T.C.
İSTANBUL ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME ANABİLİM DALI
SAYISAL YÖNTEMLER BİLİM DALI

DOKTORA TEZİ

# YAPAY ARI KOLONİSİ ALGORİTMASI İLE FİNANSAL PORTFÖY OPTİMİZASYONU

Burcu KARTAL
2502100300

Tez Danışmanı
Doç.Dr.Tunçhan CURA

İSTANBUL 2015

T.C. İSTANBUL ÜNİVERSİTESİ
SOSYAL BİLİMLER
1982 ENSTİTÜSÜ



# DOKTORA

# TEZ ONAYI

**ÖĞRENCİNİN**

Adı ve Soyadı : **BURCU KARTAL** Numarası : **2502100300**

Anabilim/Bilim Dalı : **SAYISAL YÖNTEMLER** Danışman Öğretim Üyesi: **DOÇ.DR TUNÇHAN CURA**

Tez Savunma Tarihi : **16.03.2015** Tez Savunma Saati : **13:00**

Tez Başlığı : **YAPAY ARI KOLONİSİ ALGORİTMASI İLE FİNANSAL PORTFÖY OPTİMİZASYONU.**

**TEZ SAVUNMA SINAVI**, Lisansüstü Öğretim Yönetmeliği'nin **50. Maddesi** uyarınca yapılmış, sorulan sorulara alınan cevaplar sonunda adayın tezinin **KABULÜ'NE** OYBİRLİĞİ / ~~OYÇOKLUĞUYLA~~ karar verilmiştir.

| JÜRİ ÜYESİ | İMZA | KANAATİ (KABUL / RED / DÜZELTME) |
|---|---|---|
| 1- PROF.DR ERHAN ÖZDEMİR | | |
| 2- PROF.DR ZUHAL TANRIKULU | | Kabul |
| 3- DOÇ.DR ALP BARAY | | Kabul |
| 4-DOÇ.DR TUNÇHAN CURA | | Kabul |
| 5- DOÇ.DR BİLGE ACAR BOLAT | | Kabul |

| YEDEK JÜRİ ÜYESİ | İMZA | KANAATİ (KABUL / RED / DÜZELTME) |
|---|---|---|
| 1- PROF.DR İSMAİL HAKKI ARMUTLULU | | |
| 2- DOÇ.DR SEDA TOLUN | | Kabul. |

Yapay Arı Kolonisi Algoritması ile Finansal Portföy Optimizasyonu

**Burcu KARTAL**

## ÖZ

Küreselleşen dünyada insanlar karşılaştıkları sorunlara kısa sürede cevap bulmak istemektedirler. Zamanı en etkin şekilde kullanmak, gelirleri arttırmak, giderleri azaltmak her insanın çözümünü aradığı problemlerdir. Optimizasyon problemleri adı verilen bu problemlerin en önemlilerinden biri de Portföy Optimizasyonu (PO) Problemidir. Önceden çözülmesi karmaşık ve zaman alıcı olan PO problemleri, son zamanlarda kullanımı oldukça artan sezgisel tekniklerle çözülmeye başlanmıştır. Zamandan oldukça tasarruf sağlayan bu yöntemlerden biri de 2005 yılında literatüre kazandırılmış Yapay Arı Kolonisi Algoritmasıdır(YAKA). Bu yöntem, arıların besin arama davranışları gözlenerek ve arı kolonilerinden esinlenilerek ortaya çıkmıştır. Bu çalışmada YAKA ile PO problemine çözüm aranmıştır. Literatürde yer alan 5 tane örnek problem ve BIST30 verileri ile sınamalar yapılmıştır. Daha önce, Tavlama Benzetimi (BT), Tabu Arama (TA), Genetik Algoritma (GA) ve Parçacık Sürü Optimizasyonu(PSO) yöntemleriyle elde edilen sonuçlarla kıyaslanmıştır. YAKA'dan elde edilen sonuçların, TB,TA,GA ve PSO'dan elde edilen sonuçlar kadar iyi olduğu gösterilmiştir.

**Anahtar Kelimeler:** Sürü Zekası, Yapay Arı Kolonisi, Metasezgisel Yöntemler, Portföy Optimizasyonu

Financial Portfolio Optimization with Artifical Bee Colony Algorithm

**Burcu KARTAL**

## ABSTRACT

As the world gets more and more globalized, one wants to find fast answers to the faced problems. Spending time efficiently, increasing revenues, reducing costs are the problems for which every human is looking for the solution. One of the most important of these problems, also known as Optimization problems, is called Portfolio Optimization (PO) problem. Although it was quite complex and time-consuming to solve a PO problem, the number of PO problems that are successfully solved has been recently increased with the use of heuristic techniques. One of these quite timesaving methods was introduced to the literature in 2005, namely the Artificial Bee Colony Algorithm (ABCA). This method is inspired by the foraging behavior of bees and emerged from observing bee colonies. In this study, we search for a solution to PO problem using ABCA. We test our algorithm using BIST30 data set and five benchmark instances that are already used by other researchers to test their algorithms. The results found by ABCA are compared to those, which were previously obtained by Simulated Annealing (SA), Tabu Search (TS), Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) methods. Overall results show that our proposed ABC method is comparable to the other methods, namely, SA, TS GA and PSO.

**Key Words:** Swarm Intelligence, Artificial Bee Colony, Meta-Heuristic Methods, Portfolio Optimization

# ÖNSÖZ

Son yıllarda teknolojideki gelişmeler ileri boyutlara ulaşmıştır. İnsanlar en basit işlerini bile bilgisayarda yapmaya başlamışlardır. Teknoloji kullanımının her alanda artması eskiden çözümü yıllar alan veya çözüme ulaşmamış problemlerin çözülmesini sağlamıştır. Markowitz 1952'de Portföy Optimizasyonu için oldukça başarılı bir yöntem üretse de yatırım araçlarının artması problemin çözümünü zorlaştırmaktadır. Çünkü büyük yatırımcılar paralarını farklı enstrümanlara yatırarak riski azaltmak istemektedirler. Böyle durumlarda optimal veya optimale yakın çözümler sunan sezgisel ve metasezgisel yöntemler kullanılabilmektedir. Bu yöntemlerden birisi de 2005 yılında literatüre kazandırılan Yapay Arı Kolonisi Algoritmasıdır (YAKA). Nispeten yeni bir yöntem olmasına rağmen hızlı bir şekilde kullanımı yaygınlaşmıştır. Literatüre bakıldığında YAKA'nın önemli optimizasyon problemlerinin çözümünde etkili sonuçlar ürettiği görülmüştür. Portföy Optimizasyonu problemi için sezgisel ve metasezgisel tekniklerle çözümler bulunmaktadır. Fakat ilgili literatür incelendiğinde YAKA ile yapılmış bir çalışmaya rastlanmamıştır. Bu da bizi bu alanda çalışma yapmaya sevk etmiştir. Böylece YAKA'nın diğer sezgisel tekniklerle karşılaştırmalarını yaparak Portföy Optimizasyonu probleminin çözümündeki yerini belirlemek istedik.

Bu çalışmada bana pratik düşünmeyi öğreten, hayatta hiçbir şey için endişe duymamam gerektiğini gösteren ve kazandığı tüm deneyimlerini açık yüreklilikle anlatan değerli tez danışmanım Doç. Dr. Tunçhan CURA'ya, en derin bilgilerini benimle paylaşan ve yetiştirilmem de emekleri olan Sayın Prof. Dr. Erhan ÖZDEMİR'e ve Sayın Doç.Dr. Alp BARAY'a en derin teşekkürlerimi sunarım.

Gurbette bana yuva sıcaklığını yaşatan ve donanımlı bir eğitim almamı sağlayan tüm Sayısal Yöntemler Anabilim Dalı öğretim üyeleri ve öğretim elemanlarına teker teker teşekkür ederim.

Tez çalışmam boyunca gösterdikleri hassasiyet ve manevi desteklerinden dolayı Recep Tayyip Erdoğan Üniversitesi İ.İ.B.F Dekanı Sayın Prof. Dr. Osman KARAMUSTAFA'ya, verilerin elde edilmesinde değerli zamanını bana ayıran değerli meslektaşım Arş. Gör. Musa GÜN'e ve tüm çalışma arkadaşlarıma teşekkür ederim.

Benim bu çalışmayı bitirmemi benden daha çok isteyen, her konuda beni destekleyen ve yardımını esirgemeyen canım babam Turgut KEMER'e, canım annem Emine KEMER'e, bitanecik abim Barış KEMER'e ve ailemizin kıymetlisi Kübra TOMAKİN KEMER'e sonsuz teşekkürü borç bilirim.

Beni gelini değil de kızı gibi seven ve çalışmam için tüm imkanlarını önüme seren canım babam Hayrullah KARTAL'a, canım annem Sevgi KARTAL'a ve neşe kaynağımız Emre KARTAL'a teşekkür ederim.

Bana tezi yazmam gerektiğini sürekli hatırlatan, değerlim, eşim Ümit KARTAL'a ve annesini hiç üzmeyen, tezi yazmama müsaade eden canım kızım, meleğim ŞEVVAL KARTAL'a sonsuz teşekkür ederim.

Doktora Tezimi, 2211A-Yurt İçi Doktora Bursu ile maddi olarak destekleyen TÜBİTAK Bilim Adamı Yetiştirme Grubuna teşekkürü borç bilirim.

Arş. Gör. Burcu KARTAL

.

# İÇİNDEKİLER

**ÖZ** ..... **III**
**ABSTRACT** ..... **IV**
**ÖNSÖZ** ..... **V**
**İÇİNDEKİLER** ..... **VII**
**TABLOLAR LİSTESİ** ..... **X**
**ŞEKİLLER LİSTESİ** ..... **XI**
**KISALTMALAR LİSTESİ** ..... **XII**
**GİRİŞ** ..... **1**
**1. OPTİMİZASYON VE SÜRÜ ZEKASI** ..... **3**
1.1. Optimizasyon Kavramı ..... 3
1.2. Optimizasyon Problemlerinde Kullanılan Popülasyon Tabanlı Sezgisel Teknikler ..... 12
1.2.1. Gelişime Dayalı Algoritmalar ..... 12
1.2.1.1. Evrimsel Strateji ..... 16
1.2.1.2. Evrimsel Programlama ..... 19
1.2.1.3. Genetik Algoritma ..... 21
1.2.1.3.1. Genel Bilgi ..... 21
1.2.1.3.2. Genetik Algoritma Operatörleri ..... 25
1.2.1.3.2.1. Kodlama ..... 26
1.2.1.3.2.2. Seçim ..... 26
1.2.1.3.2.3. Çaprazlama ..... 27
1.2.1.3.2.4. Mutasyon ..... 29
1.2.1.3.3. Genetik Algoritmalarda Parametre Seçimi ..... 30
1.2.1.3.3.1. Popülasyon Büyüklüğü ..... 30
1.2.1.3.3.2. Üreme Parametresi ..... 30
1.2.1.3.3.3. Çaprazlama Olasılığı ..... 31
1.2.1.3.3.4. Mutasyon Olasılığı ..... 31
1.2.1.3.3.5. Kuşak Aralığı ..... 31
1.2.1.3.3.6. Seçim Stratejisi ..... 32

1.2.1.3.3.7. Fonksiyon Ölçeklemesi ..... 32
1.2.1.4. Diferansiyel Gelişim Algoritması ..... 32
1.2.1.5. Yapay Bağışıklık Algoritması ..... 35
1.2.2. Sürü Zekası Temelli Algoritmalar ..... 41
1.2.2.1. Karınca Kolonisi Optimizasyonu Algoritması ..... 45
1.2.2.2. Parçacık Sürü Optimizasyonu Algoritması ..... 52
1.2.2.3. Yapay Arı Kolonisi Algoritması ..... 56
**2. YAPAY ARI KOLONİSİ ALGORİTMASI ..... 57**
2.1. Doğada Bulunan Arılar ..... 57
2.2. Doğadaki Arıların Besin Arama Davranışları ..... 62
2.3. Yapay Arı Kolonisi Algoritması ..... 65
2.3.1. Başlangıç Besin Kaynaklarının Belirlenmesi ..... 68
2.3.2. Görevli Arıların Besin Kaynaklarına Yönlendirilmesi ..... 68
2.3.3. Gözcü Arıların Besin Kaynaklarını Seçiminde Kullanacakları Olasılık Değerlerinin Hesaplanması ..... 69
2.3.4. Gözcü Arıların Besin Kaynaklarını Seçmeleri ..... 70
2.3.5. Besin Kaynaklarının Terk Edilmesi ve Kaşif Arı Üretimi ..... 70
2.4. Yapay Arı Kolonisi Algoritması ile İlgili Yapılan Çalışmalar ..... 71
**3. PORTFÖY OPTİMİZASYONU ..... 77**
3.1. Yatırım ve Portföy Kavramı ..... 77
3.2. Getiri ve Risk ..... 78
3.2.1. Menkul Kıymetlerde Getiri ..... 81
3.2.1.1. Dönemlik Getiri ..... 81
3.2.1.2. Beklenen Getiri ..... 82
3.2.2. Menkul Kıymetlerde Risk ..... 83
3.2.2.1. Standart Sapma ve Varyans ..... 83
3.2.2.2. Değişim Katsayısı ..... 85
3.2.3. Menkul Kıymetlerin Birlikte Hareket Ölçüleri ..... 86
3.2.3.1. Kovaryans ..... 86
3.2.3.2. Korelasyon Katsayısı ..... 87
3.2.4. Portföyün Beklenen Getirisi ..... 88
3.2.5. Portföyün Riski ..... 89

3.3. Portföy Yönetimi ve Portföy Yönetimi Yaklaşımları ..... 90
3.3.1. Geleneksel Portföy Yöntemi Yaklaşımı ..... 94
3.3.2. Modern Portföy Yönetimi Yaklaşımı ..... 96
3.3.2.1. Markowitz Ortalama-Varyans Modeli ..... 101
3.3.2.2. İndeks Modelleri ..... 103
3.3.2.2.1. Tekli İndeks Modeli ..... 103
**4. UYGULAMA ..... 107**
4.1. Kardinalite Kısıtlı Ortalama Varyans Modeli ..... 107
4.2. Kardinalite Kısıtlı Ortalama Varyans Portföy Optimizasyonu Problemine Yapay Arı Kolonisi Algoritması Yaklaşımı ..... 110
4.3. Parametre Seçimi ..... 117
**SONUÇLAR VE ÖNERİLER ..... 119**
**KAYNAKÇA ..... 132**
**ÖZGEÇMİŞ ..... 144**

# ŞEKİLLER LİSTESİ


Şekil 1.1. [a, b] aralığında tanımlı $f(x)$ fonksiyonu.....9

Şekil 1.2. Optimizasyon metotlarının sınıflandırılması.....11

Şekil 1.3. Negatif seçim algoritması ile model tanıma a) detektör kümesini oluşturma, b) istenmeyen (nonself) maddelerin varlığını takip etmek.....38

Şekil 1.4. Klonal seçim algoritması.....40

Şekil 1.5. Karıncaların en kısa yolu bulmalarının gösterimi.....46

Şekil 1.6. Yıllara göre YAK ile yapılmış yayın sayısı.....56

Şekil 2.1. Arılarda görev dağılımı.....61

Şekil 2.2. Arıların besin arama davranışları.....64

Şekil 3.1. Risk bileşenleri.....79

Şekil 3.2. Toplam riskin kaynakları.....81

Şekil 3.3. Risk karşısında yatırımcı tipleri.....91

Şekil 3.4. Portföy yönetimi aşamaları.....93

Şekil 3.5. Kayıtsızlık eğrileri.....98

Şekil 3.6. Etkin set (Etkin sınır).....99

Şekil 3.6. Yatırımcının etkin sınır eğrisi üzerindeki yeri.....100

Şekil 4.1. Hang Seng karşılaştırma problemine ait standart etkin sınır.....109

Şekil 4.2. KKOV modeli için YAKA yaklaşımı.....114

Şekil 4.3. Her bir besin kaynağı için düzeltme fonksiyonu.....116

Şekil 5.1. BIST 30 endeksine ilişkin standart etkin sınır.....121

Şekil 5.2. Hang Seng endeksi için sezgisel etkin sınırlar.....124

Şekil 5.3. DAX100 endeksi için sezgisel etkin sınırlar.....125

Şekil 5.4. FTSE 100 endeksi için sezgisel etkin sınırlar.....126

Şekil 5.5. S&P 100 endeksi için sezgisel etkin sınırlar.....127

Şekil 5.6. Nikkei endeksi için sezgisel etkin sınırlar.....128

Şekil 5.7. BİST 30 endeksi için sezgisel etkin sınırlar.....129

Şekil 5.8. Problemlere ilişkin birleştirilmiş etkin sınırlar.....130

# TABLOLAR LİSTESİ

Tablo 4.1. Denenen parametre değerleri ve seçilen değerler .....117
Tablo 5.1. Test verilerine ilişkin hisse senedi sayıları .....119
Tablo 5.2. BIST 30 endeksinde işlem gören hisse senetleri.....120
Tablo 5.3. Beş sezgisel tekniğe ait uygulama sonuçları.....122

# KISALTMALAR LİSTESİ

AKOA : Arı Kolonisi Optimizasyonu Algoritması
CR : Çaprazlama Sabiti
DG : Diferansiyel Gelişim
DGA : Diferansiyel Gelişim Algoritması
EP : Evrimsel Programlama
ES : Evrimsel Strateji
F : Ölçekleme Faktörü
GA : Genetik Algoritma
KARP : Kapasite Kısıtlı Araç Rotalama Problemi
KKA : Karınca Kolonisi Algoritması
KKOV : Kardinalite Kısıtlı Ortalama Varyans
MF : Marjinal Fayda
NP : Popülasyon Büyüklüğü
PO : Portföy Optimizasyonu
PSO : Parçacık Sürü Optimizasyonu
TA : Tabu Arama
TB : Tavlama Benzetimi
YAK : Yapay Arı Kolonisi
YAKA : Yapay Arı Kolonisi Algoritması

# GİRİŞ

İnsanoğlu eline geçen en küçük bir varlığı değerlendirmek için birçok yola başvurmaktadır. Yatırımcı parasını herhangi bir gayrimenkul satışına yatırabileceği gibi borsaya girip hisse senedi de alabilir. Yatırımcı parasını nasıl değerlendirmesi gerektiği tam olarak bilmemektedir. Bu durum, konuyu önemli bir problem konusu haline getirmektedir ve karşımıza önemli optimizasyon problemlerinden biri olan Portföy Optimizasyonu Problemi çıkmaktadır. Bu problemin çözümüne ilk olarak Markowitz bir yöntem sunmuştur. Başarılı bir yöntem olmasına rağmen portföy içindeki çeşitliliğinin artması metodun çözümünü karmaşık hale getirmiştir. Daha sonraları çeşitli araştırmacılar tarafından yeni yöntemler de önerilmiştir. Fakat bu yöntemler ani kararların verilmesi gereken durumlarda aciz kalmıştır. Bu durumda kesin sonucu vermeyen ama kesin sonuca yakın değerler üreten sezgisel teknikler kullanılmaya başlanmıştır. Bu yöntemler bir çok optimizasyon problemi türü için kullanılmıştır. Kolay uygulanabilir olmaları ve hızlı, etkin sonuç üretmeleri araştırmacılar tarafından tercih sebebi olmuştur. Sezgisel teknikler genel olarak gelişime dayalı sezgisel teknikler ve sürü zekası temelli sezgisel teknikler olmak üzere ikiye ayrılmaktadır. Gelişime dayalı teknikler, sürekli bir iyileştirmenin olduğu ve en iyinin saklandığı yöntemlerdir. Sürü zekası temelli teknikler ise, ortak hareket etmenin ve haberleşmenin olduğu yöntemleri içerir.

Doğaya baktığımızda bir sürü canlının hayatta kalma mücadelesine tanık olabiliriz. Canlıların özellikle sürü mantığına sahip hayvanların besin bulma yarışında verdikleri mücadele ve bir engel karşısında ürettikleri çözümler araştırmacıları, bu konuyu detaylı olarak incelemeye yöneltmiştir. Yapılan çalışmalar sonucunda canlıların aslında günlük yaşantılarında optimizasyon yaptıklarını görmüşlerdir. Bu durum araştırmacıların kafasında bir soru oluşturmuştur: “Acaba canlıların besin arama davranışlarını modelleyerek çok karmaşık problemlere çözüm bulabilir miyiz?” Hayvanların besin arama davranışları ve arama sırasında kullandıkları sürü zekası araştırmacıların yeni yöntemler bulmasında ilham kaynağı olmuşlardır. Sürü zekasını kullanan yöntemlerden en bilinenleri karıncaların besin

arama davranışlarını kullanan Karınca Kolonisi Algoritması, kuşların besin arama ve göç hareketlerini içeren Parçacık Sürü Optimizasyonu ve son dönemlerin popüler yöntemi olmaya başlayan ve arıların besin arama davranışlarından esinlenen Yapay Arı Kolonisi Algoritmasıdır. Bu yöntemler bir çok optimizasyon problemleri üzerinde denenmiştir ve oldukça etkili ve hızlı sonuçlar üretmektedirler.

Çalışmanın birinci bölümünde optimizasyon ve sürü zekası kavramlarından bahsedilecektir. Optimizasyon tekniklerinden kısaca bahsedilip sezgisel teknikler derinlemesine incelenecektir.

İkinci bölümde önemli optimizasyon problemlerinden biri olan portföy optimizasyonu probleminden bahsedilecektir. Konuyla ilgili temel kavramlar tanıtıldıktan sonra probleme ilişkin modeller tanıtılacaktır.

Üçüncü bölümde son yıllarda kullanım sıklığı artan ve sürü zekasını temel alan yeni bir yöntem olan Yapay Arı Kolonisi Algoritmasından bahsedilecektir. Algoritmanın genel yapısı tanıtılıp, algoritmanın kullanıldığı son yıllardaki çalışmalardan bahsedilecektir.

Dördüncü bölümde Yapay Arı Kolonisi Algoritmasının, Portföy Optimizasyonu Problemindeki uygulamasına yer verilmektedir. Bunun için literatürde bulunan 5 tane veri setinden ve gerçek veri olan BIST 30 verilerinden yararlanılmıştır.

Beşinci bölümde elde edilen sonuçlar ve karşılaştırmalı grafikler gösterilecektir. Sonuçlar üzerinde yorumlar yapılıp çeşitli öneriler sunulacaktır.

# 1. OPTİMİZASYON VE SÜRÜ ZEKASI

## 1.1. Optimizasyon Kavramı

Günümüzde bilgisayarların etkinliğinin hızla gelişmesi ve aynı paralelde yazılımların gelişimiyle optimizasyon teorisi, metot ve algoritmalarına ilgi giderek artmaktadır. Optimizasyon, birçok disiplinin (mühendislik, işletme ve fen bilimleri) ilgi alanına girmiş ve daha önceleri çözümü zor olarak kabul edilen problemlerin çözümüne katkı sağlamıştır[1].

Optimizasyon insanların günlük yaşantılarında hatta doğanın sıklıkla uyguladığı bir kavram olup fiziksel sistem analizinde ve karar biliminde kullanılan önemli bir araçtır[2]. Matematiksel olarak tanımlanmış problemlerin "en iyi" (best) çözümlerini belirleyen bir bilim dalıdır[3]. Bir başka ifade ile girdilerin maksimum ya da minimum çıktısını bulmaya yarayan matematiksel bir süreç olarak tanımlanmaktadır[4].

Optimizasyon süreci, 'iyi' ve 'kötü" değişiminin ölçüldüğü durumlardaki 'en iyi'yi bulma sürecidir. Günümüzde gelirin veya kârın 'en fazla' yada 'maksimum' olması amaçlanırken; maliyet yada harcamaların da 'en az' yada 'minimum' olması istenir. Bu durumda "optimum" kavramı bu iki durumu da kapsamaktadır. Optimizasyon teorisi, bu değerin bulunması ile ilgili metotları ve optimalliği araştıran sayısal çalışmaları içeren matematiğin bir koludur. Ayrıca optimizasyon teorisi, 'en uygun' (optima) değerin hesaplanmasında kullanılabilecek teknikler, metotlar, prosedürler ve algoritmalar bütünüdür[5].

---

[1] Andreas Antoniou ve Wu-Sheng Lu, **Practical Optimization Algorithms and Engineering Application**, SpringerScience+BusinessMedia, New York, 2007, s.xv.
[2] Jorge Nocedal ve Stephen J. Wright, **Numerical Optimization**, Springer-Verlag, New York, 1999, s.1.
[3] Roger Flecther, **Practical Methods of Optimization**, Second Edition, John Wiley&Sons, 2000, s.3.
[4] Randy L. Haupt ve Sue A. Haupt, **Practical Genetic Algorithms**, John Wiley & Sons Inc., Second Edition, 2004, s.2.
[5] Andreas Antoniou ve Wu-Sheng Lu, a.g.e., 2007, s.1-2.

Optimizasyon konusunda dört farklı yaklaşımlar söz konusudur. Bunlar analitik metotlar, grafik metotlar, deneysel metotlar ve nümerik metotlar olarak tanımlanmaktadır. Analitik metotlar, klasik diferansiyel hesaplamaları esas alır. Çözümünde fonksiyonun türeviyle ilgilenmektedir. Problemin matematiksel olarak ifade edilmesi şarttır. Bilgisayarla çözüm gerektirmemektedir. Bağımsız değişkenlerin 2 veya 3 değişkeni aştığı problemlerin çözümünde ve yüksek dereceli lineer olmayan problemlerde bu yöntem uygulanamamaktadır.

Grafik metotta maksimum veya minimum yapılacak amaç fonksiyonunun grafik yardımıyla çizimi yapılır. Değişken sayısının ikiyi geçtiği problemlerde kullanılmamaktadır. Pratikte kullanılan uygulamalardaki fonksiyonların değişken sayılarının fazlalığından grafik metodun kullanımı sınırlıdır.

Bazı durumlarda sistemin optimum performansı direk deneylerle bulunmaktadır. Öncelikle sistem kurulur ve değişkenler teker teker ayarlanır. Böylece her durum için performans kriterleri hesaplanır. Bu metotta optimum ya da optimuma yakın sonuçlar elde edilebilir. Fakat birbirini etkileyen birden fazla değişken mevcut ise metodun önerdiği çözümler güvenilir olmayabilir.

Optimizasyon konusundaki en önemli yaklaşımlar nümerik metotları temel alır. Bu yaklaşımda optimizasyon problemlerine adım adım gelişen çözümler sunabilmek için aşamalı numerik prodesürler kullanır. Bazı yakınlaşma kriterleri sağlanana kadar metot uygulanmaktadır. Bu yaklaşım analitik olarak çözümü bulunamayan karmaşık optimizasyon problemlerinin çözümünde kullanılmaktadır. Bilgisayarda kolaylıkla programlanabilmektedir. Bu özellikleri sayesinde diğer yaklaşımların yerine geçmektedir.

Nümerik optimizasyon metotlarının teori ve uygulamalarını kapsayan disiplin matematiksel programlama olarak da bilinmektedir. Geçtiğimiz 40 yıl içinde gelişen matematiksel programlama modelleri aşağıda tanımlanmaktadır[6]:

- Doğrusal programlama,
- Tamsayılı programlama,
- Kuadratik (Quadratic) programlama,
- Doğrusal olmayan programlama
- Dinamik programlama

Optimizasyon modeli, amaç fonksiyonu, karar değişkenleri ve kısıtlardan oluşur ve kısıtları ihlal etmeden amaç fonksiyonunu en uygun duruma getiren karar değişkenlerinin hangi değerler olduğunu araştırır. *Amaç Fonksiyonu* modelde minimum ya da maksimum yapılmaya çalışılan fonksiyon olarak tanımlanır. Buna örnek olarak bir şirketin karını maksimize etmesi ya da maliyetlerini minimize etmesi verilebilir. Sistemin performansını etkileyen ve kontrol altında tutulan değişkenlere *Karar Değişkenleri* adı verilir. *Kısıtlar* ise karar değişkenlerinin alabileceği değerlerin sınırlarını ifade eden fonksiyonlardır[7].

Etkin bir optimizasyon modeli aşağıdaki adımlarla matematiksel ifadeye dönüşmektedir[8]:

1. *Karar Değişkenlerinin Belirlenmesi ve Tanımlanması:*

Modelin kurulmasındaki ilk ve en önemli adımı oluşturmaktadır. Probleme göre tanımlanan değişkenler olduğundan iyice araştırılıp belirlenmesi gerekmektedir. Tasarım değişkenleri olarak da adlandırılan karar değişkenleri bağımlı ve bağımsız değişkenler diye ikiye ayrılmaktadır.

[6] Andreas Antoniou ve Wu-Sheng Lu, a.g.e., 2007, s.2.
[7] Wayne L. Winston, **Operations Research: Applications and Algorithms**, Forth Edition, Brooks/Cole, 2004, s. 2.
[8] Jasbir S. Arora, **Guide to Structural Optimization**, American Society of Civil Engineers, 1997, s.2-3.

*2. Amaç Fonksiyonunun Belirlenmesi ve Tanımlanması:*

Karar (Tasarım) değişkenlerine direk ya da doğrudan bağlı olmayan amaç fonksiyonu skaler bir fonksiyondur. Modelde optimize edilecek olan fonksiyondur ve problem türüne göre bu fonksiyon maksimum (kar) veya minimum (maliyet) yapılmaya çalışılmaktadır.

*3. Kısıtların Belirlenmesi ve Tanımlanması:*

Karar değişkenlerinin sonsuz değer almamaları için sınırlandırılması gerekmektedir. Buna örnek olarak kaynak kısıtı, zaman kısıtı verilebilir. Ayrıca karar değişkenlerinin belirli değerleri alması engellenerek sınırlandırılabilir. Problem türüne göre değişim gösteren kısıt ve sınırlanmalar optimizasyon modelinde tanımlanmalı ve sağlanmalıdır.

Optimizasyon modelinin matematiksel ifadesi aşağıda tanımlanmaktadır[9]:

$x$, karar değişkenleri vektörü,

$f$, minimum ya da maksimum yapılmak istenen $x$'e bağlı fonksiyon, amaç fonksiyonu.

$c$, kısıtlar vektörü, Bu vektör, $x$ değişkenlerinin fonksiyonudur.

$$\min_{x \in R^n} \; f(x) \;\; kısıtlar \begin{cases} c_i(x) = 0, & i \in m \\ c_i(x) \geq 0, & i \in n \end{cases} \qquad (1.1.)$$

Burada $f$ amaç fonksiyonu ve her $c_i$, $x$'e bağlı skaler bir fonksiyondur. $m$, eşitlik kısıtı sayısını ve $n$ ise eşitsizlik kısıtı sayısını ifade etmektedir.

Bazı optimizasyon problemlerinde örneğin ulaştırma problemlerinde karar değişkenleri tamsayı değerini almaktadır. Problemin çözümü önce gerçel sayılarla bulunur daha sonra optimale yakın çözümü veren tamsayılı değere yuvarlatılarak karar değişkenlerinin değerleri elde edilir. Bu tür problemlerin çözümü ayrık optimizasyon araçları kullanılarak bulunabilir. Denklem 1.1. ifadesine Denklem

[9] Jorge Nocedal ve Stephen J. Wright, a.g.e., 1999, s.2.

1.2.'de gösterilen kısıt eklenerek karar değişkenlerinin tamsayılı ifadeler alması sağlanır. Bu tür problemlere *tamsayılı programlama problemi* adı verilmektedir.

$$x_{ij} \in Z, \quad tüm\ i\ ve\ j\ değerleri\ için \tag{1.2.}$$

Problemin çözümü sonlu sayıda setlerden oluşuyorsa bu tür problemlere *kesikli* (*ayrık) optimizasyon problemleri* adı verilirken, sonsuz sayıda setlerden oluşan çözümlere sahip ise *sürekli optimizasyon problemleri* adı verilmektedir. Bir başka deyişle karar değişkenleri çözüm uzayında belirli değerleri alabiliyorsa kesikli (ayrık) optimizasyon, herhangi bir değeri alabiliyorsa sürekli optimizasyon olur. Hem gerçel sayılı hem de tamsayılı değerleri içeren çözümlerin bulunduğu problem türüne ise *karma tamsayılı programlama problemi* adı verilir.

