchmark fonksiyonu ile doğrulanmıştır. I-VS algoritmasının iyi bir arama yeteneğine sahip olduğunu göstermektedir. I-VS algoritması, NOx emisyonlarını tahmin etmek için I-VS-FLN modeli oluşturulmuş ve Hızlı Öğrenme Ağını optimize etmek için uygulanmıştır. I-VS-FLN modelinin uygun performans ve genelleme kabiliyetine sahip olduğunu göstermektedir. I-VS algoritması, kazanın ayarlanabilir parametrelerini yanma verimliliğini arttırmak ve NOx emisyonlarını azaltmak için kullanılmıştır.

1.3. Tezin Organizasyonu

Bu tez çalışmasında meta-sezgisel optimizasyon algoritmaları kullanılarak çok katmanlı algılayıcı ağlarının eğitimi yapılmıştır. Çalışma toplam beş bölüm oluşmaktadır.

Birinci bölüm giriş bölümü olup konunun genel tanımı yapılmış, çalışmanın amacı ve önemi üzerinde durulmuştur. Konu ile ilgili önceden yapılmış olan benzer çalışmalar hakkında araştırmalar yapılmış, bilgiler verilmiş ve kısa bir analiz yapılmıştır.

İkinci bölümde; çalışmanın çıkış noktası olan yapay sinir ağlarının ortaya çıkışı, bu ağların yapıları ve modelleri ile eğitilmesi, eğitimde kullanılan optimizasyon algoritmaları ve optimizasyon problemleriyle olan bağlantısı üzerinde durulmuştur. Bunun yanı sıra literatürde en iyi bilinen ve yaygın şekilde kullanılan algoritmalardan Geri Yayılım, Levenberg-Marquardt, Yapay Arı Kolonisi (ABC) algoritmasının çalışması detaylarıyla anlatılmıştır ve bir diğer sürü zekâsı algoritması olan Parçacık Sürü Optimizasyonu (PSO) algoritmasının çalışması da ele alınmıştır ve son olarak bölüm değerlendirmesi yapılmıştır.

Üçüncü bölümde; tez çalışmasının üzerinde odaklanılan başlıca algoritma olan Girdap Arama (VS) algoritmasının çalışması detaylarıyla anlatılmıştır. Algoritmanın modeli ve çalışma prensipleri açıklanmıştır ve son olarak bölüm değerlendirmesi yapılmıştır.

Dördüncü bölümde; yapay sinir ağlarının sınıflandırma test problemleri için algoritmalar ile eğitilmesi. Önerilen algoritmaların çalışması detaylarıyla açıklandıktan sonra literatürde sıkça kullanılan birçok sınıflandırma problemi üzerinde uygulanmıştır. Elde edilen deneysel sonuçlar ve bu sonuçlara ait gösterge değerleri bu bölümde sunulmuştur ve son olarak bölüm değerlendirmesi yapılmıştır.

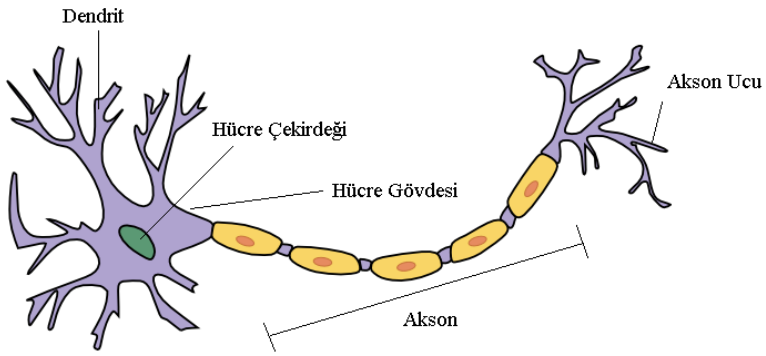
Beşinci bölüm, tez çalışmasında elde edilen sonuçlar tartışılmış ve bu konu üzerinde çalışma yapmak isteyen araştırmacılara önerilerde bulunulmuştur.

2. YAPAY SİNİR AĞLARI

Yapay Sinir Ağları (YSA) temel olarak büyük veri kümelerini modelleme aracı sağlar. Kavramsal olarak birinin verileri nasıl girdiğini, nasıl işlediğini ve nasıl öğrenildiğini hesaplayan bir temsilini sunar.

YSA biyolojik sinir ağlarından esinlenerek birbirine bağlı çok sayıda basit işlemci içeren paralel bir bilgi işlem sistemidir ve bu sistem insanda kullanılan bazı organizasyonel prensipleri kullanır.

Biyolojik sinir hücresinin yapısı Şekil 2.1'de verilmektedir. Şekilden görüldüğü gibi temel bir sinir hücresi; dendritler, hücre gövdesi, akson ve sinapslardan oluşmaktadır. Yapay nöron biyolojik nöronlardan esinlenen bir işlem birimidir. YSA'nın işlevi bilgiyi işlemek olduğundan, esas olarak veri işleme ile ilgili alanlarda kullanılmaktadır. Gerçek sinir ağlarını modellemek için kullanılan çok çeşitli YSA'lar vardır. Hayvanların ve makinelerin davranışlarını ve kontrollerini incelemek için kullanılan YSA'lar vardır aynı zamanda örüntü tanıma, tahmin etme ve veri sıkıştırma gibi mühendislik amaçları için de kullanılmıştır.



Şekil 2.1. Biyolojik nöron (Gupta ve ark., 2016)

YSA, temel olarak ağırlıklarla çarpılan girdilerden oluşur. Yapay sinir hücresine dışarıdan verilen bilgiler girdi olarak adlandırılır. Ağırlık değerleri ise hücreye girdi olarak verilen bilgilerin önemini ve hücre üzerindeki etkisini gösterir. Ağırlıklar daha sonra nöronun aktivasyonunu belirleyen bir matematiksel fonksiyondan geçirilir (Gupta ve ark., 2016).

Az sayıda nörondan oluşan bir YSA topolojisinde; nöronlar arasındaki ağırlıklar deneme yanılma yoluyla ayarlanarak, belirli girdiler için istenilen çıktılar kolayca elde edilebilir. Fakat yüzlerce veya binlerce nöronlu bir YSA topolojisinde, gerekli ağırlıkları

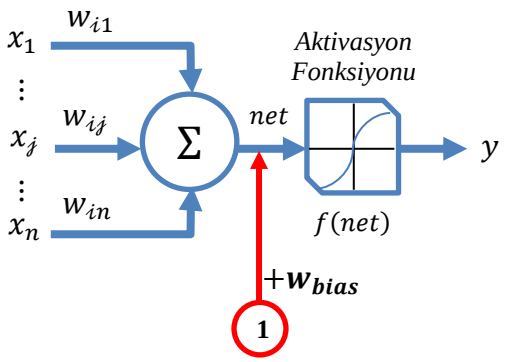
bulmak oldukça karmaşıklaşacaktır. Ağdan istenen çıktıyı elde etmek için YSA'nın ağırlıklarını ayarlayabilen algoritmalardan faydalanılır. Ağırlıkları ayarlama işlemine ağın öğrenmesi veya eğitimi olarak adlandırılır. Eğitim, rasgele ağırlıklarla başlar. YSA'da verinin dağıtık ve paralel olarak işlenebilmesini sağlayan en önemli etmen ağırlık değerleridir. Bir ağırlık değeri tek başına bir anlam ifade etmezken; yapılan toplu hesaplamalar neticesinde, uygulanan topoloji ve kullanılan mimariye göre sonuca ulaşılmasını sağlar.

YSA'da nöronlar arası bağlantıların kullanıldığı iki temel hesaplama aşaması vardır, Bunlar ileriye doğru hesaplama ve geriye doğru hesaplama aşamalarıdır (Gurney, 1997). İleri Beslemeli Ağlar genellikle statiktir. Belirli bir girdiden bir çıkış seti üretirler. İleri Beslemeli Ağlar, bir girdiye verilen yanıtın bir önceki ağ durumundan bağımsız olması anlamında belleksizdir. Geri Beslemeli Ağlar ise, çıkıştan girişe doğru ters yönlü olarak da hesaplama yapar. Geri Beslemeli Ağlar bu anlamda dinamiktir ve hafızaları vardır. Bir andaki çıkış hem o andaki hem de önceki girişleri yansıtma yeteneğine sahiptir (Benusková ve Lubica, 1994).

2.1.Yapay Sinir Hücresi

Yapay sinir ağları paralel çalışan ve nöron olarak adlandırılan basit işlem birimlerinden oluşur. Bu birimler biyolojik sinir hücrelerinden esinlenerek geliştirilmiştir. Belirli bir işlevi yerine getirmek amacıyla, birimler arasındaki bağlantıların değerleri ayarlanarak sinir ağı eğitilmiş olur. Bu bağlantılar genellikle ağırlık şeklinde ifade edilir. Yapay sinir ağları üç tür katmandan oluşur. Sırasıyla bu katmanlar; (1) ele alınan problemin öznitelik sayısıyla eşit sayıda nöron içeren giriş katmanı, (2) doğrusal olmayan NP-Zor problemlerin çözümünde muhakkak olması gereken ve en az bir nöron içeren gizli katman ve (3) son olarak ele alınan problemin çıktı adedi ile eşit sayıda nöron içeren çıkış katmanıdır.

Giriş ve çıkış katmanlarındaki nöron sayısı probleme bağlı iken; gizli katmanda bulunan nöron sayısı ve gizli katman adedi uygulanacak topolojiye göre farklılık gösterebilir. Şekil 2.2'de yapay bir nöronun yapısı görülmektedir. Yapay nöronun çıkışı (y) matematiksel olarak denklem (2.1) ve denklem (2.2) ile hesaplanır.



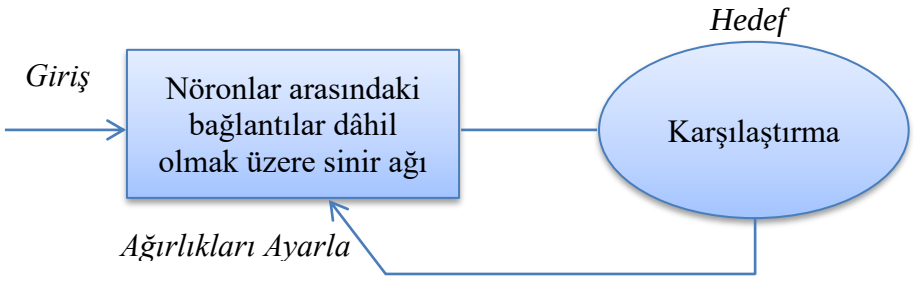
Şekil 2.2. Yapay Bir Sinir Hücresinin Temel Bileşenleri

$$net = \sum_{i=1}^{n} w_{ij} x_j + w_{bias} \quad (2.1)$$

$$y = f(net) \quad (2.2)$$

Girişlerde uygulanan değerler (X_j), bağlantılar üzerindeki ağırlıklar (W_{ij}) ile çarpılır ve toplayıcı fonksiyona (Σ) bir eşik değeri bias (b) ile birlikte uygulanır. Bias yapay sinir ağı sisteminin girdilerinden bağımsız ve giriş değeri sabit 1(bir) alınan bir nöronun ağırlık değeridir. Eğitim sürecinde ağırlık değerlerini dengelemek amacıyla gizli ve çıkış katmanlarına uygulanır.

Ağın çıkışı beklenen hedefe uyana kadar yapay sinir ağı eğitilir. Yapay sinir ağı, çıktısı bilinen girdi değerleri ile eğitildikten sonra sonucu bilinmeyen değerler için karar vermeyi gerçekleştirebilir (Billah ve ark., 2016). Bu süreç şematik olarak Şekil 2.3 ile ifade edilmektedir.



Şekil 2.3. YSA çalışma prensibi

Her bir nöronun net çıkışı, aktivasyon veya transfer fonksiyonu olarak adlandırılan bir işlemden geçirilerek hesaplanır. Doğrusal, radyal, logaritmik, sigmoid ve hiperbolik tanjant gibi farklı türlerde aktivasyon fonksiyonları kullanılmaktadır. Bu çalışmada, denklem (2.3) ile hesaplanan sigmoid fonksiyonu hem giriş hem de çıkış katmanlarındaki nöronların transfer fonksiyonu olarak kullanılmıştır.

$$Y = f(net) = \frac{1}{1+e^{-net}} \quad (2.3)$$

Yapay sinir ağı uygun bir öğrenme algoritması kullanılarak yeterli sayıda veri ile eğitilirse, en karmaşık ve doğrusal-olmayan çok değişkenli sistemlerde bile kabul edilebilir sonuçlar elde edilebilir (SINGH ve SINGH, 2005);(Vaferi ve ark., 2014). Eşlenik gardyan, geri yayılım, kademeli korelasyon, parçacık sürü optimizasyonu, evrimsel teknikler ve genetik algoritmalar YSA eğitimi için en bilinen algoritmalarından bazılarıdır. Bazı YSA metodolojilerinin veri modelleme ve fonksiyon yaklaşımı için daha yararlı olduğu, bazılarının ise kavramsal problemleri çözmeye uygun olduğu bildirilmiştir (Basheer ve Hajmeer, 2000). İleri besleme yapısına sahip olan çok katmanlı algılayıcı sinir ağı fonksiyon yaklaşımı ve regresyon temelli çalışmalar için en iyi bilinen mimari modeldir (Gibbs ve ark., 2006) ; (Maier ve ark., 2000); (Topcu ve Saridemir, 2008).

ÇKA; giriş katmanı, gizli katman ve çıkış katmanından oluşan 3 katmanlı ve tam bağlı yapay sinir ağı mimarisine sahiptir. Birinci katman, bağımsız değişkenlerin bilgilerini dış kaynaktan almakla sorumludur. Giriş sinyalleri üzerindeki Denklem (2.1)'e göre hesaplama yapıldıktan sonra sonucu bir sonraki katmana aktarır. Bu işlem, manipüle edilen giriş bilgileri gizli katmandan son katmana ulaşana kadar devam eder. Son katman, ÇKA sinir ağı modelinin çıkış sinyallerinin hesaplanmasından sorumludur.

ÇKA sinir ağının ampirik korelasyonlara kıyasla sağladığı başlıca avantajları; deneysel bağıntılar üzerinde bağlılık ve çoklu uygulanabilirlik, esneklik, geri yayılım algoritmasının gücü, nispeten az bellekli ve hızlı olma gereksinimi, giriş ve çıkış desenleri arasındaki karmaşık ilişkileri öğrenme yeteneği, istatistiksel dağılım varsayımlarından bağımsızlığı, ayarlanabilir parametrelerin sayısını arttırma kabiliyeti olarak sıralanabilir (Gholami ve ark., 2018) ; (Dach ve ark., 2016) ; (Hagan ve ark., 1996).

2.2. Yapay Sinir Ağlarının Eğitilmesi

Deneysel bir veri seti ile ÇKA sinir ağının ağırlıklarının ve biaslarının ayarlanması işlemine ağın eğitimi veya öğrenmesi denir. Ele alınan veri seti için genellikle deneme-yanılma yoluyla uygun bir ağ topolojisi belirlenir. Bu nedenle, verilen yapıya sahip bir ÇKA ağı için ağırlık ve biasların sayısı bilinir. Eğitim bu ağırlık değerleri için rastgele başlangıç değerleri ile başlatılır. Eğitim algoritması, verilen deneysel verileri kullanarak uygun ağırlık değerlerini ayarlamaya çalışır (Hassanpour ve ark., 2018) ; (Yilmaz ve Kaynar, 2011). Aslında, eğitim algoritması deneysel ve öngörülen değerler arasındaki

sapmayı en aza indirgemeye çalışır. Daha sonra, eğitimli ÇKA ağının performansı, test kümesi olarak ayrılan ve eğitim süreci sırasında kullanılmayan verilerle doğrulanmalıdır.

Kullanılan öğrenme algoritmaları ile hata azaltılıp gerçek çıkış ile uyumu en yüksek çıkış değerinin elde edilmesi öğrenme sürecinin amacıdır. Bu amaçla, sistem içerisinde ağırlıklar her bir çevrimde yenilenerek hata azaltılmaya çalışılır. Eğer yapay sinir ağı verilen giriş çıkış çiftleriyle amaca ulaşmış ise ağırlık değerleri saklanır. Ağırlıkların sürekli yenilenip istenilen sonuca ulaşılana kadar geçen işleme "öğrenme" adı verilir. Yapay sinir ağı öğrendikten sonra daha önce verilmeyen girişler verilip ağ çıkısıyla gerçek çıkısın yaklaşımı incelenir ki verilen bu girişlere "test seti" denir. Eğer yeni verilen örneklere de doğru yaklaşıyorsa sinir ağı başarılı bir şekilde öğrenmiş sayılır. Ağ tarafından öngörülen çıkış ile istenen çıkış arasındaki hata oranı, daha önce belirlenen bir sınıra erişmiş ise, ağın problemi yeterince kavradığı kabul edilir. ÇKA parametreleri için başlangıç değerleri ile başlar ve iterasyon sayısını arttırarak değerlerini değiştirir. Gözlemlenen sapma önceden belirlenmiş bir değere ulaşır ulaşılmaz eğitim tamamlanır ve ağırlıklar en iyi değerlerini bulur (Davoudi ve Vaferi, 2017).

Yapay sinir ağlarının eğitimi için literatürde birçok yöntem önerilmiştir. En çok kullanılan yöntem Meyilli Azalma (gradient descent) stratejisini kullanarak toplam hatanın asgarileştirilmesini sağlayan Geriye Yayılım (BP) algoritmasıdır. Bu yöntemde; hatanın azaltılması ve optimal ağırlık değerinin bulunması, değişim miktarının türevi alınarak yapılır. Sistemin toplam hata miktarı ise veri setindeki yerel hataların toplamıdır. Bu değer beklenen çıktıdan ne kadar farklı olduğu hesaplanarak bulunur. Bununla birlikte; BP algoritması, yavaş yakınsamasından dolayı bir dezavantajı vardır.

Meyilli Azalma metodunun yavaş yakınsaması GaussNewton algoritması ile büyük ölçüde iyileştirilebilir. Özellikle hata fonksiyonu ikinci dereceden bir yüzeye sahipse, GaussNewton her yön için uygun adım boyutlarını bulabilir ve daha ilk adımda hızlı bir şekilde yakınsama sağlayabilir. Ancak bu iyileştirme yalnızca hata fonksiyonunun kuadratik bir yakınsaması mümkün olduğunda gerçekleşir. Aksi takdirde, GaussNewton algoritması da yakınsama sağlayamaz (Billah ve ark., 2016).

2.3. Eğitim Algoritmaları

Girişler ve çıkışlar arasındaki 'öğrenme' yeteneği yapay sinir ağlarını çekici hale getiren başlıca avantajlarından birisidir. Gerçekleştirilecek hesaplamada farklı verilere göre ağın ağırlıklarını belirlemek için çeşitli öğrenme algoritmaları önerilmiştir. YSA, öğrenme yeteneği sayesinde örüntü tanıma, tıbbi teşhis, zaman serisi tahmini gibi doğrusal olmayan problem türleri için etkili bir çözüm aracı olmuştur.

Öğrenme algoritmalarının yakınsamasını hızlandırmak için literatürde önemli araştırmalar yapılmıştır. Bu çalışmalar iki kategoride toplanabilir: (1) öğrenme hızını değiştirme, momentumu kullanma ve değişkenleri yeniden ölçeklendirme gibi fikirleri içeren geçici sezgisel tekniklerin geliştirilmesi; (2) standart sayısal optimizasyon tekniklerinin geliştirilmesi. Yapay sinir ağı eğitimi için yaygın olarak kullanılan üç tür sayısal optimizasyon tekniği Eşlenik Gardyan algoritmaları, Quasi-Newton algoritmaları ve Levenberg-Marquardt algoritmasıdır (Hagan ve Menhaj, 1994).

Bugüne kadar literatürde yapay sinir ağlarının eğitimi için çok sayıda algoritma önerilmiştir. Ancak belirli bir problem için yakınsama hızı ve doğruluk açısından hangi algoritmanın en iyi çalıştığını belirlemek zordur. Sorunun karmaşıklığı, eğitimde kullanılan veri kümelerinin sayısı, ağdaki ağırlık ve biasların sayısı, hata toleransı, ağda kullanılan transfer fonksiyonu veya ağın sınıflandırma için kullanılıp kullanılmadığı gibi etmenler ağın öğrenme performansını etkilemektedir (Coskun ve Yildirim, 2003). Bu çalışmada ele alınan çeşitli eğitim algoritmaları sonraki bölümlerde kısaca açıklanmaktadır.

2.3.1. Klasik Algoritmalar

2.3.1.1. Geri Yayılım Algoritması (BP: Backpropagation)

BP algoritması danışmanlı bir öğrenme yöntemidir. Yapay sinir ağı öğrenme modelleri arasında en çok kullanılanlardan birisidir. Bu algoritma bilinen bir veri seti girdi-hedef çıktı örnekleri ile uygulanır. Tahmin ve sınıflandırma uygulamalarında sıkça kullanılmıştır.

BP'nin arkasındaki temel fikir, öğrenme işlemi sırasında toplam çıktı hatasını aşamalı olarak en aza indirmektir. Bununla birlikte, eğitim setleri doğru çıktıyı tahmin etmek için girdi katmanı aracılığıyla yinelemeli olarak uygulanır. Geri yayılım algoritmasında her yinelemede, ileriye doğru hesaplama ve geriye doğru hesaplama olmak üzere iki aşamaya ayrılmıştır. İleri hesaplama aşamasında YSA’nın o andaki durumunda

YSA'ya uygulanan giriş işaretlerine karşı YSA'nın çıkışlarında oluşan değerler bulunur. Geriye doğru hesaplama aşamasında, çıkışlarda oluşan hatalardan yola çıkılarak devredeki ağırlıkların yeniden düzenlenmesi yapılmaktadır.

Geri yayılım algoritmasının adımları Şekil 2.4'de verildiği gibidir.

Adım 1: **Repeat**

Adım 2: Eğitim kümesinden rasgele bir sonraki örneği seçme ve ağ girişine giriş vektörü uygulama

Adım 3: Ağın çıkısını hesaplama

Adım 4: Ağın çıkısı ile hedef çıkış arasındaki hatayı hesaplama

Adım 5: Ağın ağırlıklarını geriye gönderilen hata yardımıyla güncelleme

Adım 6: **Until** Durma kriteri

Şekil 2.4. Geriye yayım algoritmasının sözde kodu.

Eğitimin k'nıncı yinelemesinde YSA'nın çıkısındaki i' inci nöronunun çıkış değeri y_i, beklenen değer d_i ve i nöronunun hatası e_i olarak temsil edilir ise, hata değerinin hesaplanması Denklem (2.4) ile tanımlanır:

$$e_i = (d_i - y_i(k)) \tag{2.4}$$

Geri Yayılım Sinir Ağı (BPNN) genel yapısında bir katmandan diğer katmana, aradaki katmanı atlayarak geçebilmek mümkün değildir. Ağın ilk katmanındaki nöronlara bir giriş verisi uygulandığında en üst katman olan çıkış katmanına erişinceye kadar, bu veri üzerinde çeşitli işlemler gerçekleştirilir. Bu işlemlerin sonunda elde edilen çıktı, olması gereken çıktı ile karşılaştırılır ve toplam hata Denklem (2.5) ile tanımlanır:

$$E = \frac{1}{2}\sum_i (d_i - y_i(k))^2 \tag{2.5}$$

Güncellenen her ağırlık için iteratif olarak meyilli azalma yöntemi kullanılır ve hata en aza indirilir (Lin ve Lee, 1996). Denklem (2.6) ile hesaplanır.

$$\Delta\omega_{ij} = -\eta \frac{\partial E(\vec{w})}{\partial \omega_{ij}} \tag{2.6}$$

Burada η negatif gradyan yönündeki her yinelemenin adım boyutunu temsil eden öğrenme katsayısını tanımlar.

Geri yayılımda tüm katmanlar arasındaki bağlantılar üzerindeki ağırlıklar minimum hata *E* ile en uygun ağırlıkların bulunmasına kadar hedef ile çıktı arasındaki hatayı en aza

indirgeyecek şekilde güncellenir Delta (∂) olarak adlandırılan i. katmanının nöronlarına ait hata Denklem (2.7) ile hesaplanır.

$$\frac{\partial E}{\partial w_{ij}} = -\delta_o . y_i \tag{2.7}$$

Geri yayılım aşaması girdiye göre hatanın eğimini hesaplar. Her bir ağırlığa w_{ij} eklenerek meyilli azalma gerçekleştirilir. w_{ij} ağırlığının değişim miktarı Denklem (2.8) ile hesaplanır.

$$\Delta\omega_{ij} = \eta\delta_o y_i \tag{2.8}$$

δ_o çıkış katmanın sinirleri Denklem (2.9) ile hesaplanır.

$$\delta_o = e_i(k) f_j \tag{2.9}$$

Gizli katmanlarda bulunan sinirler için ise Denklem (2.10) ile hesaplanır.

$$\delta_o = f_j \sum_m \delta_m w_{mo} \tag{2.10}$$

f_j, j sinirinin aktivasyon fonksiyonudur. Bu tanımlar ile hata işaretlerinin çıkıştan girişe doğru akışı, ileri yayılma aşamasında işaretlerin ileri doğru akışına benzetilmiştir.

BPNN, birçok araştırmacı tarafından çok çeşitli uygulamalarda kullanılmıştır. Geleneksel makine öğrenme teknikleri kullanılarak çözülemeyen karmaşık problemleri çözmek için yüksek bir kabiliyete sahiptir (Yousefi ve Hamilton-Wright, 2014). Pazarlama (Chiang ve ark., 2006), Biyobilişim (hofestadt ve Töpel, 2005), tıp (Behrman ve ark., 2007), mühendislik (Dvir ve ark., 2006) ve diğerleri (Lee ve ark., 2005) üzerinde birçok BPNN uygulaması bulunmaktadır. Bununla birlikte; algoritma performansını kısıtlayan birkaç engel bulunmaktadır. Bu engellerin üstesinden gelmek için yakınsama hızını arttıran ve meyilli azalma hatasını verimli bir şekilde azaltan momentum katsayısı önerilmiştir (Shao ve Zheng, 2009).

Momentum Katsayısı Geri-Yayılım (BP) yapay sinir ağlarının öğrenme performansını etkiler. Momentum tekniği bir önceki iterasyondaki değişimin belirli bir oranının yeni değişim değerine ilave edilmesi olarak görülmektedir. Optimum kararlı durum erişmek için yakınsama sürecini hızlandırır ve üstün bir şekilde geliştirmek için hata ayarını en aza indirmektedir. Ağırlıkların güncellemesi için düşük hata performansı sağlar Bu özellikle yerel çözümlere yönelik ağların daha iyi sonuçlar elde etmesini sağlamak amacı ile önermiştir. Bu değerin küçük olması yerel çözümlerden kurtulmayı zorlaştırabilir ve aynı zamanda çok büyük değerlerde de tek bir çözüme ulaşmada sorunlar yaşanabilir.

Literatürde tavsiye edilen bu değerin 0.6-0.8 arasında seçilmesinin uygun olacağını göstermektedir (Öztemel, 2016).

2.3.1.2. Levenberg Marquardt Algoritması

LevenbergMarquardt algoritması meyilli azalma metodunu ve GaussNewton algoritmasını birleştirir. GaussNewton algoritmasının hızından ve meyilli azalma metodunun istikrarından faydalanır. GaussNewton algoritmasından daha kararlıdır. Çünkü birçok durumda hata yüzeyi ikinci dereceden çok daha karmaşık olsa bile iyi bir şekilde yakınsayabilir.

Geri yayılım algoritmasının yakınsama hızı düşüktür ve yerel minimuma yakalanma riski oldukça yüksektir. Bu durumların ortaya çıkabileceği problemler için genelde Levenberg-Marquardt (LM) algoritması tercih edilmektedir. BP algoritması birinci dereceden türev bilgisiyle hatayı azaltmaya çalışırken LM algoritması Newton metodu ile BP metodu arasında ara değerleri hesaplar ve ikinci dereceden türev bilgisiyle hatayı azaltmaya çalışır (Hagan ve Menhaj, 1994). LM metodu, maksimum komşuluk fikri üzerine kurulmuş en küçük kareler hesaplama metodudur (Levenberg, 1944).

Meyilli azalma metodundan ve Newton algoritmasından elde edilen LM algoritmasının güncelleme kuralı Denklem (2.11) ile hesaplanır:

$$\Delta\omega = (J^T J + \mu^I)^{-1} J^T e \tag{2.11}$$

Burada ω ağırlık vektörüdür I, birim matrisi kombinasyon katsayısıdır, J Jacobian matrisi ve e, hata vektörüdür.

$$J = \begin{vmatrix} \frac{\partial e_{11}}{\partial \omega_1} & \frac{\partial e_{11}}{\partial \omega_2} & \cdots & \frac{\partial e_{11}}{\partial \omega_N} \\ \frac{\partial e_{12}}{\partial \omega_1} & \frac{\partial e_{12}}{\partial \omega_2} & \cdots & \frac{\partial e_{12}}{\partial \omega_N} \\ \cdots & \cdots & & \cdots \\ \frac{\partial e_{M}}{\partial \omega_1} & \frac{\partial e_{1M}}{\partial \omega_2} & \cdots & \frac{\partial e_{1M}}{\partial \omega_N} \\ \cdots & \cdots & & \cdots \\ \frac{\partial e_{P1}}{\partial \omega_1} & \frac{\partial e_{P1}}{\partial \omega_2} & \cdots & \frac{\partial e_{P1}}{\partial \omega_N} \\ \frac{\partial e_{P2}}{\partial \omega_1} & \frac{\partial e_{P2}}{\partial \omega_2} & \cdots & \frac{\partial e_{P2}}{\partial \omega_N} \\ \cdots & \cdots & & \cdots \\ \frac{\partial e_{PM}}{\partial \omega_1} & \frac{\partial e_{PM}}{\partial \omega_2} & \cdots & \frac{\partial e_{PM}}{\partial \omega_N} \end{vmatrix} \tag{2.12}$$

Burada P, eğitim örneklerinin sayısıdır. M, çıktıların sayısıdır. N, ağırlıkların sayısıdır. Hata vektöründeki elemanlar Denklem (2.13) ile hesaplanır:

$$e_{pm} = d_{pm} - O_{pm} \tag{2.13}$$

Burada d_{pm} ve O_{pm} eğitim modeli olduğunda, sırasıyla ağ çıkışında istenen çıktı ve hesaplanan çıktıdır.

LM algoritmasının adımları Şekil 2.5'de verilmiştir.

Adım 1: **Repeat**

Adım 2: Jakobyeni hesapla

Adım 3: Hata gradyanını ($\mathrm{g} = J^T E$) hesapla

Adım 4: Hessian değerini ($\mathrm{H} = \mathrm{J^T J}$) bul

Adım 5: δ değerini bulmak için $(\mathrm{H} + \lambda \mathrm{I})\delta = \mathrm{g}$ denklemini çöz

Adım 6: δ değerini kullanarak ağdaki ağırlıkları güncelle

Adım 7: **Until** Durma kriteri

Şekil 2.5. Levenberg Marquardt algoritmasının sözde kodu

LM algoritması aşağıda ifade edilen hatanın (E) değişim durumlarına göre çalışmaktadır.

1) Yukarıdaki kurallara göre bir güncelleme yap.
2) Yeni ağırlık vektöründeki hatayı değerlendir.
3) Güncelleme sonucunda hata artarsa, ağırlıkları önceki değerlerine sıfırla ve μ değerini artır. Ardından adım-1'e git
4) Güncelleme sonucunda hata azalmışsa, adımı kabul et (diğer bir deyişle, ağırlıkları yeni değerlerde tut) ve μ değerini azalt.

2.4. Parçacık Sürü Optimizasyonu

Parçacık Sürü Optimizasyon Algoritması (PSO), ilk kez Kennedy ve Eberhart tarafından 1995 yılında tanıtıldı (Kennedy ve Eberhart, 1995). Bu algoritma günümüzde popüler olan sürü zekâsı temelli bir algoritmadır. PSO algoritması doğadan esinlenen ve tekrarlamaya dayanan evrimsel bir algoritmadır. Bu algoritmanın ilham kaynağı, kuş ve balık gibi sürü halinde hareket eden hayvanların sosyal davranışlarıdır. Bu sistemde, üyeler kendi deneyimlerinden ve sürünün kolektif deneyiminden faydalanır. Böylece, bireysel olarak basit ajanlar olmalarından dolayı yerine getiremeyecekleri karmaşık görevleri kolektif olarak başarabilen akıllı bir sistem görevi görürler. Son zamanlarda, sürü zekasına dayanan evrimsel algoritmaları araç olarak kullanan veya performanslarını iyileştirmek için üzerinde modifikasyon önerisi yapılan birçok araştırma bulunmaktadır. Bu algoritmalardan bazıları yapay arı kolonisi (Karaboga ve Basturk, 2007), diferansiyel

evrim algoritması (Zeng ve ark., 2016), karınca koloni optimizasyonu (Dorigo ve ark., 2006), meyve sineği optimizasyon algoritması (Lv ve ark., 2017), yarasa algoritması (Yang, 2010), çekirge sürüsü algoritması (Cui ve ark., 2016) , yerçekimi arama algoritması (Rashedi ve ark., 2009) olarak sıralanabilir.

Parçacık Sürü Optimizasyonunda, popülasyondaki her bir üyeye bir parçacık denir. Popülasyon tabanlı bir algoritma olan PSO, belirli sayıda parçacıktan oluşur. Her bir parçacık için konum ve hız bilgileri vektörel olarak tutulur. Yani her parçacık, iki vektörle ifade edilir: konum vektörü X_i ve hız vektörü V_i. X_i, optimize edilmiş problemin potansiyel bir çözümünü temsil eder ve hız vektörü V_i, pozisyon değiştirme hızını yansıtır. Algoritma her bir parçacık için rastgele üretilen konum değerleri ile başlatılır. Bu parçacıklar, çok boyutlu bir arama uzayında hareket ederler. Her biri arama uzayında farklı bir noktayı ifade eden parçacıkların uygunluğu hesaplanır. Amaç diğer meta-sezgisel optimizasyon tekniklerinde olduğu gibi; tüm alanı aramadan optimal çözüm değerlerini tespit etmeye çalışmaktır. Arama alanının boyutu, optimize edilmesi gereken fonksiyon parametrelerinin sayısına eşittir. Her parçacık iterasyonlar boyunca bulduğu en iyi konumu belleğinde saklar. Tüm parçacıklar arasında bulunan en iyi konum ise başka bir bellek alanında saklanır. Bu iki bilgi kullanılarak parçacıkların nasıl hareket edeceğine karar verilir. Bu işlem sonlanma ölçütü sağlanıncaya kadar tekrar edilir. Her yinelemede, tüm parçacıklar, küresel en iyi nokta bulunana kadar çok boyutlu arama uzayında hareket eder. Parçacıklar hızlarını ve konumlarını yerel ve global en iyi cevaplara göre güncelleştirir (Kennedy ve Eberhart, 1995).

