

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**MAKİNE ÖĞRENMESİ UYGULAMALARINDA YAPAY
ARI KOLONİ PROGRAMLAMA TEMELLİ YENİ
YÖNTEMLERİN GELİŞTİRİLMESİ**

**Hazırlayan
Sibel ARSLAN**

**Danışman
Doç. Dr. Celal ÖZTÜRK**

Doktora Tezi

**Eylül 2019
KAYSERİ**

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**MAKİNE ÖĞRENMESİ UYGULAMALARINDAYAPAY
ARI KOLONİ PROGRAMLAMA TEMELLİ YENİ
YÖNTEMLERİN GELİŞTİRİLMESİ
(Doktora Tezi)**

**Hazırlayan
Sibel ARSLAN**

**Danışman
Doç. Dr. Celal ÖZTÜRK**

**Eylül 2019
KAYSERİ**

BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.



SİBEL ARSLAN

İmza

“Makine Öğrenmesi Uygulamalarında Yapay Arı Koloni Programlama Temelli Yeni Yöntemlerin Geliştirilmesi” adlı Doktora tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi’ ne uygun olarak hazırlanmıştır.

Hazırlayan

Sibel ARSLAN



Danışman

Doç. Dr. Celal ÖZTÜRK



Bilgisayar Mühendisliği ABD Başkanı v.

Prof. Dr. Veysel ASLANTAŞ

İmza

Doç. Dr. Celal ÖZTÜRK danışmanlığında **Sibel ARSLAN** tarafından hazırlanan **“Makine Öğrenmesi Uygulamalarında Yapay Arı Koloni Programlama Temelli Yeni Yöntemlerin Geliştirilmesi”** adlı bu çalışma jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalında **Doktora** tezi olarak kabul edilmiştir.

12 / 09 / 2019

JÜRİ:

Danışman : Doç. Dr. Celal ÖZTÜRK

Üye : Doç. Dr. Bahriye AKAY

Üye : Dr. Öğr. Üyesi Beyza GÖRKEMLİ

Üye : Dr. Öğr. Üyesi Rukiye KARAKIŞ

Üye : Dr. Öğr. Üyesi Selçuk ASLAN

ONAY:

Bu tezin kabulü Enstitü Yönetim Kurulunun 17.09.2019 tarih ve 2019/53-05 sayılı kararı ile onaylanmıştır.



 17 / 09 / 2019

Prof. Dr. Mehmet AKKURT

Enstitü Müdürü

TEŞEKKÜR

Çalışmalarım boyunca beni yönlendiren, hiçbir konuda bilgi, tecrübe, yardım ve yakın ilgisini esirgemeyen, üzerimde çok fazla emeği olan kıymetli danışmanım Doç. Dr. Celal Öztürk hocama saygılarımı sunar, teşekkür ederim. Tez izleme komitemde bulunarak çalışmalarımı yönlendiren, ayrıca çeşitli aşamalarda tezime katkılarda bulunan kıymetli hocalarım Doç. Dr. Bahriye Akay ve Dr. Öğretim Üyesi Beyza Görkemli 'ye teşekkürü bir borç bilirim.

Çalışmalarımda her türlü desteği sağlayan iş arkadaşlarım Nesrin Demircan, Engin Zaman, Nihat Arslan, Hasan Emre Kubat ve Canan Ekici'ye çok teşekkür ederim.

Doktora tez çalışmamı Yurt İçi Lisansüstü Burs Programı kapsamında yürütülen 2211-A Genel Yurt İçi Doktora Burs Programı ile destekleyen TÜBİTAK Bilim İnsanı Destekleme Daire Başkanlığı birimine teşekkürlerimi sunarım.

Çalışmalarım sürecinde beni her zaman destekleyen ve bu günlere getiren kıymetli anne ve babama, abim Uğur, bir tanecik yeğenim Mete'ye sonsuz teşekkürler.

Sibel ARSLAN

Eylül 2019, KAYSERİ

MAKİNE ÖĞRENMESİ UYGULAMALARINDA YAPAY ARI KOLONİ PROGRAMLAMA TEMELLİ YENİ YÖNTEMLERİN GELİŞTİRİLMESİ

Sibel ARSLAN

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü
Doktora Tezi, Eylül 2019
Danışman: Doç Dr. Celal ÖZTÜRK

ÖZET

Otomatik programlama, bir sistemde girdi ve çıktı değişkenleri arasındaki ilişkiyi model üretmek için çalışarak bir makine öğrenmesi yaklaşımıdır. Genetik Programlama (GP), Genetik Algoritmanın (GA) uzantısı olarak geliştirilen ve günümüzde en çok kullanılan otomatik programlama yöntemidir. Yapay Arı Koloni Programlama (ABCP), bal arılarının yiyecek kaynağı arama davranışlarını simüle eden Yapay Arı Kolonisi (ABC) algoritmasına dayalı sembolik regresyon problemi için yeni geliştirilen yüksek seviyeli bir otomatik programlama yöntemidir. Bu tezde, ABCP algoritmasının farklı makine öğrenmesi uygulamalarındaki performansını araştırmaya yönelik çalışmalar yapılarak, sonuçlar GP ile kıyaslanmıştır. Ayrıca, ABCP algoritmasına dayalı Çok Kovanlı Yapay Arı Koloni Programlama (MHABCP) ve ABCP-tanımlayıcı ilk kez yeni yöntemler olarak önerilmiştir. MHABCP’de aday çözüm üretilirken standart ABCP’deki bilgi paylaşım mekanizmasının kullanılması ile birlikte Çok Genli Genetik Programlama’da kullanılan iki noktalı yüksek seviyeli çaprazlama operatörünün komşuluk araştırma işlemine adaptasyonundan esinlenilmiştir. MHABCP gürültü eklenmiş/eklenmemiş yüksek boyutlu sembolik regresyon problemlerinin çözümünde; ABCP-tanımlayıcı desen sınıflandırma problemlerinin çözümünde kullanılmıştır. ABCP-tanımlayıcı modelleri, her bir desen setinin sınıflandırılması için eğitim aşamasında rastgele seçtiği iki tane örnek üzerinde kayan pencereler ile çıkarmıştır. Yapılan çalışmada elde edilen sonuçlar, ABCP ve ABCP algoritmasına dayalı yöntemlerin literatürde yer alan birçok problemde kullanılabileceğini göstermektedir.

Anahtar Kelimeler: Otomatik programlama, Genetik Programlama, Yapay Arı Koloni Programlama(ABCP), Dizi Sınıflandırma, Çok Kovanlı ABCP, ABCP-tanımlayıcı.

DEVELOPMENT OF NEW METHODS BASED ON ARTIFICIAL BEE COLONY PROGRAMMING IN MACHINE LEARNING APPLICATIONS

Sibel ARSLAN

Erciyes University, Graduate School of Natural and Applied Sciences

PhD Thesis, September 2019

Supervisor: Assoc. Prof. Dr. Celal ÖZTÜRK

ABSTRACT

Automatic programming is a machine learning approach that tries to explain the relationship between input and output variables in a system by extracting models. Genetic Programming (GP) is the most widely used automatic programming method developed as an extension of Genetic Algorithm (GA). Artificial Bee Colony Programming (ABCP) is a newly developed high-level automatic programming method for the symbolic regression problem based on the Artificial Bee Colony (ABC) algorithm, which simulates the food source search behavior of honeybees. In this thesis, works were conducted to investigate the performance of ABCP algorithm in machine learning applications and the results were compared with GP. Furthermore, Multi-Hive Artificial Bee Colony Programming (MHABCP) and ABCP-descriptor based on ABCP algorithm were proposed for the first time as new methods. The candidate solution was generated in MHABCP used the information sharing mechanism in the standard ABCP and the adaptation of two-point high-level crossover operator used in Multi-Gene Genetic Programming to the neighborhood research process. MHABCP was used to solve the solution of high dimensional symbolic regression problems with/without noise; ABCP-descriptor was used to solve the texture classification problems. ABCP-descriptor models were extracted by sliding windows on two samples randomly selected during the training phase for the classification of each texture set. The results of this work show that ABCP and ABCP algorithm based algorithms can be used in many problems in the literature.

Keywords: Automatic programming, Genetic Programming, Artificial Bee Colony Programming (ABCP), Sequence Classification, Multi Hive ABCP, ABCP-descriptor.

İÇİNDEKİLER

MAKİNE ÖĞRENMESİ UYGULAMALARINDA YAPAY ARI KOLONİ PROGRAMLAMA TEMELLİ YENİ YÖNTEMLERİN GELİŞTİRİLMESİ

BİLİMSEL ETİĞE UYGUNLUK	ii
YÖNERGEYE UYGUNLUK.....	iii
KABUL VE ONAY	iv
TEŞEKKÜR	v
ÖZET.....	vi
ABSTRACT	vii
İÇİNDEKİLER	viii
KISALTMALAR	xiii
TABLolar LİSTESİ.....	xvii
ŞEKİLLER LİSTESİ.....	xix
GİRİŞ	1

1. BÖLÜM

OTOMATİK PROGRAMLAMA YÖNTEMLERİ VE UYGULAMALARI

1.1. Giriş	4
1.2. Genetik Programlama	5
1.2.1. Genetik Programlama Hazırlık Adımları	6
1.2.2. Genetik Programlama Akışı	6
1.2.3. Genetik Programlama Operatörleri	8
1.2.3.1. Seçim Operatörü	8
1.2.3.1.1. Turnuva Operatörü	8
1.2.3.1.2. Rulet Tekerleği Operatörü	9
1.2.3.2. Çaprazlama Operatörü	10

1.2.3.3. Mutasyon Operatörü	11
1.2.3. Genetik Programlama Türevleri.....	13
1.2.3.1. Geometrik Semantik Genetik Programlama.....	13
1.2.3.2. Gen İfade Programlama (Gen Expression Programming-GEP).....	13
1.2.3.3. Lineer Genetik Programlama (Linear Genetic Programming-LGP)	13
1.2.3.3.1. Genetik Paralel Programlama (Genetic Parallel Programming)	14
1.2.3.3.2. Çok Amaçlı Genetik Programlama (Multi Objective Genetic Programming- MOGP).....	14
1.2.3.4. Çok Genli Genetik Programlama (Multigene Genetic Programming)	14
1.3. Diğer Otomatik Programlama Yöntemleri.....	15

2. BÖLÜM

YAPAY ARI KOLONİ PROGRAMLAMA

2.1. Yapay Arı Koloni Programlama (ABCP).....	22
2.1.1. ABC Algoritması.....	22
2.1.2. ABCP Algoritması	23
2.1.2.1. ABCP Akış Şeması	24
2.1.2.2. ABCP Algoritmasının Bileşenleri	27
2.1.2.2.1. Kolonideki Başlangıç Çözümlerinin Üretilmesi	27
2.1.2.2.2. Uygunluk Fonksiyonu	31
2.1.2.2.3. Kontrol Parametreleri	31
2.1.2.2.4. Bilgi Paylaşım Mekanizması	32
2.1.2.2.5. Durma Kriterleri	33

3. BÖLÜM

TEST PROBLEMLERİNİN ÇÖZÜMÜNDE ABCP'NİN UYGULANMASI

3.1. Giriş	35
3.2. Deney Çalışmaları	38
3.2.1. Deneyler	38
3.2.2. Uygunluk Fonksiyonu ve Parametreler	40
3.3. Sonuçlar	42
3.3.1. Simülasyon Sonuçları	42
3.3.2. ABCP Tarafından Üretilen Modellerin Değerlendirilmesi.....	44

4. BÖLÜM

ÖZNİTELİK SEÇİMİ VE SINIFLANDIRMA PROBLEMLERİNİN YAPAY ARI KOLONİ PROGRAMLAMA İLE ÇÖZÜMÜ

4.1. Problem Tanımı.....	47
4.2. Deney Çalışmaları	50
4.2.1. Veri Setleri.....	50
4.2.2.Eğitim ve Test Setleri.....	52
4.2.3. Uygunluk Fonksiyonu Tanımlanması.....	53
4.2.4. Kullanılan Yöntemlerin Kontrol Parametreleri	54
4.3. Sonuçlar	55
4.3.1. Simülasyon Sonuçları	55
4.3.2. Çıkarılan Modellerin Analizi.....	55

5. BÖLÜM

YÜKSEK BOYUTLU SEMBOLİK REGRESYON PROBLEMLERİNDE ÖZNİTELİK SEÇİMİ İÇİN MHABCP YAKLAŞIMI

5.1. Problem Tanımı.....	60
5.2. MHABCP (Multi Hive Artificial Bee Colony Programming-MHABCP).....	61
5.3. Deney Çalışmaları	64
5.3.1. Uygunluk Fonksiyonu	64

5.3.2. Kontrol Parametreleri	64
5.3.3. Veri Setleri.....	65
5.4. Sonuçlar ve Tartışmalar	67
5.4.1. Eğitim Setleri Sonuçları- Öğrenme Kabiliyeti	67
5.4.2. Test Setleri Sonuçları – Genelleme Kabiliyeti.....	69
5.4.3. Öznitelik Seçimi Sonuçları.....	70
5.4.4. Çıkarılan Modellerin Analizi.....	71

6. BÖLÜM

ÇOK SINIFLI DESEN SINIFLANDIRMA İÇİN YAPAY ARI KOLONİ PROGRAMLAMA TANIMLAYICI

6.1. Problem Tanımı.....	78
6.2. Görüntü Tanımlayıcılar	80
6.2.1. Yerel İkili Örüntü (Local Binary Pattern-LBP)	80
6.2.2. Genetik Programlama Tanımlayıcı.....	82
6.3. Önerilen Metot	82
6.3.1. Algoritmanın Genel İşleyişi.....	82
6.3.2. ABCP Descriptor Program İşleyişi	83
6.3.3. Uygunluk Fonksiyonu	84
6.4. Deney Çalışmaları.....	89
6.4.1. Veri Setleri.....	89
6.4.2. Kontrol Parametreleri	91
6.6. Sonuçlar ve Tartışmalar	92
6.6.1. Genel Sonuçlar	93
6.6.2. Model Analiz Sonuçları	94

7. BÖLÜM

TARTIŞMA, SONUÇ VE ÖNERİLER

7.1. Tartışma ve Sonuç.....	99
7.2. Öneriler	102
KAYNAKÇA	103
ÖZGEÇMİŞ.....	119



KISALTMALAR

GP	Genetik Programlama (Genetic Programming)
GA	Genetik Algoritma (Genetic Algorithm)
LGP	Lineer Genetik Programlama (Linear Genetic Programming)
CGP	Kartezyen Genetik Programlama (Cartesian Genetic Programming)
GEP	Gen İfade Programlama (Gen Expression Programming)
AP	Karınca Programlama (Ant Programming)
AISP	Yapay Bağışıklık Sistemi Programlama (Artificial Immune System Programming)
CSP	Klon Seçim Programlama (Clone Selection Programming)
IP	Bağışıklık Programlama (Immune Programming)
ABCP	Yapay Arı Koloni Programlama (Artificial Bee Colony Programming)
ABC	Yapay Arı Kolonisi (Artificial Bee Colony)
MGP	Çok Genli Genetik Programlama (Multigene GP)
MHABCP	Çok Kovanlı Yapay Arı Koloni Programlama (Multi Hive Artificial Bee Colony Programming)
GSGP	Geometrik Semantik Genetik Programlama (Geometric Semantic Genetic Programming)
LGP	Lineer Genetik Programlama (Linear Genetic Programming)
GPP	Genetik Paralel Programlama (Genetic Parallel Programing)
ALU	Aritmetik İşlemci Ünitesi (Arithmetic Logic Unit)
EE	Evrimsel Motoru (Evolution Engine)
MOGP	Çok Amaçlı Genetik Programlama (Multi Objective Genetic Programming)
NSGA	Baskın Olmayan Sıralama Genetik Algoritma (Non-Dominated Sorting Genetic Algorithm)
EP	Evrimsel Programlama (Evolutionary Programming)
ACOP	Doğrusal Zorunlu Programlama İçin Karınca Koloni Optimizasyonu (Ant Colony Optimization For Linear Imperative Programming)

LDEP	Diferansiyel Gelişim ile Doğrusal Komut Programlama (Linear Imperative Programming with Differential Evolution)
GPU	Grafik İşlemci Ünitesi (Graphic Processing Unit)
CUDA	Hesaplanmış Birleşik Cihaz Mimarisi (Compute Unified Device Architecture)
ACP	Karınca Kolonisi Programlama (Ant Colony Programming)
MRDE	Matris Riccati Diferansiyel Denklemleri (Matrix Riccati Differential Equations)
RK4	Runge–Kutta 4th
CAP	Kartezyen Karınca Programlama (Cartesian Ant Programming)
ACO	Karınca Koloni Optimizasyonu (Ant Colony Optimization)
LNEP	Log-Normal Evrimsel Programlama (Log-Normal Evolutionary Programming)
ILNEP	Bağışıklık Log-Normal Evrimsel Programlama (Immune Log-Normal Evolutionary Programming)
MRI	Manyetik Rezonans Görüntüleme (Magnetic Resonance Imaging)
CT	Bilgisayarlı Topografi (Computed Topography)
RG	Bölge Yetiştiriciliği (Region Growing)
RF	Radyo Frekansı (Radio Frequency)
LSB	En Az Anlamlı Bit (Least-Significant-Bit)
GPPI	Permütasyon Önceliği Olan Genetik Programlama (GP with Permutation Importance)
GPC	Genetik Programlama Tabanlı Sınıflandırıcı (Genetic Programming Based Classifier)
CNN	Evrimsel Sinir Ağları (Convolutional Neural Networks)
GGP	Gramer Tabanlı Genetik Programlama (Grammar Based Genetic Programming)
ES	Evrimsel Stratejiler (Evolutionary Strategies)
GPS	Küresel Konumlama Sistemi (Global Positioning System)
DES	Veri Şifreleme Standardı (Data Encryption Standard)
SSR	Sıralı Sembolik Regresyon (Sequential Symbolic Regression)
GRSR	Genetik Özyinelemeli Sembolik Regresyon (Genetic Recursive Symbolic Regression)
SR	Sembolik Regresyon (Symbolic Regression)

PSOCoM	Kütle Merkezi ile Parçacık Sürüsü Optimizasyonu (Particle Swarm Optimization with Center of Mass Technique)
BJ	Box-Jenkins
NN	Sinir Ağları (Neural Network)
QSAR	Nicel Yapı Etkinlik İlişkisi (Quantitative Structure Activity Relationship)
MDA	Çoklu Ayrımcı Analiz (Multiple Discriminant Analysis)
MVGPC	Çoğunluk Oylu Genetik Programlama Sınıflandırıcı (Majority Voting Genetic Programming Classifier)
EPS	Deprem Tahmin Sistemi (Earthquake Predictor System)
MaXReL	Maksimal İlişki (Maximal Relevance)
mRMR	Minimal Yedeklilik- Maksimum Alaka (Minimal-Redundancy-Maximum-Relevance)
CRF	Koşullu Rastgele Alan (Conditional Random Field)
SVM	Karar Destek Makineleri (Support Vector Machines)
EEG	Epileptic Elektroansefalografi
k-NN	k-En Yakın Komşu (k-Nearest Neighbor)
FEA	Öznitelik Çıkarma Hızlandırıcısı (Feature Extraction Accelerator)
ICA	Bağımsız Bileşen Analizi (Independent Component Analysis)
PCA	Temel Bileşen Analizi (Principal Component Analysis)
MBPSO	Mutasyon Operatörlü İkili Parçacık Sürü Optimizasyonu (Binary PSO with Mutation Operator)
GP-CPS	Genetik Programlama Tabanlı Karşılaştırmalı Partner Seçimi (Genetic Programming- Comparative Partner Selection)
WDBC	Wisconsin Teşhis Göğüs Kanseri (Wisconsin Diagnostics Breast Cancer)
FSALPS	Öznitelik Seçiminin Yaş Tabanlı Popülasyon Yapısı (Feature Selection Age Layered Population Structure)
ALPS GP	Yaş Tabanlı Popülasyon Genetik Programlama (The Age Layered Population Structure Genetic Programming)
SE	Alt Topluluk (Sub-Ensemble)
PSO	Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)
MLP	Çok Katmanlı Algılayıcı (Multi-Layer Perception)
SMO	Sıralı Minimum Optimizasyon (Sequential Minimal Optimization)

NB	Naive Bayes
FSSEM	Beklenti Maksimizasyonu Kullanarak Öznitelik Altküme Seçimi (Feature Subset Selection using (EM) clustering)
EM	Beklenti Maksimizasyonu (Expectation-Maximization)
sNN	Simetrik En Yakın Komşu Kuralını (Symmetric Nearest Neighbor)
FS	Öznitelik Seçimi (Feature Selection)
HEM	Kovan Değişim Mekanizması (Hive Exchange Mechanism)
NRMSE	Normalize Edilmiş Ortalama Hatanın Karakökü (Normalised Root Mean Square Error)
RMSE	Kare Ortalamalarının Karekökü (Root Mean Square Error)
LD50	Median Lethal Dose
DLBCL	Diffüz Büyük B Hücreli Lenfoma (Diffuse Large B-Cell Lymphoma)
CCUN	Topluluklar ve Suç ile Normalleştirilmemiş Veri Seti (Non-normalized dataset of Communities and Crime)
CCN	Topluluklar ve Suç ile Normalleştirilmiş Veri Seti (Normalized dataset with Communities and Crime)
SIFT	Değişmeyen Ölçekli Dönüşüm (Scale Invariant Transform)
SURF	Hızlandırılmış Sağlam Öznitelikler (Speeded up Robust Features)
HOG	Yönlendirmiş Gradyanların Histogramı (Histogram of Oriented Gradients)
2TGP	İki Aşamalı Genetik Programlama (Two Tier Genetic Programming)
LBP	Yerel İkili Örüntü (Local Binary Pattern)
CI	Merkezi Yoğunluk (Central Intensity)
NI	Komşuların Yoğunluğu (Neighbours Intensity)
RD	Radyal Fark (Radial Difference)
AD	Açısal Fark (Angular Difference)
WLBP	Wigner Tabanlı Yerel İkili Örüntü (Wigner-Based Local Binary Patterns)
FBLBP	Öznitelik Tabanlı Yerel İkili Örüntü (Feature Based Local Binary Pattern)

TABLOLAR LİSTESİ

Tablo 1.1. Çok genli genetik programlama modeli örneği.	15
Tablo 3.1. Test problemleri.	39
Tablo 3.2. GP ve ABCP için parametreler.	41
Tablo 3.3. Simülasyon sonuçları.	42
Tablo 3.4. ABCP’de en iyi modellerin eşitlikleri.	45
Tablo 3.5. ABCP’de her veri seti için en iyi modellerin özellikleri.	46
Tablo 4.1. Veri setlerinin özellikleri.	52
Tablo 4.2. Deneylerde kullanılan parametreler.	54
Tablo 4.3. Simülasyon sonuçları.	56
Tablo 4.4. GP ve ABCP ile elde edilen en iyi modeller.	57
Tablo 4.5. Elde edilen en iyi çözüm ağaçlarının özellikleri.	58
Tablo 4.6. En iyi sonuçlarda öne çıkan öznelilikler.	59
Tablo 5.1. Parametreler.	65
Tablo 5.2. Yapay veri setlerinin özellikleri.	65
Tablo 5.3. Gerçek veri setlerinin özellikleri.	66
Tablo 5.4. En önemli 10 öznelilik.	70
Tablo 5.5. MHABCP’de F_1 , F_2 , LD50 veri setleride çıkarılan en iyi modellerin eşitlikleri.	72
Tablo 5.6. MHABCP’de DLBCL ve CCUN veri setlerinde çıkarılan en iyi modellerin eşitlikleri.	73
Tablo 5.7. MHABCP’de CCN veri setinde çıkarılan en iyi modelin eşitlikleri.	74
Tablo 5.8. MHABCP’de en iyi modellerin analizi.	75
Tablo 5.9. GP, ABCP, MHABCP simülasyon sonuçları	76
Tablo 5.10. MHABCP’nin GP ve ABCP’ye karşı Friedman testi sonuçları.	77
Tablo 6.1. Veri setleri.	91
Tablo 6.2. Deneylerde Kullanılan Kontrol Parametreleri.	92
Tablo 6.3. Tanımlayıcıların simülasyon sonuçları.	93
Tablo 6.4. DS-Broadtz için ABCP-tanımlayıcının en iyi modelinin alt ağaçlarının özellikleri.	94

Tablo 6.5. DS-Broadtz'da ABCP-Tanımlayıcı en iyi modelin alt ağaçlarının eşitlikleri.	96
Tablo 6.6. DS-Kylberg için ABCP-Tanımlayıcı en iyi model.....	97
Tablo 6.7. DS-Kylberg'de ABCP-tanımlayıcı en iyi modelin alt ağaçlarının eşitlikleri.	97



ŞEKİLLER LİSTESİ

Şekil 1.1. Genetik programlama hazırlık adımları.	6
Şekil 1.2. Genetik programlama akış şeması.	7
Şekil 1.3. Turnuva operatörünün işleyişi.	9
Şekil 1.4. Rulet tekerleği operatörünün işleyişi.	9
Şekil 1.5 Çaprazlama operatörünün işleyişi.	11
Şekil 1.6. Alt ağaç mutasyon operatörünün işleyişi.	12
Şekil 1.7. Tek düğüm mutasyon operatörünün işleyişi.	12
Şekil 2.1. ABC algoritması sözde kodu.	23
Şekil 2.2. ABC (a) ve ABCP (b)'de birey temsilleri.	24
Şekil 2.3. ABCP algoritması akış diyagramı.	25
Şekil 2.4. ABCP algoritmasının adımları.	27
Şekil 2.5. Full metodu kullanarak ağaç üretilmesi.	29
Şekil 2.6. Grow metodu kullanarak ağaç üretilmesi.	30
Şekil 3.1. Cherkassy fonksiyonunda ABCP'nin en iyi modelinin tahmini ve gerçek veri noktaları arasındaki ilişki.	43
Şekil 3.2. pH veri seti için ABCP'nin en iyi modelinin tahmini ve gerçek veri noktaları arasındaki ilişki.	43
Şekil 3.3. Beton Basınç Dayanımı veri seti için ABCP'nin en iyi modelinin tahmini ve gerçek veri noktaları arasındaki ilişki.	44
Şekil 5.1. MHABCP sembolik model örneği.	62
Şekil 5.2. MHABCP'de kovan değişim mekanizmasına örnek.	63
Şekil 5.3. En iyi bireylerin eğitim NRMSE dağılımı.	67
Şekil 5.4. Eğitim gelişim grafikleri.	68
Şekil 5.5. En iyi bireylerin test NRMSE dağılımı.	69
Şekil 6.1. Örnek LBP parametre değerleri.	81
Şekil 6.2. LBP kodunun üretilerek onluk tabana çevirmesi örneği.	81
Şekil 6.3. Uniform ve non uniform LBP kodu örnekleri.	81
Şekil 6.4. Algoritmanın genel işleyişi.	83
Şekil 6.5. Satır vektör temsili gösterimi.	83
Şekil 6.6. ABCP program temsili gösterimi.	84

Şekil 6.7. Mantıksal fonksiyonlar.	85
Şekil 6.8. İki vektör arasındaki mesafenin hesaplanması (sözde kod 1).....	87
Şekil 6.9. Sınıflar arası ve sınıf içi uzaklığın hesaplanması (sözde kod 2).....	88
Şekil 6.10. Histogramlar arasında mesafenin karşılaştırılması	89
Şekil 6.11. DS-Broadtz veri seti sınıf örnekleri.	90
Şekil 6.12. DS- Kylberg veri seti sınıf örnekleri.....	91
Şekil 6.13. DS-Broadtz’da ABCP-Tanımlayıcının en iyi modeli.	95
Şekil 6.14. DS-Kylberg’de ABCP-Tanımlayıcının en iyi modeli.	98



GİRİŞ

Tezin Amacı, Kapsamı ve Katkıları

Otomatik programlama yöntemleri, bir sistemde eldeki verileri kullanarak model çıkarmayı hedefleyen yöntemlerdir. Sistemleri iyi tanımlayan modellerin üretilmesi ile çıkışların en az hatayla tahmin edilmesi sağlanır. Yapay Arı Kolonisi (ABC) algoritması, Karaboğa tarafından önerilen, bal arılarının yiyecek kaynağı arama davranışlarını simüle ederek optimum çözümünü arayan, sürü zekasına dayalı algoritmadır [26]. Yapay Arı Koloni Programlama (ABCP), ABC algoritmasından esinlenerek sembolik regresyon problemi için geliştirilmiş otomatik programlama yöntemidir. Bu tez çalışması ile çeşitli makine öğrenmesi uygulamalarındaki birçok probleme etkin çözümler sunabilecek yeni ABCP tabanlı yöntemler önerilmiş, bu yöntemlerin performansları deneysel çalışmalar ile incelenmiş ve sonuçlar rakip yöntemlerle kıyaslanarak yöntemlerin etkinliği ortaya konmuştur. Bu problemlere örnek olarak sembolik regresyon, öznitelik seçimi, sınıflandırma ve tahmin etme verilebilir. Sembolik regresyon probleminde, bağımlı ve bağımsız değişkenler arasındaki matematiksel eşitlik belirlenir. Öznitelik seçimi problemlerinde, model çıkarımında sistemlerdeki gereksiz/ilgisiz girişlerin elenerek çıkışların daha doğru tahmin edilebilmesi amaçlanır. Sınıflandırma problemlerinde, verilerin daha etkin ve doğru kategorize edilebilmesi; Tahmin etme problemlerinde, sistemlerin çıkışlarının modellerle doğru kestirebilinmesi hedeflenir.

Problemlerde ABCP'nin başarısı, en çok bilinen ve kullanılan otomatik programlama yöntemi Genetik Programlama (GP) ile kıyaslanmıştır. Bu tezde, öncelikle GP ve ABCP yöntemleri detaylı açıklanarak yöntemlerle ilgili literatür çalışmalarına yer verilmiştir. Ayrıca, evrimsel algoritmalara dayalı otomatik programlama yöntemleri hakkında bilgi verilmiştir. ABCP'nin performansını izlemek adına yapay ve gerçek veri setleri üzerinde deneyler yapılmıştır. Ardından tezde, ABCP algoritması temel alınarak geliştirilen Çok Kovanlı ABCP (MHABCP) algoritması önerilmiş ve yüksek boyutlu sembolik regresyon problemi üzerindeki başarısı test edilmiştir. MHABCP, literatüre ilk kez ABCP'ye dayalı

yeni bir yöntem olarak sunulmuştur. Tez kapsamında, desen sınıflandırma probleminde GP-tanımlayıcıdan ilham alınarak ABCP'ye dayalı ABCP-tanımlayıcı önerilmiş ve bilinen desen veri setleri kullanılarak rakip yöntemlerle başarısı kıyaslanmıştır.

Çalışmanın Bölümleri

Tezin ilk bölümünde Genetik Programlama (GP) açıklanmıştır. Ayrıca Karınca Programlama (AP), Yapay Bağışıklık Sistemi Programlama (AISP) gibi evrimsel hesaplama algoritmalarına dayalı otomatik programlama yöntemleri ve farklı bilimsel alanlardaki uygulamaları ile ilgili mevcut literatür gözden geçirilmiştir.

İkinci bölümde, Yapay Arı Koloni Programlamanın (ABCP) detaylı analiz edilerek daha kolay anlaşılması amaçlanmıştır. Bu sebeple, öncelikle ABC algoritması açıklanmıştır. ABCP algoritmasının akış şeması ve algoritma bileşenleri incelenmiştir.

Üçüncü bölümde sembolik regresyon, tahmin ve öznitelik seçimi problemlerinde GP ve ABCP'nin performansları karşılaştırılmıştır. Sembolik regresyon problemi olarak Cherkassy fonksiyonu, tahmin problemi olarak pH veri seti modellemesi ve öznitelik seçimi problemi olarak gürültü ile zorlaştırılmış Beton Basınç Dayanımı veri seti kullanılmıştır. Her üç deneyde de, ABCP sembolik regresyonda daha doğru matematiksel model üretme, daha iyi model tahmini ve gereksiz öznitelikleri çıkararak önemli girişleri matematiksel modelde kullanma bakımından GP'den daha yüksek performans göstermiştir.

Tezin dördüncü bölümünde, sınıflandırma problemlerinde GP ve ABCP yöntemleri tarafından çıkarılan modellerin öznitelik seçme kabiliyetleri araştırılmış ve bu alanla ilgili literatür çalışmasına yer verilmiştir. Performans analizi için dört sınıflandırma veri seti kullanılmıştır. Deney sonuçlarında, her iki yöntemle çıkarılan modellerin eşitliklerinde ortak öznitelikleri kullandıkları görülmüştür. Deneyler, ABCP'nin ürettiği eşitliklerin, GP ile rekabet edecek sınıflandırma sonuçlarına sahip olduğunu göstermiştir.

Tezin beşinci bölümünde, yüksek boyutlu sembolik regresyon problemlerinde öznitelik seçim yöntemleri araştırılmış ve yüksek boyutlu öznitelik seçim problemleriyle ilgili kısa bir literatür taraması sunulmuştur. Çalışmada, son zamanlarda önerilen, yüksek seviye otomatik programlama tekniği olan ABCP yönteminden esinlenerek, yüksek boyutlu

sembolik regresyon problemi için Çok Kovanlı Yapay Arı Koloni Programlama (MHABCP) yöntemi önerilmiştir. Önerilen yöntem, altı farklı yüksek boyutlu veri setinde test edilmiş ve yöntemlerin sonuçları en iyi bilinen otomatik programlama tekniği olan GP'nin sonuçlarıyla karşılaştırılmıştır. Deneylerde; MHABCP, ABCP ve GP ile yüksek boyutlu matematiksel modeller çıkarılmıştır. Standart GP ve ABCP ile karşılaştırıldığında, MHABCP'nin eğitim setinde başarılı modelleri çıkarabilir olduğu ve test setinde genelleştirme kabiliyetine sahip olduğu görülmüştür.

Tezin altıncı bölümünde, desen sınıflandırma problemini çözmek için GP-tanımlayıcıdan esinlenerek geliştirilen ABCP-tanımlayıcı önerilmiştir. Tanımlayıcı eğitim aşamasında her desen için sadece iki örnek desen rastgele seçerek desen setlerini tanımlamaya çalışır. Önerilen yöntemde, örnekler 5x5 boyutlarında pencere kullanılarak satır vektörlerine dönüştürülmüştür. Satır vektörleri ABCP-Tanımlayıcı için modellerin girişleri olarak kullanılmıştır. LBP'nin çalışma mantığına benzer şekilde, modeller sonucunda elde edilen ikili kodlar onluk tabana çevrilerek histogramda karşılık gelen değeri 1 arttırılmıştır. Tanımlayıcıda uygunluk fonksiyonu, sınıflandırma başarısı ve mesafe olmak üzere iki bileşenden oluşturulmuştur. Sınıflandırma başarısı, histogramları değerlendirirken (1-NN) sınıflandırıcı kullanarak eğitim örneklerini doğru sınıflandırma yeteneğini ölçmektedir. Mesafe işlevi, eğitim örneklerinde sınıf içi ve sınıflar arasında uzaklıkları hesaplamaktadır. ABCP-tanımlayıcının, standart LBP ve GP-Tanımlayıcı ile kıyaslanarak performansı değerlendirilmiştir. Kıyaslamalı sonuçlar, ABCP-tanımlayıcının sınıflandırma başarısı yüksek modeller oluşturduğu ve bu alanda kullanılabilecek yeni bir metot olduğunu göstermiştir.

Yedinci bölümde, tez çalışmasında elde edilen sonuçlar değerlendirilmiş ve ileriki çalışmalarda yapılabilecekler tartışılmıştır.

1. BÖLÜM

OTOMATİK PROGRAMLAMA YÖNTEMLERİ VE UYGULAMALARI

1.1. Giriş

Otomatik programlama yöntemleri, birçok mühendislik probleminde pratik çözümler sağlayan model üretme sürecidir [1].

Genetik Programlama (Genetic Programming-GP), 1985 yılında Cramer [2] tarafından tanımlanmıştır ve Koza tarafından geliştirilmiştir [3]. GP, Genetik Algoritmanın (Genetic Algorithm-GA) bir uzantısıdır ve en tanınmış evrimsel algoritma temelli otomatik programlama yöntemidir. GP algoritma adımları GA'nın adımlarına benzerdir. Çözümler; GA'da sabit boyutlu dizilerle, GP'de yapısı terminal ve fonksiyonların kombinasyonundan oluşturulmuş parçalı ağaçlarla temsil edilir. Ağaçların en küçük birimi, düğüm olarak adlandırılır. Düğümler, problemler için özel tanımlanan terminal kümesinden (x, y gibi değişkenler ve sabitler) ve fonksiyon kümesinden (aritmetik operatörler, mantıksal fonksiyonlar, matematiksel fonksiyonlar) seçilirler. Bu düğümlerin birleşimi ile çözümleri temsil eden parçalı ağaçlar oluşturulur. Günümüzde, GP'nin Lineer Genetik Programlama (Linear Genetic Programming-LGP), Kartezyen Genetik Programlama (Cartesian Genetic Programming-CGP), Gen İfade Programlama (Gen Expression Programming-GEP) gibi türevleri bulunmaktadır. Roux ve Fonlupt, karınca kolonilerinin yuvaları ve yiyecek kaynakları arasındaki en kısa mesafeyi optimize eden karınca koloni algoritmasından ve GP yönteminden esinlenerek Karınca Programlama (Ant Programming-AP) geliştirmişlerdir [4]. Programda, ağaç yapısında çözümlerin her bir düğümü fenomen tablosuna sahiptir. Çalışmada, AP regresyon ve çoklayıcı problemleri üzerinde GP ile kıyaslanmıştır.

Yapay Bağışıklık Sistemleri, doğal bağışıklık sistemlerinin bir organizmaya yapılan saldırılara cevap vermeyi öğrenmelerinden ilham alarak geliştirilen bilgisayar algoritmalarıdır. Johnson, yapay bağışıklık sistemlerinin, genetik programlamaya alternatif olarak kullanılabileceği Yapay Bağışıklık Sistemi Programlama (artificial immune system programming-AISP) önermiştir [5]. Önerilen metot sembolik regresyon probleminde kullanılmıştır. Gan ve diğerleri, çalışmalarında çözümleri antikolların yapısından esinlenerek kodlayan Klon Seçim Programlama (Clone Selection Programming-CSP) önermişlerdir [6]. Çalışmada klon seçim prensibi bir arama stratejisi olarak geliştirilerek GEP ve Bağışıklık Programlama (Immune Programming-IP) ile kıyaslanmıştır.

Yapay Arı Koloni Programlama (artificial bee colony programming-ABCP), ABC algoritmasına dayanan yüksek seviyeli otomatik programlama metodudur [7, 8]. ABCP algoritmasının adımları, Yapay Arı Koloni (artificial bee colony-ABC) algoritmasına benzerdir. Bu algoritmalar arasındaki en temel iki fark; çözüm yapısının temsili ve çözümlerin geliştirilmesini sağlayan bilgi paylaşım mekanizmasıdır. Karaboğa ve diğerleri, ABC algoritmasına dayalı bir otomatik programlama yöntemi olan ABCP'yi ilk kez geliştirmişlerdir. Çalışmada ABCP, standart GP, GP'nin farklı versiyonları, karınca koloni optimizasyonu temelli bir otomatik programlama yöntemi ile literatürdeki en çok kullanılan test problemleri üzerinde karşılaştırılmıştır [7]. Arslan ve Ozturk, standart ABCP üzerinde bazı modifikasyonlar yapıp Çok Genli GP'den (Multigene GP-MGGP) esinlenerek Çok Kovanlı ABCP (Multi Hive ABCP-MHABCP) olarak adlandırdıkları yeni bir ABCP versiyonu ortaya koymuşlardır [8]. Önerilen versiyon, yüksek boyutlu sembolik regresyon veri setleri üzerinde standart ABCP ve GP ile test edilmiştir. MHABCP'nin eğitim kümesindeki öğrenme kabiliyeti ve test kümesindeki genelleme yeteneği bu alanda kullanılabilir bir algoritma olduğunu göstermiştir.

1.2. Genetik Programlama

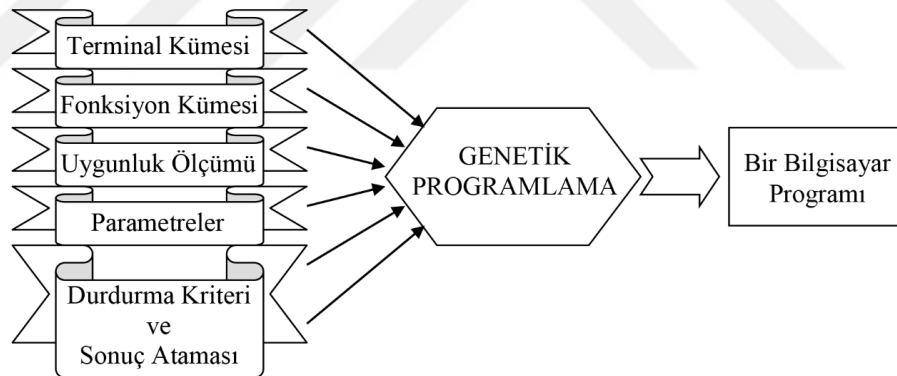
GP, GA'dan esinlenerek geliştirilmiş farklı mühendislik problemlerine uyarlanabilen bir metoddur. GA ve GP arasındaki en büyük fark, çözümlerin temsildir. GA yönteminde popülasyondaki bireyler sabit uzunlukta karakter dizileriyken, GP'de Lisp programlama dilindeki ağaç yapılarına benzer hiyerarşik değişken boyutlu ağaç yapıları

kullanılmaktadır. GP ve versiyonları, birçok farklı alanda başarılı sonuçlar ürettiğinden en çok tercih edilen yöntemlerdir. Bölüm 1.2.1’de Standart GP tanıtılmaktadır.

1.2.1. Genetik Programlama Hazırlık Adımları

1985 yılında Cramer ilk modern uygulamaya benzeyen ağaç yaklaşımını Markov karar süreçlerine uygulamıştır [2]. Bu yaklaşımı Koza, karmaşık iyileştirme ve arama problemlerine uygulamıştır. Standart GP, ağaç yapısında üretilmiş popülasyondaki çözümlere seçim, çaprazlama, mutasyon gibi operatörler kullanarak optimum sonuca ulaşmayı hedefleyen otomatik programlama yöntemidir [3].