Problemin amaç ve kısıt fonksiyonlarının her ikisi de doğrusal bir fonksiyonsa probleme *doğrusal programlama problemi*, en az biri doğrusal fonksiyon değilse *doğrusal olmayan programlama problemi* adı verilmektedir[10].

Optimizasyon probleminin çözümü zamana bağlı olarak değişmiyorlarsa bu tür optimizasyon problemlerine *statik optimizasyon* denir. Bu tür problemlerde matematiksel model, cebirsel ve soyut denklemlerden oluşur. Eğer çözüm zamana bağlı olarak değişen bir problemin çözümü ise bu tür optimizasyon modellerine *dinamik optimizasyon* adı verilmektedir. Bu tür problemler genellikle diferansiyel denklemler içeren ve amaç fonksiyonları integral olan modellerdir.

Bir çok optimizasyon problemi, maksimize ya da minimize edilmeye çalışılan amaç fonksiyonunun yanında yerine getirilmesi gereken kısıtları içerir. Eğer problemin karar değişkenlerini etkileyen bir kısıt fonksiyonu yok ise bu tür problemlere *kısıtsız optimizasyon problemleri,* var ise *kısıtlı optimizasyon problemleri* adı verilmektedir. Kısıtlı optimizasyon problemleri, kısıtın işaretine

[10] Jorge Nocedal ve Stephen J. Wright, a.g.e., 1999, s.4-5.

bağlı olarak (=) ise eşitlik kısıtlı ve ($\leq$ veya $\geq$) ise eşitsizlik kısıtlı optimizasyon problemi adını almaktadır[11].

Kısıtlanmış ve kısıtlanmamış fonksiyonların maksimum ve minimum (uç) noktalarını belirleyebilmek için diferansiyel hesabı kullanılmaktadır. Kısıtlanmamış en uç noktaların belirlenmesi için gerekli ve yeterli koşullar varken, eşitlik kısıtlı problemler için Langranj veya Jakobiyen yöntemleri, eşitsizlik kısıtlı problemler için ise Kuhn-Tucker koşulları ile çözüm bulunmaktadır[12,13].

Bir fonksiyonun $f(x)$ uç noktası, fonksiyonun maksimum veya minimum noktalarından biri olabilir. Matematiksel ifade ile tüm $j$'ler için $|h_j|$ yeterince küçük sayı olacak şekilde tüm $h = (h_1, \dots, h_j, \dots, h_n)$ değerleri için

$$f(X_0 + h) \leq f(X_0) \tag{1.3.}$$

ise, $X_0 = (x_1, \dots, x_j, \dots, x_n)$ noktası maksimumdur. Bu tür maksimuma *zayıf maksimum* da denilmektedir. Eğer ifade " $\leq$ " yerine " $<$ " olursa bu maksimuma *güçlü maksimum* denir.

Benzer şekilde, eğer $h$ için

$$f(X_0 + h) \geq f(X_0) \tag{1.4.}$$

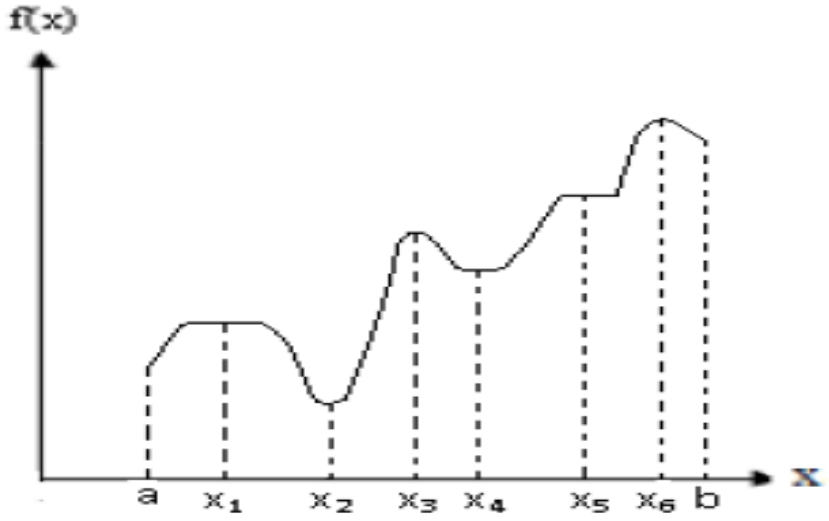
olarak tanımlı ise bu durumda $X_0 = (x_1, \dots, x_j, \dots, x_n)$ noktası minimumdur. Bu tür minimuma *zayıf minimum* da denilmektedir. Eğer ifade " $\geq$ " yerine " $>$ " olursa bu minimuma *güçlü minimum* denir.

---

[11] David A. Wismer ve R. Chattergy, **Introduction to Nonlinear Optimization: A Problem Solving Approach**, North-Holland Series, New York, 1978, s. 1-2.
[12] David A. Wismer ve R. Chattergy, a.g.e., 1978, s. 33-115.
[13] Hamdy A. Taha, **Yöneylem Araştırması**, Çev. Baray, Alp Ş. ve Esnaf, Şakir, 6. Baskıdan Çeviri, Literatür Yayıncılık, İstanbul, 2000, s. 737-771.

Tek değişkenli $f(x)$ fonksiyonuna ait [a,b] aralığında tanımlı maksimum ve minimum noktaları Şekil 1.1'de gösterilmektedir. $x_1, x_3$ ve $x_6$ fonksiyonun maksimum noktalarını belirtirken, $x_2$ ve $x_4$ noktaları ise minimum noktalarını belirtmektedirler. Maksimum uç noktalar arasında en yüksek değere sahip $f(x_6)$ noktası olduğu için bu noktaya *global veya mutlak maksimum*, diğer iki nokta olan $f(x_1)$ ile $f(x_3)$ noktalarına ise *lokal veya görece maksimum* adı verilmektedir. Benzer şekilde $f(x_4)$, lokal minimum ve $f(x_2)$, global minimumdur[14].

**Şekil 1.1. [a, b] aralığında tanımlı $f(x)$ fonksiyonu**




**Kaynak:** Hamdy A. Taha, a.g.e., 2000, s.738.

Bazı algoritmalar değişkenlerin belirli bir başlangıç değerlerini kullanarak maliyeti minimize etmeye çalışırlar. Hızlı çalışan bu algoritmalar kolaylıkla yerel minimuma takılabilirler. Geleneksel optimizasyon algoritmaları da denilen bu algoritmalar genellikle cebiri kullanır. Rasgele metotlar ise değişken değerlerini bulmak için bazı olasılıksal hesaplamaları kullanırlar. Yavaş çalışmasına rağmen global minimum noktasını bulmakta diğer modelden daha iyidir[15].

[14] Hamdy A. Taha, a.g.e., 2000, s.737.
[15] Randy L. Haupt ve Sue A. Haupt, a.g.e., 2004, s.5.

Lokal ve global optimumların bulunmasında kullanılan optimizasyon metotlarına, tek boyutta minimizasyon yapan (Golden Section), çok boyutta arama yapan türeve dayalı olmayan (konjuge yönler), birinci türeve dayalı teknikler (Quassi Newton), ikinci türeve dayalı teknikler (Newton, Levenberg, Mardquardt) ve global optimizasyon teknikleri olarak da adlandırılan modern sezgisel algoritmalar (ısıl işlem, tabu arama, genetik algoritma vb) örnek verilebilir[16].

Optimizasyon metotları, klasik teknikler ve modern sezgisel teknikler olarak sınıflandırılmaktadır (Şekil 1.2.). Klasik teknikler arama uzayını tamamen araştıran ve kesin sonucu polinominal zamanda bulabilen tekniklerdir. Global optimumu garanti eden yöntemlerdir. Problem boyutunun yüksek olduğu ya da çözüm ile uygunluğu arasındaki ilişkinin karmaşık olduğu durumlarda çözüm bulabilmesi zorlaşmaktadır[17]. Klasik teknikler, analitik ve yapısal metotlar olarak iki alt gruba ayrılmaktadır. Analitik metotlara örnek olarak doğrusal programlama, türeve dayalı algoritmalar ve bölgesel araştırma metotları verilebilir. Yapısal modeller ise dinamik programlama, ayır ve bağla, böl ve keşfet gibi metotlardan oluşur.

Modern sezgisel teknikler ise kendi aralarında deterministik ve olasılık temelli olarak ikiye ayrılmaktadır. Tabu arama algoritması, deterministik metotlar grubuna girmektedir. Olasılık Temelli metotlar ise tek çözümden iteratif olarak gelişen ve birden çok çözümden (popülasyon) iteratif olarak geliştirilen metotlar diye iki alt başlıkta incelenmektedir[18]. Bu teknikler Monte Carlo temelli algoritmalardır. Tüm arama uzayına bakmazsızın sezgisel yöntemler yardımıyla bir sonraki adımdaki çözüm ve/veya çözümleri belirleyerek global optimum nokta ve/veya noktaları bulmaya çalışmaktadır. Kısa zamanda çözüme ulaştıran bu yöntemler global optimum noktayı bulmayı garanti etmez[19].

---

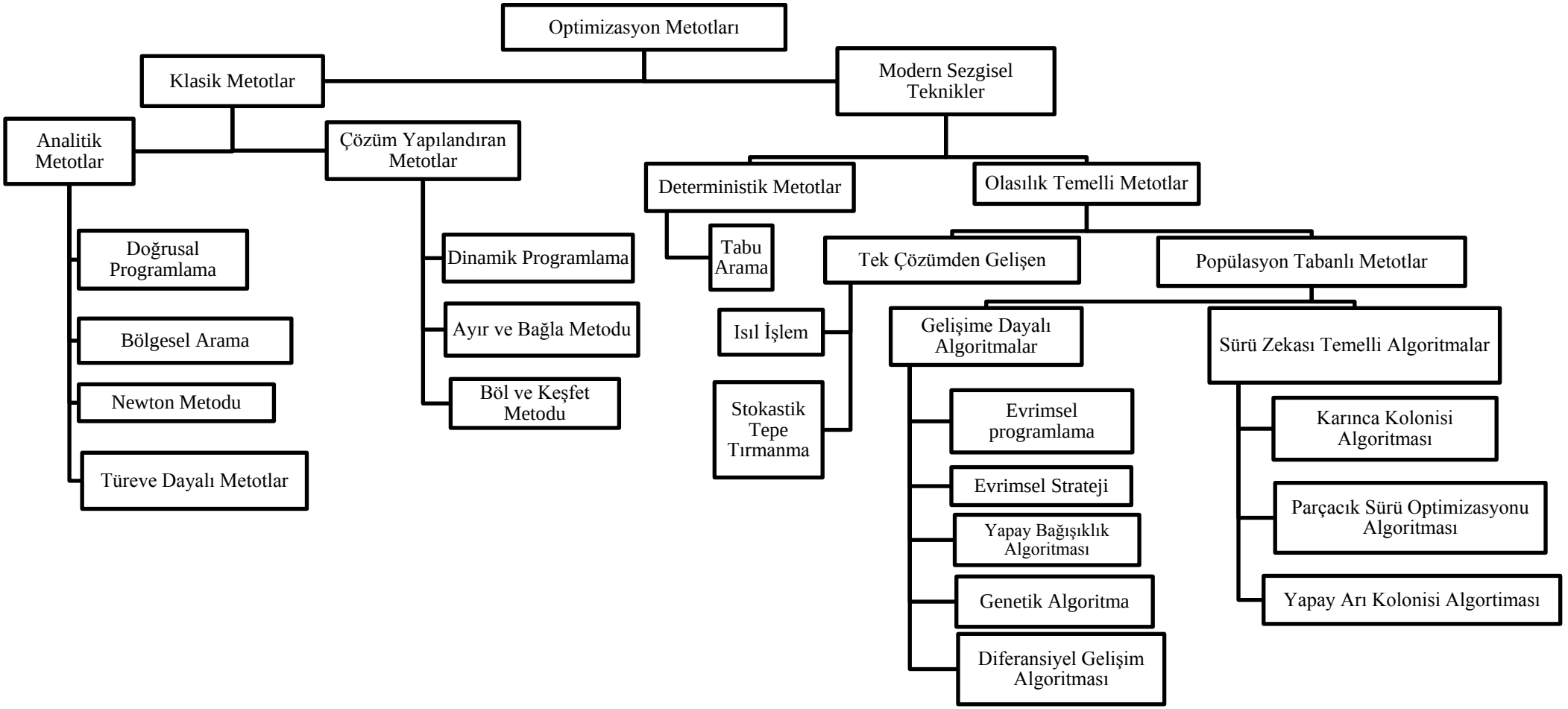
[16] Bahriye Akay, "**Nümerik Optimizasyon Problemlerinde Yapay Arı Kolonisi (Artificial Bee Colony) Algoritmasının Performans Analizi**", Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Yayımlanmamış Doktora Tezi, 2009, s.9.

[17] T. Weise, **Global Optimization Algorithms: Theory and Application**, www.it-weise.de, Second Edition, 2009, s.22.

[18] Bahriye Akay, a.g.e., 2009, s.9.

[19] T. Weise, a.g.e., 2009, s.22.

**Şekil 1.2. Optimizasyon metotlarının sınıflandırılması**




**Kaynak:** Bahriye Akay, a.g.e., 2009, s.11.



11

## 1.2. Optimizasyon Problemlerinde Kullanılan Popülasyon Tabanlı Sezgisel Teknikler

Optimizasyon problemlerinde her problem türüne göre oldukça geniş çözüm yöntemleri kullanılmaktadır. Bu tezde yapay arı kolonisi algoritması uygulaması yapılacaktır. YAK, popülasyon tabanlı bir sezgisel yöntemdir. Bu yüzden son zamanlarda sıklıkla kullanılan popülasyon tabanlı optimizasyon tekniklerinden bahsedilecektir. Popülasyon tabanlı sezgisel teknikler, gelişime dayalı algoritmalar ve sürü zekası temelli algoritmalar olmak üzere iki sınıfa ayrılabilir (Şekil 2.2.). Gelişime dayalı algoritmalara evrimsel strateji, evrimsel programlama, genetik algoritma, genetik programlama, diferansiyel gelişim algoritması ve yapay bağışıklık algoritması örnek verilebilir. Sürü zekası temelli algoritmalar sınıfına ise karınca kolonisi algoritması, parçacık sürü optimizasyonu algoritması ve yapay arı kolonisi algoritması girmektedir. Bu çalışmada genetik programlama dışındaki tüm algoritmalara kısaca değinilecektir.

### 1.2.1. Gelişime Dayalı Algoritmalar

Doğal evrim prensiplerini esas alan arama tekniklerinin uygulamaları ve gelişimi günümüzde oldukça popüler ve önemli bir alandır. Evrimsel hesaplama teknikleri (Evrimsel Algoritmalar) olarak adlandırılan gelişime dayalı algoritmalar, Darwin'in önerdiği "en iyinin hayatta kalma" prensibini temel alır[20]. Evrimsel Algoritmalar, bu evrimsel prensipleri, optimal çözümü arayabilen algoritmalarda özetlemişlerdir. Bu algoritmalar problem için uygun çözümler bulur ve mümkün zaman içinde optimal sonuca ulaşırlar[21].

Evrimsel hesaplama teknikleri, hemen hemen bütün mühendislik alanlarında kullanılan karar destek sistemleri, yapay zeka, nümerik optimizasyon ve biyolojiyi de içine alan disiplinler arası bir çalışma alanıdır. Bu algoritmalar, popülasyonlardaki bireylerin kolektif öğrenme prosesini modellemektedirler. Popülasyondaki bireyler

[20] S. Sumathi ve dğr., **Evolutionary Intelligence**, Springer-Verlag, Coimbatore, India, 2008, s.1.
[21] S. Sumathi ve dğr., a.g.e., 2008, s.2.

sadece potansiyel çözüm alanındaki bir noktayı göstermez, bunun yanında doğanın "kanunu" ile ilgili mevcut bilgisinin geçici taşıyıcısı rolündedir[22]. Doğanın uyguladığı "en iyi olan hayatta kalsın: zayıf olan ölsün" kuralını temel alan doğal seleksiyon gibi evrimsel süreçlerin bilgisayar yardımıyla modellenmesi sonucunda evrimsel hesaplama teknikleri ortaya çıkmıştır.

1960'ların ortalarından günümüze kadar adım adım geliştirilen evrimsel hesaplama teknikleri temel anlamda üç ana yöntemle açıklanabilir. Bu yöntemler, Lawrence Fogel[23] tarafından ortaya atılan *evrimsel programlama*, Holland[24] tarafından geliştirilen *genetik algoritma* ve Bienert, Rechenberg ve Schwefel[25] adlı üç arkadaşın geliştirdikleri *evrimsel strateji*'dir[26]. Bu üç temel yaklaşım dışında bireylerin program kodları olduğu, programların geliştirilmesinde kullanılan genetik algoritma temelli yöntem *genetik programlama*[27], genetik algoritmadan farklı yeniden üretim mekanizması kullanan *diferansiyel gelişim algoritması* ve bireylerin genetik ve fiziksel gelişimlerini kültürün nasıl etkilediğinden yola çıkılarak geliştirilen *kültürel gelişim algoritması* (Cultural Evolution) örnek verilebilir[28,29].

Evrimsel hesaplama teknikleri, evrimin kullandığı süreçleri bilgisayar ortamında simüle ederek problemin çözüm aşamaları olarak kullanmaktadır. Bu teknikler, seçim (selection), yeniden birleştirme (recombination), mutasyon, yer değiştirme (migration), çevre (locality) ve komşu (neighbour) gibi kurallar zinciridir.

---

[22] Thomas Bäck, **Evolutionary Algorithms in Theory and Practice**, Oxford University Press, New York, First Edition, 1996, s.7.
[23] L.J. Fogel ve dğr., **Artificial Intelligence through Simulated Evolution**, Wiley, New York, 1966.
[24] J.H. Holland, **Nonlinear Enviroment Permitting Efficient Adaptation**, Computer and Information Science II, New York, 1967.
[25] I. Rechenberg, **Cybernetic Solution Path of an Experimental** Problem, Royal Aircraft Establishment Library Translation 1122, 1965.
[26] Thomas Bäck ve dğr., **Evolutionary Computation 1 Basic Algorithms and Operators**, Bristol and Philadelphia: Institute of Physics Publishing, 2000a, s.41.
[27] J.R. Koza, **Genetic Programming: On the Programming of Computers by Means of Natural Selection**, MA:MIT Press, Cambridge, 1992.
[28] Andries Engelbrecht, **Computational Intelligence**, John Wiley & Sons Inc., Second Edition, 2007 s.8,.
[29] S.N. Sivanandam, ve S.N Deepa, **Introduction to Genetic Algorithm**, Springer-Verlag, Berlin, 2008, s.2.

Popülasyondaki her bir birey için çevreye adaptasyon ölçüsü vardır. Bu ölçü uygunluk değeri veya amaç fonksiyonu olarak adlandırılmaktadır. Seçim, bu uygunluk değerini kullanarak yüksek uygunluğu olan bireylere odaklanır. Yeniden birleştirme ve mutasyon, gelecek nesil için bireylerin farklılaşmasını sağlar.

Tipik bir evrimsel hesaplama yönteminde başlangıç popülasyonu oluşturma, değerlendirme, seçim, yeniden üretim ve mutasyon gibi birbirini takip eden bir döngü mevcuttur. Popülasyon büyüklüğü genelde sabit tutulur. Başlangıç popülasyonu da genel olarak tesadüfi olarak belirlenir. Değerlendirme her bir bireyin belirlenen çevre içindeki adaptasyon (uyum) değerine göre yapılır. Buradaki işlem amaç fonksiyonunun yapısına göre basit ya da karmaşık olabilir. Seçim ise genel olarak iki aşamalı olarak ele alınmaktadır: Ailenin (ebeveyn) seçimi ve hayatta kalma. Ebeveyn seçimi hangi bireylerin ebeveyn olacağını ve onların da kaç çocuğa sahip olacağının belirlenmesini kapsar. Yeni bireyler, ebeveynler arasındaki bilgi değişiminin yapıldığı yeniden birleştirme (recombination) ve yeni bireyleri daha da farklılaştıran mutasyon aşamalarında üretilir. Elde edilen yeni bireyler daha sonra yeniden değerlendirilir ve popülasyonda hangisinin kalacağına karar verilir[30,31,32].

Klasik evrimsel algoritmaya ait sözde-kod aşağıda tanımlanmıştır[33]:

t=0;
*Başlangıç Popülasyonu Oluştur* P(t);
*Değerlendirme* P(t);
until (Doğru){
t=t+1;
*Seçim* P(t);
*Yeniden Birleştirme* P(t);
*Mutasyon* P(t);

[30] James Kennedy ve dğr., **Swarm Intelligence**, Morgan Kaufmann Publishers, 2001, s.144.
[31] S. Sumathi ve dğr., a.g.e., 2008, s.31-32.
[32] Kemal Çakar, **Genetik Algoritmalar Yardımıyla Acil Servis İstasyonu Yerleşiminin Optimizasyonu**, Atatürk Üniversitesi, Sosyal Bilimler Fakültesi, Yayınlanmamış Doktora Tezi, 2009, s.19.
[33] S. Sumathi ve dğr., a.g.e., 2008, s.11.

*Değerlendirme* P(t);

*Hayatta Kalma* P(t);

}

Evrimsel algoritma kullanımının diğer global optimizasyon tekniklerine göre avantajları aşağıda sıralanmıştır[34,35]:

✓ Evrimsel algoritmalar kavramsal olarak basittir. Algoritma başlangıç popülasyonu oluşturma, aşamalı değişim ve seçimi kapsar.

✓ Evrimsel algoritmaların performansı, yalnızca sürekli veya kısıtlı değişkenler ile uygulanabilen diğer bazı nümerik tekniklerin aksine gösterimden bağımsızdır.

✓ Evrimsel algoritmaların uygulama alanı geniştir. Matematiksel olarak formüle edilebilen herhangi bir probleme uygulanabilmektedirler.

✓ Evrimsel algoritmalar, problemin önceki bilgilerinin kolaylıkla kıyaslanabildiği bir yapıya sahiptir. Bu gibi bilgilerin kullanılması evrimsel aramaya odaklanarak mevcut çözüm alanının daha etkin biçimde taranması sağlar.

✓ Evrimsel algoritmalar çok sayıda geleneksel optimizasyon teknikleri ile birlikte kullanılabilir. Bu ilk popülasyonun oluşturulmasında kullanılmasının yanında yöntemle eşzamanlı olarak kullanılmasını da kapsamaktadır.

✓ Her bir çözümün değerlendirmesi paralel olarak yapılabilmektedir.

✓ Geleneksel optimizasyon metotlarının aksine evrimsel algoritmalar çözümde meydana gelebilecek ani değişimlere uyarlanabilmektedir.

✓ Evrimsel algoritmaların en önemli avantajı insanların hiçbir bilgisinin olmadığı problemleri şekillendirebilme yeteneğidir. İnsan deneyimi söz konusu olduğunda ise problem çözme mekanizmasının otomatikleşmesinde daha az etkindir.

[34] Kemal Çakar, a.g.e., 2009, s.29-30.
[35] S. Sumathi ve dğr., a.g.e., 2008, s.72.

### 1.2.1.1. Evrimsel Strateji

Biyolojik süreçler, evrim tarafından optimize edilmektedir ve evrim bir biyolojik süreçtir. Bu yüzden evrim kendi kendisini optimize etmektedir, yani evrim evrimleşmektedir diye düşünen Rechenberg 1960'lı yıllarda evrimsel stratejinin temellerini atmış ve daha sonra Schwefel tarafından geliştirilmiştir[36]. Evrimsel Strateji (ES), 1960'lı yıllarda Berlin Teknik Üniversitesi'nde Bienert, Rechenberg ve Schwefel tarafından ortaklaşa geliştirilen bir yöntemdir. Bu alandaki ilk uygulama bükülen bir borunun şekil optimizasyonu, birleştirilen iki plakanın sürtünme minimizasyonu gibi hidrodinamik problemlerin deneysel çözümü olmuştur. Bu tür optimizasyon problemlerini analitik olarak veya geleneksel yöntemlerle çözülememesi sebebi ile deneysel ayarlamalara dayanak teşkil edecek biçimde tesadüfi değişimlere dayalı bir algoritmik metot geliştirilmiştir[37].

Evrimsel stratejilerin önemli özelliklerinden biri mutasyon uygulamalarını kontrol etmek için kendi kendine adapte olan mekanizmalar kullanmasıdır. Bu mekanizmaların arama sürecini optimize etmesi amaçlanır. Bu mekanizma, sadece üzerinde çalışılan problemin çözümlerini değil mutasyona uğramış bu çözümlerin bazı parametrelerini de geliştirmektedir[38].

Evrimsel strateji modelini diğer evrimsel hesaplama tekniklerinden ayıran üç önemli özellik vardır. İlk farklılık bireylerin temsili konusundadır. Evrimsel stratejide bireyler, genetik algoritmada kullanıldığı gibi ikili kodlama sistemi yerine yüzen nokta vektörleriyle bir başka deyişle gerçek değerleriyle temsil edilmektedir. İkinci önemli fark kullanılan seçim metodunda gizlidir. Genetik algoritmada uygunluk fonksiyonuna göre belirlenen yeni bireyler birden fazla kere popülasyonda yeniden seçilebilmektedir. Evrimsel stratejilerde ise tekrarlama söz konusu değildir. Seçim deterministik olarak yapılmaktadır. Üçüncü fark ise evrimsel stratejilerde ilk

[36] Andries Engelbrecht, a.g.e., 2007, s.213.
[37] Thomas Bäck, a.g.e., , 1996, s.66-67.
[38] Enrique Alba ve Carlos Cotta, **Evolutionary Algorithms**, In Olariu, S., Zomaya, A.Y. (Ed.) Handbook of Bioinspired Algorithms and Applications, Taylor&Francis Group, LLC, 2006, s.6.

önce yeniden birleştirme (çaprazlama ve mutasyon) süreci sonra seçim süreci yapılmaktadır. Bazı evrimsel strateji modellerinde sadece mutasyon operatörü kullanılmaktadır. Genetik algoritmada ise bu durum tam tersidir[39].

Evrimsel stratejiler genellikle deterministik seçim modeli, normal dağılım (Gaussian) esaslı mutasyon ve ayrık veya orta seviyeli yeniden birleştirme süreçlerini kullanır. Çaprazlama operatörüne neredeyse hiç başvurmaz çünkü bu yöntem, genetik aşamalarda evrimi simule etmemektedir[40].

Önerilen ilk evrimsel strateji modeli, (1+1) stratejisidir. Bu strateji, deneysel optimizasyon problemlerinde kullanılmıştır. Popülasyon tabanlı bir yöntem olmayıp sadece tek birey vektöründen oluşmaktadır. Bu vektörde, optimize edilecek karar değişkenleri vektörü ve strateji parametreleri vektörü bulunmaktadır. Strateji parametreleri vektörü, her boyut için performansa göre dinamik olarak değişen mutasyon sayısını göstermektedir[41].

Diğer evrimsel strateji modelleri olan (λ+μ) ve (λ,μ) , Schwefel tarafından 1981 ve 1995'de literatüre kazandırılmıştır. Burada μ, popülasyon sayısını yani ebeveyn sayısını ifade ederken; λ, tüm μ ebeveynlerinden üretilecek olan çocuk sayısını göstermektedir. (λ+μ) modelinde, en iyi μ bireyleri, hem λ çocukları hem de μ ebeveynleri içeren bir aday havuzundan seçilir. (λ,μ) modelinde ise, μ ebeveyn adayları seçilebilir değildirler. Bir sonraki nesilin oluşmasında μ bireyleri, λ çocuklarının oluşturduğu bir aday havuzundan seçilir. Bu yüzden $\lambda \geq \mu$ olması gerekmektedir. Sıklıkla kullanılan $\lambda/\mu$ oranı 7 'dir. Hangi model kullanılırsa kullanılsın bırakılan μ bireyleri, gelecek nesilde torun olabilmek ve eşleşme yapabilmek için eşit olasılıklara sahip olurlar ve tamamen deterministik olarak seçilirler. (λ+μ) modelinde elitizm söz konusu iken (λ,μ) modelinde mevcut değildir.

[39] Leszek Rutkowski, **Computational Intelligence: Methods and Techniques**, SpringerVerlag, Heidelberg, 2005, s.289-290.
[40] R. Sarker ve dğr., **Evolutionary Optimization**, Kluwer Academic Publishers, Dordrecht, 2003, s.30.
[41] Andries Engelbrecht, a.g.e., 2007, s.213.

Genellikle evrimsel strateji algoritmasının (λ,μ) modeli, daha başarılı performanslar sergilemektedir[42,43].

Evrimsel stratejilerin uygulama adımları aşağıda sıralanmaktadır[44,45]:

*Adım 1:* μ bireyden oluşan bir başlangıç popülasyonu oluşturulur. Her bir birey ikili gerçek değerli vektörden oluşturulur. $(x_i, \eta_i) \quad (i = 0,1,2, \dots, \mu)$

*Adım 2:* Her bir bireyin uygunluk değeri hesaplanır.

*Adım 3:* Her bir ebeveyn $(x_i, \eta_i)$, ortalama olarak $\lambda/\mu$ yeni bir birey $(x_i{}^*, \eta_i{}^*)$ aşağıdaki formüller yardımıyla oluşturur.

$$\eta_k^*(j) = \eta_i(j) \exp\left(\tau' N(0,1) + \tau N_j(0,1)\right) \tag{1.5.}$$

$$x_k^*(j) = x_i(j) + \eta_i^*(j) N_j(0,1) \tag{1.6.}$$

$N(0,1)$ ortalaması 0 ve standart sapması 1 olan normal dağılım gösteren tek boyutlu tesadüfi bir sayıyı ifade etmektedir. $\tau$ ve $\tau'$ faktörleri ise yaygın olarak sırasıyla $(\sqrt{2\sqrt{n}})^{-1}$ ve $(\sqrt{2n})^{-1}$değerlerini almaktadır.

*Adım 4:* Her bir yeni bireyin $(x_i{}^*, \eta_i{}^*)$ uygunluğu hesaplanır.

*Adım 5:* Bireyler $(x_i, \eta_i)$ veya $(x_i{}^*, \eta_i{}^*)$ uygunluk değerine göre sıralanır ve en iyi μ sayıda birey bir sonraki nesilde ebeveyn olmak üzere seçilir.

*Adım 6:* Durdurma koşulları sağlanıncaya kadar önceki adımlar tekrarlanır.

[42] James Kennedy ve dğr., a.g.e., 2001, s.177.
[43] R. Sarker ve dğr., a.g.e., 2003, s.30.
[44] Kemal Çakar, a.g.e., 2009, s.33.
[45] R. Sarker ve dğr., a.g.e., 2003, s.32.

### 1.2.1.2. Evrimsel Programlama

Evrimsel Programlama (EP) , insan zekasının modellenmesine odaklanıldığı zamanlardaki yapay zeka yöntemlerine alternatif yaklaşım olarak 1960'lı yılların başında Fogel tarafından geliştirilmiştir. İlk EP modeli, sonlu durum makinesi yöntemini evrimleştirmek üzere geliştirilmiştir. Bu model, gözlemlenmiş davranış uzayından en optimal davranış setinin bulunmasını içerir. Bir başka deyişle bireylerin genlerini değil direk davranışlarını dikkate almaktadır. Diğer evrimsel hesaplama tekniklerinin aksine EP'de uygunluk fonksiyonu, bireylerin çevrelerine ilişkin olarak bireylerin "davranışsal hatalarını" ölçer[46,47].