PSO amaç fonksiyonun (f) optimal çözümü $\mathrm{x}^{\mathrm{opt}}$'i bulmak için uygulanan global bir optimizasyon metodudur. D boyutlu bir optimizasyon problemi araştırılırken, parçacık $\mathrm{X_i} = (\mathrm{x_i^1}, \ldots, \mathrm{x_i^D})$, olarak temsil edilir ve hız $\mathrm{V_i} = (\mathrm{v_i^1}, \ldots, \mathrm{v_i^D})$ olarak temsil edilir. Her parçacık kendi en iyi konum bilgisi olan $\mathrm{Pbest_i} = (\mathrm{p_i^1}, \ldots, \mathrm{p_i^D})$'i hafızasında tutar. Bu parçacıklar, en iyi global konum $\mathrm{Gbest_i} = (\mathrm{g_i^1}, \ldots, \mathrm{g_i^D})$ bulmak için belirli bir hızda hareket eder. En uygun çözümü aramak için parçacık konumunu Denklem (2.14) ve hızını Denklem (2.15) 'e göre günceller:

$$V_i^D(t+1) = V_i^D(t) + c_1 r_1 \left(Pbest_i^D(t) - X_i^D(t) \right) + c_2 r_2 \left(Gbest_i^D(t) - X_i^D(t) \right) \tag{2.14}$$

$$X_i^D(t+1) = X_i^D(t) + V_i^D(t+1) \tag{2.15}$$

Burada t iterasyon numarasıdır. r_1 ve r_2, [0,1] aralığında rastgele üretilen sayılardır. c_1 ve c_2 ivme sabitleridir. Genel olarak c_1, parçacığın hızlandırma katsayısı olarak ve c_2, sürünün katsayısı olarak görülür. Denklem (2.14) 'de gösterilen hız güncelleme denklemi her bir parçacık için bir sonraki pozisyonunu belirleyecek yeni hız vektörünün hesaplanmasını sağlar. Bu hız değişimi c ve r katsayıları oranında, parçacığın hem kendi en iyi pozisyonunun hem de global en iyi pozisyonun çevresinde arama yapılmasını sağlar. Son olarak, güncellenmiş bu hız vektörüne göre her bir parçacık kendi pozisyonunu değiştirir. Her parçacığın hızı $[Vmin, Vmax]$ aralığında sınırlandırılmıştır. Arama alanının çok uzağında hareket eden boyutsal değerin önlemek için pozisyon $[Xmin, Xmax]$ aralığında sınırlandırılmıştır. Algoritma güncelleme denklemini, optimum bir çözüm bulunana kadar veya önceden belirlenmiş yineleme sayısına ulaşılana kadar tekrarlanır. PSO algoritmasının sözde kodu Şekil 2.6'de verilmiştir.

PSO Algoritması

For her parçacık için
    Parçacığı başlangıç konumuna getir
End

Do
  For her parçacık için
    Uygunluk değerini hesapla
    Eğer uygunluk değeri pbest ten daha iyi **ise**,
      Şimdiki değeri yeni pbest olarak ayarla
  End

  Tüm parçacıkların bulduğu pbest değerlerinin en iyisini, tüm parçacıkların gbest'i olarak ayarla

  For her parçacık için
    Denklem (2.14)'e göre parçacık hızını hesapla
    Denklem (2.15)'ye göre parçacık konumunu güncelle
  End
While maksimum iterasyon sayısına veya minimum hata koşulu sağlanana kadar devem et.

Şekil 2.6. PSO algoritmasının sözde kodu

Parçacık Sürüsü Optimizasyonu, güçlü bir algoritmadır ve birçok gerçek dünya problemine uygulanır. Bununla birlikte, arama alanının önyargıları geleneksel PSO'da mevcut olduğundan, ayrılmaz problemlerde arama performansı nispeten iyi değildir. Bu durumun üstesinden gelmek için Clarc Parçacık Sürüsü Optimizasyonu 2011 (PSO2011)'i önermiştir (Zambrano-Bigiarini ve ark., 2013). PSO2011'in en önemli faktörü, arama aralığının merkezi ve arama yarıçaplıdır. Merkez hem yerel hem global en iyi konumun ağırlık merkezi ile belirlenir.

PSO2011'in arama aralığı küresel bir biçimde dağıtılmıştır. Küresel arama yeteneğini geliştirilmiş bir güncelleme kuralı CEC2005 benchmark fonksiyonlarını kullanarak etkinliğini artırmaktadır. Şekil 2.7 görüldüğü gibi PSO2011 rotasyonel değişmezlik fikrinden yararlanır. Her bir parçacık için ve her bir zaman adımında, üç nokta etrafında ağırlık merkezi ($\vec{G}_i^D$) tanımlanır: mevcut konum (X_i^D), bir önceki en iyi yerel konumunun biraz "ötesinde" bir nokta ($\vec{p}_i^D$), ve bir önceki en iyi konum komşuluğun biraz "ötesine" bir nokta($\vec{l}_i^D$):

$$\vec{p}_i^D = \vec{X}_i^D + c_1 r_1\left(\mathrm{Pbest}_\mathrm{i}^\mathrm{D}(\mathrm{t}) - \mathrm{X}_\mathrm{i}^\mathrm{D}(\mathrm{t})\right) \tag{2.16}$$

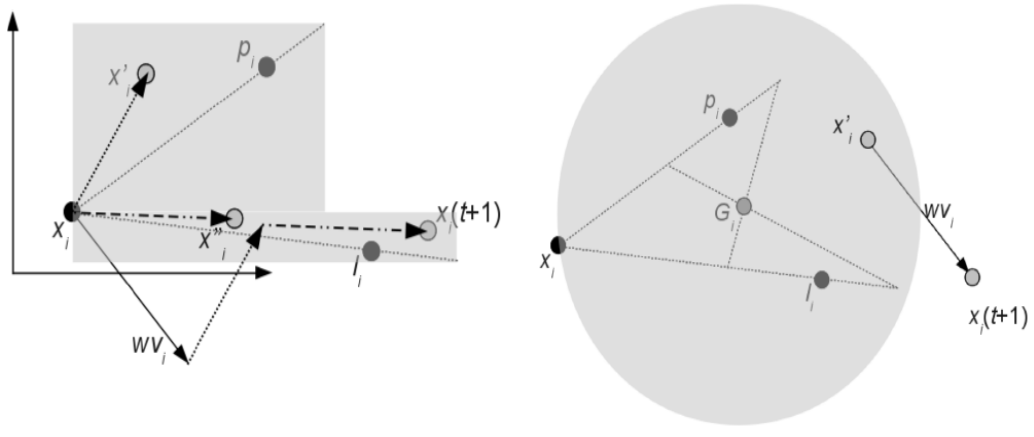
$$\vec{l}_i^D = \vec{X}_i^D + c_2 r_2\left(Gbest_i^D(t) - X_i^D(t)\right) \tag{2.17}$$

$$\vec{G}_i^D = \frac{(\vec{X}_i^D + \vec{p}_i^D + \vec{l}_i^D)}{3} \tag{2.18}$$

Denklemler (2.16) ve (2.17), arama aralığının merkezini belirler. PSO2011'in rotasyonel değişmez özelliği vardır. $\mathcal{H}_i(\vec{G}_i^D, ||\vec{G}_i^D - \vec{X}_i^D||)$ merkezi $\vec{G}_i^D$ ve yarıçap $||\vec{G}_i^D - \vec{X}_i^D||$ile küresel bir dağılımdır. PSO2011'in hız güncelleme kuralı denklem (2.19) hesaplanır:

$$\vec{V}_i^D(t+1) = \vec{V}_i^D(t) + \mathcal{H}_i(\vec{G}_i^D, ||\vec{G}_i^D - \vec{X}_i^D) - \vec{X}_i^D(t) \tag{2.19}$$

Parçacık pozisyonu denklem (2.15) sonrasında güncellenir.



(a) (b)

Şekil 2.7. (a) 2007 yılına kadar Canonical ve Standard PSO algoritmalarının geometrik yorumu. (b) Standart PSO 2011 (PSO-2011) algoritmasının geometrik yorumu (Zambrano-Bigiarini ve ark., 2013).

PSO 2011 algoritmasının işleyişi Şekil 2.8'de açıklanmaktadır.

PSO2011 Algoritması

For i=1 to N **do** {sürüdeki her parçacık için}
Parçacıkların pozisyonlarını ($\vec{X}_i$) ve hızlarını ($\vec{V}_i$) başlat
Önceki en iyi $\vec{P}_i$ ve yerel en iyi $\vec{G}$ başlat
End for
Repeat
  For i=1 to N **do**
    Denklem (2.18) kullanarak parçacığın hızını güncelle
    Denklem (2.15)'yi kullanarak parçacığın konumunu güncelle
    if $f(\vec{X}_i) < f(\vec{P}_i)$ **then** {f en aza indirgeme}
      Parçacık en bilinen konumu $\vec{P}_i = \vec{X}_i$ güncelle
      if $f(\vec{P}_i) < f(\vec{L})$ **then** {f en aza indirgeme}
        Komşunun bilinen en iyi pozisyonunu güncelle $\vec{L} = \vec{P}_i$
      end if
    end if
  end for
Until [yineleme sayısı (T) veya tolerans hatası karşılanır]

Şekil 2.8. PSO(2011) algoritmasının sözde kodu

2.5. Yapay Arı Kolonisi Optimizasyonu

Karaboğa, Yapay Arı Kolonisi Optimizasyon Algoritması (ABC) sürü zekasına dayanan meta-sezgisel ve stokastik bir optimizasyon algoritmasıdır (Karaboga, 2005). Bu algoritmada doğada besin kaynaklarını arayan bal arısı kolonisinin davranışından esinlenilmiştir. Koloninin yapay arıları üç gruba ayrılır: İşçi arılar, gözcü arılar ve kâşif arılar. İşçi arı ve gözcü arı sayısı besin kaynakları sayısına eşittir. Her arı ayrı bir besin kaynağı ile ilişkilidir. İşçi arılar hafızasındaki besin kaynaklarını arar. Ardından kovana döndüklerinde besin kaynaklarının konumu ile ilgili bilgileri dans ederek arılarla paylaşırlar. Gözcü arılar, kovanda bekler ve hangi besin kaynağını seçileceğine karar verirler. Bu nedenle, daha faydalı besin kaynakları seçme olasılığı daha yüksektir. Besin kaynağı tükenen işçi arılar; yiyecek kaynaklarından vazgeçip yenilerini arayan kâşif arılara dönüşürler. Üç aşamadan oluşan ABC algoritmasının çalışması aşağıda açıklanmaktadır.

Başlangıç: ABC algoritması, her biri arama uzayında bir besin kayağını diğer bir ifadeyle aday çözümü gösteren bir popülasyonun rasgele oluşturulmasıyla başlar. Her aday çözüm karar değişkenlerinden oluşan çok boyutlu bir parametre vektörüdür. Bu popülasyon denklem (2.20) kullanarak başlangıçta rasgele üretilir:

$$\chi_{ij}=\chi_j^{\min}+\text{rand x}(\chi_j^{\max}-\chi_j^{\min}) \tag{2.20}$$

Burada i ve j sırasıyla $[1, SN]$ ve $[1, D]$ aralığında seçilir. SN sürünün eleman sayısı, D boyut sayısı, $rand$ [0,1] aralığında rasgele bir reel sayıdır, $\chi_j^{max} ve\ \chi_j^{mın}$ sırasıyla alt sınır ve üst sınırdır.

İşçi arı aşaması: Bu aşamada, her işçi arı yalnızca belirli bir besin kaynağı (χ_i) ile bağlantılıdır ve daha sonra komşu bir aramaya dayalı olarak hafızasındaki yiyecek kaynağını (χ_i) değiştirerek yeni bir besin kaynağı (υ_i) üretir. Böylece, yeni besin kaynağı (υ_i) Denklem (2.21) ile tanımlanır:

$$\mathcal{V}_{ij} = \mathcal{X}_{ij} + \emptyset_{ij} \times (\chi_{ij} - \chi_{kj}) \tag{2.21}$$

Burada χ_i χ_i'nin $j$inci elemanını gösterir, j rasgele bir indeks, χ_{kj} popülasyondan rasgele seçilen başka bir besin kaynağının (χ_k) j inci unsurunu temsil eder. $\emptyset_{ij}$ $[-1,1]$ aralığında rasgele bir sayıdır. Bu aşamada (χ_i) ve (υ_i) besin kaynakları birbirleriyle karşılaştırılır. işçi arı aç-gözlü seçim işlemi uygulayarak daha iyi olanı kullanır. Ardından, işçi arılar kovana geri döner ve yeni yiyecek kaynakları hakkındaki bilgileri gözcü arılar ile paylaşır.

Gözlemci Arı Aşaması: Gözcü arılar işçi arılardan aldıkları bilgileri değerlendirir ve nektar miktarına orantılı olarak bir besin kaynağı seçerler. i' inci besin kaynağının seçilme olasılığı $\mathcal{P}_i$ denklem (2.22) ve denklem (2.23)' e göre hesaplanır.

$$\mathcal{P}_i = \frac{fit_i}{\sum_j fit_j} \qquad j=1,2, \dots, SN \tag{2.22}$$

$$fit_i = \begin{cases} \frac{1}{1+\mathrm{f}_i} & \mathrm{f}_i \geq 0 \\ 1 + \mathrm{abs}(\mathrm{f}_i) & \mathrm{f}_i < 0 \end{cases} \tag{2.23}$$

Burada fit_i ve f_i sırasıyla, i. parçacığın uygunluk değeri ve amaç fonksiyon değeridir. Bu nedenle, daha iyi besin kaynakları gözlemci arı aşamasında seçilip güncellenecektir. Bir gözlemci arı, bir gıda kaynağı seçtikten ve Denklem (2.21) 'yi uygulayarak yeni bir besin kaynağı (υ_i) ürettikten sonra, işçi arı aşamasına benzer şekilde χ_i ve υ_i arasında açgözlü bir seçim gerçekleştirir.

İşçi arı aşamasındaki konum ve hız güncelleme denklemleri gözcü arı aşamasındaki ile aynıdır. Aralarındaki fark, gözcü arı aşamasında her besin kaynağı güncelleme sürecine girerken, işçi arı aşamasında olasılık değerlerine göre seçilme şansına sahip çözümler kullanılarak güncelleme yapılır.

Kâşif arı aşaması: işçi ve gözcü arıların iyileştiremeyecekleri bir besin kaynağını ziyaret etmeleri durumunda; önceden belirlenmiş bir deneme sayısı (trial) boyunca besin kaynağı iyileştirilemiyorsa, o kaynak terk edilir. Onunla ilişkili olan arı kâşif arıya dönüşür. Bundan sonra, bu kâşif arı Denklem (2.19) 'u kullanarak yeni bir çözüm üretir ve işçi arıya dönüşür. Yukarıdaki açıklamalara dayanarak, ABC algoritmasının sözde kodu Şekil 2.9 'da verilmiştir.

ABC Algoritması

1: parametreleri ayarla: maksimum yineleme, SN, limit (denemelerin maksimum sayısı)
2: Gıda kaynakları popülasyonunu $\boldsymbol{\chi_i}$ burada i=1,..., SN olarak başlat ve niteliklerini değerlendir
3: for her besin kaynağı i **do**, $\boldsymbol{trail_i}$=0 ayarla end, burada i=1,…, SN
4: iterasyonu ayarla /* ABC *iterasyonun sayısı* */
5: Repeat
6: ***/* işçi arı aşaması */***
7: **for** i=1 to SN **do**
8: denklemi (2.21)'yi kullanarak işçi arı aşaması için $\boldsymbol{v_i}$ yeni bir besin kaynağı oluştur ve kalitesini değerlendir
9: daha iyi birini seçmek için $\boldsymbol{\chi_i}$ ve $\boldsymbol{v_i}$ arasında açgözlü seçim mekanizması uygula
10: **eğer** gıda kaynağı $\boldsymbol{\chi_i}$ düzelmediyse o zaman $\boldsymbol{trail_i}$= $\boldsymbol{trail_i}$+1, **yoksa** $\boldsymbol{trail_i}$ = 0
11: **End**
12: besin kaynakları için, uygunluk değerlerine dayanarak, $\boldsymbol{\mathcal{P}_i}$'nin olasılık değerlerini denkleme (2.22) kullanarak hesapla
13: ***/* Gözcü arı aşaması */***
14: t = 0, i = 0
15: **Repeat**
16: **if** random < $\boldsymbol{\mathcal{P}_i}$
17: denklem (2.21)'yi kullanarak gözcü arı aşaması için $\boldsymbol{v_i}$ yeni bir besin kaynağı oluştur ve uygunluğunu değerlendir.
18: daha iyi birini seçmek için $\boldsymbol{\chi_i}$ ve $\boldsymbol{v_i}$ arasında açgözlü seçim mekanizma uygula
19: **eğer** gıda kaynağı $\boldsymbol{\chi_i}$ düzelmediyse o zaman $\boldsymbol{trail_i}$= $\boldsymbol{trail_i}$+1**, yoksa** $\boldsymbol{trail_i}$ = 0
20: t = t + 1
21: **End**
22: **Until** (t=SN)
23: ***/* Kâşif arı aşaması */***
24: **if** max($\boldsymbol{trail_i}$) > limit **then**
25: denklem (2.20)'i kullanarak $\boldsymbol{\chi_i}$'yi yeni rastgele oluşturulmuş bir gıda kaynağı ile değiştir
26: **End**
27: Şimdiye kadar bulunan en iyi besin kaynağını sakla
28: iterasyon = iterasyon + 1
29: **Until** (durdurma ölçütleri sağlanıncaya kadar)

Şekil 2.9. ABC algoritmasının sözde-kodu

2.6. Bölüm Değerlendirmesi

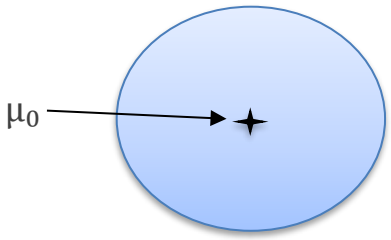
Bu bölümde, yapay sinir ağlarının yapısı ve temel kavramı açıklanmaktadır. Yapay sinir ağların yapıları ve modelleri ile birlikte eğitim teknikler anlatıldı. Yapay sinir ağlarının eğitilmesinde kullanılan teknikler veya yaklaşımlar ele alınmış. Yapay sinir ağı yapısı için yaygınca kullanılan tekniklere yer verilmiştir. Bu amaçla, türeve dayalı algoritmalardan YSA Eğitiminde yaygın olarak kullanılan Geri Yayılım ve Levenberg Marquardt öğrenme algoritmaları anlatılmıştır. Ayrıca, Optimizasyon algoritmaları kavramı tanımlanmış ve bu alanda özellikle optimizasyon problemlerinin çözümüne yönelik olarak geliştirilen Parçacık Sürü Optimizasyonu ve Yapay Arı Kolonisi Optimizasyonu algoritma detaylı bir şekilde anlatıldı. Bu algoritmalar için önerilmiş çok fazla sayıda modifikasyon çalışması yayınlanmıştır. Bu nedenle çalışma kapsamında ele alınan algoritmalar önerildikleri orijinal versiyonları temel alınarak açıklandı.

3. GİRDAP ARAMA ALGORİTMASI (VS: Vortex Search)

Girdap arama algoritması (VS) ilk olarak 2015 yılında, tek-amaçlı optimizasyon problemlerinin çözümü için geliştirilmiş yerli bir algoritmadır (Dogan ve Olmez, 2015). VS algoritması, popülasyon-temelli meta-sezgisel bir algoritmadır. Bu bağlamda; rasgele arama (RS) ve desen arama (PS) algoritmalarını içeren arama algoritmaları ailesinde incelenebilir. Algoritma, karıştırılan sıvıların girdap akışıyla oluşturulan girdap deseninden esinlenerek arama uzayını iterasyonlar boyunca daraltır. Böylece global optimuma ulaşmayı hedefler. Ayrıca VS algoritması çok az sayıda kullanıcı tanımlı parametre içerir. VS algoritmasının önerildiği çalışmada; 50 kıyaslama fonksiyonu ile test edilmiş ve sonuçlar hem tek çözüm tabanlı (Benzetimsel tavlama ve Desen Arama) hem de popülasyon tabanlı (PSO2011 ve ABC) algoritmalarla karşılaştırılmıştır. Bu karşılaştırılmanın sonucunda, VS algoritmasının oldukça başarılı bir performansa sahip olduğu ortaya koyulmuştur. Bu bölümde Girdap arama algoritması yapısı açıklandıktan sonra, VS algoritmanın özellikleri hakkında bilgi verilmekte ve son olarak bölüm değerlendirmesi yapılmaktadır.

3.1. Algoritmanın Yapısı

İki boyutlu bir arama uzayında bir girdap deseni iç içe geçmiş birçok dairelerle modellemektedir. Burada ilk olarak, girdabın dıştaki en büyük çemberinin başlangıç merkezi μ0 problemin tüm boyutları için Denklem (3.1) kullanılarak hesaplanır. Böylece çember arama alanına ortalanmış olur Şekil 3.1görüldüğü gibi.



Şekil 3.1. Girdap çemberinin başlangıç merkezi.

$$\mu_0 = \frac{\text{üst sınır} + \text{alt sınır}}{2} \tag{3.1}$$

Aday Çözüm Üretme: Gauss dağılımı kullanarak d-boyutlu uzayda başlangıç merkezinin etrafında rasgele değerler üretilir. $C_t(s)$ komşu çözüm sayısıdır. T yineleme endeksini temsil eder ve başlangıçta t = 0'dır. Burada, $C_0(s) = \{s_1, s_2, \ldots, s_k\}$ k = $1, 2, .., n$ çözümleri temsil eder ve n aday çözümlerin toplam sayısını temsil eder. Denklem (3.2) 'de çok değişkenli Gauss dağılımının genel formu verilmektedir.

$$p(x|\mu, \Sigma) = \frac{1}{\sqrt{(2\Pi)^d|\Sigma|}} \exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\} \quad (3.2)$$

Burada d boyutu temsil eder. x rasgele bir değişkenin dx1 boyutlu vektörüdür, μ merkez ve Σ kovaryans matrisidir. Σ değerlerinin diyagonal elemanları (varyansları) eşitse ve diyagonal olmayan elemanlar (kovaryans) sıfır (korelasyonsuz) ise dağılımın şekli küresel olacaktır. Dolayısıyla, Σ değeri, Denklem (3.3) kullanılarak sıfır kovaryanslı eşit varyansları kullanarak hesaplanabilir.

$$\Sigma = \sigma^2.[I]_{dxd} \quad (3.3)$$

Denklem (3.3) 'te, σ^2 dağılımın varyansını ve I, dxd eşdeğer matrisini temsil eder. Dağılımın başlangıçtaki standart sapması (σ^2), Denklem (3.4) kullanılarak hesaplanabilir.

$$\sigma_0 = \frac{\text{maximum(üst sınır)} - \text{minimum(alt sınır)}}{2} \quad (3.4)$$

Burada σ_0, dış çemberin başlangıç yarıçapı (r0) olarak da düşünülebilir. İlk aşamalarda küçük değişimlere gerek duyulduğundan, r0 büyük bir değer olarak seçilir. Böylece, ilk adımda dış çember tarafından arama alanının tam olarak kapsanması sağlanır. Bu süreç, eldeki sorunun kuşbakışı bir görünümünü sağlar.

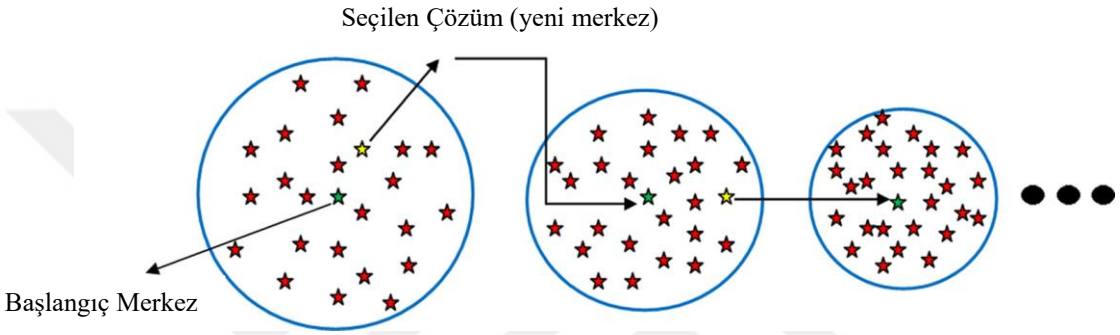
Mevcut Çözümün Değiştirilmesi: Seçim aşamasında, mevcut çember merkezi μ0'ı değiştirmek için C0(s) 'dan bir çözüm (en iyisi) s'∈C0(S) seçilir ve hafızaya alınır. Seçim aşamasından önce, aday çözümlerin arama sınırları içinde olması sağlanmalıdır. Bu amaçla, sınırları aşan çözümler, aşağıdaki Denklem (3.5) 'de olduğu gibi sınırlara çekilmiştir.

$$s_k^i = \begin{cases} \text{rand.}(\text{üst sınır}^i - \text{alt sınır}^i) + \text{alt sınır}^i & s_k^i < \text{alt sınır}^i \\ s_k^i & \text{alt sınır}^i \leqslant s_k^i \leqslant \text{üst sınır}^i \\ \text{rand.}(\text{üst sınır}^i - \text{alt sınır}^i) + \text{alt sınır}^i & \text{alt sınır}^i,\ s_k^i > \text{üst sınır}^i \end{cases} \quad (3.5)$$

Burada $k = 1,2,\ldots,n$ ve $i = 1,2,,\ldots,d$ ve rand [0,1] aralığında rasgele üretilen reel sayılardır. Daha sonra, hafızaya getirilmiş en iyi çözüm S_0, ikinci dairenin (içteki) merkezi

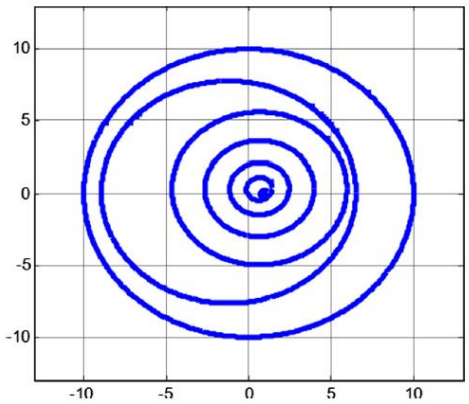
olarak atanır. İkinci adımın üretim aşamasında, bu yeni dairenin etkin yarıçapı (r_1) azaltılır. Daha sonra yeni merkez etrafında yeni çözümler kümesi $C_1(s)$ oluşturulur.

İkinci adımın seçim aşamasında, yeni çözüm kümesi $C_1(s)$ bir çözüm $s' \in C_0(s)$ seçmek üzere değerlendirilir. Seçilen çözüm şu ana kadar bulunan en iyi çözümden daha iyiyse, bu çözüm yeni en iyi çözüm olarak atanır ve hafızaya alınır. Daha sonra, üçüncü dairenin merkezi şimdiye kadar bulunan en iyi çözümü tutar. Bu işlem, sonlandırma koşulları sağlanıncaya kadar devam eder. İşlemin açıklayıcı bir çizimi Şekil 3.2 'de verilmektedir. Şekil 3.5 'de algoritmanın sözde kodu açıklanmaktadır.



Şekil 3.2. VS algoritmasının arama sürecinin örnek bir çizimi (Dogan ve Olmez, 2015)

Bu şekilde, algoritma sona erdirildiğinde, oluşan desen, en küçük dairenin merkezinin algoritma tarafından bulunan optimum nokta olduğu bir girdap benzeri yapı olarak görülür. Temsili bir desen, üst ve alt sınırların [-10, 10] aralığı arasında olduğu iki boyutlu bir optimizasyon problemi için Şekil 3.3 'de gösterilmiştir.



Şekil 3.3. VS Algoritmanın Arama İşleminden Sonra Girdap Benzeri Bir Yapı

Yarıçapın Azaltma İşlemi: VS algoritmasında, ters tamamlanmamış gama fonksiyonu, her iterasyon geçişinde yarıçapın değerini azaltmak için kullanılır. Denklem (3.6) verilen tamamlanmamış gama fonksiyonu, olasılık teorisinde, özellikle ki-kare dağılımını içeren uygulamalarda kullanılır (Andrews, 1997).

$$\gamma(x,a) = \int_0^x e^{-t}t^{a-1}dt \tag{3.6}$$

Burada a > 0 şekil parametresi olarak bilinir ve $x \geq 0$ rasgele bir değişkendir. Tamamlanmamış gamma fonksiyonu ile birlikte Denklem (3.7) ile verilen tamamlayıcı Γ(x, a) genellikle fonksiyona eklenir.

$$\Gamma(x,a)=\int_x^{\infty} e^{-t}t^{a-1}dt \qquad a>0 \tag{3.7}$$

Böylece,

$$\gamma(x,a) + \Gamma(x,a)= \Gamma(a) \tag{3.8}$$

Burada Γ(a) gama fonksiyonu olarak bilinir.

VS Algoritması

Girdi: Başlangıç merkezi μ0, **Denklem(3.1)** kullanarak hesaplanır.
Çıktı: Bulunan en iyi çözüm $\boldsymbol{Sbest}$

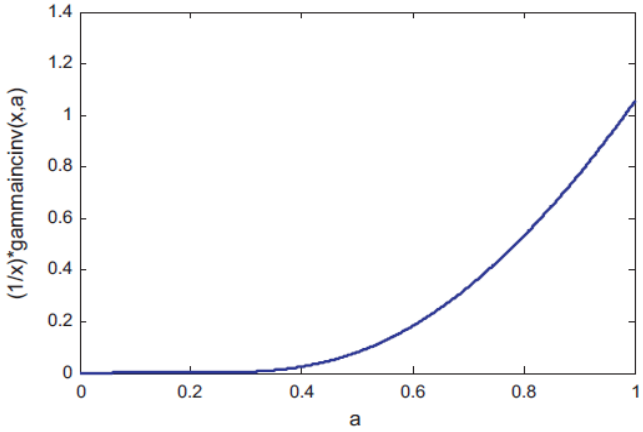
Başlangıç yarıçapı r0 (veya standart sapma, σ0) denklem (3.4) kullanılarak hesaplanır.
Şimdiye kadar bulunan en iyi çözümün uygunluğu $\boldsymbol{F(Sbest) = inf}$
t=0;
Repeat
Standart sapma (yarıçap) rt ile merkez μt etrafında Gauss dağılımı kullanarak aday çözümler üretir
(Ct (s)) Üret;
Ct (s) değerlerini, denklem (3.5)'deki gibi sınırlara kaydırır
Geçerli merkezin yerini alacak μt en iyi çözümü $\boldsymbol{Ct(s)}$ den seçilir
s'=seç(Ct(s));
If $\boldsymbol{f(s') < f(Sbest)}$
$\boldsymbol{Sbest = s'}$
$\boldsymbol{f(Sbest) = f(s')}$
Else
Şimdiye kadar en iyi çözümü tut $\boldsymbol{Sbest}$
End
Merkez her zaman şimdiye kadar bulunan en iyi çözüme kaydırılmıştır.
$\boldsymbol{\mu_{t+1} = Sbest}$
Sonraki yineleme için standart sapmayı (yarıçap) azaltma
$\boldsymbol{r_{t+1} = Azaltılmış\ r_t}$
$\boldsymbol{t = t + 1}$
Until Sonlanma ölçütü

Şekil 3.4. VS algoritmasının sözde-kodu

Şekil 3.5'de, ters tamamlanmamış gama fonksiyonu x = 0.1 ve a Є [0,1] için çizilmiştir. Burada, ters tamamlanmamış gama fonksiyonunun a parametresi, aramanın çözünürlüğünü tanımlar. Belirli bir adım boyutunda [0,1] aralığı içinde eşit olarak örnekler çıkartarak, aramanın çözünürlüğü ayarlanabilir. Bu amaçla, her yinelemede, a'nın bir değeri aşağıdaki denklemi (3.9) kullanarak hesaplanır:

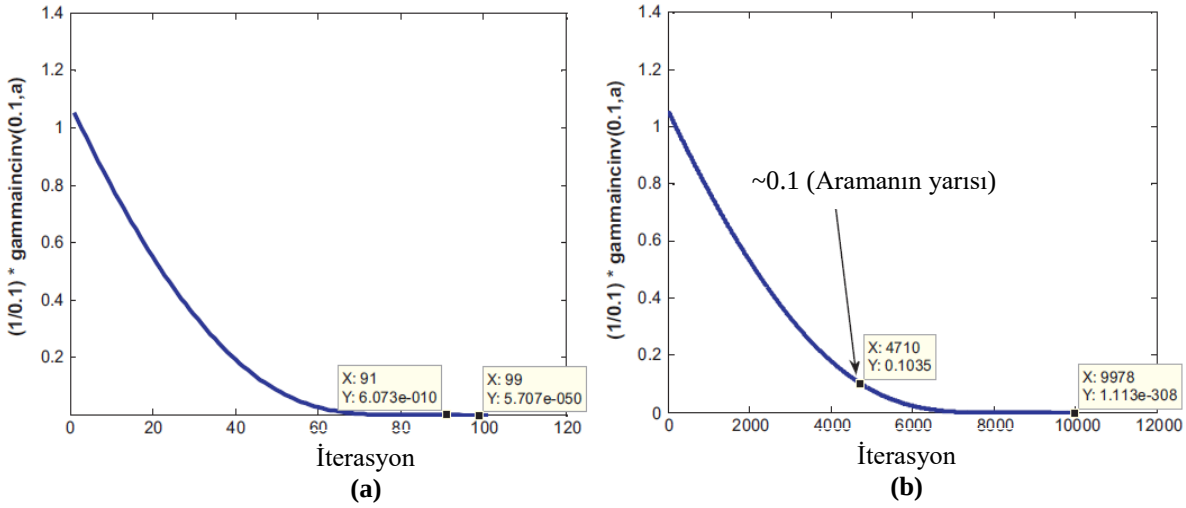
$$a_t = a_0 - t/MaxItr \tag{3.9}$$

Burada a0, birinci iterasyonda arama alanının tam olarak kapsanmasını sağlamak için a0 = 1 olarak seçilir, t iterasyon sayısı ve MaxItr, yinelemelerin maksimum sayısını temsil eder.



Şekil 3.5. (1/x) gammaincinv(x,a) burada x = 0.1 ve a Є [0,1].

Böylece, MaxItr'in farklı değerleri için, a değerlerinin örnekleme oranı değiştirilir ve böylece aramanın çözünürlüğü değiştirilir.

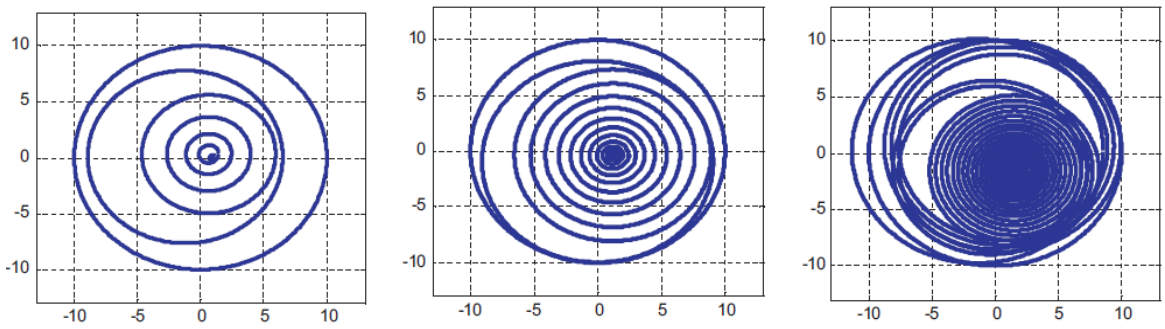


Şekil 3.6. (1/0.1) gammaincinv(0.1,a) for (a) MaxItr = 100 (b) MaxItr = 10,000. (Dogan ve Olmez, 2015)

Şekil 3.6 (a) ve (b) 'den görüldüğü gibi, iterasyon sayısı arttıkça adım büyüklüğü azalır ve fonksiyonun değeri önemli ölçüde azalır. Örneğin, Şekil 3.6 (b)'de, 9978 iterasyon için işlevin sonuç değeri 1.113e-308 gibi çok küçük bir sayıdır. Bununla birlikte, iterasyon sayısının yarısına erişilinceye kadar, fonksiyon hemen hemen doğrusal davranır. Bu davranış, fonksiyonun iki ayrı bölgede analiz edilmesine izin verir (Dogan ve Olmez, 2015).