Kullanıcı, problemin üst düzey ifadesini, belirli iyi tanımlanmış hazırlık adımlarını uygulayarak GP sistemine iletir. Şekil 1.1, standart genetik programlama için beş ana hazırlık adımını göstermektedir. Hazırlık aşamaları (solda), genetik programlama sistemine kullanıcı tarafından verilen girişlerdir. Bilgisayar programı (sağda) genetik programlama sisteminin çıktısıdır [9].



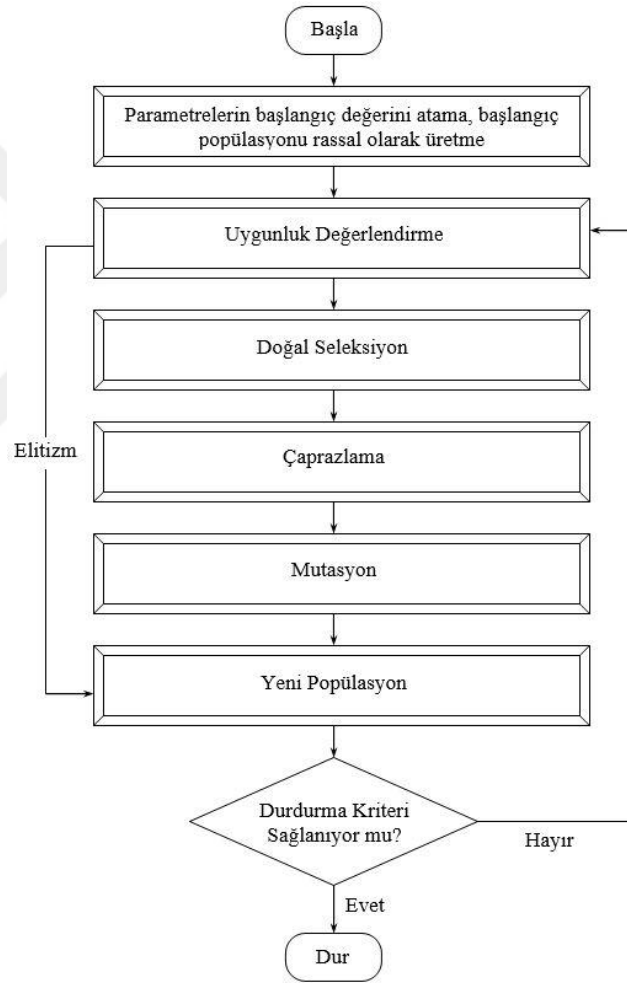
Şekil 1.1. Genetik programlama hazırlık adımları.

GA, çözümleri sabit kod dizileri şeklinde ifade ederken; GP, çözümleri parçalı ağaçlarla temsil eder. Ağaçların en küçük birimi düğüm olarak adlandırılır. Düğümler, problemler için özel tanımlanan terminallerden veya fonksiyonlardan seçilir.

1.2.2. Genetik Programlama Akışı

GP akış şeması Şekil 1.2’de gösterilmiştir. Akış şemasında ilk adım başlangıç popülasyonunun oluşturulmasıdır. Popülasyondaki bireyler, full metodu, grow metodu veya ramped half and half metodu ile üretilebilir [10]. GA’daki seçim operatörü,

çaprazlama ve mutasyon operatörlerinin çalışma mantığı, GP'ye mevcut çözümleri geliştirmek ve yeni çözümleri aramak amacıyla uyarlanır. Seçim operatörü ile popülasyondaki her bireyin probleme uygunluğu değerlendirilir. Bütün durumlarda bireyin uygunluğu önceden belirlenmiş bir uygunluk fonksiyonu ile yapılarak ve bütün ağaç analiz edilerek bulunur [6]. Uygunluk değeri iyi olan bireylerin bir sonraki nesilde bulunma olasılıkları daha fazladır. Bazı çalışmalarda uygunluk değeri kötü olan bireylerin de seçilme durumu yüksektir. Seçim operatörü, rulet çemberi, turnuva metodu gibi yöntemlerle uygulanır.



Şekil 1.2. Genetik programlama akış şeması.

Çaprazlama operatöründe iki ebeveyn birey uygunluk değerine bağlı olarak popülasyondan seçilir. Genellikle her iki ebeveyn için birer çaprazlama noktası rastgele belirlenir. Her iki çaprazlama noktasından alınan alt ağaçlar bireylerde yer değiştirilerek yeni bireylerin üretilmesi sağlanır. Mutasyon operatöründe, bireyden mutasyon noktası/düğümü rastgele belirlenir. GP'de ağaçtan seçilen düğüm yerine rastgele

oluşturulan ağaç yerleştirilir. Bu tip mutasyon alt ağaç mutasyonu olarak adlandırılır. Bir diğer mutasyon yöntemi ise tek nokta mutasyonudur. Bu yöntemde rastgele seçilen düğüm terminal ise terminal kümesinden, fonksiyon ise fonksiyon kümesinden rastgele seçilen değer ile yer değiştirilmesi sağlanır. Mutasyon operatörü yeni, görülmemiş ve araştırılmamış çözüm elemanlarının bulunmasını sağlar [11]. Elitizm ile bir önceki nesildeki en iyi çözümler mevcut nesle aktarılır. Önceden belirlenen durdurma kriterlerine (çözümlerin belirli uygunluk değeri, iterasyon sayısı gibi) ulaşması sağlandığında program sonlandırılır. Diğer bölümde GP operatörleri detaylı açıklanmıştır.

1.2.3. Genetik Programlama Operatörleri

Genetik Programlama'nın akış şeması; uygunluk fonksiyonunun değerlendirilmesi, doğal seçim, çaprazlama ve mutasyon gibi genetik operatör süreçlerini içerir. Popülasyondaki her birey, probleme ne kadar uygun olduğunu belirlemek için değerlendirilir. Genetik operatörlerle bireyler, uygunluk değeri büyüklüğü olasılığında seçilerek popülasyondaki genetik çeşitliliği sağlar ve uygunluk değeri daha iyi bireyler elde edilmeye çalışılır [12].

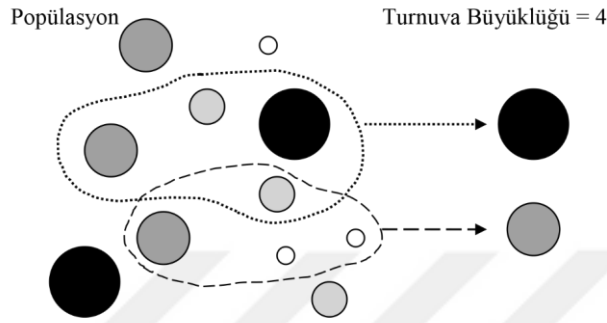
1.2.3.1. Seçim Operatörü

Seçim operatörü, bireylerin uygunluğu değerlendirildikten sonra bir seçim stratejisi kullanılarak bir sonraki nesle aktarılacak olan kromozomların seçimini yapan operatördür [13]. Operatör sonucunda uygunluk değeri yüksek olan bireylerin seçimi ile gelecek nesillerin daha yüksek uygunluk değerine sahip olması hedeflenir. Literatürde çeşitli seçim operatörlerinin, seçim olasılığını değerlendirmek için farklı yaklaşımları vardır [14]. Alt bölümlerde, en çok kullanılan seçim yöntemlerinden olan turnuva seçimi ve rulet seçimi operatörleri açıklanmıştır.

1.2.3.1.1. Turnuva Operatörü

Turnuva seçimi, problemler tarafından kolay uygulanması nedeniyle GP'de en çok kullanılan seçim operatörlerinden biridir [15-17]. Turnuva seçim operatöründe, popülasyondan turnuva büyüklüğü kadar rastgele bireyler seçilerek bir küme oluşturulur. Bu kümedeki bireyler birbirleriyle uygunluk değerleri üzerinden rekabet eder. En iyi uygunluk değerine sahip birey bir sonraki jenerasyona yerleştirilmek amacıyla seçilir [15]. Turnuva büyüklüğü, popülasyon büyüklüğü ile doğru orantılıdır. Turnuva

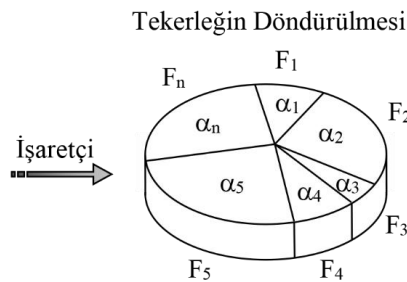
büyüklüğü değiştirilerek seçim baskısı kolayca kontrol edilebilir. Böylece turnuva büyüklüğü popülasyondaki kötü bireylerin sayısından daha büyükse, bu bireylerin seçilme olasılıkları düşük olur. Operatör, tüm bireylerin seçilmesine olanak tanıdığı için çeşitliliği korur. Bu durum optimum çözüme yakınsama hızını azaltabilir. Şekil 1.3 turnuva büyüklüğü 4 seçilmiş bir turnuva operatörünün işleyişini göstermektedir [15].



Şekil 1.3. Turnuva operatörünün işleyişi.

1.2.3.1.2. Rulet Tekerleği Operatörü

Rulet tekerleği operatöründe, uygunluk değeri en yüksek olan bireyin seçilme şansı daha yüksektir. Rulet tekerliğinde, bir sektör seçme olasılığı, sektörün merkezi açısının büyüklüğüne eşittir. GP’de, toplam nüfus çembere bölünmüştür ve her sektör bir bireyi gösterir. Bireyin uygunluk değerinin toplam popülasyonun uygunluk değerine oranı, o bireyin gelecek nesildeki seçim olasılığını belirler. Operatörün dezavantajı, her seçimde popülasyonda bulunan en iyiyi seçme olasılığı yüksek olduğundan bölgesel minimumlara/maksimumlara takılma olasılığının diğer tekniklere göre daha yüksek olmasıdır. Orantılı rulet tekerleği operatörü Şekil 1.4’de gösterilmektedir. Burada popülasyon büyüklüğü n popülasyon büyüklüğünde her bireyin uygunluk değeri F ile ifade edilmiştir. Bireyin uygunluk değeri F arttıkça tekerlekteki açı değeri artmaktadır.



Şekil 1.4. Rulet tekerleği operatörünün işleyişi.

Aşağıdaki noktalar Rulet Tekerleği seçim operatörünün temel adımlarıdır:

1. Popülasyondaki tüm bireyler için uygunluk değerlerinin toplamının hesaplanması.
2. Her bireyin uygunluk değerinin ve tüm bireyler için uygunluk değerlerinin toplamına oranının hesaplanması. Bu oran, bireyin seçilme olasılığını gösterir.
3. Rulet tekerleğini ikinci adımda elde edilen oranlara göre sektörlere ayırma. Her sektör bir bireyi temsil eder. Sektörün alanı, bireyin seçilme olasılığı ile orantılıdır.
4. Tekerleği n popülasyon büyüklüğü kez döndürmek, Rulet tekerleğinin dönmesi durduğunda, işaretçinin gösterdiği alana karşılık gelen bireyin seçilmesi.

Popülasyon büyüklüğü n , popülasyon $P=\{\alpha_1, \alpha_2, \dots, \alpha_n\}$, her α_i 'nin uygunluk değeri $F_i = f(\alpha_i)$, α_i 'nin seçilme olasılığı Eşitlik 1.1'deki gibi hesaplanabilir [18].

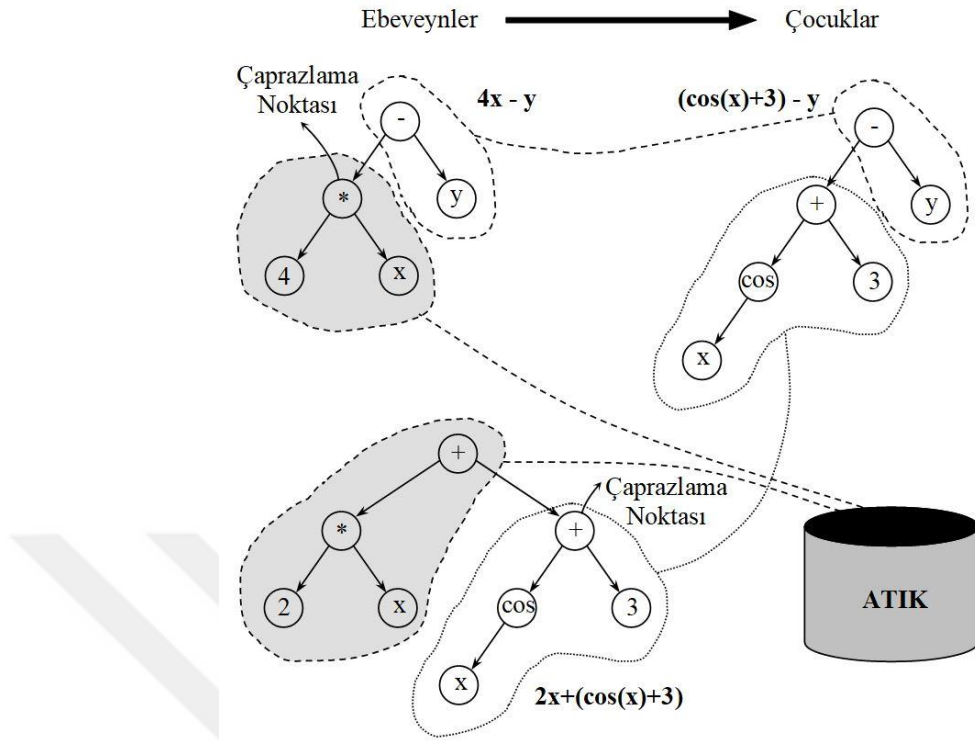
$$P(\alpha_i) = \frac{f(\alpha_i)}{\sum_{j=1}^n f(\alpha_j)}, i, j = 1, 2, \dots, n \quad (1.1)$$

1.2.3.2. Çaprazlama Operatörü

Çaprazlama operatörü, bireyler arasında bilgi alışverişi ile yeni çözümlerin oluşturularak popülasyondaki çeşitliliği artırmayı sağlayan genetik işleme operatörüdür. Seçim operatörü ile oluşturulan yeni popülasyondaki bireyler çaprazlama operatörü için ebeveyn adaylarıdır. Çaprazlama operatörüne tabi tutulacak birey sayısı çaprazlama olasılığından (P_c) ve n popülasyon büyüklüğü olmak üzere ($P_c * n$) ile hesaplanır. Çaprazlama operatörüne tabi tutulacak bireyler eşleme havuzuna konulur. Operatörün adımları alttaki maddelerde sıralanmıştır:

1. Eşleme havuzuna seçilen ebeveynler eşleştirilir.
2. Her bir ebeveyn den rastgele olarak bir alt ağaç seçilir. Seçilen düğümün terminal veya fonksiyon olma olasılığı kullanıcı tarafından tanımlanabilir.
3. Seçilen alt ağaçlar birbirleri ile yer değiştirerek iki tane yeni birey oluşturulur.

Şekil 1.5 çaprazlama operatörünün işleyişini göstermektedir. Çaprazlama operatörü, mutasyon operatörüne nazaran daha baskın bir operatördür. Operatör $[0.8,1]$ gibi yüksek olasılıklarla işletilir [3]. Programın kapalılık özelliği bulunduğundan çaprazlama operatörü sonucunda elde edilen ağaçların programa uygunluğu kontrol edilmelidir.

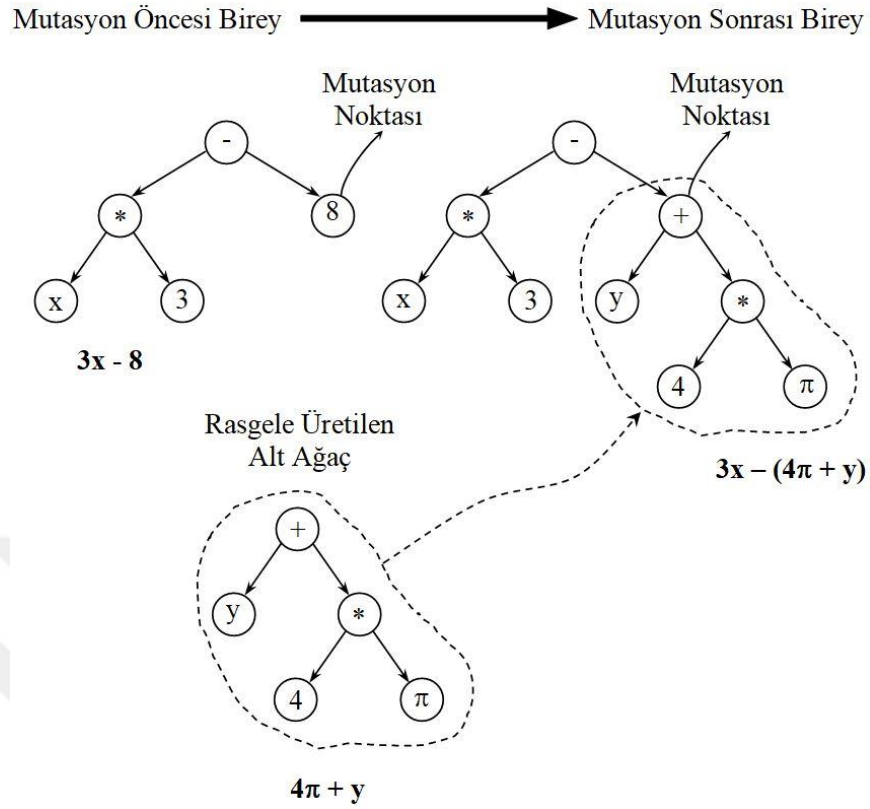


Şekil 1.5 Çaprazlama operatörünün işleyişi.

1.2.3.3. Mutasyon Operatörü

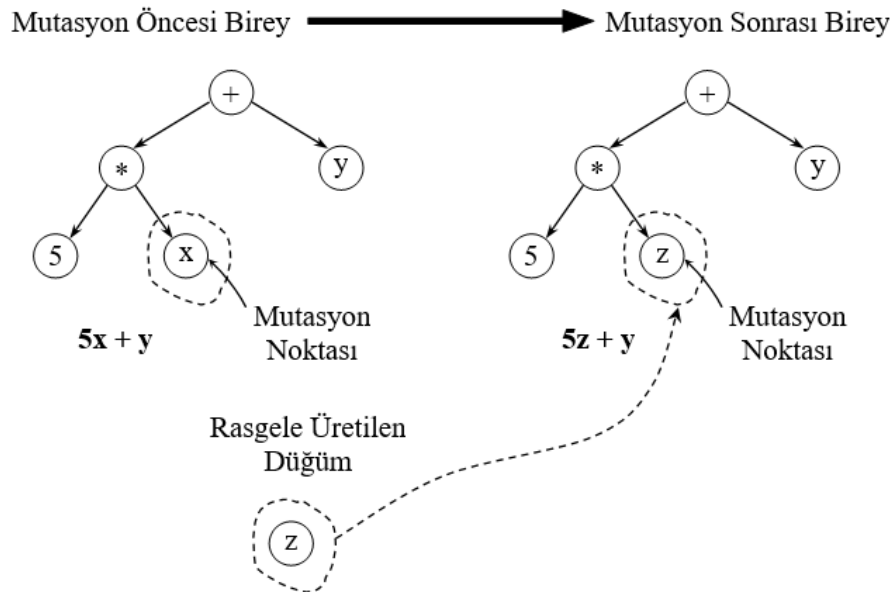
Mutasyon operatörü, GP’de seçim çaprazlama operatörlerinin işleyişinden sonra elde edilen popülasyonda çeşitliliği sağlamak amacıyla kullanılır. Mutasyon operatörü kullanılarak bireylerin benzerliği engellenmeye çalışılır. Bireylerin benzerliği, bireylerin arama uzayında yerel minimuma/maksimuma takılarak tüm çözüm uzayının taranmamasına neden olur. Mutasyon operatörü çözüm uzayında incelenmemiş bölgelere yayılmasını sağlar. Operatör, mutasyon olasılığına (P_m) bağlı olarak işlem gerçekleştirir. Mutasyon operatörü, çaprazlama operatörüne nazaran daha çekinik bir operatördür. Operatör [0.1,0.5] gibi düşük olasılıklarla işletilir [3].

Alt ağaç mutasyonunda bir birey için rastgele bir düğüm seçilir ve bu düğümden itibaren yeni rastgele üretilmiş alt ağaç yer değiştirir. Rastgele seçilen düğüm fonksiyon veya terminal kümesinden olabilir. Alt ağaç mutasyonu Şekil 1.6’da gösterilmiştir.



Şekil 1.6. Alt ağaç mutasyon operatörünün işleyişi.

Tek düğüm mutasyonunda rastgele seçilen düğüm, terminal kümesinden ise terminal kümesinden; fonksiyon kümesinden ise fonksiyon kümesinden rastgele seçilen düğümle değiştirilir. Tek düğüm için mutasyon operatörü örneği Şekil 1.7’de verilmiştir.



Şekil 1.7. Tek düğüm mutasyon operatörünün işleyişi.

Bu operatörler haricinde kullanılan bir diğer operatör yeniden eşleme operatörüdür. Yeniden eşleme operatörü ile kromozomlar yeni nesil popülasyonuna birebir kopyalanırlar [3].

1.2.3. Genetik Programlama Türevleri

Geçtiğimiz yıllarda, farklı problemlerde GP'yi geliştirmek için birçok araştırma yapılarak GP'nin farklı türevleri önerilmiştir. Uygulamada, hemen hemen GP'nin tüm türevleri Şekil 1.2'te gösterilen akış şemasına sahiptir ve farklı genetik operatörler kullanarak çözümleri iyileştirir. Tezin bu bölümünde, geliştirilen bazı GP türevlerine değinilmiştir.

1.2.3.1. Geometrik Semantik Genetik Programlama

Moraglio ve diğerleri, Geometrik Semantik GP (Geometric Semantic Genetic Programming-GSGP) olarak adlandırdıkları GP için geometrik anlamsal operatörleri tanıtmışlardır [19, 20]. GSGP'de yavrular kendi ebeveyn bireylerinden elde edilir, böylece yavrular yapısal olarak değil, anlamsal olarak ebeveynlere benzer olur [21]. GSGP, girdi verilerinin bilinen hedeflerle eşleştirilmesinden oluşan her problem için tek modlu hata düzeyi oluşturabilir. Bununla birlikte algoritmanın ebeveynlerinden daha büyük yavrular üretme konusunda sınırlamaları vardır.

1.2.3.2. Gen İfade Programlama (Gen Expression Programming-GEP)

GEP, GA ve GP ile benzer şekilde, popülasyondaki bireylerin uygunluk değerine göre seçilerek bir veya daha fazla genetik operatör ile bireylerin kalitesini arttırmayı hedefleyen algoritmadır. Bireylerin temsili, üç algoritma arasındaki temel farktır. GA'larda bireyler, sabit uzunlukta doğrusal dizelerle; GP'de bireyler, farklı boyut ve şekillerde doğrusal olmayan parçalı ağaçlarla; GEP'de ise bireyler, içerisinde farklı boyut ve şekillerde doğrusal olmayan yapıları içeren sabit uzunlukta doğrusal dizelerle temsil edilir [22]. Genler baş ve kuyruktan oluşur. Baş, terminal ve fonksiyonları temsil eden sembollerini içerirken; kuyruk sadece terminalleri içerir.

1.2.3.3. Lineer Genetik Programlama (Linear Genetic Programming-LGP)

LGP, popülasyondaki programların güçlü programlama dili veya makine kodundan bir dizi talimat sırası olarak ifade edildiği GP türevidir. Alt bölümlerde Lineer GP tabanlı geliştirilen GP türevleri örneklendirilmiştir.

1.2.3.3.1. Genetik Paralel Programlama (Genetic Parallel Programming)

Lineer Genetik Programlamaya dayanan Genetik Paralel Programlama, çoklu aritmetik işlemcisine dayanan paralel programlar geliştirilmesi için önerilmiştir [23]. GPP; Multi-Aritmetik İşlemci Ünitesi (Arithmetic Processing Unit-ALU), işlemci ve evrim motoru (Evolution Engine-EE) bileşenlerinden oluşur. Multi-ALU İşlemci, GPP'nin çekirdeğidir. İşlemci, tek bir işlemci saat döngüsünde aynı anda birden fazla işlemi gerçekleştirebilir. Birden çok ALU, bir çapraz anahtarlama ağı üzerinden paylaşılan bir kayıt dosyasına erişir. İşlemci, GPP'deki programların uygunluk değerlendirmesi için tasarlanmıştır. Evrim Motoru başlangıç popülasyonunun oluşturulması ve genetik operatörlerinin işleyişinden sorumludur. Evrim döngüsü tamamlandığında EE paralel programı üretir. GPP'nin, sınıflandırma veri setleri üzerinde yapılan deneylerde, diğer sınıflandırma algoritmalarıyla karşılaştırılabilir genelleme performansına sahip olduğu gözlemlenmiştir [23].

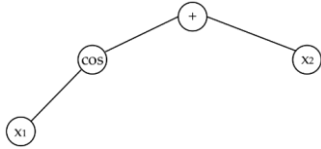
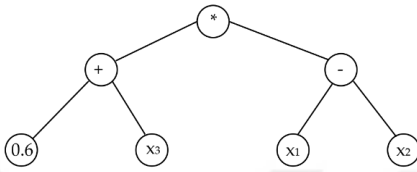
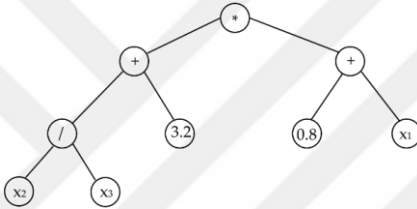
1.2.3.3.2. Çok Amaçlı Genetik Programlama (Multi Objective Genetic Programming- MOGP)

Watchareeruetai ve diğerleri, öznitelik çıkarma programlarının otomatik olarak oluşturulması için Çok Amaçlı Genetik Programlama (MOGP) önermişlerdir [24]. MOGP, Baskın Olmayan Sıralama Genetik Algoritmasına (Non-Dominated Sorting Genetic Algorithm-NSGA-II), nüfus çeşitliliğini artırmak için nüfusun kesilmesi, örneklenmesi ve yavru üretimi gibi artıklık düzenleme mekanizmaları eklenerek geliştirilmiştir.

1.2.3.4. Çok Genli Genetik Programlama (Multigene Genetic Programming)

Çok Genli GP (Multigene Genetic Programming-MGGP), standart GP modellerinden farklı olarak çoklu ağaç kullanımına izin veren GP türevidir. MGGP'de, her bir model, bir veya daha fazla ağacın doğrusal bir kombinasyonudur. Her ağaç, genel modele ağırlıklı bir katkısı olan kısmi bir model parçası olarak düşünülebilir. Modelde, ağaçlar 'genotip' olarak ve kodu çözülmüş ve basitleştirilmiş ağaç yapısı 'fenotip' olarak adlandırılırlar [25]. Tablo 1.1'de bir MGGP model örneği sunulmuştur.

Tablo 1.1. Çok genli genetik programlama modeli örneği.

	Genotip	Ağırlık	Fenotip
Genler		0.47	$\cos(x_1) + x_2$
		-0.23	$(0.6 + x_3) * (x_1 - x_2)$
		0.054	$\left(\frac{x_2}{x_3} + 3.2\right) * (0.8 + x_1)$
Tüm Model	$y = 0.47 * (\cos(x_1) + x_2) - 0.23 * (0.6 + x_3) * (x_1 - x_2) + 0.054 * \left(\frac{x_2}{x_3} + 3.2\right) * (0.8x_1)$		

1.3. Diğer Otomatik Programlama Yöntemleri

Evrimsel programlama (EP), türlerin davranışlarının evrimsel süreçte birbirine bağlanmasını taklit eder [26]. EP, kontrollü bir stokastik arama kullanarak uygunluk değerini optimize eder ve aramayı birden fazla noktadan paralel olarak gerçekleştirir. Başlangıçta küresel optimum çözüm aranır. Arama uzayında parametrelerin rastgele bozulmasından dolayı bir süre sonra çözümlerin hedefe yakınsama hızı artar [27].

Zaman içerisinde, evrimsel programlamaya dayalı sembolik regresyon, kümeleme, sınıflandırma, görüntü işleme gibi birçok mühendislik problemlerini çözmeyi amaçlayan otomatik programlama yöntemleri geliştirilmiştir. Bunlardan biri, karınca koloni davranışlarının matematiksel modellerinden esinlenerek geliştirilen karınca

programlamadır [4]. Li ve Chen, yapay karıncaların iş birliğine dayanan ve her çözümü talimatlar dizisi olarak kullanan doğrusal zorunlu programlama için karınca kolonisi optimizasyonuna dayalı otomatik programlama yöntemi (Ant Colony Optimization For Linear Imperative Programming-ACOP) önermişlerdir [28]. Yöntemde keşif sırasında bir elementin uygunluk değeri sezgisel veya fenomen miktarı ile kontrol edildiğinden anlamsız işlemler elimine edilir. Deneylerde, ACOP 12 sembolik regresyon test problemi üzerinde Lineer GP (Linear Genetic Programming-LGP) ve Diferansiyel Gelişim ile Doğrusal Komut Programlama (Linear Imperative Programming with Differential Evolution-LDEP) ile kıyaslanabilir sonuçlar elde edilmiştir. Cano ve diğerleri, grafik işlemci ünitelerini (Graphic Processing Units-GPU) ve CUDA (Compute Unified Device Architecture) programlama modelini kullanarak, sınıflandırma problemi için çok amaçlı bir AP modelinin yeni bir paralelleştirme yaklaşımını sunmuşlardır [29]. Çoklu iş parçacığı kullanılarak paralelleştirilen program, her nesilde karıncaların yaratılması, çok amaçlı stratejisi ve kuralların seçilmesinde Niching yaklaşımı ile birlikte kullanılarak sınıflandırıcıyı oluşturur. Önerilen GPU modeli, büyük veri setlerini ve çoklu GPU cihazlarını verimli bir şekilde ölçeklendirmiştir. Kamali ve diğerleri, karmaşık diferansiyel denklemleri makul hesaplama süresiyle çözebilen Karınca Kolonisi Programlama (Ant Colony Programming-ACP) metodu geliştirmişlerdir [30]. Metot, uzaklık fonksiyonu yerine fenomen seviyesinin miktarıyla ilgili olasılık fonksiyonunu kullanarak GP'den daha iyi sonuçlar elde etmiştir. Kamali ve diğerleri, başka bir çalışmada bulanık modelleme problemlerindeki Matris Rikkati Diferansiyel Denklemleri'ni (Matrix Riccati Differential Equations-MRDE) çözmek için ACP metodu önermişlerdir [31]. Metot, terminal kriterini kontrol ederek çözüme yakın sonuçları tahmin etmiştir. Deneysel sonuçlar, metodun GP'ye nazaran göreceli hatalar açısından RK4 (Runge–Kutta 4th) yöntemine daha yakın olduğunu göstermiştir.

Kartezyen Karınca Programlama (Cartesian Ant Programming-CAP), Karınca Koloni Optimizasyonundaki (Ant Colony Optimization-ACO) fenomen yoğunluğuna dayanan rota seçim yöntemi yaklaşımını ve Kartezyen Genetik Programlamadaki (Cartesian GP-CGP) bireylerin grafik gösterimini birlikte benimseyen bir otomatik programlama yöntemidir [32-33]. Kushida ve diğerleri, sembolik regresyon problemlerini çözmek için karıncalar ve yaprak bitleri arasındaki simbiyotik ilişkiyi kullanan CAP önermişlerdir [34]. Önerilen yöntem, gereksiz düğümde fonksiyon değiştirerek ve her düğümdeki

fenomen miktarı ile birlikte yaprak bitlerinin salgıladığı tatlı özüt miktarını değerlendirerek çözüm alanını arama yeteneğini arttırmıştır. Hara ve diğerleri, CAP’de karıncaların arama performansını arttırmak için gereksiz fonksiyonları düğüm değişimleriyle faydalı fonksiyonlara dönüştüren uyarlamalı düğüm değişim metodu önermişlerdir [35]. Metot, sembolik regresyon probleminde CGP ve standart CAP’dan daha başarılı sonuçlar elde etmiştir. Kushida ve diğerleri, programda verilen düğümleri daha verimli kullanmak için Attach ve Multiple olarak adlandırdıkları düğüm bırakma mekanizmasına sahip bir CAP önermişlerdir [36]. Mekanizmanın ilk aşamasında belirli düğümler kullanılamaz olarak işaretlenir. Belirli bir adımdan sonra, kullanılamayan düğümler serbest bırakılır ve tüm düğümler kullanılabilir hale getirilir. Standart CAP’ye göre önerilen mekanizma ile etkili düğümlerin sayısı artırılmıştır. Kushida ve diğerleri, başka bir çalışmada standart CAP’ye, düğümlerin kullanım sıklığını uniform hale getirmek için internod mesafesini göz önünde bulunduran geçiş kuralı önermişlerdir [37]. Kuralla birlikte CAP’de sadece işlevsel sembollerin değil, aynı zamanda terminal sembollerin de ilgili düğümlere atanabilmeleri sağlanmıştır.

Musilek ve diğerleri, bağışıklık sisteminin ilkelerinden ilham alarak Bağışıklık Programlama (Immune Programming-IP) geliştirmişlerdir [38]. IP, klonal seçimin bağışıklık prensipleri ve düşük benzerliğe sahip antikorların yer değiştirmesi ile geliştirilmiştir. Popülasyondaki bireyler GP yöntemindeki gibi ağaçlarla temsil edilmiştir. Wang ve diğerleri, geleneksel IP yöntemini geliştirmek için RankIP olarak adlandırdıkları, keşif yaklaşımında sıralama (rank) fonksiyonunu geliştirmişlerdir [39]. Yöntemde, sınıflandırma problemlerini antikorların benzerlik değerlerinin belirli aralıkta iyi dağıldığından emin olmak için haritalama mekanizması ve test de sıralamayı öğrenmek için en iyi antikoru seçmeye odaklanan formüller tasarlanmıştır. Mansor ve diğerleri, yasak operasyon alanlarında elektrik dağıtım problemini çözmek için Log-Normal Evrimsel Programlama (Log-Normal Evolutionary Programming-LNEP) ve Yapay Bağışıklık Sistemi Programlama (Artificial Immune System Programming-AISP) hibridizasyonundan oluşan Bağışıklık Log-Normal Evrimsel Programlama (Immune Log-Normal Evolutionary Programming-ILNEP) metodunu önermişlerdir [40]. Metotta, yavru oluşturmak için mutasyona tabi tutulmadan önce, AIS’den klonlama işleminin avantajı ebeveyn nüfusunun en iyi bireylerini alma şansını artırmak için ana LNEP iyileştiricisine dâhil edilmiştir. Lau ve Musilek ozon ve klor dioksit dezenfektanları

tarafından bulaşıcılıktaki azalmayı IP yaklaşımı ile tahmin etmişlerdir [41]. Yaklaşım, küçük popülasyon büyüklüğü ile hedef çözümü geliştirme yeteneğine sahiptir. Deneysel sonuçlar, IP modellerinin performansının, sıcaklık düzeltmeli Chick-Watson modeli ve sinir ağı tabanlı model tarafından elde edilen sonuçlardan daha üstün olduğunu göstermiştir. Mabrouk ve diğerleri, Manyetik Rezonans Görüntüleme (Magnetic Resonance Imaging-MRI) ve Bilgisayarlı Topografi'den (Computed Topography-CT) tıbbi görüntülerin segmentasyonu için Bölge Yetiştiriciliği (Region Growing-RG) tekniği ile IP yöntemini birlikte kullanan hibrit sistem tasarlamışlardır [42]. Sistemden elde edilen modellerle çıkarılan matematiksel eşik fonksiyonları, RG tekniği ile farklı seviyelerde Radyo Frekansı (Radio Frequency-RF) ve gürültüye sahip çeşitli MRI görüntülerinin segmentasyonu için kullanarak başarılı sonuçlar elde edilmiştir. Xu ve Xuesong, IP yöntemine kaos optimizasyon mekanizması uyarlanarak kaos bağışıklık programı önermişlerdir [43]. Geliştirilmiş yöntem, kümeleme problemlerinde global çözüme yakınsama hızını arttırmıştır. Xu ve diğerleri, bir çift olarak en az anlamlı bit (The Pair-wise Least-Significant Bit- PLSB) eşleştirme şemasında en yakın çözümü bulmak için bir bağışıklık programlama stratejisi geliştirmişlerdir [44]. Strateji, permütasyon sıraları arasında optimal çözümü aramak için IP yöntemini ve LSB eşleşmesi için farklı sıraların performansını değerlendirmek üzere bir kademe puan sistemini kullanmıştır.

GP, günümüzde en çok kullanılan ve birçok mühendislik problemini başarıyla çözen algoritmadır. Araştırmacılar model çıkarımı, sınıflandırma, kümeleme, öznitelik seçimi problemlerinde GP kullanarak performans üzerindeki etkilerini gözlemleyerek yeni stratejiler önerdiler. Demirbaş ve diğerleri, işlevsel dikdörtgen plakaların termal stres analizi için GP ile maksimum ve minimum eş değer stres seviyeleri kullanarak denklemler elde etmişlerdir [45]. Yapılan çalışmalarda GP ile çıkarılan modeller, sonlu fark yöntemine nazaran hesaplama zamanını oldukça düşürmüştür. Liu ve diğerleri, farklı kanser türlerini sınıflandırmak için GP tabanlı topluluk sistemi üzerinde çalışmışlardır [46]. Karar ağaçları, minimum, maksimum ve ortalama operatörleri bu topluluğun çerçevesi olarak kullanılmıştır. Öznitelik, örnek ve temel sınıflandırma düzeyleri kullanarak aşırı öğrenmeden kaçınılmıştır. Patraik ve Bhuyan, heterojen trafik akışına sahip Hindistan'da hizmet seviyesi (The Level of Service-LoS) tanımlamak için GP tabanlı kümeleme analizi kullanmıştır [47]. Analiz sonuçları yol ağının daha iyi hizmet

kalitesi üretmesi için geometrik iyileştirmelere ihtiyacı olduğunu göstermektedir. Feng ve diğerleri, hata verileri kullanarak hiyerarşik kümeleme için GP tabanlı metot önermişlerdir [48]. Çalışmada çok kümeli problemler birden çok iki kümeli problem haline getirilerek GP uygulanmıştır. Chen ve diğerleri, her koşmada görülen en iyi öznelilikleri toplayarak kullandıkları permütasyon ölçütü ile özneliliklerin önemini değerlendiren GP tabanlı öznelilik seçim metodu önermişlerdir [49]. Permütasyon önceliği olan GP (GP with Permutation Importance-GPPI) olarak adlandırılan metodun, yüksek boyutlu sembolik regresyon problemlerinde, standart GP'ye göre genelleştirme performansı daha yüksektir.

GP, görüntü işleme alanında, özellikle görüntü tespitinde [50], sınıflandırmada [51], tanımlamada [52] ve görüntü segmentasyonunda [53] başarıyla uygulanmıştır. Choi ve diğerleri, üç aşamalı GP tabanlı bir sınıflandırıcıya dayalı yeni bir pulmoner nodül tespit sistemi önermişlerdir [50]. İlk aşamada akciğer hacmi, eşikleme ve 3D bağlantılı bileşen etiketleme tabanlı bir metot kullanılarak bölütlendirilmiştir. İkinci aşamada nodül adaylarının tespiti için bir dizi öznelilikler, en uygun çoklu eşik kullanılarak çıkarılmıştır. Üçüncü aşamada GP tabanlı bir sınıflandırıcı (GP-based Classifier-GPC), nodülleri ve nodülleri olmayanları sınıflandırmak için eğitilmiş ve kullanılmıştır. Deneysel sonuçlar, sınıflandırıcının hata pozitif oranını önemli ölçüde azalttığını ve yüksek duyarlılığı koruduğunu göstermiştir. Evans ve diğerleri, görüntü sınıflandırma problemleri için Eşevrimsel Sinir Ağlarının (Convolutional Neural Networks-CNNs) konvolüsyon ve havuzlama operatörlerini GP'ye uyarlayan yeni bir metot önermişlerdir [51]. Sonuçlar metodun, CNN'ye göre yorumlanabilir modelleri daha hızlı test ederek CNN ile rekabet edebilir olduğunu göstermiştir. Al-Sahaf ve diğerleri, sınıf başına sadece iki örnek kullanarak öznelilik vektörünü çıkaran GP tabanlı rotasyon duyarlı görüntü tanımlayıcı önermişlerdir [52]. Tanımlayıcı, az sayıda örnek kullandığından programın zaman ve bellek kullanımı açısından karmaşıklığını azaltır. Ayrıca istatistiksel bilgileri kullanarak rotasyon duyarlı sınıflandırma yapar. Deneysel sonuçlar, tanımlayıcının rotasyon değişimine karşı dirençli olduğunu göstermiştir. Vojodi ve diğerleri, değerlendirme doğruluğunu iyileştirmek için GP yöntemine dayalı bir kompozit değerlendirme ölçütü önermişlerdir [53]. Değerlendirme ölçütü; bölge içi benzerlik, bölgeler arası eşitsizlik ve entropi düzenine dayalı değerlendirme ölçüsüne bağlı olarak oluşturulmuştur. Büyük veri

setleri kullanılarak yapılan deneylerde önerilen ölçütün görüntü segmentasyonun doğruluğunun arttırdığını göstermiştir.

ABCP önerildikten sonra birkaç yıl içinde çeşitli problemleri çözmek için kullanılmıştır [8,54-58]. Arslan ve Öztürk sembolik regresyon, tahmin ve öznitelik seçimi problemleri için ABCP ile standart GP'yi karşılaştırmışlardır [54]. Deneysel sonuçlar; ABCP'nin GP'ye kıyasla güvenilir matematiksel modeller çıkarma, iyi tahmin yeteneği ve problemler için gerekli/alakalı öznitelikleri seçmede daha yüksek performans gösterdiğini ortaya koymuştur. Boudardara ve Görkemli, ABCP'yi, temel robotik yol planlama problemlerinden biri olan yapay karınca problemini çözmek için kullanmışlardır [55]. Santa Fe ve Los Altos Hills parkurlarında yürütülen deneylerde, ABCP'nin GP'den çok daha iyi çözümler ürettiği gözlenmiştir. Ayrıca ABCP'nin, bu tür sorunları etkin bir şekilde çözmek için kullanılabileceğini göstermiştir. Arslan ve Öztürk'ün diğer çalışmasında, Yapay Arı Koloni Programlama Tanımlayıcı (ABCP-descriptor) çok sınıflı desenleri sınıflandırmak için önerilmiştir [56]. Her bir desenin sınıflandırılması için eğitim aşamasında sadece rasgele seçilen iki tane örnek kullanılmıştır. Tanımlayıcının yaygın olarak kullanılan iki desen veri setinde standart Yerel İkili Örüntü (Local Binary Pattern-LBP)'ye ve GP-tanımlayıcı'ya (GP-descriptor) kıyasla iyi bir performansa sahip olduğu görülmüştür. Bu çalışma, tezin 6. bölümünde detaylı anlatılmıştır. Arslan ve Öztürk'ün yaptığı bir başka çalışmada ABCP ve GP'nin kanser veri setinde, sınıflandırma problemlerinde ve özniteliklerin seçilmesi probleminde performansı araştırılmıştır [57]. Yazarlar [57]'deki çalışmalarını geliştirerek dört farklı problemde ABCP tarafından çıkarılan modellerin sınıflandırma başarısını ortaya koymuştur [58]. Çalışmada en iyi modeller, veri setlerindeki toplam sınıf sayısına göre tanımlanan Hassasiyet Uygunluk Fonksiyonu (Sensitivity Fitness Function-SFF) kullanılarak elde edilmiştir.