Evrimsel programlama, genetik algoritma gibi stokastik optimizasyon stratejisidir. Fakat genetik algoritmanın aksine ebeveyn ile çocuklar arasındaki bağlantı davranışsaldır. Evrimsel stratejilerden bağımsız olarak geliştirilse de bu iki yöntem birbirine benzemektedir. Evrimsel strateji ve genetik algoritmalar gibi evrimsel programlama da yararlı bir yöntemdir[48].

Sayısal optimizasyon için kullanılan evrimsel programlama, algoritma açısından evrimsel strateji yöntemine benzemektedir. Her iki yöntemde olduğu gibi bireyler gerçek değerlerle temsil edilir, normal dağılımlı mutasyon söz konusudur ve kendi kendine adaptasyon mekanizmasını kullanmaktadır. Bu iki yöntem arasındaki fark ise yeniden birleştirme ve seçim operatörlerindedir. Evrimsel programlama, yeniden birleştirme veya çaprazlamayı kullanmamaktadır. Seçim mekanizması olarak turnuva seçimi gibi olasılıksal rekabet yöntemlerine başvurmaktadır[49].

Fogel'e göre evrimsel programlamanın beş farklı yaklaşımı aşağıda tanımlanmaktadır[50,51]:

[46] Andries Engelbrecht, a.g.e., 2007, s.187.
[47] Enrique Alba ve Carlos Cotta, a.g.e., 2006, s.6.
[48] S. Sumathi ve dğr., a.g.e., 2008, s.15
[49] R. Sarker ve dğr., a.g.e., 2003, s.32.
[50] Thomas Bäck, a.g.e., 1996, s.92.
[51] Kemal Çakar, a.g.e., 2009, s.37.

✓ *Standart Evrimsel Programlama*, herhangi bir kendi kendine adaptasyon mekanizmasının yokluğu ile karakterize edilir.

✓ *Sürekli Standart Evrimsel Programlama*, jenerasyon temelli çalışan mekanizmalardan farklı olarak yeni oluşturulan birey hemen değerlendirilir ve popülasyona dahil edilir.

✓ *Meta-evrimsel Programlama*, farklılıkları kendi kendine adaptasyonlarına izin vermek için genotipe dahil eder.

✓ *Sürekli Meta-evrimsel Programlama,* yukarıdaki gibi yeni oluşturulan bireyler değerlendirilir ve popülasyona dahil edilir.

✓ *Rmeta-evrimsel Programlama,* standart sapmalara ilaveten kendi kendine adatasyon için kovaryansları da genotipe dahil eder.

Evrimsel programlama ait sözde-kod aşağıda verilmektedir[52]:

$t = 0$;
Strateji parametrelerini tanımla;
$\eta_s$ bireylerden oluşan $C(0)$ başlangıç popülasyonunu oluştur;
**For** her birey için ,$x_i(t) \in C(t)$
Uygunluk hesapla ,$f(x_i(t))$;
**End**
**While** (false)
**For** her birey için ,$x_i(t) \in C(t)$
mutasyon operatörü ile $x_i'(t)$ çocuklarını üret
uygunluk hesapla $f(x_i'(t))$;
$C'(t)$ çocuk havuzuna $x_i'(t)$ çocuklarını ekle
**End**
Seçim operatörü uygulanarak $C(t) \cup C'(t)$ havuzundan $C(t+1)$ yeni popülasyon oluştur.
$t = t + 1$;
**End**

[52] Andries Engelbrecht, a.g.e. , 2007, s.188.

### 1.2.1.3. Genetik Algoritma

#### 1.2.1.3.1. Genel Bilgi

Evrimsel hesaplama tekniklerinin bir üyesi olan Genetik Algoritma (GA), biyoloji alanındaki evrim teorisinden esinlenmektedir. Doğada bulunan tüm bireyler kısıtlı olan yiyecek, barınak vb. ihtiyaçlarını karşılayabilmek için birbirleriyle rekabet içindedirler. Bu rekabet aynı türdeki bireylerin eş seçiminde de bulunmaktadır. Bu seçim sürecinde zayıf performans gösteren bireylerin yaşamlarını sürdürebilme şansı daha düşük olurken, nispeten daha uygun yapıdaki bireyler daha fazla yavru üretebilmektedir[53]. Eşleşme sırasında her bir ebeveynin iyi özelliklerinin birleşimi, uygunluğu atasından daha iyi olan yavrular ortaya çıkarabilir. GA, çevrenin uygunluğunu kontrol ederek problemin çözümü için kromozomları geliştirir[54]. Kısaca GA'nın temeli, en iyinin hayatta kalmasına ve adaptasyonuna dayanır.

Genetik algoritmanın ilk temelleri 1950'li yıllarda doğal genetik sistemi bilgisayarlarda simüle etmek isteyen biyologlar tarafından atılmıştır. Bu biyologların bazı çalışmaları genetik algoritmanın günümüzdeki prensiplerine oldukça yakın çalışmalardı. Bu alandaki ilk yayın 1950'lerde Fraser[55] tarafından yayımlanmıştır. 1960'lı yılların başında John Holland ve öğrencileri, genetik algoritma hakkında diğer yayınlardan daha etkili bir yayın yayımlamışlardır. Makine zekası alanıyla ilgilenen Holland öğrencilerinin de yardımıyla, genetik algoritmayı yapay sistemlere uygulamış ve geliştirmişlerdir. Holland'ın birçok katkısından biri arama sürecinde tek bir birey ile değil bireyler popülasyonunu kullanmak olmuştur. Fraser'da çalışmalarında popülasyonu kullanmasına rağmen yöntemini yapay sistemlere uyarlamamıştır. Holland ayrıca şema teorisi kavramını ortaya atmıştır. Şema teorisi, uygunluk değeri iyi olan kromozomlardan başarılı jenerasyonlar üretebilmek için

[53] S.N. Sivanandam ve S.N. Deepa, a.g.e. , 2008, s.15.
[54] M. Pelikan ve dğr., **Scalable Optimization via Probabilistic Modeling**, Springer-Verlag, Berlin, 2006, s.40.
[55] A.S. Fraser, "Simulation of genetic systems by automatic digital computers", **Australian Journal of Biological Science,** 10,484–499, 1957.

kromozom bloklarını inşa etmesini temel alır. "Genetik Algoritma" söylemi ilk olarak Bagley'in doktora çalışmasında geçmektedir[56]. Genetik algoritma alanındaki en önemli yıl olan 1975'de Holland[57], "Adaptation in Natural and Artificial Systems" kitabı yayımlanmıştır. Bu kitap bu konuya ilgi duyanlar tarafından oldukça ilgi görmüştür. Holland ve öğrencileri her bir çalışmalarında genetik algoritmaya katkı da bulunmuşlardır. Günümüzde genetik algoritma konusunda bilgi sahibi olmak için iki önemli eser mevcuttur. Bunların bir tanesi, Holland'ın öğrencisi Goldberg[58] tarafından yazılan "Genetic Algorithms in Search, Optimization, and Machine Learning" adlı kitap ve diğeri Lawrence Davis[59] tarafından 1991'de yayımlanan "Handbook of Genetic Algorithms" adlı kitaptır[60].

Doğal seleksiyon ve doğal genetik mekanizması kavramlarıyla yakından ilişkili olan GA'nın en önemli özelliği; yeni üretilecek çözümlerin, var olan çözümlerin uygunluk değerine (amaç fonksiyonu değerine) göre olasılıksal bir şekilde seçilmesidir. Buna göre uygunluk değeri, diğerlerine göre yüksek olan bireyin seçilme olasılığı daha fazla olacaktır.

GA'nın kesin bir tanımı olmamasına rağmen Mitchell[61], çaprazlama, mutasyon ve ters çevirme operatörlerinden esinlenen genetiklerle "doğal seleksiyonun" birlikte kullanılarak bir kromozom popülasyonundan yeni bir popülasyona geçişi sağlayan bir metot olduğunu; Nabiyev[62], en iyinin korunumu ve doğal seçilim ilkesinin benzetilmesiyle oluşturulan ve bilgisayar uygulamasıyla elde edilen bir arama yöntemi olduğunu söylemişlerdir.

---

[56] J.D. Bagley, **The behavior of adaptive systems which employ genetic and correlation algorithms**, Ph.D. dissertation, University of Michigan, Ann Arbor, 1967.
[57] J.H. Holland, **Adaptation in Natural and Artificial Systems**, Cambridge, MA: The MIT Press, 1975.
[58] D.E. Goldberg, **Genetic Algorithms in Search, Optimization and Machine Learning**, Reading, MA: Addison-Wesley, 1989.
[59] L. Davis, **Handbook of Genetic Algorithms,** New York: Van Nostrand Reinhold, 1991.
[60] James Kennedy ve dğr., a.g.e. , 2001, s.138-139.
[61] Melanie Mitchell, **An Introduction to Genetic Algorithms**, 5. Baskı, Massachusetts: A Bradford Book The MIT Press, 1999, s.2.
[62] Vasif Vagifoğlu Nabiyev, **Yapay Zeka: Problemler Yöntemler Algoritmalar**, 1. Baskı, Ankara: Seçkin Yayınevi, 2003, s.606.

Holland tarafından formüle edilen şema teoremi, GA'nın arama mantığını belirtir. Şema, diziler arasındaki belirli kısımların benzerliklerini gösteren alt dizileri tanımlayan sabit bir şablondur[63]. Şema, 0,1 ve * sembollerinden oluşur. Belirli uzunluktaki diziler arasındaki olabilecek tüm benzerlikleri göstermeye yarayan '*' sembolü, duruma göre 0 veya 1 değerini alabilir. Teorem, en fazla şema içeren dizinin bir sonraki toplumda da hayatını devam ettireceği mantığına dayalıdır. GA, en iyi şemanın bulunmasıyla en iyi kromozoma yaklaşır[64,65.]

GA'nın kromozom popülasyonu, uygunluğa göre seçim, yeni bir yavru elde etmek için çaprazlama ve rastgele mutasyon olarak sıralanan bileşenleri diğer evrimsel algoritmalarda bulunmamaktadır[66]. Genetik algoritmalar rastsal arama yöntemleri olmalarına rağmen klasik rastsal yöntemlerden farklıdırlar. Bu temel farklılıklar şöyle sıralanmaktadır[67]:

- GA, çözümün kendisi yerine çözümün kodlanmış biçimi ile çalışır,
- GA, tek çözümde değil, çözüm kümesinde arama yapar,
- GA, amaç fonksiyonunun türevi ve benzeri yardımcı bilgilerini değil amaç fonksiyonu bilgisini kullanır,
- GA, deterministik değil stokastik kuralları kullanır.

Stokastik algoritma olan GA'da olasılık temel role sahiptir. Başlangıç popülasyonunun oluşturulmasında, genetik operasyonlar için popülasyondan bireylerin seçilmesinde ve çaprazlama veya mutasyon gibi genetik operatörleri uygulamak için seçilen birey üzerindeki noktaların belirlenmesinde olasılık içeren metotlar kullanılmaktadır[68]. GA, her zaman bir çözüm popülasyonunu dikkate alır ve

[63] David. A. Coley, **An Introduction to Genetic Algorithms for Scientists and Engineers**, London: World Scientific Publishing, 1999, s.46.
[64] W. Banzhaf, **Genetic Programming: An Introduction,** Morgan Kaufman Publishers, Inc, San Francisco, California, 1998, s.96-97.
[65] James S. Spall, **Introduction to Stochastic Search and Optimization,** John Wiley & Sons Inc., New Jersey, 2003, s.263.
[66] Melanie Mitchell, a.g.e. , 1999, s.7.
[67] Masatoshi Sakawa, **Genetic Algorithms and Fuzzy Multiobjective Optimization,** Kluwer Academic Publisher, Massachusetts, 2002, s.15.
[68] Kenneth E. Kinnear, **Advances in Genetic Programming**, MIT Press, 1994, s.22-23.

farklı çözümleri bir araya getirerek popülasyonun çeşitlenmesini sağlar. Özel gereksinimlere ihtiyaç duymayan GA, her türlü probleme uygulanabilir[69].

GA'lar global optimal çözümü garanti edemez ama probleme kabul edilebilir bir çözüm önerebilir[70]. GA, popülasyon tabanlı algoritma olduğundan özellikle gerçek hayat problemlerine uygulandığında önemli bir dezavantaja sahiptir. Bir algoritmanın belirli süre içerisinde çözüm geliştirme işlemini tamamlamış olması gerekir. Bu işlem, yerel arama algoritmalarına kıyasla popülasyon tabanlı algoritmalarda daha uzun sürede olmaktadır. Süreyi kısaltmak için az popülasyonla çalışabilen GA'ların, iyi performansa sahip olması önemli hale gelmiştir. Diğer bir dezavantajı ise global minimuma yakınsama hızının çok iyi olmamasıdır[71].

Kanonik genetik algoritma diye de adlandırılan basit genetik algoritma, bireylerin temsil edilmesinde ikili gösterimi, çaprazlama sürecinde tek noktalı çaprazlamayı ve mutasyon sürecinde ise döndürme tipi mutasyonu kullanmaktadır. İkili gösterim aday çözümlerin 0 ve 1'lerle ifade edilmesidir. Tek noktalı çaprazlama sürecinde ise sabit n uzunluğundaki iki ebeveyn kromozom (x,y) rasgele sayı (r) yardımıyla belirli bir yerden ikiye ayrılır. Birinci çocuk, x kromozomunun (1-r) ile y kromozomunun (r-n) arasındaki bitlerle n uzunluğunda yeni bir birey olurken, 2. çocuk, x kromozomunun (r-n) ile y kromozomunun (1-r) arasındaki bitleri kullanır. Döndürme tipi mutasyon ise, 0 olan bir değerini 1'e, 1 olan bit değerini 0'a dönüştürülerek yapılır[72].

[69] Kemal Çakar, a.g.e. , 2009, s.41.
[70] S.N. Sivanandam ve S.N. Deepa, a.g.e. , 2008, s.20.
[71] Derviş Karaboğa, **Yapay Zeka Optimizasyon Algoritmaları: Tabu Araştırma, Işıl İşlem, Genetik, Karınca Koloni, Bağışıklık ve Diferansiyel Gelişim Algoritmaları**, 1. Baskı, İstanbul: Atlas Yayın Dağıtım, 2004, s.89.
[72] R. Sarker ve dğr., a.g.e. , 2003, s.34.

Standart bir GA'nin adımları aşağıdaki gibi verilebilir[73]:

*1. Adım:* Problem değişkenlerini, sabit uzunluktaki kromozomlar olarak tanımla; popülasyon büyüklüğünü ($N$), çaprazlama olasılığını ($P_c$) ve mutasyon olasılığını ($P_m$) seç,

*2. Adım:* Problemdeki bireylerin performansını (uygunluğunu) değerlendirecek bir amaç (uygunluk) fonksiyonu belirle. Bu uygunluk fonksiyonu, yeniden üretim boyunca eşleşecek kromozomların seçimi için bir temel oluşturur,

*3. Adım:* $N$ boyutlu ilk başlangıç popülasyonunu rastgele olarak oluştur,

$$x_1, x_2, \dots, x_N$$

*4. Adım:* Her bireyin uygunluğunu hesapla,

$$f(x_1), f(x_2), \dots, f(x_N)$$

*5. Adım:* Var olan popülasyondan eşleme için iki tane kromozom seç. Ebeveyn kromozomlar, uygunluklarıyla ilişkili olasılıklar ile seçilir. Yüksek uygunluğa sahip birey (kromozom), eşleşme için seçilmesinde daha az uygunluğa sahip bireyden daha yüksek olasılığa sahiptir.

*6. Adım:* Çaprazlama ve mutasyon operatörlerini kullanarak yavru kromozom çifti elde et,

*7. Adım:* Yavru bireyleri yeni popülasyona ekle,

*8. Adım:* Yeni popülasyonun büyüklüğü, ilk popülasyon büyüklüğüne ($N$) eşit olana kadar 5. Adımı tekrar et,

*9. Adım:* Yavru kromozomlarla ebeveyn kromozomları yer değiştir,

*10. Adım:* 4. Adıma git ve durdurma kriteri sağlanana kadar süreç devam ettir.

### 1.2.1.3.2. Genetik Algoritma Operatörleri

Genetik algoritmalar yeni nesil üretebilmek için yeniden üretme/seçim, çaprazlama ve mutasyon olmak üzere üç ana genetik operatör kullanır[74].

[73] Michael Negnevitsky, **Artificial Intelligence: A guide to Intelligent Systems**, Second Edition, Addison-Wesley, 2005, s.222.
[74] Masatoshi Sakawa, , a.g.e. , 2002, s.13.

### 1.2.1.3.2.1. Kodlama

Bir problemin çözümünde GA'nın uygulanabilmesi için yapılacak ilk iş değişkenlerin kodlanmasıdır. Kodlama, probleme özgü bilgilerin GA'nın anlayacağı dile çevrilme şeklidir[75]. Kodlama ayrı ayrı genleri simgeleme sürecidir ve bu süreçte bitler, sayılar, ağaçlar, listeler ve başka objeler kullanılarak genler temsil edilir[76]. Günümüzde henüz tüm problem çeşitleri için tek bir kodlama çeşidi olmadığından her problem kendine özgü bir kodlamaya ihtiyaç duyar.

GA'da kullanılan bazı kodlama çeşitleri aşağıda gösterilmiştir:

- ✓ İkili Kodlama
- ✓ Permütasyon Kodlama
- ✓ Değer Kodlama
- ✓ Ağaç Kodlama
- ✓ Gri Kodlama

### 1.2.1.3.2.2. Seçim

GA uygulamalarının temelini oluşturan doğal seleksiyon, yeni bireylerin oluşturulmasının ilk adımı olan seçim aşamasında önemli yere sahiptir. En yüksek uygunluğa sahip bireylerin daha fazla çoğalabilmesine imkan tanıyan seçim aşaması, uygun bireylerin popülasyondan nasıl seçileceğiyle ve seçilen bireylerden ne kadar yeni birey oluşturulabileceğiyle ilgilenir[77].

Başlangıç popülasyonundan seçilecek ebeveynler, uygunluk fonksiyonuna göre değerlendirilerek popülasyondan ayrılırlar[78]. Seçimin amacı, popülasyondaki

[75] Çağatan Taşkın ve Gül Gökay Emel, **Sayısal Yöntemlerde Genetik Algoritmalar**, Bursa: Alfa Aktüel, 2009, s.33.
[76] S.N. Sivanandam ve S.N. Deepa, a.g.e. , 2008, s.43.
[77] Masatoshi Sakawa, a.g.e. , 2002, s.19.
[78] S.N. Sivanandam ve S.N. Deepa, a.g.e. , 2008, s.46.

bireylerden daha iyi yavrular elde etmek umudu ile popülasyondaki nispeten daha uygun bireyleri ortaya çıkarmaktır[79]. Genetik algoritma, arama performansının en iyi olması için seçim baskısı ve popülasyon çeşitliliğinin ayarlanabildiği bir seçim yönteminin bulunması gerekir[80].

En iyi kromozomları seçmenin birçok yolu vardır. Rulet Çemberi Seçimi, Sıralama Seçimi, Sabit Durum seçimi, Turnuva Seçimi, Boltzman Seçimi ve Rastgele Seçim bunlardan en çok kullanılanlarıdır.

Seçim aşamasında kullanılan bir diğer yöntem ise seçkinlik (elitizm) aşamasıdır. Seçkinlik, mevcut topluluktaki uygunluk değeri en iyi olan bireyin doğrudan yeni topluluk havuzuna aktarılmasıdır. Bir jenerasyondaki en iyi birey çaprazlama ve mutasyon gibi operatörlerle bozularak kaybolabilir. Bu işlem, en iyi bireyin kaybolmaması için kullanılır. Seçkinlik uygulanan birey, hiçbir işleme tabi tutulmadan diğer jenerasyona aktarılır, böylelikle en iyi birey korunmuş olur[81]. Kısaca seçkinlik, popülasyonun maksimum uygunluk değerinin azalmasını engelleyen bir yöntemdir[82].

### 1.2.1.3.2.3. Çaprazlama

Çaprazlama operatörü, mevcut gen havuzunda bulunan yüksek uygunluktaki bireyleri başka bir deyişle bir önceki kuşaktaki nitelikli bireyleri elde etmek için kullanılır[83]. Genetik algoritmadaki metotlardan en dikkat çekici ve en önemlisidir. Çaprazlama operatörü; en iyi bireyler arasında değişim yaparak arama uzayında ulaşılmamış yerlerde arama yapmamızı sağlar[84]. Seçim evresinde popülasyondaki en

[79] Thomas Bäck ve dğr., a.g.e. , 2000a, s.56.

[80] Melanie Mitchell, a.g.e. , 1999, s.127.

[81] Davut Özdağlar ve dğr., "Kompleks Su Dağıtım Şebekelerinin Genetik Algoritma ile Optimizasyonu", **İMO Teknik Dergi**, 253, 3851-3867, 2006, s.3856.

[82] Thomas Bäck ve dğr., **Evolutionary Computation 2 Advanced Algorithms and Operators**, Bristol and Philadelphia: Institute of Physics Publishing, 2000b, s. xxv.

[83] Çağatan Taşkın ve Gül Gökay Emel, a.g.e. , 2009, s.49.

[84] Mitsuo Gen, ve Runwei Cheng,, **Genetic Algorithms and Engineering Optimization**, John Wiley & Sons, Inc. , 2000, s.8.

iyi bireyleri seçerek eşleme havuzundaki ortalama uygunluk değeri arttırılırken yeni bireyler üretilemez sadece seçilen en iyi bireyler klonlanmış olur. Çaprazlama evresinde ise çaprazlama eşleme havuzuna uygulanarak ebeveyninden farklı yeni bireyler elde edilir[85].

Çaprazlama süreci genel olarak üç adımdan oluşur[86]:

1. Yeniden üreme operatörü eşleşme için rassal olarak bir çift birey dizisi seçer,
2. Dizi boyunca rassal olarak bir çaprazlama alanı belirlenir,
3. Bu kısımdaki değerler karşı dizidekilerle yer değiştirerek işlem tamamlanır.

Popülasyonda ne oranda bir çaprazlamaya izin verildiği 0 ile 1 arasında değer alabilen çaprazlama oranı ($P_c$) ile gösterilir[87]. Literatürde çeşitli çaprazlama operatörleri önerilmiştir. Bunlardan en yaygın kullanılanları tek noktalı ve çok noktalı çaprazlamadır.

En çok kullanılan çaprazlama teknikleri aşağıda sıralanmaktadır:

- ✓ Tek Noktalı Çaprazlama
- ✓ İki Noktalı Çaprazlama
- ✓ Uniform Çaprazlama
- ✓ Dairesel Çaprazlama

Bu çaprazlama yöntemleri dışında; Kısmı Eşleşmiş Çaprazlama, Sıralı Çaprazlama, Üstünlük Korumalı Çaprazlama, Doğrusal Sıralı Çaprazlama ve Ağaç Çaprazlama yöntemleri de vardır[88,89].

[85] Kemal Çakar, a.g.e. , 2009, s.59.
[86] S.N. Sivanandam ve S.N. Deepa, a.g.e., 2008, s.50.
[87] Masatoshi Sakawa, , a.g.e. , 2002, s.22.
[88] S.N. Sivanandam ve S.N. Deepa, a.g.e. , 2008, s.54.
[89] Çağatan Taşkın ve Gül Gökay Emel, a.g.e. , 2009, s. 52-56.

### 1.2.1.3.2.4. Mutasyon

Kromozomlar belirli bir zaman sonra aynı bit dizilimini tekrarlayabilirler. Bu da arama uzayında sürekli aynı yeri taramak anlamına gelir. Bu yüzden yeni kromozomlar üretmek için mutasyon operatörüne ihtiyaç duyulur. Mutasyon, mevcut bireylerden yeni genetik malzemeler ortaya çıkarmak ve popülasyona çeşitlilik kazandırmak amacıyla kullanılır[90]. GA uygulamalarında ikinci dereceden rol oynayan mutasyon, küçük olasılıkla bir dizideki bir ya da birkaç tane değerin rastgele olarak değiştirilmesiyle arama uzayında yeni çözümler elde etmemizi sağlar[91].

Basit bir arama operatörü olarak tanımlanabilen mutasyon operatörü, tüm arama uzayını keşfetmek için yardımcı olabilir. En önemli özelliklerinden biri ise popülasyonda yok olmuş bir bireyi tekrar bulmamızı sağlayarak popülasyondaki çeşitliliği korumasıdır[92].

En çok kullanılan mutasyon teknikleri aşağıda sıralanmaktadır.

- ✓ Döndürme
- ✓ Değiş-Tokuş
- ✓ Yenileme/ Yer Değiştirme

Genetik operatörlerin uygulanmasından sonra popülasyonda bozulmalar oluşabilir. Uygun olmayan kromozomlar *tamir operatörü (düzenleyici algoritma)* adı verilen ve başlangıç popülasyonunun bilgilerini esas alarak geliştirilen bir algoritma yardımıyla tamir edilir. Kromozomlar, tamir edildikten sonra bir sonraki nesle aktarılarak GA çalışmasına devam eder[93].

---

[90] Kemal Çakar, a.g.e. , 2009, s.63.
[91] Berna Dengiz ve Fulya Altıparmak, "Genetik Algoritmalara Genel Bir Bakış", **Endüstri Mühendisliği Dergisi**, 9(3), 3-14, 1998, s.6.
[92] S.N. Sivanandam ve S.N. Deepa, a.g.e. , 2008, s.56.
[93] Çetin Elmas, **Yapay Zeka Uygulamaları**, Ankara: Seçkin Yayıncılık, 2007, s.398.

### 1.2.1.3.3. Genetik Algoritmalarda Parametre Seçimi

Parametreler, genetik algoritmanın performansı üzerinde önemli etkiye sahip olduklarından parametre seçimi GA'daki diğer önemli bir adımdır. Kontrol parametreleri olarak da adlandırılan bu parametrelere ilişkin birçok çalışma yapılmıştır[94]. GA'nın başarısı veya başarısızlığı bu parametre değerlerinin seçimine bağlıdır[95].

#### 1.2.1.3.3.1. Popülasyon Büyüklüğü

Genetik algoritma kullanıcısı tarafından verilen en önemli kararlardan birisidir. Bu değer çok küçük olduğunda, genetik algoritma yerel bir optimuma takılabilmektedir. Popülasyonun çok büyük olması ise çözüme ulaşma zamanını arttırmaktadır[96]. Önemli bir role sahip popülasyon büyüklüğü; hesaplama etkinliği, yeterli örnekleme sağlayabilme durumu, popülasyonun çeşitliliği, çaprazlama ve seçim baskısı gibi parametreler arasında dengeyi kurmada yardımcıdır[97]. Bu konu ile ilgili yapılan çalışmalar sonucunda kısıt içeren problemlerde popülasyon büyüklüğünün dizi uzunluğuna yakın bir değer seçilmesi gerektiğini belirtmişlerdir[98].

#### 1.2.1.3.3.2. Üreme Parametresi

GA'nın arama işleminin yönlendirilmesinden üreme parametresi birinci derecede sorumludur. Üreme işlemi sonucunda en iyi çözümün sahip olduğu kopya sayısını gösteren seçim basıncı, üreme sonrası uyum değeri düşük olan bireylerin elenmesini sağlayan parametrelerden biridir.

---

[94] Berna Dengiz ve Fulya Altıparmak, a.g.e., 1998, s. 8.
[95] Melanie Mitchell, a.g.e., 1999, s.131.
[96] Gül Gökay Emel ve Çağatan Taşkın, "Genetik Algoritmalar ve Uygulama Alanları", **Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi**, XXI(1), 129-152, 2002, s.135.
[97] Thomas Bäck ve dğr,, a.g.e. , 2000b, s.134.
[98] Çağatan Taşkın ve Gül Gökay Emel, a.g.e., 2009, s.72.

### 1.2.1.3.3.3. Çaprazlama Olasılığı

Yeni kromozomların eskilerin iyi özellikli parçalarını taşıyacağı umuduyla yapılan çaprazlama operatörünün temel parametresi çaprazlama olasılığıdır. $P_c$ ile gösterilen çaprazlama olasılığı, çaprazlamanın ne kadar sıklıkta olacağını[99] bir başka deyişle eşleşme havuzuna girecek kromozomların sayısını belirler. Bu oranın yüksek olması; iyi özellikteki bireylerin yeni popülasyonda bulunma olasılığını azaltırken, düşük olması; yeterli sayıda yeni bireyin oluşmasını engeller[100].

### 1.2.1.3.3.4. Mutasyon Olasılığı

Popülasyondaki genetik çeşitliliği koruyan mutasyonun en önemli parametresi kromozomda ne kadar bir değişim olacağını gösteren mutasyon olasılığıdır. $P_m$ ile gösterilen mutasyon olasılığı sıfır olursa yeni birey mutasyona uğramadan oluşurken, %100 olursa kromozom tamamen değişir[101]. Yapılan çalışmalar sonucunda popülasyon büyüklüğü 20'den büyükken $P_m > 0,05$ olarak seçildiği durumda veya popülasyon büyüklüğü 20'den küçükken $P_m < 0,002$ olduğu durumda performansın arttığı gözlenmiştir[102].

### 1.2.1.3.3.5. Kuşak Aralığı

Her nesilde oluşan yeni kromozom oranına kuşak aralığı adı verilmektedir. Genetik operatörler için kaç tane kromozomun seçildiğini gösterir. Yüksek bir değer birçok kromozomun yer değiştirdiği anlamına gelmektedir[103].

[99] S.N. Sivanandam ve S.N. Deepa, a.g.e.,, 2008 s.56.
[100] Randy L Haupt ve Sue A. Haupt, a.g.e., 2004, s.117.
[101] S.N. Sivanandam ve S.N. Deepa, a.g.e., 2008, s.57.
[102] Thomas Bäck, a.g.e., 1996, s.123.
[103] Gül Gökay Emel ve Çağatan Taşkın, a.g.e., 2002, s.136.

#### 1.2.1.3.3.6. Seçim Stratejisi

Eski kuşağı yenilemenin çeşitli yöntemleri mevcuttur. *Kuşaksal stratejide,* mevcut popülasyondaki kromozomlar tamamen yavrular ile yer değiştirir. Popülasyonun en iyi kromozomu da yenilendiğinden dolayı bir sonraki kuşağa aktarılamaz. En iyi kromozomu geri kazanmak için bu strateji elitizm (en uygun) stratejisiyle beraber kullanılmaktadır. *Elitizm stratejisinde,* popülasyondaki en iyi kromozomlar hiçbir zaman yenilenmemektedir, bundan dolayı üreme için en iyi çözüm her zaman elverişlidir. *Denge durumu stratejisinde* ise, her kuşakta yalnızca birkaç kromozom yenilenmektedir. Genellikle, yeni kromozomlar popülasyona katıldığında en kötü kromozomlar yenilenir[104,105].

#### 1.2.1.3.3.7. Fonksiyon Ölçeklemesi

Doğrusal ölçekleme, üstsel ölçekleme gibi yöntemler mevcuttur. Probleme göre en uygun ölçekleme yönteminin seçilmesi genetik algoritmanın etkin işlemesi açısından önem taşımaktadır[106]. Kompleks yapıya sahip uygunluk fonksiyonunun değerlendirilmesinin zorlaştığı problemlerde, uygunluk fonksiyonunu ölçeklendirerek yani basitleştirerek daha kolay çözüm bulabiliriz.