Ters tamamlanmamış gama fonksiyonu, başarılı bir aramadan beklenen hedefleri karşılar. Yarıçapı ayarlamak için böyle bir yöntem kullanmanın dezavantajı yakınsama hızının yineleme sayısına bağlı olmasıdır. Yineleme sayısı arttıkça adım büyüklüğü azalır ve arama çözünürlük artar. İlgili arama alanı derinlemesine araştırıldığından bazı durumlarda bu avantaj olarak görülebilir.

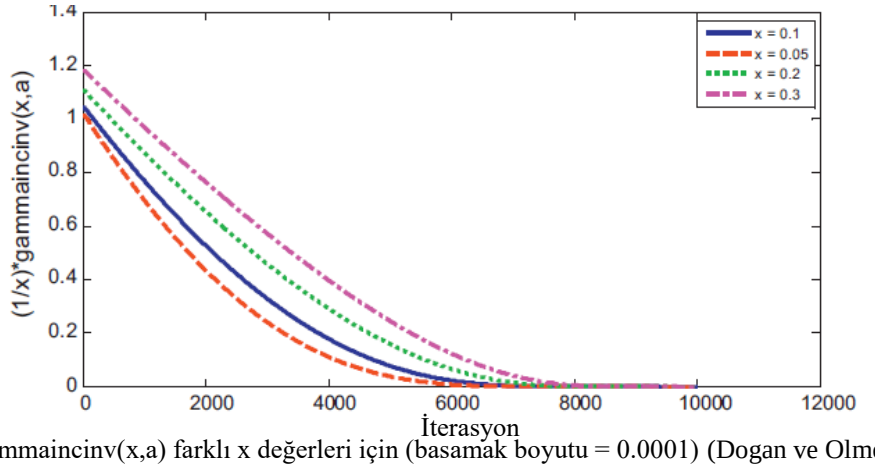
Bununla birlikte, 500.000 ve üstü bir iterasyon sayısına denk gelecek küçük bir adım boyutu yakınsama hızını düşürür. Şekil 3.7 (a), Şekil 3.7 (b) ve Şekil 3.7 (c)'de, adım boyutunun arama sınırları üzerindeki etkisi görülmektedir.



(a) adım büyüklüğü 0.1 (b) adım büyüklüğü 0.05 (c) adım büyüklüğü 0.02

Şekil 3.7. Aramanın çözünürlüğü adım boyutunda bir azalma ile artar (Dogan ve Olmez, 2015)

VS algoritması oldukça basittir itersayon sayısı, popülasyon sayısı, problemin üst ve alt sınırları ve boyutları dışında herhangi bir ek parametre gerektirmez. Olabilecek tek parametre x değeridir. Bununla birlikte, x değeri sabit bir değer olarak da seçilebilir. Şekil 3.8'de, (1 / x) gammaincinv (x, a), belirli bir adım boyutunun farklı x değerleri için çizilmiştir.



Şekil 3.8. (1/x) gammaincinv(x,a) farklı x değerleri için (basamak boyutu = 0.0001) (Dogan ve Olmez, 2015)

3.2. Bölüm Değerlendirmesi

Bu bölümde, popülasyon-temelli meta-sezgisel girdap arama algoritması detaylı bir şekilde açıklandı. VS algoritması özellikle tek-amaçlı optimizasyon problemlerinin çözümüne yönelik olarak geliştirilmiş bir arama tekniğidir. Bu algoritma ile arama uzayı içinde bir merkez belirlenir ve bu merkez noktasından belirlenen bir yarıçapı kapsayan bir alanda arama işlemi gerçekleştirilir. Algoritma için kilit nokta kullanılan yarıçapın iterasyonlar boyunca azaltılması ile daraltılan arama uzayında global optimal değere ulaşılmasını sağlamaktır. Yarıçapı azaltma işlemi RS algoritmalarında da kullanılan bir adaptif adım boyutu ayarlama işlemi olarak düşünülebilir. Bu işlem, algoritmanın ilk adımlarda keşfedici bir şekilde davranmasına ve sonraki adımlarda yakınsamasını hızlandıracak şekilde çalışmasına imkân verir. Bu tür bir işlemi gerçekleştirmek için, arama işlemi sırasında yarıçapın değeri doğru şekilde ayarlanmalıdır. Ayarlamanın yapıldığı bu yöntem, algoritmanın başarısı için kritik öneme sahiptir. Yalnızca en iyi çözümü hafızaya alması bakımından VS algoritması zayıf belleklidir. Zayıf belleğin kullanımı, tek çözüm tabanlı meta-sezgisellerde yeni değildir.

Girdap arama yeni bir algoritma olmasından dolayı literatürde henüz çok az sayıda çalışma yer almaktadır. Bu algoritma için az sayıda modifikasyon çalışması önerilmiştir. Performans iyileştirme amacıyla önerilen bu modifikasyonlar çoğunlukla belirli bir uygulama alanına yönelik olduğu için genel olarak algoritmanın orijinali ile her zaman rekabet edebilir düzeyde olamamaktadır. Bu nedenle çalışma kapsamında algoritmanın orijinal versiyonu temel alınarak açıklanmıştır.

4. OPTİMİZASYON ALGORİTMALARI İLE YSA EĞİTİMİ

Yapay sinir ağlarının kapsamlı uygulama alanında örüntü algılama; sınıflandırma, konuşma tanıma, teşhis problemleri, finans, tıp, sinyal işleme, robotik kontrol ve bilgisayar alanlarındaki çalışmalar ile en yaygın uygulama alanı olarak karşımıza çıkmaktadır. Çok farklı sinir ağı sınıflandırıcısı arasında, Çok Katmanlı İleri Beslemeli Algılayıcılar (ÇKA), evrensel yaklaşım yetenekleri ile sınıflandırma amaçlı kullanılan yapılardandır. Sinir ağlarının başarısı büyük ölçüde eğitim algoritması ve eğitimde kullanılan parametrelerin seçimine bağlıdır. YSA'da, uygun eğitim algoritması, aktivasyon fonksiyonu ve eğitim verileri tabanlı ağırlık ve eşik değerleri bulmaya yönelik ağ optimizasyonu üzerine çalışmaktadırlar. Bu yaklaşımların çoğunda topoloji tipleri ve aktivasyon fonksiyonları sabit tutulur ve ağırlıkların olası değerleri ayarlanmaya çalışılarak ağ eğitilir. Ağ yapısı sabitleştirildiğinde, örneğin gizli katman sayısı ve her bir gizli katmandaki nöron sayısı belirlendiğinde, ağ optimal ağırlık setlerini elde edene kadar ağırlık ayarlamaları yapmaya devam eder. YSA'lar kendilerine örnekler gösterildikçe, bu ağırlık değerlerini değiştirirler. Amaç, ağa gösterilen örnekler için doğru çıktıları üretecek ağırlık değerlerini bulmaktır (Lang ve ark., 1990).

Genellikle, karşılaştırma belirli bir hata oranına erişmek için gereken eğitimin hesaplama süresini bir performans ölçütü kullanarak yapılabilir. Eşleştirme hatası, veri setinde önceden belirlenen çıktı değeri ile belli bir giriş değerine karşılık üretilen çıktı değeri arasındaki farktır. Eğitim sürecinde girdilerin bulunduğu girdi veri seti ile bu girdilere karşılık gelen çıktılardan oluşan çıktı veri seti arasında eşleme yapılması beklenir. Yapay sinir ağlarının eğitilmesi nöronlar arası bağlantı ağırlıklarının uygun değerlere atanması sürecidir. Bu genelleştirme özelliğine kavuşması işlemine, "ağın öğrenmesi" denir. Yapay sinir ağının öğrenme sürecinde temel olarak üç adım bulunmaktadır; Çıktıları hesaplamak, Çıktıları hedef çıktılarla karşılaştırmak ve hatayı hesaplanmak son olarak Ağırlıkları değiştirerek süreci tekrarlamak.

İleri beslemeli çok katmanlı algılayıcı ağlar doğrusal olmayan modellemedeki başarısı ve genelleştirme yeteneği, gerçek zamanlı işlem yapabilme, adaptasyon ile kendini yönetebilme, hata toleransı gibi özellikleri sayesinde araştırma alan birçok uygulamada tercih edilmektedirler. Uygun ağ yapısının ve optimum ağırlık değerlerinin bulunması yapay sinir ağ tasarımının başarısı, eğitim algoritmasına ve eğitimde kullanılan özellik seçeneklerine bağlıdır (Birattari, 2005).

Literatürde yapay sinir ağlarının eğitimi için çok çeşitli çalışmalar yapılmıştır ve çeşitli alanlarda uygulanmıştır. Bunlardan birisi, Yapay Arı Kolonisi (ABC) algoritmasını YSA eğitimi için önermiştir. Performansı geleneksel ve modern optimizasyon yöntemleriyle karşılaştırılarak analiz edilmiştir. Çok katmanlı ileri beslemeli algılayıcı ağları, XOR, 3-Bit Parite ve Encoder-Decoder test problemleri ve UCI makine öğrenme deposundan alınan 9 farklı sınıflandırma test probleminin çözülmesi amacıyla tasarlanmış ve eğitiminde de ABC algoritmasının performansı türeve dayalı algoritmalardan eğim düşme ve Levenberg-Marquardt geri yayılım algoritmalarının, evrimsel algoritmalardan Genetik Algoritma, Parçacık Sürü Optimizasyon ve Diferansiyel Gelişim algoritmalarının performansları ile karşılaştırılmıştır. Elde edilen sonuçlarda ABC algoritmasının ÇKA yapay sinir ağlarının sınıflandırma amaçlı eğitilmesinde üstün bir performans sergilediği sonucuna varılmıştır (Öztürk ve Karaboğa, 2011).

Bu çalışmanın amacı; yapay sinir ağlarının eğitiminde daha önce literatürde denenmemiş olan Girdap Arama Algoritmasının kullanılması ve algoritmanın YSA eğitimindeki performansının araştırılmasıdır. Bu sebeple, tezde bir sınıflandırma problemi için en uygun mimariyi bulmak ya da ulaşılabilecek en düşük hatayı bulmak ana hedef olarak gösterilmemiştir. Amaç VS algoritmasının ağ eğitiminde diğer algoritmalar kadar rekabetçi olduğunu göstermek ve başarısını tartışmaktır.

Önerilen yaklaşım alternatif bir eğitim yöntemi olarak geliştirilmiş ve performansı Yapay Arı Kolonisi ve Parçacık Sürüsü Optimizasyonu ile karşılaştırılmıştır. Girdap arama algoritmasının performansını değerlendirmek için sınıflandırma problemi olarak beş veri seti kullanılmıştır.

4.1. Sınıflandırma Problemleri

Çalışmada kullanılan tüm veri setleri, test problemlerinin verilerini düzenleyen ve saklayan UCI veri-tabanından seçildi. Diyabet Problemi (Pima Indians Diabetes), Kanser Problemi (Breast Cancer Data Set), Tiroit Problemi (Thyroid Disease), Şarap Problemi (Wine Recognition Data Set) ve İris Problemi (Iris Plants Data Set) için veri setleri UCI'den alınarak önerilen VS-tabanlı eğitim yaklaşımı ve karşılaştırma amacıyla kullanılan diğer meta-sezgisel teknikler aynı koşullar altında bu verilerin sınıflandırılması için kullanıldı.

ÇKA ağının eğitimi için genellikle toplam hata fonksiyonunun minimizasyonu kullanılır. Bu çalışmada da kullanılmış olan minimum kare hata (*MSE*) fonksiyonu hata

fonksiyonu olarak uygulamalarda sıkça kullanılmaktadır. *MSE*'nin formülü matematiksel olarak Denklem (4.1) ile ifade edilir.

$$MSE(\vec{\omega}(t)) = \frac{1}{N}\sum_{j=1}^{N}\sum_{k=1}^{P}(d_k - o_k)^2 \quad (4.1)$$

Burada, $\vec{\omega}(t)$ t. iterasyondaki bağlantı ağırlıkları; d_k istenen çıktı değeri ve o_k üretilen çıktı değerini o anki üretilen değerini temsil etmektedir; P çıktı nöron sayısını ve N örnek sayısını göstermektedir.

Standart Hata Yüzdesi (*SEP*) olarak isimlendirilen ağ hata fonksiyonu minimize edilerek kullanılabilir. SEP hata fonksiyonu Denklem 4.2'de verilmiştir.

$$SEP = 100 * \frac{O_{max}-O_{min}}{N.P}\sum_{j=1}^{N}\sum_{k=1}^{P}(d_k - o_k)^2 \quad (4.2)$$

Denklemde O_{max} ve O_{min} sırasıyla çıktı nöronun çıktı sinyalinin maksimum ve minimum değerlerini göstermektedir. Veri setlerinde çıkışlar ait oldukları sınıfı temsil ederken 1 değerini, diğer durumlarda ise 0 değerini aldıklarından dolayı sırasıyla O_{max} 1 değerine ve O_{min} 0 değerine eşitlenmiştir (Öztürk ve Karaboğa, 2011).

Sınıflandırma yaklaşımının güvenilirliğini ve geçerliliğini test etmek amacıyla performans değerlendirmesi üç aşamada yapıldı. Birinci aşamada; uygulamalarda her problemin %100'ü eğitim verisi olarak kullanıldı. İkinci aşamada; uygulamalarda her problemin %75'i eğitim seti olarak geri kalan %25'i ise test verisi olarak kullanıldı. Üçüncü aşamada ise yaygın olarak kullanılan yöntem çapraz doğrulama (cross-validation) yöntemi uygulanarak sonuçlar alındı. Tüm veri setlerinin öznitelikleri eğitim işleminden önce ön işlem olarak [0-1] aralığına çekilerek normalize edildi. Seçilen topolojiler hem çok-çıkışlı hem de Tek çıkışlı sınıflandırma problemleri üzerinde uygulandı. Kullanılan veri setlerinin orijini, öznitelikleri, örnek sayıları ve kaç sınıftan oluştukları gibi bilgiler aşağıda sırasıyla açıklanmaktadır.

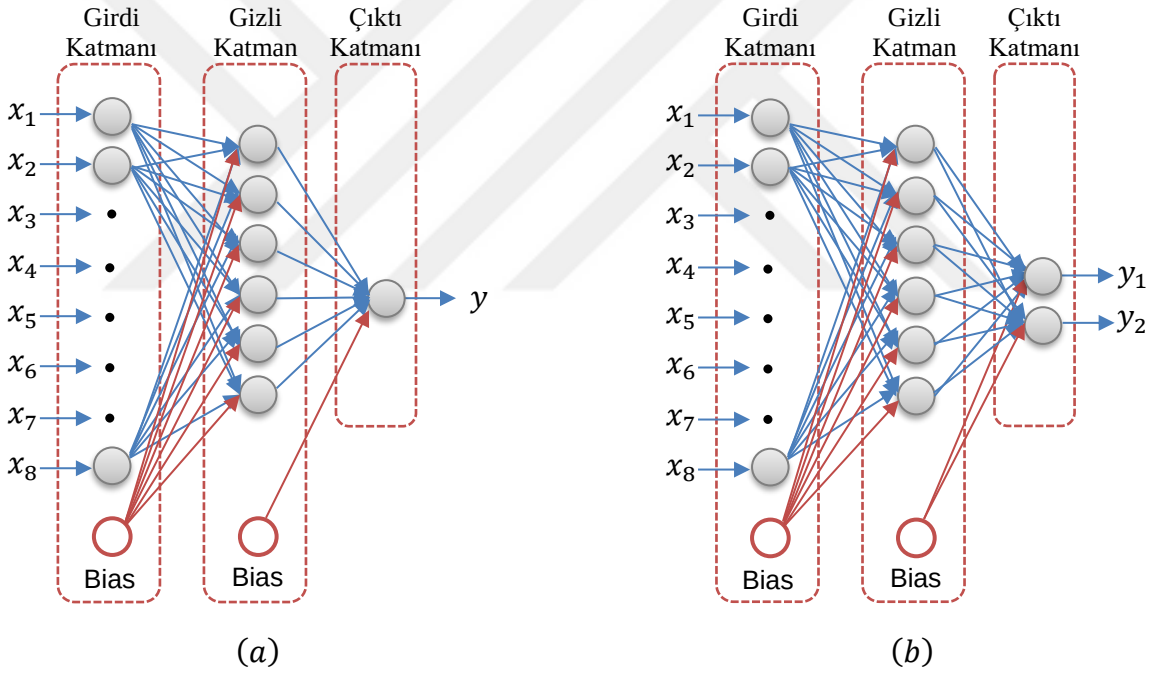
4.1.1. Diyabet Problemi (Pima Indians Diabetes)

Pima Kızılderilileri Diyabet Veri Seti, Johns Hopkins Üniversitesi'nin Uygulamalı Fizik Laboratuvarı'nda oluşturulmuştur. Bu veri kümesine kaydedilen tüm hastalar en az 21 yaşında olan kadınlardır. Veri seti 8 girdi ve 2 çıktıdan oluşur ve 768 örnek içerir (Frank ve Asuncion, 2010).

Veri setinin öznitelikleri aşağıda sıralanmıştır.

x1 = Hamile kalma sayısı [0 - 17]
x2 = Plazma glikoz konsantrasyonu oral glikoz tolerans testinde 2 saat [0 - 199]
x3 = Diyastole tansiyon (mm Hg) [0 - 122]
x4 = Triceps deri kat kalınlığı (mm) [0 - 99]
x5 = 2 saatlik serum insülini (mu U / ml) [0 - 846]
x6 = Vücut kitle indeksi (ağırlık kg / (m yükseklik) ^ 2) [0 - 67.1]
x7 = Diyabet soyağacı fonksiyonu [0.078 - 2.42]
x8 = Yaş (yıl) [21 - 81]

Sınıf etiketleri 268 örnek için diyabet pozitif olarak test edilmiştir $y_1 = (Sınıf\ 1)$. Kalan 500 örnek için ise diyabet negatif olarak test edilmiştir $y_2 = (Sınıf\ 2)$. Diyabet problemi için ÇKA ağı çıkış nöron sayıları değiştirilerek iki farklı topoloji üzerinden ayrı ayrı çalıştırılarak eğitildi. Diyabet veri seti için kullanılan farklı topolojiler şematik olarak Şekil 4.1.'den görülmektedir.



Şekil 4.1. Diyabet probleminin (a) ÇKA ağının tek çıkışlı topolojisi (b) ÇKA ağının çok çıkışlı topolojisi.

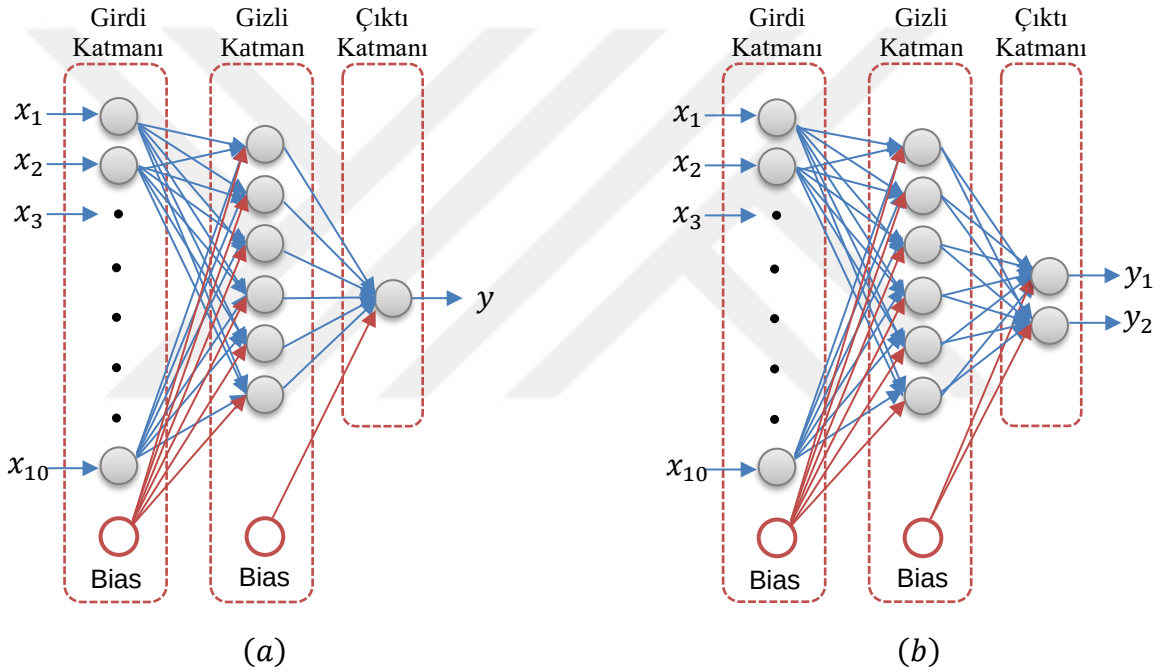
4.1.2. Kanser Problemi (Breast Cancer Data Set)

Göğüs kanseri veri seti, Dr. William H. Wolberg'den Madison Üniversitesi Wisconsin Hastanelerinden elde edilmiştir. Veri seti 10 girdi, 2 çıktıdan oluşur ve toplam 699 örnek içerir. Veri setinin öznitelikleri aşağıda sıralanmıştır.

x1 = Örnek kod numarası
x2 = Küme Kalınlığı [1 - 10]
x3 = Hücre Boyutunun Eş biçimliliği [1 - 10]
x4 = Hücre Şekillerinin Düzensizliği [1 - 10]

x5 = Marjinal Bağlılığı [1 - 10]
x6 = Tek Epitelyum Hücre Boyutu [1 - 10]
x7 = Çıplak Çekirdekler [1 - 10]
x8 = Mülayim Kromatin [1 - 10]
x9 = Normal Nükleonla [1 - 10]
x10 = Mitozlar [1 - 10]

Sınıf etiketleri 458 veri için kanser iyi huylu tespit edilmiştir $y_1 = (Sınıf\ 1: \%65.5)$. Kalan 241 veri için ise kanser kötü huylu tespit edilmiştir $y_2 = (Sınıf\ 2: \%\ 34.5)$ (Frank ve Asuncion, 2010). Kanser problemi için ÇKA ağı çıkış nöron sayıları değiştirilerek iki farklı topoloji üzerinden ayrı ayrı çalıştırılarak eğitildi. Kanser veri seti için kullanılan farklı topolojiler şematik olarak Şekil 4.2.'den görülmektedir.



Şekil 4.2. Kanser probleminin (a) ÇKA ağının tek çıkışlı topolojisi (b) ÇKA ağının çok çıkışlı topolojisi

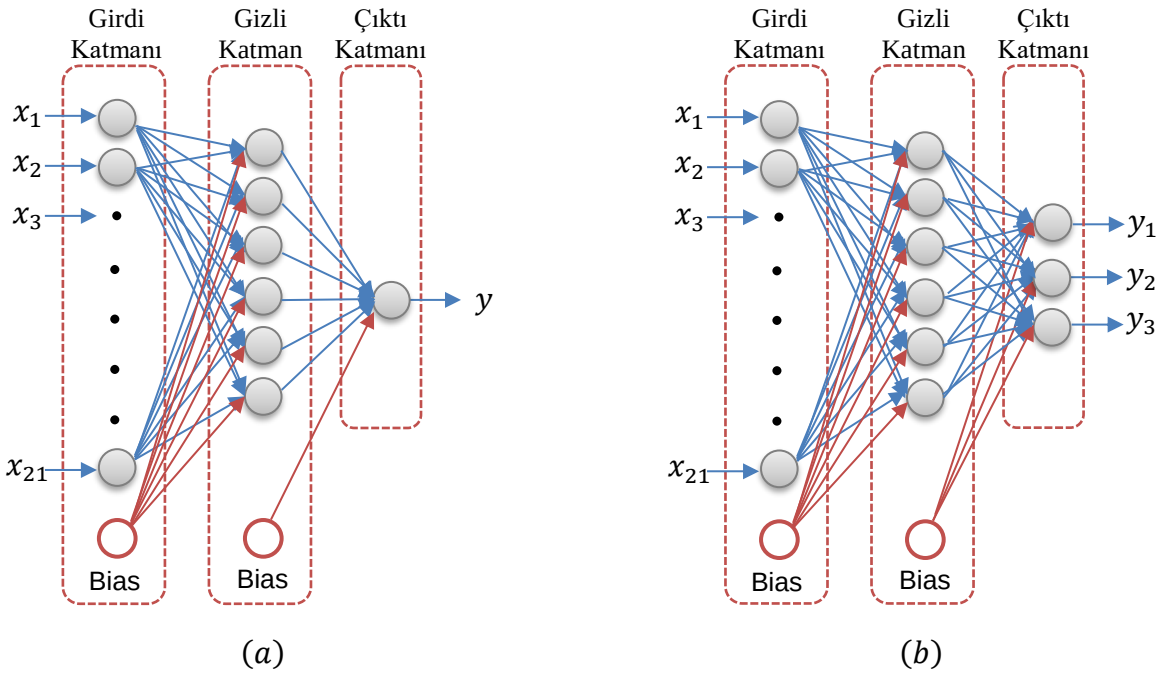
4.1.3.Tiroit Problemi (Thyroid Disease)

Thyroid Disease veri seti New South Wales Enstitüsü Sidney-Avustralya, tarafından sağlanan tiroit hastalıkları kayıtlarıdır. Tiroit hastasının ötiroidizm, hipotiroidizm veya hipertiroidi olup olmadığı teşhis edilir. Veri seti 21 girdi ve 3 çıktıdan oluşur. 7200 örnek içerir. Veri setinin öznitelikleri aşağıda sıralanmıştır.

x1 = Yaş [0.01 - 0.97]
x2 = Cinsiyet [0 - 1]
x3 = Tiroksin tamsayısı [0 - 1]
x4 = Tiroksin tamsayısına ilişkin sorgulama [0 - 1]

x5 = Anti tiroit İlacı tamsayısı [0 - 1]
x6 = Hasta tamsayısı [0 - 1]
x7 = Hamile tamsayısı [0 - 1]
x8 = Tiroit Cerrahisi tamsayı [0 - 1]
x9 = I 131 tedavi tamsayısı [0 - 1]
x10 = Hipotiroidizm tamsayı [0 - 1]
x11 = Hipertiroidi tamsayı [0 - 1]
x12 = Lityum tamsayı [0 - 1]
x13 = Guatr tamsayısı [0 - 1]
x14 = Tümör tamsayısı [0 - 1]
x15 = Hipopitüiter tamsayı [0 - 1]
x16 = Psikolojik tamsayı [0 - 1]
x17 = TSH [0.0 - 0.53]
x18 = T3 [0.0005 - 0.18]
x19 = TT4 [0.0020 - 0.6]
x20 = T4U [0.017 - 0.233]
x21 = FTI [0.0020 - 0.642]

Tiroit problemi için ÇKA ağı çıkış nöron sayıları değiştirilerek iki farklı topoloji üzerinden ayrı ayrı çalıştırılarak eğitildi. Tiroit veri seti için kullanılan farklı topolojiler şematik olarak Şekil 4.3.'den görülmektedir.



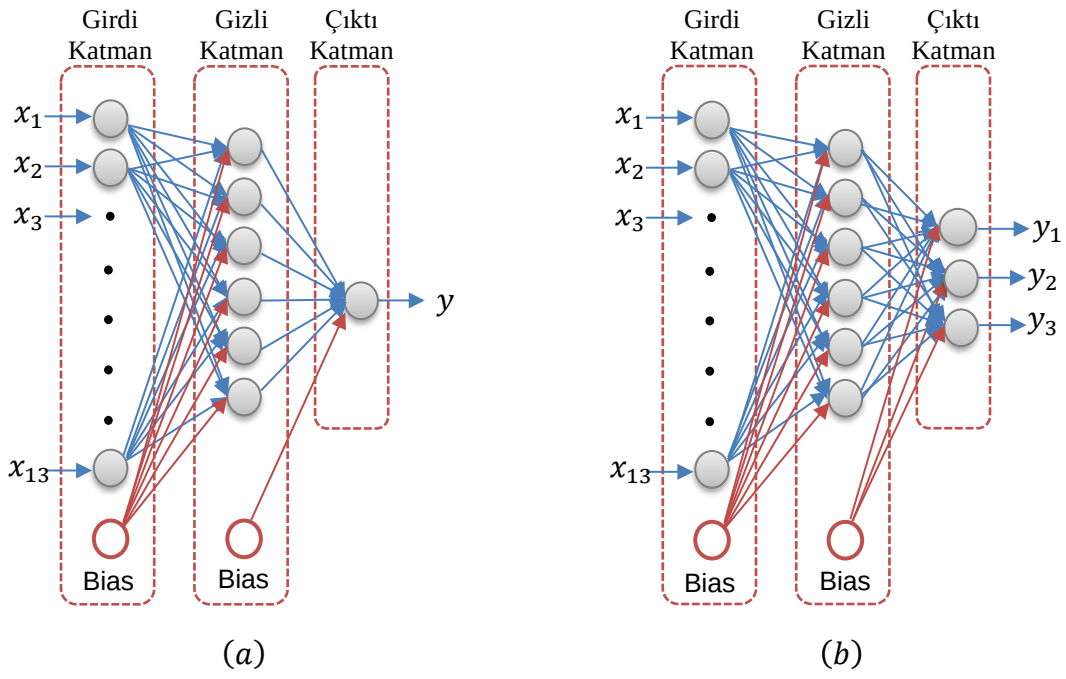
Şekil 4.3. Tiroit probleminin (a) ÇKA ağının tek çıkışlı topolojisi (b) ÇKA ağının çok çıkışlı topolojisi

4.1.4. Şarap Problemi (Wine Recognition Data Set)

Şarap tanıma verileri 1998 yılında C. Blake tarafından eklenmiştir (Forina, 1998). Bu veriler, İtalya'daki aynı bölgede yetiştirilen ve üç farklı türden üretilen şarapların kimyasal analiz sonuçlarıdır. Analiz kapsamında şarap türlerinde bulunan 13 bileşenin miktarları kaydedilmiştir. Dolayısıyla veri seti 13 girdiden oluşur. Şarap türleri bu girdilere göre 3 sınıfa ayrılmıştır. Veri seti toplam 178 örnek içerir. Veri setinin öznitelikleri aşağıda sıralanmıştır.

x1 = Alkol [11.03 - 14.83]
x2 = Malik asit [0.74 - 5.80]
x3 = Kül [1.36 - 3.23]
x4 = Külün alkalin itesi[10.60 - 30.00]
x5 = Magnezyum [70.00 - 162.00]
x6 = Toplam fenoller [0.98 - 3.88]
x7 = Flavonoidler [0.34 - 5.08]
x8 = Nonflavanoid fenoller [0.13 - 0.66]
x9 = Proantosiyanidinler [0.41 - 3.58]
x10 = Renk yoğunluğu [1.28 - 13.00]
x11 = Renk tonu [0.48 - 1.71]
x12 = Seyreltilmiş şarapların OD280 / OD315 oranı [1.27 - 4.00]
x13 = Proline [278.00 - 1680.00]

Şarap veri setinde; birinci Sınıf için 59, ikinci sınıf için 71 ve üçüncü sınıf için 48 veri örnek bulunmaktadır. Şarap türü sınıflandırma problemi için ÇKA ağı çıkış nöron sayıları değiştirilerek iki farklı topoloji üzerinden ayrı ayrı çalıştırılarak eğitildi. Şarap veri seti için kullanılan farklı topolojiler şematik olarak Şekil 4.4.'den görülmektedir.



Şekil 4.4. Şarap probleminin (a) ÇKA ağının tek çıkışlı topolojisi (b) ÇKA ağının çok çıkışlı topolojisi

4.1.5. İris Problemi (Iris Plants Data Set)

İris bitki veri seti, Örüntü tanıma literatüründe en bilinen ve en çok kullanılan veri setidir (Frank ve Asuncion, 2010). Veri seti 4 girdi ve 3 sınıftan oluşur. Veri seti her sınıf için ellişer tane olmak üzere toplam 150 örnek içerir.

Özellikleri şunlardır:

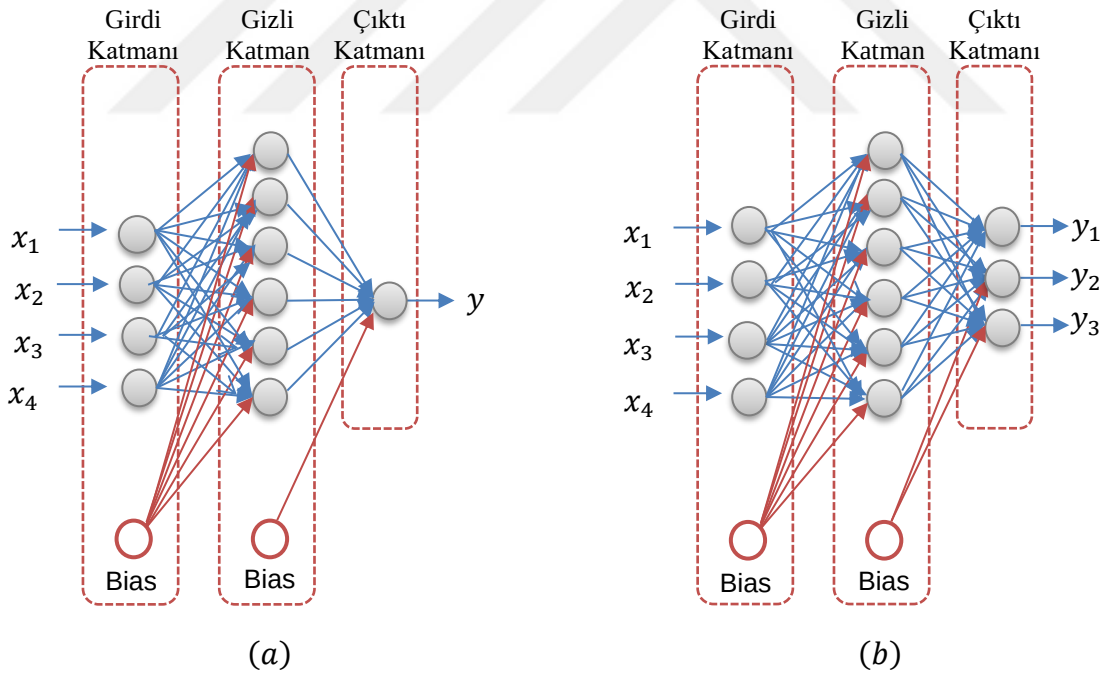
x1 = Sepal uzunluk [4.3,7.9] cm

x2 = Sepal genişliği [2.0,4.4] cm

x3 = Petal uzunluk [1.0,6.9] cm

x4 = Petal genişliği [0.1,2.5] cm

Birinci Sınıf İris Setosa, ikinci sınıf İris Versicolour ve üçüncü sınıf İris Virginica olarak tasnif edilmiştir. İris türü bitki sınıflandırma problemi için ÇKA ağı çıkış nöron sayıları değiştirilerek iki farklı topoloji üzerinden ayrı ayrı çalıştırılarak eğitildi. İris veri seti için kullanılan farklı topolojiler şematik olarak Şekil 4.5.'den görülmektedir.