Literatürde, makine öğrenmesi uygulamalarında evrimsel programlamaya dayalı farklı otomatik programlama yöntemleri geliştirilmiştir. Derhami ve Smith bulanık kural tabanlı sınıflandırma sistemlerinden optimal bulanık kuralları çıkaran tamsayı programlama yaklaşımını önerdiler [59]. Rekürsif uygulanan yaklaşımda, eski çözümlerin yeniden bulunmasını engellemek için önceki kuralları tabu kısıtlarına dönüştürerek yeni kurallar elde edilmesi, gereksiz kuralların kaldırılması, kural tabanının toplam doğruluğunun ve kapsamının maksimize edilmesi amaçlanmıştır. Yaklaşım, tahmin doğruluğu açısından geleneksel algoritmalara göre daha yüksek performans göstermiştir.

2. BÖLÜM

YAPAY ARI KOLONİ PROGRAMLAMA

2.1. Yapay Arı Koloni Programlama (ABCP)

Tezin bu bölümünde ABCP yöntemi Yapay Arı Kolonisi (Artificial Bee Colony-ABC) algoritmasını esas aldığı için öncelikle ABC algoritmasından bahsedilmiş ve ardından ABCP otomatik programlama yöntemi ayrıntılı olarak anlatılmıştır.

2.1.1. ABC Algoritması

ABC algoritması, bal arılarının yiyecek kaynağı arama davranışlarını simüle ederek çok boyutlu nümerik problemlerin en uygun çözümünü arayan, sürü zekasına dayalı optimizasyon algoritmasıdır. Algoritma, 2005 yılında Karaboğa tarafından geliştirilmiştir [60]. Literatürde, ABC algoritmasının çok sayıda farklı optimizasyon problemlerinin çözümü için kullanıldığı ve iyi bilinen algoritmalarla karşılaştırıldığında başarılı sonuçlar elde edildiği görülmüştür [61-64]. ABC algoritmasında bir yiyecek kaynağının pozisyonu, problem için mümkün olan bir çözüm; yiyecek kaynağının nektar miktarı, çözümün kalitesini ifade etmektedir. ABC yönteminde çözümler, karar değişkenleri için önerilen değerleri temsil eden sabit boyutlu diziler ile gösterilmektedir.

ABC algoritmasında görevli arılar, gözcü arılar ve kâşif arılar olmak üzere üç farklı göreve sahip arı bulunmaktadır. Görevli arılar yuvadan çıktıklarında hafızalarında belirli bir yiyecek kaynağı vardır ve yuvaya döndüklerinde faydalandıkları yiyecek kaynakları ile ilgili gözcü arılar ile bilgi paylaşımında bulunurlar. Gözcü arılar, görevli arılar tarafından paylaşılan bilgiyi değerlendirerek yiyecek kaynağının nektar miktarına göre gidecekleri yiyecek kaynağına karar verirler. Kâşif arılar ise diğer arı türlerinden farklı olarak, bilgi paylaşımında bulunmaksızın yeni yiyecek kaynağı ararlar. Yiyecek kaynaklarını bulduktan sonra çalışmalarına görevli arı olarak devam ederler.

ABC algoritmasının başlangıç aşamasında, kolonideki tüm arılar kâşif arı durumundadır. Her bir arı için rastgele çözümlerle Eşitlik 2.1 ile yiyecek kaynağı ataması yapılır.

$$x_{i,j} = l_j + rand(0,1) * (u_j - l_j) \quad (2.1)$$

Burada; $x_{i,j}$ yiyecek kaynağının (çözümün) j . boyutunda yer alan değeri, u_j j . boyut için üst sınırı, l_j j . boyut için alt sınırı ifade eder. Yiyecek kaynaklarının nektar miktarı, diğer bir deyişle kalitesi uygunluk fonksiyonuyla değerlendirilir. Görevli arılar hafızalarındaki yiyecek kaynağının daha iyisini başka bir yiyecek kaynağından da yararlanarak, Eşitlik 2.2 ile komşuluk araştırması yaparak bulmaya çalışırlar.

$$v_{i,j} = x_{i,j} + \phi_{i,j} * (x_{i,j} - x_{k,j}) \quad (2.2)$$

$k \neq i$ olmak üzere; v_i komşuluk araştırması sonucu elde edilen aday çözümü, x_k popülasyondan rastgele seçilmiş çözümü, j rastgele seçilmiş boyutu, $\phi_{i,j}$ ise $[-1,1]$ aralığında rastgele seçilen bir sayıdır. Uygunluk değeri en yüksek olan bireyi belirlemek için aday çözüm v_i ve mevcut çözüm x_i arasından açgözlü seleksiyon uygulanır. Uygunluk değeri $fit(x_i)$ 'nin belirlenmesinde Eşitlik 2.3'den yararlanılır.

$$fit(x_i) = \begin{cases} \frac{1}{1+f(x_i)} & \text{eğer } f(x_i) \geq 0 \\ 1 + abs(f(x_i)) & \text{eğer } f(x_i) < 0 \end{cases} \quad (2.3)$$

Eşitlikteki $f(x_i)$, x_i 'inci çözümün uygunluk fonksiyonu değerini ifade eder. Görevli arılar, yiyecek kaynaklarının kalitesi hakkındaki bilgileri gözcü arılarla paylaşırlar. Gözcü arılar yiyecek kaynaklarının kalitesine bağlı olarak, Eşitlik 2.4 ile elde edilen olasılıklarla, yiyecek kaynağı seçmekte ve görevli arılar aşamasında olduğu gibi Eşitlik 2.2'yi kullanarak seçilen kaynağı geliştirmektedir.

$$P_i = \frac{fit(x_i)}{\sum_{i=1}^{SN} fit(x_i)} \quad (2.4)$$

SN kolonide bulunan toplam yiyecek sayısını ifade eder. Kâşif arı aşamasında tüm yiyecek kaynaklarının tükenip tükenmediği kontrol edilmektedir. Yiyecek kaynaklarının tükenme durumunu “*limit*” olarak adlandırılan bir parametre kontrol eder. Her bir kaynak

için iyileştirme denemelerinin sayısı tutulur ve her bir çevrimde denemelerin sayısının limit parametre değerini aşıp aşmadığı kontrol edilir. Aştıysa yiyecek kaynağı tükenmiş kabul edilerek kaynak terk edilir ve o kaynağın görevli arısı kâşif arıya dönüşerek rasgele üretilen yeni bir kaynakla çalışmasına devam eder. ABC algoritmasının sözde kodu Şekil 2.1’de verilmektedir.

Başlangıç aşaması

Eşitlik 2.1 ile başlangıç çözümlerini (x_i) rassal olarak üret.

Çevrim sayacını başlat (çevrim=0)

REPEAT

Görevli arı aşaması

Her bir görevli arı için;

Eşitlik 2.2’yi kullanarak bir aday çözüm (v_i) üret.

Eşitlik 2.3 ile v_i ’nin uygunluk değerini belirle.

Uygunluk değerlerini dikkate alarak x_i ve v_i arasında açgözlü seleksiyon uygula.

Eğer seleksiyon sonucunda v_i seçilirse i . çözüme ait deneme sayacını sıfırla,

Seçilmezse deneme sayacının değerini bir artır.

Eşitlik 2.3 ve Eşitlik 2.4’ü kullanarak çözümler için olasılık değerlerini (P_i) hesapla.

Gözcü arı aşaması

Her bir gözcü arı için;

P_i ’ye bağlı olarak bir çözüm seç.

Eşitlik 2.2’yi kullanarak bir aday çözüm (v_i) üret.

Eşitlik 2.3 ile v_i ’nin uygunluk değerini belirle.

Uygunluk değerlerini dikkate alarak x_i ve v_i arasında açgözlü seleksiyon uygula.

Eğer seleksiyon sonucunda v_i seçilirse, çözüme ait deneme sayacını sıfırla,

Seçilmezse deneme sayacının değerini bir artır.

Şu ana kadar elde edilen en iyi çözümü hafızaya al.

Kâşif arı aşaması

Eğer tükenmiş bir yiyecek kaynağı (çözüm) mevcutsa kâşif arı için

Eşitlik 2.1 ile rassal olarak yeni bir çözüm üret ve üretilen çözümü tükenmiş çözümle değiştir.

İlgili çözümün deneme sayacını sıfırla.

çevrim=çevrim+1

UNTIL (çevrim = maksimum çevrim sayısı)

Şekil 2.1. ABC algoritması sözde kodu.

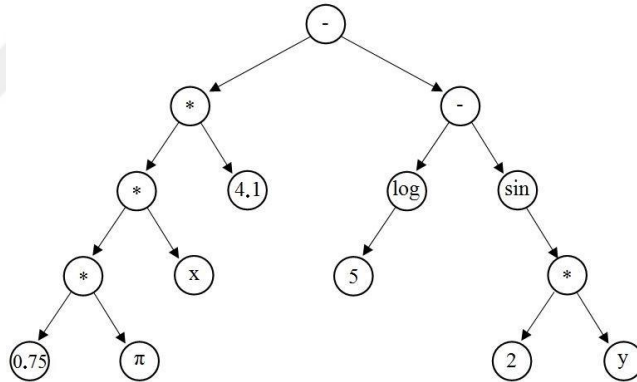
2.1.2. ABCP Algoritması

ABCP, ABC algoritmasına dayanan yüksek seviyeli otomatik programlama metodudur [7, 8]. ABCP algoritmasının adımları ABC algoritmasına benzerdir. Bu algoritmalar arasındaki en temel iki fark, çözüm yapısının temsili ve çözümlerin geliştirilmesini sağlayan bilgi paylaşım mekanizmasıdır. Çözümler; ABC’de sabit boyutlu dizilerle,

ABCP’de GP ile benzer şekilde terminal ve fonksiyonların kombinasyonundan oluşturulmuş parçalı ağaçlarla temsil edilir. Ağaçların en küçük birimi düğüm olarak adlandırılır. Düğümler, problemler için özel tanımlanan terminal kümesinden (x , y gibi değişkenler ve sabitler) ve fonksiyon kümesinden (aritmetik operatörler, mantıksal fonksiyonlar, matematiksel fonksiyonlar) seçilirler. Bu düğümlerin birleşimi ile çözümleri temsil eden ağaçlar oluşturulur. Şekil 2.2’de ABC (ikili kodlamalı) ve ABCP (ağaç yapılı) yaygın olarak kullanılan iki çözüm gösterimi verilmiştir. Şekildeki ABCP çözümünün matematiksel bağıntısı Eşitlik 2.5’de ifade edilmiştir. Bu gösterimlerde bağımsız değişkeni temsil etmek için “ x ” ve “ y ”, bağımlı değişkeni temsil etmek için “ $f(x,y)$ ” sembolleri kullanılmıştır.

1	0	0	0	1	0	1	0
---	---	---	---	---	---	---	---

(a)



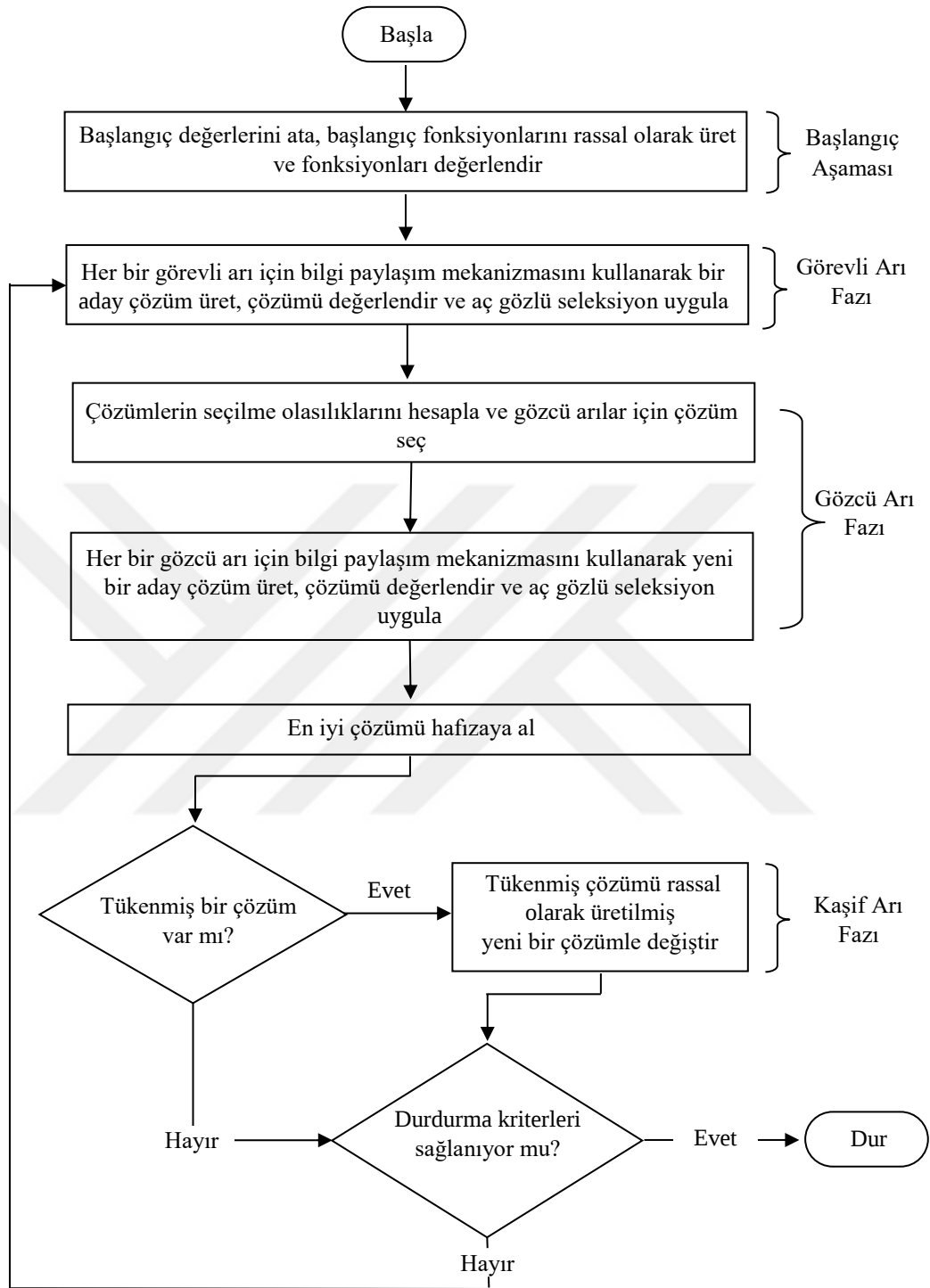
(b)

Şekil 2.2. ABC (a) ve ABCP (b)’de birey temsilleri.

$$f(x, y) = [(0.75 * \pi) * x * 4.1] - [\log(5) - \sin(2y)] \quad (2.5)$$

2.1.2.1. ABCP Akış Şeması

ABCP yöntemi için akış şeması Şekil 2.3’te verilmiştir. GP ile benzer şekilde ABCP’de başlangıç aşamasında çözümler “*full*”, “*grow*” ve “*ramped half and half*” metotlarıyla üretilebilir [3]. Bu metotlar alt bölümlerde ayrıntılı açıklanmıştır. Her bir çözüm ağacının kalitesi probleme özel olarak belirlenen uygunluk fonksiyonu dikkate alınarak belirlenir.



Şekil 2.3. ABCP algoritması akış diyagramı.

Maliyet fonksiyonlarının uygunluk değerlerine dönüştürülmesinde Eşitlik 2.3'den yararlanılır. Tez kapsamında kullanılan maliyet fonksiyonları negatif değerler almadığı için uygunluk değeri Eşitlik 2.6 ile belirlenebilir.

$$fit(x_i) = \frac{1}{1+fit(x_i)} \quad (2.6)$$

Gözcü arılar çözümlerin uygunluk değerlerine göre belirlenen olasılıkları kullanarak çözümleri seçerler. Bu tezde Eşitlik 2.7 kullanılarak yiyecek kaynaklarının seçilme olasılıkları hesaplanmıştır.

$$P_i = \frac{\alpha * fit(x_i)}{fit(x_{best})} + (1 - \alpha) \quad (2.7)$$

$fit(x_{best})$, en iyi uygunluk değerine sahip aday çözümü; α , 0 ile 1 arasında değer alan bir parametredir. Tez kapsamında $\alpha=0.9$ alınarak olasılıklar Eşitlik 2.8'deki gibi genelleştirilmiştir.

$$P_i = \frac{0.9 * fit_i}{fit_{best}} + 0.1 \quad (2.8)$$

ABCP'de çözümler ağaçlarla temsil edildiğinden ABC'de Eşitlik 2.2 ile komşuluk araştırması yapılarak bulunan aday çözüm üretme işlemi kullanılmamaktadır. Bunun yerine Koza'nın geliştirdiği çaprazlama operatörü [3], ABC'de komşuluk araştırma işlemine uyarlanarak bilgi paylaşım mekanizması olarak adlandırılmıştır [7]. Mekanizmanın aşamaları alt bölümde detaylandırılmıştır.

Kâşif arı aşamasında “*limit*” parametresi ile tüm yiyecek kaynaklarının tükenip tükenmediği kontrol edilmektedir. Limitin aşılması durumunda yiyecek kaynağı tükenmiş kabul edilerek kaynak terk edilir ve o kaynağın görevli arısı kâşif arıya dönüşür. Kâşif arı için grow metodu ile üretilen yeni çözüm kolonideki çözümlerden farklı olmalıdır. ABCP algoritmasının sözde kodu Şekil 2.4 ile gösterilmiştir.

Başlangıç aşaması

Kolonideki başlangıç çözümlerini *ramped half and half* metodu (x_i) rassal olarak üret. Çözümleri değerlendir.

En iyiyi hafızaya al.

Çevrim sayacını başlat (çevrim=0)

REPEAT**Görevli arı aşaması**

Her bir görevli arı için;

Bilgi paylaşım mekanizmasını kullanarak bir aday çözüm (v_i) üret.

Problemin maliyet fonksiyonu ile v_i 'nin uygunluk değerini belirle.

Uygunluk değerlerini dikkate alarak x_i ve v_i arasında açgözlü seleksiyon uygula.

Eğer seleksiyon sonucunda v_i seçilirse i . çözüme ait deneme sayacını sıfırla,

Seçilmezse deneme sayacının değerini bir artır.

Eşitlik 2.6 ve Eşitlik 2.7'yi kullanarak çözümler için olasılık değerlerini (P_i) hesapla.

Gözcü arı aşaması

Her bir gözcü arı için;

P_i 'ye bağlı olarak bir çözüm x_i seç.

Bilgi paylaşım mekanizmasını kullanarak bir aday çözüm (v_i) üret.

Problemin maliyet fonksiyonu ile v_i 'nin uygunluk değerini belirle.

Uygunluk değerlerini dikkate alarak x_i ve v_i arasında açgözlü seleksiyon uygula.

Eğer seleksiyon sonucunda seçilirse, çözüme ait deneme sayacını sıfırla,

Seçilmezse deneme sayacının değerini bir artır.

Şu ana kadar elde edilen en iyi çözümü hafızaya al.

Kâşif arı aşaması

Eğer tükenmiş bir yiyecek kaynağı (çözüm) mevcutsa kâşif arı için grow metodu rassal olarak yeni bir çözüm üret ve üretilen çözümü tükenmiş çözümle değiştir.

İlgili çözümün deneme sayacını sıfırla.

çevrim=çevrim+1

UNTIL (çevrim = maksimum çevrim sayısı)

Şekil 2.4. ABCP algoritmasının adımları.

2.1.2.2. ABCP Algoritmasının Bileşenleri

Standart ABCP algoritmasının çalışması için gerekli bileşenler alt bölümlerde açıklanmıştır.

2.1.2.2.1. Kolonideki Başlangıç Çözümlerinin Üretilmesi

ABCP'de algoritmanın ilk adımı kolonideki başlangıç çözümlerinin üretilmesidir. ABCP'de GP'de yaygın olarak kullanılan “*ramped half and half*” metodu ile çözümler üretilir. Ramped half and half metodu, full metodu ve grow metodunun birlikte kullanılması ile geliştirilmiştir [3]. Maksimum ağaç derinliği, bu metotların hepsi için bir

kontrol parametresidir. Takip eden kısımlarda koloninin üretiminde kullanılan fonksiyon, terminal kümesi ve üretim metotları açıklanacaktır.

2.1.2.2.1.1. Fonksiyon ve Terminal Kümesi

ABCP’de çözümler yapısı terminal ve fonksiyonların kombinasyonundan oluşturulmuş parçalı ağaçlarla temsil edilir. Ağaçların en küçük birimi düğüm olarak adlandırılır. Düğümler, problemler için özel tanımlanan terminal kümesinden (x, y gibi değişkenler ve sabitler) ve fonksiyon kümesinden (aritmetik operatörler, mantıksal fonksiyonlar, matematiksel fonksiyonlar) seçilirler. Bu düğümlerin birleşimi ile çözümleri temsil eden ağaçlar oluşturulur. Her bir çözüm ağacının kalitesi, her probleme özel olarak belirlenen maliyet fonksiyonu dikkate alınarak belirlenir. ABCP’de çözüm ağaçları üretilirken kullanılabilecek fonksiyon türleri:

- Aritmetik fonksiyonlar (-, +, *, ...)
- Matematiksel fonksiyonlar (sin, cos, sqrt, ...)
- Mantıksal fonksiyonlar (and, or, not, ...)
- Koşul ifadeleri (if-else)
- Döngü ifadeleri (for, while, ...) ve
- Probleme özel tanımlanan fonksiyonlardan oluşabilir.

Algoritmada kullanılan fonksiyon setinin kapalılık özelliği olmak zorundadır. Kapalılık özelliği gereği, bir fonksiyonun ürettiği sonucu, ağaçtaki diğer fonksiyonların kabul edebilmesi ve hata vermeden işleyebilmesi gerekmektedir. Örneğin, aritmetik operatörlerden bölme fonksiyonu, korumalı fonksiyon olarak tanımlanmalıdır. Çünkü payda sıfır olduğunda sıfıra bölünme hatası alınır ve hatadan dolayı algoritma çalışmasını durdurur. Bu nedenle paydanın sıfır olması durumunda işlemin sonucu olarak sıfır gibi bir tamsayı atanabilir. Ayrıca çözüm ağaçlarında algoritmada kod yazımında kullanılan veri tipinin aldığı değerden büyük değerler elde edilebilmektedir. Böyle bir durumla karşılaşıldığında Koza [3] ilgili ağaca değer atayarak algoritmanın çalışmasının devam edebileceği kanaatine varmıştır. Bu tez kapsamında problemlerin çözümünde söz konusu durumla karşılaşıldığında 1000000 sayısal değeri atanarak işlemlere devam edilmiştir.

2.1.2.2.1.2. Ağaçların Karmaşıklığı

ABCP’de ağacın kök düğümünün derinliği sıfır kabul edilir. Bir düğümün derinliği, kök düğümünden başlayarak düğüme ulaşmak için gereken minimum kenar sayısıdır. Bir ağacın derinliği, en derin düğümünün derinliği kabul edilir. Ağacın karmaşıklığı, ağacın derinliği ve o derinlikte bulunan düğüm sayısı ile orantılı olarak Eşitlik 2.9’daki gibi hesaplanır [3].

$$C = \sum_{k=0}^d k * n_k \quad (2.9)$$

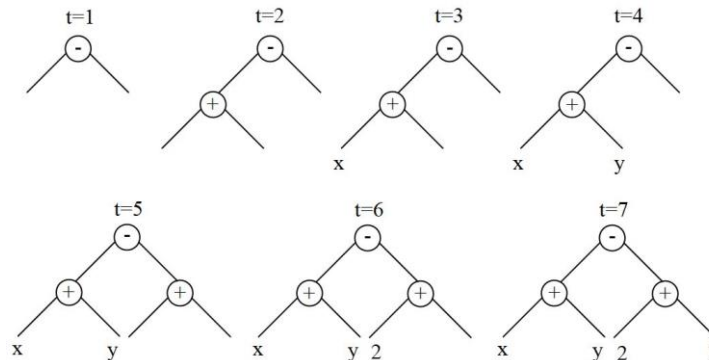
Burada C ağacın karmaşıklığı, d ağacın toplam derinliği, k derinlik, n_k k .ncı derinlikteki toplam düğüm sayısıdır. Örneğin Şekil 2.2(b)’deki bireyin ağaç derinliği 4 olup karmaşıklığı Eşitlik 2.10’daki gibi hesaplanır.

$$C = \sum_{k=0}^d k * n_k = 0 * 1 + 1 * 2 + 2 * 4 + 3 * 4 + 4 * 4 = 38 \quad (2.10)$$

2.1.2.2.1.3. Ramped Half and Half Metodu

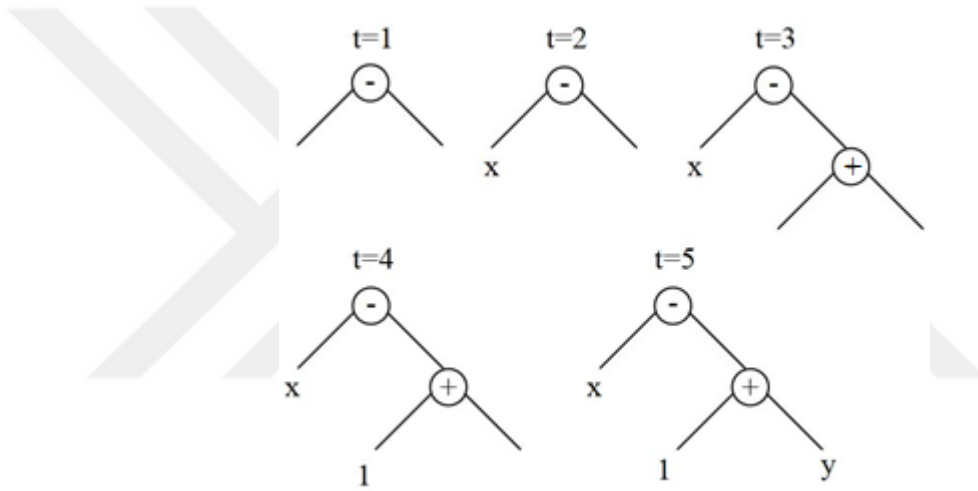
Kolonide başlangıçtaki çözümler diğer evrimsel algoritmalarla benzer şekilde maksimum ağaç derinliğini geçmeyecek şekilde “*ramped half and half*” metoduyla rastgele üretilir. Kâşif arı aşamasında tükenmiş kaynak yerine “*grow*” metodu ile üretilen yeni çözümün kolonideki diğer çözümlerden farklı olması gerekmektedir. Bu alt bölümde “*full*”, “*grow*” ve “*ramped half and half*” metodları açıklanacaktır.

“*Full*” metodunda, düğümler maksimum ağaç derinliğine ulaşmaya kadar fonksiyon kümesinden rastgele seçilir. Maksimum ağaç derinliğinde, düğümler terminal kümesinden seçilir. Şekil 2.5, ağaç derinliği 2 olan bir ağacın full metodu ile üretimini göstermektedir. Şekilde her bir adım $t=1,2,...,7$ ile gösterilmiştir.



Şekil 2.5. Full metodu kullanarak ağaç üretilmesi.

“Grow” metodu, daha farklı büyüklükte ve şekillerde ağaçların üretilmesine izin verir. Düğümler, maksimum ağaç derinliğine sahip olana kadar fonksiyon ve terminal kümesinden rastgele seçilir. Maksimum ağaç derinliğine ulaşıldığında full metodunda olduğu gibi sadece terminaller seçilir. Şekil 2.6, maksimum ağaç derinliği 2 olan bir ağacın grow metodu ile üretilme işlemini gösterir. Burada kök düğümün fonksiyon kümesinden seçilmesi (-) maksimum ağaç derinliğine ulaşarak fazla büyümesini engeller. Fonksiyon kümesinden seçilen diğer düğüm (+)’dır. Ağacın maksimum ağaç derinliğini aşmamasını sağlamak için maksimum ağaç derinliğinde terminallerin seçilmesi zorunludur.



Şekil 2.6. Grow metodu kullanarak ağaç üretilmesi.

Full ve grow metodları, farklı büyüklüklerde ve şekillerde ağaçlar üretmedikleri için Koza [3] “*ramped half and half*” olarak adlandırdığı metodu bu iki yöntemin kombinasyonu olarak önermiştir. Metot, başlangıç popülasyonunun yarısını “*full*”, yarısını “*grow*” metodunu kullanarak üretir. Farklı maksimum ağaç derinlikleri kullanılarak (bu nedenle ramped-artan terimi kullanılır) yapılan üretim, farklı büyüklük ve şekillerde ağaçların üretimini sağlar. Bu metodun uygulanması kolay olsa da, istatistiksel dağılımda ağaç büyüklüğü ve şekli gibi önemli parametrelerin kontrolünü zorlaştırır [10]. Grow metoduyla üretilen ağaçların büyüklüğü, terminal ve fonksiyon kümelerinin büyüklüklerine karşı oldukça hassastır. Örneğin program, fonksiyon kümesine nazaran daha büyük terminal kümesine sahipse grow metodu maksimum ağaç derinliğinden bağımsız olarak kısa ağaçlar üretir. Terminal kümesinin daha büyük olduğu

tam tersi durumda ise grow metodu, full metoduna benzer şekilde davranarak ağaçları üretir.

Kolonideki başlangıç çözümlerinin üretimi rastgele olmayabilir. Hedef çözümün özellikleri hakkında bilgi varsa, bu bilgiler kullanılarak kolonideki başlangıç çözümleri üretilir.

2.1.2.2.2. Uygunluk Fonksiyonu

Uygunluk fonksiyonu, kolonideki bireylerin problemi ne kadar iyi çözüp optimum çözüme ne kadar yaklaştığını belirleyen fonksiyondur. Amaç değeri iyi olan bir bireyin diğer nesle aktarılma olasılığı artar. ABCP’de uygunluk fonksiyonunun probleme uygun olarak seçilmesi, programın arama uzayında doğru yön tayini yapmasını sağlar. Aşağıda belirtilen uygunluk fonksiyonları tek başına ya da birleştirilmiş şekilde kullanılır [12].

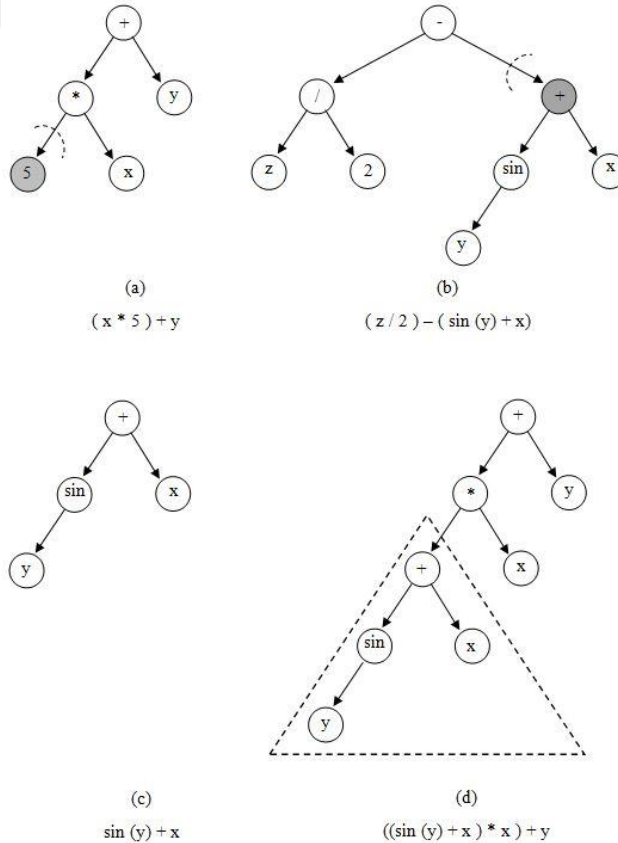
- Fayda Tabanlı (Benefit-based): Alınan faydayla doğru orantılı olan uygunluk fonksiyonlarıdır.
- Hata Tabanlı (Error-based): Toplam hataya göre ters orantılı olan uygunluk fonksiyonlarıdır.
- Amaç Tabanlı (Cost-based): Kullanılan kaynakla (ağacın yaprak sayısı, zaman, yer, amaç, vb.) ters orantılı olan uygunluk fonksiyonlarıdır.
- Cimrilik Tabanlı (Parsimony-based): Bireylerin sadeliği ile doğru orantılı olan uygunluk fonksiyonlarıdır.

2.1.2.2.3. Kontrol Parametreleri

Kontrol parametreleri, ABCP’nin çalışması sırasında kullanılan parametrelerdir. Parametrelerin doğru olarak belirlenmesi hedef çözüme/çözümlere yakınsama hızını artırır. Parametrelerin değerleri, daha önce yapılmış ve doğruluğu ispatlanmış deneylerden ya da deneme-yanılma yöntemi ile bulunabilir. Kontrol parametrelerine koloni büyüklüğü, maksimum ağaç derinliği, limit parametre değeri, maksimum iterasyon sayısı örnek olarak verilebilir.

2.1.2.2.4. Bilgi Paylaşım Mekanizması

ABCP’de aday çözüm (v_i) üretmek için GP’de Koza tarafından önerilen çaprazlama operatörü [3], ABC’deki komşuluk araştırmanın yerine bilgi paylaşım mekanizması adıyla önerilmiştir [7]. Mekanizmada aday çözüm (v_i) üretilirken, kolonide yer alan farklı bir kaynaktan alınan komşu düğüm çözümü x_k , ($i \neq k$) olmak üzere P_{ip} olasılığı göz önüne alınarak rastgele seçilir. Komşu çözümden rastgele seçilen düğüm x_k güncel çözümde paylaşılacak bilginin ne olacağı ve ne kadar paylaşılacağını belirler. Benzer olarak, x_i mevcut çözümünden P_{ip} olasılığı kullanılarak rastgele bir terminal veya fonksiyon düğümü seçilir. Aday çözüm v_i , güncel çözüm düğümü x_i ve komşu çözüm düğümü x_k ’nın düğümlerinin yer değiştirilmesiyle üretilir. Bilgi paylaşım mekanizması örneği Şekil 2.7’de verilmiştir. Şekil 2.7a ve 2.7b sırasıyla güncel çözümü ifade eden düğüm x_i ve ağaçtan alınan komşu düğüm çözümü x_k , Şekil 2.7c komşu çözümden alınan bilgiyi ve üretilen aday çözüm v_i , Şekil 2.7d’de verilmiştir. ABCP algoritmasında da ABC algoritmasında olduğu gibi, aday çözüm üretildikten sonra, güncel çözümü ifade eden düğüm x_i ve aday çözüm v_i arasında açgözlü (greedy) seçim süreci uygulanır.



Şekil 2.7. ABCP algoritması bilgi paylaşım mekanizması.

2.1.2.2.5. Durma Kriterleri

Durma kriterleri, algoritmanın hangi koşul ya da koşullarda durması gerektiğini belirleyen kriterlerdir. ABCP'nin durma kriterleri; iterasyon sayısı, önceden belirlenmiş uygunluk değerine istenilen düzeyde yakınsama olabilir. En çok kullanılan durma kriteri, iterasyon sayısıdır. Bu çalışmada durdurma kriteri olarak iterasyon sayısı kullanılmıştır.



3. BÖLÜM

TEST PROBLEMLERİNİN ÇÖZÜMÜNDE

ABCP’NİN UYGULANMASI

3.1. Giriş

Sembolik regresyon (Symbolic Regression-SR), bir sistemin giriş ve çıkış verileri arasında en uygun matematiksel bağıntının belirlenmesi işlemidir. Sembolik regresyon problemlerinde bağıntının oluşturulması, veriler arasında gerçekten bir ilişkinin var olup olmadığı analiz edilmeye çalışılır. Literatürde, sembolik regresyon problemini çözebilmek için otomatik programlama yöntemleri kullanılarak çok fazla araştırma yapılmıştır. Motta ve diğerleri, Gramer Tabanlı Genetik Programlama (Grammar Based Genetic Programming-GGP) ile Evrimsel Stratejileri (Evolutionary Strategies-ES) birlikte kullanan bir metot önermişlerdir [65]. Metotta, GGP tarafından çıkarılan modellerin sayısal katsayılarını geliştirmek için ES kullanılarak test problemlerinde en iyi medyan sonuçlar elde edilmiştir. Wu ve diğerleri, GPS uygulamalarında koordinatları dönüştüren regresyon fonksiyonlarını bulmak için genetik tabanlı bir yöntem sunmuşlardır [66]. Önerilen yöntemin elipsoidin tanımlanmasındaki kusurdan ve Küresel Konumlama Sistemi (Global Positioning System-GPS) alıcılarının güvenilirliğinden kaynaklanan sistematik hataları içermemesine rağmen, doğru kartezyen koordinatlar bağımsız kaynaklardan bilindiğinde, istatistiksel hataları etkili bir şekilde azalttığı görülmüştür. Smetka ve diğerleri, belirli bir anahtarla başlatılan simetrik bir şifreleme algoritması olan Veri Şifreleme Standardının (Data Encryption Standard-DES) davranışını modelleyen bir fonksiyonu GP ile elde etmeye çalışmışlardır [67]. Yapılan çalışmalar, GP’nin bilgisayar performansını arttırmasına rağmen, DES algoritmasının davranışının içyapısını ortaya çıkaramadığını göstermiştir. Kumar ve diğerleri, üç sediment taşınım problemi (vejetatif akış, toplam yatak yükü taşınımı ve sediment eşik tahmini için başlangıç kayması) için modeller geliştirmede Çok Genli GP kullanmışlardır [68].

Deneysel sonuçlar, GP'nin gürültülü verilere karşı dirençli modeller geliştirdiğini ve bu modellerin, modelleri eğitmek için kullanılmayan yeni veriler üzerinde yüksek tahmin doğruluğuna sahip olduğunu göstermiştir. Oliveira ve diğerleri, Sıralı Sembolik Regresyon (Sequential Symbolic Regression-SSR) tekniğini kanonik GP, Geometrik Anlamsal GP (GSGP) ve Genetik Özyinelemeli Sembolik Regresyon (GRSR) ile karşılaştırmışlardır [69]. Deney sonuçlarında SSR'nin, semantik geometrik operatörlerin GSGP tarafından kullanılmasından kaynaklanan üssel büyümeyi sınırlayan daha küçük çözümleri benzer hata oranları ile ürettiği gözlemlenmiştir. Haeria ve diğerleri, her bir GP ağacının her bir düğümü için değişkenlik, ortalama ve korelasyon katsayısı gibi istatistiksel bilgileri hesaplayan İstatistiksel GP (Statistical Genetic Programming-SGP) yöntemini önererek sembolik regresyon problemlerini çözmüşlerdir [70]. Çalışmada, SGP iyi yapılandırılmış bir ağaca daha yakın olan çözümler bulmak için arama sürecini engellemesi, çözümlerin doğruluğunu ve genelleme yeteneğini arttırdığı ve kod büyüme oranını azalttığı gösterilmiştir. Sotto ve Melo Doğrusal Genetik Programlamaya (Linear Genetic Programming-LGP), efmüt olarak adlandırdıkları makro ve mikro mutasyonlar ekleyerek yeni bir yöntem önermişlerdir [71]. Çalışmada yöntem, bir dizi sembolik regresyon seti üzerinde, geleneksel bir LGP algoritması ile karşılaştırılmıştır. Sonuçların analizinde, yöntemde kullanılan çifte turnuva ve orantılı turnuvanın etkili kodun yüzdesini arttırdığı ancak çözüm kalitesini iyileştirmediği gözlemlenmiştir. Barmpalexis ve diğerleri, standart ve çok popülasyonlu bir GP kombinasyonunu, kontrollü bir ilaç dağıtım sisteminin optimizasyonunda YSA ile karşılaştırmışlardır [72]. Deneysel sonuçlar, önerilen kombinasyondan çıkarılan modellerin karmaşık olsa da çeşitli bağımsız değişkenlerin önemine dair çıkarımlar ortaya koyarak sürecin altında yatan mekanizmaların anlaşılması için daha yüksek bir potansiyele sahip olduğunu göstermiştir. Arslan ve Öztürk yüksek boyutlu sembolik regresyon problemleri için ABCP'ye dayalı yeni bir öznetelik seçim yöntemi, Çok Kovanlı ABCP (MHABCP)'yi önermişlerdir [8]. MHABCP ve ABCP altı farklı yüksek boyutlu veri setinde test edilmiş ve yöntemlerin sonuçları Genetik Programlamanın sonuçlarıyla karşılaştırılmıştır. Deneysel sonuçlar, MHABCP'nin, standart GP ve ABCP ile karşılaştırıldığında eğitim setinde başarılı modelleri çıkarabildiğini ve test setini genelleştirme yeteneğine sahip olduğunu göstermiştir.