### 1.2.1.4. Diferansiyel Gelişim Algoritması

Diferansiyel Gelişim Algoritması (DGA), sürekli optimizasyon problemlerinin çözümünde kullanılmak üzere Storn ve Price tarafından ortaya atılmıştır[107]. Genetik Algoritmanın temel prensiplerini kullanan DGA'da bireyler gerçek değerleriyle kodlanmaktadır. Popülasyon tabanlı bir yöntem olan DGA, kodlanması basit ama etkili sonuçlar veren bir direk araştırma yöntemidir. Bilimde ve özellikle mühendislik

[104] Kemal Çakar, a.g.e., 2009, s.69.
[105] Gül Gökay Emel ve Çağatan Taşkın, a.g.e., 2002, s.136.
[106] Çağatan Taşkın ve Gül Gökay Emel, a.g.e., 2009, s.80.
[107] Rainer Storn ve Kenneth Price, "Differential Evolution: A Simple and Efficent Heuristic Strategy for Global Optimization over Continuous Spaces", **Journal of Global Optimization,** 11 , 1997, 341-359.

alanındaki problemler gerçek parametreli optimizasyon problemleri olarak adlandırılmaktadır. Gerçek değerle kodlama sayesinde DGA bu alanda kullanılan önemli yöntemlerden biridir. Özetle, Evrimsel hesaplama tekniklerinin bir üyesi olan DGA, stokastik direk arama ve küresel optimizasyon algoritmasıdır. Ayrık optimizasyon algoritması değil nümerik optimizasyon için geliştirilmiş gelişime dayalı bir algoritmadır. DGA'nın basit ve sade yapısı, kullanım ve kodlama kolaylığı, hızı ve dinçliği kullanımı konusunda önemli avantajlar sağlamaktadır[108,109,110].

Tüm evrimsel algoritmalarda varolan nesilden yeni bir nesil oluşturma işlemi çaprazlama ve/veya mutasyon operatörleri tarafından yapılmaktadır. Şayet bu iki operatörde kullanılıyorsa çaprazlama operatörü ilk sırada işlem görmektedir. Daha sonra üretilen yeni nesiller mutasyona uğramaktadır. Bu algoritmalarda mutasyon sayısı, aynı olasılık dağılım fonksiyonu tarafından belirlenmektedir. DGA'da ise ilk önce mutasyon operatörü ile deneme vektörü oluşturulur. Çaprazlama operatörü ile birlikte yapılan bu mutasyon işlemi sonucunda yeni yavru üretilmektedir. Mutasyon sayısı ise aynı olasılık dağılımından belirlenmez. Bu sayı, varolan popülasyon bireyleri arasındaki farklardan bulunan fark vektörü tarafından belirlenir[111]. DGA'nı diğerlerinden ayıran özellik olan farklı mutasyon ve çaprazlama işlemi, mutasyon işlemine tabi tutulacak bir amaç vektörüne, rasgele seçilen amaç vektörler çiftinin ağırlıklaştırılmış farkının ilave edilmesiyle yapılır.

DGA'nın adım adım işleyişi aşağıda anlatılmaktadır[112]:

Adım 1: Kontrol parametrelerinin $D, G_{max}, NP \geq 4, F \in (0,1+), CR \in [0,1]$, değerlerini ve parametre sınırlarını $\bar{x}^{(ed)}, \bar{x}^{(ey)}$ ata.

Adım 2: Başlangıç Popülasyonunu oluştur.

[108] Rainer Storn ve Kenneth Price, a.g.e., 1997, s.342.
[109] T. Weise, a.g.e., s. 229, 2009.
[110] Derviş Karaboğa, **Yapay Zeka Optimizasyon Algoritmaları**, Genişletilmiş 2. Baskı, Nobel Yayın Dağıtım, 2011, s.161.
[111] Andries Engelbrecht, a.g.e., 2007, s.237.
[112] Derviş Karaboğa, a.g.e., 2011, s.163.

$$\forall i \leq NP \ \wedge \forall j \leq D: x_{j,i,G=0} = x_j^{(ed)} + rand_j[0,1] * \left(x_j^{ey} - x_j^{(ed)}\right)$$
$$i = (1,2,\dots,NP);\ j = (1,2,\dots,D);\ G = 0;\ \ rand_j[0,1] \in [0,1] \quad (1.7.)$$

**REPEAT**

Adım 3: Mutasyon ve rekombinasyon(çaprazlama)

$$r_1, r_2, r_3 \in \{1,2,\dots,NP\};\ r_1 \neq r_2 \neq r_3 \neq i\ (rasgele\ seçilmiş)$$
$$j_{rand} \in \{1,2,\dots,D\}, (rasgele\ seçilmiş)$$
$$n_{j,i,G+1} = x_{j,r_3,G} + F * \left(x_{j,r_1,G} - x_{j,r_2,G}\right) \quad (1.8.)$$
$$\forall j \leq D, u_{j,i,G+1} = \begin{cases} n_{j,i,G+1}, & eğer\ (rand_j[0,1) < CR\ \vee j = j_{rand}) \\ x_{j,i,G}, & d.d. \end{cases} \quad (1.9.)$$

Adım 4: Seleksiyon

$$\bar{x}_{i,G+1} = \begin{cases} \bar{u}_{i,G+1}, & eğer\ f(\bar{u}_{i,G+1}) \leq f(\bar{x}_{i,G}) \\ \bar{x}_{i,G}, & d.d. \end{cases} \quad (1.10.)$$

**UNTIL** ( Durdurma Koşulu Sağlanana Kadar)

$NP$ : Populasyon Büyüklüğü (kromozom sayısı) NP ≥4 (1, 2, 3, …, i)

$D$ : Değişken Sayısı (gen sayısı) (1, 2, 3, …, j)

$CR$ : Çaprazlama Oranı [0.1,1.0]

$G$ : Jenerasyon $(1, 2, 3, \dots, G_{max})$

$F$ : Ölçekleme Faktörü

$x_{j,i,G}$ : G jenerasyonunda, i kromozomunun j parametresi (gen)

$n_{j,i,G+1}$ : Mutasyon ve çaprazlamaya tabi tutulmuş ara kromozom

$u_{j,i,G+1}$ : $x_{j,i,G}$'den bir sonraki jenerasyon için üretilen kromozom(child-trial)

$r_1, r_2, r_3$ :Yeni kromozomun üretilmesinde kullanılacak rasgele seçilmiş kromozomlar

$\bar{x}^{(ed)}, \bar{x}^{(ey)}$: Değişkenlere ait alt ve üst sınır değerleri

DGA'nın kullandığı en önemli parametreler: $NP$ (Number of Population-Popülasyon Büyüklüğü), $CR$ (Crossover Rate-Çaprazlama Sabiti), $F$ (Scaling Factor-Ölçekleme Faktörü) olarak sayılabilir. Bu parametre değerlerinin tüm gelişime dayalı algoritmalarda olduğu gibi iyice değerlendirilip seçilmesi gerekmektedir. Bu parametreler problemin çözümünü direk etkileyen faktörlerdir.

Çeşitli DGA modelleri aşağıda gösterilmiştir[113]:

*Ayrık Optimizasyon Problemleri İçin*

- Binary Diferansiyel Gelişim Algoritması
- Açısal (Angle Modulated) Diferansiyel Gelişim Algoritması

*Sürekli Optimizasyon Problemleri İçin*

- Basit Diferansiyel Gelişim Algoritması
- Hibrid Diferansiyel Gelişim Stratejileri
  - Gradyan Temelli Hibrid DGA
  - Gelişimsel Algoritma Temelli DGA
  - Parçacık Sürü Optimizasyonu Temelli DGA
- Popülasyon Temelli Diferansiyel Gelişim Algoritması
- Kendi kendine Organize Olabilen Diferansiyel Gelişim Algoritması

### 1.2.1.5. Yapay Bağışıklık Algoritması

İnsan vücudu, kendi yapısını korumak için dışarıdan gelebilecek saldırılara karşı birçok savunma mekanizmasına sahiptir: Bu mekanizmalardan biri de Bağışıklık Sistemidir. Kendi kendine programlanabilen bu mekanizma, yabancı antijenleri tespit edebildiği gibi yeniden saldırılarına karşı bu antijenlere ait bir hafıza oluşturmaktadır. Hafıza oluşumu, adaptif bağışıklık sisteminin hızlı reaksiyon göstermesine böylelikle yabancı antijenlerin vücuttan kolaylıkla atılmasını sağlamaktadır[114].

[113] Andries Engelbrecht, a.g.e., 2007, s.245-255.
[114] Andries Engelbrecht, a.g.e., 2007, s.415.

Bağışıklık sistemi birbiriyle etkileşim halinde olan iki alt bağışıklık sistemine sahiptir. Bunlardan biri yabancı patojen ve antijenleri doğuştan tanıyabilme yeteneğine sahip doğal bağışıklık sistemidir. Bu sistemde antijenler ilk görüşte tanınıp yok edilebilmektedir. Doğal bağışıklık cevabının en önemli bileşeni, antikor hareketlerine yön veren komplement olarak bilinen kan proteinleri sınıfındadır. Diğer önemli özelliği ise kazanılmış bağışıklık cevabının yapılmasında rol oynayan T hücrelerinin aktifleşmesinden sorumlu antijen sunucu hücrelerde ikincil uyarım sinyallerinin oluşmasını sağlamaktır. Diğer tip bağışıklık sistemi ise kazanılmış bağışıklık sistemidir. Bu sistem, B hücreleri ve T hücreleri diye ikiye ayrılan lenfositlerde klonal olarak dağıtılmış somatik üretilen antijen reseptörlerini kullanır. Daha önceden herhangi bir işgalciyle karşılaşılmamış olsa bile kazanılmış bağışıklık, o mikrobu tanıma ve cevap verme yeteneğini kazanabilir[115].

Bağışıklık sisteminin temel çalışanları lökositler ve lenfositlerdir. Bunların üremesinden, büyümesinden ve gelişiminden sorumlu organlara ise lenfosit organlar adı verilmektedir. Bunlar: tonsiller ve adenoidler, lenfotik damarlar, kemik iliği, lenf düğümleri, timüs, dalak, peyer's plakaları ve apandisttir. Bağışıklık hücreleri temelde üçe ayrılmaktadır. Lenfositler, hücre yiyiciler olarak bilinen fogositler, grarulositler ve benzerleri ve son olarak komplementlerdir. Lenfositler kendi aralarında üçe ayrılmaktadır. B hücreleri ve antikorlar; B hücreleri, exogeneous proteinlere karşı antikor sağlanmasında ve üretilmesinde görevlidir. B hücrelerinin her biri özel bir antikoru üretir. Antikorlar ise başka özel proteini tanıyan ve ona bağlanan, bağışıklık sisteminin en önemli tanıma elemanıdır. T Hücreleri ve Lenfokinler; Timus içinde üretilen T hücreleri fonksiyonu, diğer hücre hareketlerinin rsegülasyonunu sağlar ve enfekte olmuş host hücrelerine direk olarak saldırır. Üç tip T lenfositleri vardır: Yardımcı Th, Öldürücü T ve Baskılayıcı T hücreleri. Lenfokinler ise hücresel büyümeyi, aktivasyonu ve regülasyonu sağlarlar. Bunların dışında hedef hücreleri yok eder, makrofajları uyarırlar. Doğal Öldürücü Hücreler, özellikle tümörlere saldıran, güçlü kimyasallarla dolu granülleri içeren öldürücüsü lenfositlerin bir

[115] L.N. De Castro ve F. J. Von Zuben, **Artificial Immune Systems: Part 1- Basic Theory and Applications**, Technical Report, TR-DCA 01/99, 1999, s. 3-4.

türüdür. Öldürme dışında bağışıklık regülasyonuna katkıda bulunur ve lenfokinleri salgılar[116].

Antijenlere karşı lenfositlerin aralarında yapmış oldukları organizasyonel davranışlar ve fonksiyonlar, bağışıklık bilimi çalışmalarında farklı teorilerin oluşmasını sağlamıştır. Bu teoriler, Klasik Görüş, Klonal Seçim Mekanizması, Ağ Teorisi ve Tehlike Teorisidir. Klasik Görüş, diğer üç teorinin temellerini oluşturmaktadır.

Klasik görüşe göre, bağışıklık sistemi, vücuttaki öz/self ve yabancı/non-self hücre ayrımını yapar. Antijenlerin varlığı, özelleşmiş aktive edilmiş hücrelerin oluşmasına neden olur. Bu hücreler antijenlerin yok edilmesinde rol oynamaktadır. Doğal bağışıklık sistemindeki öz/yabancı ayrımını yapabilen ve öz-toleranslı olgunlaşmış T hücreleri, klasik görüşünün en önemli özelliklerindendir. Klasik görüşteki yapay bağışıklık sistemi modellerinden biri negatif seçim mekanizmasıdır[117].

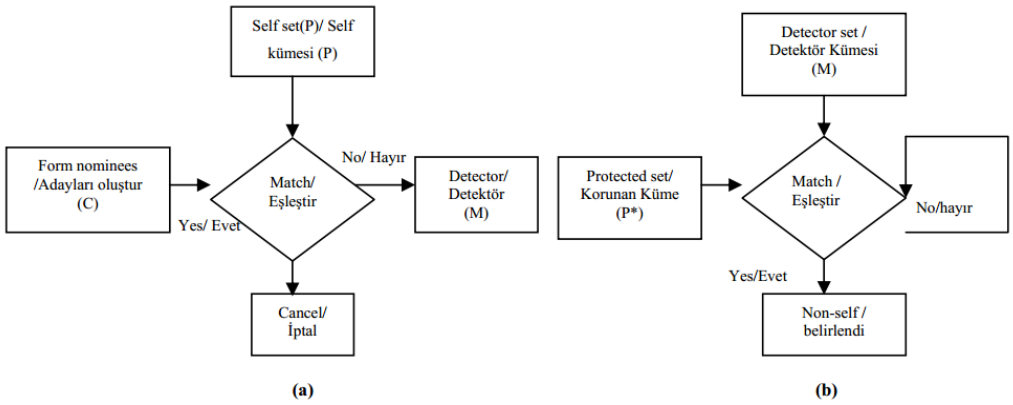
Bağışıklık sistemi, hücrelerinin öz ya da yabancı ayrımını yapması gerekir. Şayet bu ayrımı yapamazsa kendi öz hücrelerine saldırı emri verecektir. Bu durum, otobağışıklık hastalıklarına sebep olur. Sistemin kendi öz hücresine saldırmamasına öz-tolerans ya da sadece tolerans adı verilmektedir[118]. Negatif Seçim Algoritması da bağışıklık sisteminin yapmış olduğu bu özelliği esas alarak oluşturulmuştur. Algoritma Forrest tarafından literatüre kazandırılmıştır[119]. Yöntemde öncelikle bir aday dedektör kümesi üretilir ve bu aday dedektör kümesi öz küme ile eşleştirilir. Eğer eşleşme var ise bu adaylar asıl dedektör kümesine aktarılır. Daha sonrasında ise

[116] Derviş Karaboğa, a.g.e., 2011, s.140-141.
[117] Andries Engelbrecht, a.g.e., 2007, s.428.
[118] Derviş Karaboğa, a.g.e., 2011, s.145.
[119] İlhan Aydın ve dğr., "Genetik Algoritma Kullanan Yapay Bağışık Sistem Tabanlı Arıza Teşhis Modeli", **DEÜ Mühendislik Fakültesi Fen ve Mühendislik Dergisi**, Cilt:11 Sayı:31, 57-72, 2009, s. 60.

test verileri girilerek dedektör kümesi ile karşılaştırılır. Eğer herhangi bir dedektör tarafından bu test verisi tanınırsa istenilen işlem gerçekleştirilir[120].

Bu algoritmanın kullanım alanları genellikle bilgisayar ve ağ saldırıları, zaman serileri tahmini, şekil denetimi ve kesimlemesi, ve donatım hatası toleransı gibi anormallik belirleme problemleridir. Negatif Seçim Algoritmasının şematik gösterimi Şekil 1.3.'de gösterilmiştir.

**Şekil 1.3. Negatif seçim algoritması ile model tanıma a)detektör kümesini oluşturma, b)istenmeyen (nonself) maddelerin varlığını takip etmek**




**Kaynak:** Orhan Engin ve Alper Döyen, "Artificial Immune Systems and Applications in Industrial Problems", G.U. Journal of Science, 17(1), 71-84, 2004, s.74.

Bir aday detektör ile öz hücre arasındaki eşleşmenin hesaplanması için Euclidian, Hamilton ve Manhattan mesafe ölçümleri kullanılabilir[121].

Negatif seçim mekanizmasının tamamlayıcısı olan Klonal Seçme Mekanizması, bir "nonself" hücre ile karşılaşan bir "B" hücresinin nasıl bir

[120] Mehmet Bayğın ve Mehmet Karaköse, "Adaptif Yapay Bağışık Sistem Tabanlı Grup Asansör Kontrol Algoritması", **Elektrik- Elektronik ve Bilgisayar Sempozyumu**, 205-210, 2011, s.206.
[121] İlhan Aydın ve dğr., a.g.e., 2009, s.61.

bağışıklık tepkisi verileceğini açıklamada kullanılmaktadır[122]. Kısaca mekanizma şöyle çalışmaktadır: Bir antijen vücutta tespit edildiğinde, kemik iliğinden antikor üretilir. Böylece bağışıklık tepkisi veren B lenfositleri salgılanır. Bu antikorlar (hücre algılayıcıları) antijenlerle birleşerek yardımcı-T hücrelerinden gelen sinyallerin de etkisi ile B-hücresini uyarır. B-hücresi uyarma ile birlikte çoğalır (bölünür) ve plazma hücreleri denilen terminal (bölünmeyen) antikor salgılayan hücrelere dönüşür (farklılaşma). Hücre bölünmesi süreci (mitoz) bir klon (kopya) oluşturur. Bu klon bir tek yavru hücre veya bir küme yavrudur. Beyaz plazma hücreleri en aktif antikor salgılayıcılarıdır. Ayrıca, hızlı şekilde bölünen büyük B lenfositleri de daha düşük bir oranda antikor salgılarlar[123].

Castro ve Zuben tarafından ortaya atılan klonal seçim algoritmanın akış şeması ve adımları Şekil 1.4.'de gösterilmektedir[124].

Adım 1 : Vücut antikor repertuarını oluşturan antikorlar, başlangıç çözüm kümesini oluştururlar ($A_b$).

Adım 2 : Antikorların benzerlik dereceleri hesaplanır ($f_1$).

Adım 3 : n adet en yüksek benzerlikli antikor seçilir ($A_{bn}$).

Adım 4 : Seçilen n adet antikorun benzerlik dereceleri ile doğru orantılı olarak, yüksek benzerlik dereceli antikorun daha fazla olacak şekilde klonlanması ($C$).

Adım 5 : Antikorların benzerlik derecesi yüksek olan, daha az olacak şekilde mutasyona uğratılır ($C_n$).

Adım 6 : Mutasyona uğramış klonların benzerlik dereceleri belirlenir ($f_2$).

Adım 7 : n yüksek benzerlik dereceli antikorların yeniden seçilmesi ($A_{bn}$).

Adım 8 : d adet en düşük benzerlik derecesindeki antikorların yeni üretilen antikorlarla değiştirilmesi ($A_{bn}$)

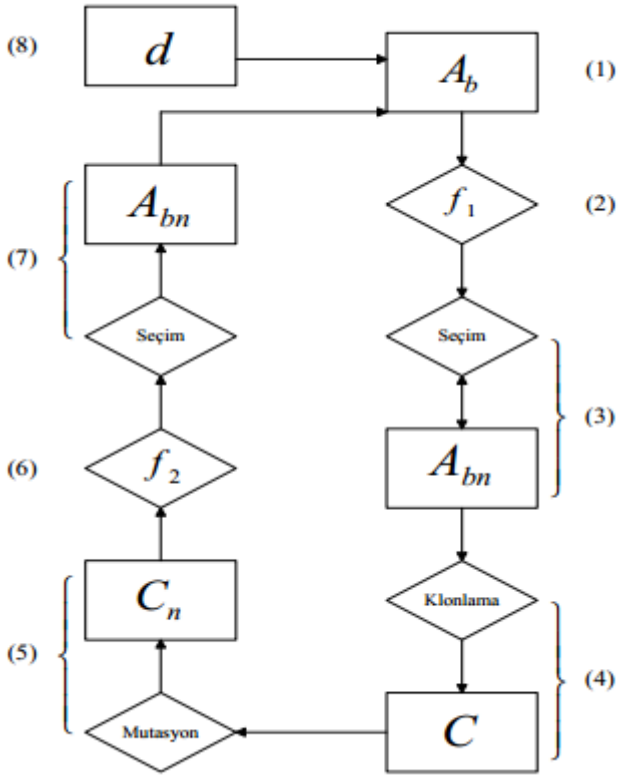
---

[122] L.N. De Castro ve J. Timmis, **"A Novel approach to pattern recognition"**. In L Alonso,J Corchado, and C Fyfe, editors, Artificial Neural Networks in pattern Recognition, University of Paisley: 67-84, 2002, s.70.

[123] L.N. De Castro ve F.J. Von Zuben, "The Clonal Selection Algorithm with Engineering Applications", **GECCO 2000**, Las Vegas, Nevada, USA, July 8, 36-42, 2000, s.36.

[124] L.N. De Castro ve F.J. Von Zuben, "Learning and optimization Using the Clonal Selection Principle", **In the Special Issue on Artificial Immune Systems of the Journal IEEE Transactions on Evolutionary Computation**, June 6(3), 239-251, 2002, s.246-247.

**Şekil 1.4. Klonal seçim algoritması**




**Kaynak:** L.N. De Castro ve F.J. Von Zuben, a.g.e., 2002, s.246-247.

Bağışıklık sisteminin en önemli özelliği öğrenebilme yeteneğidir. Bağışıklık sistemindeki öğrenme iki durumdan oluşur. Birincisi antijenleri tanıyarak kendini kanıtlayan lenfositlerin göreceli popülasyon büyüklüğünü (kopyalama ile) ikincisi ise antijenlere olan benzerliklerini artırmalarıdır (farklılaşma). Bağışıklık sisteminde destekli öğrenme stratejisi mevcuttur. Sistem bu strateji şu şekilde kullanmaktadır: Organizmalar tüm hayatları boyunca bir antijen ile defalarca karşılaşmaktadır. Bir antijen saldırısına ilk tepki B hücreleri tarafından verilir. Bu hücreler, her biri farklı benzerlik derecesinde antikor üreten, düşük benzerlik değerine sahip hücrelerdir. Bağışıklık sisteminin 2. saldırıya verdiği tepkinin etkinliği ilk enfeksiyon sonucunda oluşan hafıza hücrelerinin varlığı ile ilişkilidir. Hafıza hücreleri, yüksek benzerlikte antikor üretebilirler. Bu strateji ile her enfeksiyon aşaması sonrası bağışıklık tepkisinin sürati ve doğruluğu artar. Bu mutasyon ve seçim algoritmasını defalarca

tekrar ederek, bağışıklık sistemi yüksek benzerlikteki antikorları üretmeyi "öğrenir"[125].

İlk defa Jerne tarafından ortaya atılan Ağ teorisi, hücre ağı içindeki B hücrelerinin içsel iletişimini içerir. Eğer B hücresi, herhangi bir antijene karşı cevap verirse, ağa bağlı tüm B hücrelerini de uyarmış olur. Bu yüzden lenfositler sadece antijenler varlığıyla değil komşu lenfositler tarafından da uyarılabilir.

Matzinger tarafından ortaya atılan tehlike teorisi ise, öz-yabancı ayrımı yapmaz. Bu yöntemde amaç, yabancı hücrelerden hangisinin vücut için tehlikeli olup olmadığının araştırılmasıdır[126].

### 1.2.2. Sürü Zekası Temelli Algoritmalar

Sürü zekası, optimizasyon problemlerine çözüm bulmak için sosyal hayvanların ve sürülerin besin arama süreçlerindeki davranışlarından esinlenerek ortaya çıkan sayısal ve davranışsal bir kavramdır[127]. Doğada topluluk halinde yaşayan, aralarında belirli bir iş paylaşımının olduğu ve önlerine çıkan sorunlara yardımlaşarak birlikte çözüm bulan hayvanlara sosyal hayvan denilmektedir. Sosyal hayvanların doğadaki yaşam mücadeleleri incelendiğinde bir çok optimizasyon probleminin çözümüne katkı sağlamışlardır. Karıncalar, arılar, termitler gibi böceklerin yanında kuş ve balıklar sosyal hayvanlar olarak sınıflandırılabilir[128].

Bir yapay zeka disiplini olan sürü zekası, sosyal hayvanların kolektif davranışlarından esinlenilerek türetilmiş zeki çok ajanlı (multi-agent) sistem dizaynıdır. Kolektif davranış ise kendi kendine organize olan, merkez kontrolü

[125] L.N. De Castro ve F.J. Von Zuben, a.g.e., 2000, s.37.
[126] Andries Engelbrecht, a.g.e., 2007, s.422.
[127] M. Belal, ve dğr., **"Swarm Intelligence"**, In Olariu, S., Zomaya, A.Y. (Ed.) Handbook of Bioinspired Algorithms and Applications, Taylor&Francis Group, LLC, 2006, s.55.
[128] Derviş Karaboğa, a.g.e., 2011, s.109.

gerektirmeyen davranışlardır. Karıncaların kendilerinden çok büyük bir yaprağı birlikte yuvalarına taşımaları bir kolektif davranış örneğidir[129].

Kolektif davranışın temel prensipleri aşağıda tanımlanmaktadır[130]:

*Homojenlik:* Sürüdeki her birey aynı davranış biçimine sahiptir. Geçici liderleri varmış gibi görünmesine rağmen lidersiz hareket ederler.

*Mekan:* Sürüdeki her bir birey, sadece bulunduğu bölgenin yakınındaki bireyleri etkileyebilir.

*Çarpışmayı Önleme:* Sürü çiftleri karşılaşmamalıdır. Bağımsız olmalıdır.

*Hız Eşleme:* Her bir sürü civarındaki sürü ile aynı hıza sahip olmalıdır.

*Sürü Merkeziyeti:* Sürü çiftleri birbirlerinin yakınında kalmalıdırlar.

Sürü zekası terimi, ilk olarak Beni ve arkadaşları[131] tarafından kullanılmıştır. Yaptıkları araştırma en yakın komşuluk etkileşimi ile kendi kendine organize olabilen hücresel robotik sistemleri ele almaktadır. Probleme çözüm getiren algoritmaların sosyal hayvan gruplarının kollektif davranışlarından ilham alınarak geliştirilmesinden dolayı çalışmalarında sürü zekası terimini kullanmışlardır[132].

Bir sürünün zeki olarak tanımlanabilmesi için beş temel özelliğinin bulunması gerekmektedir[133]:

*Yakınlık Prensibi:* Popülasyon, temel uzay ve zaman hesaplamalarını uygulayabilmelidir.

*Kalite Prensibi:* Popülasyon, çevresindeki kalite faktörlerine cevap verebilmelidir.

---

[129] Christian Blum ve Daniel Merkle, **Swarm Intelligence Introduction and Application**, Springer Verlang, Berlin, 2008, s.43.

[130] Abraham Ajith, ve dğr., **"Swarm Intelligence in Data Mining"**, Grosan,Crina, Abraham, Ajith and Chis, Monica, Swarm Intelligence in Data Mining, Springer, 2006, s.1-2.

[131] G. Beni, "The concept of Cellular Robotic Systems", **In Proceedings of the IEEE International Symposiumon Inteligent Systems**, IEEE Press, Piscataway,NJ, 57-62, 1988.

[132] Eric Bonabeau ve dğr., **Swarm Intelligence: From Natural to Artifical Systems**, Oxford University Press, 1999, s.7.

[133] James Kennedy ve dğr., a.g.e., 2001, s.xx.

*Dağılım Cevabı Prensibi:* Popülasyon, kaynaklarını dar boğazlarda kullanmamalıdır.

*Kararlılık Prensibi:* Popülasyon, çevresel değişimlerde davranış biçimini sürekli değiştirmemelidir.

*Uyarlanabilirlik Prensibi:* Popülasyon hesaplama maliyetinin yüksek olduğu durumlarda davranış biçimini değiştirebilmelidir.

Sosyal hayvanlar, birçok duyusal girdiyi değerlendirebilen, çevresel uyarılara göre davranışlarını şekillendiren ve birçok bilgi yığını içerisinde karar verebilen karmaşık yapılı varlıklardır. Bireysel davranışlarının karmaşıklığı, sürü halinde yaptıkları davranışlarının karmaşıklığını açıklamada yetersiz kalmaktadır. Sosyal hayvanların işbirliği içinde hareket etmeleri incelendiğinde bireylerin anatomik farklılıklarının iş bölümü kavramını ortaya çıkarmıştır. Fakat sosyal hayvanların kolektif davranışlarının birçoğu kendi kendine organize olmaktan kaynaklanmaktadır. Kendi kendine organize olma teorisine göre sosyal hayvanların karmaşık kolektif davranışlar göstermesi, basit davranışlar sergileyen bireylerin aralarındaki etkileşimden kaynaklanmaktadır. Birçok araştırma gösteriyor ki kendi kendine organize olma kolektif davranış sergilemenin en önemli parçasıdır[134].

Kendi kendine organize olabilmek dört özellik ile tanımlanmaktadır[135]:

*Pozitif Geri Besleme:* Toplanma ve güçlendirme kavramlarını içerir. Karıncaların birbirini takip ederek ya da arıların dans ederek diğer bireyleri yiyecek kaynağına toplaması pozitif geri beslemeye örnek olarak verilebilir.

*Negatif Geri Besleme:* Pozitif geri beslemeyi dengeler. Sürünün doyum noktasına gelme, kaynakların tükenmesi ve diğer besin kaynakları arasındaki rekabet gibi kolektif durumları istikrarlı hale getirmek için ihtiyaç duyulan bir mekanizmadır.

[134] Eric Bonabeau ve dğr., a.g.e., 1999, s.7.
[135] Eric Bonabeau ve dğr., a.g.e., 1999, s.9.

*Dalgalanma:* Kendi kendine organize olma, rasgele yürüyüş veya hatalar gibi dalgalanmalardan oluşur. Bu tür dalgalanmalar yeni çözümlerin keşfini sağlar. Yeni çözüm kaynakları bulmak yaratıcılık ve yenilik getirdiğinden önemlidir.

*Çoklu etkileşimler:* Kendi kendine organize olmanın tüm örnekleri, çoklu etkileşime dayanmaktadır. Etkileşim sayesinde bireyler birbirleriyle haberleşirler böylece sürü içinde bilgi alış verişi sağlanmış olur.

Sürü zekasında iki farklı tür iletişim vardır: Direk İletişim ve Dolaylı İletişim.

Direk iletişim, bireylerin aralarında yapmış oldukları dolaysız iletişimdir. Bu tür iletişime örnek, bal arılarının yapmış oldukları sallanma dansı verilebilir.

Dolaylı iletişim ise, bireylerin aralarındaki iletişimi çevre yardımıyla yapması olarak tanımlanabilir. Bu iletişim, Stigmergy iletişim olarak da adlandırılmaktadır. Stigmergy iletişimde mutlaka bir uyarıcı bulunmaktadır. Bireyler bu sayede kendi davranışlarıyla diğer bireylerin davranışları arasındaki farkı ayırt edebilmektedir. Çevredeki değişimlere bu sayede kolaylıkla adapte olabilmektedirler. Stigmergy iletişimde iki farklı model söz konusudur. İlk modelde iletişim, çevredeki fiziksel değişimlerle yapılmaktadır. Çevredeki ısının, nemin, oksijen miktarının değişimi gibi. İkinci modelde ise direk bir değişim söz konusu değildir. İletişim, görevle ilgili ardışık davranışların etkisi ile olmaktadır. Bu tür iletişime en iyi örnek karıncaların besin kaynağına giderken feromen maddesini salgılamalarıdır[136].

Sürü zekası kavramı, etkili optimizasyon ve kümeleme algoritmalarının gelişmesine sebep olmuştur. Kuşların zarif ama tahmin edilemeyen göç hareketleri, parçacık sürü optimizasyonu algoritmasının tasarlanmasını sağlarken, karıncaların ve arıların besin arama davranışları karınca kolonisi algoritmasını ve yapay arı kolonisi algoritmasını ortaya çıkarmasına sebep olmuştur[137]. Bu yöntemlerde aynı işi yapan birçok ajan bulunmasının yanında yerel iletişim ağları ve yüksek kontrol

[136] M. Belal ve dğr., a.g.e., 2006, s.57-58.
[137] Andries Engelbrecht, a.g.e., 2007, s.9.

mekanizmaları mevcuttur. Farklı tip problemlerin çözümü için kolayca tasarlanabilen güçlü ve pratik tekniklerdir. Bir çok optimizasyon modeline uygulanabildiği gibi haberleşme ağları ve robotik işlemler gibi mühendislik uygulamalarında da etkilidirler[138].