Şekil 4.5. İris probleminin (a) ÇKA ağının tek çıkışlı topolojisi (b) ÇKA ağının çok çıkışlı topolojisi

4.2. ÇKA Ağlarının Eğitiminde Yapılan Ayarlamalar

Girdap Arama algoritmasının yapay sinir ağı eğitimindeki performansını karşılaştırmak amacıyla literatürde başarılı performanslarından dolayı sıkça tercih edilen

meta-sezgisel tekniklerden ABC ve PSO optimizasyon algoritmaları kullanıldı. Performansı testi amacıyla seçilen sınıflandırma problemleri üzerinde algoritmaların hepsi aynı parametre değerleri ve topolojiler kullanılarak çalıştırıldı. Ağ yapılarında ara ve çıkış katmanlardaki nöronlara bias değerleri uygulandı ve aktivasyon fonksiyonu olarak sigmoid fonksiyonu kullanıldı.

Giriş katmanı, gizli katman ve çıkış katmanından oluşan 3 katmanlı ÇKA ağ yapısı her problem için kullanıldı. Giriş ve çıkış katmanlarındaki nöron sayısı, problemlerin özelliklerine bağlı olarak sabit olmakla birlikte; çıktı katmanı için tek-çıkışlı ve çok-çıkışlı olacak şekilde iki farklı topoloji seçilerek eğitim işlemi gerçekleştirildi. Gizli katmandaki nöron sayıları ise test problemleri için literatürde önerilen değerlerden belirlendi. Deneme yanılma yoluyla yapılan çok sayıda eğitimden sonra bu problemler için literatürde tavsiye edilen 6 nöronlu bir gizli katmanın kullanıldığı bir topoloji oluşturuldu (Öztürk ve Karaboğa, 2011).

Eğitimde kullanılan ÇKA ağının topolojisine dair parametreler Tablo 4.1 ve Tablo 4.2'de verildi. Tablo 4.1'de çok-çıkışlı yapay sinir ağlarının parametreleri ve Tablo 4.2'de tek-çıkışlı yapay sinir ağlarının parametreleri yer almaktadır. Tabloların ilk sütununda problem için kullanılan topoloji, ikinci sütununda biaslarla birlikte optimal değerleri aranan toplam ağırlık sayısı ve üçüncü sütunda veri setinde bulunan toplam örnek sayısı görülmektedir. Problemlerin giriş ve çıkış katmanlarındaki nöron sayılarına göre problemin boyutunu temsil eden büyüklük değerleri eğitim aşamasında ayarlanacak ağırlıkların sayılarıdır.

Tablo 4.1. Sınıflandırma Problemleri için Çok-Çıkışlı ÇKA Ağında Kullanılan Parametreler

Problem	YSA Yapısı	Boyutu	Örnek Sayısı
Diyabet	8-6-2	68	768
Kanser	10-6-2	80	963
Tiroit	21-6-3	153	7193
Şarap	13-6-3	105	178
İris	4-6-3	51	150

Tablo 4.2. Sınıflandırma Problemleri için Tek-Çıkışlı ÇKA Ağında Kullanılan Parametreler

Problem	YSA Yapısı	Boyutu	Örnek Sayısı
Diyabet	8-6-1	61	768
Kanser	10-6-1	73	963
Tiroit	21-6-1	139	7193
Şarap	13-6-1	91	178
İris	4-6-1	37	150

Önerilen yaklaşımın performans değerlendirmesi üç aşamada yapıldı. Birinci aşamada; tüm problemler için veri setinin tamamı eğitim seti olarak kullanıldı. İkinci aşamada; veri setinin %75'i eğitim seti olarak kullanılırken, geri kalan %25'i ise test verisi olarak ayrıldı. Üçüncü aşamada ise k-katlamalı-çapraz-doğrulama yöntemi kullanıldı.

Bağımsız çalıştırma işlemleri, her bir problem için rasgele oluşturulan popülasyonlar ile 30 kez tekrarlandı. Ağın eğitim işlemi, MSE değeri 0.01 olduğunda veya maksimum iterasyona ulaşıldığında durduruldu. Her aşamada elde edilen eğitim sonuçları ayrı ayrı kaydedildi ve algoritmaların performans değerlendirmeleri yapıldı. Maksimum iterasyon sayısı 100, ağırlıkların değer aralıkları olarak [-20, 20] ve popülasyon boyutu 30 olarak alındı. VS algoritması için limit değeri 100 olarak kullanıldı. PSO algoritması için eylemsizlik faktörü 0.6 ve hızlandırma faktörü 1.8 seçildi. ABC algoritması için limit değeri 100 olarak kullanıldı. Bu seçilen değerler her bir algoritmalar için literatürde tavsiye edilen değerlerdir (Han ve May, 1996); (Carvalho ve Ludermir, 2006).

Her bir algoritmanın çoklu çalıştırılmaları neticesinde ulaştığı değerlerin objektif bir karşılaştırma ile değerlendirilebilmesi için bazı istatistiksel ölçütlerden faydalanıldı. Sonuçların doğruluğunu karşılaştırmak amacıyla ortalama ve ortanca değerleri; algoritmaların tutarlı çalışıp çalışmadığını saptamak amacıyla ise istatistiksel dağılımları göstermek için kullanılan çeyrekler açıklığı (IQR) ve standart sapma (std) değerleri hesaplandı.

Ortalama ($\bar{x}$) değeri bulmak için her bir çalıştırma sonunda elde edilen MSE değerlerinin aritmetik ortalaması hesaplandı. Ortalama formülü Denklem (4.3) ile ifade edilir.

$$\bar{x} = \frac{\sum x}{n} \tag{4.3}$$

Standart sapma (std) değeri varyansın karekköküdür. Standart sapma formülü Denklem (4.4) ile ifade edilir.

$$std = \sqrt{\frac{\sum(x-\bar{X})^2}{n-1}} \tag{4.4}$$

Çeyrekler açıklığı (IQR) formülü Denklem (4.5) ile ifade edilir.

$$IQR = Q_3 - Q_1 \tag{4.5}$$

Çeyrekler açıklığı sıralanmış bir veri setinin orta yarısını %50sini kapsayan ve üçüncü dörttebirlik ve birinci dörttebirlik farkını gösteren bir istatistiksel yayılma

ölçüsüdür. Q_1'lik kısım x değerlerinin % 25 den fazla ve % 75 den az olan kısımdır. Q_3 ise x değerlerinin % 75 den fazla ve % 25 den az olan kısımdır. Ortadaki 50% aralığı ise dörtlükler arası aralık,

Test problemlerinde en iyi sonucu veren sonuçlar koyu gri arka plan rengiyle vurgulanmıştır. Birden fazla algoritma aynı en iyi değere ulaşabilirse, daha küçük IQR değerine sahip algoritma en iyi kabul edilmiştir.

Literatür çalışmaları tablo 4.3'da özet olarak verilmiştir. Tabloda veri setleri ilgili sınıflandırmada kullanılan yapay zekâ yöntemleri ve sonuçta elde edilen doğruluk oranları bulunmaktadır. Çalışmalarda başarılı sonuçlara ulaşıldığı görülmektedir.

Tablo 4.3. Sınıflandırma Problemlerine Ait Deneysel Başarılar

Referans	Veri Seti	Yöntem	Başarı (%)	Önerilen Yöntem ÇKA-VS (%)
(Öztürk ve Karaboğa, 2011)	Diyabet	YSA-ABC	74	76
(Dordinejad, 2016)	Diyabet	SVM	77	
(Arce ve ark., 2018)	Diyabet	MLP-BMNN	71	
(Kaymak ve ark., 2017)	Göğüs Kanseri	RBNN	70	95
(Bektaş ve Babur, 2016)	Göğüs Kanseri	ML-Rastgele Orman Algoritması	90	
(İ.İşeri ve ark., 2011)	Göğüs Kanseri	EWD-MLPNN	95	
(Uslu, 2013)	Göğüs Kanseri	YSA-LM	95	
(Öztürk ve Karaboğa, 2011)	Tiroit	YSA-ABC	93	94
(Yurtay ve ark, 2016)	Tiroit	MLNN with LM	92	
(Çomak ve Arslan, 2008)	Tiroit	BEKKDVM	94	
(Arce ve ark., 2018)	Şarap	SVM-BMNN	57	88
(Çomak ve Arslan, 2008)	Şarap	EKKDVM	94	
(P. Rijnbeek ve ark, 2010)	Şarap	CBA	86	
(Arce ve ark., 2018)	İris	DE-BMNN	100	100
(Çomak ve Arslan, 2008)	İris	EKKDVM	94	
(Duch W., 2001)	İris	C-MLP2LN	98	

4.2.1. Birinci Aşama Performans Değerlendirmesi

Bu aşamada, veri setlerinin tamamı eğitim için kullanıldı. VS, ABC ve PSO algoritmaları eşit koşullarda tüm problemler için otuzar defa çalıştırıldı. Çıktı katmanı için tek-çıkışlı ve çok-çıkışlı olacak şekilde iki farklı topoloji seçilerek eğitim işlemi gerçekleştirildi.

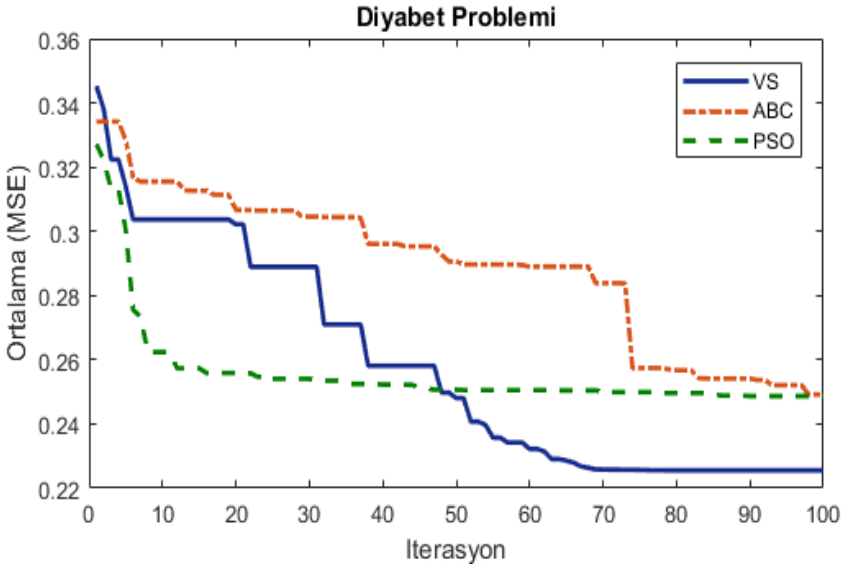
4.2.1.1. Çok-Çıkışlı ÇKA Ağlarının Birinci Aşama Performans Değerlendirmesi

Çok-çıkışlı ÇKA ağlarının sınıflandırma test problemleri için eğitilmesi amacıyla algoritmaların 30'ar kez çalıştırılmaları sonucu her bir problem için elde edilen ortalama ve ortanca MSE değerleri ile bu değerlere ait IQR ve standart sapma değerleri Tablo 4.4'de verilmektedir. Sonuçlar incelendiğinde ortalama MSE değerlerine göre VS algoritması beş test probleminin üçünde daha başarılı; diğer iki problem için de oldukça iyi sonuçlar verdiği görülmektedir.

Diyabet probleminde VS algoritmasının ABC ve PSO'dan daha iyi eğitim performansı elde ettiği açıkça görülmektedir. ÇKA-VS modelinin ortalama ve ortanca MSE değerleri sırasıyla 0.232657 ve 0.228187'dir. Bu değerler ile diyabet veri seti için en iyi performans gösteren algoritmadır. Aynı şekilde, IQR ve standart sapma değerleri sırasıyla 0.007135 ve 0.014212'dir. Bu değerler de VS algoritmasının tutarlı bir çalışma istikrarına sahip olduğunu gösterir. PSO 0.243124 ortalama değeriyle ikinci, ABC ise 0.244650 ortalama değeri ile üçüncüdür. Şekil 4.6'da VS, ABC ve PSO algoritmalarının diyabet problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.

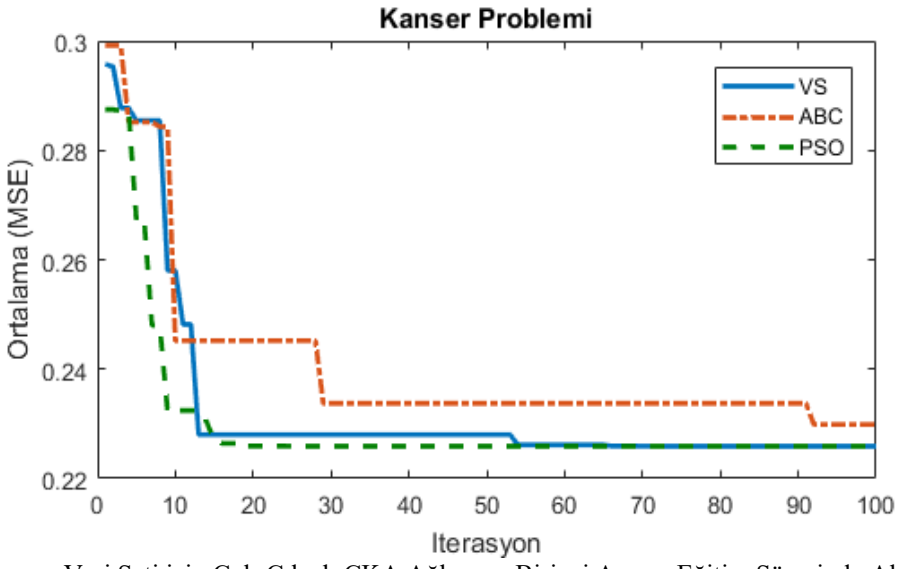
Tablo 4.4. Birinci Aşama Eğitim Süreci Sonunda 30 Bağımsız Çalıştırma için Çok-Çıkışlı ÇKA Ağlarında Algoritmaların Elde Ettiği MSE Değerleri

Veri Seti		VS	ABC	PSO
Diyabet	Ortanca	0.228187	0.241065	0.230711
	IQR	0.007135	0.022720	0.025444
	Ortalama	0.232657	0.244650	0.243124
	std	0.014212	0.014467	0.024070
Kanser	Ortanca	0.226215	0.229048	0.226217
	IQR	2.775557e-15	0.009717	1.933154e-07
	Ortalama	0.226212	0.232972	0.228207
	std	6.990853e-14	0.008754	0.010913
Tiroit	Ortanca	0.046835	0.049372	0.047296
	IQR	0.001670	4.667226e-04	0.002955
	Ortalama	0.046441	0.049356	0.046685
	std	0.001364	0.001024	0.002354
Şarap	Ortanca	0.175750	0.226986	0.172570
	IQR	0.030034	0.019739	0.052577
	Ortalama	0.174335	0.223959	0.172399
	std	0.030584	0.024445	0.038813
Iris	Ortanca	0.111057	0.067992	0.024490
	IQR	0.104282	0.058490	0.098379
	Ortalama	0.082300	0.073413	0.054430
	std	0.057370	0.033083	0.054384



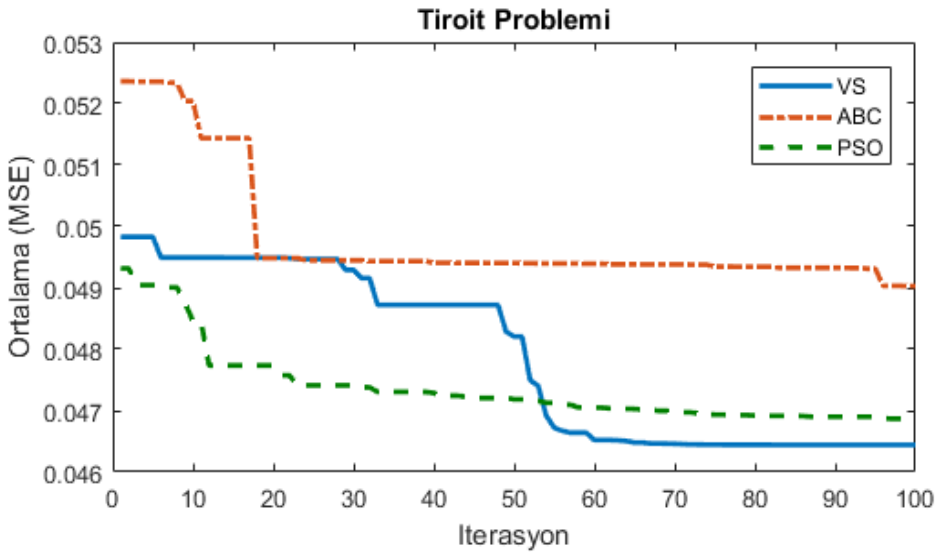
Şekil 4.6. Diyabet Veri Seti için Çok-Çıkışlı ÇKA Ağlarının Birinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

Kanser veri seti, girdi ve parametre boyutlarına göre Diyabet veri seti ile benzerlik gösterir. Böyle olmasına rağmen sonuçlara bakıldığında; Diyabet'den daha zor bir problem olduğu gözlemlenmektedir. Algoritmaların ortalama değerleri birbirine yakın ve 0.22 civarındadır. Ancak VS 2.775557e-15 IQR ve 6.990853e-14 standart sapma değerleriyle istatistiksel olarak daha güvenli bir aralıkta diğer algoritmalardan daha tutarlı bir çalışma eğilimi göstermiştir. Bu bakımdan VS algoritmasının daha iyi bir performans sergilediği söylenebilir. Sırasıyla 0.226212 ve 0.226215 ortalama ve ortanca MSE değerleri ile VS en iyi performans gösteren algoritmadır. PSO 0.228207 ortalama değeriyle ikinci, ABC ise 0.232972 ortalama değeri ile üçüncüdür. Şekil 4.7'de VS, ABC ve PSO algoritmalarının kanser problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



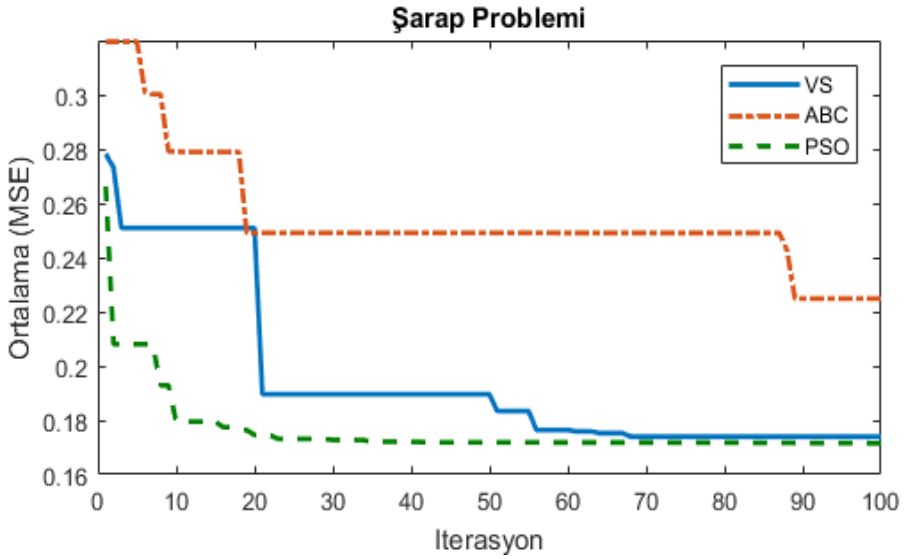
Şekil 4.7. Kanser Veri Seti için Çok-Çıkışlı ÇKA Ağlarının Birinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

Tiroit problemi 21 girdi ve 7193 örnek ile ele alınan problemler arasında en fazla girdi ve parametreye sahip olan test problemidir. Algoritmaların ortalama değerleri birbirine yakın ve 0.04 civarındadır. VS ortanca ve IQR değerlerin sırasıyla 0.046835 ve 0.001670 değerleri ile diğer algoritmaları geride bırakmış ve iyi başarılı sonucu elde etmiştir. Benzer şekilde, 0.046441 otalama ve 0.001364 standart sapma değerlerine göre değerlendirilirse yine en iyi performansı sergilediği görülür. Tiroit probleminde ortalama değerlerine göre algoritmaların performansları VS, PSO, ABC şeklinde sıralanmaktadır. Şekil 4.8'de VS, ABC ve PSO algoritmalarının Tiroit problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



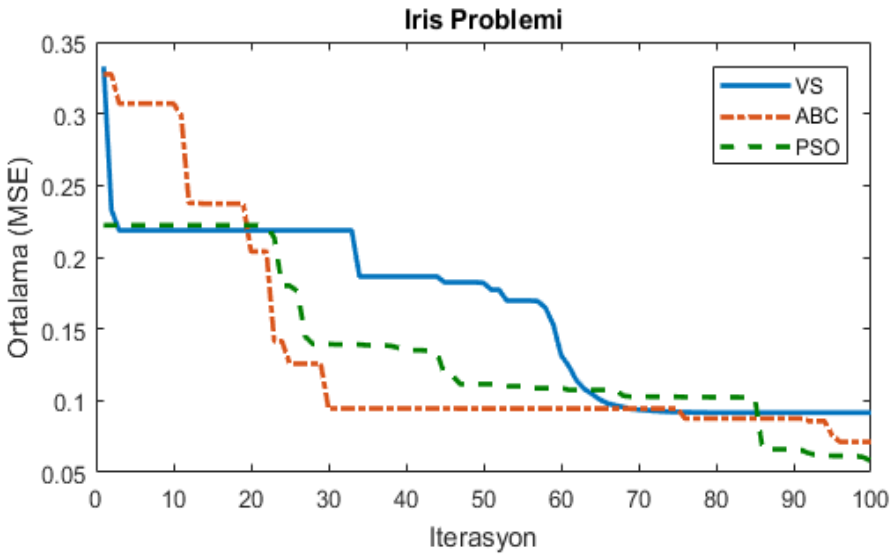
Şekil 4.8. Tiroit Veri Seti için Çok-Çıkışlı ÇKA Ağlarının Birinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

Şarap probleminde 3 çıktı sınıfı ile 178 örnek ele alınmıştır. PSO ve VS algoritmaları birbirlerine yakın performanslar sergilemiştir. PSO 0.172399 ortalama değeri ve 0. 038813 standart sapma değeri ile VS algoritmasını çok az bir farkla geride bırakmıştır. PSO'nun ortanca ve IQR değerleri sırasıyla 0.172399 ve 0. 052577'dir. VS 0.174335, ABC 0.223959 ortalama değerlerini üretmişlerdir. Şarap probleminde ortalama değerlerine göre algoritmaların performansları PSO, VS, ABC şekilde sıralanmaktadır. Şekil 4.9'da VS, ABC ve PSO algoritmalarının şarap problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



Şekil 4.9. Şarap Veri Seti için Çok-Çıkışlı ÇKA Ağlarının Birinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

İris veri setinde ortalama değerlerine bağlı olarak VS algoritmasının göstermiş olduğu performans diğer veri setleri kadar iyi değildir. PSO algoritması ortanca ve IQR değerleri sırasıyla 0.024490 ve 0.098379 değerler ile diğer algoritmaları geride bırakmış ve iyi performans sergilemiştir. Benzer şekilde 0.054430 ortalama değeri ve 0.054384 standart sapma değeri ile PSO diğer algoritmaları az farkla geride bırakmıştır. ABC 0.073413 ortalama değeriyle ikinci, VS 0.082300 ortalama değerleriyle üçüncüdür. Şekil 4.10'da VS, ABC ve PSO algoritmalarının iris problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



Şekil 4.10. İris Veri Seti için Çok-Çıkışlı ÇKA Ağlarının Birinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

4.2.1.2. Tek-Çıkışlı ÇKA Ağlarının Birinci Aşama Performans Değerlendirmesi

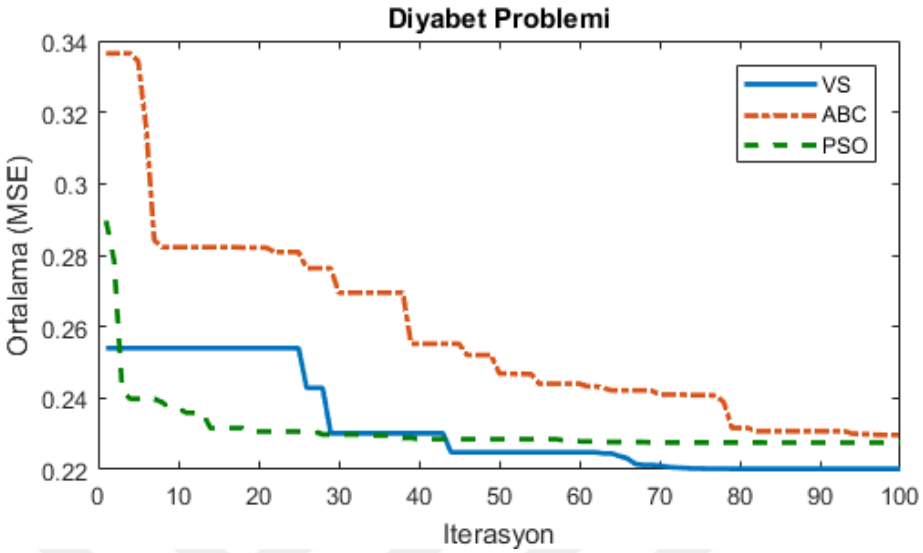
Tek-Çıkışlı ÇKA ağlarının sınıflandırma test problemleri için eğitilmesi amacıyla algoritmaların 30'ar kez çalıştırılmaları sonucu her bir problem için elde edilen ortalama ve ortanca MSE değerleri ve bunların IQR ve standart sapma değerleri Tablo 4.5'de verilmektedir. Sonuçlar incelendiğinde MSE değerlerine göre VS algoritması test problemlerinin tamamında ya daha iyi ya da en az diğer algoritmalar kadar iyi sonuçlara ulaşmıştır.

Tablo 4.5. Birinci Aşama Eğitim Süreci Sonunda 30 Bağımsız Çalıştırma için Tek-Çıkışlı ÇKA Ağlarında Algoritmaların Elde Ettiği MSE Değerleri

Veri Seti		VS	ABC	PSO
Diyabet	Ortanca	0.222608	0.227202	0.221281
	IQR	0.007621	0.006371	0.010272
	Ortalama	0.220531	0.226283	0.220471
	Std	0.005760	0.005060	0.013211
Kanser	Ortanca	0.225906	0.227548	0.225906
	IQR	7.494005e-16	1.162234e-04	1.823186e-09
	Ortalama	0.225906	0.227640	0.220423
	Std	5.109564e-16	2.803210e-04	0.030029
Tiroit	Ortanca	0.942856	0.948815	0.948422
	IQR	0.006589	1.048063e-04	6.340784e-04
	Ortalama	0.943994	0.948777	0.947452
	std	0.003135	8.374547e-05	0.002248
Şarap	Ortanca	1.477528	1.477528	1.477528
	IQR	0	0	0
	Ortalama	1.477528	1.477528	1.477528
	std	6.775215e-16	6.775215e-16	6.775215e-16
Iris	Ortanca	0.333333	0.3333334	0.333333
	IQR	2.670641e-13	4.194422e-13	0
	Ortalama	0.333333	0.333343	0.333335
	std	4.937758e-09	3.845957e-11	7.635772e-13

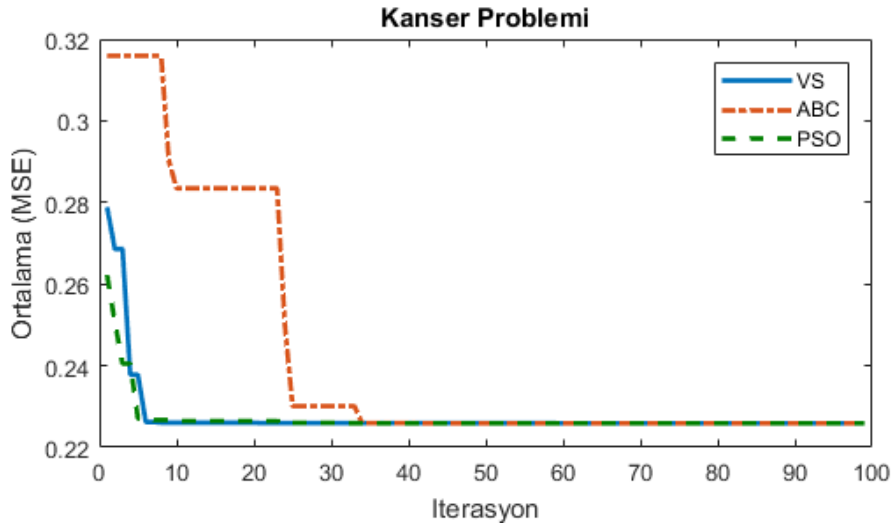
Diyabet problemine bakıldığında algoritmaların ortalama ve ortanca değerleri birbirine çok yakın ve 0.22 civarındadır. PSO algoritmasının 0.221281 ortanca ve 0.010272 IQR değerleri ile çok az bir farkla daha iyi bir eğitim performansı elde ettiği görülmektedir. VS algoritmasının ortalama ve standart sapma değerleri sırasıyla 0.220531 ve 0.005760 ile diğer algoritmaları geride bırakmış ve iyi performans sergilemiştir. Aslında bu durum yani ortanca değerlerinin hemen hemen aynı olması ve ortalama değer bakımından da daha iyi bir değer elde edebilmiş olması; VS algoritmasının istatistiksel dağılım açısından daha tutarlı çalıştığını ve daha düşük bir sapmayla tutarlı sonuçlar elde

edebildiğinin göstergesidir. PSO 0.220471 ortalama değeriyle ikinci. ABC 0.226283 ortalama değeriyle üçüncüdür. Şekil 4.11'da VS, ABC ve PSO algoritmalarının diyabet problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



Şekil 4.11. Diyabet Veri Seti için Tek-Çıkışlı ÇKA Ağlarının Birinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri.

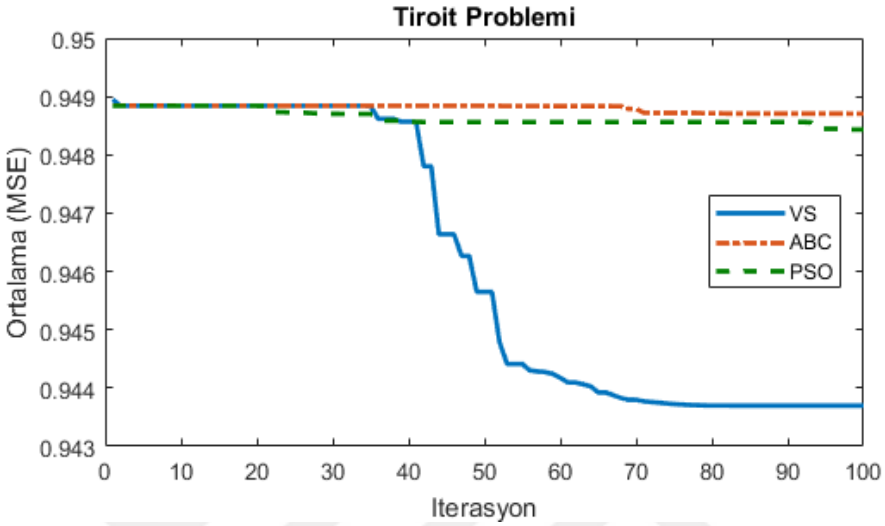
Kanser probleminde algoritmaların ortanca ve ortalama değerleri birbirine çok yakın ve 0.22 civarındadır. Ortanca değerleri birbirine yakın olsa da IQR 7.494005e-16 değeri ile VS iyi performans sağlamış ve öne geçmiştir. Ortalama ve standart sapma değerleri sırasıyla 0.220423 ve 0.030029 değeriyle PSO'da az farkla iyi sonuçlanmıştır. Şekil 4.12'da VS, ABC ve PSO algoritmalarının kanser problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



Şekil 4.12. Kanser Veri Seti için Tek-Çıkışlı ÇKA Ağlarının Birinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri.

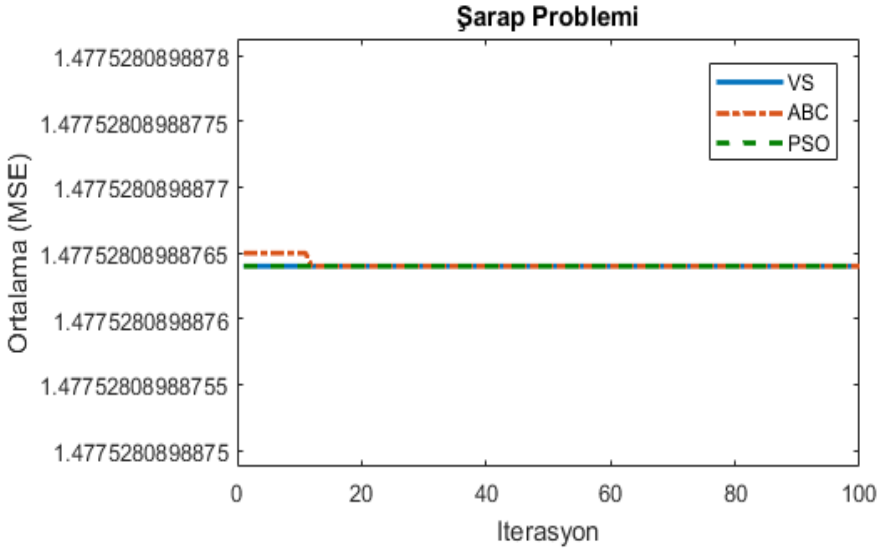
Tiroit probleminde algoritmaların ortalama ve ortanca değerleri birbirine yakın ve 0.94 civarındadır. VS algoritmasının ABC ve PSO'dan daha iyi eğitim performansı elde

ettiği görülmektedir. ÇKA-VS modeli, sınıflandırma doğruluğu olarak kullanılan ortalama ve ortanca MSE değerleri sırasıyla 0.943994 ve 0.942856 değerler ile en iyi performans gösteren algoritmadır. PSO 0.947452 ortalama değeriyle ikinci, ABC ise 0.948777 ortalama değeri ile üçüncüdür. Şekil 4.13'te VS, ABC ve PSO algoritmalarının tiroit problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



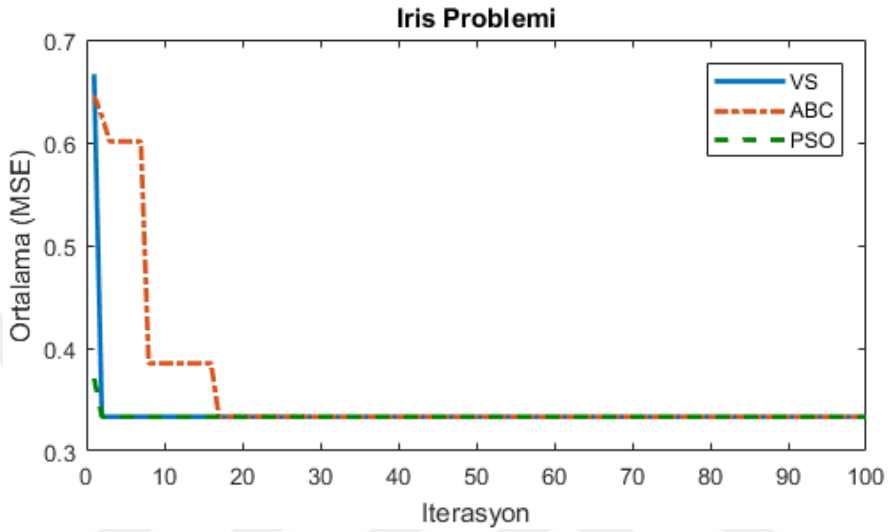
Şekil 4.13. Tiroit Veri Seti için Tek-Çıkışlı ÇKA Ağlarının Birinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

Şarap probleminde ortalama MSE değerlerine bağlı olarak algoritmaların göstermiş oldukları performanslar diğer veri setleri kadar iyi değildir. VS, ABC ve PSO algoritmalarının başarısız olduğu ve tüm algoritmalar aynı ortalama 1.477528 değeri ile eşdeğer performans sergiledikleri görülmektedir. Şekil 4.14'te VS, ABC ve PSO algoritmalarının şarap problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



Şekil 4.14. Şarap Veri Seti için Tek-Çıkışlı ÇKA Ağlarının Birinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

İçerdiği örnekler sayısının az olması ve gürültünün hemen hemen hiç olmaması nedeniyle sınıflandırması en kolay veri setlerinden birisi olan İris veri setinde ortalama MSE değerlerine bağlı olarak algoritmaların tamamının aynı başarılı sonuçlara ulaşarak çok benzer performanslar sergiledikleri görülmektedir. Tüm algoritmalar aynı ortalama 0.3333 değerine ulaşmıştır. Şekil 4.15'te VS, ABC ve PSO algoritmalarının iris problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



Şekil 4.15. İris Veri Seti için Tek-Çıkışlı ÇKA Ağlarının Birinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

Tablo 4.4 ve Tablo 4.5'deki sınıflandırma hatalarından görüldüğü üzere VS algoritması yapay sinir ağlarının eğitilmesinde en az diğer meta-sezgisel optimizasyon algoritmaları kadar iyi bir performansa sahiptir. Buna rağmen VS algoritmasını tüm problemlerde en iyi performansı gösterdiği söylenemez. Farklı yapay sinir ağı mimarileri her bir veri setine uygulandığında daha iyi sınıflandırma sonucu elde etmek mümkün olabilir. Problemlerde özellikle sinir ağı yapısı seçme veya ayarlama işlemi yapılmamıştır. Bunun yerine literatürde en fazla kullanılan ağ yapıları referans alınmıştır.