Otomatik programlama teknikleri, tahmin problemlerini çoğu evrimsel optimizasyon tekniklerine dayanan çalışmalarla çözmüştür. Seidy, tahmin doğruluğuna göre, Kütle Merkezi ile Parça Sürüsü Optimizasyonunu (Particle Swarm Optimization With Center Of Mass Technique-PSOCoM) kullanarak parçacık sürüsü optimizasyonuna dayalı modellerden daha başarılı sonuçlar veren yeni bir borsa tahmin modeli önermiştir [73]. Manjusha ve diğerleri, benzer semptomları olan potansiyel ölümcül dermatolojik hastalıkları teşhis etmek için Naive Bayes ve J48 algoritmalarını kullanmışlardır [74]. Her bir hastalığın nüksetme ihtimalinin tahmin edildiği bir arayüz geliştirmişlerdir. BuHamra ve diğerleri, Box-Jenkins (BJ) ve Yapay Sinir Ağları, Kuveyt'te aylık su tüketimini modellemek için birlikte kullanmışlardır [75]. Çalışmada, Yapay Sinir Ağındaki girdi katmanı değişkenlerinin BJ ile elde edilmesi ve ortalama hata dikkate alınması ile geleneksel yöntemlere göre daha başarılı sonuçlar elde edilmiştir. Martin ve diğerleri, iflas tahmin kurallarını üretmek için Karınca Koloni Optimizasyonu (ACO) tekniği kullanmışlardır [76]. Etamadi ve diğerleri, Çok Genli GP kullanarak çıkarılan T. pyriformis sulu toksisitesinin tahmin edilebilir sembolik Nicel Yapı Etkinlik İlişki (Quantitative Structure Activity Relationship-QSAR) modeli, aynı verilerdeki QSAR modelleri kadar başarılı olduğu bir uygulama ile göstermişlerdir [77]. Çalışmada GP ve çoklu ayrımcı analiz (Multiple Discriminant Analysis-MDA) teknikleri birlikte kullanılarak iflas tahmin modellemesi yapılmıştır. Sonuçlar, GP modelinin geleneksel MDA modelinden daha yüksek doğruluk oranlarına ulaştığını göstermiştir. Paul ve Iba mikro dizi verilerinin sınıflandırılması için çoğunluk oylu genetik programlama sınıflandırıcı (Majority Voting Genetic Programming Classifier-MVGPC) önermişlerdir [78]. Kanser veri setleri üzerinde yapılan deneylerde, çoğunluk oyuyla elde edilen doğruluğun, tek kuralların veya kuralların ortalama doğruluğundan daha iyi olduğu görülmüştür. Asima ve diğerleri, GP ve AdaBoost yöntemleri ile birleştirerek deprem tahmin sistemi (Earthquake Predictor System-EP-GPBoost) sunmuşlardır [79]. Yöntemle, sismik parametrelerin hesaplanması deprem tahmin performansını arttırmıştır. Arslan ve Öztürk sembolik regresyon, tahmin ve öznitelik seçimi problemleri için ABCP ile standart GP'yi karşılaştırmışlardır [54]. Deneysel sonuçlar; ABCP'nin GP'ye kıyasla güvenilir matematiksel modelleme çıkarmada, iyi tahmin yeteneği ve problemler için gerekli/alakalı öznitelikleri seçmede daha yüksek performans gösterdiğini ortaya koymuştur. Arslan ve Öztürk'ün diğer çalışmasında ABCP, lineer olmayan Cherkassky fonksiyon tahmin probleminde kullanılmış ve performansı GP yönteminin performansı

ile karşılaştırılmıştır [80]. Gorkemli ve diğerleri, ABCP zaman serileri tahmin problemi için Box-Jenkins veri setine uygulayarak performansı ABC ile eğitilmiş Yapay Sinir Ağıyla kıyaslamışlardır [81].

Son araştırmalarda, veri setlerindeki özniteliklerin önemini belirleme çabası, öznitelik seçim yöntemlerinin kullanılmasını gerektirmiştir. Bu yöntemler toplanan verilerdeki gürültülü ve gereksiz girişleri ortadan kaldırmak için kullanılır, böylece veri seti daha güvenilir bir şekilde ifade edilebilir ve aynı zamanda sınıflandırma yüksek başarı oranları elde eder. Farklı alanlarda öznitelik seçimi problemlerini çözmek için çeşitli optimizasyon yöntemleri uygulanmıştır. Lujan ve diğerleri, çok sınıflı problemler için ikinci dereceden fonksiyon optimizasyonuna dayanan, ikinci dereceden programlama adı verilen otomatik bir programlama tekniği önermişlerdir [82]. Çalışma, büyük veri setlerinde Maksimal İlişki (Maximal Relevance-MaxRel) ve Minimal Yedeklilik-Maksimum Alaka (Minimal-Redundancy-Maximum-Relevance-mRMR)'dan daha verimli bulunmuştur. Novaković ve diğerleri, istatistiksel ve entropi temelli öznitelik sıralama yöntemleri farklı veri setleriyle karşılaştırmışlardır [83]. Sınıflandırıcının doğruluğunun, sıralama endekslerinin seçiminden etkilendiği gösterilmiştir. Sikdar ve diğerleri, Diferansiyel Gelişim Tabanlı Varlık Tanıma (Named Entity Recognition-NER) adı verilen iki aşamalı otomatik programlama tekniği önermişlerdir [84]. İlk aşamada, öznitelik seçim problemlerinde Koşullu Rastgele Alan (Conditional Random Field-CRF) ve SVM sınıflandırıcıları kullanılmıştır. İkinci aşamada, F ölçeği puanına göre sınıflandırıcılar seçilerek ve diferansiyel gelişim temelli sınıflandırıcı toplama tekniği kullanılarak birleştirilmiştir. Viegas ve diğerleri, dengeli olmayan (skewed) veri setlerine, dirençli GP tabanlı öznitelik seçim yaklaşımı önermişlerdir [85]. Yaklaşım, farklı öznitelik seçim ölçütlerinin farklı öznitelik alanı ürettiği varsayımından yola çıkarak verilerde en ayırt edici öznitelik setlerini birleştirmiştir. Deneysel sonuçlar, yaklaşımın özellikle biyolojik verilerde öznitelik sayısından büyük oranda azalış ile etkinlik bakımından büyük kazançlar sağlamıştır. Mei ve diğerleri, atölye çizelgeleme problemi için GP ile kural geliştirmeye yönelik ilk pratik öznitelik seçim algoritması tasarlamışlardır [86]. Algoritmada iyi kurallar dizisi çıkarmak için niching tabanlı arama çerçevesi geliştirilerek taşıyıcı modelin karmaşıklığı azaltılmıştır. Guo ve diğerleri, sınıflandırıcının performansını geliştirmek ve öznitelik uzayının boyutunu azaltmak amacıyla GP tabanlı yaklaşım önermişlerdir [87]. İki yaygın epileptik

elektroensefalografi (EEG) tespit problemi üzerinde yapılan deneylerde önerilen yaklaşımın k-NN sınıflandırıcı performansını önemli ölçüde iyileştirdiği görülmüştür. Jia ve Verma, düşük enerjili sensör çıkarım uygulamaları için heterojen mikroişlemci tasarlamışlardır [88]. Tasarımda, donanım hızlandırmanın programlanabilir hesaplamalarının üstesinden gelmek için GP kullanan Öznitelik Çıkarma Hızlandırıcısı (feature extraction accelerator-FEA) önerildi. Tasarımın, yüksek seviyede hızlandırıcı programlanabilirliği sağlamış olduğu görülmüştür.

Bu bölümde Yapay Arı Koloni Programlama GP ile kıyaslanarak sembolik regresyon, tahmin ve öznitelik seçim problemlerinde performansları değerlendirilmiştir.

3.2. Deney Çalışmaları

Tezin bu bölümünde, karşılaştırmalı veri setleriyle üç farklı deney yürütülmüş ve GP ve ABCP'nin performans değerleri benzer parametreler kullanılarak karşılaştırılmış ve sonuçlar tartışılmıştır.

3.2.1. Deneyler

İlk deneyde, GP ve ABCP tarafından geliştirilen modellerin performansı sembolik regresyon probleminde değerlendirilmiştir. Eşitlik 3.1'de ifade edilen 4 girişli lineer olmayan Cherkassy fonksiyonuna [89] en yakın olabilecek fonksiyon tahmin edilmeye çalışılmıştır. GP ve ABCP kullanılarak, çıkış y 'ye çok yakın olan x_1, x_2, x_3 ve x_4 ile matematiksel bağıntıyı geliştirmek amaçlanmıştır. Fonksiyonun tahmini için eğitim aşamasında 400 adet nokta setinden (x_1, x_2, x_3, x_4, y) yararlanılmıştır. Ayrıca test aşamasında 100 adet nokta seti kullanılmıştır. Hem eğitim hem test için giriş nokta seti değerleri (x_1, x_2, x_3, x_4) $[-0.25, 0.25]$, çıkış nokta seti değerleri (y) $[0.5, 1.5]$ aralığında tanımlıdır.

$$y = \exp(2x_1 \sin(\pi x_4)) + \sin(x_2 x_3) \quad (3.1)$$

İkinci deneyde, pH veri setindeki çıkış değerleri tahmin edilmeye çalışılmıştır. pH veri seti, pH nötrleştirme işleminin bir simülasyonunda, dört girişin lineer olmayan bir fonksiyonu olan bir pH çıkış değeri elde edilerek oluşturulmuştur [90]. Veri setinde 700 adet kararlı nokta kümesi (x_1, x_2, x_3, x_4, y) ile eğitim seti, 299 adet kararlı nokta kümesi ile test veri seti oluşturulmuştur. Çıkış parametresi y ve lineer bağımsız giriş parametreleri

(x_1, x_2, x_3, x_4) olarak verilmiştir. Hem eğitim hem test için giriş nokta seti değerleri (x_1, x_2, x_3, x_4) $[0.5, 49]$, çıkış nokta seti değerleri (y) $[2.5, 11]$ aralığında tanımlıdır.

Üçüncü deneyde, GP ve ABCP'nin öznetelik seçim performansını incelemek için beton basınç dayanımı verileri [91] alınmıştır. Veri setindeki giriş değerleri: çimento (x_1), yüksek fırın cürufu (x_2), uçucu kül (x_3), su (x_4), süperplastikleştirici (x_5), kaba agregası (x_6), ince agregası (x_7), yaş (x_8)'dir. Çalışmada sekiz giriş parametresine ($x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8$) 50 adet gürültü giriş parametresi ($x_9, x_{10}, \dots, x_{57}, x_{58}$) eklenerek toplamda 1030 satır, 58 girişli ve 1 çıkış parametre değerine (y) sahip veri seti kullanılmıştır. Bu 1030 satır veri setinden 773 satır eğitim verisi, 257 satır test verisi olarak kullanılmıştır. Gürültü parametreleri eklenmeden giriş parametreleri $[0, 1135]$, çıkış nokta seti değerleri (y) $[2.5, 11]$ aralığında tanımlıdır. 50 adet gürültü parametresi $(-500, 500)$ aralığında rastgele değerlerden oluşturularak veri setine eklenir. Tüm deneylerde kullanılan girişlerin sayısı ve kayıt örnekleri sayısı Tablo 3.1'de gösterilmektedir.

Tablo 3.1. Test problemleri.

Veri Seti	Giriş Sayısı	Toplam Kayıt Örneği	Eğitim Kayıt Örneği	Test Kayıt Örneği	Gürültü
Cherkassy	4	500	400	100	-
pH	4	999	700	299	-
Beton Basınç Dayanımı	58(50+8)	1030	773	257	50 gürültü giriş parametresi [-500,500]

3.2.2. Uygunluk Fonksiyonu ve Parametreler

Yöntemler tarafından elde edilen modellerin performansı Eşitlik 3.2’de gösterilen Kare Ortalamalarının Karekökü (Root Mean Square Error-RMSE) fonksiyonu ile değerlendirilir.

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (y_{tahmin} - y_{gercek})^2}{n}} \quad (3.2)$$

Eşitlik 3.2’de ifade edilen y_{gercek} , veri setinden alınan gerçek y değeri; y_{tahmin} , elde edilen çözümün değişken setinin değerlerinin girilmesiyle elde edilen tahmini y değeri ve n veri büyüklüğüdür.

GP ve ABCP parametreleri Tablo 3.2’de özetlenmiştir. *add3* fonksiyonu üç değişkenin toplamıdır ($x_1 + x_2 + x_3$) ve *mult3* fonksiyonu üç değişkenin çarpımıdır ($x_1 * x_2 * x_3$). Bölme fonksiyonu korumalı fonksiyondur. Fonksiyonda bölen değeri sıfıra eşitse, sonuç birdir, aksi takdirde normal bölme gerçekleştirilir. *Ifbte* ve *iflte*, düğümlerin koşul durumunu belirtir. Eşitlik 3.3 ve Eşitlik 3.4 koşul fonksiyonlarının nasıl çalıştığını açıklar.

$$X = ifbte(A, B, C, D)$$

$$\text{Eğer } A \geq B \text{ ise } X = C \text{ değilse } X = D \quad (3.3)$$

$$X = iflte(A, B, C, D)$$

$$\text{Eğer } A < B \text{ ise } X = C \text{ değilse } X = D \quad (3.4)$$

Literatürdeki veri setlerinin büyüklüğü ile orantılı olarak popülasyon büyüklüğü yüksek alınmıştır. Deneyler değerlendirildiğinde, popülasyon/koloni büyüklüğü Cherkassy için 100, pH için 200 ve Beton Dayanımı veri seti için 300 olarak ayarlandığı durumlarda optimum sonuçların elde edildiği görülmüştür. Deneylerde, GP ve ABCP’nin durdurma kriteri hem GP hem de ABCP için maksimum iterasyon sayısı olarak alınmıştır.

Tablo 3.2. GP ve ABCP için parametreler.

Parametreler	Problemler					
	Cherkassy		pH		Beton Basınç Dayanımı	
	GP	ABCP	GP	ABCP	GP	ABCP
Popülasyon/ Koloni Büyüklüğü	100	100	200	200	300	300
İterasyon Sayısı	100	100	200	200	500	500
Maksimum Ağaç Derinliği	12	12	12	12	12	12
Turnuva Büyüklüğü	6	-	25	-	15	-
Mutasyon Oranı	0.14	-	0.14	-	0.14	-
Çaprazlama Oranı	0.84	-	0.84	-	0.84	-
Yeniden Eşleme Oranı	0.02	-	0.02	-	0.02	-
Fonksiyonlar	+,-, *, tan, sin, cos, square, max, min, exp, ifbte, iflte		+, -, *, tanh, add3, mult3		+,-,*,/, sin, cos, exp, rlog, add3,mult3	
Sabitler	[-10,10]		[-10,10]		[-10,10]	

3.3. Sonular

Bu alt blm, yapılan deneylerde, GP ve ABCP'nin sembolik regresyon, tahmin ve zellik seim kabiliyetlerini deęerlendirmektedir.

3.3.1. Simlasyon Sonuları

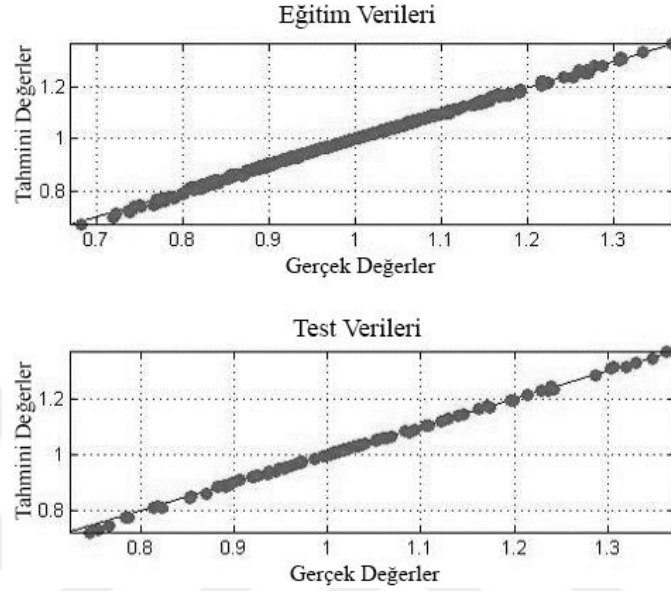
Deneyler, yntemler iin 30 defa baęımsız olarak yrtlmřtr. Problemler iin elde edilen sonular Tablo 3.3'te gsterilmektedir. Tablo 3.3'te veri setlerinde her lt iin en iyi deęerlere sahip yntem koyu renk ile belirtilmiřtir.

Tablo 3.3. Simlasyon sonuları.

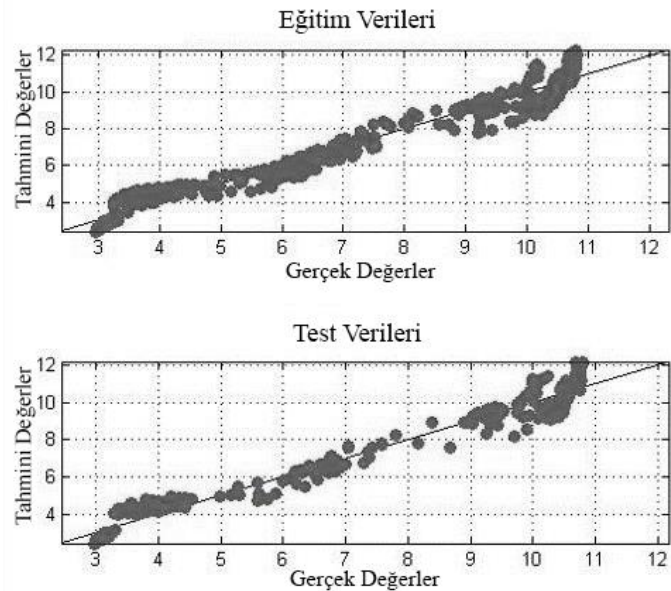
lt	Problemler					
	Cherkassy		pH		Beton Basın Dayanımı	
	GP	ABCP	GP	ABCP	GP	ABCP
Ortalama	0.07	0.03	0.92	0.77	11.64	10.50
Standart Sapma	0.03	0.02	0.14	0.07	2.38	1.51
Maksimum	0.02	0.01	0.60	0.59	8.83	8.26
Minumum	0.11	0.09	12.60	0.88	16.77	14.57

Aık kaynak sembolik regresyon aracı olan GPTIPS [92], bu alıřmada GP yntemi olarak sembolik regresyon problemi iin uyarlanmıř ve kullanılmıřtır. Tablo 3.2'de grldę zere tm deneylerde, ABCP'nin GP'den daha iyi eęitim performansına sahip olduęu grlmektedir. Tm kořulardaki en iyi uygunluk deęeri ABCP ile 0.0323 Cherkassy sembolik regresyon probleminde bulunmuřtur. pH veri seti iin ABCP'nin standart sapması % 10 ve GP'nin standart sapması %20 aralıęındadır. Ancak, Beton Basın Dayanımı veri setine eklenen grlt nedeniyle, tm ltlere dayalı sonular dięer veri setlerine gre daha kt olarak elde edilmiřtir.

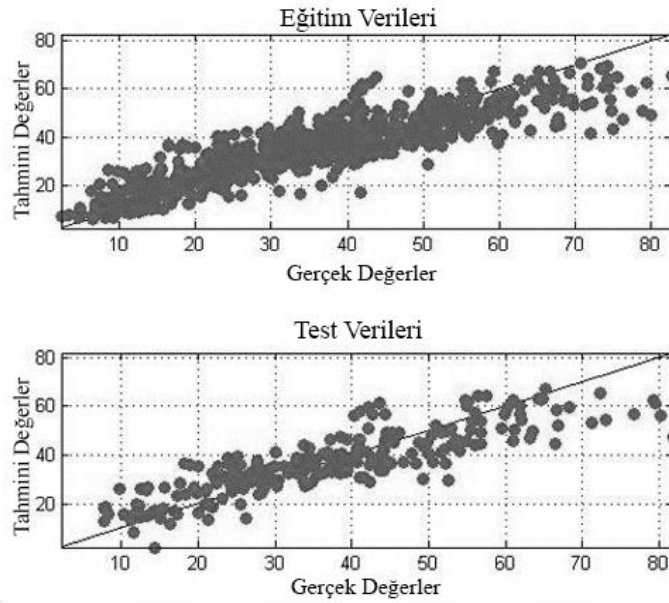
ABCP yönteminde, eğitim ve test verileri için tahmini ve gerçek veriler arasındaki ilişki, Cherkassy için Şekil 3.1’de, pH için Şekil 3.2’de ve Beton Basınç Dayanımı veri seti için Şekil 3.3’te ifade edilmiştir.



Şekil 3.1. Cherkassy fonksiyonunda ABCP’nin en iyi modelinin tahmini ve gerçek veri noktaları arasındaki ilişki.



Şekil 3.2. pH veri seti için ABCP’nin en iyi modelinin tahmini ve gerçek veri noktaları arasındaki ilişki.



Şekil 3.3. Beton Basınç Dayanımı veri seti için ABCP'nin en iyi modelinin tahmini ve gerçek veri noktaları arasındaki ilişki.

Şekil 3.1, Şekil 3.2 ve Şekil 3.3 tüm veri setleri için evrim grafiklerini göstermektedir. Şekillerden anlaşıldığı üzere, $y_{gerçek}$ ve y_{tahmin} değerlerinin birbirine yakınlığı aralarındaki hatanın artmasıyla azalmıştır.

3.3.2. ABCP Tarafından Üretilen Modellerin Değerlendirilmesi

Tüm deneylerde, ABCP'deki en iyi çözümlere sahip modeller Tablo 3.4'te gösterilmiştir. Tabloda, Cherkassy ve pH veri setlerindeki tüm girişlerin matematiksel modellerde kullanıldığı görülmüştür. Beton Basınç Dayanımı için geliştirilen modelde, 8 adet giriş parametresinden yüksek fırın cürufu, yaş, çimento, plastik giriş parametreleri ve eklenen 50 gürültü parametresinden sadece x_{17} gürültü parametresi yer almaktadır. Gürültü parametrelerinden sadece 1 tanesinin modelde bulunup 49 tanesinin bulunmaması, ABCP'nin gürültüyü başarılı bir şekilde ayıkladığının göstermektedir.

Tablo 3.4. ABCP’de en iyi modellerin eşitlikleri.

Veri Seti	En İyi Model	Giriş Sayısı
Cherkassy	$y = ((\exp(3x_4x_1) * (\exp(x_2x_3) + x_4x_1)) + x_4x_1) * \exp(x_4x_1))$	4
pH	$y = A + B + C$ $A = (\tanh(\tanh(x_4 + (x_3 * \tanh(x_2)))) + 2x_3) * \tanh(\tanh(x_2)) - x_2$ $B = x_2^2 * (\tanh((x_4 - (x_1 - 8.935 + (x_1 + x_3 + x_4) * \tanh(\tanh(x_2)))) * \tanh(x_2))) + x_2$ $C = (x_4 + (x_2 + \tanh(\tanh(\tanh(x_2)))) + \tanh(\tanh(2x_2 + x_4))) + ((x_2^2 * x_3) + x_2)) + \tanh(x_2)$ $+ \tanh(\tanh(x_2) - (x_1 - x_3)) - x_4$	4
Concrete	$y = \left \log\left(\frac{A}{B}\right) + (C) \right $ $A = \left (\tanh(\tanh(\sqrt{\sqrt{Yaş}}))) + \left(\frac{\sqrt{\log(\text{Çimento})}}{(-2.997)} + \text{Cüruf} \right) * Yaş * \frac{\tanh(\tanh(\log(\frac{Plastik}{-3.877})))}{\frac{\tanh(\tanh(x_{17}))}{(-3.877)}} \right $ $B = (\sqrt{\log(\text{Çimento})})^2 + (\text{Çimento} * (-3.877))^2 $ $C = (-3.877) + \left \left(\sqrt{\log(Yaş)^2} * \sqrt{\log(\text{Çimento})} - \left(\frac{\tanh(\tanh(Plastik))}{\tanh(\frac{\tanh(\text{Çimento})}{(-3.877)})} \right) \right) \right $ $- \tanh\left(\tanh\left(\log\left(\frac{Plastik}{(-3.877)}\right)\right)\right) / \tanh\left(\tanh\left(\tanh\left(\frac{\tanh(\text{Çimento})}{(-3.877)}\right)\right)\right)$	5

En iyi model ağaçlarının genel özellikleri hakkındaki bilgiler Tablo 3.5’te verilmiştir. Ağaçların karmaşıklığı Eşitlik 2.9 ile hesaplanmıştır. Tablo 3.5’te görüldüğü gibi, giriş parametrelerine eklenen gürültü parametreleri problemin zorluğunu arttırmıştır. Zorluğun arttırılması, çözüm ağaçlarını genişleterek karmaşıklığını arttırmıştır. Cherkassy fonksiyonu diğer problemlerden daha kolay olduğu için ABCP, modeller içinde en az karmaşıklığa sahip ağacı bu problemde çıkarmıştır.

Tablo 3.5. ABCP’de her veri seti için en iyi modellerin özellikleri.

Problem	ABCP		
	Ağacın Toplam Düğüm Sayısı	Ağaç Derinliği	Ağacın Karmaşıklığı
Cherkassy	25	7	118
pH	72	11	467
Beton Basıç Dayanımı	80	12	657

4. BÖLÜM

ÖZNİTELİK SEÇİMİ VE SINIFLANDIRMA PROBLEMLERİNİN YAPAY ARI KOLONİ PROGRAMLAMA İLE ÇÖZÜMÜ

4.1. Problem Tanımı

Öznitelik seçme yöntemleri, veri setlerinde ilgisiz ve gereksiz nitelikler çıkarılarak daha doğru sonuçlar elde etmek için modeller oluşturmaya izin verir. Bu yöntemler aracılığıyla model öngörüsü, y çıktı parametresi ile veri setinin x giriş parametreleri arasındaki fonksiyonel ilişki sağlanır. Model için gerekli olmayan özniteliğin kaldırılması, modelin boyutunu da azaltır, böylece alan karmaşıklığı ve hesaplama süresi azalır [93,94].

Öznitelik seçme yöntemleri, temel olarak, filtre yöntemleri, gömülü yöntemler ve sarmalayıcı yöntemler olmak üzere üç ana kategoride incelenmektedir [95]. Filtreleme yöntemlerine örnek olarak Bağımsız Bileşen Analizi (Independent Component Analysis-ICA) ve Temel Bileşen Analizi (Principal Component Analysis-PCA) ilk akla gelen örneklerdir. Filtreleme yöntemleri, öznitelikler arasındaki korelasyonlara (öznitelik alaka düzeyi) ve özniteliklerin sınıf etiket vektörleriyle ilişkilendirilmesine dayanan yöntemlerdir. Bu yöntemler bir seçim kriterinin yardımıyla öznitelikleri sınıflandırır. Özniteliklerin normal dağıldığı durumlarda PCA, normal olmayan dağılımın olduğu durumlarda ICA yöntemi kullanılabilir. Sarmalayıcı yöntemler, sınıflandırma doğruluğunun başarısını hesaba katar ve bir nesnenin modelde yer alıp almayacağına karar verir. Yüksek başarı değerine sahip modelleri elde etmek için, zaman kısıtlı problemlerde veri seti birçok kez eğitildiğinden ve test edildiğinden dolayı tercih edilmez. Model yapımının bir parçası olarak öznitelik seçimi gerçekleştiren gömülü yöntemler, en iyi bölünün belirlenmesine dayanır [96].

Son yıllarda öznitelik seçim metotları üzerine yapılan çalışmaların sayısı oldukça fazla rastlanmaktadır [97-101]. Zhang ve diğerleri, spam olmayan elektronik postaları spam olarak etiketleme hata oranını düşürmek için Mutasyon Operatörlü İkili Parçacık Sürü Optimizasyonu (Binary PSO with Mutation Operator-MBPSO) spam belirleme yöntemi önermişlerdir [97]. Yöntemin Genetik Algoritma, Parçacık Sürü Optimizasyonu, İkili Parçacık Sürü Optimizasyonu, Karınca Koloni Optimizasyonu gibi birçok sezgisel yöntemle göre daha başarılı olduğu gözlemlenmiştir. Sikora ve Piramuthu, Hausdorff uzaklık ölçüsünü kullanan öznitelik seçim problemi için genetik algoritma tabanlı veri madenciliği tekniği önermişlerdir [98]. Gerçek veri madenciliği problemlerinde, tahmin doğruluğu ve hesaplama verimliliği konusunda tekniğin oldukça başarılı olduğu görülmüştür.

Öznitelik seçimi probleminin çözümüne yönelik GP tabanlı yöntemler geliştirilmiştir. Aslam ve diğerleri, diyabet verilerinin sınıflandırılması için GP tabanlı yöntem sunmuşlardır [99]. Çalışmada seçilen özniteliklerin kombinasyonu yapılarak yeni öznitelikler üretilmiştir. Genetik Programlama tabanlı Karşılaştırmalı Partner Seçimi (Genetic Programming-Comparative Partner Selection-GP-CPS) olarak adlandırılan yöntem sekiz giriş boyutunu tek boyuta indirgemıştır. Harvey ve diğerleri, probleme özgü terminal ve fonsiyonları kolayca dâhil edebilen AutoFeat olarak adlandırdıkları GP tabanlı otomatik öznitelik tasarımı önermişlerdir [100]. AutoFeat, büyük sayısal dizin verilerden anlamlı/ilgili öznitelikleri seçerek karar verme süreçlerini oluşturmada oldukça başarılı olduğunu göstermiştir. Muharram çalışmasında karar ağacı sınıflandırma tekniklerinin karmaşık, gerçek veriler üzerindeki performansını artırmak amacıyla öznitelik alanını yeniden yapılandıran GP tabanlı yaklaşım önermiştir [101]. Yaklaşım, lineer olmayan öznitelik kombinasyonlarından yeni öznitelikleri çıkarır ve uygunluk fonksiyonu olarak karar ağacı bölme kriterlerini kullanır. Deneysel sonuçlara göre, önerilen yaklaşım başarılı tahmin doğruluğuna sahiptir.

Sınıflandırma, veriler hakkında bilgi edinmeyi ve verileri izlemeyi kolaylaştırdığı için yapılan çalışmaların sayısı oldukça fazladır [102-108]. Bagui ve Hemasinha, iki büyük meme kanseri veri setini lineer, kuadratik, k en yakın komşu (k Nearest Neighbor- k-NN) gibi birçok yöntemle sınıflandırmaya çalışmışlardır [102]. Çalışmada, 9 değişkenden oluşan WBC (Wisconsin Breast Cancer) ve 30 değişkenden oluşan WDBC (Wisconsin Diagnostics Breast Cancer) veri setlerini sırasıyla 6 ve 7 değişkene indirgemişlerdir.

Awuley ve Ross, öznitelik seçimi ve sınıflandırma için GP tabanlı Öznitelik Seçiminin Yaş Tabanlı Popülasyon Yapısı (Feature Selection Age Layered Population Structure-FSALPS)'nı yeni bir algoritma olarak sunmuşlardır [103]. Algoritma, farklı özniteliklere sahip iki sınıf içeren dört farklı veri seti üzerinden kanonik GP, Yaş Tabanlı Popülasyon Genetik Programlama (The Age Layered Population Structure Genetic Programming-ALPSGP) ile karşılaştırılmıştır. Sonuçlar değerlendirildiğinde yeni geliştirilen algoritmanın öznitelik azaltmada oldukça başarılı olduğu görülmüş ve bu alanda kullanılabilir bir yöntem olduğu kanısına varılmıştır. Liu ve Gui, mikro dizi veri setlerinin analizinde küçük örneklem büyüklüğü kullanarak analiz etmişlerdir [104]. Çalışmada GP ile bireyler alt-topluluk (sub-ensemble-SE) olarak adlandırılan ağaç kümesinden oluşturulur. SE'lerde yüksek çeşitliliği sağlamak için açgözlü algoritma tasarlanmıştır. Sonuçlar, GP tabanlı yöntemin, karmaşık çok sınıflı mikro dizi veri setlerinin sınıflandırılmasında kullanılabileceğini göstermiştir. Chaudhari ve diğerleri, çok sınıflı problemler için her ağacın belirli bir sınıfın örüntüsünü tanıyan, diğer örüntüleri reddeden GP metodunu önermişlerdir [105]. Çalışmada mutasyon işleminin yıkıcılığını azaltan yeni bir mutasyon operatörü kullanılmıştır. Performansı kötü olan ağaçlara genetik işlemlere katılma ve performansını iyileştirme şansı tanımlamışlardır. Mausa ve Grbac, bir türün çoklu alt popülasyonlarında kullanılan bir ek göçmenlik operatörü ve iki fenotipik olarak farklı eş-evrimsel türde kullanılan kolonizasyon operatörü önermişlerdir [106]. Yazarlar geliştirdikleri Eş-evrimsel GP (Co-Evolutionary Multi-Population Genetic Programming)'de, farklı seçim stratejileri kullanarak dengesiz veri setlerinde yazılım hata tahmini problemlerini test etmişlerdir. Abdulhamid ve diğerleri, sınıflandırma için üç farklı GP döngü yapısı önermişlerdir [107]. Döngü yapıları klasik yöntemler ve standart GP ile kıyaslanmıştır. Döngü yapıları aramaya ekstra karmaşıklık getirirse de Poker problemi gibi zor problemlerde sınıflandırma başarısının yüksek olduğu görülmüştür. Yan ve Zeng sınıflandırma kurallarını, PSO ve GP yöntemlerini hibritleştirerek çıkarmışlardır [108]. Deneysel sonuçlar, en iyi kuralların, PSO ve GP'nin art arda kullanıldığı iki kez optimizasyondan sonra elde edildiğini göstermiştir.

Tezin bu kısmında yapılan çalışmada aşağıdaki işlemlerin yapılması hedeflenmiştir:

- GP ve ABCP tarafından çıkarılan modellerin performanslarının sınıflandırma doğruluğu, karmaşıklık gibi parametrelerle karşılaştırılması,
- ABCP yönteminin gerçekten ilgili/bağılantılı özellikleri seçip seçemeyeceği,
- Aşırı öğrenme durumu olup/olmadığını belirlemek için yöntemlerin eğitim performanslarının değerlendirilmesi.

4.2. Deney Çalışmaları

Bu alt bölümde, yürütülen deneylerle GP ve ABCP'nin öznelik seçimi ve sınıflandırma kabiliyetleri gösterilmeye çalışılmıştır.

4.2.1. Veri Setleri

Deneyler dört gerçek veri seti üzerinde yürütülmüştür. Tüm veri setleri UCI'dan alınmıştır [109]. Deneylerde kullanılan ilk veri seti Wisconsin meme kanseri veri setidir. Veri seti 569 hastaya ait tümörün iyi ya da kötü huylu olduğunu belirleyen 30 adet giriş parametresinden oluşmaktadır. Veri seti incelendiğinde yaklaşık olarak tümörlerin %60'ı iyi huylu, geri kalanının kötü huylu olduğu gözlemlenmiştir. Çalışmanın amacı, veri setinde sunulan 30 adet giriş parametresi kullanılarak tümörün iyi ya da kötü huylu olduğunu teşhis edebilmektir. Veri setinde kötü huylu tümör 1, iyi huylu tümör 0 ile tanımlanmıştır. Giriş seti 10 adet şüphelenilen tümöre ait parametre içerir. Bu giriş parametreleri; yarıçap, doku, çevre uzunluğu, alan, akıcılık, yoğunluk, konkavlık, konkavlık noktaları, simetri ve fraktal olarak verilmiştir. Her bir kayıt için ortalama, standart hata ve en kötü hata değeri bulunmaktadır. Böylece 30 adet giriş parametresi elde edilmiş olunur.

Son çalışmalarda, makine öğrenmesi algoritmaları kanser sınıflandırması probleminin çözümü için kullanılmıştır. Bagui ve diğerleri, WDBC veri seti üzerinde lineer, kuadratik, lojistik ve k-NN kuralları uygulayarak sınıflandırma yapmışlardır [102]. Salama ve diğerleri, J48 karar ağaçları, Çok Katmanlı Algılayıcı (Multi-Layer Perception-MLP), Naive Bayes (NB), Sıralı Minimum Optimizasyon (sequential minimal optimization-SMO), Uzaklık Tabanlı k-En Yakın Komşu (Instance Based For K-Nearest Neighbor-IBK) sınıflandırıcıları ile WDBC veri setini sınıflandırmışlardır [110]. Kathija ve diğerleri, WDBC'yi sınıflandırmak için Karar Destek Makineleri (Support Vector

Machines-SVM) ve Naive Bayes sınıflandırıcılarını kullanmışlardır [111]. Deney sonuçları değerlendirildiğinde, Naive Bayes modelinin daha yüksek sınıflandırma başarısı olduğu görülmüştür.

Deneylerde kullanılan ikinci veri seti dermatoloji veri setidir. Bu veri seti, 33'ü doğrusal değerli ve bir tanesi nominal olan 34 nitelik içermektedir. Veri setinde, hastalar önce klinik olarak 12 öznitelikle değerlendirilmiştir. Daha sonra, 22 histopatolojik özniteliklerin değerlendirilmesi için cilt örnekleri alınmıştır. Veri setinde histopatolojik özniteliklerin değerleri, numunelerin bir mikroskop altında analizi ile belirlenmiştir.

Günümüzde dermatoloji hastalıklarını teşhis için birden çok araştırma mevcuttur [74,112-116]. Güvenir ve diğerleri, Bayesian tekniğini öznitelik seçimi olarak kullanmışlardır [112]. Doğruluk, hassaslık, özgüllük gibi birçok ölçüte göre çalışma değerlendirilerek 34 öznitelige sahip dermatoloji veri seti için 15 öznitelikten oluşturulan model sınıflandırmada oldukça başarılı sonuçlar elde edilmiştir. Pappa ve diğerleri, sınıflandırmada öznitelik seçimi için C4.5 olarak adlandırılan çok nesneli genetik algoritma yaklaşımı önermişlerdir [116]. Dermatoloji veri setinin de içinde bulunduğu 6 farklı veri seti üzerinde yapılan deneylerde minimize edilen hata oranı ve C4.5 karar ağacı ile ilgili başarılı sonuçlar elde edilmiştir.

Deneylerde kullanılan üçüncü veri seti, İtalya'nın aynı bölgesindeki üç farklı çeşit şarabın kimyasal analizinin sonuçları olan Şarap veri setidir. Analiz, üç şarap çeşidinin her birinde bulunan 13 unsurun miktarlarına göre belirlenmiştir. Bu veri seti, toplamda 13 nitelik içermektedir.

Zhong ve Fukushima yaptığı çalışmada, düzgün olmayan Newton metodunun düzenlenmiş bir yaklaşımını önererek içerisinde şarap veri setinin de bulunduğu altı farklı veri seti üzerinde standart v-K-SVCR yöntemi ile kıyaslamasını yapmışlardır [117]. Deney sonuçları önerilen yaklaşımın standart metoda göre daha hızlı ve büyük veri setlerinde daha başarılı olduğunu göstermiştir. Dy ve Brodley etiketlenmemiş veriler üzerinde dağılım ayrılabilirliği ve maksimum olasılık kriterlerine göre FSSEM (Feature Subset Selection Using (EM)) clustering, Beklenti Maksimizasyonu (Expectation-Maximization-EM) kümeleme kullanarak öznitelik alt karakter seçimi yapmışlardır [118]. Şarap veri setinin de kullanıldığı bu çalışmada, k değerinin verildiği FSSEM- k

yöntemi başarılı bulunmuştur. Fisher ve Poland, spektral kümeleme yöntemleri için blok yapılı yakınlık matrisi önererek standart sınıflandırma metotları ile karşılaştırmasını yapmışlardır [119]. İçerisinde şarap veri setinin de bulunduğu 10 farklı veri setinde yapılan çalışmada genel olarak başarılı sonuçlar elde edilmiştir.

Son olarak bu tez çalışmasında at kolik veri setinden faydalanılmıştır. At kolik, atların çeşitli patolojik değerlerine bağlı olarak kolik rahatsızlığının bulunup bulunmadığı veya bilinmediği durumlar derlenerek oluşturulmuş veri setidir. Bu veri seti, toplamda 26 nitelik içermektedir. Nock ve diğerleri, klasik en yakın komşuluk ilişkilerinin puanlarını hesaplayan Simetrik En Yakın Komşu Kuralını (Symmetric Nearest Neighbor-sNN) önermişlerdir [120]. At kolik veri seti ile birlikte toplam yirmi dokuz veri seti üzerinde, yeni önerdikleri metotla klasik en yakın komşu metotlarını kıyaslamışlardır. Deneysel sonuçlar, simetrik en yakın komşu kuralının klasik metotlara göre oldukça başarılı olduğunu göstermiştir.

4.2.2.Eğitim ve Test Setleri

Çalışmada her veri seti, GP ve ABCP’de çıkarılan modellerin öznitelik seçimi ve sınıflandırma performansını araştırmak için bir eğitim setine ve test setine bölünmüştür. Dört veri setinin öznitelik sayısı, eğitim örnekleri ve test örnekleri Tablo 4.1’de gösterilmektedir. Tüm veri setlerinden rastgele seçilen örneklerin setlerinin yaklaşık olarak % 70’i örnek eğitim ve diğer kalan %30’u ise örnek test setini oluşturmuştur. Her koşmada, eğitim ve test kayıtları, rastgele veri setleri örneklerinden seçilerek yeniden oluşturulmuştur.

Tablo 4.1. Veri setlerinin özellikleri.

Veri Seti	Öznitelik Sayısı	Toplam Kayıt Sayısı	Eğitim Kayıt Sayısı	Test Kayıt Sayısı	Sınıf Sayısı
WDBC	30	569	427	142	2
Dermatoloji	34	366	274	92	6
Şarap	13	178	133	45	3
At Kolik	26	364	273	91	3

4.2.3. Uygunluk Fonksiyonu Tanımlanması

Bu çalışmada kullanılan veri setlerinin giriş değerleri reel olduğu için çözümlerden elde edilen sonuçlar teorik olarak $[-\infty, \infty]$ aralığında bulunmaktadır. Dolayısıyla sonuç değerlerinin ayrık sınıf değerlerini tanımlayabilmesi için (sınıf 0, sınıf 1 gibi) öncelikle, daha önce tanımlanan ve içerisinde toplam sınıf sayısını içeren bir aralığa çekilmesi gerekmektedir. Bunun için tanımlanan fonksiyon Eşitlik 4.1’de verilmiştir.

$$(N_c - 1) \left(\frac{1}{(1 + \exp(-g_o))} \right) \quad (4.1)$$

Eşitlikte tanımlanan N_c çıkış sınıflarının sayısını, g_o ise o anki çözümün sonucunu ifade eder. Örneğin, 4 sınıflı bir problem için Eşitlik 4.1’in çıkışı $[0-3]$ aralığında olur. Bulunan reel değerler en yakınındaki tamsayı değerine yuvarlanarak çözümün sınıf değeri ‘0’, ‘1’, ‘2’, ‘3’ olarak tahmin edilmiş olunur. Bu çalışmada benzer olarak 2 sınıflı bir problemde bahsedildiğinden (WDBC veri seti, iyi huylu sınıf 0, kötü huylu sınıf 1) Eşitlik 4.1 sonucunda elde edilen çıkış $[0,1]$ aralığında olup, çıkışlar en yakın tam sayı değerine yuvarlanarak sınıf değeri ‘0’ veya ‘1’ olarak elde edilmiş olunur.