### 1.2.2.1. Karınca Kolonisi Optimizasyonu Algoritması

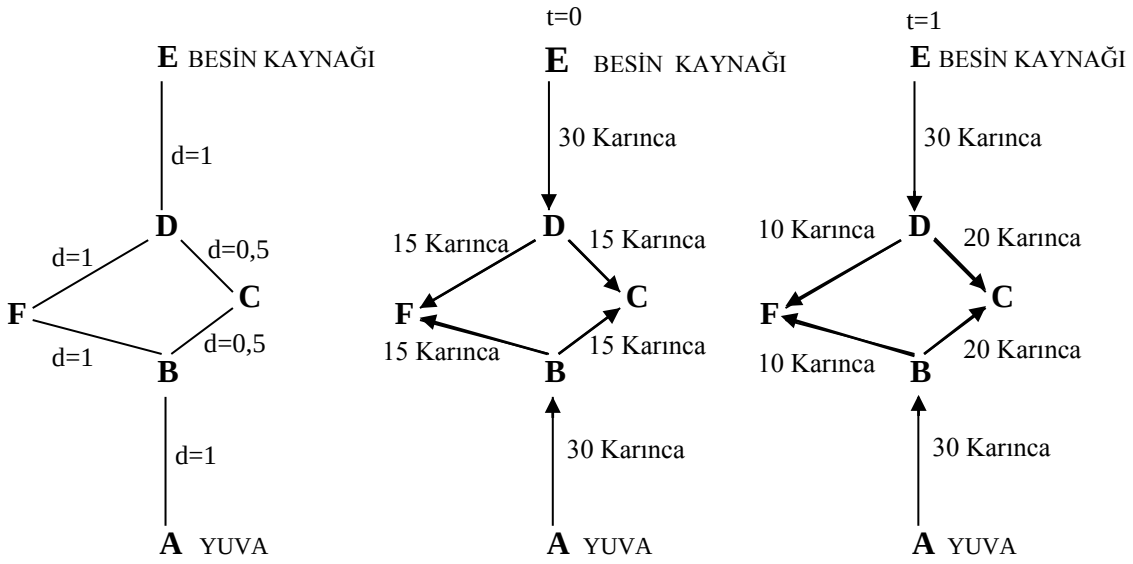
Doğadaki bazı sosyal sistemler, sınırlı yeteneklere sahip basit bireyler tarafından oluşturulmalarına rağmen kolektif yani ortaklaşa zekâ davranışı gösterirler. Bu bireylerin kendi aralarındaki organizasyonları ve birbirlerini dolaylı yollardan etkilemeleri, problemlere zeki çözümler sunmalarını sağlar[139]. Sosyal bir canlı olan karıncalar[140], tek başlarına etkili bir avlanma yapamazken ve diğer karıncalarla iletişim kuramazken, grup halindeyken karışık problemleri çözme ve başarılı bir şekilde yiyecek bulma ve toplama yeteneğine sahiptirler[141]. Bir başka ifadeyle genellikle kör olan karıncalar grup halindeyken verilen görevleri başarılı bir şekilde yaparlarken, bireysel olarak başarısız olurlar[142].

Koloniler halinde yaşayan karıncalar, bireysel olarak hayatta kalmaktan ziyade grup olarak hayatta kalmayı amaçlarlar ve hareketlerini bu amaç doğrultusunda yönetirler. Örneğin; bir karınca yemek ararken, rastgele hareketlerle besinini çevreleyen bölgeyi araştırır. Hareket ederken feromen adı verilen bir kimyasal salgı salgılar. Bunu hisseden diğer karıncalar, feromen salgı miktarının çok olduğu yolu tercih ederler. Karınca besine ulaştığında, besinin kalitesini ve miktarını belirler. Yuvaya besini taşımaya başlayan karıncanın yola bırakacağı feromen miktarı, besinin kalitesine ve miktarına bağlıdır. Feromen sayesinde karınca, diğer karıncalara besin kaynağının yerini bildirir ve yuva ile besin arasındaki en kısa yolun

[138] M. Belal ve dğr., a.g.e., 2006, s.55.
[139] Vasif Vagifoğlu Nabiyev, , a.g.e., 2003, s.628.
[140] Blum, Christian, "Ant Colony Optimization: Introduction and Recent Trends", **Physics of Life Reviews**, 2, 353-373, 2005, s.355.
[141] John E. Bell ve Patrick R. McMullen, "Ant Colony Optimization Techniques for The Vehicle Routing Problem", **Advanced Engineering Informatics**, 18, 41-48, 2004, s.42.
[142] B. Bullnheimer ve dğr., "Applying The Ant System to The Vehicle Routing Problem", **2nd International Conference on Metaheuristics – MIC97**, 1997, s.2, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.7946 (08.08.2014).

bulunmasını sağlar. Böylece karıncalar feromen salgısı sayesinde birbirlerini dolaylı olarak etkilerler. Karıncaların bu davranışları en kısa yolu nasıl bulduklarını açıklamaktadır[143]. Bu hareketin temeli, feromen adı verilen kimyasal salgı vasıtasıyla karıncaların arasındaki dolaylı iletişimdir[144].

**Şekil 1.5. Karıncaların en kısa yolu bulmalarının gösterimi**




**Kaynak:** B. Bullnheimer ve dğr, a.g.e., 1997, s.3.

Karıncalar, yuvalarından bir gıda kaynağına giden en kısa yolu, herhangi görsel ipucu kullanmadan bulma yeteneğine sahiptirler ve çevrelerindeki değişikliklere çok kısa sürede uyum sağlayabilirler[145]. Bu bilgiler ışığında Şekil 1.5.'de; karıncaların A ile gösterilen yuvalarından E ile gösterilen besin kaynağına giderken izledikleri yollar uzunlukları ile birlikte gösterilmektedir. t=0 anında 30 karıncanın yuvadan ayrılarak besin kaynağına ulaşabilmek için farklı uzunlukta alternatif iki yoldan birini tercih etmek zorundadır. Besin kaynağına ilk gidişte feromen maddesi olmadığından karıncaların yarısı uzun yolu, diğer yarısı da kısa yolu tercih ederler.

[143] Xiaoxia Zhang ve Lixin Tang, "A New Hybrid Ant Colony Optimization Algorithm for The Vehicle Routing Problem", **Pattern Recognition Letters**, 30, 848-855, 2009, s.849.
[144] Christian Blum, a.g.e., 2005, s.354.
[145] Gökhan Dalkılıç ve Fatih Türkmen, Karınca Kolonisi Optimizasyonu, **1. Ulusal Yüksek Performanslı Bilişim Sempozyumu**, Kocaeli, 2003, s.1.

Belirli bir zaman periyodu geçtikten sonra feromen salgısının etkisiyle 5 karınca daha kısa yolu tercih etmiştir. Her zaman periyodu geçtikçe kısa yoldaki feromen miktarı artacak ve karıncalar tamamen bu yolu tercih etmeye başlayacaklardır[146].

Yapay karıncaların özellikleri ve doğal karıncalar arasındaki farkları aşağıda açıklanmaktadır[147,148]:

✓ Yapay karıncalar, ayrık bir dünyada yaşarlar ve hareketleri durumdan duruma farklılık göstermektedir,

✓ Yapay karıncalar, daha önce gittiği yolların geçmişini tutan özel bir iç hafızaya sahiptirler,

✓ Yapay karıncalar, bulunan çözümün kalitesine göre artan miktarda feromen salgılarlar,

✓ Yapay karıncaların feromen salgılaması, çözüm üretildikten sonra yapılmaktadır. Bu davranış, gerçek karıncaların davranışı ile farklılık göstermektedir,

✓ Tüm sistemin etkinliğini artırmak için KKA'ya ek olarak görünürlük derecesi, yerel optimizasyon, geri besleme gibi özellikler ilave edilebilmektedir. Bu özellik gerçek karıncalarda bulunmamaktadır,

✓ Her yapay karınca, arama uzayında adım adım hareket ederek verilen probleme uygun çözümler bulmaya çalışır,

✓ Rehberlik faktörü, bir yerden bir yere hareket etmeden önce uygulanan geçiş kurallarını kullanan karınca hareketlerini içerir. Bu geçiş kuralları, probleme özel kısıtların ve karınca iç hafızasını içerebilir.

Doğal karınca kolonisinden yapay karınca kolonisine geçiş, yapay feromen yolları boyunca iletişim kurarak işbirliği içinde çalışan basit sayısal işlemlerin

[146] B. Bullnheimer ve dğr., a.g.e., 1997, s.2-3.
[147] Marco Dorigo ve dğr., Ant Algorithms for Discrete Optimization, **Artificial Life**, 5, 137-172, 1999, s.142.
[148] R.J. Mullen ve dğr., A Review of Ant Algorithms, **Expert Systems with Applications**, 36, 9608-9617, 2009, s. 9609-9610.

kullanımını içerir. Her karıncanın i noktasından j noktasına hareket etmesine aşağıdaki iki neden rehberlik etmektedir[149,150] :

- ✓ *İzlenebilirlik (Sezgisel Bilgi):* Öncelik olarak bilinen bilgidir ve i noktasında j noktasına hareket için sezgisel tercih ölçüsüdür,
- ✓ *Yapay feromen yolları (İz):* i noktasından j noktasına gidilmeden önce karıncalar tarafından bırakılan feromen miktarıdır.

Günümüze kadar birçok araştırmacı tarafından çeşitli karınca kolonisi algoritmaları önerilmiştir. Bu algoritmaların temelinde ise 1991'de Dorigo ve arkadaşları tarafından önerilen Karınca Algoritması vardır. Bu algoritma esas olarak üç ayrı algoritmayı içermektedir. Bunlar karınca yoğunluk, karınca miktar ve karınca çevrim algoritmalarıdır. Karınca yoğunluk ve karınca miktar algoritmalarının arasındaki tek fark karıncaların yollara bıraktıkları feremon maddesi miktarının hesaplanmasındadır. Karınca çevrim algoritmasının diğerlerinden farkı ise feremon maddesi güncellemesinin her adım da değil tur tamamlandıktan sonra yapılmasıdır. Bu sayede karınca çevrim algoritması değerlendirme yapmak için küresel bilgiyi kullanmaktadır[151].

Karınca Kolonisi Algoritmalarının temelini oluşturan Karınca Sistemine ait gezgin satıcı probleminin çözümde kullanılan algoritma aşağıda gösterilmektedir[152]:

$\tau_{ij}$ ve $\eta_{ij}$ $\forall(ij),\ başlangıç\ değerleri\ belirle$

**do**

**for** $k = 1\ to\ m$ {

$S_p^k = \emptyset$ $\quad$ $//\ oluşum\ adımı\ başladı$

**for** $i = 1\ to\ N$ {

[149] Marco Dorigo ve dğr., Positive feedback as a search strategy, **Technical Report, 91-016**, June 1991, s.3-4.
[150] R.J. Mullen ve dğr., a.g.e., 2009, s.9609.
[151] Marco Dorigo ve dğr., a.g.e., 1991, s.2-6.
[152] Tunçhan Cura, **Uygulamalı Modern Sezgisel Teknikler ve Uygulamalar**, Papatya Yayıncılık, İstanbul, 2008, s.117.

**if** $(i = 1)$ {

$t_i^k = rastlantı\ bir\ başlangıç\ noktası\ seç$

}

**else** {

$t_i^k = t_{i-1}^k$ "$şehrinden\ sonra\ gidilecek\ şehri\ Denklem$ 1.11. $göre\ seç$

$$p_{ij}^k = \begin{cases} \dfrac{\tau_{ij}^{\alpha} + \eta_{ij}^{\beta}}{\sum_{(il)\notin s_p^k} \tau_{il}^{\alpha} + \eta_{il}^{\beta}} & eğer\ (ij) \notin s_p^k, \\ 0 & değilse \end{cases} \tag{1.11}$$

$$s_p^k = s_p^k \cup (t_i^k, t_{i-1}^k) \cup (t_{i-1}^k, t_i^k) \tag{1.12}$$

}

} // $oluşum\ adımı\ bitti$

$$\Delta\tau_{ij}^k = \begin{cases} \dfrac{Q}{L_k}, & eğer\ k,\ \ (i,j) yolunu\ kullandıysa \\ 0, & d.d. \end{cases} \tag{1.13}$$

$Denklem$ 1.13. $ve\ t^k$ çözüm $vektörüne\ göre\ \Delta\tau_{ij}^k\ \ \forall(ij)güncellenir$

}

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^k \tag{1.14.}$$

$Denklem$ 1.14. $yardımıyla\ \Delta\tau_{ij}\ \forall(ij)güncellenir$

$$\tau_{ij}^t = \rho\tau_{ij}^{t-1} + \Delta\tau_{ij} \tag{1.15.}$$

$Denklem$ 1.15.$'e\ göre\ \tau_{ij}\ \forall(ij)güncellenir$

**While** $(bitme\ koşulu\ sağlanmadı)$

Burada, $\tau_{ij}, i$ noktası ile $j$ noktası arasındaki feromen (iz) miktarını; $\eta_{ij}, i$ noktası ile $j$ noktası arasındaki hareketin cazibe değerini; m, toplam karınca miktarını; N, toplam şehir sayısını; $S_p^k, k$. karınca tarafından ziyaret edilen şehirleri; $\Delta\tau_{ij}^k$, birim zamanda $k$. karınca tarafından $(i, j)$ hattına bırakılan feromen miktarını; $L_k, k$. karıncanın mevcut çözüm turunda yola çıktığı şehre dönene kadar kat ettiği toplam mesafeyi; $t^k, k$. karıncanın çözüm turunu; $\Delta\tau_{ij}$, birim zamanda $(i, j)$ hattına bırakılan feromen miktarını; $\rho$, buharlaşma parametresini; $p_{ij}^k, k$. karıncanın $i$ şehrinden $j$ şehrine gitme olasılığını göstermektedir. $\alpha, \beta$ ve Q, kullanıcı tarafından belirlenen parametrelerdir.

Başarılı karınca kolonisi algoritmalarından biri de 1996 yılında Stützle ve Hoos tarafından önerilen Min-Max Karınca Sistemidir[153]. Bu sistemin, karınca sisteminden farkı aşağıda anlatılmıştır[154]:

- ✓ Bir çevrimde en iyi sonuçların kullanılabilmesi için her çevrimde tek bir karınca feromon izini günceller. Bu karınca, ya mevcut çevrimdeki en iyi sonucu bulan karınca (çevrim_eniyi) ya da algoritmanın başından beri en iyi sonuç bulan karınca (global_eniyi) olmalıdır.
- ✓ Çözüm uzayındaki tıkanmalardan kaçınmak için mümkün feromen izlerinin güncellemesi belirli bir aralık içinde kısıtlanır. Böylece güncellenen feromen aralık dışında değer alamaz.
- ✓ Bu sistemde başlangıç feromen değerleri max değere eşitlenerek algoritmanın daha geniş çözüm araştırması sağlanmıştır.

Diğer bir başarılı karınca kolonisi algoritması ise karınca koloni sistemi algoritmasıdır. Bu algoritma 1995'de Dorigo ve Gambardella tarafından ortaya atılan

[153] T. Stützle, ve H.H Hoos,. "Improving the Ant System: A detailed report on the MAX–MIN Ant System," FG Intellektik, FB Informatik, TU Darmstadt, Germany, **Tech. Rep. AIDA–96–12**, Aug. 1996.
[154] Tunçhan Cura, a.g.e., 2008, s.117.

Ant-Q algoritması[155] geliştirilerek aynı araştırmacılar tarafından 1996'da önerilmiştir[156]. Bu algoritmada, oluşum aşamasının sonunda gerçekleşen feromen güncellemesine ek olarak yerel feromen güncellemesi kavramı vardır. Yerel feromen güncellemesi, her oluşum adımı sonrası sistemdeki bütün karıncalar tarafından yapılır. Bu güncellemedeki amaç, iterasyon boyunca ardışık karıncaların gerçekleştirdikleri aramayı çeşitlendirmektir[157].

Yukarıda kısaca anlatılan karınca algoritmaları dışında Elitizm Karınca Sistemi[158,159], Sıralama Temelli Karınca Sistemi[160], ANTS algoritması[161], BWAS algoritması[162] ve Hyber-cube Karınca Sistemi[163] algoritmaları vardır[164].

Karınca Kolonisi Optimizasyonu, Dorigo ve arkadaşları[165,166] tarafından kombinasyonal optimizasyon problemlerinin çözümü için formüle edilmiştir. Algoritma üç aşamadan oluşmaktadır. İlk aşama karınca çözüm yapısının oluşturulması aşamasıdır. Bu aşamada karıncalar tarafından rasgele çözümler oluşturuluyor. İlk önce boş olan çözüm rotası, stokastik bir mekanizma yardımıyla seçilen uygun çözümlerin eklenmesiyle bulunmaktadır. Stokastik kuralı farklı karınca kolonisi optimizasyonu algoritmalarına göre farklılık göstermektedir. İkinci

---

[155] L. M. Gambardella and M. Dorigo, **"Ant-Q: A reinforcement learning approach to the traveling salesman problem,**" in Proceedings of the Twelfth International Conference on Machine Learning (ML-95), A. Prieditis and S. Russell, Eds. Morgan Kaufmann Publishers, 1995, s.252–260.

[156] M. Dorigo, ve L.M. Gambardella, , "Ant colonies for the traveling salesman problem," **BioSystems**, 43(2), 1997, s. 73–81.

[157] Marco Dorigo ve dğr., Ant Colony Optmization: Artificial Ants as a Computational Intelligence Technique, IRIDIA-Technical Report Series, **TR/IRIDIA/2006-023**, September 2006, s. 4.

[158] Marco Dorigo ve dğr., a.g.e., 1991.

[159] Marco Dorigo, **"Optimization, learning and natural algorithms (in italian),"** Ph.D. dissertation, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1992.

[160] B. Bullnheimer, ve dğr., "A new rank based version of the Ant System — a computational study," Institute of Management Science, University of Viena, **Tech. Rep.,** 1997.

[161] V. Maniezzo, "Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem", **INFORMS Journal on Computing**, 11(4), 358–369, 1999.

[162] O. Cordon ve dğr., **"A new ACO model integrating evolutionary computation concepts: The best-worst Ant System,**" in Abstract proceedings of ANTS 2000 – From Ant Colonies to Artificial Ants: Second International Workshop on Ant Algorithms, M. Dorigo et al., Eds. IRIDIA, Universit´e Libre de Bruxelles, Belgium, 2000, s. 22–29.

[163] C. Blum ve dğr., "HC–ACO: The hyper-cube framework for Ant Colony Optimization," in **Proceedings of MIC'2001 –Metaheuristics International Conference**, Porto, Portugal, 2, 399–403, 2001.

[164] Marco Dorigo ve dğr., a.g.e., 2006, s.2.

[165] Marco Dorigo ve dğr., a.g.e., 1999.

[166] Marco Dorigo ve G. Di Caro, **"The Ant Colony Optimization metaheuristic,"** in New Ideas in Optimization, D. Corne et al., Eds. McGrawHill, London, UK, 1999, s. 11–32.

aşamada lokal arama ile çözüm iyileştirilmektedir. Bu aşamanın yapılması tamamen algoritma geliştiriciye bağlıdır. Üçüncü aşama ise feromen miktarlarının güncellenmesi ile ilgilidir. Bu aşamadaki amaç, kötü çözümlerin feromen miktarlarını azaltılarak ve iyi çözümlerin feromen miktarlarını artırarak çözümlerin iyileştirilmesidir. Bu üç aşama durdurma koşulu sağlanana kadar devam etmektedir[167].

### 1.2.2.2. Parçacık Sürü Optimizasyon Algoritması

Sürü zekası tekniklerinden biri olan parçacık sürü optimizasyonu (PSO) algoritması, kuş ve balık sürüleri gibi ortak hareket eden sosyal hayvanların davranışlarından esinlenilerek ortaya atılmıştır. PSO, hayvanlar arasındaki işbirlikçi çalışmayı esas alan populasyon temelli paralel çalışan evrimsel hesaplama tekniğidir[168].

Kennedy ve Eberhart tarafından 1995 yılında ortaya atılan PSO bir stokastik hesaplama tekniğidir[169]. Genetik Algoritma gibi evrimsel hesaplama yöntemleriyle benzerlik göstermektedir. GA'da olduğu gibi rasgele üretilen bir başlangıç popülasyonu oluşturulur ve her iterasyon sonunda iyileştirilmeler yapılıp optimum nokta bulunmaya çalışılır. Benzerliğin yanında PSO'da çaprazlama ve mutasyon gibi genetik operatörler yoktur. PSO'da parçacık diye adlandırılan popülasyon bireyleri, çok boyutlu bir arama uzayı içinde hem kendi geçmiş bilgilerini hem de komşularının bilgilerini değerlendirip hareket ederler[170,171].

PSO algoritması, her bir parçacığın probleme ait muhtemel çözümleri temsil ettiği bir populasyon ile çalışır. Evrimsel hesaplama literatüründe birey diye adlandırılan aday çözümler PSO'da parçacık, tüm aday çözümleri içeren popülasyon

[167] Marco Dorigo ve dğr., a.g.e., 2006, s.3.
[168] M. Belal ve dğr., a.g.e., 2006, s.61.
[169] J.Kennedy ve R.C. Eberhart, "Particle Swarm Optimization", **Proc. IEEE int'l Conf. On Neural Networks**, vol IV, 1942-1948, 1995.
[170] S.N. Sivanandam ve S.N. Deepa, a.g.e., 2008, s.403.
[171] M. Belal ve dğr., a.g.e., 2006, s.61.

ise sürü diye ifade edilmektedir. Arama uzayında parçacıkların hareketini iki faktör belirlemektedir: Parçacıkları pozisyonları ve hızları.

Optimizasyonun yönlendirilmesinde hız vektörü etkilidir. Hız vektörünün hesaplanmasında önemli iki bileşen vardır. Birincisi parçacığın kendi tecrübelerinden elde ettiği deneysel bilgiyi içeren bilişsel bileşen, diğeri ise sosyal olarak komşularından elde ettiği bilgileri içeren sosyal bileşendir. Orijinal PSO, komşuluk boyutlarına göre ikiye ayrılmaktadır. Komşuluk boyutu eğer tüm arama uzayını içeriyorsa buna global en iyi PSO (geniyiPSO) adı verilirken, komşuluk sadece belirli boyuttaki arama uzayını içeriyorsa bölgesel en iyi PSO (leniyiPSO) adını almaktadır. Global en iyi PSO, yıldız topolojisini kullanırken, bölgesel en iyi PSO, halka topolojisini kullanmaktadır. Global en iyi PSO, çok geniş parçacık ilişkisinden ötürü daha çabuk yakınsamaktadır ama yerel minimuma takılma riski vardır. Bölgesel en iyi PSO, daha yavaş hareket etmesine rağmen yerel minimuma takılma riski azdır[172].

Klasik PSO algoritması ait parçacıkların pozisyonları ve hızları aşağıdaki denklemler yardımıyla hesaplanmaktadır.

$$v_i(t+1) = \alpha v_i(t) + b_1 r_1 \left(p_i^{(1)} - x_i(t)\right) + b_2 r_2 \left(p_i^{(2)} - x_i(t)\right) \quad (1.16.)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (1.17.)$$

Burada, $v_i(t)$, $i$ parçacığına ait hız vektörünü; $x_i(t)$, $i$ parçacığına ait pozisyon bilgisini; $\alpha$, atalet (eylemsizlik ağırlığı) katsayısını; $b_1, b_2$, pozitif ivme sabitlerini; $r_1, r_2$, [0,1] arasında uniform dağılıma sahip rasgele sayıları; $p_i^{(1)}, i.$ parçacığa ait en iyi pozisyon bilgisini; $p_i^{(2)}$, global PSO'da tüm parçacıklar içindeki en iyi pozisyon bilgisini, bölgesel PSO'da ise komşular arasındaki herhangi bir parçacığa ait en iyi pozisyon bilgisini ifade etmektedir.

[172] Andries Engelbrecht, a.g.e., 2007, s.289-290.

Pozitif ivme sabitleri, bilişsel ve sosyal bileşenlerin katkısını ölçmek amacıyla geliştirilmiştir. Parçacığın maksimum adım sayısını kontrol eder. Genellikle bu iki sayı ($b_1 = b_2 = 2$) birbirine eşit olacak şekilde 2 değerini alır[173].

Atalet katsayısı, sürünün global araştırma ve bölgesel araştırma yeteneklerini kontrol etmek amacıyla geliştirilmiştir. Önceki uçuş yönünün yeni hızı ne kadar etkileyeceğini belirler. $\alpha > 1$ olması durumunda hızlar zamanla artar ve maksimum hıza ulaşır. Böylelikle sürü ıraksar. $\alpha = 0$ durumunda hız kavramı tamamen ortadan kalkar. Parçacık geçmiş hız bilgisini kullanmaz ve herhangi bir yöne hareket eder. $\alpha < 1$ durumunda ise hızlı yön değişimlerine izin verilir[174].

PSO algoritmasında ilk önce her bir parçacık için konum ve hız değerleri rasgele olarak atanır. Her bir adımda hız ve konum değerleri güncellenip uygunluk fonksiyonu değerleri hesaplanır. Parçacıkların yerel en iyi değeri ve sürünün global en iyi değeri güncellenir. Belirli bir iterasyon sonucunda sürünün en iyi değeri PSO'nun probleme sunduğu çözüm olur. PSO algoritmasının aşamalı gösterimi aşağıda gösterilmektedir[175].

Adım 1 : Arama uzayında parçacıkları içeren sürü rasgele olarak belirlenir

Adım 2 : Her bir parçacığın uygunluk fonksiyonu değeri hesaplanır.

Adım 3 : $p_i^{(1)}$, $p_i^{(2)}$ ve $v_i(t)$ değerleri güncellenir.

Adım 4 : $v_i(t+1)$ , $x_i(t+1)$ değerleri hesaplanır.

Adım 5 : Durdurma koşulu sağlandı ise algoritmayı bitir yoksa Adım 2'ye git.

Klasik PSO algoritmasındaki hız vektörü, hem bilişsel bileşeni hem de sosyal bileşeni içeriyorsa bu modele tam(full) model, sadece bilişsel bileşeni içeriyorsa bilişsel model, sadece sosyal bileşeni içeriyorsa sosyal model olarak adlandırılmaktadır.

[173] M. Belal ve dğr., a.g.e., 2006, s.61.
[174] Derviş Karaboğa, a.g.e., 2011, s.190.
[175] S.N. Sivanandam ve S.N. Deepa, a.g.e., 2008, s.407.

PSO algoritmasının ortaya atılmasından sonra yapılan çalışmalar hep klasik algoritmanın çözüm kalitesini ve çözüm hızını geliştirmek üzere olmuştur. Yapılan geliştirmeler sonucunda var olan klasik PSO'ya bazı eklentiler yapılmıştır. Bunlar, eylemsizlik ağırlığı (atalet katsayısı), hız engelleme ve hız sınırlamadır. Atalet katsayısından daha önceden bahsettiğimiz için sadece hız engelleme ve hız sınırlama eklentilerinden kısaca bahsedilecektir.

PSO'nun özellikle komşuluk içerisindeki en iyi ve kişisel en iyi pozisyonlarından uzakta kalan parçacıklar hızlı bir şekilde büyük değerler almaktadır. Bu da parçacıkların büyük değişimler yaşamasına ve ıraksamasına neden olmaktadır. Parçacıkların küresel arama kabiliyetini sürdürebilmesi için hızlarının belirli değerler içinde olması gerektiği vurgulanmıştır. Bunun için parçacıkları hızları belirli bir değeri aşarsa o parçacığa ait hız o değere eşitlenmektedir (Denklem 1.18.).

$$v_{ij}(t+1) = \begin{cases} v'_{ij}(t+1), & v'_{ij}(t+1) < V_{max,j} \\ V_{max,j}, & v'_{ij}(t+1) \geq V_{max,j} \end{cases} \qquad (1.18.)$$

Hız sınırlama yöntemi ise eylemsizlik ağırlığına benzer bir yöntemdir. Burada hızlar sabit bir sayı ile hızlandırılmaktadır. Bu sayıya sınırlama katsayı adı verilmektedir. $\chi$'nin küçük değerleri algoritmanın küresel araştırma yeteneğini azaltıp bölgesel araştırma yeteneğini artırmaktadır. Büyük değerlerde ise tersi durum söz konusudur. Bu model verilen kısıtlar içinde yakınsamayı garanti etmektedir. Söz konusu modele ilişkin denklemler aşağıda verilmiştir[176]

$$v_{ij}(t+1) = \chi\left[v_{ij}(t) + \phi_1\left(y_{ij}(t) - x_{ij}(t)\right) + \phi_2\left(\hat{y}_j(t) - x_{ij}(j)\right)\right] \qquad (1.19.)$$

$$\chi = \frac{2K}{\left|2 - \phi - \sqrt{\phi(\phi-4)}\right|} \qquad \phi \geq 4, \qquad K \in [0,1] \qquad (1.20.)$$

$$\begin{aligned} \phi &= \phi_1 + \phi_2 \\ \phi_1 &= b_1 r_1 \\ \phi_2 &= b_2 r_2 \end{aligned} \qquad (1.21.)$$

[176] Derviş Karaboğa, a.g.e., 2011, s.191.

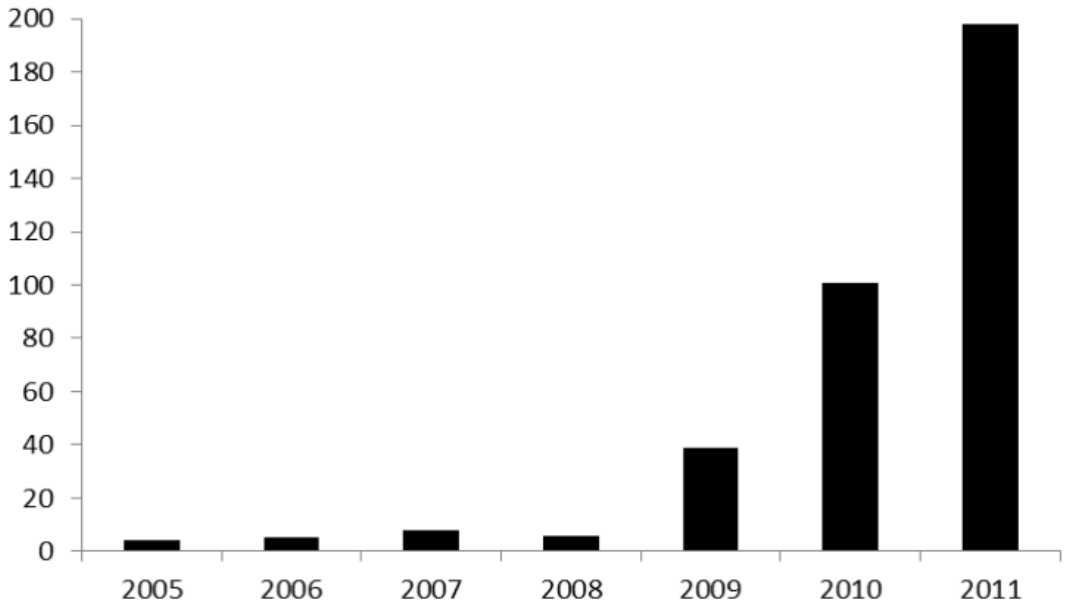
### 1.2.2.3. Arı Kolonisi Optimizasyon Algoritması

Yapay Arı Kolonisi Algoritması (YAK), 2005 yılında Derviş Karaboğa tarafından optimizasyon problemlerinin çözümü için geliştirilen popüler bir algoritmadır. Sürü zekası algoritmalarından biri olan YAK, arıların besin arama davranışlarından esinlenmektedir. Yeni bir algoritma olmasına rağmen oldukça hızlı gelişmiştir. Yıllar itibariyle YAK ile yapılmış çalışma sayıları Şekil 1.6.'da yer almaktadır.