Sonuç olarak; iki tabloya bakıldığında VS algoritması, Çok-Çıkışlı yapay sinir ağının eğitilmesinde Tek-Çıkışlı yapay sinir ağına kıyasla daha iyi performans sergilediğini görüldü. Özellikle şarap ve iris verilerinde Çok-Çıkışlı ileri beslemeli ağ yapısında Tek-Çıkışlı ağ yapısına kıyasla tüm algoritmalar daha başarılı sonuçlar elde etti. Daha önce YSA eğitimi için literatürde denenmemiş olan Girdap Arama Optimizasyon algoritmasının en az diğer algoritmalar kadar iyi sonuçlar elde ettiği ve sınıflandırma problemlerinde başarıyla ileri beslemeli çok katmanlı algılayıcı yapay sinir ağlarının eğitiminde kullanılabilir bir araç olduğu görüldü.

4.2.2. İkinci Aşama Performans Değerlendirmesi

Bu aşamada veri kümelerinden her problemin %75'i eğitim verisi, geriye kalan %25'i ise test verisi olarak kullanıldı. Uygulamalarda bir veri seti ağa eğitim seti ve test seti olmak üzere iki parça halinde uygulanır. Eğitim verileriyle yapay sinir ağı eğitildikten sonra, ağın görmediği test verileri ağa uygulanır ve ağın vereceği tepki hesaplanır. Bunun sebebi ağın ezberlemesini engellemektir. Ezberleme (overfitting) olarak adlandırılır ve bu durum çoğunlukla ayarlanabilen ağırlık sayısı çok fazla olduğunda oluşur. Ezberlemenin oluşmasını engellemenin çeşitli yolu vardır; örneğin eğer eğitim seti gürültü içeriyorsa, çok sayıda gizli nöron olan bir ağ gürültüyü öğrenebilir, bunun yanında ağın genelleştirme yeteneğini azaltır. Ağda, gizli katmanda bulunan nöron sayısı sınırlanırsa, ağ gürültüyü öğrenemez hale gelir ve genelleştirme performansı artar.

VS, ABC ve PSO algoritmaları eşit koşullarda tüm problemler için otuzar defa çalıştırıldı. Her bir algoritmanın bu çoklu çalıştırılmaları neticesinde ulaştığı değerlerin objektif bir karşılaştırma ile değerlendirilebilmesi için bazı istatistiksel ölçütlerden faydalanıldı. Ortalama-Standart Sapma ve Ortanca-Çeyrekler Açıklığı (IQR) değerleri hesaplandı.

Bir girdi vektörü eğitilmiş bir YSA'ya sunulduğunda ağ önceki tecrübelerine göre çıktıyı hesaplar. Eğer beklenen çıktı ile hesaplanan çıktı aynı değilse örnek yanlış sınıflandırılmış demektedir. Bu işlem, tüm test verileri için tekrarlanır ve toplamda doğru sınıflandırılmış örnek sayısı, test veri setinin eleman sayısına bölünerek sınıflandırma doğruluğunun yüzdesi hesaplanır. Bu işlem matematiksel olarak Denklem (4.6)'deki gibi ifade edilir.

$$\text{Sınıflandırma Doğruluğu (\%)} = \frac{|\text{Doğru Sınıflandırılmış Test Verileri}|}{|\text{Test Veri Seti}|} \times 100 \qquad (4.6)$$

Algoritmaların sınıflandırma başarılarının analizi için ağın eğitimi ve test edilmesi aşamalarında elde edilen bulgular ayrı tablolar halinde sunuldu. Eğitim sırasında hesaplanan ortalama ve ortanca sınıflandırma hataları (MSE) ve bu değerlere ait IQR ve standart sapma değerleri hesaplandı. Bununla birlikte eğitimden sonra test aşamasında yüzde olarak hesaplanan sınıflandırma doğrulukları ayrıca raporlandı.

4.2.2.1. Çok-Çıkışlı ÇKA Ağlarının İkinci Aşama Performans Değerlendirmesi

Algoritmaların 30'ar kez çalıştırılmaları sonucu eğitim aşamasında her bir problem için elde edilen ortalama ve ortanca MSE değerleri ve bunların IQR ve standart sapma değerleri Tablo 4.6'de verilmektedir. Tablo 4.6'e bakıldığında, çok-çıkışlı ÇKA ağlarının sınıflandırma test problemleri için eğitilmesinde elde edilen eğitim aşaması sonuçları görülmektedir. Ortalama MSE değerlerine göre VS algoritması test problemlerinin geneli üzerinde daha iyi sonuçlar vermiştir. Aynı zamanda VS, ABC ve PSO algoritmalarının her problem için eğitim aşamasındaki yakınsama grafikleri verilmiştir.

Tablo 4.6. İkinci Aşama Eğitim Süreci Sonunda 30 Bağımsız Çalıştırma için Çok-Çıkışlı ÇKA Ağlarında Algoritmaların Elde Ettiği MSE Değerleri

Veri Seti		VS	ABC	PSO
Diyabet	Ortanca	0.230063	0.243424	0.228266
	IQR	0.007252	0.026819	0.030318
	Ortalama	0.231582	0.247545	0.241037
	Std	0.008384	0.017514	0.027646
Kanser	Ortanca	0.226691	0.231809	0.226691
	IQR	2.470246e-15	0.015960	7.34803e-07
	Ortalama	0.226101	0.231120	0.236217
	Std	0.003228	0.009725	0.015340
Tiroit	Ortanca	0.046136	0.048201	0.046297
	IQR	0.001240	2.493799e-04	0.002189
	Ortalama	0.045958	0.048238	0.046115
	Std	0.001209	0.001414	0.001941
Şarap	Ortanca	0.158283	0.225705	0.170936
	IQR	0.032233	0.027916	0.069569
	Ortalama	0.166324	0.222914	0.175938
	Std	0.037729	0.020273	0.036239
Iris	Ortanca	0.109961	0.069505	0.053993
	IQR	0.079144	0.069981	0.093542
	Ortalama	0.086928	0.076992	0.066433
	Std	0.042511	0.037675	0.053925

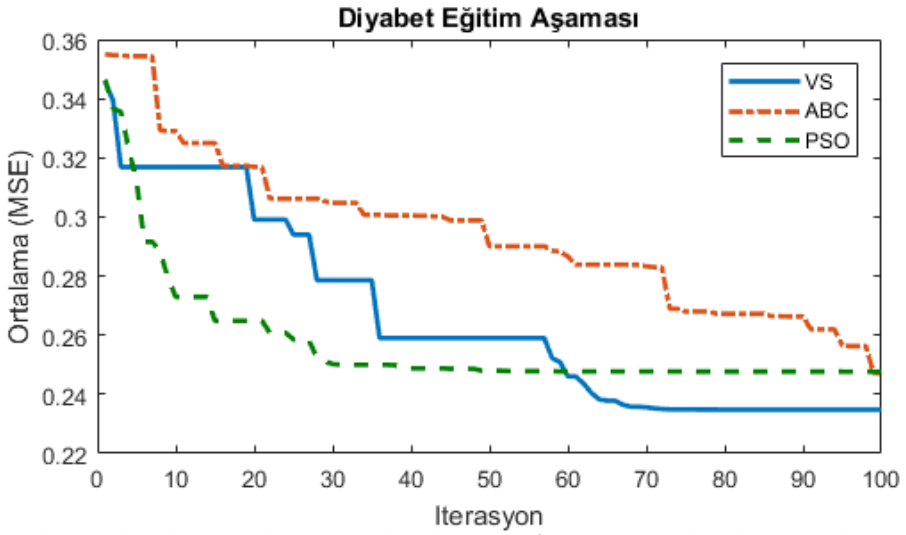
Çok-Çıkışlı topolojide algoritmaların 30'ar kez çalıştırılmaları sonucu test aşamasında her bir problem için elde edilen ve yüzde olarak hesaplanan sınıflandırma doğruluklarının maksimum, minimum, ortalaması ve ortanca değerleri ve bu değerlere ait IQR ve standart sapma değerleri Tablo 4.7'de verilmiştir. Ayrıca 30 çalıştırma içinde algoritmaların ulaştığı maksimum ve minimum sınıflandırma doğrulukları da Tablo 4.7'da yer almaktadır.

Tablo 4.7. İkinci Aşama Test İşlemi Sonunda 30 Bağımsız Çalıştırma için Çok-Çıkışlı ÇKA Ağlarında Algoritmaların Elde Ettiği Yüzde Olarak Sınıflandırma Doğrulukları

Veri Seti		VS	ABC	PSO
Diyabet	Ortanca	65.36	63.54	66.40
	IQR	2.604166	9.895833	1.562500
	Ortalama	65.13	59.02	62.56
	Std	2.069518	12.15961	9.562435
	Maks	67.19	68.75	67.18
	Min	60.41	7.8125	30.20
Kanser	Ortanca	66.28	66.28	66.28
	IQR	0	0	0
	Ortalama	66.28	66.28	66.28
	std	4.336137e-14	4.336138e-14	4.336137e-14
	Maks	66.28	66.28	66.28
	Min	66.28	66.28	66.28
Tiroit	Ortanca	92.01	91.93	92.04
	IQR	0.111234	0.222469	0.111234
	Ortalama	91.96	91.80	91.96
	Std	0.144097	0.708545	0.217989
	Maks	92.05	92.04	92.38
	Min	91.54	88.09	91.15
Şarap	Ortanca	57.95	19.31	48.86
	IQR	25	31.81818	43.18181
	Ortalama	44.84	22.12	39.24
	Std	22.75024	20.01827	23.76122
	Maks	65.90	61.36	63.63
	Min	0	0	0
Iris	Ortanca	59.45	75.67	67.56
	IQR	27.02703	27.02702	35.13513
	Ortalama	68.46	73.78	72.97
	Std	15.47438	15.29126	21.81876
	Maks	97.29	97.29	97.29
	Min	48.65	35.13	24.32

Sonuçlar incelendiğinde ortalama yüzde değerlerine göre VS algoritması problemlerin geneli üzerinde daha iyi sonuçlar verdiği görülmektedir. Tablo 4.6'deki verilere göre eğitim aşaması için diyabet probleminde VS algoritması 0.231582 ortalama değeri ile ABC ve PSO'dan daha iyi bir eğitim performansı elde ettiği görülmektedir. ÇKA-VS modeli 0.231582 ortalama ve 0.008384 standart sapma değerleri ile en iyi performans gösteren algoritmadır. PSO 0.228266 ortanca ve 0.030318 IQR değerleri ile iyi performans sergilemiştir. PSO 0.243124 ortalama değeriyle ikinci, ABC ise 0.244650 ortalama değeri ile üçüncüdür.

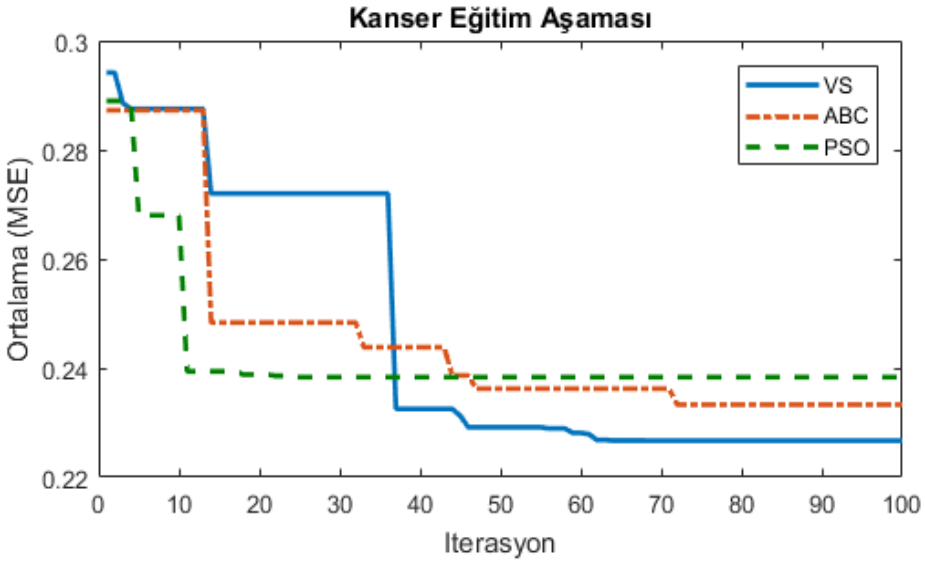
Test aşamasında ise diyabet probleminde VS algoritması %65.13 ortalama doğruluk ve 2.069518 standart sapma değerleri ile diğer algoritmaları geride bırakarak en iyi performans gösteren algoritma olmuştur. PSO %66.40 ortanca ve 1.562500 IQR ile en iyi değerlere ulaşmakla birlikte VS %65.36 ortanca doğruluk değeri ile çok yakın bir performansa sahiptir. Maks ve minimum değerlerini incelediğimizde %68.75 maks değeri ile ABC ve %60.41 min değeri ile VS algoritması iyi performans sergilemiştir. Şekil 4.16'da VS, ABC ve PSO algoritmalarının diyabet problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



Şekil 4.16. Diyabet Veri Seti için Çok-Çıkışlı ÇKA Ağlarının İkinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

Kanser probleminin sonuçlarına bakıldığında; eğitim aşamasında en iyi sonucu 0.226101 ortalama ve 0.003228 standart sapma değerleri ile VS en iyi performans gösteren algoritmadır. Ortanca değerleri incelendiğinde VS ve PSO 0.226691 değeriyle eşdeğer bir performans göstermiştir. Ancak VS 2.47024e-15 IQR değeriyle daha güvenli bir çalışma göstererek diğer algoritmaları geride bırakmıştır. MSE değerlerine göre VS algoritması en iyi performans sergilemiştir. ABC 0.231120 ortalama değeriyle ikinci, PSO ise 0.236217 ortalama değeri ile üçüncüdür.

Test aşamasında ise, ortalama MSE değerlerine bağlı olarak tüm algoritmalar göstermiş oldukları performanslar bir birine yakındır. VS, ABC ve PSO algoritmalarının aynı ortalama %66.28 doğruluk elde edebilmişlerdir. Şekil 4.17'da VS, ABC ve PSO algoritmalarının kanser problemi için eğitim ve test aşamasındaki yakınsama grafikleri verilmiştir.

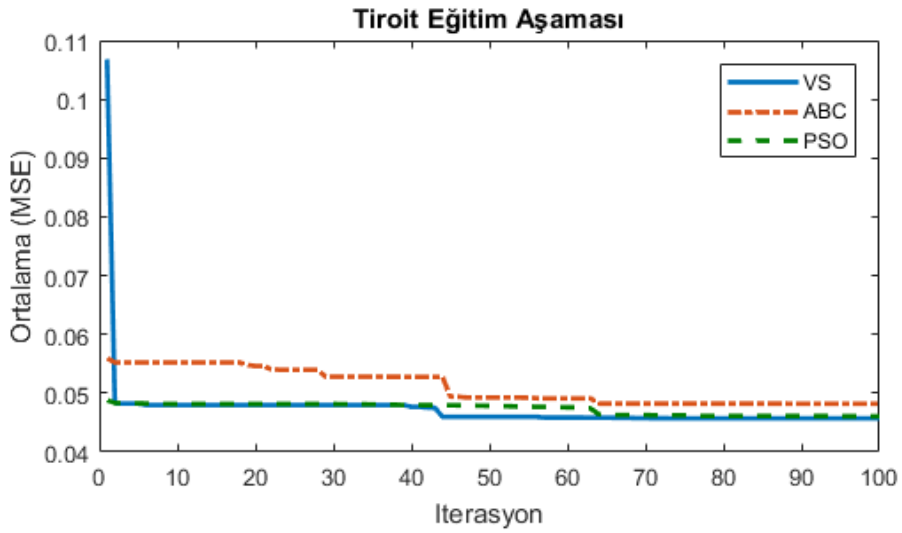


Şekil 4.17. Kanser Veri Seti için Çok-Çıkışlı ÇKA Ağlarının İkinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

Tiroit probleminde eğitim aşamasında, algoritmaların ortalama ve ortanca değerleri birbirine yakın ve 0.04 civarındadır. Bununla birlikte az da olsa VS algoritması hem ortalama hem de ortanca değerleri ile ABC ve PSO'dan daha iyi eğitim performansı elde ettiği görülmektedir. ÇKA-VS modeli 0.045958 ortalama ve 0.001209 standart sapma değerleri değerler ile en iyi performans gösteren algoritmadır. Aynı şekilde, VS algoritması 0.046136 ortanca ve 0.001240 IQR değeriyle en iyi performansı sergilemiştir. Tiroit probleminde ortalama değerlerine göre algoritmaların performans VS, PSO ve ABC şeklinde sıralanır.

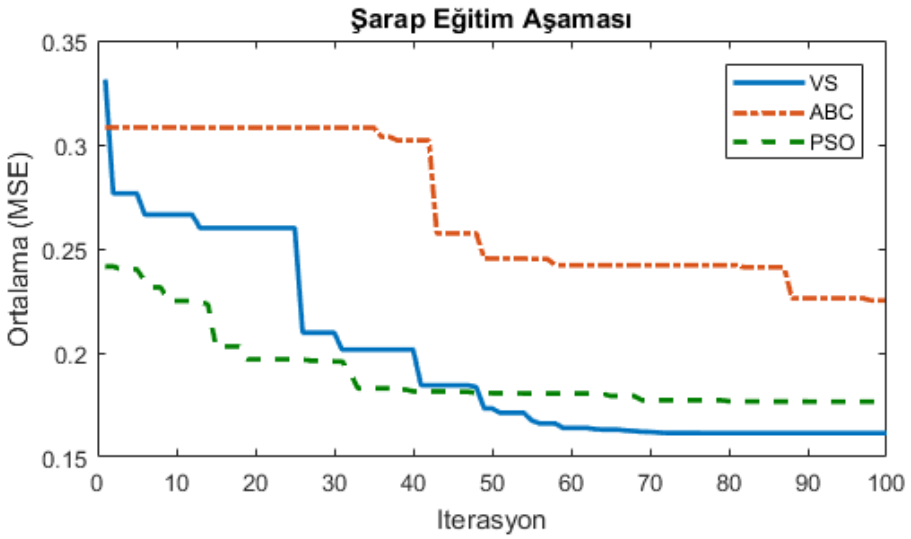
Test aşamasında ise %92 ortanca doğruluk değerleri ile VS ve PSO algoritmaların göstermiş oldukları performanslar birbirlerine çok yakındır. Ortalama %91.96 ile VS ve PSO aynı doğruluğa ulaşırken standart sapmaya bakıldığında VS 0.144097 değeri ile en iyi performans gösteren algoritmadır. Maks ve minimum değerlerini incelediğimizde maks değerleri tüm algoritmalar %92 performans göstermiş fakat küsurat farkıyla PSO %92.38 maks değeri ile diğer algoritmaları geçmiştir. min değer ise %91.54 VS algoritması iyi performans sergilemiştir. Şekil 4.18'de VS, ABC ve PSO algoritmalarının tiroit problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.

Şarap probleminde eğitim aşamasında, VS 0.166324 ortalama değeri ve 0.037729 standart sapma değeri ile tüm algoritmaları geride bırakmıştır. Aynı şekilde, 0.158283 ortanca ve 0.032233 IQR değerleri ile VS en iyi performansı sergilemiştir. PSO 0.17593 ortalama değeriyle ikinci, ABC ise 0.22291 ortalama değeri ile üçüncüdür.



Şekil 4.18. Tiroit Veri Seti için Çok-Çıkışlı ÇKA Ağlarının İkinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

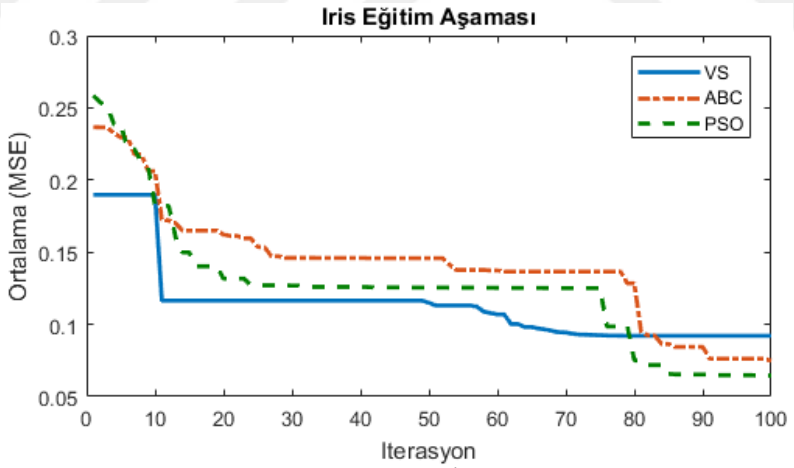
Test aşamasında ise, VS algoritmasının ortalama değeri ile ABC ve PSO'dan daha iyi eğitim performansı elde ettiği açıkça görülmektedir. ÇKA-VS modeli %44.84 ortalama doğruluk ve 22.75024 standart değerler ile en iyi performans gösteren algoritmadır. VS algoritması %57.95 ortanca doğruluk ve 25 IQR değeriyle en iyi performans sergilemiştir. Maks ve minimum değerlerini incelediğimizde %65.90 maks değeri ile VS iyi performans sergilemiştir. min değeri ise tüm algoritmalar 0 değer elde etmiştir. PSO %39.24 ortalama doğruluk değeriyle ikinci, ABC ise %22.12 ortalama doğruluk değeri ile üçüncüdür. Şekil 4.19'da VS, ABC ve PSO algoritmalarının şarap problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



Şekil 4.19. Şarap Veri Seti için Çok-Çıkışlı ÇKA Ağlarının İkinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

İris problemi için eğitim aşamasında, ortalama MSE değerlerine bağlı olarak VS algoritmasının göstermiş olduğu performans diğer veri setleri kadar iyi değildir. PSO algoritması 0.053993 ortanca ve 0.93542 IQR değerler ile en iyi performansı sergilemiştir. 0.066433 ortalama değeri ve 0.053925 standart sapma değeri ile PSO diğer algoritmaları az farkla geride bırakmıştır. ABC 0.076992, VS 0.086928 ortalama değerlerine ulaşmışlardır. İris probleminde ortalama performans değerlerine göre algoritmalar PSO, ABC ve VS şeklinde sıralanabilir.

Test aşamasında ise, ortalama doğruluk değerlerine bağlı olarak algoritmaların göstermiş oldukları performanslar birbirlerine yakındır. ABC algoritması %75.67 ortanca doğruluk ve 27.02702 IQR değerler ile en iyi performansı sergilemiştir. %73.78 ortalama doğruluk değeri ve 15.29126 standart sapma değeri ile ABC diğer algoritmaları az farkla geride bırakmıştır. Maksimum ve minimum değerlerini incelediğimizde maksimum değerleri tüm algoritmalar %97 performans göstermiştir. Minimum değer ise %48.65 VS algoritması diğer algoritmaları geride bırakmış ve iyi performans sergilemiştir. PSO %72.97, VS %68.46 ortalama doğruluk değerlerine ulaşmışlardır. Şekil 4.20'da VS, ABC ve PSO algoritmalarının iris problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



Şekil 4.20. İris Veri Seti için Çok çıkışlı ÇKA Ağlarının İkinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

4.2.2.2. Tek-Çıkışlı ÇKA Ağlarının İkinci Aşama Performans Değerlendirmesi

Tablo 4.8 incelendiğinde, tek-çıkışlı ÇKA ağlarının sınıflandırma test problemleri için eğitim aşamasında elde edilen MSE sonuçları görülmektedir. Algoritmaların 30'ar kez çalıştırılmaları sonucu eğitim aşamasında her bir problem için elde edilen ortalama ve ortanca MSE değerleri ve bunların IQR ve standart sapma değerleri Tablo 4.8'de verilmektedir.

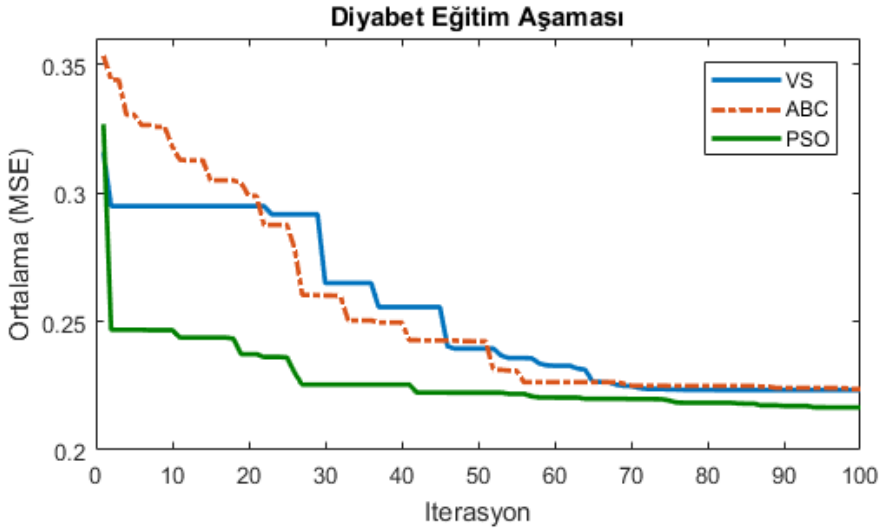
Tablo 4.8. İkinci Aşama Eğitim Süreci Sonunda 30 Bağımsız Çalıştırma için Tek çıkışlı ÇKA Ağlarında Algoritmaların Elde Ettiği MSE Değerleri

Veri Seti		VS	ABC	PSO
Diyabet	Ortanca	0.224809	0.224786	0.216343
	IQR	0.005371	0.007574	0.010509
	Ortalama	0.223267	0.223444	0.216601
	Std	0.005344	0.006255	0.007283
Kanser	Ortanca	0.226691	0.226755	0.226691
	IQR	3.053113e-16	2.369317e-04	4.581840e-10
	Ortalama	0.226691	0.227159	0.221965
	Std	3.450155e-16	0.001571	0.025674
Tiroit	Ortanca	0.9479381	0.950141	0.949596
	IQR	0.0061336	4.971256e-04	7.414149e-04
	Ortalama	0.9468119	0.949943	0.949295
	Std	0.0029572	5.218397e-04	0.001182
Şarap	Ortanca	1.5149253	1.5149253	1.5149253
	IQR	0	0	0
	Ortalama	1.5149253	1.5149253	1.5149253
	Std	9.033620e-16	1.393369e-13	9.033620e-16
Iris	Ortanca	0.3628318	0.3628318	0.3628318
	IQR	4.962696e-14	3.533839e-13	0
	Ortalama	0.3628318	0.3628318	0.3628318
	Std	2.728962e-10	9.675381e-10	6.478473e-13

Tek-Çıkışlı topolojide algoritmaların 30'ar kez çalıştırılmaları sonucu test aşamasında her bir problem için elde edilen ve yüzde olarak hesaplanan sınıflandırma doğruluklarının maksimum, minimum, ortalama ve ortanca değerleri ve bu değerlere ait IQR ve standart sapma değerleri Tablo 4.9'de verilmiştir. Sonuçlar incelendiğinde ortalama yüzde değerlerine göre VS algoritması diğer algoritmalarla rekabet halindedir.

Tablo 4.8'deki verilere göre eğitim aşaması için diyabet probleminde PSO algoritmasının ortalama 0.216601 MSE değeri ve 0.007283 standart sapma değerleri ile ABC ve VS algoritmalarından az farkla iyi bir performans elde ettiği görülmektedir. PSO 0.216343 ortanca ve 0.010509 IQR değerleri ile en iyi performans gösteren algoritmadır. VS 0.223267 ortalama değeriyle ikinci, ABC ise 0.223444 ortalama değeri ile üçüncüdür.

Test aşamasında ise VS algoritmasının %65.80 ortalama doğruluk ve 1.328456 standart sapma değerleri ile ABC ve PSO'dan daha iyi bir performansı elde ettiği görülmektedir. PSO %66.40 ortanca doğruluk ve 2.08333IQR değerleri ile en iyi performansı gösteren algoritmadır. Maks ve minimum değerlerini incelediğimizde %69.79 maks değeri ile ve %59.37 min değeri ile PSO algoritması iyi performans sergilemiştir. Bu anlamda VS ve PSO algoritmaları iyi ve yakın performanslar sergilemişlerdir. Şekil 4.21'da VS, ABC ve PSO algoritmalarının diyabet problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



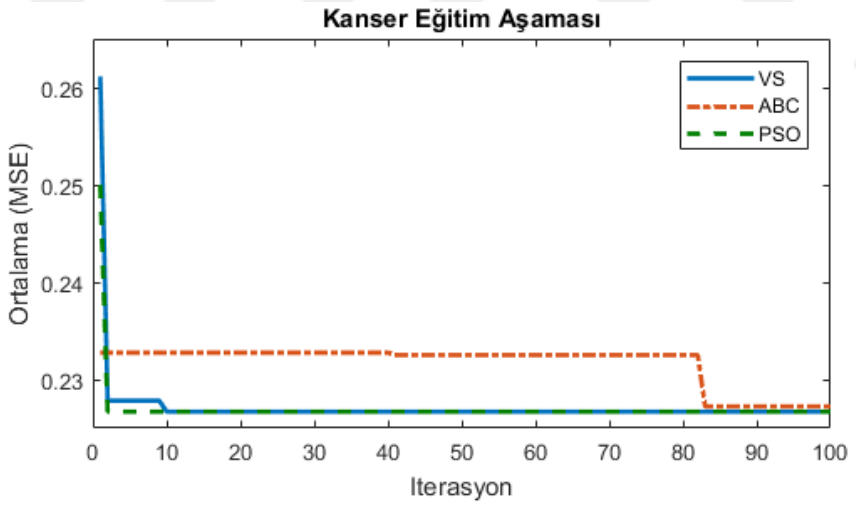
Şekil 4.21. Diyabet Veri Seti için Tek-Çıkışlı ÇKA Ağlarının İkinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

Tablo 4.9. İkinci Aşama Test İşlemi Sonunda 30 Bağımsız Çalıştırma için Tek-Çıkışlı ÇKA Ağlarında Algoritmaların Elde Ettiği Yüzde Olarak Sınıflandırma Doğrulukları

Veri Seti		VS	ABC	PSO
Diyabet	Ortanca	66.14	65.36	66.40
	IQR	1.562500	4.166666	2.083333
	Ortalama	65.80	64.80	65.62
	Std	1.328456	2.696576	2.454385
	Maks	67.19	68.75	69.79
	Min	54.69	57.81	59.37
Kanser	Ortanca	66.28	66.28	66.28
	IQR	0	0	0
	Ortalama	66.28	66.30	67.16
	Std	4.336137e-14	0.104328	4.799092
	Maks	66.28	66.85	92.57
	Min	66.28	66.28	66.28
Tiroit	Ortanca	5.812	5.561	5.609
	IQR	0.723025	0	0.055617
	Ortalama	5.893	5.587	5.609
	Std	0.327080	0.063216	0.104033
	Maks	6.40	5.78	5.95
	Min	5.56	5.56	5.56
Şarap	Ortanca	38.63	38.63	38.63
	IQR	0	0	0
	Ortalama	38.63	38.63	38.63
	Std	7.226896e-15	7.226896e-15	7.226896e-15
	Maks	38.63	38.63	38.63
	Min	38.63	38.63	38.63
Iris	Ortanca	75.67	75.67	75.67
	IQR	0	0	0
	Ortalama	75.67	75.67	75.67
	Std	2.890758e-14	2.890758e-14	2.890758e-14
	Maks	75.67	75.67	75.67
	Min	75.67	75.67	75.67

Kanser probleminin sonuçlarına bakıldığında; eğitim aşamasında algoritmaların ortalama ve ortanca değerleri birbirine yakın ve 0.22 civarındadır. Ortalama değerlerine bakıldığında en iyi sonucu 0.221965 ortalama MSE ve 0.025674 standart sapma değerleri ile PSO sahiptir. Ortanca değerleri incelendiğinde bütün algoritmalar aynı değeri elde etmiş ama VS 3.053113e-16 IQR değeriyle daha güvenli bir çalışma göstererek diğer algoritmaları geride bırakmıştır. Kanser veri setinde ortalama performans değerlerine göre algoritmalar PSO, VS ve ABC şeklinde sıralanır.

Test aşamasında ise, %67 ortalama ve 4.79909 standart sapma değerleri ile PSO az farkla diğer algoritmaları geride bırakmış ve iyi performans sergilemiştir. Ortanca ve IQR değerleri incelendiğinde bütün algoritmalar %66 ortanca ve 0 IQR değeriyle aynı performans göstermiştir. VS, ABC %66 ortalama değeriyle aynı ama standart sapma değer ise VS de 4.3361e-14 değeri ile iyi sonuç elde etmiştir. Maks ve minimum değerlerini incelediğimizde %92.57 maks değeri ile PSO iyi performans sergilemiştir. Min değeri ise tüm algoritmalar %66.28 değeri ile aynı performans elde etmiştir. Şekil 4.22'de VS, ABC ve PSO algoritmalarının kanser problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.