Eşitlik 4.1’den sağlanan değerlerle doğru sınıflandırma yapılabilmesi için uygun bir uygunluk fonksiyonu gerekmektedir. Bu çalışmada uygunluk fonksiyonu, doğru tahmin edilen sınıf sayılarının, veri setindeki toplam sınıf sayılarının oranlarının ağırlıklı toplamı ile bulunmuştur. Örneğin, iki sınıfa sahip veri seti için 0 sınıf değeri için doğru tahmin edilen 0 sınıf sayısının veri setindeki toplam 0 sınıf sayısına oranıyla, 1 sınıf değeri için doğru tahmin edilen 1 sınıf sayısının veri setindeki toplam 1 sınıf sayısına oranının ağırlıklı toplamından elde edilmektedir. İki sınıfa sahip problemler için Hassas Uygunluk Fonksiyonu (sensitivity fitness function-SFF) olarak tanımlanan bu fonksiyon Eşitlik 4.2’de verilmiştir [121].

$$SFF = w \frac{n_c(i,0)}{n_a(i,0)} + (1 - w) \frac{n_c(i,1)}{n_a(i,1)} \quad (4.2)$$

Eşitlik 4.2’de $k = 0$ veya $k = 1$ olmak üzere $n_c(i, k)$, i . çözüm için k sınıfından veri setindeki k sınıfı ile kıyaslandığında doğru tahmin edilen durumların sayısını; $n_a(i, k)$, veri setindeki k sınıfında bulunan tüm kayıtların sayısını; w ise sınıflarının önem durumuna göre $[0,1]$ aralığında tanımlanan reel bir sayıyı ifade eder. Çalışmada

Tablo 4.2’de görüldüğü gibi, w ağırlık değeri, çalışmada her bir sınıf eşit öneme sahip olacak şekilde, her bir veri setinin sınıf sayısına oranla tanımlanmıştır. *maxx* fonksiyonu vektörün maksimum değerini, *minx* fonksiyonu vektörün minimum değerini, *ifbte* ve *iflt* düğümlerin koşul durumunu belirtir. Eşitlik 3.3 ve Eşitlik 3.4’te koşul fonksiyonlarının nasıl çalıştığı açıklanmıştır.

4.3. Sonuçlar

Bu alt bölüm, yapılan deneylerde, GP ve ABCP’nin simülasyon sonuçlarını ve çıkarılan modellerin analizini içermektedir.

4.3.1. Simülasyon Sonuçları

Problemler için elde edilen sonuçlar her veri seti için ortalama, en iyi, en kötü değerler açısından Tablo 4.3’te gösterilmektedir. Tablo 4.3, ABCP’nin her veri seti için eğitimde, tüm performans ölçütlerinde, GP’den daha başarılı olduğunu göstermektedir. Tablo 4.3’te veri setlerinde test başarı yüzdesi en yüksek değere sahip yöntem koyu renk ile belirtilmiştir. Tabloda WDBC’de, yöntemler arasında ortalama eğitim başarı yüzdeleri arasında %1 fark, test başarı yüzdeleri arasında ortalama %0,05 fark olduğu görülmüştür. Her iki yöntem için dermatoloji veri setinde elde edilen sonuçlar, at kolik veri seti için elde edilen sonuçlardan sonra, elde edilen en az başarılı sonuçlardır. Dermatoloji veri setinde, GP ile çıkarılan başarı yüzdeleri arasında oldukça yüksek fark bulunduğu için standart sapma yüksek çıkmıştır. Şarap veri setinde, her iki yöntemde test başarı yüzdesi %100 bulunmasına rağmen uygunluk fonksiyonu 0,98’de kalmıştır. At kolik veri setinde elde edilen sonuçlar diğer veri setlerine nazaran düşüktür. GP ve ABCP’nin, klasik yöntemle yapılan çalışmayla kıyaslandığında [120], benzer sınıflandırma sonuçları elde ettiği görülmüştür.

4.3.2. Çıkarılan Modellerin Analizi

Tüm deneylerdeki GP ve ABCP’de en iyi çözümlere sahip modellerin eşitlikleri Tablo 4.4’te gösterilmiştir. Tabloda görüldüğü üzere eşitliklerde en fazla öznelilik dermatoloji ve at kolik veri setlerinde kullanılmıştır. WDBC ve şarap veri setlerinde 4 girişle yöntemler en iyi modellerini üretmişlerdir.

Tablo 4.3. Simülasyon sonuçları.

		GP				ABCP			
Veri Seti	Ölçüt	$SFF_{eğitim}$	Eğitim Başarı Yüzdesi	SFF_{test}	Test Başarı Yüzdesi	$SFF_{eğitim}$	Eğitim Başarı Yüzdesi	SFF_{test}	Test Başarı Yüzdesi
WDBC	Ortalama	0.91	92.33	0.9	91.01	0.92	93.27	0.9	91.48
	Standart Sapma	0.02	2.56	0.03	3.8	0.02	2.01	0.03	3.07
	En İyi	0.94	95.32	0.94	95.77	0.95	96.25	0.96	97.89
	En Kötü	0.86	86.42	0.81	77.46	0.87	87.82	0.84	84.51
Dermatoloji	Ortalama	0.81	81.96	0.77	78.66	0.89	92.27	0.85	89.17
	Standart Sapma	0.1	15	0.11	13.96	0.02	1.93	0.05	4.4
	En İyi	0.92	95.26	0.94	96.74	0.93	97.08	0.97	98.91
	En Kötü	0.6	48.54	0.48	46.74	0.84	89.42	0.77	80.43
Şarap	Ortalama	0.88	88.7	0.85	84.9	0.92	93.43	0.88	88.22
	Standart Sapma	0.06	5.94	0.07	7.59	0.02	2.59	0.05	6.83
	En İyi	0.95	98.5	0.98	100	0.97	98.5	0.98	100
	En Kötü	0.76	76.69	0.71	73.33	0.88	88.72	0.78	73.33
At Kolik	Ortalama	0.62	58.81	0.49	50.4	0.67	62.52	0.54	54.76
	Standart Sapma	0.06	5.42	0.09	8.35	0.03	3.53	0.07	4.92
	En İyi	0.71	67.4	0.65	71.43	0.73	69.96	0.65	61.54
	En Kötü	0.51	47.99	0.3	38.46	0.62	56.78	0.36	45.05

Tablo 4.4. GP ve ABCP ile elde edilen en iyi modeller.

Veri Seti	Yöntem	En İyi Model Eşitlikleri	Eşitlikteki Öznitelik Sayısı	Ortak Öznitelik/Öznitelikler
WDBC	ABCP	$\cos(x_8 * x_{22}) + \min(x_5, \cos(x_5 * x_{21}))$	4	-
	GP	$\cos(\text{iflte}(\tan(0.7604 * x_{27} * x_{28} * x_{24}), \tan(x_{14}), \tan(0.7390), \cos(x_{17})))$	5	
Dermatoloji	ABCP	$(\text{iflte}(\tan(x_{15}, x_6, (((\tan(\text{iflte}(x_{22}, x_{22}, (\text{iflte}(x_{33}, x_{31}, x_{31}, \text{iflte}(x_{23}, x_{31}, \tan(-2.161), x_{15}))) - x_{31}, x_{15}))) - x_{22}) + (\tan(x_5 * x_{34}))), (2x_7 - x_{20})) - x_{22}))$	11	$x_5, x_{15}, x_{22}, x_{23}, x_{31}$
	GP	$\text{iflte}(x_{13}, x_{20}, \text{iflte}(x_{13}, x_{27}, \cos(\exp(\text{iflte}(x_{23}, x_5, \cos(x_{15}), x_{31}), x_{23}), (x_{13} - x_{23})))) + (x_{31} - (x_{22} * \exp(x_{22})))$	8	
Şarap	ABCP	$\text{iflte}(-2.452, (\exp(x_{12}) - x_{10}), 7.5519, ((-2.452 * x_{10}) * \cos(x_7)) * \min(\cos(\cos(x_{10} - x_7)) - x_{13})))$	4	x_7
	GP	$\min(\exp(\sin(x_1), \min(\exp(\cos(x_7)), \cos(\text{iflte}(x_9, x_7, \text{iflte}(\cos(\exp(\sin(x_1))), \text{iflte}(\cos(x_7), x_8, x_7, x_8)))x_7,)x_{11})), \tan(\text{iflte}(x_{11}, \sin(x_1), x_8, x_7)))$	4	
At Kolik	ABCP	$(\text{iflte}(\text{square}(\cos(x_{10} + x_{24})), \sin(x_{22}), \min(x_8, x_{12}) - 4.1591, \sin(\text{iflte}(x_{19}, \exp(\exp(\text{iflte}(x_{23}, x_{22}, x_{22}, \sin(x_{22}))))), \tan(x_{23}) - x_{22}, x_{22}))))$	7	$x_8, x_{10}, x_{22}, x_{23}$
	GP	$\min(x_6, x_{21}) - \text{square}(\min(\text{iflte}(x_{22}, x_{10}, \text{iflte}(\text{square}(x_1), x_{10}), \tan(-6.653), x_1), x_{26}), \min(x_{14}, x_{21})) - \text{square}(\min(\text{iflte}(\text{square}(x_{14}), \text{square}(x_8), \tan(x_{23}), -6.655), (x_{23} - x_1))))$	8	

Tablo 4.4'te görüldüğü üzere her iki yöntem de en fazla dermatoloji veri setinde, en az şarap veri setinde ortak öznitelik çıkarmıştır. WDBC veri seti için her iki yöntem de farklı özniteliklerle modelini oluşturmuştur.

Her iki yöntemin de birkaç özniteliğe sahip başarılı modelleri çıkardığı görülmektedir. En iyi çözüm ağaçlarının sahip olduğu toplam düğüm sayısı, ağaç derinliği, ağaç karmaşıklığı hakkında detaylı bilgi her veri seti için Tablo 4.5'te verilmiştir. Ağaçların karmaşıklığı Eşitlik 2.9 ile hesaplanmıştır.

Tablo 4.5. Elde edilen en iyi çözüm ağaçlarının özellikleri.

Problem	GP			ABCP		
	Toplam Düğüm Sayısı	Ağaç Derinliği	Ağaç Karmaşıklığı	Toplam Düğüm Sayısı	Ağaç Derinliği	Ağaç Karmaşıklığı
WDBC	16	7	67	11	5	36
Dermatoloji	25	8	107	37	12	249
Şarap	32	9	177	21	7	81
At Kolik	34	9	197	33	9	163

Tablo 4.5'te veri setlerinde daha az karmaşıklığa sahip yöntem koyu renk ile belirtilmiştir. Tabloda görüldüğü gibi, dermatoloji veri seti hariç, tüm veri setlerinde ABCP düğüm, ağaç derinliği ve karmaşıklığı daha az olan eşitlikleri oluşturduğu görülmektedir. Tüm koşmalarda, her iki yöntemde elde edilen en iyi çözümlerin matematiksel modelleri değerlendirilmiş ve modellerde yöntemlerin kullandıkları ortalama öznitelik sayısı, standart sapma ve eşitliklerdeki öznitelikler Tablo 4.6'da verilmiştir. Tabloda 30 koşma için toplamda özniteliklerin karşılaşma sıklığı da verilmiştir. Örneğin; WDBC için x_{28} özniteliği koşmalarda, ABCP tarafından çıkarılan en iyi modellerde 15 kez eşitliklerde yer almıştır. Her iki yöntemde de eşit veya yakın ortalama değerler ve standart sapmalar ile eşitlikler oluşturulduğu görülmüştür. Tabloda görüldüğü üzere, tüm koşmalarda her iki yöntemin de eşitliklerinde ortak kullandığı öznitelikler veri setlerinde sırasıyla; WDBC'de 3 (x_7, x_8, x_{28}), dermatolojide 7 ($x_7, x_{14}, x_{15}, x_{22}, x_{27}, x_{31}, x_{33}$), şarapta 4 ($x_7, x_{10}, x_{11}, x_{12}$), at kolikte 8 ($x_1, x_8, x_{10}, x_{19}, x_{21}, x_{22}, x_{23}, x_{26}$) özniteliktir.

Tablo 4.6. En iyi sonuçlarda öne çıkan öznitelikler.

Veri Seti	Yöntem	Ortalama	Standart Sapma	En Çok Kullanılan Öznitelikler ve Kullanılma Sıklıkları	Sık Kullanılan Öznitelik Sayısı	Ortak Öznitelik Sayısı
WDBC	ABCP	4,13	1,33	$x_5(7), x_7(12), x_8(11), x_{28}(15)$	4	3
	GP	3,13	1,36	$x_7(12), x_8(12), x_{27}(11), x_{28}(8)$	4	
Dermatoloji	ABCP	7,20	1,90	$x_6(9), x_7(12), x_{14}(23), x_{15}(29), x_{22}(25), x_{27}(10), x_{31}(30), x_{33}(15)$	8	7
	GP	6,23	1,74	$x_5(10), x_7(13), x_{14}(15), x_{15}(9), x_{20}(10), x_{22}(15), x_{27}(9), x_{30}(8), x_{31}(23), x_{33}(8)$	10	
Şarap	ABCP	4,07	1,18	$x_7(30), x_{10}(19), x_{11}(26), x_{12}(17)$	4	4
	GP	3,17	1,58	$x_7(29), x_{10}(14), x_{11}(12), x_{12}(12)$	4	
At Kolik	ABCP	6,93	1,41	$x_1(21), x_8(13), x_{10}(7), x_{15}(7), x_{19}(25), x_{21}(13), x_{22}(23), x_{23}(27), x_{26}(13)$	9	8
	GP	5,97	2,36	$x_1(15), x_2(7), x_7(11), x_8(13), x_{10}(9), x_{14}(9), x_{19}(14), x_{21}(14), x_{22}(12), x_{23}(15), x_{26}(14)$	11	

5. BÖLÜM

YÜKSEK BOYUTLU SEMBOLİK REGRESYON PROBLEMLERİNDE ÖZNİTELİK SEÇİMİ İÇİN MHABCP YAKLAŞIMI

5.1. Problem Tanımı

Öznitelikler süreçlerin bireysel ölçülmüş özellikleridir. Veriler toplanırken alakasız ve gereksiz özniteliklerin bulunması veri setlerinin boyutunun artmasına neden olmaktadır. Bununla birlikte çok fazla özniteliğin birlikte elde edilmesi sırasında gürültünün de toplanması olası olduğundan, öğrenme algoritmalarının anlamlı model çıkarımı zorlaşarak aşırı öğrenme riski ve hesaplama maliyeti artmaktadır [122].

Öznitelik Seçimi (Değişken Eleme), yüksek boyutlu veri setlerinin işlenmesinde, gürültüyü azaltarak boyutsallık zorluğunun etkisini gidermeye ve tahmin performansını geliştirmeye yardımcı olan bir makine öğrenmesi tekniğidir [123]. Öznitelik seçimi ile ilgili çok çalışma bulunmaktadır [103,124-126]. Bu çalışmalarda genelde öznitelik seçimi ile birlikte sınıflandırma yapılmıştır. ABCP [7] ve GP [3] ile yapılan çalışmalarda yüksek boyutlu verilerde öznitelik seçim problemi konusuna nadiren değinilmiştir. GP, model oluştururken öznitelikleri eleme kabiliyetine sahiptir. Ancak yüksek boyutlu veriler için öznitelikleri seçme kabiliyetinin yeterince güçlü olmadığı ifade edilmektedir [54].

Genelleme, öğrenme algoritmalarından eğitim sonucunda elde edilen başarılı modellerin test verilerinde düşük hataya sahip olma kabiliyetidir. Eğer bir öğrenme algoritması, iyi bir genelleme kabiliyeti varsa, öğrenilmiş verilerdeki başarısını test edilmeyen verilerde de gösterebilir. Aşırı uydurma, öğrenilen modellerin eğitim verisini ezberleyerek düşük eğitim hatasına sahipken, test verileri üzerinde yüksek hata elde ederek başarısız olma

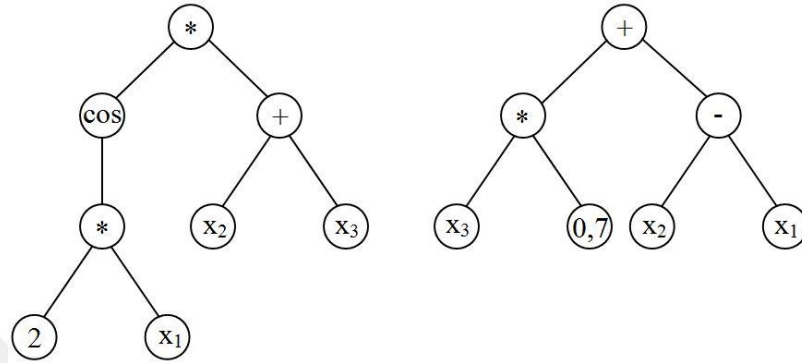
durumudur. Bu nedenle öğrenme algoritmalarının genelleme kabiliyeti ne kadar yüksekse, aşırı uydurma diğer bir deyişle eğitim verisini ezberleme riski de o kadar azdır. Yüksek boyutlu verilerden model çıkarımında öğrenme algoritmalarının eğitimde zorlanması genelleme kabiliyetini azaltır. Öğrenme algoritmalarında, çıkarılan modellerin makul test hatalarına sahip olması hedeflenir.

Verilerdeki öznitelik sayısının artmasıyla (diğer bir deyişle verinin boyutsallığı) genelleme kabiliyetinin geliştirilmesinin önemi gittikçe artmaktadır. Genelleme ile ilgili yapılan çalışmalar [127,128] olmasına rağmen, yüksek seviyeli programlama yöntemlerinin genelleme kabiliyetinin değerlendirilmesi ile ilgili az sayıda çalışma bulunmaktadır. Yapılan çalışmalar, yapay sinir ağlarının eğitiminde ABC algoritması ve türevlerinin etkileyici genelleme kabiliyetine sahip olduğunu göstermiştir [129,130]. Kushchu genellemeye ilişkin araştırmaları inceleyerek, genelleme kabiliyetini arttırmak için GP tabanlı bir yöntem önermiştir [131]. Gonçaves ve diğerleri, GSGP’de mutasyon adımını deterministik ve optimal olarak adapte eden geometrik semantik mutasyonun iki varyantını önermişlerdir [132]. Bu varyantlarla ilgili yapılan deneylerde GSGP’nin eğitim verilerini öğrenmede ve genelleme kabiliyetinde oldukça başarılı oldukları görülmüştür. Uy ve diğerleri, semantik tabanlı çaprazlama kullanarak GP’nin genelleme yeteneği üzerindeki belirgin bir pozitif etkisi olduğu gerçek sembolik regresyon problemleri ile test ederek görmüşlerdir. ABCP yeni geliştirilen bir programlama yöntemi olduğundan genelleme kabiliyeti ile ilgili çalışma bulunmamaktadır. Bu bölümde, yüksek boyutlu sembolik regresyon probleminde otomatik programlama yöntemlerinin öznitelik seçimi ile birlikte genelleme kabiliyeti performansı değerlendirilmiştir.

5.2. MHABCP (Multi Hive Artificial Bee Colony Programming-MHABCP)

Bu bölümde, ABCP algoritmasının verimliliğini arttırmak ve yüksek boyutlu sembolik regresyon problemlerinde güvenilir model çıkarımı yapabilmek için ABCP tabanlı Çok Kovanlı ABCP (Multi Hive Artificial Bee Colony Programming-MHABCP) önerilmiştir. Standart ABCP algoritmasında, kolonideki her bir çözüm ağacı, çıkış değişkenleri y ’yi tahmin etmek için kullanılır. Burada her bir sembolik model tek bir ağaçla temsil edilir. Ancak MHABCP’de her model (kolonideki her bir çözüm), her bir ağacın bir kovan olarak düşünülebileceği bir dizi ABCP ağacından elde edilen çıktıların ağırlıklı lineer bir birleşimidir. Şekil 5.1, x_1, x_2 ve x_3 bağımsız değişkenler kullanılarak oluşturulmuş birçok

kovanlı model örneğini tanımlar. Şekil 5.1'in modelinin matematiksel eşitliği Eşitlik 5.1'deki gibidir. Burada y bağımlı çıktı değişkenini temsil eder. Eşitlik elde edilirken modeldeki her bir kovan bir w ağırlık katsayısı ile çarpılır ve w_0 (bias) eklenir.



Şekil 5.1. MHABCP sembolik model örneği.

$$y = w_0 + w_1(\cos(2x_1)) * (x_2 + x_3)) + w_2(0.7x_3 + (x_2 - x_1)) \quad (5.1)$$

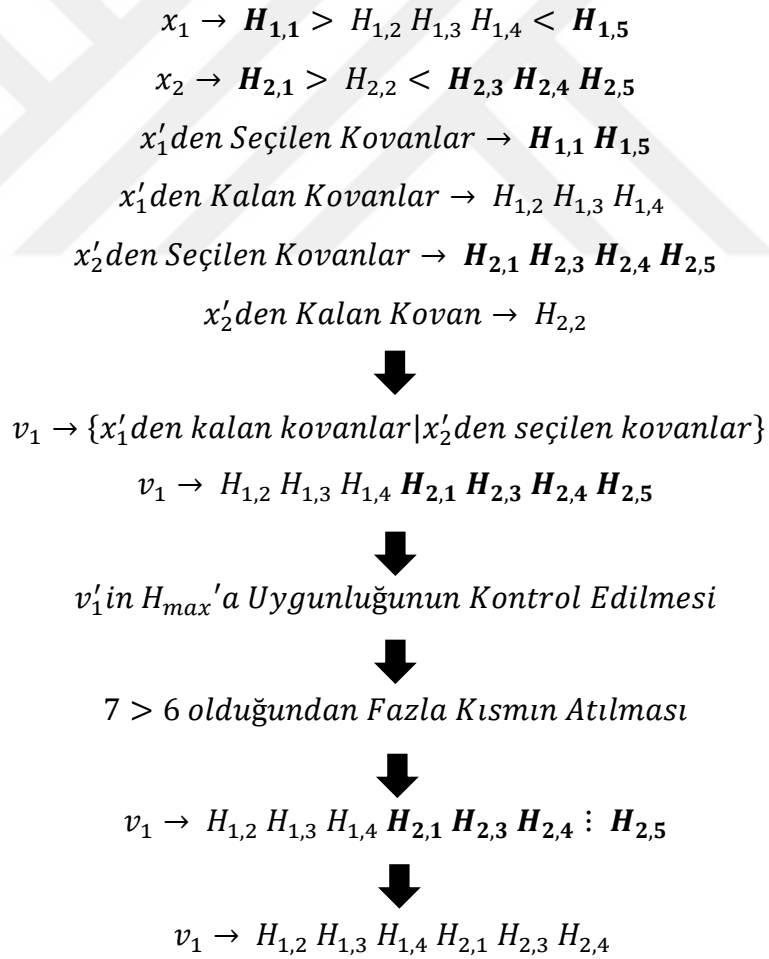
MHABCP'de başlangıç aşamasında kolonideki çözümler, 1 ile maksimum kovan sayısı H_{max} arasında, kovanlardan rasgele üretilerek oluşturulur. Kolonideki her bir çözümün genel matematiksel ifadesi Eşitlik 5.2'deki gibidir.

$$x_i = w_{i0} + \sum_{j=1}^k (w_{ij} * H_{i,j}) \quad (5.2)$$

Burada x_i kolonideki i 'inci çözümü, w_{i0} başlangıç ağırlığını (bias), w_{ij} i 'inci çözüm için j 'inci kovanın ağırlığını, $H_{i,j}$ i 'inci çözümün, j 'inci kovanın matematiksel modelini ifade eder.

MHABCP algoritmasında aday çözüm v_i üretilirken standart ABCP'deki bilgi paylaşım mekanizmasının kullanılması ile birlikte MultiGen Genetik Programlama'da kullanılan iki noktalı yüksek seviyeli çaprazlama operatörünün [25], MHABCP'deki komşuluk araştırma işlemine adaptasyonu gerçekleştirilmiştir. Burada çalıştırılan komşuluk araştırmaya Kovan Değişim Mekanizması (hive exchange mechanism) adı verilmiştir. Mekanizma sayesinde kolonideki farklı çözümlerin kovanları yer değiştirebilmekte ve ayrıca kovanlarda bilgi paylaşım mekanizması kullanılarak çözümlerin iyileştirilmesine olanak sağlanmaktadır.

Mekanizmanın bir örneği Şekil 5.2’de gösterilmiştir. Bu mekanizmada bilgisinden yararlanılan mevcut kaynak çözümü x_1 , rasgele seçilen komşu çözüm x_2 , aday çözüm v_1 ve bu çözümlerin her bir kovanı $H_{i,j}$ ile ifade edilmiştir. Örneğin maksimum kovan sayısı $H_{max} = 6$ olarak verilmiş olsun. Mekanizmada ilk olarak mevcut çözümden ve komşu çözümden değişime uğrayacak kovanlar rasgele seçilir. Kovanlar belirlendikten sonra mevcut çözümden kalan kovanlar ile komşu çözümden seçilen kovanlar birleştirilerek aday çözüm v_i üretilir. Üretimden sonraki aşamada v_i ’nin maksimum kovan sayısı H_{max} ’ı aşıp aşmadığına bakılır. Şekil 5.2’de görüldüğü üzere üretilen aday çözüm 7 kovandan oluşmaktadır. Bu durumda maksimum kovan sayısından fazla olan kovan (Şekil 2.6’da $7 - 6 = 1$) çözümün sonundan çıkarılarak aday çözüm üretilmiş olunur.



Şekil 5.2. MHABCP’de kovan değişim mekanizmasına örnek.

5.3. Deney Çalışmaları

Bu kısımda, önerilen MHABCP'nin verilerdeki önemli (gerçek) öznelilikleri seçme becerisi ve genelleme kabiliyeti araştırılmıştır. MHABCP ile diğer yüksek seviyeli otomatik programlama yöntemleri (ABCP, GP) yapay ve gerçek verilerle yapılan deneylerle kıyaslanmıştır.

5.3.1. Uygunluk Fonksiyonu

MHABCP, ABCP, GP'den çıkarılan modeller Normalize Edilmiş Ortalama Hatanın Karakökü (Normalised Root Mean Square Error-NRMSE) fonksiyonu kullanılarak değerlendirilmiştir. NRMSE fonksiyonu Eşitlik 5.3'te gösterilmiştir.

$$NRMSE = \frac{RMSE}{Y_{max} - Y_{min}} \quad (5.3)$$

Burada $RMSE$ ortalama hatanın karekökünü, $Y_{max} - Y_{min}$ verilerin çıkış değer aralığını ifade eder. $RMSE$ fonksiyonu Eşitlik 5.4'te sunulmuştur.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (f(x_i) - Y_i)^2} \quad (5.4)$$

N örneklerin sayısını, $f(x_i)$ çıkarılan modellerin çıkışlarını, Y_i gerçek çıkışları ifade eder. Yöntemlerden elde edilen çözüm ağaçlarının karmaşıklığı, ağaçların derinliği ve düğüm sayısı ile orantılı olarak Eşitlik 2.9 ile hesaplanmıştır.

5.3.2. Kontrol Parametreleri

Otomatik programlama yöntemlerinin tüm koşullarda kullandıkları parametre değerleri Tablo 5.1'de gösterilmiştir. Tüm yöntemlerde başlangıç çözümleri “*Ramped Half and Half*” metoduyla üretilmiştir. GP'de bireyleri iyileştirmek için çaprazlama, mutasyon ve yeniden eşleme değişim operatörleri kullanılmıştır. ABCP'de kaynakların kalitesini arttırmak için tüm çözümlerde Bilgi Paylaşım Mekanizması kullanılırken, MHABCP'de 0.2 oranında Kovan Değişim Mekanizmasından yararlanılmıştır. Buradaki bölme (/) fonksiyonu korunmuş versiyondur. Bu fonksiyonda da bölen değer 0'a eşitse sonuçta 1 döndürülmekte, diğer durumlarda normal bölme işlemi gerçekleştirilmektedir.

Tablo 5.1. Parametreler.

Parametreler	GP	ABCP	MHABCP
Popülasyon Büyüklüğü	512	-	-
Koloni Büyüklüğü	-	512	512
İterasyon Sayısı	100	100	100
Maksimum Ağaç Derinliği	10	10	5
Çaprazlama Oranı	0.14	-	-
Mutasyon Oranı	0.84	-	-
Yeniden Eşleme Oranı	0.02	-	-
Maksimum Kovan Sayısı	-	-	4
Bilgi Paylaşım Mekanizması Kullanma Oranı	-	1	0.8
Kovan Değişim Mekanizması Kullanım Oranı	-	-	0.2
Kullanılan Fonksiyonlar	+, -, *, /(korumalı)		

5.3.3. Veri Setleri

Bu bölümde yapılan çalışmalarda iki yapay, dört gerçek olmak üzere toplamda altı yüksek boyutlu veri seti kullanılarak yöntemlerin performansı analiz edilmiştir. Yapay veri setlerinde ilgili girişlere göre çıkış parametresi üretilmiş, daha sonra her iki veri setine de 50 gürültü giriş parametresi eklenmiştir. Veri setlerinde çıkış parametresinin üretiminde kullanılan girişler belirli olduğundan bu veriler yöntemlerin öznetelik seçme kabiliyetlerini değerlendirmek için kullanılabilir [134]. Yapay veri setleri Tablo 5.2’de, gerçek veri setleri Tablo 5.3’te gösterilmiştir. Bu bölümdeki çalışmada [49]’daki veri setleri kullanılarak yöntemlerin kıyaslanması hedeflenmiştir.

Tablo 5.2. Yapay veri setlerinin özellikleri.

Fonksiyon	Eğitim Örnekleri	Test Örnekleri	Gürültü
$F_1 = -g \frac{x_1 x_2}{x_3^2}$	70 nokta $x_1, x_3 = rnd[0,1]$ $x_2 = rnd[1,2]$	30 nokta $x_1, x_3 = rnd[0,1]$ $x_2 = rnd[1,2]$	50 gürültü faktörü $= rnd[0,1]$
$F_2 = \frac{30x_1 x_3}{(x_1 - 10)x_2^2}$	1000 nokta $x_1, x_3 = rnd(-1,1)$ $x_2 = rnd(1,2)$	10000 nokta $x_1, x_3 = rnd(-1,1)$ $x_2 = rnd(1,2)$	50 gürültü faktörü $= rnd[0,1]$

F_1 fonksiyonu Newton'un yerçekimi kanunu formülüdür. Formülde yer alan g yerçekimi katsayısıdır ve çalışmamızda katsayı olarak $6.67408E-11$ değeri kullanılmıştır. F_2 fonksiyonu [135]'teki çalışmadan alınmıştır. Gürültü parametreleri, yöntemlerin gerçekten alakalı öznelilikleri seçebilme kabiliyetini değerlendirmek için eklenmiştir.

Tablo 5.3. Gerçek veri setlerinin özellikleri.

Veri Seti	Öznelilikler	Toplam Örnek Sayısı	Eğitim Örnek Sayısı	Test Örnek Sayısı
F_1	53	100	70	30
F_2	53	11000	1000	10000
LD50	626	234	163	71
DLBCL	7399	240	180	60
CCUN	102	1994	1396	598
CCN	100	1994	1396	598

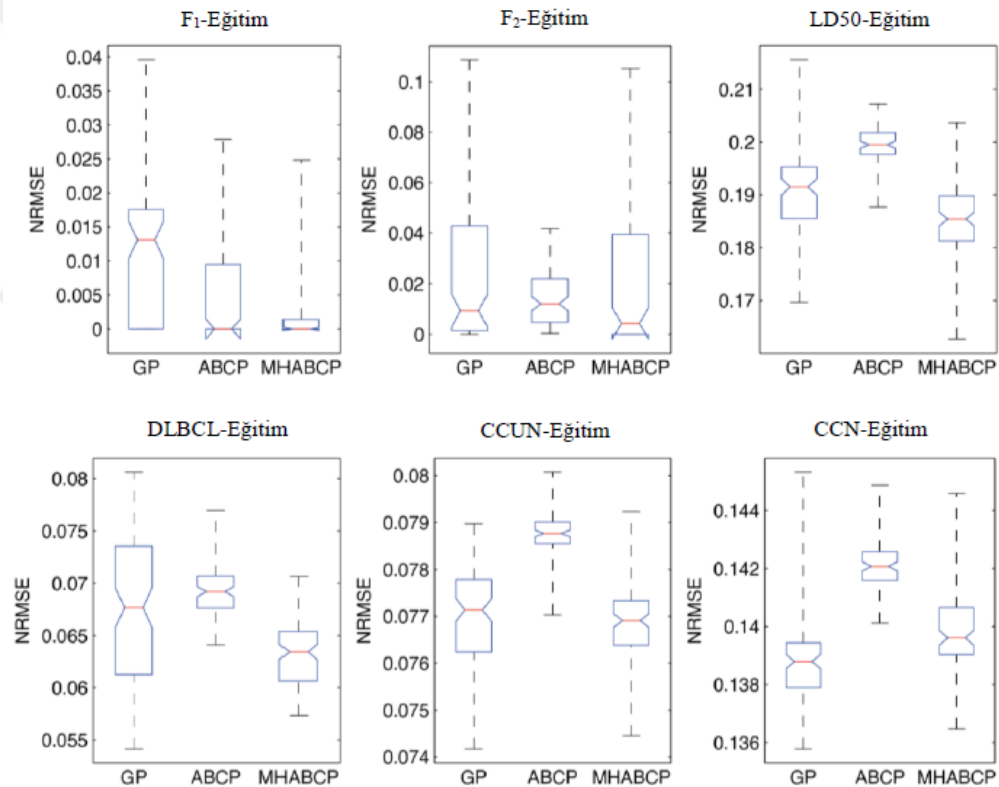
Gerçek veri setleri UCI'dan [109] alınmıştır. Veri setleri yüksek boyutlu olduğundan gerçek/ilgili öznelilikleri seçerek model çıkarımı yapmak oldukça zordur. LD50 veri seti, LD50 olarak temsil edilen farmakokinetik parametrenin (Medyan Ölümcül Doz) değerini tahmin etmek için kullanılır. Vanneschi ve diğerleri, GP için geometrik anlam bilimsel operatörlerin kullanıldığı GP tabanlı GSGP'yi önermişlerdir [136]. Bu operatörler LD50 veri setinde GP'nin genelleme kabiliyetini geliştirmesinde yüksek performans göstermiştir. Deneylerde kullanılan ikinci veri seti, Rosenwald tarafından toplanan Diffüz Büyük B Hücreli Lenfoma (Diffuse Large B-Cell Lymphoma-DLBCL)'dir [137]. Bu veri setinde DLBCL'ye sahip olan ve kemoterapi gören hastaların yaşam süreleri tahmin edilmeye çalışılmıştır. Diğer veri setleri Topluluklar ve Suçun Normalleştirilmemiş Veri Seti (Non-Normalized Dataset Of Communities and Crime-CCUN) ve Topluluklar ve Suç ile Normalleştirilmiş Veri Seti (Normalized Dataset with Communities and Crime-CCN) ABD'de kişi başına düşen suç oranını tahmin etmek için kullanılmıştır. Çalışmamızda CCN ve CCUN veri setlerindeki kayıp veriler çıkartılmıştır. Tablo 5.3'te görüldüğü üzere tüm veri setleri, makine öğrenmesi tekniklerinde yaygın olarak kullanılan verileri ayırma oranları (toplam örnek sayının %70'i eğitim ve %30'u test örnekleri olacak şekilde) kullanılmıştır.

5.4. Sonuçlar ve Tartışmalar

Otomatik Programlama yöntemlerinin deney sonuçları bu alt bölümde verilmiş ve tartışılmıştır. Yüksek boyutlu sembolik regresyon problemleri için yeni önerdiğimiz MHABCP'nin ilgili/gerekli öznelilikleri seçme, aşırı uydurma (overfitting) karşı direnci standart GP ile ABCP algoritmalarıyla karşılaştırılarak değerlendirilmiştir.

5.4.1. Eğitim Setleri Sonuçları- Öğrenme Kabiliyeti

Yöntemlerin her biri 100 iterasyona sahip birbirinden bağımsız 100 koşma ile çalıştırılmıştır. Şekil 5.3, her koşmanın en iyi bireyinin NRMSE'lerinin eğitim dağılım grafiğini; Şekil 5.5, karşılık gelen test NRMSE'lerinin dağılım grafiğini göstermektedir.

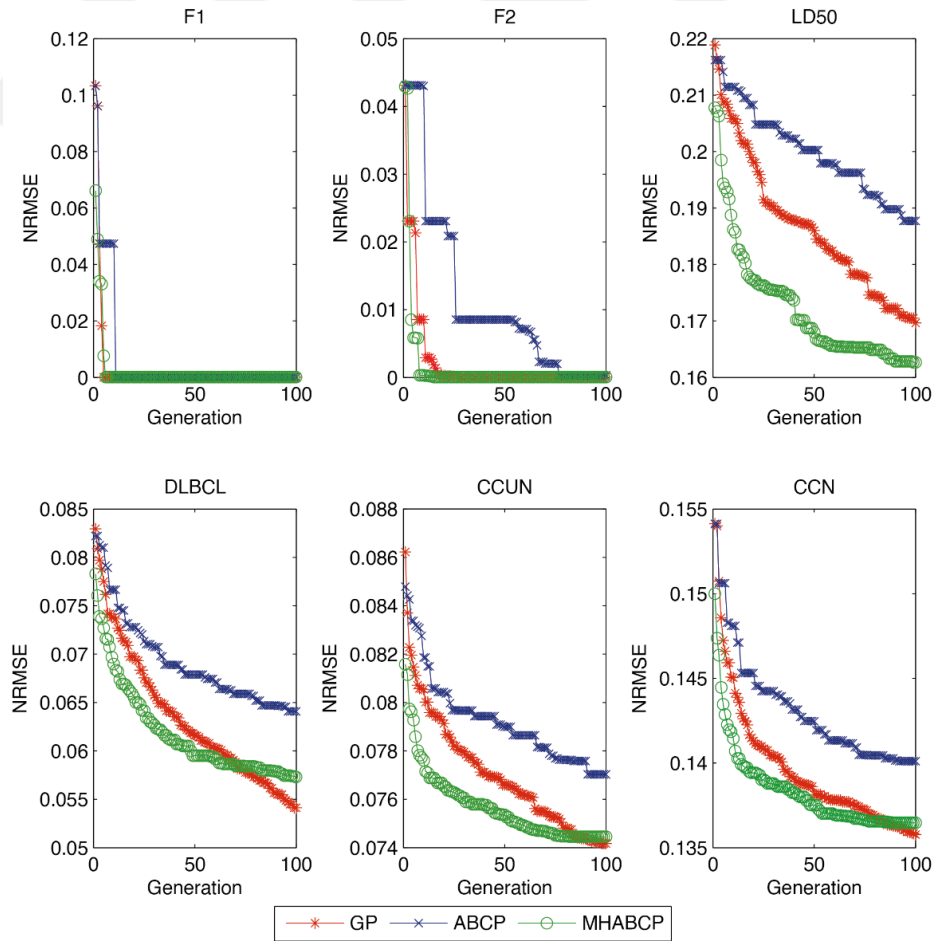


Şekil 5.3. En iyi bireylerin eğitim NRMSE dağılımı.

Grafiklerde, her kutu bir yöntemi temsil eder. Kutulardaki hata çubukları NMRSE'lerde minimum/maksimum değerlerini ve çizgiler NMRSE'lerin ortanca değerlerini göstermektedir. Ortanca değerlerin, aykırı değerlere daha dirençli olduğu düşünüldüğünden [132], değerlendirmelerde ortanca değerler ortalamalar ile birlikte kullanılmıştır. GP'nin NMRSE aralığı çok daha geniş dağılım gösterdiğinden GP'nin

standart sapmaları diğer yöntemlerden daha yüksektir. F_2 'de, dağılımlarda üç yöntemde minimum değerler arasında önemli bir fark yoktur. F_1 , F_2 ve DLBCL için ABCP, GP'den daha iyi bir performansa sahiptir. Şekil 5.3'de ifade edildiği üzere, MHABCP'nin performansının, neredeyse tüm problemlerde diğer yöntemlerden daha başarılı olduğu görülmektedir.

Şekil 5.4, gerçek ve yapay veri setleri için üç yöntemin eğitim gelişimi performansını İterasyon –NMRSE grafikleri ile göstermektedir. Her bir yöntemde eğitim veri setindeki her iterasyon için tüm koşmalardaki NRMSE'ler kaydedilir. Toplamda 100x100'lük iterasyon-koşmadaki NMRSE matrisi elde edilmiş olunur. Sonraki aşamada, her iterasyonun 100 koşmada elde edilen NMRSE değerlerinden minimum NMRSE'si işaretlenerek 100x1'lik iterasyon-100 koşmanın minimum NMRSE'si matrisi çıkarılır. Bahsedilen adımlar tüm yöntemler için gerçekleştirilerek yöntemlerin eğitim evrimsel gelişim grafiği çizdirilmiştir.

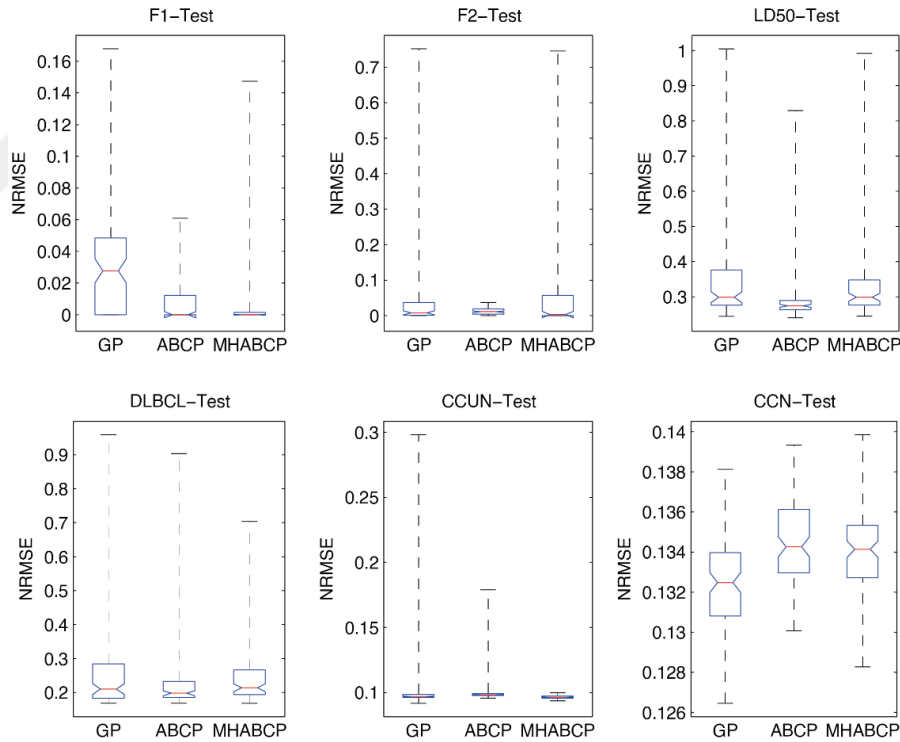


Şekil 5.4. Eğitim gelişim grafikleri.