Orijinal YAK algoritması dışında literatürde birçok arı ile ilgili algoritmalar mevcuttur. Arı Algoritması, Arı Sürüsü Algoritması, Arı Kolonileri Optimizasyon Algoritması, Sanal Arı Kolonisi Algoritması, Arı Kovanı Algoritması, Bal Arıları Çiftleşme Optimizasyon Algoritması bu algoritmalardan bazılarıdır[177].

YAK bu tezin üçüncü bölümünü oluşturan Yapay Arı Kolonisi Algoritması başlığı altında ayrıntılı olarak incelenecektir.

**Şekil 1.6. Yıllara göre YAK ile yapılmış yayın sayısı**




**Kaynak:** Derviş Karaboğa ve dğr., A comprehensive survey: artificial bee colony (ABC) algorithm and applications, **Artificial Intelligence Review**, Springer Science+Business Media B.V., 2012, s.23.

[177] M. Marinaki ve dğr., "Honey Bees Mating Optimization algorithm for financial classification problems", **Applied Soft Computing**, 10, 806-812, 2010, s. 806.

# 2. YAPAY ARI KOLONİSİ ALGORİTMASI

## 2.1. Doğada Bulunan Arılar

Doğadaki en ilginç sürülerden biri de bal arılarıdır. Sosyal bir çevrede yaşayan arılar, kendine kendilerine içgüdüsel olarak bir düzen kurar ve bu düzen içinde hareket ederler. Kovan içinde bulunan her arının belirli bir görevi vardır. Bu görev başka bir görev verilene kadar sadık bir şekilde yapılmaya devam eder. Bal arıları fotoğrafik hafızaya, uzay çağı sensörlerine, navigasyon sistemine, sezgisel kavrama yeteneğine, yeni yuva seçimi sırasında kullanılan grup olarak karar verme yeteneğine sahiptir. Bunların dışında yiyeceklerin saklanması, balın yuvaya getirilmesi ve dağıtılması, yiyecek arama sürecinin ve birbirleri arasındaki iletişimin en iyi şekilde yerine getirilmesinden sorumludur.

Kovan içerisinde bulunan arılar üç sınıfa ayrılmaktadır[1]: Kraliçe Arı (Queen), Erkek Arı (Drone) ve İşçi Arı (Worker). Bunlar hakkında kısa bilgi aşağıda verilmektedir.

**Kraliçe Arı (Queen):** Kovan içinde yumurtlama yeteneğine sahip tek arıdır. Bu özelliğinden dolayı koloni için önemli bir arıdır. Bir kovan içinde sadece bir tane kraliçe arı vardır. Fiziksel olarak erkek arı ve işçi arılardan uzundur. Kraliçe arı, salgıladığı maddelerle koloni bütünlüğünü ve kovan içindeki işleyişi kontrol eder. Kraliçe arı salgıladığı bu madde sayesinde kovan içindeki arıları işaretlemiş olur. Bu sayede kovana giriş yapan yabancı arılar fark edilir. Böylelikle kraliçe arı kolonin bütünlüğünü sağlamış olur. Kraliçe arı, ömrü boyunca sadece bir kere erkek arı ile çiftleşmektedir. Çiftleşmeden sonra aldığı spermleri vücudunun belirli bölümünde depo etmektedir ve bu spermlerle 2-3 yıl boyunca yumurta üretebilmektedir. Sağlıklı bir kraliçe arı günde 2000'e yakın yumurta üretmektedir. Yıl bazında bu rakam

[1] Derviş Karaboğa ve Bahriye Akay, "A survey: algorithms simulating bee swarm intelligence", **Artif Intell Rev**, 31, 61-85, 2009, s.63.

ortalama 175.000-200.000 arasında değişmektedir[2]. Kraliçe arının üreme dışındaki bir diğer önemli görevi koloni büyüklüğünün kontrol edilmesidir. Öyle ki eğer kolonideki besin kaynakları azalmışsa yumurta üretmeye başlar, eğer koloni kalabalıklaşmışsa yumurtlamayı durdurur. Kraliçe arı senelerce yaşayabilir. Yeni kraliçe arının seçilmesinde de rol oynamaktadır. Dişi arılardan birini yumurtlamanın devamlılığı için kraliçe arı olarak seçer.

**Erkek Arılar (Drones):** Erkek arıların tek görevi kraliçe arı ile çiftleşmektir. Bunun için koloninin babası olarak da adlandırılmaktadır. Yazın bir kolonide birkaç yüz erkek arı vardır. Diğer arılara göre daha iridirler. Arıların savunma amaçlı kullandıkları iğneye sahip değillerdir. Ayrıca besin toplayacak organları yoktur. Erkek arılar diğer arılardan farklı olarak döllenmemiş yumurtadan meydana gelmektedir. Altı aydan fazla yaşayamamaktadırlar. Kraliçe arı ile çiftleşme gerçekleştikten sonra hemen ölürler.

**İşçi Arılar (Workers):** İşçi arılar koloninin sağlıklı bir yaşam sürmesinde önemli görevlere sahip arılardır. Bu görevler besinlerin toplanması, toplanan besinlerin saklanması, kovanın ölü arı ve molozlardan temizlenmesi, havalandırılması ve güvenliliğinin sağlanması olarak sıralanabilir. Bunların dışında kraliçe arının yumurtladığı yer olan bal mumu hücrelerini yapar. Tükürük bezlerinde bulunan bir salgı sayesinde kraliçe arıyı, erkek arıları ve larvaları besler. İşçi arının yapacağı iş koloninin ihtiyacına ve işçi arının yaşına bağlı olarak değişmektedir. Yaşamlarının ikinci yarısında işçi arılar yiyecek arayıcı olarak çalışırlar. İlk önce yuvadan ayrılarak kısa uçuşlar gerçekleştirir. Bu uçuşlarla çevrenin topolojisini ve yuvanın yerini tespit ederler. İşçi arılar dişi arılardır. Fakat üreme yetenekleri yoktur. Kolonide en çok bulunan arı türüdür. Yaz aylarında 6 hafta, kış aylarında ise 4-9 ay arasında yaşayabilmektedirler.[3]

**Çiftleşme Uçuşu (Mating-Flight):** Kraliçe arı, çiftleşmek üzere yuvadan uzaklara uçuş gerçekleşir. Bu uçuş kraliçe arının bir çeşit dansından sonra başlar.

[2] Derviş Karaboğa ve Bahriye Akay, a.g.e., 2009, s.63.
[3] Derviş Karaboğa ve Bahriye Akay, a.g.e., 2009, s.63.

Erkek arılar uçuş esnasında kraliçe arıyı takip eder ve çiftleşme havada uçuş esnasında gerçekleşir. Bir erkek arının kraliçe ile çiftleşme olasılığı, erkek arı ile kraliçe arının uygunluğuna ve kraliçe arının hızına bağlıdır. Kraliçe arı erkek arılardan aldığı spermleri spermetika adı verilen bir organında depolar. Bu sayede kraliçe arı tarafından üretilecek olan potansiyel yavru arıların genetik havuzu oluşturulmuş olur.

**Besin Arama (Foraging):** Besin arama, kovandaki en önemli görevdir. Besin arama davranışlarını, koku, kuyruk (waggle) dansından gelen konum bilgisi, besin kaynaklarında ya da kaynak yolunda diğer arıların bulunması gibi dışsal bilgiler ile hatırlanan kaynak konumu ya da kokusu gibi içsel bilgiler etkilemektedir. Bu konu hakkında çeşitli çalışmalar yapılmıştır [4,5,6]. Besin arama süreci, bir besin arayıcının kovandan ayrılmasıyla başlar. Besin alabileceği kaynağı bulduktan sonra besin arayıcı arı, nektarı karın bölgesine depolar. Bu depolama süreci kaynağın zenginliğine ve kaynağın kovana uzaklığına bağlıdır. Yaklaşık olarak 30-120 dakika boyunca besin arayıcı arı nektarı karın bölgesine doldurur. Besin arayıcı arı salgıladığı enzimler yardımıyla karnındaki nektarı bal yapmaya başlar. Kovana döndükten sonra nektarı boş petek hücrelerine boşaltır. Nektarı, bakteri saldırılarından korumak ve nektarın fermantasyon geçirmemesi için ekstra maddeler ekler. Bal ve enzimlerle dolu petek hücreleri bal mumu ile kapatılır.

**Dans:** Nektar boşaltımından sonra besin arayıcı arı, kaynak hakkındaki bilgiyi diğer arılara bildirmek üzere belirli bir alanda dans adı verilen özel bir hareket sergiler. Dans esnasında diğer arılar antenleriyle birlikte besin arayıcı arıya dokunurlar ve besin kaynağıyla ilgili bilgileri elde ederler. Bu bilgiler besin kaynağının zenginliği, yuvaya uzaklığı ve yön bilgileridir. Bu bilginin tüm arılar tarafından görülebilmesi için bu dans kovanın çeşitli yerlerinde tekrar edilmektedir. Besin kaynağının uzaklığa göre arıların dansları da değişim göstermektedir. Arılarda

[4] Karl Von Frisch, **The dancing bees: an account of the life and senses of honey bee**, Harcourt, Brace&World, 1954.
[5] Karl Von Frisch ve M. Lindauer, "The "language" and orientation of the honey bee" **Annual Review of Entomology**, 1, 45–58, 1956.
[6] T. Seeley, **Honeybee ecology: a study of adaptation in social life**, Princeton University Press, Princeton, 1985.

üç çeşit dans görülmektedir[7]: Dairesel Dans (round dance), Kuyruk Dansı (waggle dance) ve Titreme Dansı (tremble dance).

Dairesel dans, besin kaynağının yuvanın 50-100 metre uzağında olduğu durumlarda görülen bir dans çeşididir. Kaynak yuvaya çok yakın olduğu için bu dans yön bilgilerini içermemektedir.

Titreme dansında, arı yavaş tempoda ve düzensiz tarzda bacaklarını titreterek ileri, geri, sağa ve sola hareket eder. Bu dans, besin kaynağının oldukça zengin olduğunu fakat kovanın işleyebileceğinden fazla nektarın geldiğini ve bu yüzden nektarı işleme görevine geçmek istediğini ifade etmektedir. Bu dansın amacı, kovan kapasitesi ve besin getirme süreci arasındaki dengeyi sağlamaktır.

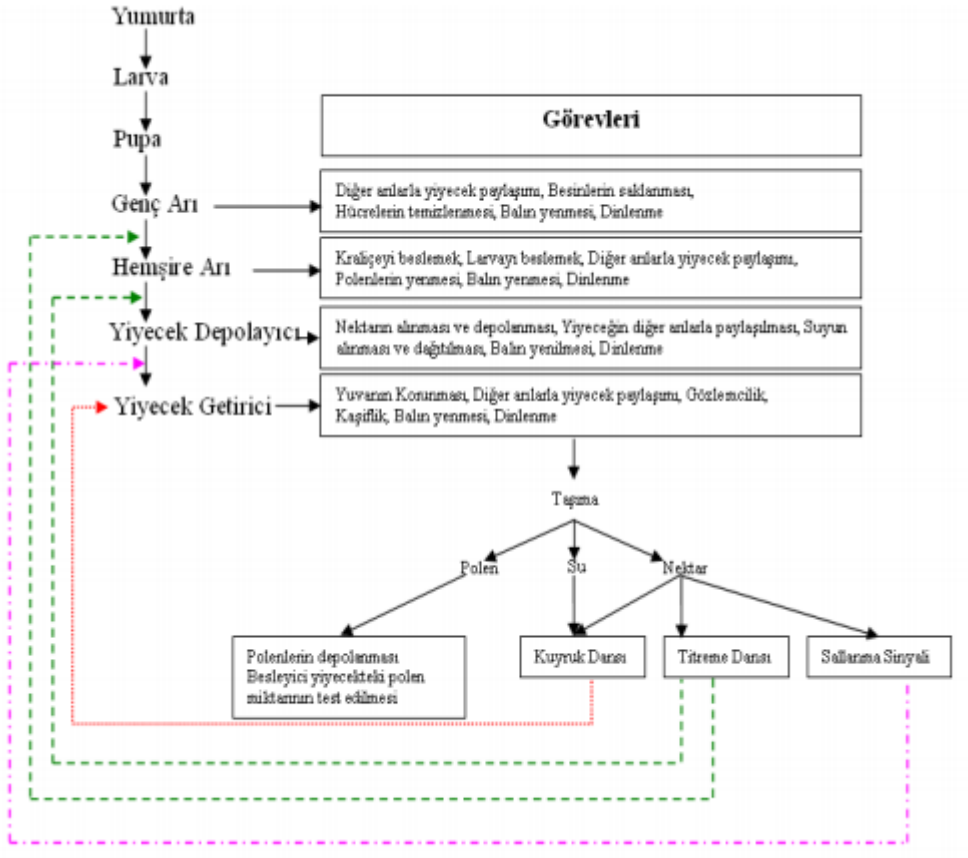
Kuyruk dansı ise 100 metreden 10 kilometreye kadarki geniş bir alan içindeki kaynaklar hakkındaki bilgi paylaşımı için kullanılır. Bu dansta besin kaynağının yönü güneşe olan açıyla hesaplanır. Her 15 saniyede dansın tekrarlanma sayısı nektarın uzaklığını ifade etmektedir. Az tekrarlanma sayısı kaynağın uzakta olduğunu belirtmektedir. Kuyruk dansı yapan arı diğer arıların bir titreşim hareketi yapması sonucunda dansına son verir.

**Görev Paylaşımı (Task Selection):** Bal arıları kolonisinde kovan içindeki işlerin yürütülmesi için belirli sayıdaki arıların belirli görevlere getirilmesine ihtiyaç vardır[8]. Her arı görevleri konusunda uzmanlaşmıştır. Yaş, hormonlar, genetik yatkınlık gibi faktörlerin hangisinin arıların uzmanlaşmasında etkili olduğu tam olarak belirlenememiştir[9]. Arılar çevrelerindeki ani değişimlere hemen cevap verebilen sosyal varlıklardır. Bir arı görevli olmadığı bir işte ihtiyaç doğrultusunda

[7] Bahriye Akay, a.g.e., 2009, s. 54-55.
[8] Madeleine Beekman vd., "What makes a honeybee scout?", **Behav Ecol Sociobiol,** 61, 985–995, 2007, s. 985.
[9] Anna Dornhaus vd., Task selection in honeybees—Experiments using multi-agent simulation. **In Proceedings of GWAL'98**, Bochum, September 18-19, 1998, s. 5.

görevli hale gelebilmektedir. Arıların kovan içindeki yaşam evreleri ve görevlerine ilişkin şema Şekil 2.1.'de gösterilmektedir[10].

**Şekil 2.1. Arılarda görev dağılımı**




**Kaynak:** Bahriye Akay, a.g.e., 2009, s.32.

Bir arının gelişim süreci, kraliçe arı tarafından yumurta olarak bırakılmasıyla başlar. Yumurta zaman içinde larva ve pupaya dönüşür. Son olarak da genç arı olarak yumurtadan çıkar. Yumurta ve larva döneminde arıların beslenme biçimi besin alanından çıktıktan sonraki görevlerinde de etkilidir. Larva beslenirken hemşire arı tarafından verilen arı sütünün zamanına göre ya dişi arı olarak kalır ya da kraliçe arı olurlar. Yumurtadan çıkan genç arının yapacağı ilk iş beslenmektir. Daha sonra yeni görevleri yapmaya başlayacaktır. Yetişkin arılar yuva çıkışında dururken, genç arı ve

[10] Bahriye Akay, a.g.e., 2009, s.32.

hemşire arılar beslenme alanında çalışırlar. Bir arının yetişken bir arı olana kadar besleme, depolama, bal ve polenlerin elde edilmesi ve dağıtılması, iletişim ve besin arama gibi görevleri vardır. Belli bir anda bir arının hangi işi gerçekleştireceği o andaki davranışsal rolüne, çevreden topladığı algılara ve bu algılara verdiği tepki eşiğine bağlıdır. Davranışsal roller, arılar arasındaki fizyolojik farklılıklardan modellenmiştir. Bundan dolayı bazı arıların görevleri gereği bazı organları diğer arılarınkinden daha gelişmiş olabilir[11].

## 2.2. Doğadaki Arıların Besin Arama Davranışları

Karmaşık ve sürekli değişen çevre koşulları altında zekice düşünülmüş kararlar alabilme yeteneğine sahip arı kolonilerinin yapması gereken ana görevlerinden biri besin kaynaklarının aranmasıdır. Bal arıları, farklı besin kaynakları içinde kovana daha faydalı olacak besin kaynağını seçer. Bu seçimde kaynaktaki şeker miktarı, nektarın zenginliği ve yuvaya yakınlığı rol oynar. Arıların besin arama davranışlarını Tereshko, reaktif difüzyon denklemlerine dayandırarak modellemiştir[12]. Bu model üç temel bileşenden oluşmaktadır. Bunlar besin kaynağı, görevli işçi arılar ve görevsiz işçi arılardır. Ayrıca bu modelin besin kaynağına yönelme ve besin kaynağını terk etme adı verilen iki tane davranış modu söz konusudur. [13]

**Besin Kaynağı:** Yuvaya bal, polen ve su taşıyan arılar geniş bir arazide besin kaynaklarını tespit ederler. Besin kaynağının değerine göre kendilerine en uygun olan besin kaynağını seçerler. Besin kaynağının değerini belirleyen arılar birçok faktörü değerlendirmesi gerekmektedir. Yuvaya yakınlık, nektarın yoğunluğu ya da zenginliği, nektarın yuvaya taşınmasındaki kolaylık bu faktörlerden bazılarıdır. Tereshko ve Loengarow'un birlikte önerdikleri modelde kolaylık olması açısından sadece nektarın zenginliği faktörü kullanılmıştır.

---

[11] Anna Dornhaus vd., a.g.e., 1998, s. 5-7.

[12] Valery Tereshko, Reaction-diffusion model of a honeybee colony's foraging behaviour, M. Schoenauer, et al, Eds., **Parallel Problem Solving from Nature VI** (Lecture Notes in Computer Science,Vol. 1917) Springer-Verlag: Berlin, 2000, s. 807-816.

[13] V. Tereshko ve A. Loengarov, "Collective decision making in honey-bee foraging Dynamics". **Computing and Informaton Systems**, 9(3), 1-7, 2005, s.2.

**Görevli İşçi Arılar:** Tüketme olduğu ya da görevlendirildiği besin kaynağını yuvaya getirmekten sorumlu olan arılardır. Bu arılar kaynak hakkındaki bilgileri de taşımaktadır. Kaynağın yuvaya olan uzaklığı, yönü ve koordinatları, kaynağın zenginliği bilgisini görevli arıların, kovandaki diğer arılarla paylaşması belirli olasılıklara bağlıdır. Kaynak ne kadar zengin ise, dans alanında yapılacak kuyruk dansının olasılığı o kadar yüksektir. Görevli arılar yapmış oldukları işi bitirdiklerinde görevsiz arı olurlar.

**Görevsiz İşçi Arılar:** Görevsiz arılar tüketilmek üzere besin kaynağı ararlar. İki çeşit görevsiz arı vardır: Kaşif arı ve gözcü arı. Kaşif arılar, herhangi bir bilgi olmadan yeni besin kaynaklarını tespit etmeye çalışır. 14 km ve fazlası gibi geniş bir alanda arama yapabilmektedir. Gözcü arılar ise, yuvada bekler ve görevli arılar tarafından yapılan dansla kaynak hakkındaki bilgiyi elde eder. Ortalama olarak bir kovanda yaklaşık %10 oranında kaşif arı bulunmaktadır[14].

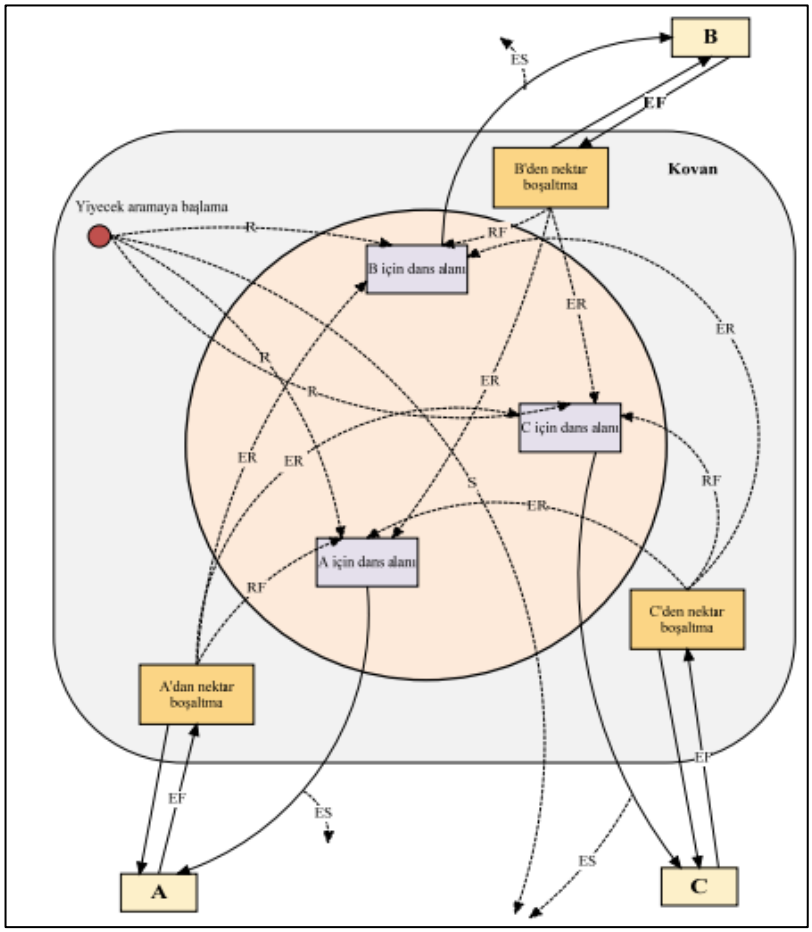
Arıların birbirleriyle bilgi alış verişi yapması kollektif bilgiyi kullandıklarını göstermektedir. Kovan içinde bulunan birçok bölümden bir tanesi de bu bilgi paylaşımın yapıldığı dans alanıdır. İşçi arılar taşımakla yükümlü olduğu besin kaynağı hakkında bu dans alanında dans ederek onu izleyen gözcü arılara bilgi verir. Dansı izleyen gözcü arılar hakkında bilgi verilen besin kaynağına yönlenebileceği gibi yerinde durup diğer arıların danslarını izleyebilir. Bu durum gözcü arının kendine karlı bir besin kaynağı seçene kadar devam eder. Kaynağın zenginliği işçi arının yapacağı dansın uzunluğuna bağlı olarak değişim göstermektedir. Dansın uzunluğu da besin kaynağının yuvaya uzaklığına bağlıdır. İşçi arılar tarafından yapılan danslar taşınan bilgiye göre değişmektedir. Bu konudaki en önemli faktör nektarın tatlılığıdır. Nektarın çıkarılmasındaki kolaylık, yuvaya yakınlığı, nektarın kıvamı, hava koşulları, günün hangi saatinde olduğu dansı etkileyen diğer faktörlerdir. Arılar besin kaynağına ulaşmak konusunda güneşten yararlanmaktadırlar. Güneş ışığı ile kendi koordinatları arasındaki açıyı hesaplayarak hareket etmektedirler. Ayrıca arılar, harcadıkları enerji tüketimini göz önünde

[14] V. Tereshko ve A. Loengarov, a.g.e., 2005, s. 2.

bulundurarak kaynağa ulaşabilmek için farklı yüksekliklerde uçarak enerjilerini ayarlarlar.[15]

Arıların besin toplama sırasında sergiledikleri davranışların görsel gösterimi Şekil 2.2.'de gösterilmiştir.

**Şekil 2.2. Arıların besin arama davranışları**




**Kaynak:** Pınar Z. Tapkan, Kombinatoryal optimizasyon problemlerinde arı sistemi yaklaşımı, Erciyes Üniversitesi, Yayımlanmamış Doktora Tezi, Kayseri, 2010, s.28

[15] Derviş Karaboğa, a.g.e., , 2011, s. 202.

Besin arayıcı arı, besin arama sürecine görevsiz arı olarak başlar. Kovan çevresindeki besin kaynakları hakkında herhangi bir bilgiye sahip olmadan dolanmaktadır. Bu arının iki seçeneği vardır:

- ✓ Kaşif arı olarak yoluna devam eder ve hiçbir ön bilgi olmaksızın çevrede besin kaynaklarını araştırmaya başlar (S).
- ✓ Gözcü arı olarak yoluna devam eder. Diğer arılar tarafından sergilenecek olan kuyruk danslarını izler, en karlı besin kaynağını seçer ve kaynağa yönelir (R).

Besin kaynağına ulaşan arı, kaynağın yerini hafızasında tutarak hemen kaynağı tüketmeye başlar. Bundan sonra görevli arı olarak çalışmaya başlar. Besin kaynağından bir miktar nektar alarak yuvaya geri döner ve besin deposuna nektarı boşaltır. Boşaltma işlemini tamamlayan görevli arı aşağıdaki 4 durumdan birini seçerek yoluna devam eder.

- ✓ Kaynaktaki nektarın azalmasından dolayı ilgili kaynağı terk eder ve görevsiz arıya dönüşür (ES).
- ✓ İlgilendiği besin kaynağına dönmeden önce dans ederek bu kaynak hakkında diğer arılara bilgi verir (RF).
- ✓ Herhangi bir bilgi paylaşımı yapmadan ilgilendiği besin kaynağına yönlenir (EF).
- ✓ İlgilendiği besin kaynağını terk edip dans alanından yeni besin kaynakları hakkında bilgi alır (ER).

## 2.3. Yapay Arı Kolonisi Algoritması

Hayvanların ve böceklerin doğada var olma çabaları birçok araştırmacının dikkatini çekmiştir. Özellikle besin arama sürecinde kullandıkları zeki davranışlar optimizasyon teknikleriyle yakından ilgilenen araştırmacılara ilham kaynağı olmuştur. Bu tekniklerden biri de arıların davranışlarından esinlenilerek literatüre

kazandırılan Yapay Arı Kolonisi Algoritmasıdır (YAKA). Algoritma, 2005 yılında Derviş Karaboğa tarafından modellenerek oluşturulmuştur[16]. Gerçek arıların besin arama süreçlerinin karmaşıklığını sadeleştirerek modeli basitleştirmiştir. Ortaya atılan bu yeni model bazı varsayımlar içermektedir. Bunlardan biri besin kaynağı nektarının sadece bir arı tarafından toplanıyor olmasıdır. Bu durumda besin kaynaklarının sayısı, görevli arı sayısına eşit olmaktadır. Diğer varsayım ise gözcü arılarının sayısı görevli arıların sayısına eşit olmasıdır. Besin kaynaklarının yerleri optimizasyon problemindeki olası çözümlere denk gelirken kaynaktaki nektar zenginliği, çözümlerin kalitesine karşılık gelmektedir[17]. YAKA, çok boyutlu ve çok doruklu optimizasyon problemlerinin çözümü için geliştirilmiş popülasyon tabanlı bir algoritmadır. YAKA'da görevli arı, kaşif arı ve gözcü arı olmak üzere üç tip arı bulunmaktadır. Koloninin yarısı görevli arılardan diğer yarısı ise gözcü arılardan oluşmaktadır. Görev arı sayısı da besin kaynağı sayısına yani olası çözümlerin sayısına eşittir. Kaşif arılar ise görevli arıların işleri bitince ortaya çıkmaktadır. Yeni besin kaynaklarını ön bilgi olmadan rasgele olarak bulduğundan algoritmanın daha önceden keşfedemediği zengin besin kaynaklarını yani yüksek uygunluktaki olası çözümler kümesini bulabilir[18].

YAKA besin kaynaklarının yani olası çözüm değerlerini arama sürecinde 4 farklı seleksiyon yöntemi kullanmaktadır[19].

*Küresel Seleksiyon İşlemi:* Gözcü arıların gidecekleri besin kaynağının seçiminde kullanılan olasılık değerlerinin hesaplanması

*Bölgesel Olasılık Tabanlı Seleksiyon İşlemi:* Besin kaynağının komşuluğunun hesaplanması

*Aç Gözlü Seleksiyon İşlemi:* Uygunluk değerlerinin kıyaslanması

[16] Derviş Karaboğa, An Idea Based On Honey Bee Swarm For Numerical Optimization, TR-06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
[17] Derviş Karaboğa, a.g.e., 2011, s. 206.
[18] Pınar Z. Tapkan, a.g.e., 2010, s. 29.
[19] Bahriye Akay, a.g.e., 2009, s. 62-63.

*Rasgele Seleksiyon İşlemi:* Kaşif arıların yeni besin kaynağı seçimi

YAKA ile ilgili temel adımlar aşağıda yer almaktadır.

**Adım 1:** Rasgele olarak besin kaynakları oluşturulur. Besin kaynak sayısı kadar görevli arı ve gözcü arı oluşturulur. Besin kaynağının bırakılmasında kullanılacak olan limit değerleri tanımlanır ve bir üst limit değeri tespit edilir ve son olarak döngü sayısını oluşturan sayaç değişkeni tanımlanır.

**Adım 2:** Elde edilen besin kaynakların olası çözümleri probleme özgü amaç fonksiyonuna göre hesaplanır.

**Adım 3:** Bu adımda döngü başlamaktadır. Bunun için maksimum döngü sayısı belirlenerek görevli arılar besin kaynaklarına gönderilir. Arılar bu kaynağı işler ve kalitesini ölçerler. Böylelikle çözüm değerleri oluşur. Bu değerler içinde en iyi olan önceki çözüm değeriyle kıyaslanır. Eğer yeni çözüm değeri, bir öncekinden iyi ise en iyi çözüm olarak hafızada tutulur ve limit değeri sıfırlanır. Tersi durumda limit değeri bir arttırılır. Limit değeri üst değeri aşıp aşmadığı kontrol edilip algoritmanın sonsuz döngüye girmesi engellenir.

**Adım 4:** Görevli arılardan sonra gözcü arılar besin kaynaklarına yönlendirilmektedir. Gözcü arıların hangi besin kaynağına gidecekleri görevli arılar tarafından getirilen bilgiye göre hesaplanan olasılıksal değerlerle belirlenmektedir. Gözcü arı gittiği kaynağın kalitesini ölçer ve çözüm değerlerini hesaplar. Bu değerler içinde en iyi olan önceki çözüm değeriyle kıyaslanır. Eğer yeni çözüm değeri, bir öncekinden iyi ise en iyi çözüm olarak hafızada tutulur ve limit değeri sıfırlanır. Tersi durumda limit değeri bir arttırılır. Limit değeri üst değeri aşıp aşmadığı kontrol edilir.

**Adım 5:** Görevli arı ve gözcü arı adımlarından sonra kaşif arı aşamasına geçilir. Kaşif arılar tamamen yeni besin kaynakları belirler. Böylelikle algoritmanın yerel minimum ve maksimum noktalarına takılması engellenir. Yeni besin

kaynaklarının çözüm değerleri hesaplanır ve bir önceki çözümle kıyaslanır. Eğer yeni çözüm değeri, bir öncekinden iyi ise en iyi çözüm olarak hafızada tutulur ve limit değeri sıfırlanır. Tersi durumda limit değeri bir arttırılır. Limit değeri üst değeri aşıp aşmadığı kontrol edilir.

**Adım 6:** Maksimum döngü sayısına ulaşana kadar görevli arı, gözcü arı ve kaşif arı adımları tekrar edilir.

Algoritmanın adımları incelendiğinde algoritmayı 4'e ayırmak mümkündür. Bunlar sırasıyla başlangıç besin kaynaklarının belirlenmesi, Görevli arıların besin kaynaklarına gönderilmesi, gözcü arıların olasılıksal seçimle elde edilen besin kaynaklarına gönderilmesi, tükenen besin kaynaklarının terk edilmesi ve kaşif arıların yeni besin kaynakları üretmeleridir.