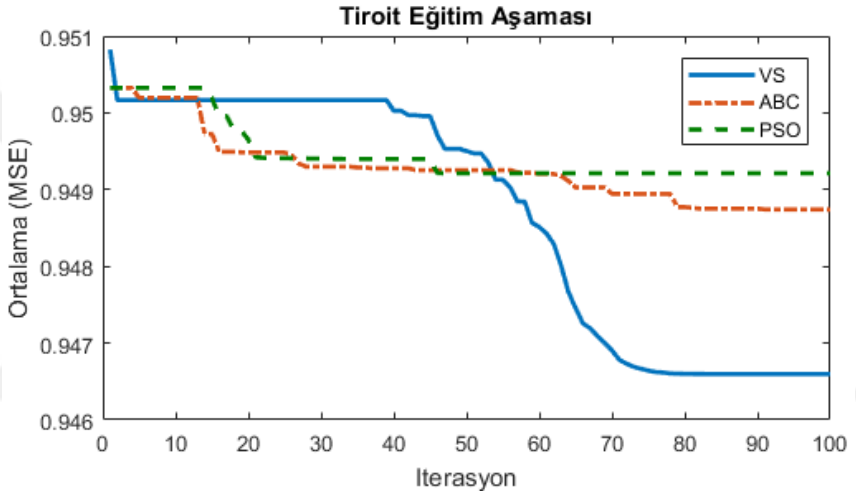


Şekil 4.22. Kanser Veri Seti için Tek-Çıkışlı ÇKA Ağlarının İkinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

Tiroit probleminde algoritmaların ortalama ve ortanca değerleri birbirine yakın ve 0.94 civarındadır. VS algoritmasının ABC ve PSO'dan daha iyi eğitim performansı elde ettiği görülmektedir. ABC algoritması 0.949943 ortalama MSE ve 5.218397e-04 standart sapma değerleri ile en iyi performansı sergilemiştir. VS algoritmasının 0.9479381 ortanca ve 0.0061336 IQR değerleri ile en iyi performans gösteren algoritma olmuştur. PSO

0.949295 ortalama değerine ulaşmıştır. Tiroit probleminde ortalama MSE değerlerine göre algoritmaların performansı VS, ABC ve PSO şeklinde sıralanır.

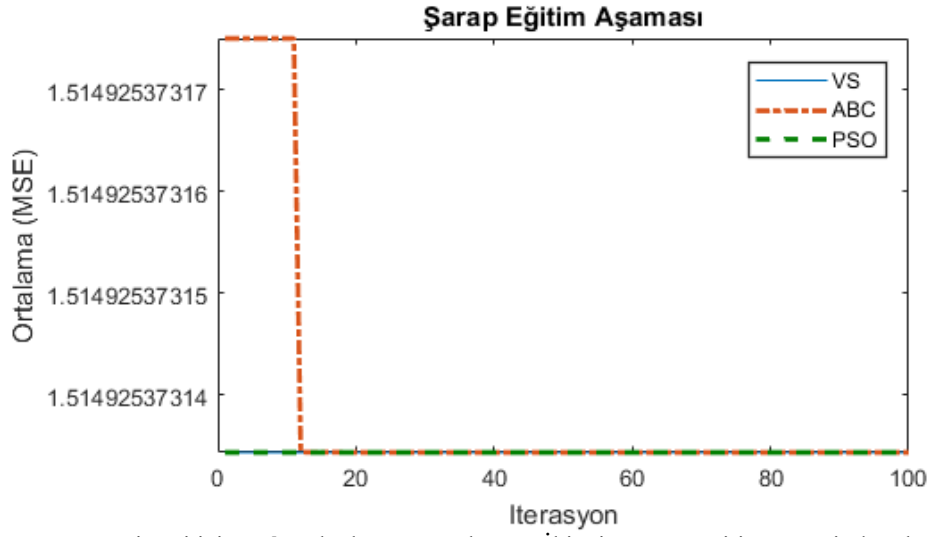
Test aşamasında ise, tüm algoritmalar düşük performans sergilemiştir. Ortanca ve ortalama MSE yaklaşık olarak %5.8 değerleri ile VS algoritması diğer algoritmalara göre daha iyi performans göstermiştir. IQR ve standart sapma değerleri sırasıyla 0.723025 ve 0.327080 olan VS en iyi performans gösteren algoritmadır. Maks ve minimum değerlerini incelediğimizde %6.40 maks değeri ile VS diğer algoritmalara göre iyi performans sergilemiştir. Min değeri ise tüm algoritmalar %5.56 değeri ile aynı performans elde etmiştir. Şekil 4.23'da VS, ABC ve PSO algoritmalarının tiroit problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



Şekil 4.23. Tiroit Veri Seti için Tek-Çıkışlı ÇKA Ağlarının İkinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

Şarap probleminde eğitim aşamasında, ortalama MSE değerlerine bağlı olarak algoritmaların göstermiş oldukları performanslar diğer veri setleri kadar iyi değildir. VS, ABC ve PSO algoritmalarının başarısız olduğu ve tüm algoritmalar aynı ortalama 1.5149 değerliyle oranında olduğu görülmektedir.

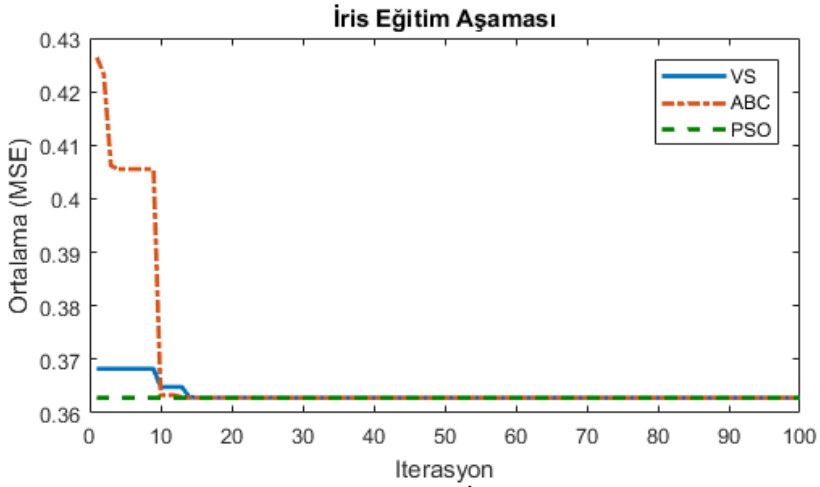
Test aşamasında ise, ortalama MSE değerlerine bağlı olarak algoritmaların göstermiş oldukları performanslar diğer veri setleri kadar iyi değildir. VS, ABC ve PSO algoritmalarının başarısız olduğu ve tüm algoritmalar aynı ortalama %38 doğruluk elde edebilmişlerdir. Şekil 4.24'te VS, ABC ve PSO algoritmalarının şarap problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



Şekil 4.24. Şarap Veri Seti için Tek-Çıkışlı ÇKA Ağlarının İkinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

İris veri setinde eğitim aşamasında, ortalama değerlerine bağlı olarak algoritmaların göstermiş oldukları performanslar tamamının aynı başarılı sonuçlara ulaşarak çok benzer performanslar sergiledikleri görülmektedir. PSO algoritması IQR ve std değerleri sırasıyla 0 ve 6.4784e-13 değerler ile iyi performans sergilemiştir. VS ve ABC ise 0.362831 ortalama değerlerine ulaşmışlardır.

Test aşamasında ise, ortalama değerlerine bağlı olarak algoritmaların göstermiş oldukları performanslar tamamının aynı başarılı sonuçlara ulaşarak çok benzer performanslar sergiledikleri görülmektedir. Tüm algoritmalar %75 ortalama doğruluk değerine ulaşmışlardır. Şekil 4.25'te VS, ABC ve PSO algoritmalarının iris problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



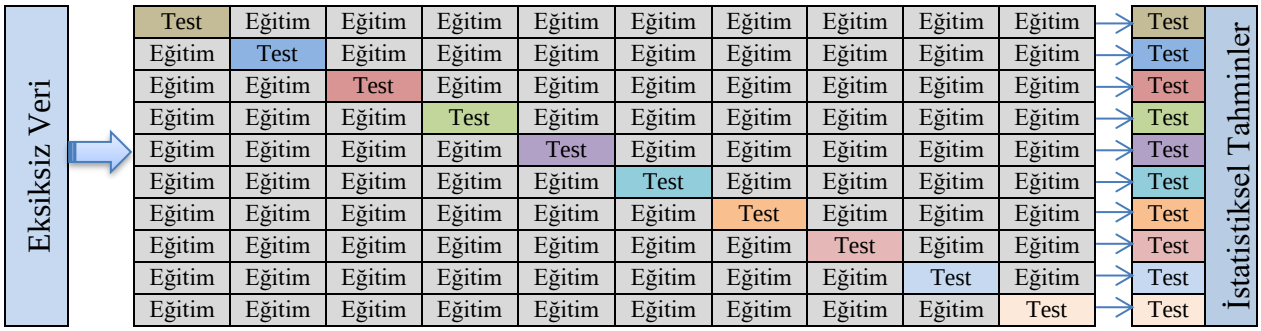
Şekil 4.25. İris Veri Seti için Tek-Çıkışlı ÇKA Ağlarının İkinci Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

Sonuç olarak, sınıflandırma hatalarında görüldüğü üzere VS algoritması yapay sinir ağlarının eğitilmesinde en az diğer algoritmalar kadar iyi bir performansa sahiptir. Buna rağmen VS algoritmasını tüm problemlerde en iyi performansı göstermemiştir. En iyi performanstan düşük performans sergileyene göre algoritmalar; VS, PSO, ABC şeklinde sıralanabilir. VS algoritması çok-çıkışlı yapay sinir ağı tek-çıkışlı yapay sinir ağına kıyasla daha iyi performans sergilediğini görülmektedir. Özellikle diyabet, şarap ve iris verilerinde çok-çıkışlı ileri beslemeli ağ yapısında tek-çıkışlı ağ yapısına kıyasla algoritmalar daha başarılı ve verimli sonuçlar elde etmiştir.

4.2.3. Üçüncü Aşama Performans Değerlendirmesi

Bu aşamada, test problemleri üzerinde eğitim işleminin doğruluğunu analiz etmek amacıyla k-katlamalı çapraz doğrulama (k-fold cross validation) yöntemi kullanıldı. Çapraz doğrulama (Sato ve ark., 2013), tahmini bir modelin performansını değerlendirmenin bir ölçümüdür. İstatistiksel analiz bağımsız bir veri kümesine geneleme yapar. Yinelenen rasgele alt örnekleme doğrulama, K katlama çapraz doğrulama, K * 2 çapraz doğrulama, bir defaya mahsus çapraz doğrulama vb. türleri vardır. Bu yöntem yapay sinir ağı ve veri madenciliği gibi birçok çalışmada kullanılmaktadır (Shao ve ark., 2013). Bu çalışmada, modelin parametrelerini seçmek için K-katlamalı çapraz doğrulama seçildi. K değeri olarak 10 kullanıldı.

Yapay sinir ağlarında genelleme hatasını tahmin etmek için sıkça kullanılan yöntem çapraz doğrulama yöntemidir. K katlamalı çapraz doğrulama, orijinal örneği rasgele K alt örneklerine bölme tekniğidir. Veri seti rastgele eşit büyüklükte k ($k \geq 2$) alt kümeye bölünür. Model, bir tanesi dışındaki tüm alt grupları eğitim seti olarak atar ve geriye kalan alt küme de test seti olarak kullanılır. Bu işlem, k kez tekrarlanır ve her seferinde doğrulama için farklı bir alt küme kullanılır (Lu ve ark., 2001). k adım sonra elde edilen sınıflandırma sonuçlarının ortalaması alınarak genel sınıflandırma sonucu elde edilmektedir. Veri setini eğitim ve test seti olarak ayırmamızın amacı, olası ağ ezberlemesinden kaçınmak ve modelin daha önceden görmediği veri seti üzerinde nasıl performans gösterdiğini anlamak içindir (Jiang ve Chen, 2016).



Şekil 4.26. K katlamalı çapraz doğrulamasının akış şeması (Jiang ve Chen, 2016)

Uygulamalarda veri seti ağa eğitim seti ve test seti olmak üzere Çapraz Doğrulama yöntemi uygulanır. k parametresi 10 olarak belirlenen bir çapraz doğrulama işlemi uygulanır. 10 parçaya bölünen veri kümesindeki 9 parça eğitim kümesi ve geriye kalan diğer parça test kümesi olarak kullanılır ve bu işlem 10 adımda ve her adımda farklı bir parça test kümesi alınarak gerçekleştirilir. Şekil 4.26' da K katlamalı çapraz doğrulama akış şeması göstermektedir. Eğitim verileriyle sinir ağı eğitildikten sonra, ÇKA'nin görmediği test verileri ağa verilir ve ağın bu verilere vereceği cevabı hesaplanır. Deneyler her bir problem için farklı sayılarıyla oluşturulan rasgele popülasyonlar ile 30 kez tekrarlandı.

Eğitim aşamasında kullanılan algoritmaların parametreleri belirlenirken maksimum jenerasyon sayısı 100 olarak ayarlandı. Tüm parametrelerin sınır aralıkları $[-20, 20]$ alındı. VS parametrelerinin ayarları ise popülasyon büyüklüğü 30 ve limit değeri 100 olarak seçildi. ABC algoritması için koloni büyüklüğü 30 ve limit değeri 100 şeklindedir. PSO algoritması için sürü büyüklüğü 30, eylemsizlik faktörü 0.6 ve hızlandırma faktörü 1.8 değerleri şeklinde ayarlandı.

Tüm algoritmalar, 100 yinelemeden oluşan k katlamalı çapraz doğrulama yöntemini kullanarak çalıştırıldı. Bu işlem k=10 için uygulandı. Her veri seti için tüm algoritmalar 30 kez tekrarlandı. Her seferinde aynı ayarlarla ve farklı rasgele popülasyonlar ile başlatıldı. Sonuçta çapraz doğrulama yöntemini kullandıktan sonra elde edilen sınıflandırma sonuçlarının ortalaması, maksimum ve minimum değerleri hesaplandı. K-Katlamalı Çapraz Doğrulama yöntemini kullanarak Çok çıkışlı ÇKA ve Tek çıkışlı ÇKA ağlarının sınıflandırma test problemleri için eğitilmesinde elde edilen sonuçlar Tablo 4.10'da ve Tablo 4.12'da verilmiştir.

4.2.3.1. Çok-Çıkışlı ÇKA Ağlarının Üçüncü Aşama Performans Değerlendirmesi

K-Katlamalı Çapraz Doğrulama yöntemini kullanarak Çok-Çıkışlı ÇKA ağlarının sınıflandırma test problemleri için eğitilmesinde elde edilen sonuçlar Tablo 4.10'de verilmiştir. Diyabet problemi için VS algoritmasının ABC ve PSO'dan daha iyi bir eğitim performansı elde ettiği açıkça görülmektedir. VS %63.91 ortalama sınıflandırma doğruluğu ile en iyi performans gösteren algoritmadır. PSO %60.83 ortalama doğruluk değeriyle ikinci, ABC ise %57 ortalama doğruluk değeri ile üçüncüdür. Minimum değerlerini incelediğimizde VS %54, PSO %35, ABC %24 değerleri ile VS algoritması en iyi performans gösteren algoritmadır. Maksimum değeri ise tüm algoritmalar %75 değeri ile aynı performansı sergilemiştir. Sonuç olarak, diyabet veri setinde ÇKA-VS modeli algoritmalar arasında en iyi ortalama doğruluğu ile daha başarılı sonuçlar elde edilmiştir.

Tablo 4.10. K-Katlamalı-Çapraz-Doğrulama Yöntemi ile Eğitilen Çok-Çıkışlı ÇKA Ağlarının Sınıflandırma Doğruluk Değerleri

	VS			ABC			PSO		
	Min (%)	Maks (%)	Ort (%)	Min (%)	Maks (%)	Ort (%)	Min (%)	Maks (%)	Ort (%)
Diyabet	54.85	75.32	63.91	24.64	75.32	57.19	35.68	75.32	60.83
Kanser	61.43	74.29	65.23	45.05	74.28	62.53	57.90	75.61	65.04
Tiroit	91.01	94.16	92.52	90.75	94.15	92.44	90.99	94.15	92.50
Şarap	3.88	88.23	46.15	0	88.23	27.61	4.63	88.23	45.42
Iris	42.44	100	75.29	42.66	100	76.60	44.66	100	77.24

Kanser problemi için Tablo 4.10 incelendiğinde algoritmaların ortalama doğruluk değerlerinin birbirine çok yakın olduğu görülür. VS az bir farkla %65.23 ortalama doğruluk değeri ile en iyi performans gösteren algoritmadır. Hemen ardından %65.04 ortalama doğruluk değeri ile PSO gelir. ABC ise %62 ortalama değeri ile üçüncüdür. Minimum doğruluk değerlerine bakıldığında; VS %61.43, PSO %57.90, ABC %45.05 değerleri ile VS algoritması yine en iyi performans gösteren algoritmadır. Maksimum değeri ise VS ve ABC algoritmaları yaklaşık olarak %74 doğruluk değeri ile yakın performanslar sergilemiştir. PSO %75 ile az farkla diğer algoritmaları geride bırakmıştır.

Tiroit problemi için VS algoritması %92.52 ortalama doğruluk değeri ile az bir farkla ilk sırada yer alır. Onu sırasıyla %92.50 ve %92.44 ortalama doğruluk değerleriyle PSO ve ABC takip etmiştir. Maksimum doğruluk değerinde ise tüm algoritmalar yaklaşık %94 değeri ile yakın performanslar sergilerken; minimum doğruluk değerleri incelendiğinde VS %91, PSO %90, ABC %90 değerleri ile VS algoritması çok az farklarla önde gelmektedir.

Şarap probleminde VS %46.15 ortalama doğruluk değeri ile tüm algoritmaları geride bırakmıştır. PSO algoritması %45.42 ortalama değeriyle ikinci, ABC %27.61 ortalama değeri üçüncüdür. Minimum değerlerini incelediğimizde VS %3 PSO %0 ABC %4 değerleri ile tüm algoritmalar kötü performans göstermiştir. Maksimum değeri ise tüm algoritmalar %88.23 değeri ile aynı performansı sergilemiştir.

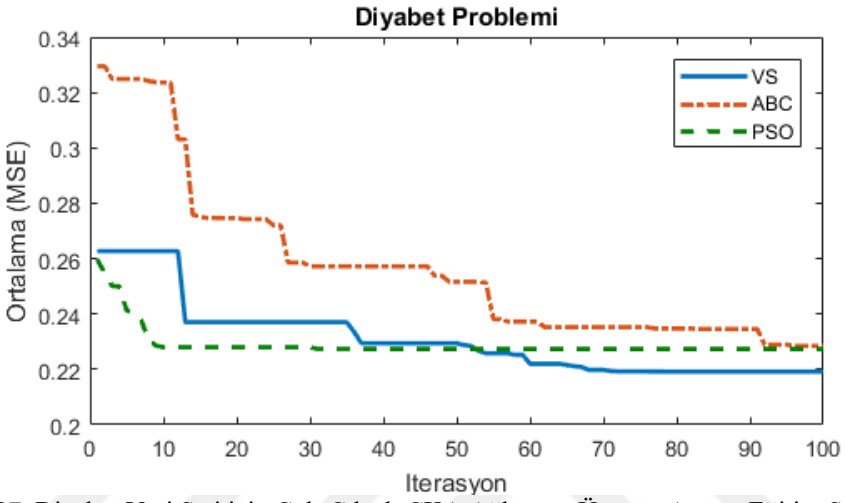
İris probleminde ortalama değerlerine bağlı olarak algoritmaların göstermiş oldukları performanslar birbirine yakındır. PSO %77.24 ortalama doğruluk değeri ile diğer algoritmaları az farkla geride bırakmıştır. ABC %76.60, VS %75.29 ortalama doğruluk değerlerini üretmişlerdir. Minimum değerlerini incelediğimizde PSO %44 değeri ile en iyi performans gösterir. Maksimum değeri ise tüm algoritmalar %100 değeri ile aynı performansı sergilemiştir.

K-Katlamalı Çapraz Doğrulama yöntemini kullanarak, algoritmaların otuzar kez çalıştırılmaları sonucu çok çıkışlı ÇKA ağlarının sınıflandırma test problemleri için eğitilmesinden elde edilen ortalama ve ortanca MSE hata değerleri ve bunların IQR ve standart sapma değerleri Tablo 4.11'da verilmiştir. Sonuçlar incelendiğinde ortalama MSE değerlerine göre VS algoritmasının problemlerin geneli üzerinde daha başarılı sonuçlar verdiği görülür.

Tablo 4.11. Üçüncü Aşama Eğitim Süreci Sonunda Çapraz doğrulama yöntemi Kullanılarak yapılan 30 Bağımsız Çalıştırma için Çok-Çıkışlı ÇKA Ağlarında Algoritmaların Elde Ettiği MSE Değerleri

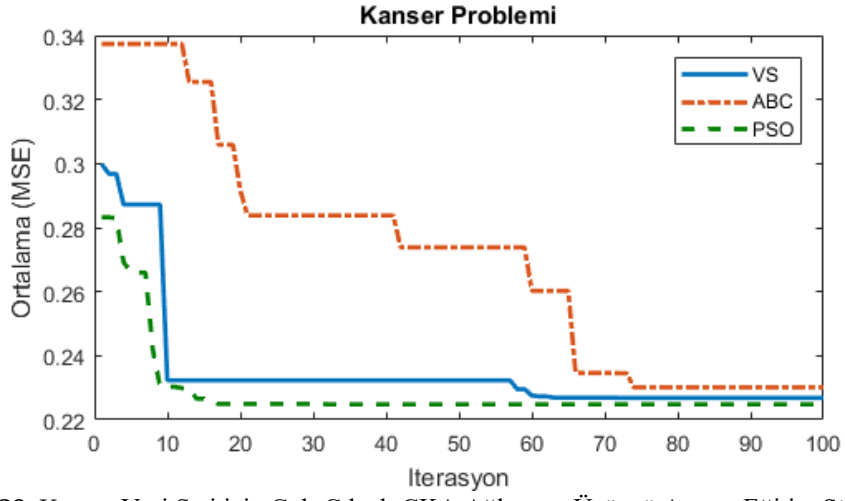
Database		VS	ABC	PSO
Diyabet	Ortanca	0.229465	0.243459	0.234252
	IQR	0.012356	0.022667	0.030991
	Ortalama	0.234963	0.246528	0.241369
	Std	0.015204	0.014890	0.022421
Kanser	Ortanca	0.225733	0.230905	0.225853
	IQR	0.001663	0.010101	0.002136
	Ortalama	0.226816	0.233826	0.224899
	Std	0.001361	0.008561	0.011126
Tiroit	Ortanca	0.047279	0.049233	0.047622
	IQR	0.001600	0.001309	0.002490
	Ortalama	0.047070	0.049291	0.047286
	Std	0.001410	0.001133	0.001937
Şarap	Ortanca	0.179883	0.224092	0.170547
	IQR	0.046701	0.039355	0.052387
	Ortalama	0.149883	0.220635	0.174864
	Std	0.032634	0.027707	0.034943
Iris	Ortanca	0.068085	0.058504	0.059950
	IQR	0.083122	0.058714	0.084919
	Ortalama	0.071915	0.065496	0.068098
	Std	0.050442	0.037127	0.052844

Eğitim süreci sonunda Diyabet problemi için VS algoritmasının ABC ve PSO'dan daha iyi eğitim performansı elde ettiği açıkça görülmektedir. VS için 0.234963 ortalama MSE ve 0.015204 standart sapma değerleri ile en iyi performans gösteren algoritmadır. Aynı şekilde, 0.229465 ortanca ve 0.012356 IQR değerleri değerler ile VS en iyi performans gösteren algoritmadır. PSO 0.241369 ortalama değeriyle ikinci, ABC ise 0.246528 ortalama değeri ile üçüncüdür. Şekil 4.27'de VS, ABC ve PSO algoritmalarının diyabet problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



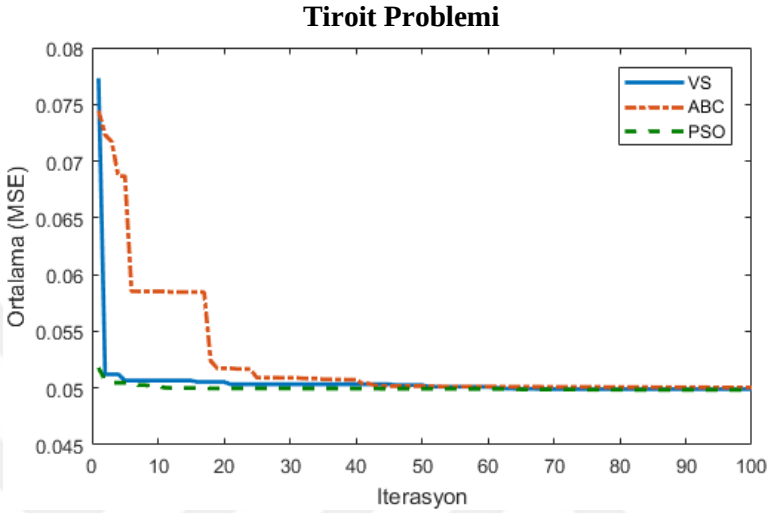
Şekil 4.27. Diyabet Veri Seti için Çok-Çıkışlı ÇKA Ağlarının Üçüncü Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

Kanser probleminde VS ve PSO ortalama değerleri birbirine çok yakın performanslar sergilemiştir. VS 0.225733 ortanca ve 0.001663 IQR değerleriyle diğer algoritmaları geride bırakmış ve iyi performansı sergilemiştir. PSO 0.226399 ortalama ve 0.011126 standart sapma değerleriyle ikinci, ABC ise 0.233826 ortalama değeri ile üçüncüdür. Şekil 4.28'de VS, ABC ve PSO algoritmalarının kanser problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



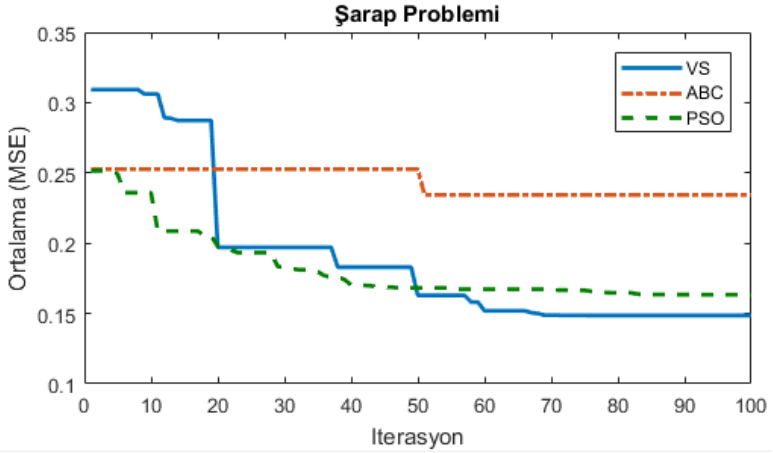
Şekil 4.28. Kanser Veri Seti için Çok-Çıkışlı ÇKA Ağlarının Üçüncü Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

Tiroit probleminde algoritmaların ortalama değerleri birbirine çok yakın ve 0.04 civarındadır. VS algoritması, bu problemde 0.047070 ortalama ve 0.047279 ortanca oranıyla en iyi performansı göstermiştir. Tiroit probleminde performans değerlerine göre algoritmaların sıralaması şu şekildedir: VS, PSO, ABC. Şekil 4.29'da VS, ABC ve PSO algoritmalarının Tiroit problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



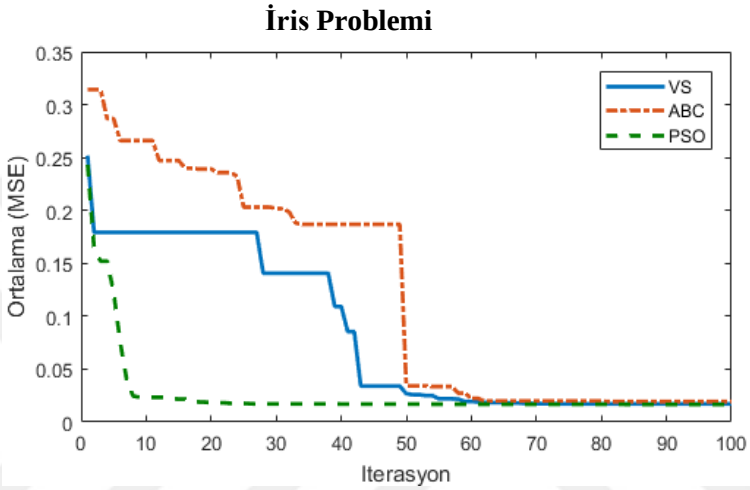
Şekil 4.29. Tiroit Veri Seti için Çok-Çıkışlı ÇKA Ağlarının Üçüncü Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

Şarap probleminde PSO ve VS algoritmasının ortanca değeri birbirine yakın değerler elde etmiştir ama ortalama değerlerine bakacak olursak VS 0.149883 değeriyle diğer algoritmaları geride bırakmıştır. PSO algoritmasının ortanca ve IQR değerleri sırasıyla 0.170547 ve 0.052387'dir. Bu değerler diğer algoritmalardan az farkla daha iyidir. Bu kapsamda performans değerlerine göre algoritmaların sıralaması şu şekildedir: VS, PSO, ABC. Şekil 4.30'da VS, ABC ve PSO algoritmalarının şarap problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



Şekil 4.30. Şarap Veri seti için Çok-Çıkışlı ÇKA ağlarının Üçüncü Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

İris probleminde algoritmaların göstermiş oldukları performanslar birbirine yakındır ama ortalama MSE değerlerine bakılırsa ABC 0.065496 ortalama değeri ve 0.037127 standart sapma değeri ile diğer algoritmaları az farkla geride bırakmıştır. PSO 0.068098, VS 0.071915 ortalama değerlerine ulaşmışlardır. ABC algoritmasının ortanca ve IQR değerleri sırasıyla 0.058504 ve 0.058714'tür. Bu değerler ile diğer algoritmalardan az farkla daha iyi performans sergilemiştir. Şekil 4.31'de VS, ABC ve PSO algoritmalarının iris problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



Şekil 4.31. İris Veri Seti için Çok-Çıkışlı ÇKA Ağlarının Üçüncü Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

4.2.3.2. Tek-Çıkışlı ÇKA Ağlarının Üçüncü Aşama Performans Değerlendirmesi

K-Katlamalı Çapraz Doğrulama yöntemini kullanarak Tek-Çıkışlı ÇKA ağlarının sınıflandırma test problemleri için eğitilmesinde elde edilen sonuçlar Tablo 4.12'de verilmiştir. Diyabet problemi için tüm algoritmalar yaklaşık %64 doğruluk değerleriyle birbirine yakın eğitim performansı elde etmiştir. Bununla birlikte VS az farkla önde gelmektedir. Minimum değerlerini incelediğimizde VS %56, PSO %55, ABC %55 değerleri ile VS algoritması daha iyi performans gösteren algoritmadır. Maksimum değeri ise VS ve ABC algoritmaları %76.62 ile aynı performansı sergilemiştir ama PSO %79 diğer algoritmaları geçmiştir. Sonuç olarak, diyabet veri setinde ÇKA-VS modeli algoritmalar arasında en iyi ortalama doğruluğu ile daha başarılı sonuçlar elde edilmiştir.

Tablo 4.12. K-Katlamalı-Çapraz-Doğrulama Yöntemi ile Eğitilen Tek-Çıkışlı ÇKA Ağlarının Sınıflandırma Doğruluk Değerleri

	VS			ABC			PSO		
	Min (%)	Maks (%)	Ort (%)	Min (%)	Maks (%)	Ort (%)	Min (%)	Maks (%)	Ort (%)
Diyabet	56.31	76.62	64.90	55.14	76.62	64.53	55.32	79.22	64.49
Kanser	57.33	95.71	64.14	57.14	71.43	63.51	57.48	88.57	64.69
Tiroit	4.005	7.093	5.45	3.908	6.258	5.134	3.95	7.093	5.28
Şarap	16.67	50	33.27	16.67	50	33.26	16.67	50	33.26
Iris	46.67	86.67	66.58	46.67	86.67	66.62	46.67	86.67	66.63

Kanser problemi için Tablo 4.12 incelendiğinde, algoritmaların ortalama doğruluk değerleri birbirine oldukça yakındır. VS ile PSO %64 ortalama değeri ile en iyi performans gösteren algoritmalardır. ABC ise %63 ortalama değeri ile üçüncüdür. Minimum değeri ise tüm algoritmalar %57 değeri ile aynı performansı sergilemişlerdir. Maksimum değerlerini incelediğimizde VS %95, PSO %88, ABC %71 değerleri ile VS algoritması en iyi performans gösteren algoritmadır.

Tiroit probleminde ortalama değerlerine bağlı olarak algoritmaların göstermiş oldukları performanslar diğer veri setleri kadar iyi değildir. VS, ABC ve PSO algoritmalarının başarısız olduğu ve tüm algoritmalar %5 ortalama oranlarında başarı olduğu görülmektedir. Minimum değerlerini incelediğimizde VS %4, PSO %3, ABC %3 doğruluk değerleri algoritmaların tamamı başarısız olmuştur. Maksimum değeri ise VS ve PSO algoritmaları %7 değeri ile aynı performansı sergilemiştir. ABC %6 değeri elde etmiştir.

Şarap problemine incelediğimizde VS, ABC ve PSO algoritmalarının %33 oranlarında ortalama değerleri görülmektedir. Minimum değerlerini incelediğimizde tüm algoritmalar %16 değerleri performans sergilemişlerdir. Aynı şekilde, maksimum değeri tüm algoritmalar %50 değerleri ile performans elde etmiştir.

İris probleminde ise tüm algoritmalar %66 ortalama değerlerini üretmişlerdir. Minimum değerlerini incelediğimizde tüm algoritmalar %46 değerleri performans sergilemişlerdir. Aynı şekilde, Maksimum değeri tüm algoritmalar %86 değerleri ile performans elde etmiştir.

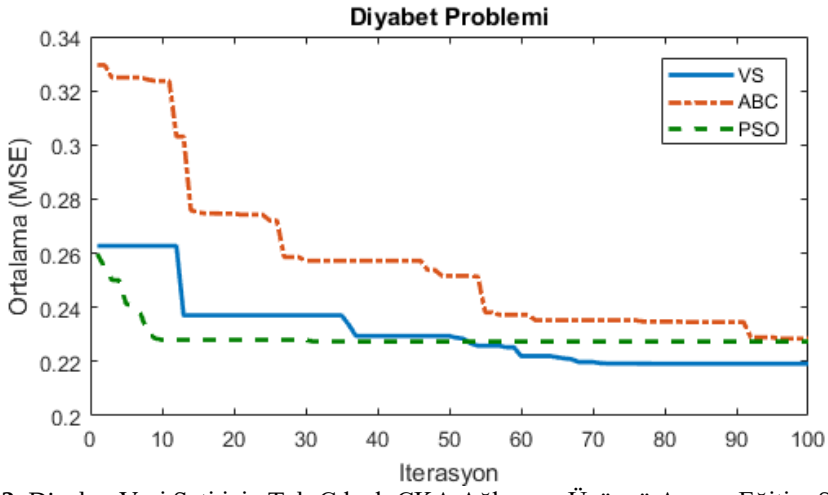
K-Katlamalı Çapraz Doğrulama yöntemini kullanarak, algoritmaların otuzar kez çalıştırılmaları sonucu Tek-Çıkışlı ÇKA ağlarının sınıflandırma test problemleri için

eğitilmesinden elde edilen ortalama ve ortanca MSE hata değerleri ve bunların IQR ve standart sapma değerleri Tablo 4.13'de verilmiştir.