MHABCP, DLBCL dışındaki tüm veri setlerinde başarılı performans göstermektedir. CCUN için GP ve MHABCP'nin gelişimi birbirine oldukça yakındır. ABCP karşılaştırılabilir bir performansa sahip olmasına rağmen, gelişimi yetersizdir. NRMSE'lerin GP ve MHABCP arasındaki benzerliği sonraki nesillerde artar. MHABCP, diğer yöntemlerden daha düşük bir NRMSE elde etmiştir. F_1 ve F_2 için, ABCP diğer yöntemlere nazaran daha yavaş en düşük NRMSE'ye ulaşmıştır. Bu veri setlerindeki gelişim incelendiğinde, ABCP diğer iki yöntemden başarısız olmuştur.

5.4.2. Test Setleri Sonuçları – Genelleme Kabiliyeti

Şekil 5.5 her koşmanın en iyi bireyinin karşılık gelen test NRMSE'lerinin dağılım grafiklerini göstermektedir. Grafiklerde görüldüğü üzere genel olarak, test dağılımının grafikleri, CCUN hariç, eğitim dağılımlarının grafiğine benzemektedir.



Şekil 5.5. En iyi bireylerin test NRMSE dağılımı.

F_2 için ABCP birbirine çok yakın test sonuçları elde etmiştir. GP, diğer yöntemlerle karşılaştırıldığında CCN veri seti dışında en yüksek test NRMSE'ye sahiptir. CCUN'de, üç yöntemde de test hata değerleri birbirine yakın değerlerdir. LD50 ve DLBCL'de, tüm yöntemler yüksek test hata değerlerine sahiptir. Test NMRSE dağılımlarına bakıldığında

her üç yöntemin başarılı test sonuçları elde etmesi, yöntemlerin aşırı öğrenmeye karşı dirençli olduğunu ve genelleme kabiliyetlerinin yüksek olduğunu göstermektedir.

5.4.3. Öznitelik Seçimi Sonuçları

Bu alt bölümde öznitelik seçim sonuçları ve MHABCP'den çıkarılan modeller analiz edilecektir. Tablo 5.4, azalan önem sırasına göre yöntemlerin seçtiği en önemli 10 özniteliği göstermektedir. Yöntemlerin, ortak öznitelikleri modellerinde kullanması, bu özniteliklerin çıktı parametreleri tahmini için önemli olduğunu göstermektedir.

Tablo 5.4. En önemli 10 öznitelik.

Proble m	Yönte m	10 Önemli Öznitelik										Ortak Öznitlikler
F ₁	GP	6	1	7	2	3	8	25	31	17	23	6 , 1 , 7 , 2 , 3 , 8
	ABCP	3	1	2	6	7	8	31	25	42	50	
	MHABCP	3	1	7	2	6	8	29	9	26	43	
F ₂	GP	3	1	2	30	5	24	22	10	16	44	1 , 2 , 24 , 3 , 44 , 5
	ABCP	1	3	2	34	43	24	21	44	5	6	
	MHABCP	3	1	2	5	44	48	34	22	24	31	
LD50	GP	30	147	55	593	549	287	123	253	289	471	30 , 593
	ABCP	30	10	55	288	56	550	593	518	287	502	
	MHABCP	30	253	518	593	508	285	288	550	561	39	
DLBCL	GP	4140	4671	4675	5760	4678	4691	3939	4669	4681	4689	4675
	ABCP	4688	4675	4677	4676	4678	181	6884	4690	3713	4667	
	MHABCP	4675	4696	4673	4677	4688	4671	1284	6559	2727	3713	
CCUN	GP	51	45	4	75	73	102	39	42	49	35	39 , 4 , 42 , 45 , 49 , 51 , 73
	ABCP	51	45	4	39	42	73	49	41	102	69	
	MHABCP	51	45	4	39	42	72	69	73	41	49	
CCN	GP	45	4	51	91	72	3	12	69	49	39	3 , 39 , 4 , 45 , 51 , 69 , 72 , 91
	ABCP	45	51	4	72	91	39	3	42	69	41	
	MHABCP	45	3	72	4	51	69	91	39	12	70	

F₁'de, ABCP öncelikle tüm eşitliklerinde ortak öznitelikleri kullanmıştır. F₂'de tüm yöntemler ilk olarak gürültü parametreleri yerine gerçek öznitelikleri kullanmıştır. Bu durum, üç yöntemin de öznitelik seçiminde veri setleri ile tutarlı olduğunu göstermektedir. Veri setlerinin boyutu, bir diğer deyişle sahip oldukları giriş sayısı arttıkça, çıkarılan modellerdeki ortak özniteliklerin sayısı azalmıştır. Genel olarak MHABCP'nin yüksek boyutlu gerçek dünya veri setleriyle uğraşırken önemli öznitelikleri seçmede daha başarılı olduğu görülmektedir.

5.4.4. Çıkarılan Modellerin Analizi

Tablo 5.5, Tablo 5.6, Tablo 5.7 MHABCP tarafından gerçekleştirilen tüm çalışmalar sonucunda elde edilen en iyi modellerin eşitliklerini göstermektedir. F₁ ve F₂'nin en iyi modelleri sadece x_1 , x_2 , x_3 içererek gürültü parametrelerini elimine etmiştir. Özellikle F₁'de en iyi modelin eşitliği F₁'in gerçek formülüne çok yakındır.

MHABCP'nin en iyi modellerinin ağaç yapılarının özellikleri Tablo 5.8'de gösterilmektedir. Ağaçların karmaşıklığı Eşitlik 2.9 ile hesaplanmıştır. Veri setlerinde boyut arttıkça, modellerin karmaşıklığının ve düğüm sayısının arttığı görülmektedir.

Tablo 5.9 tüm otomatik programlama yöntemlerinin; ortalama, standart sapma, maksimum, minimum ve ortanca değerlerle karşılaştırmalı sonuçlarını göstermektedir. Tablo 5.9'da veri setlerinde ortalaması en iyi değere sahip yöntem koyu renk ile belirtilmiştir. Tabloda üzere F₁'de hem eğitim hem de test hataları açısından, MHABCP, diğer iki yönteme göre daha düşük hata değerine sahiptir. Bu durum, F₁ veri seti için diğer yöntemlere nazaran MHABCP modelinin en az hatayla en doğru model bulunduğunu gösterir. F₂'de, en az hata MHABCP'de ve en iyi ortalama değer ABCP'de bulunmuştur. LD50 ve DLBCL'de tüm yöntemler için ortalama değerler çok yakındır. DLBCL'de GP, eğitim setinin en düşük ortalama değerine sahiptir. CCUN'de ABCP'nin standart sapması diğer yöntemlerden çok daha düşüktür. CCUN ve CCN için tüm yöntemler benzer hata oranlarına sahiptir. Sonuçlar genel olarak değerlendirildiğinde, GP, minimum ve ortanca değerleri açısından ABCP'den daha başarılıdır. Ancak, GP'nin standart sapması diğer iki yöntemden daha yüksektir. MHABCP'nin başarılı sonuçlar gösterdiği ve diğer yöntemlerle karşılaştırılabilecek performansa sahip olduğu görülmektedir.

Tablo 5.5. MHABCP’de F₁, F₂, LD50 veri setleride çıkarılan en iyi modellerin eşitlikleri.

Problem	En İyi Modeller		Basitleştirilmiş Model
	Terim	Eşitlik	
F ₁	Bias	2.02e-28	$y = \frac{6.67 e^{-11} x_1 x_2}{x_3^2} + 2.02e - 28$
	Kovan 1	0.0	
	Kovan 2	$\frac{6.67 e^{-11} x_1 x_2}{x_3^2}$	
F ₂	Bias	-37.9	$y = 3.86e^{-8} - ((3x_1x_3 + 0.3x_1^2x_3 + 0.0303x_1^3x_3 + 0.00304x_1^4x_3))/x_2^2$
	Kovan 1	$-(0.00304x_1^4x_3)x_2^2$	
	Kovan 2	$-(0.3x_1^2x_3)/x_2^2$	
	Kovan 3	$37.9 - (3x_1x_3)/x_2^2$	
	Kovan 4	$-(0.0303x_1^3x_3)/x_2^2$	
LD50	Bias	8611.0	$y = 1300x_{346} - 2877x_{56} + 354x_{348} - 496x_{424} - \frac{(1.4e^{12}x_{285} - 4.4e^{12}x_{285} - 4.4e^{12}x_{424})}{6.23e^{15}x_{27}} - 13004.4x_{30}x_{346}$ $- \frac{3.52e^{13}x_{13} + 3.52e^{13}x_{13}x_{346} - 3.52e^{13}x_{348} - 3.52e^{13}x_{478}}{6.23e^{15}x_{27}} - 999x_{27}x_{56}x_{346}^2 + 638x_{272}x_{232}x_{346}^2$ $- 999x_{56}x_{232}x_{346} + 1300x_{232}x_{285}x_{285}x_{346}x_{369} - 1300x_{232}x_{346}x_{369}x_{424} + 1300x_{30}x_{232}x_{346}^2x_{369}$ $+ 8611$
	Kovan1	$638x_{27}^2x_{232}x_{346}^2 - (1.4e^{15}x_{348})/(4.4e^{12}x_{285} - 4.4e^{12}x_{424}) - 319x_{424}$	
	Kovan 2	$1300x_{346} - 1300x_{30}x_{346} + 1300x_{232}x_{346}x_{369}(x_{285} - x_{424} + x_{30}x_{346})$	
	Kovan 3	$-999x_{56}(x_{232}x_{346} + x_{27}x_{346}^2 + 2.87)$	
	Kovan 4	$354x_{348} - 177x_{424} - (177x_{27})/(x_{13} + x_{346} - x_{346} - x_{478})$	

Tablo 5.6. MHABCP’de DLBCL ve CCUN veri setlerinde çıkarılan en iyi modellerin eşitlikleri.

Problem	En İyi Modeller		Basitleştirilmiş Model
	Terim	Eşitlik	
DLBCL	Bias	0.00201	$y = 0.156x_{2766} - 0.312x_{3909} + 0.312x_{3922} + 0.132x_{4681} + 0.156(x_{475} + x_{1777}x_{5238})(x_{2766} - x_{3909} + x_{4887}x_{5238}) + 0.132x_{1777}x_{6866} \\ + 0.156x_{4887}x_{5238} + 0.288x_{4887}x_{6816} + \frac{0.00621(x_{1777} + x_{1777}x_{4887})(x_{5238} - x_{6816} + x_{4887}x_{5238})}{x_{6816}} \\ + 0.132x_{5214}^2x_{5238}(x_{475} + x_{2766}x_{5238}) - \frac{0.00102x_{4887}x_{5214}x_{5238}(x_{4681} + x_{4887}x_{5238})(x_{3909} - x_{2766} + x_{4887}x_{5214})}{x_{2729}} \\ + 0.00201$
	Kovan 1	$-(0.00102x_{4887}x_{5214}x_{5238}(x_{4681} + x_{4887}x_{5238}) * (x_{3909} - x_{2766} + x_{4887}x_{5214})/x_{2729}$	
	Kovan 2	$(0.00621x_{1777}(x_{4887} + 1)(x_{5238} - x_{6816} + x_{4887}x_{5238}))/x_{6816}$	
	Kovan 3	$0.156x_{2766} - 0.312x_{3909} + 0.156(x_{475} + x_{1777}x_{5238}) * (x_{2766} - x_{3909} + x_{4887}x_{5238}) + 0.156x_{4887}x_{5238} + 0.156x_{4887}x_{6816}$	
	Kovan 4	$0.132x_{3922} + 0.132x_{4681} + 0.132x_{1777}x_{6816} + 0.132x_{4887}x_{6816} + 0.132x_{5214}^2x_{5238}(x_{475} + x_{2766}x_{5238})$	
CCUN	Bias	261.0	$y = 62x_{51} - 0.533x_{49} + 62x_{69} - 0.533x_{70} - 0.533x_{69}(x_{30} + x_{45}) + 0.00116x_{39}(x_{69} + x_{72}) - \frac{7.11e^{-15}(8.72e^{15}x_{49} + 8.72e^{15}x_{70})}{+0.533x_{42}x_{70}} \\ - \frac{0.533x_{72}}{x_{39}^2} + 5.17e^{-5}x_3(x_4x_{39} + 2x_{39}^2)(x_{87} + x_{10}x_{69}) + 0.00116x_{10}x_{69}^2x_{89} + \frac{0.00116x_{39}x_{100}}{x_{30}} \\ - \frac{0.00116x_{100}(x_4 + x_{30})}{x_{10}} + 261$
	Kovan 1	$-(62(x_{49} + x_{70} - x_{51}x_{89} - x_{69}x_{89}))/x_{89}$	
	Kovan 2	$0.533x_{42}x_{70} - 0.533x_{70} - 0.533x_{69}(x_{30} + x_{45}) - 0.533x_{49} - (0.533x_{72})/x_{39}^2$	
	Kovan 3	$0.00116x_{39}(x_{69} + x_{72}) + 0.00116x_{10}x_{69}^2x_{89} + \frac{0.00116x_{39}x_{100}}{x_{30}} - (0.00116x_{100}(x_4 + x_{30}))/x_{10}$	
	Kovan 4	$5.17e - 5x_3x_{39}(x_{87} + x_{10}x_{69})(x_4 + 2x_{39})$	

Tablo 5.7. MHABCP’de CCN veri setinde çıkarılan en iyi modelin eşitlikleri.

Problem	En İyi Modeller		Basitleştirilmiş Model
	Terim	Eşitlik	
CCN	Bias	0.352	$y = 0.12x_3 - 0.241x_{14} + 0.0138x_{30} + 0.12x_{34} + 0.198x_{39} - 0.319x_{45} - 0.0138x_{51} + 0.0138x_{72} + 0.261x_3x_{39} + 0.12x_{14}x_{45} - 0.261x_{30}x_{39} + 0.212x_{18}x_{60} - 0.261x_{39}x_{45} + 0.261x_{39}x_{69} + 0.261x_{39}x_{72} + 0.12x_{39}x_{75} + 0.212x_{51}x_{69} + 0.319x_{34}x_{91} + 0.319x_{45}x_{51}x_{75} - 0.319x_{51}x_{72}x_{75} + 0.352$
	Kovan 1	$0.261x_{39}(x_3 - x_{30} - x_{45} + x_{69} + x_{72})$	
	Kovan 2	$0.319x_{39} - 0.319x_{45} + 0.319x_{34}x_{91} + 0.319x_{51}x_{75}(x_{45} - x_{72})$	
	Kovan 3	$0.0138x_{30} - 0.0138x_{51} + 0.0138x_{72} + 0.212x_{18}x_{60} + 0.212x_{51}x_{69}$	
	Kovan 4	$0.12x_3 - 0.241x_{14} + 0.12x_{34} - 0.12x_{39} + 0.12x_{14}x_{45} + 0.12x_{39}x_{75}$	

Tablo 5.8. MHABCP’de en iyi modellerin analizi.

Problem	Kriterler	Değer
F₁	Kovan Sayısı	2
	Düğüm Sayısı	16
	Karmaşıklık	48
	Ağaç Derinliği	5
	Giriş Sayısı	3
	Modelde Kullanılan Girişler	X ₁ , X ₂ , X ₃
F₂	Kovan Sayısı	4
	Düğüm Sayısı	42
	Karmaşıklık	136
	Ağaç Derinliği	5
	Giriş Sayısı	3
	Modelde Kullanılan Girişler	X ₁ , X ₂ , X ₃
LD50	Kovan Sayısı	4
	Düğüm Sayısı	70
	Karmaşıklık	258
	Ağaç Derinliği	5
	Giriş Sayısı	11
	Modelde Kullanılan Girişler	X ₁₃ , X ₂₇ , X ₃₀ , X ₅₆ , X ₂₃₂ , X ₂₈₅ , X ₃₄₆ , X ₃₄₈ , X ₃₆₉ , X ₄₂₄ , X ₄₇₈
DLBCL	Kovan Sayısı	4
	Düğüm Sayısı	86
	Karmaşıklık	336
	Ağaç Derinliği	5
	Giriş Sayısı	11
	Modelde Kullanılan Girişler	X ₄₇₅ , X ₁₇₇₇ , X ₂₇₂₉ , X ₂₇₆₆ , X ₃₉₀₉ , X ₃₉₂₂ , X ₄₆₈₁ , X ₄₈₈₇ , X ₅₂₁₄ , X ₅₂₃₈ , X ₆₈₁₆
CCUN	Kovan Sayısı	4
	Düğüm Sayısı	74
	Karmaşıklık	280
	Ağaç Derinliği	5
	Giriş Sayısı	15
	Modelde Kullanılan Girişler	X ₃ , X ₄ , X ₁₀ , X ₃₀ , X ₃₉ , X ₄₂ , X ₄₅ , X ₄₉ , X ₅₁ , X ₆₉ , X ₇₀ , X ₇₂ , X ₈₇ , X ₈₉ , X ₁₀₀
CCN	Kovan Sayısı	4
	Düğüm Sayısı	60
	Karmaşıklık	220
	Ağaç Derinliği	5
	Giriş Sayısı	13
	Modelde Kullanılan Girişler	X ₃ , X ₁₄ , X ₁₈ , X ₃₀ , X ₃₄ , X ₃₉ , X ₄₅ , X ₅₁ , X ₆₀ , X ₆₉ , X ₇₂ , X ₇₅ , X ₉₁

Tablo 5.9. GP, ABCP, MHABCP simülasyon sonuçları

Problem	Yöntem		Ortalama	Standart Sapma	Maksimum	Minimum	Ortanca
F ₁	GP	Eğitim	0.012	0.010	0.040	2.99E-17	0.013
		Test	0.032	0.035	0.168	4.19E-17	0.028
	ABCP	Eğitim	0.005	0.008	0.028	3.50E-17	4.37E-17
		Test	0.009	0.015	0.061	4.19E-17	5.62E-17
	MHABCP	Eğitim	0.001	0.003	0.025	2.64E-17	5.42E-17
		Test	0.004	0.016	0.147	3.73E-17	8.46E-17
F ₂	GP	Eğitim	0.028	0.037	0.109	3.21E-08	0.009
		Test	0.085	0.176	0.752	2.85E-08	0.008
	ABCP	Eğitim	0.014	0.011	0.042	3.06E-04	0.012
		Test	0.013	0.009	0.037	2.59E-04	0.011
	MHABCP	Eğitim	0.024	0.030	0.105	1.25E-10	0.004
		Test	0.074	0.150	0.745	1.18E-10	0.003
LD50	GP	Eğitim	0.191	0.008	0.216	0.170	0.192
		Test	0.367	0.161	1.005	0.244	0.299
	ABCP	Eğitim	0.199	0.003	0.207	0.188	0.200
		Test	0.295	0.081	0.830	0.240	0.274
	MHABCP	Eğitim	0.185	0.008	0.204	0.163	0.185
		Test	0.358	0.159	0.992	0.245	0.299
DLBCL	GP	Eğitim	0.068	0.007	0.081	0.054	0.068
		Test	0.273	0.158	0.960	0.169	0.210
	ABCP	Eğitim	0.069	0.003	0.077	0.064	0.069
		Test	0.226	0.090	0.904	0.169	0.198
	MHABCP	Eğitim	0.063	0.003	0.071	0.057	0.063
		Test	0.252	0.101	0.704	0.168	0.214
CCUN	GP	Eğitim	0.077	0.001	0.079	0.074	0.077
		Test	0.101	0.021	0.298	0.092	0.097
	ABCP	Eğitim	0.079	4.53E-04	0.080	0.077	0.079
		Test	0.099	0.008	0.179	0.096	0.098
	MHABCP	Eğitim	0.077	0.001	0.079	0.074	0.077
		Test	0.096	0.001	0.100	0.094	0.096
CCN	GP	Eğitim	0.139	0.001	0.145	0.136	0.139
		Test	0.133	0.002	0.138	0.126	0.132
	ABCP	Eğitim	0.142	0.001	0.145	0.140	0.142
		Test	0.134	0.002	0.139	0.130	0.134
	MHABCP	Eğitim	0.140	0.001	0.145	0.136	0.140
		Test	0.134	0.002	0.140	0.128	0.134

MHABCP yönteminin, ABCP ve GP'ye göre önemini % 5 önem düzeyinde test etmek için Friedman testi yapılmıştır. Testin sonuçları Tablo 5.10'da verilmiştir. Tabloda (+) simgesi yöntemler arasında farkın bulunduğu, (-) yöntemler arasında farkın bulunmadığını göstermektedir. Sonuçlara göre yöntemin, yüksek boyutlu verilerdeki diğer otomatik programlama tabanlı öznelilik seçim yöntemlerine kıyasla önemli ölçüde daha iyi performans gösterdiği gözlemlenmiştir.

Tablo 5.10. MHABCP'nin GP ve ABCP'ye karşı Friedman testi sonuçları.

	GP-MHABCP	ABCP-MHABCP
Problem	Anlamlılık Testi (eğitim, test)	Anlamlılık Testi (eğitim, test)
F ₁	(+, +)	(-, -)
F ₂	(-, -)	(-, -)
LD50	(+, -)	(+, +)
DLBCL	(+, -)	(+, +)
CCUN	(-, -)	(+, +)
CCN	(+, +)	(+, -)

6. BÖLÜM

ÇOK SINIFLI DESEN SINIFLANDIRMA İÇİN YAPAY ARI KOLONİ PROGRAMLAMA TANIMLAYICI

6.1. Problem Tanımı

Resim tanımlayıcısı (Image Descriptor) şekil, renk gibi öznitelikleri çıkararak/belirleyerek resim hakkında bilgi sağlar. Resim tanımlayıcılar yoğunluk ve seyreklik tipleri olmak üzere iki farklı tipe sahiptirler. Yoğunluk tipli tanımlayıcılar, resimde piksel piksel ilerleyerek öznitelikleri çıkaran yaklaşıma sahiptirler. En çok kullanılan yoğunluk tipi tanımlayıcılardan biri Yerel İkili Örüntü (LBP)'dür. Seyrek tipi tanımlayıcılara SIFT (Scale Invariant Transform) ve SURF (Speeded up Robust Features) örnek verilebilir. Bu tanımlayıcılar resimdeki her pikseli değerlendirerek modeli oluştururlar [138].

GP ve ABCP, sistemlerin bağımlı ve bağımsız değişkenleri arasındaki ilişkiyi belirleyerek model üreten yüksek seviyeli otomatik programlama yöntemleri oldukları önceki bölümlerde gösterilmiştir. Bu bölümde desen sınıflandırma problemi için GP-tanımlayıcıdan esinlenerek ABCP resim tanımlayıcısı ilk kez önerilmiş ve elde edilen modellerin desen sınıflandırma probleminde başarısı araştırılmıştır.

Son yıllarda görüntü işleme problemlerinde otomatik programlama yöntemleri daha sık kullanılmıştır. Tackett [139] GP'yi, resimlerden elde edilen öznitelik vektörlerini kategorilere sınıflandırmak için kullanmıştır. [140]'ta görüntü sınıflandırması için otomatik olarak alan-uyarlamalı global özellik tanımlayıcıları oluşturmak için Çok Amaçlı Genetik Programlama (Multi Objective Genetic Programming-MOGP) temelli, etki alanı uyarlamalı öğrenme yöntemi uyarlanmıştır. [141]'de öznitelik tespiti için GP önerilmiştir. Yöntem, görüntüyü alt bölgelerine ayırarak bu bölgelerin SURF noktalarını

çıkarmaktadır ve SVM sınıflandırıcı kullanarak başarılı sınıflandırma sonuçları elde etmiştir. Iqbal ve diğerleri, transfer öğrenme ile GP'yi birlikte kullanan resim sınıflandırma yöntemi önermişlerdir [142]. Transfer öğrenmeden çıkarılan kod parçaları (fragment) ile elde edilen GP ağaçları daha kullanışlı başlangıç popülasyonu üreterek sınıflandırma performansını geliştirmiştir. Lensen ve diğerleri, resimdeki farklı bölgeleri otomatik olarak belirleyip, bu bölgelerin Yönlendirilmiş Gradyanların Histogramı (Histogram of Oriented Gradients-HOG)'nı çıkararak sınıflandırma yapan yeni bir GP yöntemi sunmuşlardır [143]. Geniş arama alanı ile yöntemin tüm aşamaları optimize edilmiştir. HOG ve GP adaptasyonunun, SURF metoduna nazaran yüksek düzey öznetelik çıkarmada daha etkili bir yöntem olduğu gösterilmiştir. Al- Sahaf ve diğerleri, üç farklı varyasyona sahip iki aşamalı GP (2 Tier GP, 2 Tier GP-line, 2 Tier GP-mix) önermişlerdir [144]. Aşamaların ilkinde, ham satır piksel değerlerini giriş değerleri olarak kullanmak yerine; satır piksel değerlerini tek bir sayısal değere dönüştürebilen özel fonksiyonlar tanımlamışlardır. Diğer aşamada ise özel fonksiyonların çıktıları kullanılarak sınıflandırma yapılmıştır. Örnek pencere aralığının esnek olması nedeniyle 2TGP-mix en başarılı yöntem olarak değerlendirilmiştir.

Metasezgisel algoritmalarından ABC ile PSO, GA ve diğer evrimsel hesaplama algoritmalarıyla yapılan karşılaştırmalı çalışmalar, ABC'nin mühendislik problemlerinde yerel ve küresel optimum sonuca ulaşma açısından performansının daha iyi olduğunu göstermiştir [145]. ABC görüntü işleme alanında; özellikle görüntü kümelemede [146], görüntü çakıştırma [147], örüntü tanıma [148], görüntü segmentasyonunda [149-153] ve görüntü sınıflandırmada [154] başarıyla uygulanan bir algoritmadır. Agrawal ve diğerleri, Bilgisayarlı Tomografi görüntülerinde teşhis için ABC algoritması kullanmışlardır. ABC ile öznetelikler seçilerek k-En Yakın Komşu (k-Nearest Neighbour-k-NN) ve SVM sınıflandırıcıları ile görüntüler sınıflandırılmıştır [154]. ABC-SVM hibriti oldukça başarılı sınıflandırma sonucu elde etmiştir. Bu bölümde yapılan çalışma, ABC tabanlı ABCP algoritmasının desen sınıflandırmasında performansını izleyen ve matematiksel modeller geliştiren ilk çalışmadır.

Günümüzde, hesaplama basitliği ve ışık değişimlerine karşı sağlamlığı nedeniyle Yerel İkili Örüntü (Local binary pattern-LBP) tabanlı tanımlayıcılar; görüntü sınıflandırması problemi, desen tanıma ve bilgisayar görüşü gibi çeşitli alanlardaki önemi nedeniyle geniş çapta araştırılmıştır [138, 155-158]. Liu ve diğerleri, merkezi piksel yoğunluğunu

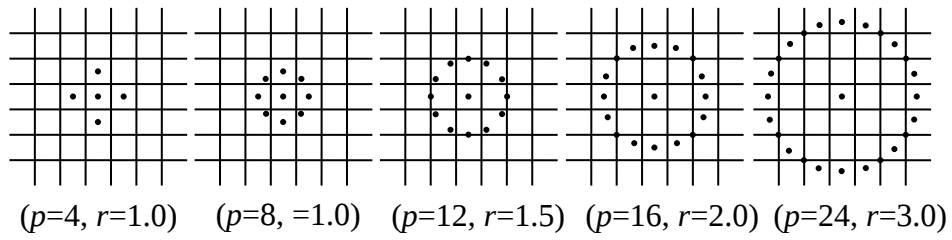
(Central Intensity LBP- CI-LBP), komşu piksel yoğunluğunu (Neighbours Intensity LBP- NI-LBP) dikkate alan yoğunluk tabanlı ve radyal farkı (Radial Difference LBP- RD-LBP), açısal farkı (Angular Difference LBP- AD-LBP) dikkate alan fark tabanlı olmak üzere dört LBP tabanlı tanımlayıcı önermişlerdir [138]. Tanımlayıcılar standart LBP ile aynı yapıya sahip olup eğitim aşamasına ve parametre ayarlamasına ihtiyaç duymamaktadır. Üç farklı desen kümesinde yapılan çalışmada, tanımlayıcılar geleneksel LBP'ye göre gri ölçekli ve rotasyon duyarlı desen sınıflandırmada daha iyi performans göstermiştir. Al-Sahaf ve diğerleri, sınıf başına bir tek örnek kullanan GP tabanlı LBP görüntü tanımlayıcı önermişlerdir [155]. Metodun GP tabanlı olan/olmayan beş farklı yöntemle karşı daha başarılı olduğu gösterilmiştir. Ojala ve diğerleri, rotasyon duyarlı, basit ve gri ölçekli varyanslar açısından sağlam düzgün LBP (Uniform LBP) yaklaşımını sunmuşlardır [156]. Sinha ve diğerleri, Wigner dağılımını LBP ile birlikte kullanan Wigner tabanlı LBP tanımlayıcı (Wigner-Based Local Binary Patterns-WLBP) önermişlerdir [157]. Farklı sınıflandırıcılar kullanılarak yapılan deneylerde metod, geleneksel LBP'ye nazaran daha iyi performans göstermiştir. Pan ve diğerleri, iki bölümlü öznetelik tabanlı LBP (Feature Based Local Binary Pattern-FBLBP) yaklaşımı sunmuşlardır [158]. Çalışmanın ilk aşamada geleneksel LBP kullanılırken diğer aşamada büyüklük vektörünün ortalama ve varyansı kullanılmıştır.

6.2. Görüntü Tanımlayıcılar

Bu bölümde, çalışmayla direk ilgili görüntü tanımlayıcıları hakkında bilgi verilecektir.

6.2.1. Yerel İkili Örüntü (Local Binary Pattern-LBP)

Yerel İkili Örüntü [159], görüntü işlemede en çok ve yaygın kullanılan metotlardan biridir. LBP operatörü görüntüde, kayan pencereler şeklinde ilerleyerek ve bu pencerelerin ortasındaki merkez piksel değerini eşik seviyesi olarak seçerek birbirlerine göre karşılaştırması sonucu ikili değerler atar. Oluşturulan ikili sayı dizisine LBP kodu denir. LBP kodu görüntüdeki kenarlar, köşeler, aydınlık veya karanlık bölgeler, çizgi bölgeleri gibi farklı özellikleri belirlemek için kullanılır. Pencerenin büyüklüğü farklı özellikleri yakalayabilmek için problemlere göre değiştirilebilir. LBP kodunun uzunluğu, Şekil 6.1'de görüldüğü gibi örnek piksel sayısı p ve simetrik dairesel komşuluğun yarıçapı r olmak üzere iki parametre ile dairesel tanımlanmaktadır. $LBP_{p,r}$ p tane komşu piksel setinden uygun 2^p farklı çıkış değeri tanımlar.

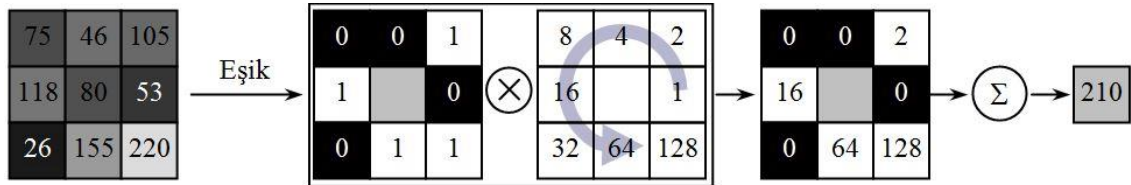


Şekil 6.1. Örnek LBP parametre değerleri.

LBP ile etiketlenen görüntünün histogramı Eşitlik 6.1 ile elde edilir. (x_c, y_c) verilen pikselin yerini, i_p ve i_c ise merkez pikselin gri seviyesini tanımlar.

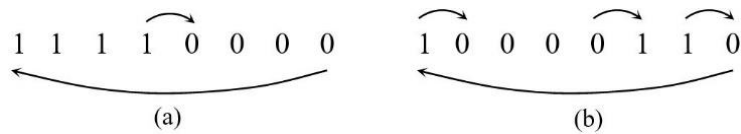
$$LBP_{p,r}(x_c, y_c) = \sum_{p=0}^{p-1} s(i_p, i_c) 2^p \quad s(x) = \begin{cases} 1, & \text{eğer } x \geq 0 \\ 0, & \text{eğer } x < 0 \end{cases} \quad (6.1)$$

LBP operatörünün kayan pencereler kullanarak öznitelik vektörü (histogram) çıkarma işlemi Şekil 6.2’de gösterilmiştir. Görüntüde kayan pencereler kullanılarak pencerenin merkez pikseli eşik değeri olarak belirlenir. Eşik değerinden büyük komşu piksel değerleri 1’e, küçük değerler ise 0’a atanır. Elde edilen değerler Şekil 6.2’deki gibi maske ile çarpılıp toplanılarak onluk tabandaki karşılığı bulunur.



Şekil 6.2. LBP kodunun üretilerek onluk tabana çevirmesi örneği.

Ojala ve diğerleri, [156,160], LBP kodlarını düzgün ve düzgün olmayan (uniform ve nonuniform) olmak üzere iki farklı sınıfa ayırmışlardır. Düzgün dağılıma sahip bir kod en fazla 2 defa farklı bit geçişine sahipken, düzgün dağılıma sahip olmayan bir kod 2’den fazla bit geçişine sahiptir. Düzgün ve düzgün olmayan LBP kodu örnekleri Şekil 6.3’te gösterilmiştir. Şekil 6.3(a), 2 geçişe sahip olduğundan düzgün LBP kodu; Şekil 6.3(b), 4 geçişe sahip olduğundan düzgün olmayan LBP kodudur.



Şekil 6.3. Uniform ve non uniform LBP kodu örnekleri.

6.2.2. Genetik Programlama Tanımlayıcı

GP-tanımlayıcı (GP-Descriptor), LBP algoritmasından esinlenerek Al-Sahaf ve diğerleri tarafından önerilen GP tabanlı görüntü tanımlayıcıdır [161]. GP-tanımlayıcı, LBP'ye benzer görüntü tanımlayıcısını otomatik olarak oluşturmayı amaçlar. Standart LBP'de olduğu gibi onluk tabandaki elde edilen değerin histogramdaki karşılık gelen bölümüne 1 arttırılır. Tanımlayıcıda uygunluk fonksiyonu, sınıflandırma başarısı ve mesafe olmak üzere iki bileşenden oluşur. Sınıflandırma başarısı, histogramları değerlendirirken k-NN sınıflandırıcı kullanarak eğitim örneklerinin doğru sınıflandırma yeteneğini ölçer. Mesafe işlevi, eğitim örneklerinde sınıf içi ve sınıflar arasında uzaklıkları hesaplar.

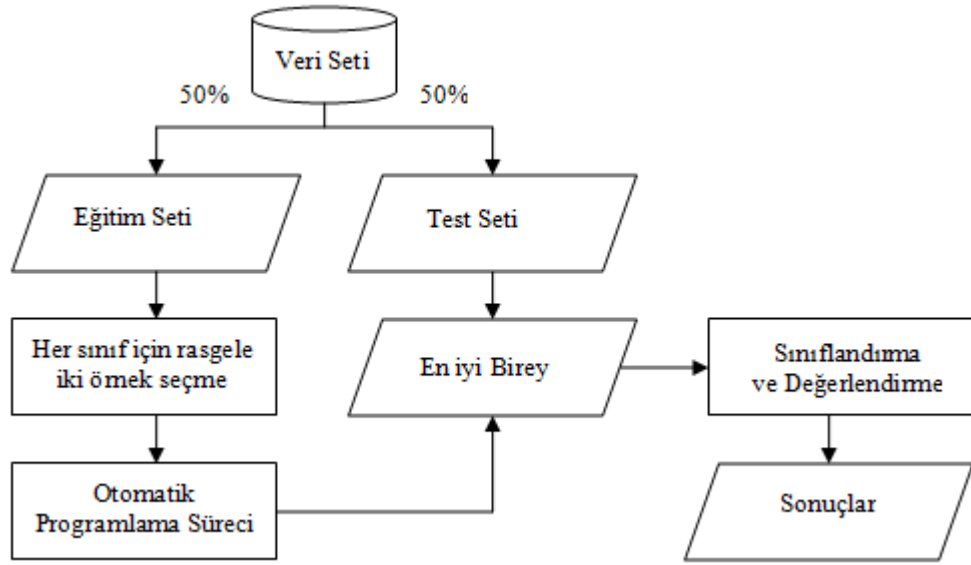
Bu bölümde gerçekleştirilen çalışmada [161]'de, önerilen GP-tanımlayıcıdan esinlenerek ilk kez ABCP-tanımlayıcı önerilmiş ve iki desen veri seti ile performansı izlenmiştir.

6.3. Önerilen Metot

Bu alt bölümde, ABCP tabanlı görüntü tanımlayıcının genel işleyişi, program yapısı, terminal ve fonksiyon kümeleri ile birlikte programın değerlendirilmesi hakkında bilgi verilmiştir.

6.3.1. Algoritmanın Genel İşleyişi

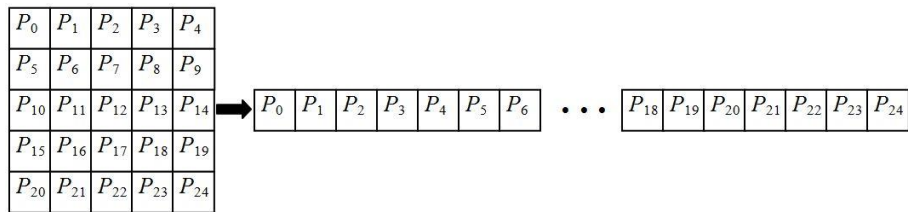
Veri seti %50 eğitim, %50 test olmak üzere iki parçaya rasgele seçilerek ayrılmıştır. Algoritma eğitim aşamasında her sınıftan iki örnek kullanarak ABCP'yi besler. Bu nedenle eğitim kümesinden her sınıftan rasgele iki örnek seçilir. Uygunluk fonksiyonu kullanılarak ABCP çözümleri iyileştirilir ve durdurma kriteri sağlanınca en iyi uygunluk değerine sahip olan çözüm ile ABCP modeli elde edilir. Test kümesindeki her bir örnek için bu model kullanılarak öznitelik vektörleri (histogramlar) elde edilir. Histogramlar kendi aralarında basit ve hızlı bir yöntem olan 1-NN ile sınıflandırılır. Sınıflandırma başarısı ile modelin performansı değerlendirilir. Algoritmanın aşamalarının detayları bölüm 6.3.2'de sunulmuştur. Tanımlayıcının adımlarını gösteren akış şeması Şekil 6.4'te verilmiştir.



Şekil 6.4. Algoritmanın genel işleyişi.

6.3.2. ABCP Descriptor Program İşleyişi

ABCP-tanımlayıcı, ham piksel değerlerini kullanarak model üreten ABCP tabanlı görüntü tanımlayıcıdır. Tanımlayıcı, belirli büyüklükte kayan pencerelerin içine düşen piksel değerlerini satır vektör olarak ABCP algoritmasına giriş olarak sunar. 5x5 pencere büyüklüğü kullanarak toplamda 25 piksel değerine sahip satır vektörler elde edilmiştir. Şekil 6.5'te satır vektörünün temsili gösterimi verilmiştir. Dolayısıyla ABCP'nin girişleri $\{P_0, P_1, \dots, P_{24}\}$ olmaktadır.



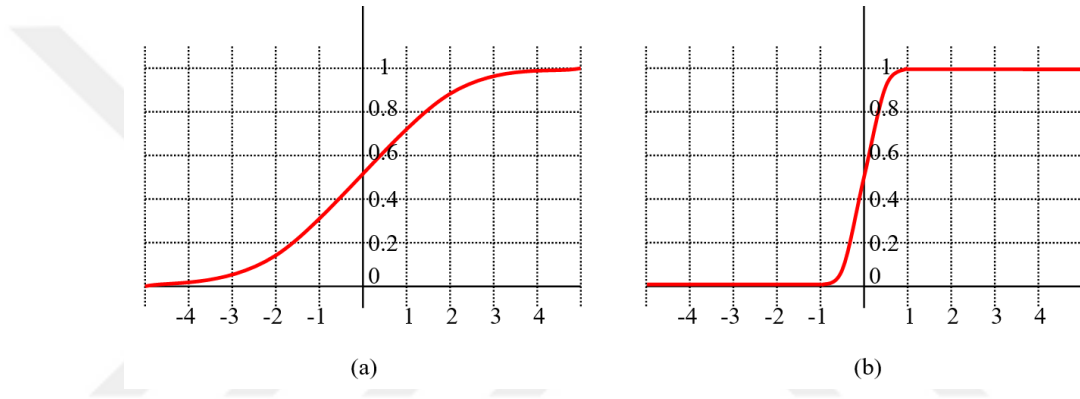
Şekil 6.5. Satır vektör temsili gösterimi.

ABCP fonksiyon kümesi basit aritmetik operatörlerden $\{+, -, *, /\}$ ve dönüştür başlangıç düğümünden oluşur. Aritmetik operatörlerden bölme fonksiyonu korumalı fonksiyondur. Payda 0 olduğunda sonuç 0 değeri döndürür. Dönüştür başlangıç düğümü ise satır vektörlerindeki girişleri kullanarak ikili kod oluşturur. Başlangıç düğümünün çocuk sayısı, üretilen kodun uzunluğunu ve aralığını belirler. Örneğin çocuk sayısı 5 ise ikili

Sınıflandırma doğruluğu (accuracy), doğru sınıflandırılan örnek sayısının toplam örnek sayısına oranı ile elde edilir. Doğruluk formülü Eşitlik 6.3'te gösterilmiştir.

$$Accuracy = \frac{N_{correct}}{N_{total}} \quad (6.3)$$

$N_{correct}$ doğru sınıflandırılan örnek sayısını, N_{total} ise toplam örnek sayısını tanımlar. Uzaklık (distance), sınıf içi ve sınıflar arası uzaklığı hesaplayan mantıksal fonksiyondur. Mantıksal fonksiyonun aralığı $[-5,5]$ aralığında olduğundan fonksiyona uyarlanarak $[-1,1]$ aralığına çekilmiştir. Mantıksal fonksiyonun gösterimi Şekil 6.7'de verilmiştir.