### 2.3.1. Başlangıç Besin Kaynaklarının Belirlenmesi

Algoritmaya ilk olarak besin kaynaklarının yerlerinin belirlenmesiyle başlanılır. Olası çözüm değerlerini ifade eden besin kaynaklarının yerleri aşağıdaki Denklem 3.1. yardımıyla belirlenir.

$$x_{ij} = x_j^{min} + rand(0,1)(x_j^{max} - x_j^{min}) \qquad (2.1.)$$

Burada i, besin kaynağı sayısını; j, parametre sayısını ifade etmektedir. Burada max ve min değerleri ile besin kaynağının belirli sınırlar içinde üretilmesi yani arama uzayı içinde kalması sağlanmaktadır.

### 2.3.2. Görevli Arıların Besin Kaynaklarına Yönlendirilmesi

Görevli arı ilgilendiği besin kaynağının komşuluğunda yeni bir besin kaynağı belirler ve çözüm değerlerini hesaplar. Elde edilen yeni çözüm değeri öncekinden

daha iyi ise hafızaya alınır. Yeni kaynağın belirlenmesinde Denklem 2.2. kullanılmaktadır.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{2.2.}$$

Burada her bir besin kaynağı $(x_{ij})$ için parametrelerinden bir tanesi değiştirilerek yeni bir besin kaynağı $(v_{ij})$ elde edilir. Elde edilen yeni besin kaynağının çözüm uzayı içinde kalması Denklem 2.3. kullanılmaktadır.

$$v_{ij} = \begin{cases} x_j^{min}, & v_{ij} < x_j^{min} \\ v_{ij}, & x_j^{min} \leq v_{ij} \leq x_j^{max} \\ x_j^{max}, & v_{ij} > x_j^{max} \end{cases} \tag{2.3.}$$

Bu bilgiler ışığında besin kaynağının uygunluk değeri aşağıdaki denklem yardımıyla hesaplanmaktadır.

$$uf_i = \begin{cases} 1/(1+f_i), & f_i \geq 0 \\ 1 + abs(f_i), & f_i < 0 \end{cases} \tag{2.4.}$$

Burada $f_i$ ile besin kaynağının uygunluk değeri temsil edilmektir. Problemin türüne göre uygunluk hesabı değişim göstermektedir. Elde edilen bu uygunluk değerleri ile çözüm değerleri hesaplanmaktadır. Eğer elde edilen çözüm önceki çözümden daha iyi ise sayaç sıfırlanır ve yeni çözüm değerleri hafızaya alınır. Eğer bir iyileşme olmamış ise sayaç bir arttırılır.

### 2.3.3. Gözcü Arıların Besin Kaynaklarını Seçiminde Kullanacakları Olasılık Değerlerinin Hesaplanması

Görevli arılar besin kaynakları hakkında gözcü arılara bilgi vermektedir. Gözcü arılar bu kaynaklara yönelmeden önce tüm çözüm değerleri olasılıksal bir seçim sürecinden geçer. Birçok farklı seçim yöntemleri bulunmaktadır. YAKA rulet

çemberi seçim mekanizmasını kullanmaktadır[20]. Tekerlekteki her dilim uygunluk fonksiyona denk gelmektedir. Bu yüzden uygunluğu büyük olan besin kaynaklarının seçilme şansı yüksektir. Gözcü arıların yönelecekleri besin kaynakların seçilme olasılıkları aşağıdaki Denklem 2.5. yardımıyla belirlenmektedir.

$$\rho_i = \frac{f_i}{\sum f_i} \qquad (2.5.)$$

Bu denklemle seçilme olasılıkları hesaplanan uygunluk değerinin toplam uygunluk değerine oranlanmasıyla hesaplanmaktadır.

### 2.3.4. Gözcü Arıların Besin Kaynaklarını Seçmeleri

Elde edilen besin kaynaklarının seçilme olasılıklarına göre gözcü arılar besin kaynaklarını seçerler ve uygunluklarını hesaplarlar. Hesaplanan yeni değer önceki değerle kıyaslanır. Eğer iyileşme varsa çözüm hafıza alınıp sayacı sıfırlanır. Tersi durumda sayacı arttırılarak döngüye devam edilir. Bu döngü tüm gözcü arıların besin kaynaklarını seçmesine kadar sürer.

### 2.3.5. Besin Kaynağının Terk Edilmesi ve Kaşif Arı Üretimi

Görevli arıların ve gözcü arıların işlerini bitirmesinden hemen sonra besin kaynaklarının sayaçları kontrol edilir. Limit değerini aşan besin kaynaklarının terk edilmesi gerekmektedir. Yani besin kaynağındaki nektarlar azalmış ya da tükenmiştir. Bu durumda kaşif arılar devreye girerek yeni besin kaynakları araştırırlar. Bir başka deyişle besin kaynağındaki nektarı yuvaya taşıyan görevli arı kaynağın tükenmesiyle birlikte yeni kaynaklar aramak için kaşif arı olmaktadır. Kaşif arıların buldukları yeni besin kaynakları değerlendirilip uygunlukları hesaplanmaktadır. Elde edilen değerler iyileşme sağlıyorsa çözüm değerleri hafızaya alınıp sayaç değeri sıfırlanır aksi takdirde sayaç bir arttırılır.

[20] Bahriye Akay, a.g.e., 2009, s. 61.

Sonuç olarak YAKA oldukça basit, esnek ve sürü zekasına dayalı bir algoritmadır. Bazı varsayımları olmasına rağmen gerçek arıları iyi bir şekilde modelleyen bir algoritmadır. Az sayıda parametre içermektedir. Sürekli optimizasyon problemleri için üretilmiş olmasına rağmen ayrık optimizasyon problemlerinde de kullanılabilir. Algoritma, hem kaşif arılar tarafından gerçekleştirilen küresel arama hem de görevli ve gözcü arılar tarafından gerçekleşen bölgesel arama yeteneğine sahiptir[21].

## 2.4. Yapay Arı Kolonisi Algoritması ile İlgili Yapılan Çalışmalar

YAKA, 2005 yılında Kayseri'de bulunan Erciyes Üniversitesi Öğretim üyesi olan Derviş Karaboğa tarafından teknik bir rapor olarak literatüre kazandırılmıştır[22]. Bu raporda doğada bulunan gerçek arılar hakkında kısa bir bilgi verildikten sonra arıların besin arama davranışlarından esinlenilerek elde edilmiş olan model ve parametreleri tanıtılmıştır. 2006 yılında YAKA ile ilgili ilk konferans Bahriye Baştürk ile Derviş Karaboğa tarafından yapılmıştır[23]. Yayımlanan ilk makale ise 2007 yılında aynı yazarlar tarafından yazılmıştır[24]. Bu makalede YAKA'nın etkin ve güçlülüğü gösterilmiş ve diğer algoritmalarla karşılaştırılmalarına yer verilmiştir. Yine 2007 yılında yayımlanan bir makalede ise YAKA'nın kısıtlı optimizasyon problemlerinde de etkili bir algoritma olduğu gösterilmiştir[25]. YAKA'nın performansının değerlendirildiği bir diğer makale ise 2008 yılında yayımlanmıştır[26]. Bu çalışmada çok boyutlu optimizasyon problemleriyle ilgilenilmiş ve diferansiyel gelişim algoritması, parçacık sürü optimizasyonu ve evrimsel algoritmalar

[21] Derviş Karaboğa, a.g.e., 2011, s. 213.
[22] Derviş Karaboğa, a.g.e., 2005
[23] Bahriye Baştürk ve Derviş Karaboğa, "An artificial bee colony (abc) algorithm for numeric function optimization", **In: IEEE swarm intelligence symposium** Indianapolis,IN,USA, , 2006 s. 49-53.
[24] Derviş Karaboğa ve Bahriye Baştürk, "A Powerful and Efficent Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm", **J. Glob. Optim**., 39, 459-471, 2007a.
[25] Derviş Karaboğa ve Bahriye Baştürk, "Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems", **Proceedings of the 12th International Fuzzy Systems Association World Congress on Foundations of Fuzzy Logic and Soft computing**, Springer, Berlin, 789–798, 2007b.
[26] Derviş Karaboğa ve Bahriye Baştürk, "On the Performance of Artificial Bee Colony (ABC) Algorithm", **Applied Soft Computing**, 8, 687-697, 2008.

kullanılmıştır. Ayrıca çeşitli programlama kodlarıyla yazılmış YAKA, Erciyes Üniversite'nin oluşturduğu bir web sitesinde[*] yayımlanmaktadır. Bu sitede YAKA ile ilgili tüm bilgilere erişilebilmektedir.

YAKA'nın kullanım sayısı, 2009 yılından sonra artmaya başlamıştır. Bu artışın sebeplerinden biri basit ve kolay uygulanabilir bir algoritma olmasıdır. Birçok alanda uygulamalarına rastlamak mümkündür. Genel olarak yapay sinir ağlarının eğitilmesi, veri madenciliği, kablosuz sensör ağları, resim işleme gibi ayrık ve kombinatoryal optimizasyon problemlerinde kullanılmıştır. Bu örnekleri genişletmek mümkündür. Derviş Karaboğa önderliğinde yapılmış bir literatür çalışması ile 2012 yılına kadarki YAKA'nın kullanıldığı makaleler bulunmuştur. Bu makalede YAKA, karşılaştırmaların yapıldığı çalışmalar, melez çalışmalar ve uygulamaların yer aldığı çalışmalar diye üç gruba ayrılmıştır[27]. Detaylı bir çalışma olduğu için yapacağımız literatür çalışmasına 2013 yılından devam etmeye karar verilmiştir.

Alvarado-Iniesta ve diğerleri, imalat sektöründe hammaddenin farklı üretim hatlarına dağıtımı konusunda bir çalışma yapmışlardır. Araç Rotalama problemine benzeyen optimizasyon probleminin çözümünde Yapay Arı Kolonisi algoritmasını kullanmışlar ve uygun sonuçlar bulmuşlardır[28]. Baştürk ve Akay, kaba tanecikli paralel model yapısındaki problemleri, yapay arı kolonisi ile çözmüşlerdir. Hesaplama hızı ve performansı hakkında detaylı bilgi vermişlerdir. YAK, evrimsel algoritmalardan daha etkin ve kaliteli sonuçlar ürettiğini göstermişlerdir[29]. Alizadegan ve diğerleri, yapay arı koloni algoritmasını modifiye ederek iki ayrı versiyonunu oluşturmuşlardır. Bu modellerin yapay arı kolonisi algoritmasına olumlu etkisinin olduğunu göstermişlerdir[30]. Kang ve diğerleri, küresel optimizasyon için YAKA ile Hooke-Jeeve örüntü arama algoritmasını birleştirerek yeni bir melez algoritma geliştirmişlerdir. Önerdikleri algoritma, yapay arı kolonisi tarafından

[*] http://mf.erciyes.edu.tr/abc/

[27] Derviş Karaboğa ve dğr., a.g.e., 2012

[28] Alejandro Alvarado-Iniesta ve dğr., "Optimization of the material flow in a manufacturing plant by use of artificiai bee colony algorithm", **Expert Systems with Applications**, 40, 4785-4790, 2013.

[29] Alper Baştürk ve Rüştü Akay, "Performance analysis of the coarse-grained parallel model of the artificial bee colony algorithm", **Information Sciences**,253, 34-55, 2013.

[30] Amir Alizadegan ve dğr., "Two Modified versions of artificial bee colony algorithm", **Applied Mathematics and Computation**, 225, 601-609, 2013.

yapılan keşif aşaması ile örüntü arama tarafından tamamlanan kullanım aşamasından oluşmaktadır. Yapılan testler sonucunda yakınsama hızı, sonuçların etkinliği ve başarı oranının tatmin edici olduğunu göstermişlerdir[31]. Kalaycı ve Gupta, sıraya dayalı demontaj hattı dengeleme probleminin çözümünde YAKA kullanmışlardır. Yapılan testler sonucunda YAKA'yı, diğer altı teknik ile karşılaştırmışlardır. Sonuç olarak YAKA'nın performansının iyi olduğunu ve amaç fonksiyonu değerine göre diğer tekniklerden üstün olduğunu göstermişlerdir[32]. Nikolić ve Teodorović, arı kolonisi optimizasyonu algoritmasını literatürde bulunan tüm sayısal test fonksiyonlarına uygulamışlardır. Yapılan testler sonucunda AKOA'nın diğer tekniklerle rekabet edecek düzeyde sonuçlar verdiğini ve CPU zamanı ihmal edilirse kaliteli sonuçlar verdiğini göstermişlerdir[33]. Akay, görüntü işlemenin bölümlendirme aşamasında optimal yüksek seviyeli eşik değerlerinin bulunmasında PSO ve YAKA'yı kullanmıştır. İki eşik değerine sahip durumlarda PSO ve YAKA aynı performansı gösterirken daha fazla sayıdaki eşik değerleri durumunda ise YAKA, diğer tekniklerden daha iyi performans göstermiştir[34]. Xiang ve An, global yakınsamayı geliştirmek için var olan YAKA'nı geliştirmiştir. Lokal minimuma takılmamak için kaşif arı aşamasında dağınık arama yöntemi kullanmıştır. Ayrıca global yakınsamayı iyileştirmek için popülasyon üretiminde dağınık ilklendirme yapmıştır. Önerdikleri algoritmayı literatürde bulunan 23 veri seti üzerinde test etmişler. Önerilen algoritmanın, iki YAK tabanlı algoritmaya göre çok iyi sonuçlar ürettiğini göstermişlerdir[35]. Yao ve diğerleri, periyodik araç rotalama probleminin çözümünde YAKA'nı kullanmışlardır. Algoritmanın performansını arttırmak için arama strateji üzerinde çok boyutlu sezgisel bilgiyi ve yerel optimizasyonu kullanmışlardır. Önerilen model, bilinen veri setleri ile test edilmiş ve güçlü bir

[31] Fei Kang ve dğr., "Artificial bee colony algorithm and pattern search hybridized for global optimization", **Applied Soft Computing**, 13, 1781-1791, 2013.
[32] Can B. Kalaycı ve Surendra M. Gupta, "Artificial bee colony algorithm for solving sequence-dependent disassembly line balancing problem", **Expert Systems with Applications**, 40, 7231-7241, 2013.
[33] Miloš Nikolić ve Dušan Teodorović, "Empirical study of the Bee Colony Optimization (BCO) algorithm", **Expert Systems with Applications**, 40, 4609-4620, 2013.
[34] Bahriye Akay, "A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding", Applied Soft Computing, 13, 3066-3091, 2013.
[35] Wan-li Xiang ve Mei-qing An, "An efficient and robust artifical bee colony algorithm for numerical optimization", **Computers&Operation Research**, 40, 1256-1265, 2013.

model olduğu gösterilmiştir[36].Zhang ve diğerleri, iş çizelgeleme probleminde YAKA'nı kullanmışlardır. Bu çalışmada toplam üretim süresi yerine ağırlıklandırılmış işe geç kalma kriterini minimize etmeyi çalışmışlardır. YAKA'nın işleme yeteneğini geliştirmek için ağaç arama algoritmasını kullanmışlardır. Yapılan testler sonucunda etkili çözümler elde edilmiştir[37].

Liu, YAKA'nın yavaş yakınmasını düzeltmek için dağınık tersine öğrenme stratejisini kullanmıştır. Yapılan test sonucunda bu iyileştirme ile yakınması oranında artış gözlenmiş ve lokal minimumdan daha hızlı kurtulduğunu söylemiştir[38]. Imaniam ve diğerleri, YAKA'yı geliştirerek hız vektörünü içeren YAK algoritmasını önermişlerdir. Bu algoritma PSO' un kullandığı arama strateji olan hız vektörünü, gözcü arı aşamasında kullanmaktadır. Böylece yakınsama problemi olan YAKA'nın daha hızlı yakınsamasını sağlamışlardır. Önerilen algoritma, PSO ve YAKA'dan daha iyi sonuçlar ürettiğini göstermişlerdir[39]. Saif ve diğerleri, sıralama ve montaj hattı dengeleme işinin aynı anda yapan çok amaçlı YAKA algoritmasını önermişlerdir. Çalışmalarının amacının, her istasyondaki farklı modellerin iş yükünü dengelemek, bir istasyonun ortalama iş gücünün bütün istasyonların ortalama iş gücünden sapmasını azaltmak ve farklı istasyonlardaki modellerin toplam akış zamanını azaltmak olduğunu söylemişlerdir. Modeli, literatürde bilinen ünlü algoritmalarla karşılaştırmışlardır. Önerilen model, NSGA II algoritmasına göre daha iyi pareto optimal sonuçlar verdiğini göstermişlerdir[40]. Chun-Feng ve diğerleri, yeni bir YAK algoritması önermişlerdir. Bu algoritma, ilk popülasyonun oluşturulmasında rastgele üretmektense nokta küme teorisini kullanarak yakınsama hızını iyileştirmişlerdir. İkinci olarak ise işçi arı, gözcü arı ve kaşif arı aşamalarında PSO arama mekanizmalarını kullanarak keşfetme yeteneğini geliştirmişler. Önerilen

[36] Baozhen Yao ve dğr., "Artificial bee colony algorithm with scanning strateji for the periodic vehicle routing problem", **Simulation: Transactions of the Society for Modeling and Simulation International**, 89(6), 762-770, 2013.
[37] Rui Zhang ve dğr., "A hybrid artificial bee colony algorithm for the job shop scheduling problem", **Int. J. Production Economics**, 141, 167-178, 2013.
[38] Wen Liu, "A multistrategy optimization improved artificial bee colony algorithm", **The Scientific World Journal**, vol. 2014, Article ID 129483, 10 p., http://dx.doi.org/10.1155/2014/129483, 2014.
[39] Nafiseh Imanian ve dğr., "Velocity based bee colony algorithm for high dimensional continuous optimization problems", **Engineering Applications of Artificial Intelligence**, 36, 148-163, 2014.
[40] Ullah Saif ve dğr., "Multi-objective artificial bee colony algorithm for simultaneous Sequencing and balancing of mixed model assembly line", **Int. J. Manuf. Technol.**, 75, 1809-1827, 2014.

modeli, literatürde bilinen problem türleri ile test etmişler ve birçok problem için daha iyi sonuçlar verdiğini göstermişlerdir[41]. Chen, işlem maliyetini içeren belirsiz portföy seçim problemine geliştirdiği YAKA ile çözüm bulmuştur. Modelin çözümü için sayısal örnekler vermiştir ve algoritmanın etkin olduğundan bahsetmektedir[42]. Chen ve diğerleri, işlem maliyetini içeren portföy atama problemi üzerine çalışma yapmışlardır. Ortamda var olan belirsizliği bulanık sayılar kullanarak belirgin hale getirmeye çalışmışlardır. Önerdikleri modelin çözümünde geliştirdikleri YAKA kullanmışlardır. Sayısal örnekler vererek modelin etkinliğini göstermişlerdir[43]. Zhang ve diğerleri, araç rotalama problemini çevresel araç rotalama problemi olarak düzenlemişlerdir. Bu modeli de melez YAKA ile çözmüşlerdir. Alınan sonuçlar çok bilenen KARP örnekleri ile kıyaslanmıştır. Melez YAKA, orijinal YAKA'dan % 5 daha iyi sonuçlar vermiştir. Bu çalışma, yeşil lojistikle ilgilenenler için pratik bir bakış açısı sağladığını vurgulamaktadırlar[44]. Xiang ve diğerleri, yeni bir melez evrimsel algoritma geliştirmişlerdir. Bu algoritmayı YAK ile DG algoritmalarının birleşiminden oluşturmuşlardır. YAKA'nın yeni adaylar seçimindeki zayıflığını global en iyi tabanlı YAKA'dan esinlenerek geliştirdikleri yeni bir YAKA ile çözmüşlerdir. DG algoritması ile önerdikleri yeni evrimsel algoritmaya hızlı yakınsama özelliği eklemişlerdir. Önerilen modeli, 20 tane bilenen veri setleri ile test etmişlerdir. Aldıkları sonuçlara göre geliştirilen algoritma, YAKA, DG ve DG algoritmasının diğer versiyonlarından daha iyi sonuç verdiğini göstermişlerdir[45]. Cura, araç rotalama probleminin bir türü olan zaman pencereli takım oryantiring problemini geliştirdiği YAKA ile çözmüştür. Böylece zor kesikli optimizasyon problemlerinden birine YAKA ile çözüldüğünü vurgulamaktadır. Önerdiği algoritmada yeni besin kaynağı seçim mekanizması ve yeni kaşif arı arama davranışı

[41] Wang Chun-Feng ve dğr., "Hybrid Artificial Bee Colony Algorithm and Particle Swarm Search for Global Optimization", **Mathematical Problem in Engineering**, Vol. 2014, Article ID: 832949, 8 pages, 2014.
[42] Wei Chen, "An Artificial Bee Colony Algorithm for Uncertain Portfolio Selection", **The Scientific World Journal**, Volume 2014, Article ID 578182, 12 pages, 2014.
[43] Wei Chen, Hui Ma, Yiping Yang, Mengrong Sun, " Application of Artificial Bee Colony Algorithm to Portfolio Adjustment Problem with Transaction Costs", **Journal of Applied Mathematics**, Volume 2014, Article ID 192868, 12 pages, 2014.
[44] Shuzhu Zhang ve dğr., "Design and development of a hybrid artificial bee colony algorithm for the enviromental vehicle routing problem", **Transportation Research Part D**, 31, 85-99, 2014.
[45] Wanli Xiang ve dğr., "hABCDE: A hybrid evolutionary algorithm based on artificial bee colony algorithm and differential evolution", **Applied Mathematics and Computation**, 238, 370-386, 2014.

olduğunu belirtmektedir. Alınan sonuçlar doğrultusunda diğer yöntemlere göre daha etkin, başarılı ve kıyaslanabilir bir yöntem olduğunu söylemektedir[46]. Karaboğa ve Görkemli, YAKA'nın yeni versiyonu olan hızlı yapay arı kolonisi algoritmasını geliştirmişlerdir. Bu algoritmanın hem gözcü arıların daha kesin ve doğru sonuçlar verdiğini hem de orijinal YAKA'nın yerel arama yeteneği açısından performansını arttırdığından bahsetmişlerdir. Bu çalışma ile yeni modelin tanıtımını yapmışlar ve modeli literatürde bulunan veri setlerini kullanarak komşuluk etkin alanına göre test etmişlerdir. Ayrıca modeli farklı parametre değerleri ile de test etmişlerdir. Sonuç olarak etkin bir yeni YAK versiyonunu geliştirdiklerini belirtmektedirler[47]. Milan ve Nebojsa, kardinalite kısıtlı ortalama varyans portföy seçim probleminin çözümünde YAKA kullanmışlardır. YAKA'nın optimal noktaya yavaş yakınsama probleminin çözümüne ateş böceği algoritması ile çözüm bulmuşlardır. Hibrid yeni bir model ortaya koyan yazarlar, özellikle ortalama öklid uzaklığı açısından diğer meta sezgisel yöntemlerden daha iyi sonuç bulduklarını söylemişlerdir[48].

[46] Tunçhan Cura, " An artificial bee colony algorithm approach for the team orienteering problem with time Windows", **Computers&Industrial Engineering**, 74, 270-290, 2014.
[47] Derviş Karaboğa ve Beyza Görkemli, " A quick artificial bee colony (qABC) algorithm and its performance on optimization problems", **Applied Soft Computing**, 23, 227-238, 2014.
[48] Tuba Milan ve Bacanin Nebojsa, "Artificial Bee Colony Algorithm Hybridized with Firefly Algorithm for Cardinality Constrained Mean-Variance Portfolio Selection Problem", **Applied Mathematics&Information Sciences**, 8(6), 2831-2844, 2014.

# 3. PORTFÖY OPTİMİZASYONU

Günümüzde finansal piyasalar, dünyanın herhangi bir yerinden, ellerindeki parayı etkin bir şekilde yönetip arttırmayı hedefleyen milyonlarca kişi tarafından takip edilmektedir. Bu piyasada, yapılan yatırımlar doğrultusunda her bir yatırım enstrümanının fiyatları yeniden hesaplanmaktadır. Bu açıdan da oldukça dinamik bir yapıya sahiptir. Büyük kazançların elde edilebiliyor olması bu piyasayı cazip haline getirmektedir. İyi kazançlar elde etmenin yanında yanlış stratejinin uygulanması ve yeterli bilginin elde edilememesinden dolayı büyük zararlara da sebep olmaktadır. Bu ikilemin olması bilim adamlarını "Nasıl bir yatırım kararı verelim ki yatırımcı için en iyi karar olsun?" sorusunun cevabını aramaya sevk etmiştir.

Milyonlarda kişinin, binlerce yatırım enstrümanı arasından kendi çıkarlarına en uygun yatırım kararını verebilmesi kolay bir iş değildir. Bu durum yeni bir karar verme probleminin ortaya çıkmasına neden olmuştur. Portföy Optimizasyonu (PO) adı verilen bu problemde amaç, yatırımcıya en uygun yatırım enstrümanı birleşimini sunmaktır. Portföy seçimi olarak da bilinen bu konu literatürde oldukça geniş yer tutmaktadır.

## 3.1. Yatırım ve Portföy Kavramı

Yatırım, ileriki dönemlerde gelir elde etmek ümidiyle belirli bir paranın çeşitli yatırım araçlarına belirli bir süre için bağlanması olarak tanımlanmaktadır[1]. Yatırımda esas olan karın elde edilebilmesidir. Bu yüzden yatırımcıların kar elde edebilecekleri her varlık bir yatırım aracı olabilir. Bu reel olarak ev, araba, arsa...vb almak olabileceği gibi finansal varlıklar olan hisse senedi, tahvil, bono…vb almak da olabilir.

[1] Serpil Canbaş ve Hatice Doğukanlı, **Finansal Pazarlar, Finansal Kurumlar ve Sermaye Pazarı Analizleri**, Genişletilmiş 3. Baskı, Beta Yayıncılık, 2001, s.286.

Portföy ise, genellikle hisse senedi, tahvil gibi finansal varlıklardan meydana gelen ve belirli bir kişinin ya da grubun sahip olduğu finansal nitelikteki kıymetlerdir[2].

Portföydeki her bir finansal varlık birbirleriyle ilişkilidir. Bu açıdan bakıldığında çeşitli finansal varlıkların birleşimi gibi gözükse de bu varlıklardan ayrı kendine öz ve ölçülebilir bir yapıya sahiptir. Daha geniş bir ifadeyle portföy, çeşitli amaçlarını gerçekleştirmek isteyen yatırımcıların sahip oldukları, birbiriyle ilişkileri olan finansal varlıkların oluşturduğu, kendine öz ve ölçülebilir yeni bir varlıktır[3].

Yatırım sürecinde iki temel kavram önemli rol oynamaktadır. Bunlar getiri ve risktir. Getiri, ileriki dönemde yatırım araçlarından elde edilecek kazançları ifade etmektedir. Gelecek dönemde finansal varlıkların ne oranda kazanç getireceği bilinmemektedir. Yatırımlar finansal varlıkların geçmiş verilerinden hareketle elde edilen beklenen getiriler ile yapılmaktadır. Yatırımcının beklediği getiri gerçekleşen getiriden fazla olmaktadır. Bu durum bir belirsizlik yaratmaktadır. Oluşan bu belirsizliğe risk adı verilmektedir. Bu iki kavram detaylı olarak Bölüm 3.2. 'de açıklanmaktadır.

## 3.2. Getiri ve Risk

Getiri, yatırımcının, bir yatırımdan belirli dönem içinde yaptığı yatırıma karşılık elde ettiği gelir olarak ifade edilmektedir[4]. Yatırımcı bir dönem içinde, yaptığı yatırımdan iki tür gelir elde etmektedir. Bunlardan birincisi faiz veya kar payı, diğeri ise finansal varlığın dönem sonu ile dönem başı fiyatı arasındaki olumlu veya olumsuz fark yani sermaye kazancı veya zararıdır.

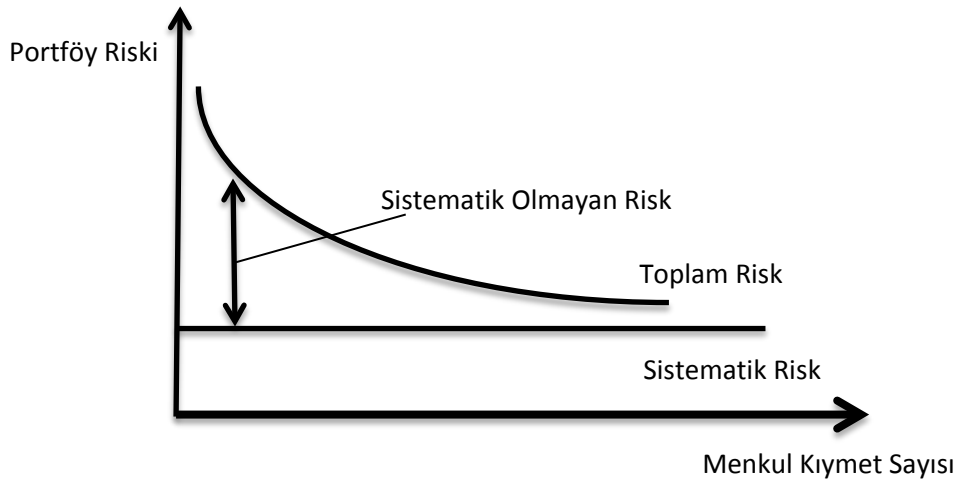
---

[2] Ali Ceylan ve Turhan Korkmaz, **Sermaye Piyasası ve Menkul Değer Analizi**, Ekin Kitabevi, Bursa, 2000, s. 257.
[3] Ali Ceylan ve Turhan Korkmaz, **Borsada Uygulamalı Portföy Yönetimi**, 3. Baskı, Ekin Kitabevi, Bursa, 1998, s. 7-8.
[4] Mehmet B. Karan, **Yatırım Analizi ve Portföy Yönetimi**, 3. Baskı, Gazi Kitabevi, Ankara, 2011, s. 139.

Getirinin, beklenen düzeyden farklı çıkmasına risk denilmektedir. Yani risk, yatırımdan sağlanacak getirinin tam olarak bilinememesinden kaynaklanmaktadır[5]. Her yatırımın bir riski mevcuttur. Yapılan yatırımın riski yatırımın boyutuna ve hangi varlıklara yatırım yapıldığına bağlıdır. Eğer risksiz bir yatırım istenirse devlet tahvillerine ve hazine bonolarına yatırım yapılabilir.

Yatırımcılar, minimum riskle maksimum kazanç elde etmeyi ümit ettikleri portföy kümesini bulmayı amaçlamaktadırlar. Bunun içinde finansal varlıkların getiri ve riskleri hakkında bilgiye sahip olmaları gerekir. Bir finansal varlığın getirisi, dönemsel getirilerinden ve sermaye kazançlarından oluşmaktadır ve kolaylıkla hesaplanabilir. Risk ise, her biri farklı bir şekilde tanımlanıp yorumlanan toplam riskin ölçülmesiyle belirlenir. Toplam risk, sistematik ve sistematik olmayan risk diye ikiye ayrılmaktadır (Şekil 3.1.)[6].

**Şekil 3.1. Risk bileşenleri**




**Kaynak:** Ali Ceylan ve Turhan Korkmaz, a.g.e., 2000, s.265.

Sistematik risk, bir finansal varlığın getirilerinin varyansı ile ölçülür. Sistematik olmayan risk ise, finansal varlığa özgü olaylar ile ortaya çıkan sadece o varlığa ait riski ifade eder. Bu risk genellikle birçok finansal varlığı içeren

[5] Serpil Canbaş ve Hatice Doğukanlı, a.g.e., 2001, s.287.
[6] Serpil Canbaş ve Hatice Doğukanlı, a.g.e., 2001, s.288.

çeşitlendirilmiş portföylerde ortadan kalkabilir. Bu yüzden finansal pazarlarda değerlendirilmeye alınmaz.