Tablo 4.13. Üçüncü Aşama Eğitim Süreci Sonunda Çapraz doğrulama yöntemi Kullanılarak yapılan 30 Bağımsız Çalıştırma için Tek-Çıkışlı ÇKA Ağlarında Algoritmaların Elde Ettiği MSE Değerleri

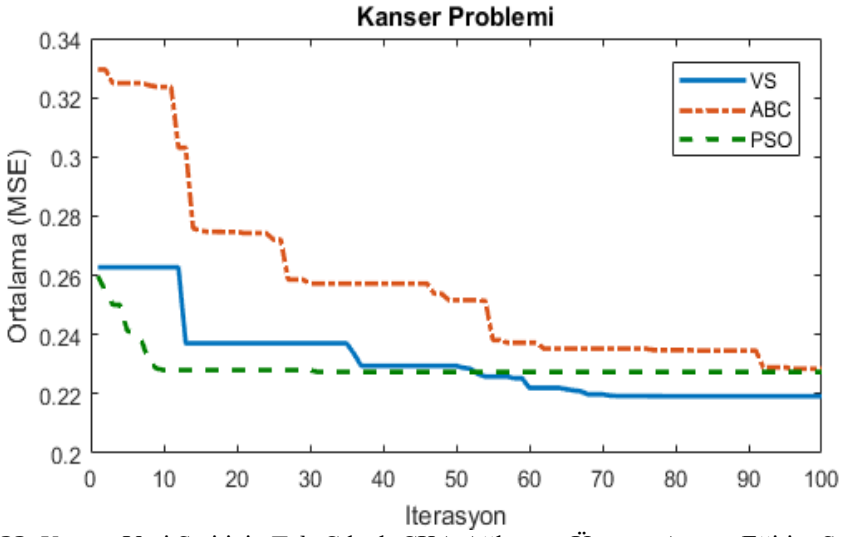
Database		VS	ABC	PSO
Diyabet	Ortanca	0.222277	0.224720	0.217874
	IQR	0.007998	0.007822	0.009212
	Ortalama	0.221166	0.224779	0.217406
	Std	0.005943	0.006463	0.007660
Kanser	Ortanca	0.227028	0.227673	0.227093
	IQR	0.001891	0.002007	0.002477
	Ortalama	0.224896	0.227978	0.221472
	Std	0.009068	0.001612	0.018418
Tiroit	Ortanca	0.945557	0.948671	0.947785
	IQR	0.005708	0.001422	0.003399
	Ortalama	0.945285	0.948616	0.946880
	Std	0.003397	9.460742e-04	0.002749
Şarap	Ortanca	1.462499	1.462499	1.462499
	IQR	0.062500	0.062500	0.062500
	Ortalama	1.477465	1.477465	1.477465
	Std	0.038015	0.038015	0.038015
Iris	Ortanca	0.333353	0.333334	0.333333
	IQR	0.007461	0.007407	0.007408
	Ortalama	0.333337	0.333334	0.333335
	Std	0.012025	0.012024	0.012023

Diyabet problemi için PSO algoritması 0.217406 ortalama doğruluk değeriyle diğer algoritmaları az farkla geride bırakmıştır. VS 0.221166 ortalama ve 0.222277 ortanca değerler ile ikinci, ABC ise 0.224779 ortalama değeri ile üçüncüdür. Diyabet probleminde performans değerlerine göre algoritmaların sıralaması şu şekildedir: PSO, VS, ABC. Şekil 4.32'de VS, ABC ve PSO algoritmalarının diyabet problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



Şekil 4.32. Diyabet Veri Seti için Tek-Çıkışlı ÇKA Ağlarının Üçüncü Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

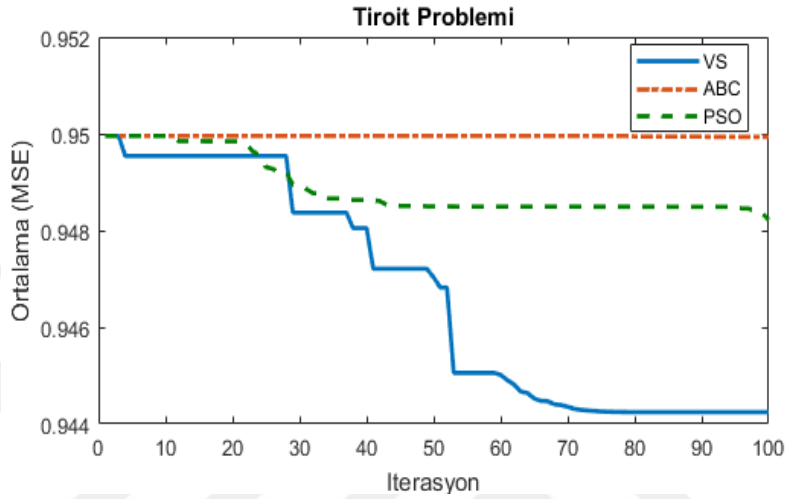
Kanser probleminde en iyi performansı gösteren algoritma 0.227028 ortanca doğruluk değeri ve 0.001891 IQR değeri ile VS'dir. Algoritmaların ortalama değerleri birbirine çok yakın ve 0.22 civarındadır. PSO'nun ortanca doğruluğu ve standart sapma değerleri sırasıyla 0.221472 ve 0.018418'dir. Bu değerler ile diğer algoritmaları geride bırakmıştır. VS 0.224896 ortalama değeriyle ikinci, ABC ise 0.227978 ortalama değeri ile üçüncüdür. Kanser veri setinde algoritmaların performans değerlerine birbirine az farkla şu şekildedir: PSO, VS, ABC sırlanmaktadır. Şekil 4.33'de VS, ABC ve PSO algoritmalarının kanser problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



Şekil 4.33. Kanser Veri Seti için Tek-Çıkışlı ÇKA Ağlarının Üçüncü Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

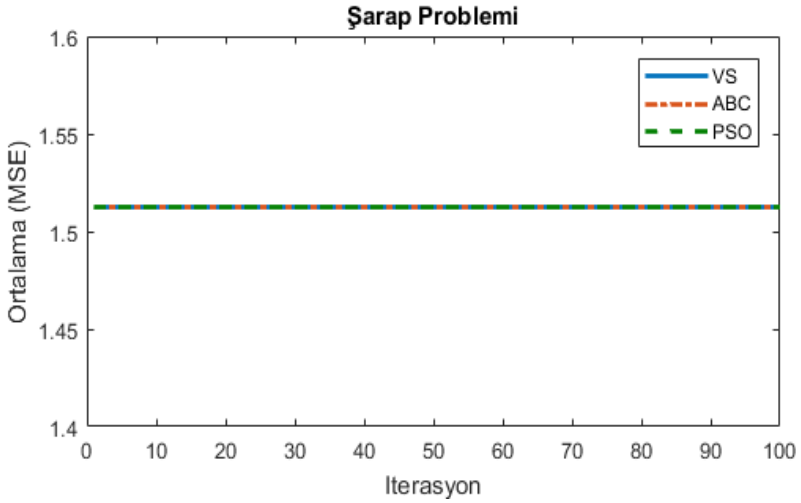
Tiroit probleminde algoritmaların ortalama değerleri birbirine çok yakın ve 0.94 civarındadır. VS algoritması, bu problemde en iyi sonucu 0.945285 ve 0.945557 ortalama

ve ortanca MSE değeriyle en iyi performansı göstermiştir. Aynı şekilde, VS algoritması IQR ve standart sapma değerleri sırasıyla 0.005708 ve 0.003397'dir. Bu değerler ile en iyi performansı sergilemiş diğer algoritmaları geride bırakmıştır. Tiroit probleminde performans değerlerine göre algoritmaların sıralaması su şekildedir: VS, PSO, ABC. Şekil 4.34'de VS, ABC ve PSO algoritmalarının tiroit problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



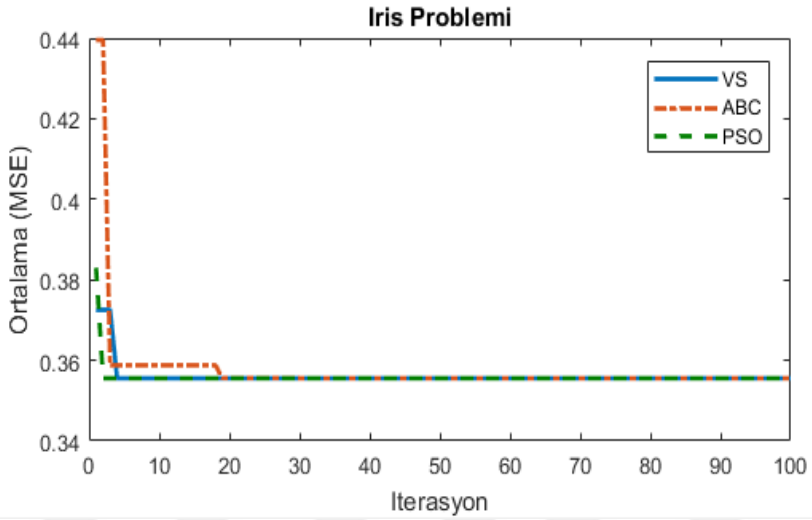
Şekil 4.34. Tiroit Veri Seti için Tek-Çıkışlı ÇKA Ağlarının Üçüncü Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

Şarap probleminde ortalama MSE değerlerine bağlı olarak algoritmaların göstermiş oldukları performanslar diğer veri setleri kadar iyi değildir. VS, ABC ve PSO algoritmalarının başarısız olduğu ve tüm algoritmalar aynı ortalama 1.477465 değerliyle oranında olduğu görülmektedir. Şekil 4.35'de VS, ABC ve PSO algoritmalarının şarap problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



Şekil 4.35. Şarap Veri Seti için Tek-Çıkışlı ÇKA Ağlarının Üçüncü Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

İçerdiği örnek sayısının az olması ve gürültünün hemen hemen hiç olmaması nedeniyle sınıflandırması en kolay veri setlerinden birisi olan İris veri setinde ortalama MSE değerlerine bağlı olarak algoritmalar tamamının aynı başarılı sonuçlara ulaşarak çok benzer performanslar sergiledikleri görülmektedir. Şekil 4.36'da VS, ABC ve PSO algoritmalarının iris problemi için eğitim aşamasındaki yakınsama grafikleri verilmiştir.



Şekil 4.36. İris Veri Seti için Tek-Çıkışlı ÇKA Ağlarının Üçüncü Aşama Eğitim Sürecinde Algoritmaların Yakınsama Grafikleri

Sonuçlar incelendiğinde ortalama değerlerine göre VS algoritması problemlerin en az diğer meta-sezgisel optimizasyon algoritmaları kadar iyi bir performansa sahiptir. Buna rağmen VS algoritmasını tüm problemlerde en iyi performansı gösterdiği söylenemez. Tablo 4.11 ve 4.13 deki sınıflandırma hatalarında görüldüğü üzere tek çıkışlı ÇKA ağları çok çıkışlı ÇKA ağlarına kıyasladığımızda VS algoritması çok çıkışlı ÇKA ağlarında daha iyi sonuçlar elde etmiştir. Özellikle İris ve Şarap probleminde çok çıkışlı ÇKA ağlarının eğitilmesinde tek çıkışlıya göre daha iyi performans sergilemiştir.

4.3. Bölüm Değerlendirmesi

Bu bölümde YSA eğitimi ve optimizasyon algoritmaları arasındaki ilişki hakkında bilgi verildikten sonra, çalışmada kullanılan veri setleri detaylı bir şekilde açıklandı. Bu veri setleri üzerinde yapılan çalıştırma sonuçları ve izlenen metodoloji detaylı bir şekilde ele alındı. Ulaşılan çalıştırma sonuçları için ortalama ve standart sapma değerleri ile ortanca ve IQR değerleri birlikte hesaplandı. Ayrıca tüm algoritmalar için yakınsama grafikleri verildi.

Deneysel çalışmalar; sınıflandırma problemlerinin çözümünde kullanılmak üzere VS tabanlı eğitim yaklaşımı ve karşılaştırma amacıyla kullanılan diğer algoritmalar; Çok-Çıkışlı ve Tek-Çıkışlı ÇKA ağı olmak üzere iki farklı topoloji üzerinde çalıştırıldı. Algoritmaların performans değerlendirmeleri üç aşamada yapıldı. Sonuçlar incelendiğinde VS algoritması problemlerin en az diğer meta-sezgisel optimizasyon algoritmaları kadar iyi bir performansa sahip olduğu görüldü.

Çok-Çıkışlı YSA yapıları ve Tek-Çıkışlı YSA yapılarının sınıflandırma performansları incelendi. Performans değerlendirmesi için üç farklı yaklaşım kullanıldı. Birinci aşamada, test problemlerini kullanarak ÇKA ağlarının eğitilmesinde veri setlerinin tamamı eğitim seti olarak kullanıldı ve sonuçlar incelendi. İkinci aşamada, ÇKA ağlarının eğitilmesinde veri setlerinin rasgele %75'i eğitim verisi olarak alınırken, geri kalan %25'i test verisi olarak kullanıldı ve başarısı incelendi. Son aşamada; problemlerin verileri düzenlenirken çapraz-doğrulama yöntemi kullanılarak eğitim ve test verilerinin değiştirilmesinin eğitim ve test başarılarına etkileri incelendi. Bu aşamada, test verileri 10-katlamalı çapraz doğrulama yöntemi uygulandı ve veri setleri eşit şekilde %10'luk on parçaya bölündü. Her parça test veri setini oluşturacak şekilde 30'ar kez çalıştırıldı ve toplam 300 çalıştırmanın sonucunda elde edilen hata oranlarının ortalaması ve ortanca değerleri istatistiksel dağılım değerleri ile birlikte kaydedildi. 10 çalıştırmada ulaşılan eğitim hatalarının minimum, maksimum, ortalama ve ortanca değerleri verildi. Eğitimlerin tamamlanmasının ardından test aşamasında elde edilen ortalama ve ortanca değerleri de ayrıca verildi.

5. ÇKA AĞLARININ VS İLE EĞİTİLMESİNİ ETKİLEYEN FAKTÖRLERİN İNCELENMESİ

Girdap arama algoritmasında popülasyon büyüklüğü, itersayon sayısı, üst ve alt sınırları olmak üzere üç temel kontrol parametresi bulunmaktadır. Popülasyon boyutu ve her bir popülasyonun kaç sefer yenilenip değerlendirileceğini belirleyen jenerasyon sayısı olarak yer almaktadır. VS algoritmasında bir çözümün iyileştirme çalışmalarının ne zaman sonlandırılacağına karar veren itersayon sayısı algoritmanın performansını direkt etkileyen parametredir. Bu sebeple, itersayon sayısını ele alınan sınıflandırma test problemlerinden araştırmalarda en çok tercih edilen kanser, diyabet ve tiroit problemleri için ÇKA ağlarının eğitilmelerindeki etkileri incelenmiştir.

Yapay sinir ağlarının eğitilmesinde sınıflandırma doğruluğunun ve minimum kare hata yüzdesinin kullanılması eğitim için sıklıkla tercih edilen yaklaşımlardandır. Kanser, diyabet ve tiroit problemlerinin eğitim aşamasında ortalama ve ortanca MSE yüzdesi, IQR ve standart sapma değerleri hesaplanmıştır. İterasyon sayısı etkisinin analizi için yapılan çalışmalarda popülasyon büyüklüğü 30 ve parametrelerin değer aralıkları olarak [-20, 20] alınmıştır. Bu değerle elde edilen sonuçlarla birlikte iterasyon değerleri 2000,1000, 500 ve 100 seçildiğinde VS algoritmasının performansındaki değişiklikler 30'ar koşma sonucunda elde edilen ortalama ve ortanca MSE değerleri kaydedilerek incelenmiştir. Tablo 5.1'de test problemleri üzerindeki sınıflandırma başarısı farklı iterasyon değerleriyle 30'ar koşma sonucu ortalama ve ortanca MSE, IQR ve standart sapma değerleriyle sunulmuştur.

Tablo 5.1. Iterasyon sayısının ÇKA ağlarının VS algoritması ile eğitilmesinin etkisi

Veri Seti		İterasyon sayısı			
		2000	1000	500	100
Diyabet	Ortanca	0.225975	0.225125	0.226081	0.228187
	IQR	0.004709	0.005421	0.003835	0.007135
	Ortalama	0.223657	0.224070	0.224349	0.232657
	std	0.004898	0.005144	0.005650	0.014212
Kanser	Ortanca	0.226212	0.226212	0.226212	0.226215
	IQR	4.718447e-16	5.273559e-16	3.330669e-16	2.775557e-15
	Ortalama	0.209831	0.225828	0.226212	0.226212
	std	0.043512	0.002102	4.277266e-16	6.990853e-14
Tiroit	Ortanca	0.040145	0.043154	0.044239	0.047483
	IQR	0.004788	0.004709	0.003444	0.001280
	Ortalama	0.040171	0.043295	0.044366	0.047181
	std	0.003302	0.002589	0.002510	0.001309

Tablo 5.1 incelendiğinde VS algoritmasında iterasyon sayısı arttıkça adım büyüklüğü azalır, arama çözünürlüğü de artar. İterasyon parametresinin değeri artıkça özellikle kanser problemine göre daha zor problemler olan diyabet ve tiroit problemlerinde ortalama MSE değerleri daha başarılı ve daha düşük sınıflandırma hatasının elde edilmesine yol açmıştır. Bunun sebebi, küçük iterasyon değerleriyle algoritma çalıştırıldığında algoritmanın bazı çözümlerin etrafında yeterince arama gerçekleştirmeden o çözümleri terk etme ihtimali olmasındandır.

5.1. Popülasyon Sayısı ve Parametre Aralık Değerlerinin Etkisi

Bu bölümde şu ana kadar yapılan çalışmalarda kullanılan yapay sinir ağlarının eğitilmesinde ağırlık değerlerinin alabileceği değer aralığı [-20,20] olarak ayarlanmıştı. Ağırlık değerlerinin alabileceği aralığın değiştirilmesinin etkisinin araştırılması amacıyla ağırlıkların alabileceği değerler [-10,10] ve [-50,50] şeklinde ayrı ayrı seçilerek sonuçlar alınmıştır. Parametre aralığı [-50,50] seçildiğinde kullanılan eğitim algoritmasının daha büyük bir arama uzayında arama yapacak olmasından dolayı daha fazla değerlendirme sayısına ihtiyaç duyulabileceği göz önünde bulundurularak popülasyon büyüklüğü da 30 yerine 60 olarak ayarlanmıştır. Yapılan değişikliklerin diğer değişikliler ile tutarlı olması amacıyla parametre aralığı [-10,10] seçildiğinde de popülasyon büyüklüğü 20 olarak alınmıştır. Bu ayarlamalar ile 30'ar koşma sonucu yapılan eğitimler ile elde edilen ortalama ve ortanca MSE hata değerleriyle bunların standart sapma ve IQR değerleri sırayla Tablo 5.2 ve 5.3'de sunulmuştur.

Tablo 5.2. Parametre Aralığı [-10,10], Popülasyon büyüklüğü 20 seçilmesi sonucu MSE değerleri

Veri Seti	VS				
	Diyabet	Kanser	Tiroit	Şarap	Iris
Ortanca	0.2262748	0.2262124	0.0463774	0.162247	0.0596591
IQR	0.0030097	1.051936e-14	0.0010055	0.036656	0.0799588
Ortalama	0.2247420	0.2262124	0.0460435	0.166335	0.0660262
Std	0.0045488	2.177784e-14	0.0013627	2.762811e-02	0.0403997

Parametre aralığı [-10,10] alındığında elde edilen ve Tablo 5.2 'de verilen sonuçlara göre YSA'ların sınıflandırma başarıları oldukça yüksektir. Diyabet için ortalama ve ortanca MSE değerleri sırasıyla 0.224742 ve 0.226274, kanser için 0.226212 ve 0.226212, tiroit için 0.046043 ve 0.046377, şarap için 0.166335 ve 0.162247 ve iris için 0.066026 ve 0.0596591 ortalama sınıflandırma hata değerleri elde edilmiştir. Parametre aralığı [-50,50]

olarak ayarlandığında elde edilen Tablo 5.3'de sunulan sonuçlar değerlendirildiğinde beş problem için de sınıflandırma başarısının [-10,10] aralığına göre daha düşük olduğu [-20,20] aralığı seçildiğinde elde edilen sonuçlara göre daha başarısız olduğunu görülmektedir.

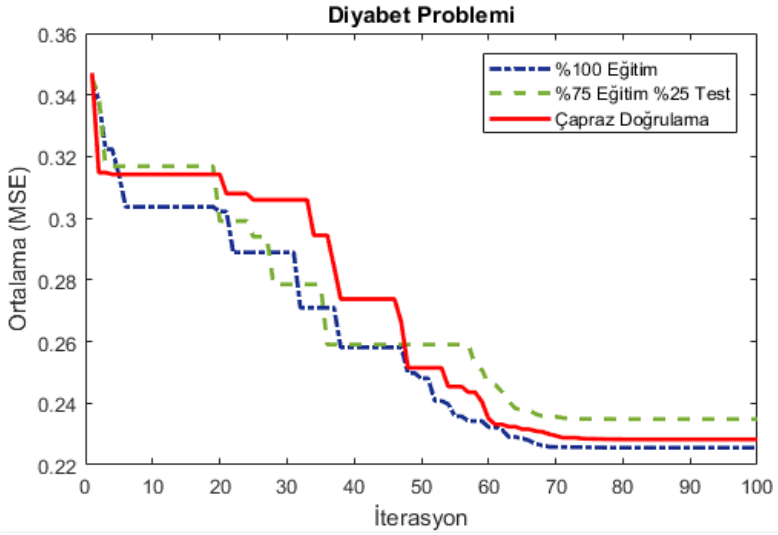
Tablo 5.3. Parametre Aralığı [-50,50], Popülasyon sayısı 60 seçilmesi sonucu MSE değerleri

Veri Seti	VS				
	Diyabet	Kanser	Tiroit	Şarap	İris
Ortanca	0.240653	0.226212	0.0476306	0.1744680	0.0259696
IQR	0.057576	1.304512e-15	0.0016571	0.0306517	0.0998086
Ortalama	0.255349	0.232190	0.0472216	0.1813654	0.0641484
Std	0.027769	0.018239	0.0018315	0.0323539	0.0532462

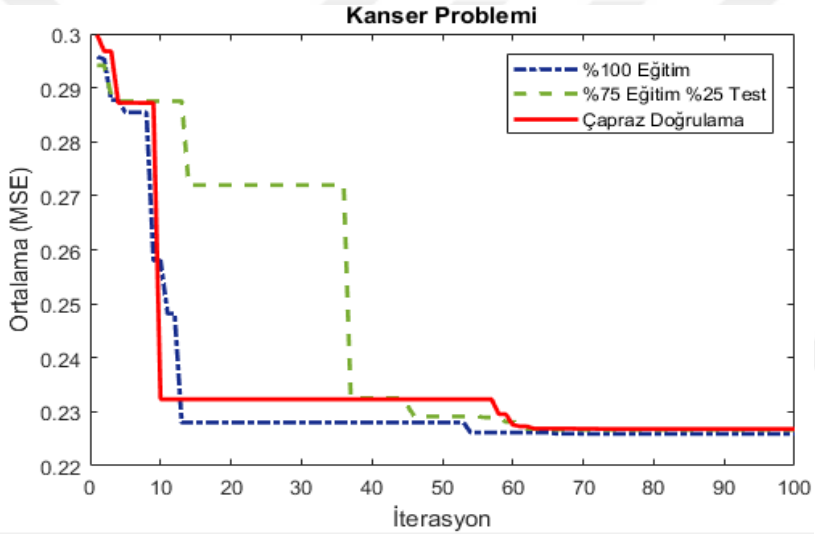
Tablo 5.2 ve 5.3 incelediğimizde daha düşük popülasyon boyutları daha iyi performans sergilediğinin ve yakınsama hızı daha iyi olduğunun sonucuna varılmıştır. Bu sonuçlara ek olarak birlikte her problem için farklı popülasyon sayısı, parametre aralık değerleri için sonuçlar alınmasına rağmen birbirlerine benzer sonuçlar olduğundan dolayı her problem için aynı ifadelerin tekrarına yer vermemek adına hepsi ayrı ayrı sunulmamıştır.

5.2. ÇKA Ağlarının VS Algoritması ile Eğitilmesini Etkileyen Faktörler

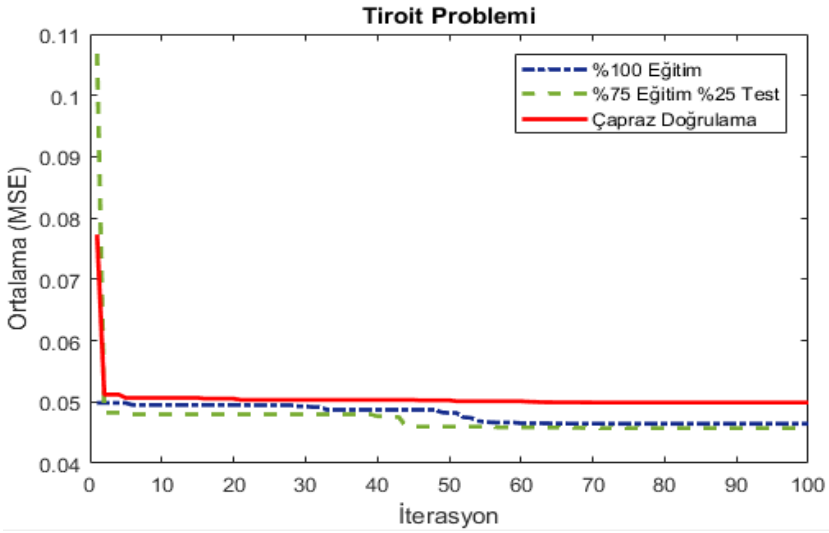
UCI veri tabanından alınan 5 sınıflandırma test problemlerinin çözülmesi amacıyla VS algoritması ile eğitilen Çok-Çıkışlı ve Tek-Çıkışlı ÇKA ağı modellerinin sınıflandırma başarıları aşağıdaki Şekil (5.1, 5.2, 5.3, 5.4 5.5, 5.6, 5.7, 5.8, 5.9 ve 5.10) incelendiğinde Çok-Çıkışlı ÇKA ağlarının Tek-Çıkışlı ÇKA ağlarına göre çok daha başarılı olduğu ÇKA ağlarının, özellikle diğer problemlere göre daha zor problemler olarak nitelendirebileceğimiz daha fazla örnek, giriş ve çıkışa sahip problemler olan tiroit problemlerinde ağlarından başarılı sınıflandırma yaptığı sonucuna varılmıştır. Kanser, diyabet ve diğer problemlerinde üç aşama performans değerlendirmesi ile yakın sonuçlar üretirken bu üç yöntemde de VS başarılı performans göstermiştir.



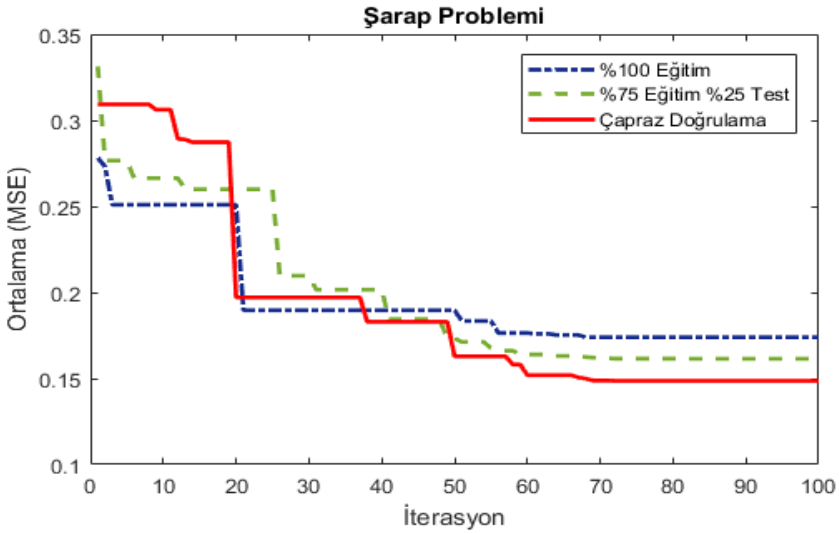
Şekil 5.1. Diyabet Veri Seti için Çok-Çıkışlı ÇKA ağlarının eğitilmesinde VS algoritmasının Ortalama Eğitim Hata yakınsama grafiği



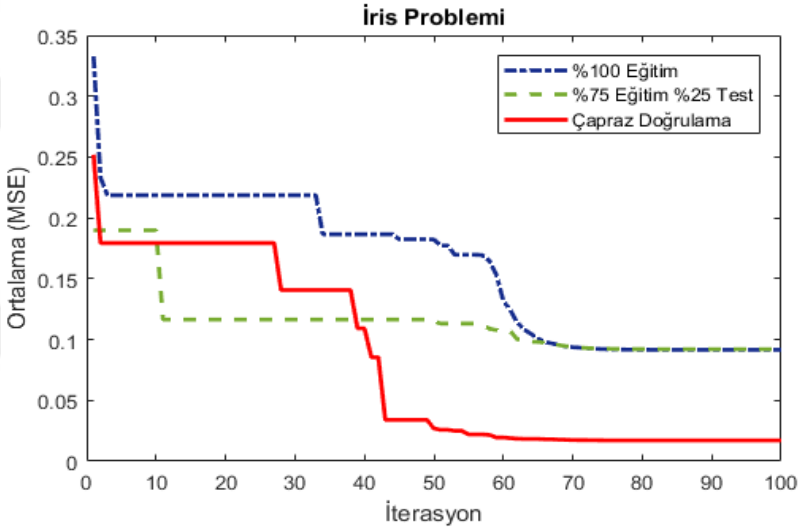
Şekil 5.2. Kanser Veri Seti için Çok-Çıkışlı ÇKA ağlarının eğitilmesinde VS algoritmasının Ortalama Eğitim Hata yakınsama grafiği



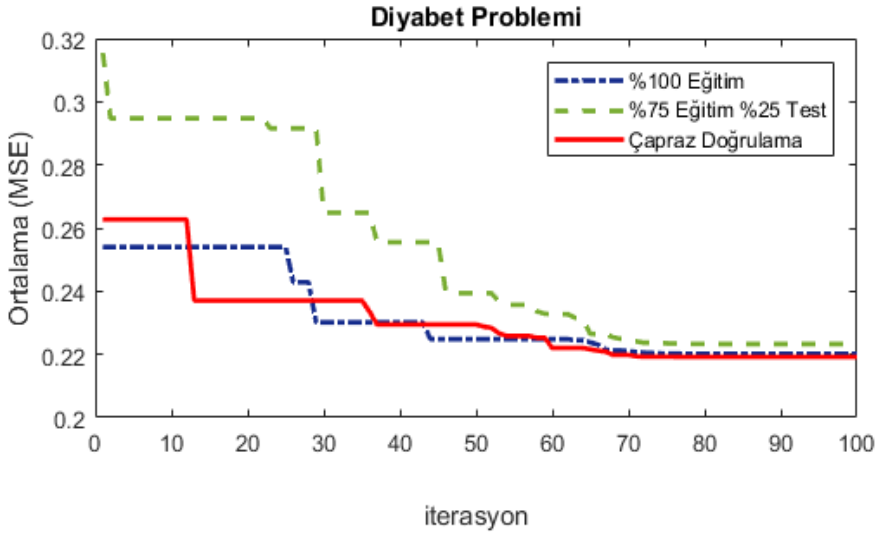
Şekil 5.3. Tiroit Veri Seti için Çok-Çıkışlı ÇKA ağlarının eğitilmesinde VS algoritmasının Ortalama Eğitim Hata yakınsama grafiği



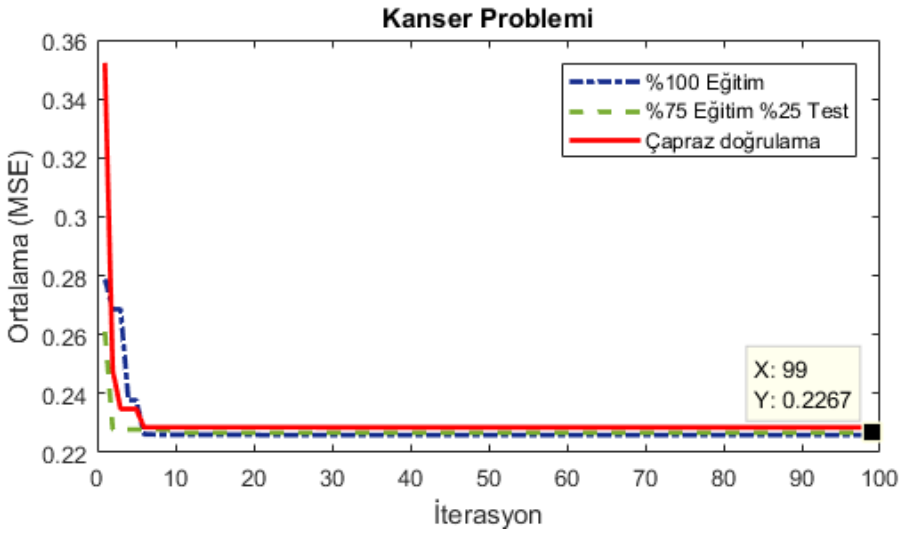
Şekil 5.4. Şarap Veri Seti için Çok-Çıkışlı ÇKA ağlarının eğitilmesinde VS algoritmasının Ortalama Eğitim Hata yakınsama grafiği



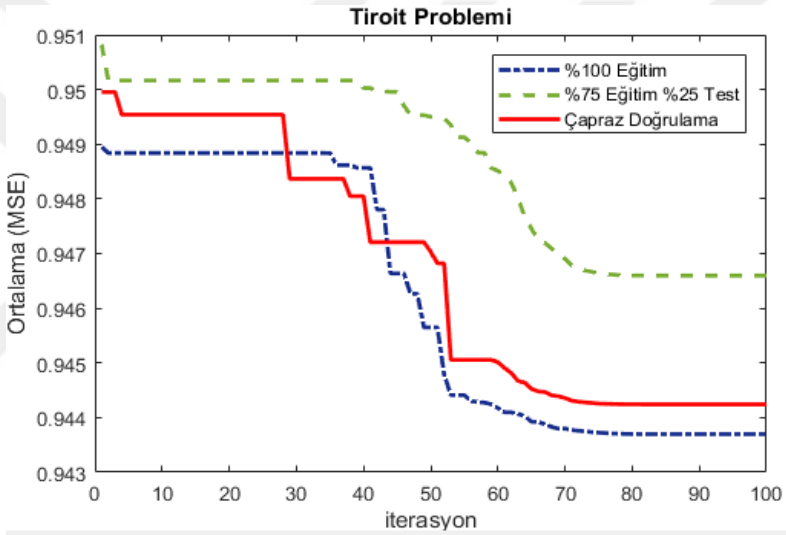
Şekil 5.5. İris Veri Seti Çok çıkışlı ÇKA ağlarının eğitilmesinde VS algoritmasının Ortalama Eğitim Hata yakınsama grafiği