Şekil 6.7. Mantıksal fonksiyonlar.

Şekil 6.7'de gösterilen mantıksal fonksiyonlar a'nın formülü Eşitlik 6.4 ile b'nin formülü Eşitlik 6.5 ile aşağıda gösterilmiştir.

$$S(t) = \frac{1}{1+exp^{-t}} \quad (6.4)$$

$$S(t) = \frac{1}{1+exp^{-5t}} \quad (6.5)$$

Uzaklık fonksiyonu Eşitlik 6.6'da verilmiştir. Buna göre, D_b sınıflar arası örneklerin ortalama uzaklığını ve D_w sınıf içi örneklerin ortalama uzaklığını hesaplar. D_b ve D_w sırasıyla Eşitlik 6.7 ve Eşitlik 6.8'de tanımlanmıştır.

$$Distance = \frac{1}{1 + exp^{-5(D_b - D_w)}} \quad (6.6)$$

$$D_b = \frac{1}{z(z-n)} \sum_{\substack{u^\alpha, v^\beta \in S_{tr} \\ \forall \vec{u} \in u^\alpha \\ \forall \vec{v} \in v^\beta}} \chi^2(\vec{u}, \vec{v}), \alpha, \beta \in [1, c], \alpha \neq \beta \quad (6.7)$$

$$D_w = \frac{1}{z(n-1)} \sum_{\substack{u^\alpha, v^\beta \in S_{tr} \\ \forall \vec{u} \in u^\alpha \\ \forall \vec{v} \in v^\beta}} \chi^2(\vec{u}, \vec{v}) \alpha \in [1, c] \quad (6.8)$$

Burada; $S_{tr} = \{(\vec{x}_i, y_i)\}$ eğitim kümesinde seçilen örnek kümesini, $\vec{x}_i$ her varlığın öznitelik vektörünü, y_i vektörün sınıf etiketini tanımlar. $i \in \{1, \dots, z\}$; c ve n sırasıyla toplam sınıfların sayısı ve sınıf başına düşen örnek sayısını tanımlar. Eğitim kümesindeki toplam örnek sayısı $z(c*n)$; x^α S_{tr} ’de α . sınıfın tüm örneklerini ifade eder. Normalize edilmiş iki eşit eleman sayısına sahip vektörler arasındaki uzaklık için $\chi^2(\dots)$ formülü Eşitlik 6.9’daki gibi hesaplanır.

$$\chi^2 = \frac{1}{2} \sum_i \frac{(u_i - v_i)^2}{u_i + v_i} \quad (6.9)$$

Sırasıyla u_i ve v_i u ve v vektörlerinin i . elemanlarıdır. Eşitlik 6.9 fonksiyonu korunmuş versiyondur. Bu fonksiyonda bölen değer sıfıra eşitse sonuçta 0 döndürülmekte, diğer durumlarda normal bölme işlemi gerçekleştirilmektedir. Her iki uygunluk fonksiyonunda 1 değeri en kötü, 0 değeri en iyi olarak tanımlanabilir. Dolayısıyla otomatik programlama yöntemi için problem, minimizasyon problemi olarak tanımlanabilir.

Özet olarak, öznitelik vektörünün görüntü tanımlayıcıdan çıkarımı için aşağıdaki adımlar takip edilir:

- Adım1. Eğitim aşamasında rasgele seçilen örnekler önceden tanımlanan pencere büyüklüğünde (örneğin 5x5) satır vektörlere dönüştürülür. (Şekil 6.5)
- Adım2. Yaprak düğümler kök düğümü (dönüştür) besleyerek ağaçlardan dalların negatiflik durumu kontrol edilerek ikili kodlar elde edilir.
- Adım3. İkilik tabandaki kod onluk tabana çevrilerek elde edilen sayının karşılığı kadar ilgili varlığın histogramındaki karşılık değeri 1 arttırılır.

Adım4. Örneklerin histogramları kullanılarak:

- i. Aralarındaki mesafe hesaplanır.
- ii. Mesafeye göre sınıf ataması yapılır.
- iii. Sınıflar arasındaki (D_b), sınıf içi (D_w) uzaklıkların hesaplanır.
- iv. Ağaçların (bireylerin) uygunluk değerleri hesaplanır.

Adım5. Uygunluk değerlerine göre ABCP algoritması işletilir.

Vektörler arasındaki mesafe Eşitlik 6.9 kullanılarak Sözde Kod 1 ile aşağıdaki gibi elde edilmiştir.

fonksiyon mesafe_hesapla($\vec{u}, \vec{v}$)

mesafe değerini sıfırla (mesafe=0)

eleman sayısını ata (e=eleman_say)

for i =1: e (Her bir vektör elemanı için)

Eğer vektör elemanlarının toplamı 0 dan farklıysa

Mesafeye hesaplanan mesafe değerini ekle ($\text{mesafe} = \text{mesafe} + \left(\frac{(u_i - v_i)^2}{u_i + v_i} \right)$)

end (Eğer)

end (for)

Mesafe değerini elde et ($\text{mesafe} = 0.5 * \text{mesafe}$)

return mesafe

end (fonksiyon)

Şekil 6.8. İki vektör arasındaki mesafenin hesaplanması (sözde kod 1).

Sözde kod 1 ile örneklerin histogramları arasında elde edilen mesafe değerleri hesaplanır, histogramlar arasındaki mesafeye göre sınıf değeri atama, sınıf içi / sınıflar arası olma durumuna göre uzaklık değerleri (sınıflar arası (D_b) ve sınıf içi (D_w)) Sözde kod 2'deki gibi hesaplanır.

Fonksiyon uzaklık_hesapla ($\vec{S}, c, n$) $\vec{S}$ eğitim kümesinde seçilen örnek kümesi,
 c toplam sınıfların sayısı,
 n sınıf başına düşen örnek sayısı

z değerini ata ($z=c*n$)

for each (x_i, c_i) (Her sınıfın i. vektörü için)

enkucuk değerini ata($enkucuk=inf$)

i.vektöre tahmin edilen sınıf değerini ata($pred_class_i=0$)

for each (x_j, c_j) (Her sınıfın j. vektörü için)

Vektörler arasındaki mesafeyi hesapla ($dis=mesafe_hesapla$
 (x_i, x_j))

Eğer ($i \neq j \ \&\& \ dis < \text{enkucuk}$) (Aynı vektör değilse ve mesafe
enkucukten küçükse)

enkucuk'e mesafeyi ata($enkucuk=dis$)

x_i 'nin sınıfı j.nin sınıfına ata ($pred_class_i = class_j$)

end

Eğer ($c_i \neq c_j$) (Vektörlerin sınıfları farklıysa)

Sınıflar arasına mesafe ekle ($D_b = D_b + dis$)

Değilse

Sınıf içine mesafe ekle ($D_w = D_w + dis$)

end (*eğer* $c_i \neq c_j$)

end (for each (x_j, c_j))

end (for each (x_i, c_i))

Sınıflar arası uzaklık değerini ata $D_b = D_b / (z * (z - n))$

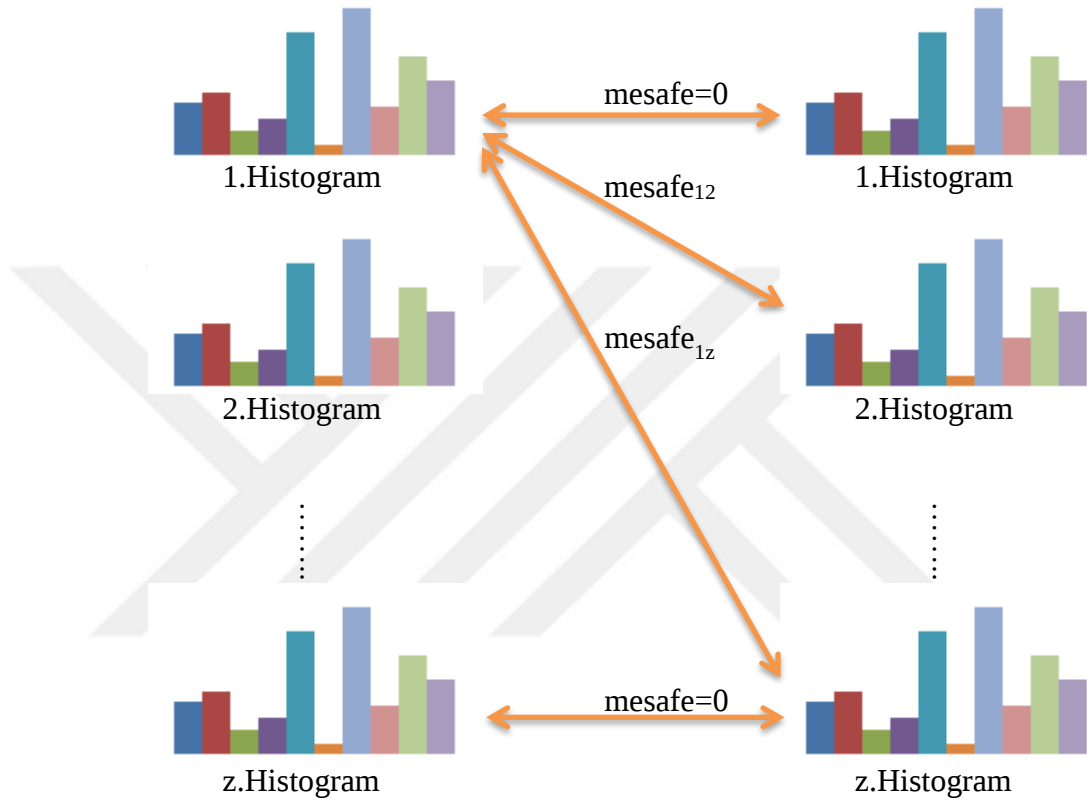
Sınıf içi uzaklık değerini ata $D_w = D_w / (z * (n - 1))$

return (D_b, D_w)

end (fonksiyon)

Şekil 6.9. Sınıflar arası ve sınıf içi uzaklığın hesaplanması (sözde kod 2).

ABCP-tanımlayıcıda ağaçların uygunluk değerlerine göre iyileştirmesi beklenir. Durdurma kriteri sağlanınca en iyi birey ile test verileri sınıflandırılır. Sınıf ataması yapılırken her bir histogramın arasındaki mesafenin karşılaştırılması temsili gösterimi Şekil 6.10'da verilmiştir.



Şekil 6.10. Histogramlar arasında mesafenin karşılaştırılması

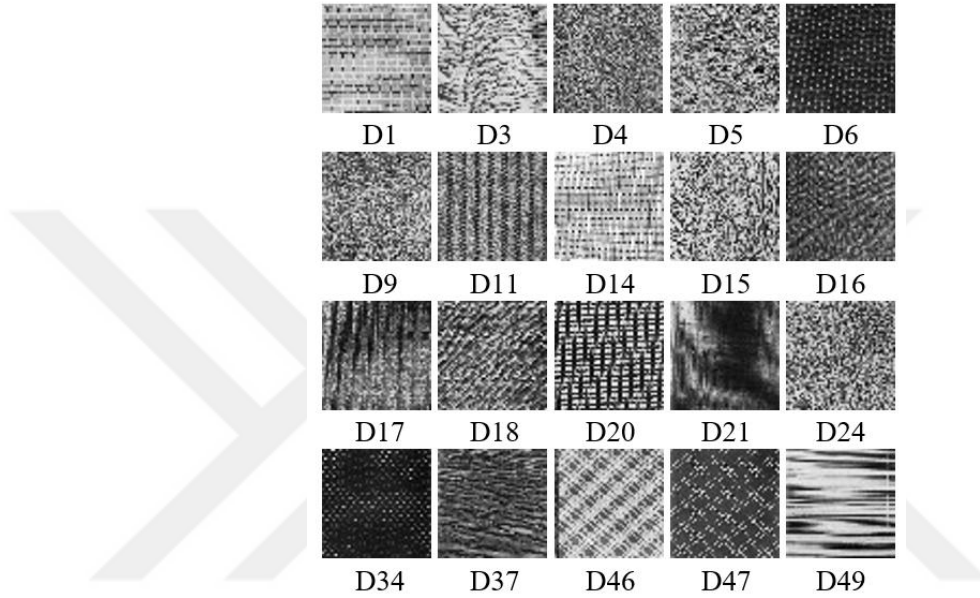
6.4. Deney Çalışmaları

Önerilen metodun performans analizi için iki farklı veri seti kullanılarak deneyler yürütülmüş ve en iyi bilinen iki farklı görüntü tanımlayıcı ile sonuçlar kıyaslanmıştır. Bu bölümde veri setleri ve otomatik programlama yöntemlerinde kullanılan parametreler hakkında detaylı bilgi verilmiştir.

6.4.1. Veri Setleri

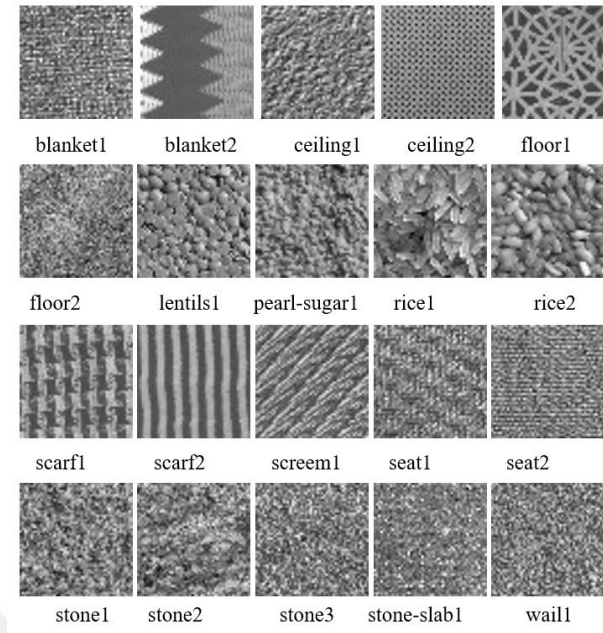
Yöntemler, görüntü işlemede yaygın olarak kullanılan iki farklı desen seti üzerinde (DS-Broadtz ve DS-Kylberg) test edilmiştir.

İlk veri seti DS-Broadtz, Broadtz desen setinden [162] elde edilmiştir. Desen setinin orijinali, her biri 640x640 piksel olmak üzere 112 sınıftan oluşmaktadır. Çalışmada, 112 sınıftan [161]'de seçilen 20 sınıf seçilerek karşılaştırma yapılması amaçlanmıştır. Seçilen 20 sınıfa ait desen örnekleri Şekil 6.11'de gösterilmiştir. Her sınıfa ait desenler, 64x64 piksel örneklere bölünerek toplamda 100 örnek elde edilmiş olur.



Şekil 6.11. DS-Broadtz veri seti sınıf örnekleri.

İkinci veri seti DS-Kylberg, Kylberg veri setinden [163] elde edilmiştir. Kylberg veri seti orijinalinde döndürülmüş ve döndürülmemiş olmak üzere iki kısma ayrılır. Çalışmada 28 sınıfa sahip olan döndürülmemiş veri setinden [161]'de 20 sınıf seçilerek kullanılmıştır. Her bir sınıf 576x576 piksellik 160 örnek içermektedir. Çalışmada hesaplama maliyetini düşürmek adına her bir örnek 115x115 piksele düşürülmüştür. Seçilen Kylberg veri setine ait desen örnekleri Şekil 6.12'de gösterilmiştir.



Şekil 6.12. DS- Kylberg veri seti sınıf örnekleri.

Tablo 6.1’de Broadtz ve Kylberg veri setlerinden toplam seçilen sınıf sayısı ($N_{classes}$), her sınıfın içerdiği örnek sayısı ($N_{instances}$), eğitimde ve testte kullanılan örnek sayısı (N_{train_in} , N_{test_in}), örnek boyutları (yükseklik x genişlik) özetlenmiştir.

Tablo 6.1. Veri setleri.

Veri Seti	$N_{classes}$	$N_{instances}$	N_{train_in}	N_{test_in}	Örnek Boyutu
Broadtz	20	100	50	50	64*64
Kylberg	20	160	80	80	115*115

Tablo 6.1’de görüldüğü üzere Broadtz, toplamda 1000 örnek (20 sınıf x 50 örnek); Kylberg, toplamda 1600 örnek (20 sınıf x 80 örnek) eğitim ve test için kullanmıştır.

6.4.2. Kontrol Parametreleri

Otomatik programlama yöntemlerinin tüm koşullarda kullandıkları parametre değerleri Tablo 6.2’de gösterilmiştir. Tüm yöntemlerde başlangıç çözümleri “*ramped half and half*” metoduyla üretilmiştir. ABCP’de, kaynakların kalitesini arttırmak için tüm

çözümlerde Bilgi Paylaşım Mekanizmasından yararlanılmıştır. Başlangıç düğümü olan “Dönüştür” düğümüne bağlanan alt ağaç sayısı/sahip olduğu dal sayısı 7 olarak belirlenmiştir. Böylece histogram $2^7=128$ farklı değer $(0,1,...,127)$ alabilmektedir. Programda yöntemlerin dal sayısı sabit tutulmuştur. Fonksiyon kümesindeki bölme (/) fonksiyonu korunmuş fonksiyondur. Bu fonksiyonda da bölen değer 0’a eşitse sonuçta 0 döndürülmekte, diğer durumlarda normal bölme işlemi gerçekleştirilmektedir. ABCP-tanımlayıcıda, yiyecek kaynaklarının tükenme durumunu kontrol eden limit parametre değeri koloni büyüklüğünün yarısı alınmıştır. ABCP algoritmasında limit değeri aşıldıysa yiyecek kaynağı tükenmiş kabul edilerek kaynak terk edilir. Terk edilen kaynağın görevli arısı kâşif arıya dönüşerek rastgele üretilen yeni bir kaynakla çalışmasına devam eder.

Tablo 6.2. Deneylerde Kullanılan Kontrol Parametreleri.

Parametreler	ABCP-Tanımlayıcı
Koloni Büyüklüğü	100
İterasyon Sayısı	30
Maksimum Ağaç Derinliği	6
Dönüştür’ün Sahip Olduğu Dal Sayısı	7
Başlangıç	Ramped Half and Half
Kullanılan Fonksiyonlar	+, -, *, / (korunmalı)
Limit	50

6.6. Sonuçlar ve Tartışmalar

Yöntemlerin deney sonuçları bu alt bölümde verilmiş ve tartışılmıştır. ABCP-Tanımlayıcı ve LBP_{8,1} 10 kez birbirinden bağımsız koşturulmuştur. Yöntemler, her koşmada veri setlerinden rastgele seçilen örnekleri analiz ederek test örneklerini doğru desen sınıfına atamaya çalışmıştır. İki farklı desen seti için önerdiğimiz ABCP-Tanımlayıcı, eğitim kümesini öğrenme kabiliyeti, ilgili/gerekli öz nitelikleri seçme konusunda GP-Tanımlayıcı ve LBP_{8,1} algoritmalarıyla karşılaştırılarak değerlendirilmiştir.

6.6.1. Genel Sonuçlar

ABCP-Tanımlayıcı, LBP_{8,1} ve GP-tanımlayıcı [161] için her iki veri setine ait simülasyon sonuçları Tablo 6.3'te sunulmuştur. Tablo 6.3'te veri setlerinde her ölçüt için en iyi test sınıflandırma başarısına sahip yöntem koyu renk ile belirtilmiştir. [161]'de GP-tanımlayıcının makine öğrenmesinde yaygın olarak kullanılan dokuz klasik yöntemle karşı performansı test edilmiştir. Yöntemlerin performansı GP-tanımlayıcı ile karşılaştırıldığında oldukça düşüktür. Bu nedenle, klasik yöntemlerden daha yüksek bir sonuç veren GP-tanımlayıcıya rakip olabilecek yeni bir yöntem olan ABCP-tanımlayıcı önerilerek sonuçları GP-tanımlayıcının sonuçları [161] ile karşılaştırılmıştır.

Tablo 6.3. Tanımlayıcıların simülasyon sonuçları.

Tanımlayıcı	Veri Seti	Kriter	En İyi Test Değeri	Test Sınıflandırma Başarısı (%)
ABCP-tanımlayıcı	Broadtz	Ortalama	0.03	94.55
		Maksimum	0.03	97.3
		Minimum	0.01	93.3
		Standart Sapma	0.01	1.17
	Kylberg	Ortalama	0.05	89.71
		Maksimum	0.14	95.19
		Minimum	0.02	71.63
		Standart Sapma	0.03	6.38
LBP _{8,1}	Broadtz	Ortalama		90.63
		Maksimum		93.8
		Minimum		87.3
		Standart Sapma		1.92
	Kylberg	Ortalama		90
		Maksimum		93.13
		Minimum		87.13
		Standart Sapma		1.9
GP-tanımlayıcı [161]	Broadtz	Ortalama		94.40
		Standart Sapma		0.81
	Kylberg	Ortalama		93.21
		Standart Sapma		1.14

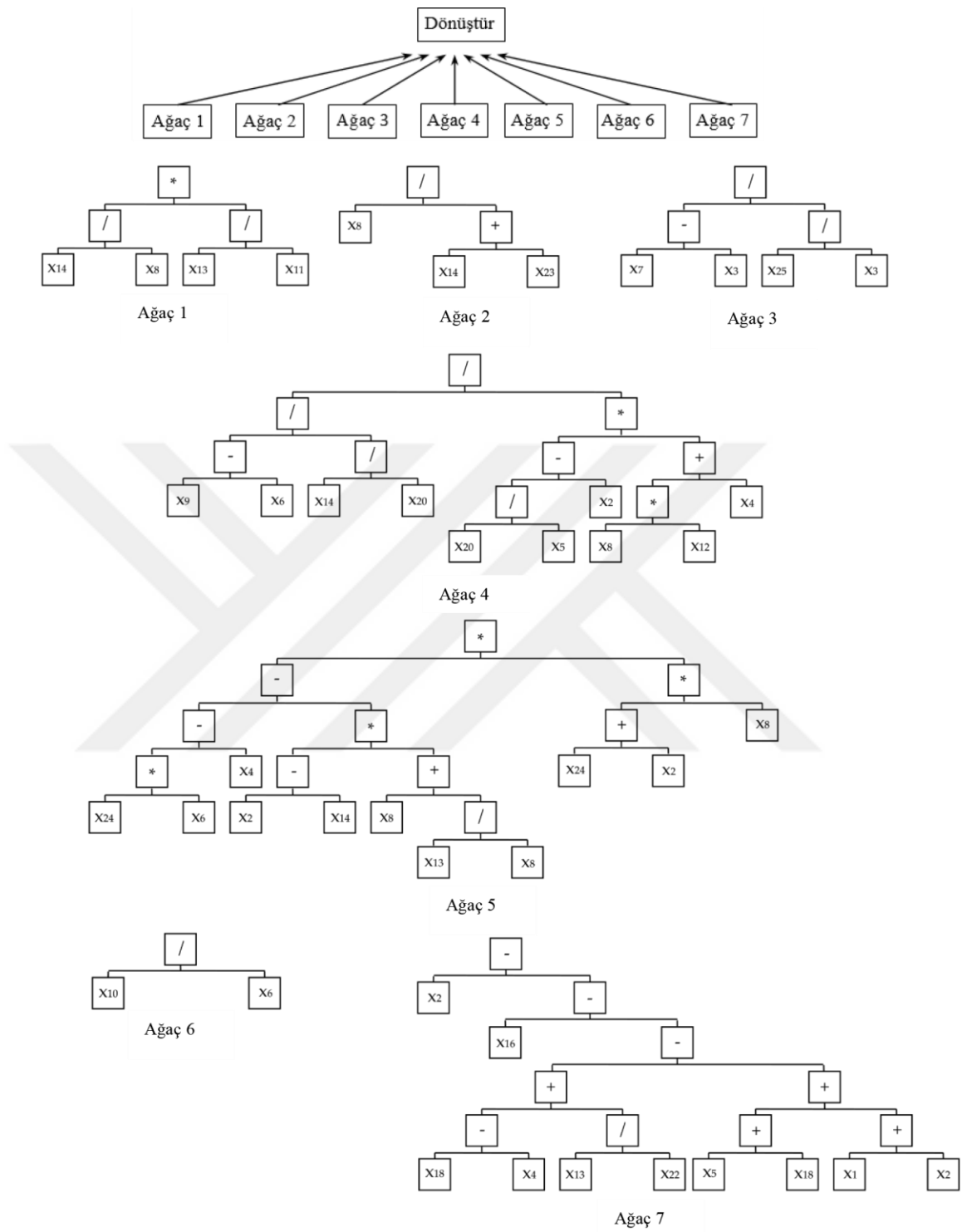
Sonuçlar değerlendirildiğinde ABCP tanımlayıcı ve LBP ile oldukça başarılı sonuçlar elde edilmiştir. Tanımlayıcıların, Broadtz veri setinde ortalama test başarıları Kylberg veri setine göre daha fazladır. En yüksek test doğruluğu Broadtz veri seti için %97.3 ve Kylberg veri seti için %96.188 olarak ABCP-Tanımlayıcıda bulunmuştur. Aynı zamanda ABCP-tanımlayıcı Kylberg veri setinde 6.38 ile en yüksek standart sapma değerine sahiptir. Sapmanın yüksek olması LBP'ye göre maksimum ile minimum test doğrulukları arasındaki farkın daha fazla olduğunu gösterir. GP-tanımlayıcı [161], %93.21 ile Kylberg veri setinde en yüksek ortalama başarıya sahiptir. ABCP-tanımlayıcının sınıflandırma sonuçlarının yüksek olması, GP-tanımlayıcı ve LBP'ye alternatif bir yöntem olduğunu göstermektedir.

6.6.2. Model Analiz Sonuçları

Bu alt bölümde, her veri seti için ABCP-tanımlayıcı ile en iyi test doğruluğuna sahip modeller analiz edilmiştir. Tablo 6.4, Broadtz'da %97.3 ile en yüksek test sınıflandırmasına sahip modelin alt ağaçları hakkında genel bilgiler gösterilmiştir. Tablo 6.4'teki karmaşıklık Eşitlik 2.9'a göre hesaplanmıştır. Tabloda görüldüğü üzere "Dönüştür" düğümüne bağlanan alt ağaçlar farklı derinlik ve düğüm sayısına bağlıdır. Broadtz veri setinin en iyi modelinin ağaçlarının temsili Şekil 6.13'de gösterilmektedir. Şekil 6.13'deki en iyi modelin alt ağaçlarının eşitlikleri Tablo 6.5'te gösterilmiştir.

Tablo 6.4. DS-Broadtz için ABCP-tanımlayıcının en iyi modelinin alt ağaçlarının özellikleri.

	Düğüm	Derinlik	Karmaşıklık
Ağaç 1	7	3	17
Ağaç 2	5	3	11
Ağaç 3	7	3	17
Ağaç 4	19	5	69
Ağaç 5	21	6	83
Ağaç 6	3	2	5
Ağaç 7	19	6	87



Şekil 6.13. DS-Broadtz’da ABCP-Tanımlayıcının en iyi modeli.

Tablo 6.5. DS-Broadtz'da ABCP-Tanımlayıcı en iyi modelin alt ağaçlarının eşitlikleri.

	Eşitlik
Ağaç 1	$(x_{14}/x_8) * (x_{13}/x_{11})$
Ağaç 2	$x_8/(x_{14} + x_{23})$
Ağaç 3	$(x_7 - x_3)/(x_{25}/x_8)$
Ağaç 4	$((x_9 - x_6)/(x_{14}/x_{20}))/((\left(\frac{x_{20}}{x_5}\right) - x_2) * ((x_8 * x_{12}) + x_4))$
Ağaç 5	$((x_{24} * x_6) - x_4) - \left((x_2 - x_{14}) * \left(x_8 + \left(\frac{x_{13}}{x_8} \right) \right) \right) * ((x_{24} + x_2) * x_8)$
Ağaç 6	(x_{10}/x_6)
Ağaç 7	$x_{16} - \left((x_{18} - x_4) + \left(\frac{x_{18}}{x_{22}} \right) \right) - ((x_5 + x_{18}) + (x_1 + x_2))$

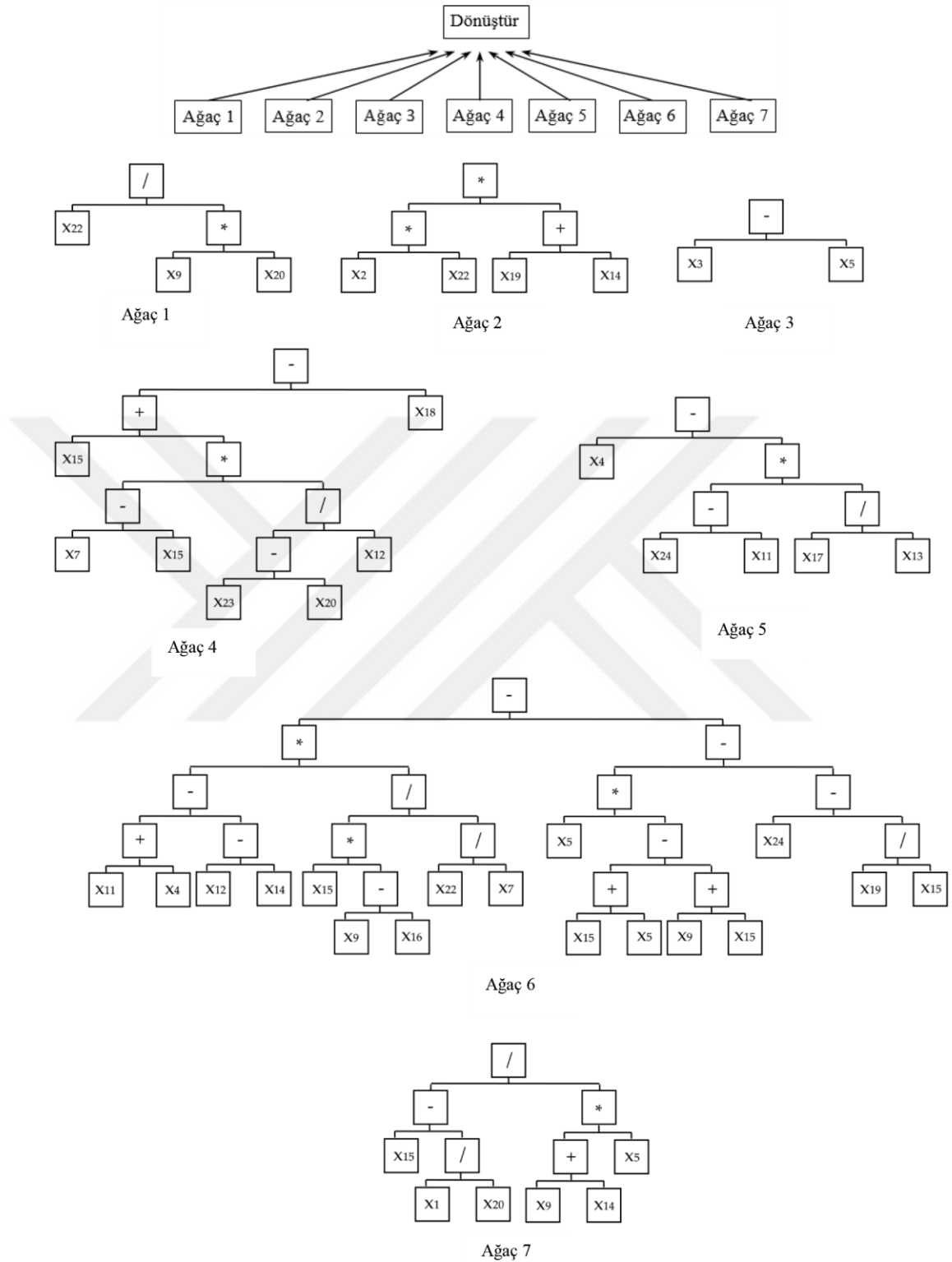
Tablo 6.6, Kylberg'de en yüksek test sınıflandırmasına sahip modelin alt ağaçları hakkında genel bilgileri göstermektedir. Kylberg'de, veri setinin en iyi modelinin ağaçlarının temsili Şekil 6.14'te gösterilmektedir. Şekil 6.14'teki en iyi modelin alt ağaçlarının eşitlikleri, Tablo 6.7'de gösterilmiştir. Modeller analiz edildiğinde, Kylberg modelinin Broadtz modelinden daha karmaşık olduğu görülmüştür. Kylberg modelinde, alt ağaçlardaki birçok giriş (x_{14} , x_{15} , x_{20} gibi), elde edilen modellerde ortaktır. Her iki veri seti için en iyi modellerin dönüştür düğümü 7 alt ağaçtan oluşmaktadır ve ağaçların derinliği 2-6 arasındadır. Düğüm sayısı ve ağacın derinliği arttıkça, ağacın karmaşıklığı artmıştır.

Tablo 6.6. DS-Kylberg için ABCP-Tanımlayıcı en iyi model.

	Düğüm	Derinlik	Karmaşıklık
Ağaç 1	5	3	11
Ağaç 2	7	3	17
Ağaç 3	3	2	5
Ağaç 4	13	6	51
Ağaç 5	9	4	27
Ağaç 6	33	6	145
Ağaç 7	11	4	33

Tablo 6.7. DS-Kylberg’de ABCP-tanımlayıcı en iyi modelin alt ağaçlarının eşitlikleri.

	Eşitlik
Ağaç 1	$x_{22}/(x_9 * x_{20})$
Ağaç 2	$(x_2 * x_{22}) * (x_{19} + x_{14})$
Ağaç 3	$(x_3 - x_5)$
Ağaç 4	$\left(x_{15} + (x_7 - x_{15}) * \left(\frac{x_{23} - x_{20}}{x_{12}} \right) \right) - x_{18}$
Ağaç 5	$x_4 - ((x_{24} - x_{11}) * \left(\frac{x_{17}}{x_{13}} \right))$
Ağaç 6	$\left(((x_{11} + x_{14}) - (x_{12} - x_{14})) * \left(\frac{x_{15} * (x_9 - x_{16})}{\frac{x_{22}}{x_7}} \right) \right) - ((x_5$ $* ((x_{15} + x_5) - (x_9 + x_{15})) - \left(x_{24} - \left(\frac{x_{19}}{x_{15}} \right) \right))$
Ağaç 7	$(x_{15} - (x_1/x_{20}))/((x_9 + x_{14}) * x_5)$



Şekil 6.14. DS-Kylberg’de ABCP-Tanımlayıcının en iyi modeli.

7. BÖLÜM

TARTIŞMA, SONUÇ VE ÖNERİLER

7.1. Tartışma ve Sonuç

Bu tez çalışması kapsamında, sembolik regresyon problemi için yeni geliştirilen ABCP otomatik programlama yöntemi, farklı mühendislik problemlerine uygulanarak yöntemin başarısı test edilmiştir. Tez kapsamında, ABCP algoritmasına dayalı Çok Kovanlı ABCP (MHABCP) otomatik programlama yöntemi ilk kez geliştirilmiş ve yüksek boyutlu sembolik regresyon probleminin çözümünde kullanılmıştır. ABCP ve MHABCP test edilirken kendi aralarında ve literatürde yer alan GP, GP tabanlı versiyonlar ve literatürde bilinen rakip yöntemlerle kıyaslanmıştır. Tezde öncelikle, GP ve ABCP yöntemleri hakkında detaylı bilgi verilmiştir. ABCP'nin ilk olarak, Cherkassy fonksiyonunun ve pH veri setindeki çıkış değerlerinin tahmininde ve gürültü eklenmiş beton basınç dayanımı veri setinde öznitelik seçme kabiliyeti araştırılarak sonuçları GP ile kıyaslanmıştır. Bahsedilen üç problemde de ABCP'nin GP'den daha iyi eğitim performansına sahip olduğu görülmüştür. Tüm sonuçlarda GP'nin standart sapması ABCP'ye nazaran yüksek çıkmıştır. “Gürültü eklenmiş beton basınç dayanımı veri setinde, diğer iki probleme kıyasla her iki yöntemde de daha yüksek hata oranına sahiptir”. ABCP tarafından çıkarılan modellerin tahmini ve gerçek çıkış değerleri izlendiğinde en çok uyumun Cherkassy fonksiyonunda, en az uyumun ise beton basınç dayanımı veri setinde olduğu gözlemlenmiştir. Ayrıca, beton basınç dayanımı veri setine eklenen 50 gürültü parametresinden sadece x_{17} gürültü parametresi ABCP modelinde yer almaktadır. Gürültü parametrelerinden sadece 1 tanesinin modelde bulunması ve diğer gürültü parametrelerinin elimine edilmesi ABCP'nin öznitelik seçiminde kabiliyetli olduğunun bir göstergesidir.

Tezde dördüncü bölümde yapılan çalışmalarda, beton basınç dayanımı veri setinde ABCP'nin öznitelik seçimi konusunda başarılı sonuçlar elde etmesiyle, tezin üçüncü bölümünde algoritmanın, giriş sayıları yüksek veri setleri üzerinde öznitelik seçimi ve sınıflandırma etkinliğinin gözlemlenmesi amaçlanmıştır. Deney çalışmalarında, sahip oldukları giriş sayıları 13 ile 34, sınıf sayıları 2 ile 6 arasında değişen; Wisconsin meme kanseri (WDBC), dermatoloji, şarap, at kolik olarak adlandırılan 4 adet gerçek veri setinden yararlanılmıştır. ABCP ve GP'nin doğru sınıflandırma yapabilmesi için her sınıfa eşit önem veren Hassas Uygunluk Fonksiyonu (sensitivity fitness function-SFF) kullanılmıştır. Veri setlerinin büyüklüğüne göre yöntemlerin koloni/popülasyon büyüklüğü arttırılmıştır. Sonuçlarda, hata oranına en çok at kolik veri setinde rastlanmıştır. WDBC veri setinde yöntemler arasında anlamlı bir fark gözlemlenmemiştir. Şarap veri setinde her iki yöntemin de test başarısı %100'e ulaşmıştır. Her iki yöntem de klasik yöntemle yapılan çalışma [120] ile benzer sınıflandırma sonuçlarına sahiptir. Yöntemlerin çıkardığı girişler izlendiğinde en fazla dermatoloji veri setinde, en az şarap veri setinde ortak girişe rastlanmıştır. Genel olarak veri setinin büyüklüğü arttıkça modellerin karmaşıklığı artmıştır. Tüm koşullarda çıkarılan en iyi sonuçlar değerlendirildiğinde GP ve ABCP, 3 ile 8 arasında değişen ortak giriş sayısına sahip modeller çıkarmıştır.

Tezin beşinci bölümünde, ABCP'nin performansını geliştirmeye yönelik MHABCP olarak adlandırılan yeni bir algoritma önerilmiştir. Algoritmada çözümlerin iyileştirilmesinde ABCP'deki bilgi paylaşım mekanizmasının kullanılması ile birlikte Çok Genli Genetik Programlama'da kullanılan iki noktalı yüksek seviyeli çaprazlama operatörünün komşuluk araştırma işlemine adaptasyonu gerçekleştirilmiştir. Böylece ABCP'ye paralellik kazandırılarak optimum çözüme daha hızlı ulaşılması hedeflenmiştir. MHABCP'de her çözüm, her bir ağacın bir kovan olarak düşünüldüğü bir dizi ağacın lineer bir birleşimi olarak tasarlanmıştır. Algoritmanın, iki yapay ve dört gerçek olmak üzere toplamda altı yüksek boyutlu veri seti kullanılarak GP ve ABCP'ye kıyasla performansı analiz edilmiştir. Kullanılan yapay veri setleride F_1 fonksiyonu Newton'un yerçekimi kanunu, F_2 fonksiyonu [135]'teki çalışmadan alınan fonksiyondur. Yapay veri setlerine 50 gürültü parametresi eklenmiştir. UCI'dan [109] alınan gerçek veri setlerinin sahip oldukları giriş sayıları 100 ile 7399 arasında değişmektedir. LD50, DLBCL, CCUN ve CCN deneylerde kullanılan gerçek veri setleridir. Yöntemlerin performansı, her biri

100 iterasyona sahip birbirinden bağımsız 100 koşma ile izlenmiştir. Deneylerde, diğer yöntemlere nazaran GP'nin hata oran dağılımları daha geniş ve standart sapmaları daha yüksek çıkmıştır. Eğitim hata oranlarına bakıldığında MHABCP, CCUN ve CNN veri setleri hariç diğer yöntemlere kıyasla daha başarılı sonuçlar çıkarmıştır. F_1 , F_2 ve DLBCL veri setlerinde, ABCP GP'den daha düşük eğitim hata oranına sahiptir. Deneylerde, eğitimdeki tüm koşmaların iterasyonlarındaki minimum hata oranına bakılarak yöntemlerin evrimsel gelişim grafiği çizdirilmiştir. Grafikler detaylı incelendiğinde MHABCP'nin, DLBCL dışındaki tüm veri setlerinde, minimum hata oranına diğer yöntemlerden daha hızlı ulaştığı görülür. ABCP, tüm veri setlerinde en iyi sonuca en yavaş ulaşan yöntemdir. Test hata oranlarının grafiklerine bakıldığında yöntemler benzer minimum test hata oranları elde etmiştir. En yüksek test hata oranı LD50'de elde edilmiştir. GP tüm veri setlerinde test hata oran aralığı en geniş olan yöntemdir. Yöntemlerin, her koşmada en iyi modellerinde bulunan girişler analiz edildiğinde her üç yöntemin de en fazla CCN veri setinde, en az LD50 veri setinde ortak öznetelik çıkardığı görülmüştür. F_1 ve F_2 'nin tüm koşmalarının, MHABCP tarafından çıkarılan en iyi modellerinde sadece x_1 , x_2 , x_3 girişleri bulunması yöntemin gürültü parametrelerini elimine ettiğini göstermektedir. F_1 modeli, F_1 'in gerçek eşitliğine oldukça yakındır. Deneylerde eğitim hatası ve karşılık gelen test hatalarına bakıldığında MHABCP'nin, F_1 'de eğitim ve testte en az hata oranına sahip yöntem olduğu görülmüştür. F_2 'de, ortalamada en az hata ABCP'de bulunmuştur, GP ve MHABCP'nin sonuçları birbirine çok yakındır. LD50 ve DLBCL'de tüm yöntemler için ortalama hata değerleri birbirine oldukça yakındır. Genel olarak GP'nin tüm veri setlerinde, minimum ve ortanca değerlerinde ABCP'den daha başarılı olduğu ancak standart sapmasının diğer yöntemlerden yüksek olduğu görülmektedir. Deneylerin tüm sonuçları ve anlamlılık testinin sonuçlarıyla, MHABCP'nin yüksek boyutlu sembolik regresyon problemini başarılı bir şekilde çözdüğü ve başka mühendislik problemlerini çözebileceği sonucuna varılmıştır.