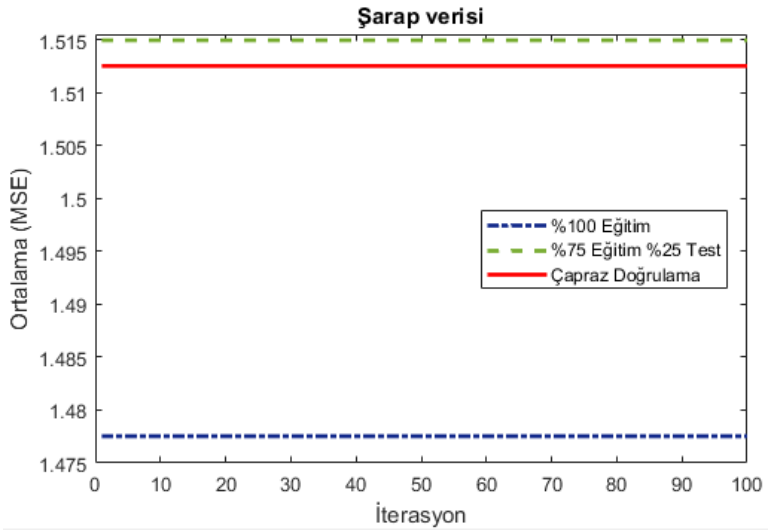
Şekil 5.6. Diyabet Veri Seti Tek çıkışlı ÇKA ağlarının eğitilmesinde VS algoritmasının Ortalama Eğitim Hata yakınsama grafiği



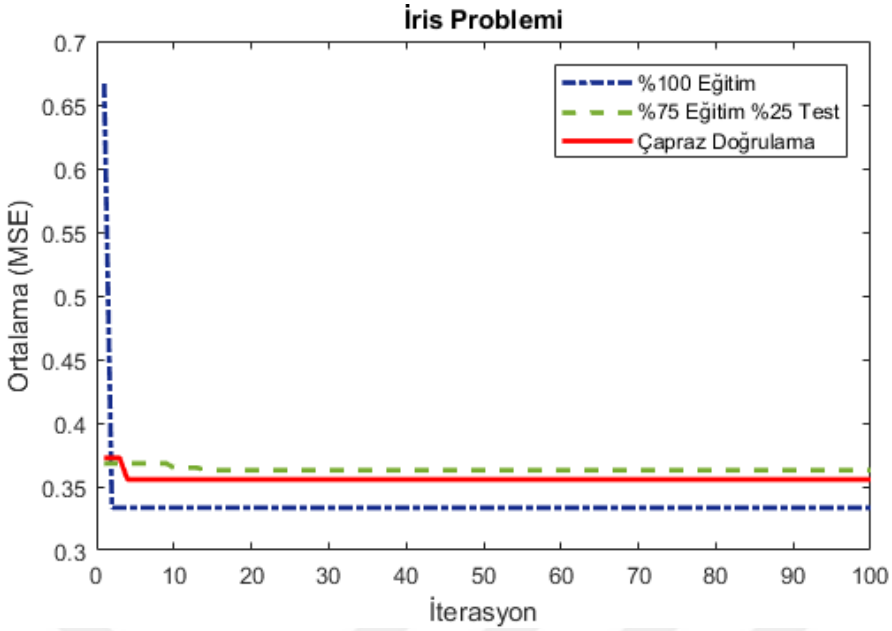
Şekil 5.7. Kanser Veri Seti Tek çıkışlı ÇKA ağlarının eğitilmesinde VS algoritmasının Ortalama Eğitim Hata yakınsama grafiği



Şekil 5.8. Tiroit Veri Seti Tek çıkışlı ÇKA ağlarının eğitilmesinde VS algoritmasının Ortalama Eğitim Hata yakınsama grafiği

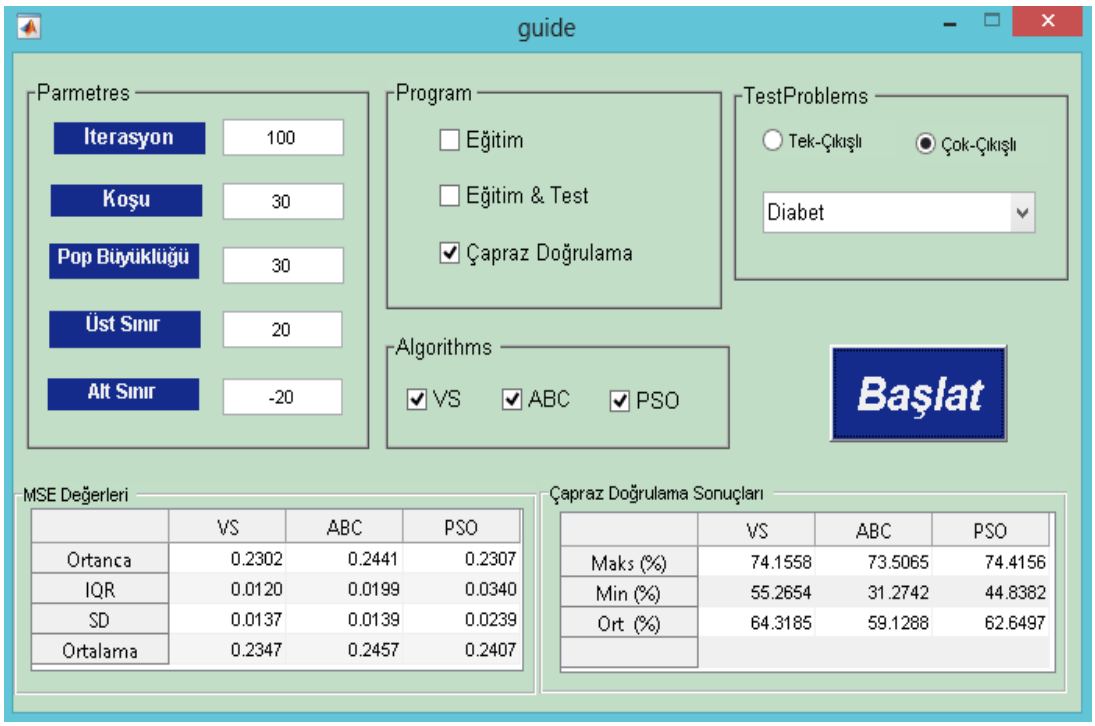


Şekil 5.9. Şarap Veri Seti Tek çıkışlı ÇKA ağlarının eğitilmesinde VS algoritmasının Ortalama Eğitim Hata yakınsama grafiği



Şekil 5.10. İris Veri Seti Tek çıkışlı ÇKA ağlarının eğitilmesinde VS algoritmasının Ortalama Eğitim Hata yakınsama grafiği

MATLAB R2017a da yazılmış olan program çalıştırıldı. Matlab'te sıkça kullanılan grafiksel kullanıcı ara yüzleri, yazılım uygulamalarının kontrolünü sağlar ve uygulamayı çalıştırmak için bir dil veya tip komutlarını öğrenmeye gerek kalmaz. Şekil 5.11 'de programın ara yüzü göstermektedir.



Şekil 5.11. Programın MATLAB ara yüzü

5.3. Bölüm Değerlendirmesi

Bu bölümde, girdap arama algoritması ile yapay sinir ağlarının eğitilmesinde performansı etkileyen faktörler incelendi. Deneylerde VS algoritmasının sınıflandırma problemlerinin çözümünde kullanılmak üzere Çok-Çıkışlı ÇKA ağı ve Tek-Çıkışlı ÇKA ağı olmak üzere iki farklı yaklaşımın üç aşamada performans değerlendirmesi ile ortalama değerleri elde edildi. Popülasyon büyüklüğü, iterasyon sayısı, ağırlık değerlerinin üst ve alt sınırları olmak üzere kontrol parametreleri incelendi. VS algoritmasında iterasyon sayısı arttıkça adım büyüklüğü azalırken arama çözünürlüğü de artar. Düşük popülasyon boyutları ile VS diğer algoritmalardan daha iyi performans sergiledi ve yakınsama hızının çok daha iyi olduğu sonucuna varıldı.

Sonuç olarak; VS algoritması ile eğitilen Çok-Çıkışlı ve Tek-Çıkışlı ÇKA ağı modellerinin sınıflandırma başarıları diğer algoritmalar ile karşılaştırmalı analizleri dikkate alındığında, VS algoritmanın verimli, rekabet düzeyi yüksek ve uygulanabilir nitelikte bir yöntem olduğunu görülmektedir.

6. SONUÇLAR VE ÖNERİLER

Yapay sinir ağları; örüntü tanıma, regresyon, fonksiyon tahmini, işlem kontrolü sınıflandırma, robotik ve daha birçok alanda başarıyla uygulanmıştır. Literatürde birçok gerçek-dünya problemi için sınıflandırma ve tahmin amaçlı YSA uygulaması yer almaktadır. Öğrenme yeteneğine sahip olan yapay sinir ağlarının problem çözmedeki başarısı büyük ölçüde ağın doğru bir şekilde eğitimine ve eğitiminde kullanılan öğrenme algoritmasının performansına bağlıdır. Bu noktada amaç optimal ağırlık değerlerinin elde edilmesidir ve NP-Zor problemlerin çözümünde çok yaygın bir şekilde kullanılan yapay sinir ağlarının eğitimi, doğrusal olmayan önemli bir optimizasyon problemi olarak ele alınabilir.

Problem türlerinin doğrusal olmaması ve boyutlarının çok büyük olması gibi etmenlerden dolayı YSA'ların eğitiminde türeve dayalı klasik öğrenme algoritmaları yetersiz kalmış ve bu tür problemlerin çözümü için literatürde pek çok sezgisel optimizasyon algoritması ağırlıkların optimal değerlerini belirmek amacıyla önerilmiştir. Örneğin Genetik Algoritmalar, Benzetimsel Tavlama, Tabu Araştırma en yaygın kullanılan tekniklerden bazılarıdır. Bunların yanı sıra son yıllarda Sürü Zekasına dayalı optimizasyon yaklaşımlardan bal arılarının yiyecek arama davranışlarını örnek alan Yapay Arı Kolonisi optimizasyon algoritması ve kuş ve balık sürülerinin kolektif davranışlarından esinlenerek geliştirilmiş olan Parçacık Sürü Optimizasyon algoritması bu alanda sıklıkla kullanılmaktadır.

Bu tez çalışmasında, ÇKA ağlarının eğitilmesinde daha önce literatürde bu amaçla kullanılmamış olan Girdap Arama Algoritmasının kullanılabilirliği araştırılmıştır. Girdap Algoritması karıştırılan sıvıların girdap akışıyla oluşturulan girdap deseninden esinlenir ve tek-amaçlı optimizasyon problemlerinin çözümü için geliştirilmiştir. VS algoritması çok katmanlı ileri beslemeli yapay sinir ağlarının sınıflandırma problemlerinin çözümüne uygulanması esnasında ağların eğitilmesi amacıyla kullanılmış ve sergilediği performans analiz edilmiştir. Çok katmanlı ileri beslemeli algılayıcı ağları UCI makine öğrenme deposundan alınan 5 farklı sınıflandırma test probleminin çözülmesi amacıyla tasarlanmış ve eğitiminde de VS algoritmasının performansı Yapay Arı Kolonisi ve Parçacık Sürüsü Optimizasyonu meta-sezgisel teknikleriyle karşılaştırılmıştır.

Önerilen eğitim yaklaşımının güvenilirliğini ve geçerliliğini test etmek amacıyla Çok-Çıkışlı ÇKA ağı ve Tek-Çıkışlı ÇKA ağı olmak üzere iki farklı yaklaşım ve

algoritmaların performans değerlendirmeleri üç aşamada yapıldı. Birinci aşamada; uygulamalarda her problemin %100'ü eğitim seti olarak kullanıldı. Bu aşamada, VS algoritması, Çok-Çıkışlı yapay sinir ağının eğitilmesinde Tek-Çıkışlı yapay sinir ağına kıyasla daha iyi performans sergilediği ortaya konuldu. İkinci aşamada; uygulamalarda her problemin %75'i eğitim seti olarak geri kalan %25'i ise test seti olarak kullanıldı. Bu aşamada ise VS algoritması yapay sinir ağlarının eğitilmesinde en az diğer algoritmalar kadar iyi bir performansa sahip olduğu görüldü. Üçüncü aşamada ise k-katlamalı çapraz doğrulama yöntemi ile eğitimler gerçekleştirildi. En iyi performanstan düşük performans sergileyene göre algoritmalar; VS, PSO, ABC şeklinde sıralandığı görüldü. Deneyler her bir problem için rasgele oluşturulan popülasyonlar ile 30 kez tekrarlandı. Ağın eğitim işlemi, MSE değeri 0.01 olduğunda veya maksimum iterasyon sayısına ulaşıldığında durduruldu. Maksimum iterasyon sayısı 100 ve parametrelerin (ağırlıkların) değer aralıkları olarak [-20, 20] alındı. Popülasyon sayısı 30 olarak alındı. Tüm aşamalardan elde edilen eğitim sonuçları ayrı ayrı listelenerek; algoritmaların performans değerlendirmeleri yapıldı. Alınan sonuçlar ortanca, çeyrekler açıklığı, standart hata yüzdesi, ortalama karesel hata ve sınıflandırma hata yüzdesi hesaplanarak sunuldu. Elde edilen sonuçlarda VS algoritmasının ÇKA yapay sinir ağlarının sınıflandırma amaçlı eğitilmesinde uygulanabilir bir yaklaşım olduğu sonucuna varıldı.

Bu tez çalışmasında VS algoritmasının orijinal yapısına sadık kalınarak standart VS algoritmasının performansı ele alınan problemler için incelendi Sonraki çalışmalarda. Girdap arama algoritmasında uygulanacak problemin yapısında düzenlenmeler yapılarak performansındaki değişiklikler gözlemlenebilir. Bu amaçla algoritmanın adımlarına eklenecek modifikasyonlar yapay sinir ağlarının eğitiminde performansı olumlu yönde etkileyebilir. Bununla birlikte farklı amaç fonksiyonları kullanımının eğitimler üzerindeki etkileri incelenebilir.

KAYNAKLAR

Ali, W., Qyyum, M. A., Qadeer, K. ve Lee, M., 2018, Energy optimization for single mixed refrigerant natural gas liquefaction process using the metaheuristic vortex search algorithm, *Applied Thermal Engineering*.

Andrews, L. C., 1997, Special Functions of Mathematics for Engineers, *Oxford University*.

Arce, F., Zamora, E., Sossa, H. ve Barrón, R., 2018, Differential evolution training algorithm for dendrite morphologicalneural networks, *Applied Soft Computing*, 68, 303–313.

Arora, S. ve Singh, S., 2015, Butterfly Algorithm with Levy Flights for Global Optimization, *2015 International Conference on Signal Processing, Computing and Control (Ispcc)*, 220-224.

Back, T. ve Schwefel, H. P., 1993, An Overview of Evolutionary Algorithms for Parameter Optimization, *Evolutionary Computation*, 1 (1), 1-23.

Basheer, I. A. ve Hajmeer, M., 2000, Artificial neural networks: fundamentals, computing, design, and application, *Journal of Microbiological Methods*, 43 (1), 3-31.

Behrman, M., Linder, R., Assadi, A. H., Stacey, B. R. ve Backonja, M. M., 2007, Classification of patients with pain based on neuropathic pain symptoms: Comparison of an artificial neural network against an established scoring system, *European Journal of Pain*, 11 (4), 370-376.

Bektaş, B. ve Babur, S., 27-29 Ekim, 2016, Machine Learning Based Performance Development for Diagnosis of Breast Cancer. *tıp teknolojileri kongresi*. IC Hotel Santai Family Resort, Antalya.

Benusková ve Lubica, 1994, Modelling the effect of the missing fundamental with an attractor neural network, *Network-Computation in Neural Systems*, 5 (3), 333-349.

Billah, M., Waheed, S. ve Hanifa, A., 2016, Stock Market Prediction Using an Improved Training Algorithm of Neural Network, *2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (Icecte)*.

Birattari, M., 2005, The problem of tuning metaheuristics as seen from a machine learning perspective, *Berlin, Germany*, Dissertationen zur Kunstlichen Intelligenz.

Birattari, M., Paquete, L., Stützle, T. ve Varrentrap, K., 2001, Classification of Metaheuristics and Design of Experiments for the Analysis of Components, *Technical Report AIDA-2001-05. Intellektik, Technic University Darmstadt, Germany.*

Carvalho, M. v. L., T., 2006, Particle swarm optimization of feed-forward neural networks with weight decay, In Proceedings of Sixth International Conference on Hybrid Intelligent Systems, 2662–2666.

Charalambous, C., 1992, Conjugate-Gradient Algorithm for Efficient Training of Artificial Neural Networks, *Iee Proceedings-G Circuits Devices and Systems*, 139 (3), 301-310.

Chiang, W.-y. K., Zhang, D. ve Zhou, L., 2006, Predicting and explaining patronage behavior toward web and traditional stores using neural networks: a comparative analysis with logistic regression, *Decision Support Systems*, 41, 514-531.

Chu, S. C., Tsai, P. W. ve Pan, J. S., 2006, Cat Swarm Optimization, *Pricai 2006: Trends in Artificial Intelligence, Proceedings*, 4099, 854-858.

Çomak, E. ve Arslan, A., 2008, DESTEK VEKTÖR MAKİNELERİNİN ETKİN EĞİTİMİ İÇİN YENİ YAKLAŞIMLAR, *SELÇUK ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ Elektrik-Elektronik Mühendisliği Bölümü*, KONYA, 125.

Coskun, N. ve Yildirim, T., 2003, The effects of training algorithms in MLP network on image classification, *Proceedings of the International Joint Conference on Neural Networks 2003, Vols 1-4*, 1223-1226.

Cui, L. G., Deng, J., Wang, L., Xu, M. Z. ve Zhang, Y. J., 2016, A novel locust swarm algorithm for the joint replenishment problem considering multiple discounts simultaneously, *Knowledge-Based Systems*, 111, 51-62.

Dach, J., Koszela, K., Boniecki, P., Zaborowicz, M., Lewicki, A., Czekala, W., Skwarcz, J., Qiao, W., Piekarska-Boniecka, H. ve Bialobrzewski, I., 2016, The use of neural modelling to estimate the Methane production from slurry fermentation processes, *Renewable & Sustainable Energy Reviews*, 56, 603-610.

Davoudi, E. ve Vaferi, B., 2018, Applying artificial neural networks for systematic estimation of degree of fouling in heat exchangers, *Chemical Engineering Research & Design*, 130, 138-153.

Dogan, B. ve Olmez, T., 2015, A new metaheuristic for numerical function optimization: Vortex Search algorithm, *Information Sciences*, 293, 125-145.

Dogan, B. ve Ölmez, T., 2015, Vortex search algorithm for the analog active filter component selection problem, *International Journal of Electronics and Communications (AEÜ)*, 69, 1243–1253.

Dogan, B., 2016, A Modified Vortex Search Algorithm For Numerical Function Optimization, *International Journal of Artificial Intelligence and Applications (IJAIA)*, 7.

Dordinejad, G. G., 2016, Çok Yüzlü Konik Sınıflandırıcılarda Gürbüz Koni Tepe Noktası Tahmini, *Osmangazi Üniversitesi Fen Bilimleri Enstitüsü Elektrik ve Elektronik Mühendisliği*, Eskişehir, 62.

Dorigo, M., Birattari, M. ve Stutzle, T., 2006, Ant colony optimization - Artificial ants as a computational intelligence technique, *Ieee Computational Intelligence Magazine*, 1 (4), 28-39.

Duch W., A. R., Grąbczewski K., 2001, A new methodology of extraction,optimization and application of crisp and fuzzy logical rules, *IEEE Transactions on Neural Networks*, 12, 277-306.

Dvir, D., Ben-David, A., Sadeh, A. ve Shenhar, A. J., 2006, Critical managerial factors affecting defense projects success: A comparison between neural network and regression analysis, *Engineering Applications of Artificial Intelligence*, 19 (5), 535-543.

Elisseeff, A. ve Paugam-Moisy, H., 1999, JNN, a randomized algorithm for training multilayer networks in polynomial time, *Neurocomputing*, 29 (1-3), 3-24.

Forina, M., 1998, An Extendible Package for Data Exploration, Classification and Correlation., *Institute of Pharmaceutical and Food Analysis and Technologies, Via Brigata Salerno, Italy.*

Frank, A. ve Asuncion, A., 2010, Uci machine learning repository, *Irvine, CA: University of California, School of Information and Computer Science,*, 21.

Gholami, E., Vaferi, B. ve Ariana, M. A., 2018, Prediction of viscosity of several alumina-based nanofluids using various artificial intelligence paradigms - Comparison with experimental data and empirical correlations, *Powder Technology*, 323, 495-506.

Gibbs, M. S., Morgan, N., Maier, H. R., Dandy, G. C., Nixon, J. B. ve Holmes, M., 2006, Investigation into the relationship between chlorine decay and water distribution parameters using data driven methods, *Mathematical and Computer Modelling*, 44 (5-6), 485-498.

Gopalakrishnan, K., 2010, Effect of training algorithms on neural networks aided pavement diagnosis., *International Journal of Engineering, Science and Technology*, 2, 83-92.

Gupta, P., Gupta, N. ve Dey, P., 2016, Artificial Neural Network for Cytodiagnosis of Endometrial Carcinoma, *Analytical and Quantitative Cytopathology and Histopathology*, 38 (4), 249-254.

Gurney, K., 1997, An Introduction to Neural Networks, *first ed. CRC press, New York.*, p.

Hagan, M. T. ve Menhaj, M. B., 1994, Training Feedforward Networks with the Marquardt Algorithm, *Ieee Transactions on Neural Networks*, 5 (6), 989-993.

Hagan, M. T., Demuth, H. B., Beale, M. H. ve Jesús, O. D., 1996, Neural Network Design.

Hameed, A. A., Karlik, B. ve Salman, M. S., 2016, Back-propagation algorithm with variable adaptive momentum, *Knowledge-Based Systems*, 114, 79-87.

Han, S. S. ve May, G. S., 1996, Optimization of neural network structure and learning parameters using genetic algorithms, *Eighth Ieee International Conference on Tools with Artificial Intelligence, Proceedings*, 200-206.

Hassanpour, M., Vaferi, B. ve Masoumi, M. E., 2018, Estimation of pool boiling heat transfer coefficient of alumina water-based nanofluids by various artificial intelligence (AI) approaches, *Applied Thermal Engineering*, 128, 1208-1222.

Hofestadt, R. ve Töpel, T., 2005, medical bionformatics: Detecting molecular diseases with case-based reasoning, *Germany* Springer.

İ.İşeri, Öz, C., Dağkurs, L. ve Kara, M., 2011, Equal Width Discretization and Artifical Neural Network Based Breast Cancer Detection, *6th International Advanced Technologies Symposium (IATS'11)*,Elazığ, Turkey.

Ince, T., Kiranyaz, S., Pulkkinen, J. ve Gabbouj, M., 2010, Evaluation of global and local training techniques over feed-forward neural network architecture spaces for computer-aided medical diagnosis, *Expert Systems with Applications*, 37, 8450-8461.

Jiang, P. ve Chen, J. J., 2016, Displacement prediction of landslide based on generalized regression neural networks with K-fold cross-validation, *Neurocomputing*, 198, 40-47.

JosephTarigan, Nadia, RyandaDiedan ve YayaSuryana, 2017, Plate Recognition Using Appropriation Neural Network and Genetic Algorithm, *2nd International Conference on Computer Science and Computational Intelligence (ICCSCI 2017)*, 116, 365–372.

Kalderstam, J., Eden, P., Bendahl, P. O., Strand, C., Ferno, M. ve Ohlsson, M., 2013, Training artificial neural networks directly on the concordance index for censored data using genetic algorithms, *Artificial Intelligence in Medicine*, 58 (2), 125-132.

Karaboga, D., 2005, An Idea Based on Honey Bee Swarm for Numerical Optimization, *Erciyes University*, Kayseri, Turkey.

Karaboga, D. ve Basturk, B., 2007, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Journal of Global Optimization*, 39 (3), 459-471.

Kaur, G. ve Arora, S., 2018, Chaotic Whale Optimization Algorithm, *Journal of Computational Design and Engineering.*

Kaymak, S., Helwan, A. ve Uzun, D., 2017, Breast cancer image classification using artificial neural networks, *9th International Conference on Theory and Application of Soft Computing, Computing with Words and Perception*, Icsccw 2017, 120, 126-131.

Kennedy, J. ve Eberhart, R., 1995, Particle swarm optimization, *1995 Ieee International Conference on Neural Networks Proceedings, Vols 1-6*, 1942-1948.

Lang, K. J., Waibel, A. H. ve Hinton, G. E., 1990, A Time-Delay Neural Network Architecture for Isolated Word Recognition, *Neural Networks*, 3 (1), 23-43.

Lanza-Gutierrez, J. M. ve Gomez-Pulido, J. A., 2015, Assuming multiobjective metaheuristics to solve a three-objective optimisation problem for Relay Node deployment in Wireless Sensor Networks, *Applied Soft Computing*, 30, 675-687.

Lecun, Y., Bottou, L., Bengio, Y. ve Haffner, P., 1998, Gradient-based learning applied to document recognition, *Proceedings of the Ieee*, 86 (11), 2278-2324.

Lee, K. D., Booth, D. ve Alam, P., 2005, A comparison of supervised and unsupervised neural networks in predicting bankruptcy of Korean firms, *Expert Systems with Applications*, 29 (1), 1-16.

Levenberg, K., 1944, A method for the solution of certain nonlinear problems in least squares, *Quart. Appl. Math.*, 164–168.

Li, X., Niu, P. ve Liu, J., 2018, Combustion optimization of a boiler based on the chaos and Lévy flight vortex search algorithm, *Applied Mathematical Modelling*, 1–16.

Lin, C.-T. ve Lee, C. S. G., 1996, Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems, *Englewood Cliffs, Prentice-Hall International*.

Lu, C., De Brabanter, J., Van Huffel, S., Vergote, I. ve Timmerman, D., 2001, Using artificial neural networks to predict malignancy of ovarian tumors, *Proceedings of the 23rd Annual International Conference of the Ieee Engineering in Medicine and Biology Society, Vols 1-4*, 23, 1637-1640.

Maier, S. H., Powell, R. S. ve Woodward, C. A., 2000, Calibration and comparison of chlorine decay models for a test water distribution system, *Water Research*, 34 (8), 2301-2309.

Mata, J., de Miguel, I., Duran, R. J., Merayo, N., Singh, S. K., Jukan, A. ve Chamania, M., 2018, Artificial intelligence (AI) methods in optical networks: A comprehensive survey, *Optical Switching and Networking*, 28, 43-57.

Mirjalili, S., Mirjalili, S. M. ve Lewis, A., 2014, Grey Wolf Optimizer, *Advances in Engineering Software*, 69, 46-61.

Mirjalili, S., 2015, The Ant Lion Optimizer, *Advances in Engineering Software*, 83, 80-98.

Mirjalili, S. ve Lewis, A., 2016, The Whale Optimization Algorithm, *Advances in Engineering Software*, 95, 51-67.

Özkış, A. ve Babalık, A., 2017, A novel metaheuristic for multi-objective optimization problems: The multi-objective vortex search algorithm, *Information Sciences*, 124–148.

Öztemel, E., 2016, Yapay sinir ağları, *istanbul*, papatya yayıncılık eğitim, p. 232.

Öztürk, C. ve Karaboğa, D., 2011, Hybrid Artificial Bee Colony algorithm for neural network training, *2011 Ieee Congress on Evolutionary Computation (Cec)*, 84-88.

P.Bhargavi ve Dr.S.Jyothi, 2010, SOIL CLASSIFICATION USING GATREE *International journal of computer science & information Technology (IJCSIT)*, 2.

P. Rijnbeek, R. ve J.A., K., 2010 Finding a Short and Accurate Decision Rule in Disjunctive Normal Form by Exhaustive Search,Machine Learning, 80, 33–62.

Pao, Y. H. ve Takefuji, Y., 1992, Functional-Link Net Computing - Theory, System Architecture, and Functionalities, *Computer*, 25 (5), 76-79.

Poggio, T. ve Girosi, F., 1990, Networks for Approximation and Learning, *Proceedings of the Ieee*, 78 (9), 1481-1497.

Rashedi, E., Nezamabadi-Pour, H. ve Saryazdi, S., 2009, GSA: A Gravitational Search Algorithm, *Information Sciences*, 179 (13), 2232-2248.

Rios-Mercado, R. Z. ve Borraz-Sanchez, C., 2015, Optimization problems in natural gas transportation systems: A state-of-the-art review, *Applied Energy*, 147, 536-555.

Rumelhart, D. E., Hinton, G. E. ve Williams, R. J., 1986, Learning Representations by Back-Propagating Errors, *Nature*, 323 (6088), 533-536.

Sajedi, H. ve Razavi, S. F., 2016, MVSA: Multiple Vortex Search Algorithm, *17th Ieee International Symposium on Computational Intelligence and Informatics (Cinti 2016)*, 169-173.

Sato, T., Muto, I., Hasegawa, M., Aono, T., Sakai, T. ve Oya, T., 2013, Ductal carcinoma in situ with isolated tumor cells in the sentinel lymph node in a 17-year-old adolescent girl, *Breast Cancer*, 20 (3), 271-274.

Shao, C. H., Paynabar, K., Kim, T. H., Jin, J. H., Hu, S. J., Spicer, J. P., Wang, H. ve Abell, J. A., 2013, Feature selection for manufacturing process monitoring using cross-validation, *Journal of Manufacturing Systems*, 32 (4), 550-555.

Shao, H. M. ve Zheng, G. F., 2009, A New BP Algorithm with Adaptive Momentum for FNNs Training, *Proceedings of the 2009 Wri Global Congress on Intelligent Systems*, 16.

Sheng-Xiang Lv, Yu-Rong Zeng ve Lin Wang, 2017, An effective fruit fly optimization algorithm with hybrid information exchange and its applications, *International Journal of Machine Learning and Cybernetics*, 1–26.

Simon, D., 2008, Biogeography-Based Optimization, *Ieee Transactions on Evolutionary Computation*, 12 (6), 702-713.

Singh, T. N. Ve Singh, V., 2005, An intelligent approach to prediction and control ground vibration in mines, *Geotechnical and Geological Engineering*, 23, 249–262.

Sitton, J. D., Zeinali, Y. ve Story, B. A., 2017, Rapid soil classification using artificial neural networks for use in constructing compressed earth blocks, *Construction and Building Materials*, 138, 214-221.

Suksri, S. ve Kimpan, W., 2016, Neural Network Training Model for Weather Forecasting Using Fireworks Algorithm, *20th International Computer Science and Engineering Conference (Icsec)*.

Topcu, I. B. ve Saridemir, M., 2008, Prediction of mechanical properties of recycled aggregate concretes containing silica fume using artificial neural networks and fuzzy logic, *Computational Materials Science*, 42 (1), 74-82.

Uslu, M., 2013, yapay sinir ağları ile sınıflandırma Hacettepe Üniversitesi , Fen Fakültesi İstatistik Bölümü, Anakara, Türkiye.

Vaferi, B., Samimi, F., Pakgohar, E. ve Mowla, D., 2014, Artificial neural network approach for prediction of thermal behavior of nanofluids flowing through circular tubes, *Powder Technology*, 267, 1-10.

Vaferi, B., Eslamloueyan, R. ve Ayatollahi, S., 2015, Application of Recurrent Networks to Classification of Oil Reservoir Models in Well-testing Analysis, *Energy Sources Part a-Recovery Utilization and Environmental Effects*, 37 (2), 174-180.

Villarrubia, G., De Paz, J., Chamoso, P. ve De la Prieta, F., 2018, Artificial neural networks used in optimization problems, *Neurocomputing*, 272, 10-16.

Wan, L., Zeiler, M., Zhang, S., LeCun, Y. ve Fergus, R., 2013, Regularization of Neural Networks using DropConnect *Proceedings of the nth International Conference on Machine Learning, Atlanta, Georgia, USA,*.

Wang, Y., Ma, X. L., Xu, M. Z., Liu, Y. ve Wang, Y. H., 2015, Two-echelon logistics distribution region partitioning problem based on a hybrid particle swarm optimization-genetic algorithm, *Expert Systems with Applications*, 42 (12), 5019-5031.

Yang, X. S., 2009, Firefly Algorithms for Multimodal Optimization, *Stochastic Algorithms: Foundations and Applications, Proceedings*, 5792, 169-178.

Yang, X. S., 2010, A New Metaheuristic Bat-Inspired Algorithm, *Nicso 2010: Nature Inspired Cooperative Strategies for Optimization*, 284, 65-74.

Yilmaz, I. ve Kaynar, O., 2011, Multiple regression, ANN (RBF, MLP) and ANFIS models for prediction of swell potential of clayey soils, *Expert Systems with Applications*, 38 (5), 5958-5966.

Yousefi, J. ve Hamilton-Wright, A., 2014, Characterizing EMG data using machine-learning tools, *Computers in Biology and Medicine*, 51, 1-13.

Yurtay, Y., Ak, G. ve Bacınoğlu, N. Z., *Bilgisayar ve Bilişim Bilimleri Fak., Bilgisayar Müh. Bölümü, Sakarya Üniversitesi*, Türkiye, Tıbbi Karar Destek Sistemlerinin Yöntemsel Olarak Değerlendirilmesi Üzerine Bir Çalışma.

Zambrano-Bigiarini, M., Clerc, M. ve Rojas, R., 2013, Standard Particle Swarm Optimisation 2011 at CEC-2013: A baseline for future PSO improvements, *2013 Ieee Congress on Evolutionary Computation (Cec)*, 2337-2344.

Zeng, Y. R., Peng, L., Zhang, J. L. ve Wang, L., 2016, An Effective Hybrid Differential Evolution Algorithm Incorporating Simulated Annealing for Joint Replenishment

and Delivery Problem with Trade Credit, *International Journal of Computational Intelligence Systems*, 9 (6), 1001-1015.

Zhang, Q., Wang, R., Yang, J., Ding, K., Li, Y. ve Hu, J., 2017, Collective decision optimization algorithm: A new heuristic optimization method, *Neurocomputing*, 221, 123–137.

Zhang, X.-L., 2014, Nonlinear Dimensionality Reduction of Data by Deep Distributed Random Samplings, *JMLR: Workshop and Conference Proceedings*, 39, 221–233.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Zainab Abdullah Jalil JALİL
Uyruğu : IRAK
Doğum Yeri ve Tarihi : KERKÜK / 1993
Telefon : (539) 924 29 18
e-mail : zeynepcelil@yahoo.com

EĞİTİM

Derece		Adı, İlçe, İl	Bitirme Yılı
Lise	:	Kerkük Kız Lisesi	**2011**
Üniversite	:	Kerkük Teknik Üniversitesi Bilgisayar Mühendisliği Bölümü	**2015**
Yüksek Lisans	:	Selçuk Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı	**2018**

YABANCI DİLLER

İngilizce, Arapça

YAYINLAR

1. Zainab Abdullah Jalil, Tahir SAĞ. (2017) Classification of Diabetes Dataset with ANN Trained by Vortex Search. International Conference on Engineering Technologies (ICENTE'17), Dec 07-09, 2017, Konya, Turkey.

   (Yüksek Lisans tezinden yapılmıştır)