Altıncı bölümde, resimlerdeki belirgin noktaları çıkararak desenleri sınıflandırmaya çalışan ABCP-tanımlayıcı ilk kez önerilmiştir. ABCP-tanımlayıcının işleyişi GP-tanımlayıcı algoritmasının işleyişinden ilham alınarak düzenlenmiştir. Çalışmada, her desen seti için rastgele iki örnek seçilerek ABCP-tanımlayıcı eğitilmiştir. Örnekler, boyutu 5x5 olan kayan pencerelerin içine düşen piksel değerlerini satır vektör olarak

ABCP-tanımlayıcıya sunmaktadır. Tanımlayıcıda kullanılan dönüştür başlangıç düğümü ise satır vektörlerindeki girişleri kullanarak ikili kodlar elde etmiştir. İkili kodlar onluk tabana çevrilerek karşılığındaki öznitelik vektöründeki bölmelerin değerleri arttırılmıştır. Örneklerin eğitiminde, algoritmanın ezberlemesinden kaçınmak için mesafe ve sınıflandırma doğruluğu eşit öneme sahip olan uygunluk fonksiyonu kullanılmıştır. Deneylerde ABCP-tanımlayıcının sonuçları GP-tanımlayıcı ve LBP ile kıyaslanmıştır. Yöntemler sık kullanılan iki farklı desen seti (DS-Broadtz ve DS-Kylberg) ile test edilmiştir. Al-Sahaf ve diğerleri, GP-tanımlayıcı ile yaygın olarak kullanılan dokuz klasik yöntemle kıyaslayarak tanımlayıcının klasik yöntemlere nazaran daha başarılı olduğunu göstermişlerdir [161]. Bu nedenle bölümdeki çalışmada, klasik yöntemlerle kıyaslama sunulmamıştır. Simülasyon sonuçları değerlendirildiğinde; test sınıflandırma doğruluğu, DS-Broadtz veri setinde en yüksek ABCP-tanımlayıcıda, DS-Kylberg veri setinde en yüksek GP-tanımlayıcıda elde edilmiştir. Ayrıca tanımlayıcıların, Broadtz veri setinde ortalama test başarısı Kylberg veri setine göre daha fazladır. ABCP-tanımlayıcı tarafından çıkarılan modeller analiz edildiğinde dönüştür düğümünün alt ağaçlarında ortak girişlerin kullanıldığı görülmüştür. Görüntü sınıflandırma problemlerinde ABCP-tanımlayıcı kullanılarak yüksek sınıflandırma doğruluğu elde edilmesi ile bu alanda kullanılabilecek bir yöntem olduğu gösterilmiştir.

7.2. Öneriler

Gelecek çalışmalarda ABCP ve MHABCP yöntemleri kullanılarak farklı gerçek dünya problemlerinde performansları değerlendirilebilir. Ayrıca ABCP-tanımlayıcının, görüntü sınıflandırma problemlerinde rotasyon duyarlı desenlerdeki başarısı test edilebilir. Yöntemlerin hesaplama maliyetini, işlem zamanını vb. parametrelerini iyileştirmek için çalışmalar yapılabilir. Bahsedilen yöntemlerin uygulanabilirliğini arttırabilmek adına yöntemler bir araç kutusu olarak tasarlanabilir. Böylece programlama bilgisi orta düzeyde olan kullanıcıların, parametreleri ayarlayarak otomatik programlama yöntemlerini hızlı ve etkin bir şekilde kullanması sağlanabilir.

KAYNAKÇA

1. Biermann, A., 1985. Automatic programming: A tutorial on formal methodologies. **J. Symbolic Computation**, **1** (2): 119-142.
2. Cramer, N. L., 1985. A representation for the adaptive generation of simple sequential program, pp. 183-187. *1st International Conference on Genetic Algorithms and Their Applications*, July 24-26, 1985, Erlbaum.
3. Koza, J. R., 1992. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, 819 pp.
4. Roux, O., Fonlupt, C., 2000. Ant programming: Or how to use ants for automatic programming. *ANTS'2000 From Ant Colonies to Artificial Ants: 2nd International Workshop on Ant Algorithms*, September 8-9, 2000, Brussels, Belgium.
5. Johnson, C. G., 2003. Artificial immune system programming for symbolic regression, pp. 350-358. *6th European Conference on Genetic Programming (EuroGP 2003)*, April 14-16, 2003, Univ Essex, Colchester, England.
6. Gan, Z., Chow, T.W.S., Chau, W.N., 2009. Clone selection programming and its application to symbolic regression. **Expert Systems with Applications**, **36** (2): 3996-4005.
7. Karaboga, D., Ozturk, C., Karaboga, N., Gorkemli, B., 2012. Artificial bee colony programming for symbolic regression. **Information Sciences**, **209**: 1-15.
8. Arslan, S., Ozturk, C., 2019. Multi hive artificial bee colony programming for high dimensional symbolic regression with feature selection. **Applied Soft Computing Journal**, **78**: 515–527.
9. Koza, J.R., 2003. Preparatory Steps of Genetic Programming. (Web page: <http://www.genetic-programming.com/gppreparatory.html>), (Date accessed: May 2019).
10. Poli R., Langdon W. B., McPhee N. F., 2008. A field guide to genetic programming. (Web page: <http://cswww.essex.ac.uk/staff/rpoli/gp-field-guide>).
11. Karaboga, D., 2006. Yapay Zeka Optimizasyon Algoritmaları. Nobel Akademik Yayıncılık, Çankaya, 246 s.

12. Alikalfa, M., 2013. Genetik Programlama İle Öncelik Kuralları ve Çizelgeleme Algoritmalarının Keşfi, Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, Eskişehir, 169 s.
13. Boussaïd, I., Lepagnot, J., Siarry, P., 2013. **A survey on optimization metaheuristics. Information Sciences, 327:** 82-117.
14. Hancock, P.J.B., 1994. An empirical comparison of selection in evolutionary algorithms, pp. 80-94. *AISB Workshop on Evolutionary Computing*, April 11-13, 1994, Department of Psychology University of Stirling, Scotland.
15. Maza, M.D., and Tidor, B., 1992. Increased flexibility in genetic algorithms: The use of variable boltzmann selective pressure to control propagation, pp. 425-440. *In: Computer Science and Operations Research: New Developments in Their Interfaces* (Eds: O., Balcı). Pergamon Press, Oxford.
16. Goldberg, D. E., Deb, K., 1991. Comparative analysis of selection schemes used in genetic algorithms, pp. 69-93. *In: Foundations of Genetic Algorithms* (Eds: G.C.E., Rawlins). Elsevier Science Publishers, Amsterdam.
17. Fogel, D. B., Back, T., Michalewicz, Z., 1997. *Handbook of Evolutionary Computation*. IOP Publishing Ltd., Oxford University Press, Bristol, 1130 pp.
18. Ahvanooey, M.T., Li, Q., Wu, M., Wang, S., 2019. A survey of genetic programming and its applications. **KSII Transactions on Internet and Information Systems, 13** (4): 1765-1793.
19. Moraglio, A., Kwawiec, K., Jhonson, C. G., 2012. Geometric semantic genetic programming in parallel problem solving from nature, pp.21-31. *PPSN'12 Proceedings of the 12th International Conference on Parallel Problem Solving from Nature*, September 1-5, 2012, Taormina, Italy.
20. Vanneschi, L., Castelli, M., Manzoni, L., Silva, S., 2013. A new implementation of geometric semantic GP and its application to problems in pharmacokinetics, pp. 205-216. *EuroGP'13 Proceedings of the 16th European Conference on Genetic Programming*, April 3-5, 2013, Vienna, Austria.
21. Hara, A., Kushida, J., Kisaka K., Takahama, T., 2015. Geometric semantic genetic programming using external division of parents. *2015 IIAI 4th International Congress on Advanced Applied Informatics*, July 12-16, 2015, Okayama, Japan.
22. Ferreira, C., 2001. Gene expression programming: A new adaptive algorithm for solving problems. **Complex Systems, 13** (2): 87-129.

23. Man, S., Lee, K. H., Kwong, S. L., 2003. Evolving data classification programs using genetic parallel programming. *The 2003 Congress on Evolutionary Computation*, 2003. CEC '03, December 8-12, 2003, Canberra, Australia.
24. Watchareeruetai, U., Matsumoto, T., Takeuchi, Y., Kudo, H., Ohnishi, N., 2009. Construction of image feature extractors based on multi-objective genetic programming with redundancy regulations, pp. 1328-1333. *Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics*, October 11-14, 2009, San Antonio, TX, USA.
25. Searson, D.P., 2015. GPTIPS 2: An open-source software platform for symbolic data mining, pp. 551-573. *In: Handbook of Genetic Programming Applications* (Eds. D. P., Searson). Springer International Publishing, New York.
26. Gao, W., 2004. Fast immunized evolutionary programming, pp. 659-1318. *Proceedings of 2004 International Conference on Machine Learning and Cybernetics, IEEE Cat. No.04EX826*, August 26-29, 2004, Shanghai, China.
27. Sarkar, M., Yegnanarayana, B., 1997. A clustering algorithm using evolutionary programming. **Pattern Recognition Letters**, **18** (10): 975-986.
28. Li, D., Chen, Y., 2018. An efficient ant colony programming approach. *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovations*, October 8-12, 2018, Guangzhou, China.
29. Cano, A., Olmo, J.L., Ventura, S., 2013. Parallel multi-objective ant programming for classification using GPUs. **Journal of Parallel Distributed Computing**, **73** (6) : 713–728.
30. Kamali, M.Z.M., Kumaresan, N., Ratnavelu, K., 2015. Solving differential equations with ant colony programming. **Applied Mathematical Modelling**, **39**: 3150–3163.
31. Kamali, M.Z.M., Kumaresan, N., Ratnavelu, K., 2017. Takagi–Sugeno fuzzy modelling of some nonlinear problems using ant colony programming. **Applied Mathematical Modelling**, **48**: 635–654.
32. Hara, A., Watanabe, M., Takahama, T., 2011. Cartesian ant programming, pp. 3161-3166. *Proceeding of 2011 IEEE International Conference on Systems, Man and Cybernetics*, October 9-12, 2011, Alaska, USA.

33. Luis, S., Dos Santos, M.V., 2013. On the evolvability of a hybrid ant colony-cartesian genetic programming methodology, pp. 109-120. *Proceedings of 16th European Conference on Genetic Programming (EuroGP2013)*, LNCS 7831, April 3-5, 2013, Vienna, Austria.
34. Kushida, J., Hara, A., Takahama, T., Mimura, N., 2017. Cartesian ant programming introducing symbiotic relationship between ants and aphids, pp. 115-120. *IEEE 10th International Workshop on Computational Intelligence and Applications*, November 11-12, 2017, Hiroshima, Japan.
35. Hara, A., Kushida, J., Fukuhara, K., Takahama, T., 2014. Cartesian ant programming with adaptive node replacements. *2014 IEEE 7th International Workshop on Computational Intelligence and Applications*, November 7-8, 2014, Hiroshima, Japan.
36. Kushida, J., Hara, A., Takahama, T., 2015. Cartesian ant programming with node release mechanism. *2015 IEEE 8th International Workshop on Computational Intelligence and Applications*, November 6-7, 2015, Hiroshima, Japan.
37. Kushida, J., Hara, A., Takahama, T., Nagura, S., 2016. Cartesian ant programming with transition rule considering internode distance. *2016 IEEE 9th International Workshop on Computational Intelligence and Applications*, November 5, 2016, Hiroshima, Japan.
38. Musilek, P., Lau, A., Reformat, M., Wyard-Scott, L., 2006. Immune programming. **Information Sciences**, **176** (8): 972–1002.
39. Wang, S., Ma, J., He, Q., 2010. An immune programming-based ranking function discovery approach for effective information retrieval. **Expert Systems with Applications**, **37** (8): 5863–5871.
40. Mansor, M.H., Musirin, I., Othman, M.M., 2017. Immune log-normal evolutionary programming (ILNEP) for solving economic dispatch problem with prohibited operating zones. *2017 4th International Conference on Industrial Engineering and Applications*, April 21-23, 2017, Nagoya, Japan.
41. Lau, A., Musilek, P., 2009. Immune programming models of cryptosporidium parvum inactivation by ozone and chlorine dioxide. **Information Sciences**, **179** (10): 1469–1482.

42. Mabrouk, E., Ayman, A., Raslan, Y., Hedar, A., 2019. Immune system programming for medical image segmentation. **Journal of Computational Science**, **31**: 111–125.
43. Xu, S., Xuesong, X., 2016. Improved immune programming algorithm and its application in the model clustering. *2016 28th Chinese Control and Decision Conference*, May 28-30, 2016, Yinchuan, China.
44. Xu, H., Wang, J., Kim, H.J., 2010. Near-optimal solution to pair-wise LSB matching via an immune programming strategy. **Information Sciences**, **180** (8): 1201–1217.
45. Demirbas, D.M., Cakır, D., Arslan, S., Ozturk, C., 2018. Equivalent stress analysis of functionally graded rectangular plates by genetic programming. **International Scientific and Vocational Journal (ISVOS JOURNAL)**, **2** (1): 67-80.
46. Liu, K., Tong, M., Xie, S., Ng, V., 2015. Genetic programming based ensemble system for microarray data classification. **Hindawi Publishing Corporation Computational and Mathematical Methods in Medicine**, **2015** (2).
47. Patnaik, A.K., Bhuyan, P.K., 2016. Application of genetic programming clustering in defining LOS criteria of urban street in Indian context. **Travel Behaviour and Society**, **3**: 38–50.
48. Feng, Q., Lian, H., Zhu, J., 2017. Multi-level genetic programming for fault data clustering. *2017 Prognostics and System Health Management Conference (PHM-Harbin)*, July 9-12, 2017, Harbin, China.
49. Chen, Q., Zhang M., and Xue, B., 2017. Feature selection to improve generalization of genetic programming for high-dimensional symbolic regression, pp. 792-806. *IEEE Transactions on Evolutionary Computation*, October, 2017.
50. Choi, W.J., Choi, T.S., 2012. Genetic programming-based feature transform and classification for the automatic detection of pulmonary nodules on computed tomography images. **Information Sciences**, **212**: 57–78.
51. Evans, B., Al-Sahaf, H., Xue, B., Zhang, M., 2018. Evolutionary deep learning: A genetic programming approach to image classification. *2018 IEEE Congress on Evolutionary Computation (CEC)*, July 8-13, 2018, Rio de Janeiro, Brazil.
52. Al-Sahaf, H., Zhang, M., Al-Sahaf, A., Johnston, M., 2017. Keypoints detection and feature extraction: A dynamic genetic programming approach for evolving

- rotation-invariant texture image descriptors, pp. 825-844. *IEEE Transactions on Evolutionary Computation*, December, 2017.
53. Vojodi, H., Fakhari, A., Moghadam, A.M.E., 2013. A new evaluation measure for color image segmentation based on genetic programming approach. **Image and Vision Computing**, **31** (11): 877–886.
 54. Arslan, S., Ozturk, C., 2019. A comparative study of automatic programming techniques. **Informatica an International Journal of Computing and Informatics**, 43: 281-289.
 55. Boudardara, F., Gorkemli, B., 2018. Application of artificial bee colony programming to two trails of the artificial ant problem. *2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, October 19-21, 2018, Ankara, Turkey.
 56. Arslan, S., Ozturk, C., 2019. Artificial bee colony programming descriptor for multi-class texture classification. **Applied Sciences**, **9** (9): 1930.
 57. Arslan, S., Ozturk, C., 2018. Artificial bee colony programming for feature selected cancer data classification. **International Journal of Scientific and Technological Research**, **4** (7).
 58. Arslan, S., Ozturk, C., Feature Selection for Classification with Artificial Bee Colony Programming. Book Chapter, *Swarm Intelligence – Recent Advances, New Perspectives And Applications*.
 59. Derhami, S., Smith, A.E., 2017. An integer programming approach for fuzzy rule-based classification systems. **European Journal of Operational Research**, **256** (23): 924–934.
 60. Karaboga, D., 2010. Artificial bee colony (ABC) algorithm. **Scholarpedia**, **5** (3): 6915-6915.
 61. Ghambari, S., Rahati, A., 2018. An improved artificial bee colony algorithm and its application to reliability optimization problems. **Applied Soft Computing**, **62**: 736–767.
 62. Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N., 2012. A comprehensive survey: artificial bee colony (ABC). **Artificial Intelligence Review**, **42** (1): 21–57.

63. Bansal, J.C., Joshi, S.K., Sharma, H., 2018. Modified global best artificial bee colony for constrained optimization problems. **Computers and Electrical Engineering**, **67**: 365-382.
64. Karaboga, D., Akay, B., 2009. A comparative study of artificial bee colony algorithm. **Applied Mathematics and Computation**, **214**: 108–132.
65. Motta, F.A.A., Freitas, J.M., Souza, F.R., Bernardino, H.S., Oliveira, I.L., Barbosa, H.J.C., 2018. A hybrid grammar-based genetic programming for symbolic regression problems. *2018 IEEE Congress on Evolutionary Computation (CEC)*, July, 2018.
66. Wu, C.H., Chou, H.J., Su, W.H., 2008. Direct transformation of coordinates for GPS positioning using the techniques of genetic programming and symbolic regression. **Engineering Applications of Artificial Intelligence**, **21** (8): 1347–1359.
67. Smetka, T., Homoliak, I., Hanacek, P., 2016. On the application of symbolic regression and genetic programming for cryptanalysis of symmetric encryption algorithm, pp. 1-8. *2016 IEEE International Carnahan Conference on Security Technology (ICCST)*, October 24-27, 2016, Orlando, FL, USA.
68. Kumar, B., Jha, A., Deshpande, V., Sreenivasulu, G., 2014. Regression model for sediment transport problems using multi-gene symbolic genetic programming. **Computers and Electronics in Agriculture**, **103**: 82–90.
69. Otavio L., Oliveira, V.B., Otero, F.E.B., Miranda, L.F., Pappa, G.L., 2016. Revisiting the sequential symbolic regression genetic programming. *2016 5th Brazilian Conference on Intelligent Systems*, October 9-12, 2016, Recife, Brazil.
70. Haeria, M.A., Ebadzadeha, M.M., Folino, G., 2017. Statistical genetic programming for symbolic regression. **Applied Soft Computing**, **60**: 447–469.
71. Sotto, L.F.P., Melo, V.V., 2016. Studying bloat control and maintenance of effective code in linear genetic programming for symbolic regression. **Neurocomputing**, **180** (C): 79–93.
72. Barmapalexis, P., Kachrimanis, K., Tsakonas, A., Georgarakis, E., 2011. Symbolic regression via genetic programming in the optimization of a controlled release pharmaceutical formulation. **Chemometrics and Intelligent Laboratory Systems**, **107** (1): 75–82.

73. Seidy, E.E., 2016. A new particle swarm optimization based stock market prediction technique. **International Journal of Advanced Computer Science and Applications (IJACSA)**, 7: (4).
74. Manjusha, K.K., Sankaranayanan, K., Seena P., 2015. Data Mining in dermatological diagnosis: A method for severity prediction. **International Journal of Computer Applications**, 117 (11): 975 – 8887.
75. BuHamra, S., Smaoui, N., Gabr M., 2003. The box-jenkins analysis and neural networks: prediction and time series modelling. **Applied Mathematical Modelling**, 27 (10): 805–815.
76. Martin, V., Martin, V.A., Venkatesan V.P., 2012. Framing qualitative bankruptcy prediction rules using ant colony algorithm. **International Journal of Computer Applications**, 41 (21): 0975 – 8887.
77. Etemadi, H., Rostamy, A.A.A., Dehkordi, H.F., 2009. A genetic programming model for bankruptcy prediction: Empirical evidence from Iran. **Expert Systems with Applications**, 36 (2): 3199–3207.
78. Paul, T.K., Iba, H., 2009. Prediction of cancer class with majority voting genetic programming classifier using gene expression data. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, 6 (2): 353-367.
79. Asima, K.M., Idrisb, A., Iqbala, T., Álvarezc, F.M., 2018. Seismic indicators based earthquake predictor system using genetic programming and AdaBoost classification. **Soil Dynamics and Earthquake Engineering**, 111: 1–7.
80. Arslan, S., Ozturk, C., 2017. Prediction of cherkassky function by artificial bee colony programming. **Akıllı Sistemlerde Yenilikler ve Uygulamalar (ASYU17)**, pp. 1-5.
81. Gorkemli B., Ozturk C., Karaboga D., 2012. Yapay arı kolonisi programlama ile sistem modelleme, pp. 857-860. *Otomatik Kontrol Türk Milli Komitesi 2012 Ulusal Toplantısı (TOK'2012)*, Ekim 11-13, 2012, Niğde.
82. Rodriguez-Lujan, I., Huerta, R., Elkan, C., Santa Cruz, C., 2010. Quadratic programming feature selection. **Journal of Machine Learning Research**, 11: 1491-1516.
83. Novaković, J., Strbac, P., Bulatović, D., 2011. Toward optimal feature selection using ranking methods and classification algorithms. **Yugoslav Journal of Operations Research**, 21 (1): 119-135.

84. Sikdar, U.K., Ekbal, A., Saha, S., 2012. Differential evolution based feature selection and classifier ensemble for named entity recognition, pp. 2475–2490. *Proceedings of COLING 2012: Technical Papers*, December, 2012, Mumbai, India.
85. Viegas, F., Rocha, L., Gonçalves, M., Mourão, F., Saa, G., Salles, T., Andrade, G., Sandin, I., 2018. A genetic programming approach for feature selection in highly dimensional skewed data. **Neurocomputing**, **273**: 554–569.
86. Mei, Y., Nguyen, S., Xue, B., Zhang, M., 2017. An efficient feature selection algorithm for evolving job shop scheduling rules with genetic programming. **IEEE Transactions on Emerging Topics in Computational Intelligence**, **1** (5): 339-353.
87. Guo, L., Rivero, D., Dorado, J., Munteanu, C.R., Pazos, A., 2011. Automatic feature extraction using genetic programming: An application to epileptic EEG classification. **Expert Systems with Applications**, **38** (8): 10425–10436.
88. Jia, H., Verma, N., 2018. Exploiting approximate feature extraction via genetic programming for hardware acceleration in a heterogeneous microprocessor. **IEEE Journal Of Solid-State Circuits**, **53** (4): 1016-1027.
89. Cherkassky, V., Gehring, D., Mulier, F., 1996. Comparison of adaptive methods for function estimation from samples. **IEEE Transactions on Neural Networks**, **7** (4): 969- 984.
90. Searson, D., 2017. Genetic Programming and Symbolic Data Mining Platform for MATLAB. (Web page: <https://sites.google.com/site/gptips4matlab/file-cabinet>), (Date accessed: 01.10.2016).
91. Searson, D. Genetic Programming and Symbolic Data Mining Platform for MATLAB.(Web page: <https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength>), (Date accessed: October 2016).
92. Searson, D.P., 2015. GPTIPS 2: An open-source software platform for symbolic data mining, pp. 551-573. *In: Handbook of Genetic Programming Applications* (Eds. D.P., Searson). Springer, New York, NY.
93. Cai R., Hao Z., Yang X., Wen W., 2009. An efficient gene selection algorithm based on mutual information. **Neurocomputing**, **72** (4-6): 991-999.

94. Saeys, Y., Inza, I., Larranaga, P., 2007. A review of feature selection techniques in bioinformatics. **Bioinformatics**, **23** (19): 2507-2517.
95. Guyon, I., Elisseeff, A., 2003. An introduction variable and feature selection. **Journal of Machine Learning Research**, **3**: 1157-1182.
96. Gulgezen, G., 2009. Kararlı ve Başarımı Yüksek Öznitelik Seçimi. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 85 s.
97. Zhang, Y., Wanga, S., Phillips, P., Ji, G., 2014. Binary PSO with mutation operator for feature selection using decision tree applied to spam detection. **Knowledge-Based Systems**, **64** (1): 22–31.
98. Sikora, R., Piramuthu, S., 2007. Framework for efficient feature selection in genetic algorithm based data mining. **European Journal of Operational Research**, **180** (2): 723–737.
99. Aslam, M.W., Zhu, Z., Nandi, A.K., 2013. Feature generation using genetic programming with comparative partner selection for diabetes classification. **Expert Systems with Applications**, **40** (13): 5402–5412.
100. Harvey, D.Y., Todd, M.D., 2015. Automated feature design for numeric sequence classification by genetic programming. **IEEE Transactions Evolutionary Computing**, **19** (4): 474–489.
101. Muharram, M.A.Y., 2005. The Application of Genetic Programming for Feature Construction in Classification. University of East Anglia, School of Computing Sciences, Doctoral Thesis, Norwich, 187 pp.
102. Bagui, S., Hemasinha, R., 2016. The statistical classification of breast cancer data. **International Journal of Statistics and Applications**, **6** (1): 15-22.
103. Awuley, A., Ross, B.J., 2016. Feature selection and classification using age layered population structure genetic programming. *CEC 2016, Vancouver BC*, July 24-29, 2016, Vancouver, BC, Canada.
104. Liu, K., Xu, C., 2009. A genetic programming-based approach to the classification of multiclass microarray datasets. **Bioinformatics**, **25** (3): 331–337.
105. Chaudhari, N.S., Purohit, A., Tiwari, A., 2008. A multiclass classifier using genetic programming. *2008 10th International Conference on Control, Automation, Robotics and Vision*, December 17-20, 2008, Hanoi, Vietnam.

106. Mause, G., Grbac, T.G., 2017. Co-evolutionary multi-population genetic programming for classification in software defect prediction: An empirical case study. **Applied Soft Computing**, 55 (C): 331–351.
107. Abdulhamid, F., Song, A., Neshatian, K., Zhang, M., 2012. Evolving genetic programming classifiers with loop structures. *WCCI 2012 IEEE World Congress on Computational Intelligence*, June 10-15, 2012, Brisbane, Australia.
108. Yan, L., Zeng, J., 2006. Using particle swarm optimization and genetic programming to evolve classification rules. *Proceedings of the 6th World Congress on Intelligent Control and Automation*, June 21 - 23, 2006, Dalian, China.
109. UCI Machine Learning Repository. (Web page: <http://archive.ics.uci.edu/ml/index.php>), (Date accessed: October 2016).
110. Salama, G.I., Abdelhalim, M.B., Zeid, M.A., 2012. Breast cancer diagnosis on three different datasets using multi-classifiers. **International Journal of Computer and Information Technology**, 1 (1): 36-43.
111. Kathija, A., Nisha, S., 2016. Breast cancer data classification using SVM and naïve bayes techniques. **International Journal of Innovative Research in Computer and Communication Engineering**, 4 (12).
112. Güvenir, H.A., Demiröz, G., İltar, N., 1998. Learning differential diagnosis of erythematous-squamous diseases using voting feature intervals. **Artificial Intelligence in Medicine**, 13 (3): 147–165.
113. Rambhajan, M., Deepanker, W., Pathak, N., 2015. A survey on implementation of machine learning techniques for dermatology diseases classification. **International Journal of Advances in Engineering & Technology**, 8 (2): 194-202.
114. Barati, E., Saraee, M., Mohammadi, A., Adibi, N., Ahamadzadeh, M.R., 2011. A survey on utilization of data mining approaches for dermatological (Skin) diseases prediction, cyber journals: Multidisciplinary journals in science and technology. **Journal of Selected Areas in Health Informatics (JSHI)**.
115. Parikh, K.S., Shah, T.P., Kota, R., Vora, R., 2015. Diagnosing common skin diseases using soft computing techniques. **International Journal of Bio-Science and Bio-Technology**, 7 (6): 275-286.

- 116.Pappa, G.L., Freitas, A.A., Kaestner C.A.A., 2002. Attribute selection with a multiobjective genetic algorithm, pp. 280-290. *SBIA 2002: Advances in Artificial Intelligence*, November 11-14, 2002, Brazil.
- 117.Zhong, P., Fukushima, M., 2007. A regularized nonsmooth newton method for multi-class support vector machines. **Optimization Methods and Software**, 22 (1): 225-236.
- 118.Dy, J.G., Brodley, C.E., 2004. Feature selection for unsupervised learning. **Journal of Machine Learning Research**, 5: 845–889.
- 119.Fischer, I., Poland, J., 2005. Amplifying the block matrix structure for spectral clustering, pp. 21-28. IDSIA-03-05.
- 120.Nock, R., Sebban, M., Bernard, D., 2003. A simple locally adaptive nearest neighbor rule with application to pollution forecasting. **International Journal of Pattern Recognition and Artificial Intelligence**, 17 (8): 1369-1382.
- 121.Morrison, G.A., Searson, D. P., Willis, M. J., 2010. Using genetic programming to evolve a team of data classifiers. **World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering**, 4 (12): 1815-1818.
- 122.Bellman, R.E., 2003. Dynamic Programming. Dover Publications, Mineola, NY, USA, 366 pp.
- 123.Chandrashekar, G., Sahin, F., 2014. A survey on feature selection methods. **Computers and Electrical Engineering**, 40 (1): 16–28.
- 124.Neshatian, K., Zhang, M., 2009. Pareto front feature selection: Using genetic programming to explore feature space, pp. 1027–1034. *Proceedings 11th Annual Conference Genetic Evolution Computing Conference (GECCO)*, July 8-12, 2009, Montreal, QC, Canada.
- 125.Shanthi, S., Bhaskaran, V.M., 2014. Modified artificial bee colony based feature selection: A new method in the application of mammogram image classification. **International Journal of Science, Engineering and Technology Research (IJSETR)**, 3 (6): 1664-1667.
- 126.Uzer, M.S., Yilmaz, N., Inan, O., 2013. Feature selection method based on artificial bee colony algorithm and support vector machines for medical datasets classification. **The Scientific World Journal**.

- 127.Chong, S.Y., Tino, P., Ku, D.C., Yao, X., 2012. Improving generalization performance in co-evolutionary learning. **IEEE Transaction Evolutionary Computing**, **16** (1): 70–85.
- 128.Cohn, D., Atlas, L., Ladner, R., 1994. Improving generalization with active learning. **Machine Learning**, **15** (2): 201–221.
- 129.Karaboga, D., Kaya, E., 2017. Training ANFIS by using the artificial bee colony algorithm. **Turkish Journal of Electrical Engineering & Computer Sciences**, **25**: 1669-1679.
- 130.Dhahri, H., Alimi, A.M., Abraham, A., 2012. Designing beta basis function neural network for optimization using artificial bee colony (ABC), pp. 2923-2929. *WCCI 2012 IEEE World Congress on Computational Intelligence*, June 10-15, 2012, Brisbane, Australia.
- 131.Kushchu, I., 2002. Genetic programming and evolutionary generalization. **IEEE Transaction Evolutionary Computing**, **6** (5): 431–442.
- 132.Gonçalves, I., Silva, S., Fonseca, C.M., 2015. On the generalization ability of geometric semantic genetic programming, pp. 41-52. *Proceedings 18th European Conference Genetic Program (EuroGP)*, April 8-10, 2015, Copenhagen, Denmark.
- 133.Uy, N.Q., Hien, N.T., Hoai, N.X., O'Neill, M., 2010. Improving the generalisation ability of genetic programming with semantic similarity based crossover, pp. 184-195. *Proceedings 13th European Conference Genetic Program (EuroGP)*, April 7-9, 2010, Istanbul, Turkey.
- 134.Stijven, S., Minnebo, W., Vladislavleva, K., 2011. Separating the wheat from the chaff: On feature selection and feature importance in regression random forests and symbolic regression, pp. 623-630. *Proceedings 13th Annual Conference Genetic Evolutionary Computing Conference (GECCO)*, July 12-16, 2011. Dublin, Ireland.
- 135.Keijzer, M., 2003. Improving symbolic regression with interval arithmetic and linear scaling, pp. 70–82. *Proceedings 6th European Conference Genetic Program (EuroGP)*, April 14-16, 2003, Berlin.
- 136.Vanneschi, L., Silva, S., Castelli, M., Manzoni, L., 2014. Geometric semantic genetic programming for real life applications, pp. 191–209. *In: Genetic Programming*

- Theory and Practice XI. (Eds. R. Riolo, J.H. Moore, M. Kotanchek). Springer, New York.
137. Rosenwald, A., 2002. The use of molecular profiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma. **New England Journal of Medicine**, **346** (25): 1937–1947.
 138. Liu, L., Zhao, L., Long, Y., Kuang, G., Fieguth P., 2012. Extended local binary patterns for texture classification. **Image and Vision Computing**, **30** (2): 86-99.
 139. Tackett, W. A., 1993. Genetic programming for feature discovery and image discrimination, pp. 303-311. *Proceedings of the 5th International Conference on Genetic Algorithms*, December, 1993, San Francisco, CA, USA.
 140. Shao, L., Liu, L., Li, X., 2014. Feature learning for image classification via multiobjective genetic programming. **IEEE Transactions on Neural Networks and Learning Systems**, **25** (7): 1359–1371.
 141. Lensen, A., Sahaf, H.A., Zhang, M., Xue, B., 2015. A hybrid genetic programming approach to feature detection and image classification. *International Conference on Image and Vision Computing*, November 23-24, 2015, Auckland, New Zealand.
 142. Iqbal, M., Xue, B., Al-Sahaf, H., Zhang, M., 2017. Cross-domain reuse of extracted knowledge in genetic programming for image classification. **IEEE Transactions on Evolutionary Computation**, **21** (4): 569-587.
 143. Lensen, A., Al-Sahaf, H., Zhang, M., Xue, B., 2016. Genetic programming for region detection, feature extraction, feature construction and classification in image data, pp. 51-67. *EuroGP 2016: Proceedings of the 19th European Conference on Genetic Programming*, March, 2016,
 144. Al-Sahaf, H., Song, A., Neshatian, K., Zhang, M., 2012. Two-tier genetic programming: Towards raw pixel-based image classification. **Expert Systems with Applications**, **39** (16): 12291–12301.
 145. Karaboga, D., Basturk, B., 2008. On the Performance of Artificial Bee Colony (ABC) Algorithm. **Applied Soft Computing**, **8** (1): 687-697.
 146. Hancer, E., Ozturk, C., Karaboga, D., 2012. Artificial bee colony based image clustering method. *WCCI IEEE World Congress on Computational Intelligence*, June 10-15, 2012, Brisbane, Australia.

147. Wang, S., 2011. Artificial bee colony used for rigid image registration. **International Journal of Research and Reviews in Soft and Intelligent Computing (IJRRSIC)**, **1** (2): 1936-1953.
148. Karaboga, D., Ozturk, C., 2009. Neural networks training by artificial bee colony algorithm on pattern classification. **Neural Network World**, **19** (3): 279-292.
149. Maa, M., Lianga, J., Guoa, M., Fana, Y., Yinb, Y., 2011. SAR image segmentation based on artificial bee colony algorithm. **Applied Soft Computing**, **11** (8): 5205–5214.
150. Zhihui, H., Weiyu, Y., Shanxiang, L., Jiuchao, F., 2013. Multi-level threshold image segmentation using artificial bee colony algorithm. *Fifth Conference on Measuring Technology and Mechatronics Automation*, January 16-17, 2013, Hong Kong, China.
151. Dakshitha, B. A., Deekshitha, V., Manikantan, K., 2015. A novel bi-level artificial bee colony algorithm and its application to image segmentation. *IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, December 10-12, 2015, Madurai, India.
152. Sag, T., Cunkas, M., 2015. Color image segmentation based on multiobjective artificial bee colony optimization. **Applied Soft Computing**, **34**: 389–401.
153. Hancer, E., Ozturk, C., Karaboga, D., 2013. Extraction of brain tumors from MRI images with artificial bee colony based segmentation methodology, pp. 516-520. *8th International Conference on Electrical and Electronics Engineering (ELECO)*, November 28-30, 2013, Bursa, Turkey.
154. Agrawal, V., Chandra, S., 2015. Feature selection using artificial bee colony algorithm for medical image classification. *Eighth International Conference on Contemporary Computing (IC3)*, August 20-22, 2015, Noida, India.
155. Al-Sahaf, H., Zhang, M., Johnston, M., 2013. Binary image classification using genetic programming based on local binary patterns. *28th International Conference on Image and Vision Computing*, November 27-29, 2013, Wellington, New Zealand.
156. Ojala, T., Pietikaè, M., Maèenpaè, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, **24** (7): 971-987.

- 157.Sinha, A., Banerji, S., Liu, C., 2014. Scene image classification using a wigner-based local binary patterns descriptor. *2014 International Joint Conference on Neural Networks (IJCNN)*, July 6-11, 2014, Beijing, China.
- 158.Pan, Z., Li, Z., Fan, H., Wu, X., 2017. Feature based local binary pattern for rotation invariant texture classification. **Expert Systems with Applications**, **88**: 238–248.
- 159.Ojala, T., Pietikäinen, M., Harwood, D., 1994. Performance evaluation of texture measures with classification based on kullback discrimination of distributions, pp. 582-585. *Proceedings of the 12th International Conference on Pattern Recognition*, October 9-13, 1994, Jerusalem, Israel.
- 160.Ojala, T., Pietikäinen, M., Mäenpää, T., 2000. Gray scale and rotation invariant texture classification with local binary patterns, pp. 404-420. *Proceedings 6th European Conference Computing Vision*, June 26 - July 1, 2000, Dublin, Ireland.
- 161.Al-Sahaf, H., Zhang, M., Johnston, M., Verma, B., 2015. Image descriptor: A genetic programming approach to multiclass texture classification. *Proceedings IEEE Congress Evolutionary Computing*, May 25-28, 2015, Sendai, Japan.
- 162.Brodatz, P., 1999. Textures: A Photographic Album for Artists and Designers. Dover Pubns, New York, USA, 112 s.
- 163.Kylberg, G., 2011. The Kylberg texture dataset V. 1.0, Centre Image Analysis, Swedish University Agricultural Sciences, Uppsala University, Uppsala, Sweden.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı: Sibel ARSLAN
Uyruğu: Türkiye (T.C.)
Doğum Tarihi ve Yeri: 20.01.1988 - Sivas
e-mail: sibel.arslan2@icisleri.gov.tr
Yazışma Adresi: Mimar Sinan Mah. Yol İş Sitesi B Blok No:4 Sivas/Merkez

EĞİTİM

Derece	Kurum	Mezuniyet Tarihi
Doktora	Bilgisayar Mühendisliği, Erciyes Üniversitesi	2019
Yüksek Lisans	Bilgisayar Mühendisliği, Erciyes Üniversitesi	2014
Lisans	Bilgisayar Mühendisliği, Kocaeli Üniversitesi	2011
Lisans	Endüstri Mühendisliği, Kocaeli Üniversitesi	2011
Lise	Kongre Lisesi (Yabancı Dil Ağırlıklı), Sivas	2006

İŞ DENEYİMLERİ

Yıl	Kurum	Görev
2012-Halen	Sivas Valiliği Bilgi İşlem Şube Müdürlüğü	Mühendis
2011	Kocaeli Üniversitesi Bilgisayar Mühendisliği	Araştırma Görevlisi

YABANCI DİL

İngilizce

SCI-E, IJSTR, ROAD, EMERGING INDEX, WEB OF SCIENCE (CLARIVATE ANALYTICS) İNDEKSLERİNE GİREN DERGİLERDE YAYINLANAN MAKALELER

1. Arslan S., Okdem S., “Düşük Hızlı Kablosuz Kişisel Alan Ağlarında Gürültülü Sinyallere Yönelik Adaptif Veri Dizisi Üretici”, JOURNAL OF THE FACULTY OF ENGINEERING AND ARCHITECTURE OF GAZI UNIVERSITY, vol.30, no.3, pp.371-380, 2015.
2. Arslan S., Ozturk C., “Multi Hive Artificial Bee Colony Programming for High Dimensional Symbolic Regression with Feature Selection”, APPLIED SOFT COMPUTING JOURNAL, vol.78, pp.515–527, 2019.
3. Arslan S., Ozturk C., “Artificial Bee Colony Programming Descriptor for Multi-Class Texture Classification”, APPLIED SCIENCES , 9(9):1930, 2019.
4. Arslan S., Ozturk C., “Artificial Bee Colony Programming for Feature Selected Cancer Data Classification”, INTERNATIONAL JOURNAL OF SCIENTIFIC AND TECHNOLOGICAL RESEARCH, vol.4, no.7, 2018.
5. Arslan S., Ozturk C., “A Comparative Study of Automatic Programming Techniques”, INFORMATICA AND INTERNATIONAL JOURNAL OF COMPUTING AND INFORMATICS, vol. 43, pp. 281-289, 2019.
6. Demirbaş M.D., Çakır D., Arslan S., Ozturk C., “Equivalent Stress Analysis of Functionally Graded Rectangular Plates by Genetic Programming”, INTERNATIONAL SCIENTIFIC AND VOCATIONAL JOURNAL (ISVOS JOURNAL), vol.2, no.1, pp.67-80, 2018.

KİTAP BÖLÜMÜ

1. Arslan S., Ozturk C., “Feature Selection for Classification with Artificial Bee Colony Programming”, SWARM INTELLIGENCE – RECENT ADVANCES, NEW PERSPECTIVES AND APPLICATIONS.

HAKEMLİ KONGRE / SEMPOZYUMLARIN BİLDİRİ KİTAPLARINDA YER ALAN YAYINLAR

1. Arslan S., Ozturk C., “Feature Selected Cancer Data Classification with Genetic Programming”, 21. National Meeting of Biomedical Engineering (BİYOMUT 2017), TÜRKİYE, 2017.
2. Arslan S., Ozturk C., “Çok Sınıflı Desenli Sınıflandırma Problemi için Genetik Programlama Tanımlayıcı”, International Conference on Innovative Engineering Applications (CIEA 2018), SİVAS, TÜRKİYE, 20-22 Eylül 2018.
3. Arslan S., Ozturk C., “Yapay Arı Koloni Programlama ile Cherkassky Fonksiyon Tahmini”, Akıllı Sistemlerde Yenilikler ve Uygulamaları Konferansı (ASYU 2017), TÜRKİYE, 5-7 Ekim 2017.