Sistematik risk, tüm finansal piyasaları ve bu piyasalarda işlem gören tüm menkul kıymetleri etkileyen, politik ve sosyal yaşamdaki değişimlerden kaynaklanan risk türüdür. Sistematik risk kaynakları faiz oranı riski, satın alma gücü (enflasyon) riski, piyasa riski, politik risk, kur riski diye sınıflandırılabilir (Şekil 3.2.).

*Faiz oranı riski*, piyasadaki genel faiz oranlarının yükselme olasılığını ifade eder. Faiz oranlarındaki değişimler finansal varlıkların fiyatlarını etkilemektedir.

*Satın alma gücü (enflansyon) riski*, yani paranın satın alma gücündeki azalış piyasada işlem gören varlıkların fiyatlarını farklı oranlarda etkilemektedir.

*Piyasa riski,* kabul edilebilir ekonomik nedene dayanmayan ve çoğunlukla psikolojik nedenlere bağlı olarak ortaya çıkan, menkul kıymet fiyatlarındaki düşüşler sebebiyle oluşur[7].

Sistematik olmayan risk, ilgili işletme veya işletmenin faaliyette bulunduğu iş koluna ait özelliklerin değişiminden kaynaklanan risk türüdür. Yönetim hataları, teknolojik gelişmeler bu risk türüne örnek olarak verilebilir. Sistematik olmayan risk, finansal risk, yönetim riski ve iş riski olarak sınıflandırılabilir (Şekil 3.2.).

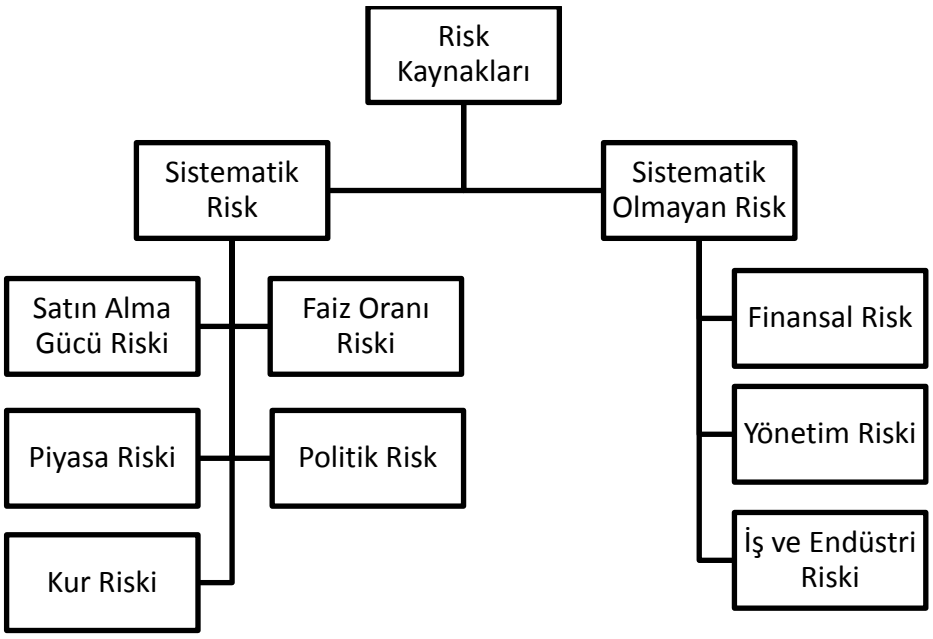
*Finansal risk,* ortaklığın sermaye yapısı içinde, var olan ortaklığa sabit yük getiren kaynakların var oluşundan kaynaklanmaktadır.

*Yönetim riski*, firma yöneticilerinin yeteneklerine bağlı olarak değişir. Yönetim hataları sonucu ortaya çıkan bir risktir.

[7] Serpil Canbaş ve Hatice Doğukanlı, a.g.e., 2001, s.293.

*İş riski*, dış etmenlere bağlı olarak oluşur. Tüketici davranışlarındaki değişim, rekabetin şiddetlenmesi, yoğun grevlerin oluşu bu riski etkileyen durumlardır[8].

**Şekil 3.2. Toplam riskin kaynakları**




**Kaynak:** Ali Ceylan ve Turhan Korkmaz, a.g.e., 2000, s.264.

## 3.2.1. Menkul Kıymetlerde Getiri

### 3.2.1.1. Dönemlik Getiri

Dönemlik getiri yatırımın belirli bir zaman içindeki toplam getirisini ifade eder. Başka bir ifadeyle o dönem içinde yatırımdaki artış miktarını gösterir.

Dönemlik getiri:

$R$ : Menkul kıymetin dönemlik getirisi

[8] Serpil Canbaş ve Hatice Doğukanlı, a.g.e., 2001, s.294.

$P_t$ : Menkul kıymetin dönem sonu fiyatı

$P_{t-1}$ : Menkul kıymetin dönem başı fiyatı

$D_t$ : Menkul kıymetin $t$. dönemde ödediği temettü (kar payı) miktarı

$$R = \frac{[(P_t - P_{t-1}) + D_t]}{P_{t-1}} \qquad (3.1.)$$

denklemi ile hesaplanır[9].

### 3.2.1.2. Beklenen Getiri

Yatırımcılar, finansal varlıkların türüne göre yaptıkları yatırımlardan kazanç elde ederler. Bu kazanç, faiz geliri, kar payı ve finansal varlığın fiyatındaki artıştan sağlanan değerlerden oluşmaktadır. Finansal varlıkların getirisi, ekonominin, sektörün ve işletmenin durumundaki değişimlerden etkilenmektedir. Bu yüzden yatırımcılar istikrarlı varlıklara yatırım yapıp risklerini azaltmak isterler. Devlet tahvilleri ve hazine bonolarından beklenen getiri risksiz faiz oranı olarak kabul edilir. Çünkü bu varlıkların sağladığı getiri kesin olarak bilinmektedir ve devlet garantisi altındadır. Bu varlıklar dışındaki menkul kıymetlere yatırım yapmak isteyen yatırımcı risksiz faiz oranının üstünde bir getiri beklemektedir. Çünkü parasının geri dönmeme riski bulunmaktadır.

Beklenen getiri, risksiz faiz oranı ve bir risk priminden meydana gelmektedir. Başka bir ifadeyle riski yüksek olan varlığın beklenen getirisi de yüksek olacaktır. Matematiksel olarak bir tanım vermek gerekirse beklenen getiri, menkul kıymete ait belirli bir dönem getirileri ile bu getirilerin gerçekleşme olasılığının çarpımının toplamı olarak ifade edilmektedir.

Belirli bir dönem içinde bir menkul kıymete ait gerçekleşmiş veriler kullanılarak elde edilen ortalama getiri,

[9] Mehmet B. Karan, a.g.e., 2011, s. 140.

$\bar{R}$ : Menkul kıymetin ortalama getirisi

$R_t$ : Menkul kıymetin t. dönemdeki getirisi

$N$ : Dönem Sayısı

$$\bar{R_i} = \sum_{t=1}^{N} \frac{R_{i,t}}{N} \qquad (3.2.)$$

denklemi ile hesaplanır.

Bir menkul kıymetin beklenen getirisi ise,

$E(R)$ : Menkul kıymetin beklenen getirisi

$P_t$ : $t$. senaryonun gerçekleşme olasılığı

$R_t$ : Menkul kıymetin $t$. senaryo durumundaki getirisi

$n$ : Senaryo Sayısı

$$E(R) = \sum_{t=1}^{n} P_t R_t \qquad (3.3.)$$

denklemi ile hesaplanır[10].

### 3.2.2. Menkul Kıymetlerde Risk

#### 3.2.2.1. Standart Sapma ve Varyans

Bir menkul kıymete yatırım kararı yalnız beklenen getirisine bakılarak verilmemelidir. Doğru bir karar için belirli bir döneme ait getirilerin bu beklenen getiriden ne kadar saptığına ilişkin bir ölçünün de belirlenmesi gerekir. Standart sapma ve varyans olarak tanımlanan bu ölçüt, menkul kıymetinde riskini temsil

[10] Ali Ceylan ve Turhan Korkmaz, a.g.e., 1998, s.80.

etmektedir. Bu değerin düşük olması ilgili varlığın riskinin düşük olduğunu belirtmektedir. Standart sapma, varyansın karekökü alınarak hesaplanır[11].

Geçmiş verilerden hesaplanmış menkul kıymete ait varyans,

$\sigma^2$ : Menkul kıymetin varyansı
$R_t$ : Menkul kıymetin $t$. dönemdekii getirisi
$\bar{R}$ : Menkul kıymetin ortalama getirisi
$N$ : Dönem sayısı

$$\sigma^2 = \sum_{t=1}^{N} \frac{(R_t - \bar{R})^2}{N} \tag{3.4.}$$

denklemi ile hesaplanmaktadır. Böylece menkul kıymete ait standart sapma,

$$\sigma = \sqrt{\sum_{t=1}^{N} \frac{(R_t - \bar{R})^2}{N}} \tag{3.5.}$$

denklemi ile hesaplanmaktadır.

Her bir senaryonun meydana gelme olasılığı farklı kabul edilirse menkul kıymete ait varyans,

$\sigma^2$ : Menkul kıymetin varyansı
$P_t$ : $t$. senaryonun gerçekleşme olasılığı
$R_t$ : Menkul kıymetin $t$. senaryo durumundaki getirisi
$E(R)$ : Menkul kıymetin beklenen getirisi
$n$ : Senaryo Sayısı

[11] Aydın Ulucan, **Portföy Optimizasyonu Kuadratik Programlama Tabanlı Modelleme**, Siyasal Kitabevi, 2004, s. 10.

$$\sigma^2 = \sum_{t=1}^{n} [P_t(R_t - E(R))^2] \quad (3.6.)$$

denklemi ile hesaplanmaktadır. Böylece menkul kıymete ait standart sapma,

$$\sigma = \sqrt{\sum_{t=1}^{n} [P_t(R_t - E(R))^2]} \quad (3.7.)$$

denklemi ile hesaplanmaktadır[12].

### 3.2.2.2. Değişim Katsayısı

Menkul kıymetin risk hesabının yapılmasında kullanılan bir diğer ölçüt değişim katsayısıdır. Bu ölçü, her birim getiri için riski ölçer. Beklenen getirisi aynı olmayan finansal varlıkların değerlendirilmesinde daha uygun bir ölçüttür. Düşük değişim katsayısı, o finansal varlığın daha az riskli olduğunu göstermektedir. Değişim katsayısı varyasyon katsayısı olarak da tanımlanmaktadır. Bir menkul kıymete ait değişim katsayısı,

$$DK = \frac{\sigma}{E(R)} \quad (3.8.)$$

denklemi ile hesaplanmaktadır[13].

[12] Ali Ceylan ve Turhan Korkmaz, a.g.e., 1998, s.83.
[13] Mehmet B. Karan, a.g.e., 2011, s. 146.

### 3.2.3. Menkul Kıymetlerin Birlikte Hareket Ölçüleri

#### 3.2.3.1. Kovaryans

İki ya da daha fazla menkul kıymetin riskinin belirlenmesinde standart sapma ve varyans kullanılmamaktadır. Bu ölçütün yerine iki menkul kıymetin belirli dönem içindeki hareketliliğinin aynı andaki uyumunu ölçen kovaryans değeri kullanılmaktadır. Bir başka ifadeyle kovaryans, menkul kıymetlerin dönem içindeki getirilerinin o menkul kıymete ait beklenen getiriden sapmalarının çarpımlarının toplamıdır[14].

Kovaryans değeri -∞ ile +∞ arasında değişen bir değerdir. Bu değerin pozitif olması iki menkul kıymetin hareketinin aynı yöne olduğu göstermektedir. Yani bir menkul kıymetin getirisi ortalama getiriden fazla ise diğer menkul kıymetinde getirisi fazladır. Aynı şekilde bir menkul kıymetin getirisi ortalama getiriden az ise diğer menkul kıymetin getirisi de azdır. Eğer bir menkul kıymetin getirisi ortalama getiriden fazla iken diğer menkul kıymetin getirisi az ise negatif kovaryans vardır. Bir başka ifadeyle iki menkul kıymetin hareketi zıt yönlüdür. Kovaryans katsayısının sıfır veya sıfıra yakın olması iki varlık arasında doğrusal bir ilişkinin olmadığını gösterir[15].

Geçmiş veriler kullanılarak hesaplanan kovaryans değeri,

$\sigma_{i,j}$ : $i$ ve $j$ menkul kıymetleri arasındaki kovaryans değeri

$R_{i,t}$ : $i$. menkul kıymetin $t$. dönemdeki getirisi

$\overline{R_i}$ : $i$. menkul kıymetin ortalama getirisi

$R_{j,t}$ : $j$. menkul kıymetin $t$. dönemdeki getirisi

$\overline{R_j}$ : $j$. menkul kıymetin ortalama getirisi

[14] Mehmet B. Karan, a.g.e., 2011, s.148.
[15] Ali Ceylan ve Turhan Korkmaz, a.g.e., 1998, s.84.

$N$ : Dönem Sayısı

$$\sigma_{i,j} = \frac{\sum_{t=1}^{N}[(R_{i,t} - \bar{R}_i)(R_{j,t} - \bar{R}_j)]}{N-1} \quad (3.9.)$$

denklemi ile hesaplanmaktadır.

Menkul kıymetlere ait senaryoların gerçekleşme olasılıklarının farklı olduğu durumlarda kovaryans değeri,

$\sigma_{i,j}$ : $i$ ve $j$ menkul kıymetleri arasındaki kovaryans değeri
$P_t$ : $t$. senaryonun gerçekleşme olasılığı
$R_{i,t}$ : $i$. menkul kıymetin $t$. senaryo durumundaki getirisi
$E(R_i)$ : $i$. menkul kıymetin beklenen getirisi
$R_{j,t}$ : $j$. menkul kıymetin $t$. senaryo durumundaki getirisi
$E(R_j)$ : $j$. menkul kıymetin beklenen getirisi
$n$ : Senaryo Sayısı

$$\sigma_{i,j} = \sum_{t=1}^{n} P_t[(R_{i,t} - E(R_i))(R_{j,t} - E(R_j))] \quad (3.10)$$

denklemi ile hesaplanmaktadır[16].

### 3.2.3.2. Korelasyon Katsayısı

Kovaryans değerini, menkul kıymetler arasında negatif ve pozitif ilişki olup olmadığını değerlendirmek dışında yorumlamak oldukça güçtür. Çünkü kovaryans değerinin büyüklüğünü açıklamak mümkün değildir. Bu sebeple daha anlamlı yorum yapabilmek için kovaryans değeri yerine korelasyon katsayısı kullanılmaktadır.

[16] Ali Ceylan ve Turhan Korkmaz, a.g.e., 1998, s.84-85.

Korelasyon katsayısı, iki menkul kıymet arasındaki ilişkinin derecesini göstermektedir. Bir başka deyişle iki menkul kıymetin hareketlerindeki değişimin uygunluk derecesini belirler.

Korelasyon katsayısı, -1 ile +1 arasında değerler almaktadır. Bu katsayının 1'e doğru yaklaşması iki menkul kıymet arasındaki ilişkinin güçlü olduğunu göstermektedir. Eğer 0'a yakın bir değerse bu varlık arasında ilişki yoktur denilebilmektedir. Bu katsayıdaki işaretler ise ilişkinin yönünü göstermektedir. Eğer pozitif ise aynı yönlü negatif ise ters yönlü bir ilişki vardır.

Korelasyon katsayısı, iki menkul kıymete ait kovaryans değerinin, ilgili menkul kıymetlerin standart sapmalarına bölünmesiyle bulunmaktadır. Matematiksel olarak korelasyon katsayısı,

$\rho_{i,j}$ : $i$. menkul kıymet ile $j$. menkul kıymet arasındaki korelasyon katsayısı
$\sigma_{i,j}$ : $i$. menkul kıymet ile $j$. menkul kıymet arasındaki kovaryans değeri
$\sigma_i$ : i. menkul kıymetin standart sapması

$$\rho_{i,j} = \frac{\sigma_{i,j}}{\sigma_i \sigma_j} \qquad (3.11.)$$

denklemi yardımıyla hesaplanır[17].

### 3.2.4. Portföyün Beklenen Getirisi

Bir portföyün beklenen getirisi, portföye dahil olan menkul kıymetlerin beklenen getirilerinin ağırlıklı ortalamasıdır. Buna göre $M$ adet menkul kıymet içeren bir portföyün beklenen getirisi,

$\mu$ : Portföyün beklenen getirisi

[17] Mehmet B. Karan, a.g.e., 2011, s.148-149.

$w_i$ : $i$. menkul kıymetin portföydeki ağırlığı

$E(R_i)$ : $i$. menkul kıymetin beklenen getiri oranı

$M$ : Menkul kıymet sayısı

$$\mu = \sum_{i=1}^{M} w_i E(R_i) \qquad (3.12.)$$

denklemi ile hesaplanır.

### 3.2.5. Portföyün Riski

Portföyün riski, menkul kıymetlerin risklerinin ağırlıklı ortalaması değildir. Çünkü menkul kıymetlerin ayrı ayrı birbirini ortadan kaldırmaktadır. Böylece portföyün riski daha da küçük değer olmaktadır. Portföy içinde birden fazla menkul kıymet bulunduğu için portföyün risk hesabı, bir önceki bölümde açıkladığımız kovaryans ve korelasyon değerleri yardımıyla yapılmaktadır. Portföyün riski, portföyün standart sapması ile ölçülmektedir[18]. Buna göre $M$ adet menkul kıymet içeren bir portföyün varyansı:

$\sigma_P^2$ : Portföyün varyansı

$w_i$ : $i$. menkul kıymetin portföydeki ağırlığı

$\sigma_{i,j}$ : $i$ ve $j$ menkul kıymetleri arasındaki kovaryans değeri

$$\sigma_P^2 = \sum_{i=1}^{M} \sum_{j=1}^{M} w_i w_j \sigma_{i,j} \qquad (3.13.)$$

denklemi ile hesaplanmaktadır[19].

[18] Ali Ceylan ve Turhan Korkmaz, a.g.e., 2000, s.260.
[19] Aydın Ulucan, a.g.e., 2004, s.15.

Korelasyon katsayısı kullanılarak portföyün varyansı,

$\sigma_P^2$ : Portföyün varyansı

$w_i$ : $i$. menkul kıymetin portföydeki ağırlığı

$\rho_{i,j}$ : $i$. menkul kıymet ile $j$. menkul kıymet arasındaki korelasyon katsayısı

$\sigma_i$ : i. menkul kıymetin standart sapması

$$\sigma_P^2 = \sum_{i=1}^{M}\sum_{j=1}^{M} w_i w_j \sigma_i \sigma_j \rho_{i,j} \qquad (3.14.)$$

denklemi ile hesaplanmaktadır.

## 3.3. Portföy Yönetimi ve Portföy Yönetimi Yaklaşımları

Ekonomik koşulların zaman içinde değişmesi portföyü oluşturan finansal varlıkların tekrar değerlendirilmesine sebep olmaktadır. Böylece hangi varlıkların ne zaman portföyden çıkartılıp satılacağına, bu varlıkların yerine hangi finansal varlıkların ne zaman getirileceğine karar verilir. Portföy yönetimi adı verilen bu işte uzmanlık esastır. Genellikle bankalar, aracı kurumlar ve uzman kuruluş tarafından yapılan bu iş belirli bir politika izlenir. Yatırım politikası, portföyden sağlanacak gelir ile portföyün riski arasındaki dengeyle alakalıdır. Bu politikaya portföy yöneticisi karar verir. Portföy yöneticisi, portföy sahibi olabileceği gibi uzman bir kişi de olabilir. Portföy oluşturmada ve yönetme de en önemli husus yatırım yapılacak finansal varlığın iyi tanınmasıdır[20].

Portföy yönetiminde temel amaç, oluşturulacak bir portföyde hangi finansal varlığın hangi oranda yer alacağını belirleyebilmektir[21]. Bir başka ifadeyle, yatırımcıların ihtiyaçlarına göre portföye finansal varlıklar almak ve yatırımın

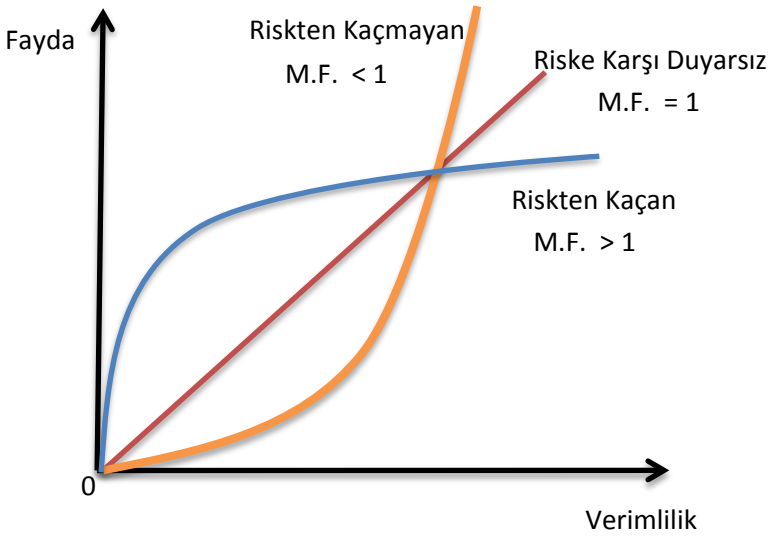
[20] Muharrem Karslı, **Sermaye Piyasası, Borsa, Menkul Kıymetler**, 4. Baskı, İrfan Yayımcılık, İstanbul, 1989, s.513-514.

[21] Mehmet Bolak, **Sermaye Piyasası: Menkul Kıymetler ve Portföy Analizi**, 4. Baskı, Beta Yayıncılık, 2001, s.247.

amacına uygun portföyü yönetebilmektir. Bu yönetimin iyi olabilmesi için yatırımcı veya portföy sahibi, portföy yöneticisiyle diyalog halinde olmalıdır. Portföy sahiplerinin nasıl bir risk ya da getiri beklediklerini portföy yöneticine bildirmeleri gereklidir. Böylece portföy yöneticisi, alınan riske uygun olarak yatırımcının karını maksimum yapmaya çalışmaktadır[22].

Portföy yönetimi açısından yatırımcı tiplerinin bilinmesi önemlidir. Risk karşısında riskten kaçan yatırımcı, riske karşı duyarsız yatırımcı ve riski seven yatırımcı olmak üzere üç tip yatırımcı vardır (Şekil 3.3.).

**Şekil 3.3. Risk karşısında yatırımcı tipleri**




**Kaynak:** Ali Ceylan ve Turhan Korkmaz, a.g.e., 2000, s.258.

Riskten kaçan yatırımcı, riski sevmez ve korkar. Bu nedenle getirileri belli olan yatırımlardan riski az olanı tercih eder. Bu yatırımcı türü için paranın marjinal faydası negatiftir. Marjinal fayda, ilave her bir birimin toplam faydaya yaptığı katkı olarak ifade edilir. Eğer yatırımcının serveti artarken riskli yatırımlara yaptığı yatırım miktarı azalıyorsa bu riskten kaçan bir yatırımcıdır.

[22] Ali Ceylan ve Turhan Korkmaz, a.g.e., 1998, s.8.

Riske karşı duyarsız yatırımcı, riskle ilgilenmez. Onlar için nereye yatırım yapıldığı önemli değildir. Bu sebeple yatırımın getiri ve riski arasında kayıtsızdırlar. Bu yatırımcılar için paranın marjinal faydası 1'e eşittir. Eğer yatırımcının serveti artarken riskli finansal varlıklara yaptığı yatırım miktarı değişmiyorsa bu yatırımcı riske duyarsız yatırımcıdır.

Riski seven yatırımcılar için yapılan yatırımın beklenen faydası, yaptırım yapmamanın beklenen faydasından büyüktür. Bu tarz yatırımcının, her ek ünitede kazanacağı verimliliğin sağladığı fayda giderek artmaktadır. Bir başka ifadeyle marjinal faydası 1'den büyüktür. Eğer yatırımcı servetini arttırırken riskli finansal varlıklara yaptığı yatırım miktarı da artıyorsa bu yatırımcı riski seven bir yatırımcıdır[23].

Dinamik bir süreç olan portföy yönetimi süreci, portföy planlaması, yatırım analizi, portföy seçimi, portföy değerlendirmesi ve portföy revizyonu olmak üzere 5 aşamadan oluşmaktadır. Söz konusu aşamalar Şekil 3.4.'de gösterilmektedir.

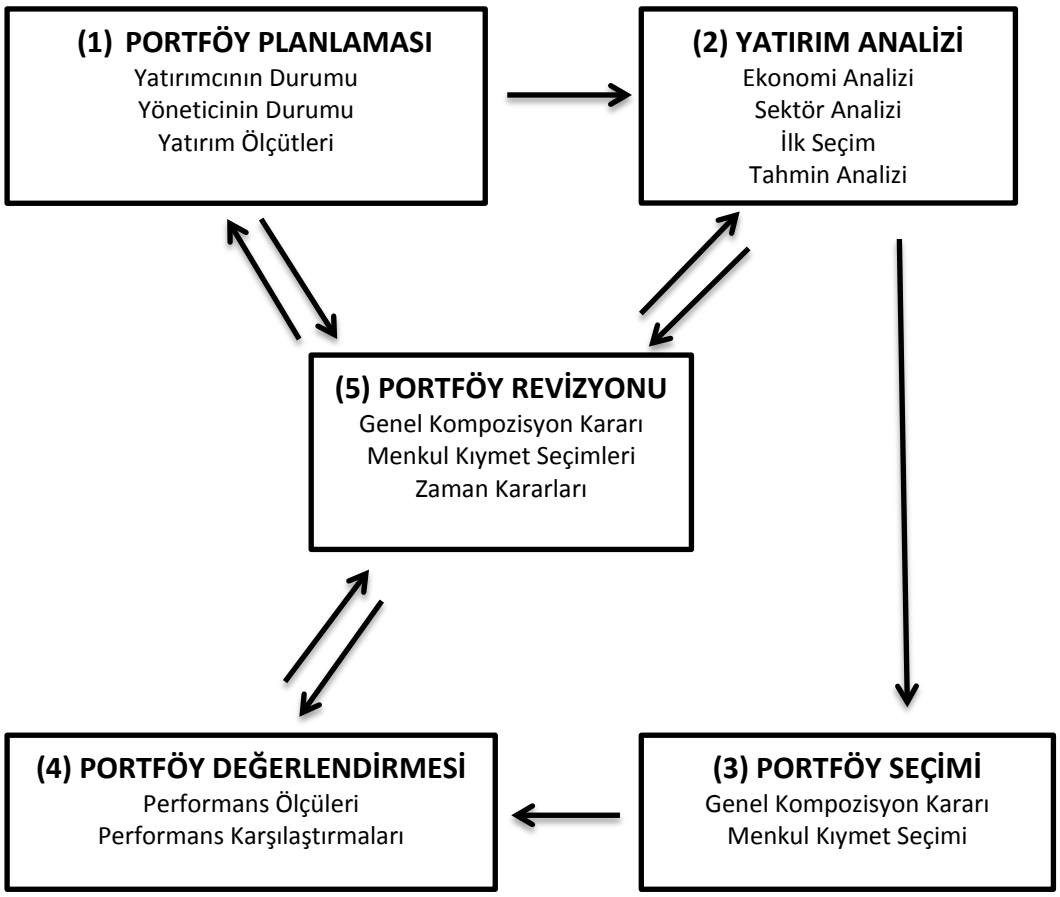
Portföy planlaması, yatırımcının, portföy yöneticisinin durumlarının incelendiği ve portföy yöneticisine yol gösterecek olan yatırım ölçütlerinin belirlenmesini içeren ilk aşamadır.

Yatırım analizi, portföye alınacak finansal varlıkların niceliklerinin incelenmesi, ölçülmesi, belirli bir dönem içinde değişik finansal varlıkların performanslarının ne olabileceğinin nicel olarak tahmin edilmesidir.

Portföy seçimi aşamasında her yatırım kategorisine yapılacak tutar belirlenir. Bu kategoriler için değişik finansal varlıklara yapılabilecek yatırım tutarları saptanır. Yatırımcı ilk defa bu aşamada reel bir girişimde bulunur. Portföy içindeki finansal varlıklar ve bu varlıklara ait ağırlıklar belirlenir.

[23] Ali Ceylan ve Turhan Korkmaz, a.g.e., 2000, s.258-260.

**Şekil 3.4. Portföy yönetimi aşamaları**




**Kaynak:** Ali Ceylan ve Turhan Korkmaz, a.g.e., 1998, s.16.

Portföy değerlendirmesi aşamasında, yatırım yapılan portföyün süreç içindeki değer değişimi ve verimi tespit edilir. Yatırım amaçlarına ve yatırım kriterlerine uygunluğu test edilir. Buna göre portföyün yapısında bir değişikliğe ihtiyaç olup olmadığına karar verilir.

Portföy revizyonu, portföy yönetiminin dinamik olmasını sağlayan aşamadır. Bu aşamada portföyün performansı ölçülür, alınması gereken önlemler belirlenir ve gerekli girişimler yapılır. Burada amaç, belirli risk seviyesinde portföyün getirisini maksimum yapmaktır[24].

[24] Ali Ceylan ve Turhan Korkmaz, a.g.e., 1998, s.16-23.

Portföy oluşturmanın temel amacı riskin dağıtılmasıdır. Yani portföyde çeşitlendirmeye gidilerek sistematik olmayan riskin azaltılmasına mümkünse yok edilmesi amaçlanır. Ancak oluşturulacak portföyde çeşitlendirme yapılarak risk azaltılırken çeşitlendirilmenin fazla olması, düşük getiriye sahip finansal varlıkların portföye girmesine sebep olur. Bu durumda daha az getiri elde edilmiş olur. Bu bağlamda portföy yönetimiyle ilgili iki yaklaşım vardır. Bunlar geleneksel portföy yönetimi ve modern portföy yönetimidir. Geleneksel portföy yönetimi yaklaşımında, portföy içindeki menkul kıymetlerin sayısı arttırılarak bir başka deyişle basit çeşitlendirme yapılarak bu amacın sağlanabileceği ileri sürülmüştür. Modern portföy yönetimi yaklaşımı ise bu görüşe karşı çıkıp finansal varlıkların getirilerinin riskin dağıtılmasında çok önemli olduğunu vurgulamışlardır. .

### 3.3.1. Geleneksel Portföy Yöntemi Yaklaşımı

Geleneksel portföy yaklaşımına göre portföy yönetimi bir sanattır. Bunun içinde çok dikkatli çalışmak gerekmektedir. Teorik araçları etkin bir şekilde kullanabilmek kişinin değişen bilgisine ve deneyimine bağlıdır. Bu yüzden bu yaklaşım, sezgi ve içe doğuş gibi sübjektif kararlar içermektedir. Geleneksel yönetimde amaç yatırımcının faydasını maksimum yapabilmektir. Bir başka ifadeyle yatırımın risk ve getirisine ilişkin fayda tercihlerini maksimum kılan portföyü belirleyebilmektir.

Geleneksel yönetimde portföyün getirisi, finansal varlıkların temettü değerleri ile belirli dönem içindeki fiyat artışıdır. Bu yüzden finansal varlıklara ilişkin olası getirilerini tahmin etmek gerekmektedir. Bir yandan da getirilere göre ortaya çıkan risk hesaplanmalıdır.

Birden fazla finansal varlıktan oluşan portföyün riski, tek bir finansal varlığın oluşturacağı riskten küçüktür. Çünkü portföy içindeki varlıkların getirileri aynı yönde hareket etmemektedir. Yani belirli dönem içinde bazı menkul kıymetler zarar ederken bazılarını kar sağlayacağından portföyün riski, tek bir finansal varlığın

riskinden küçük olacaktır. Bu prensip geleneksel portföy yönetimi yaklaşımına ilham vermiştir. Böylece geleneksel portföy yönetimi portföy içindeki finansal varlık sayısını arttırarak riskin dağıtılabileceğini öne sürmüştür. Bu yaklaşım, "bütün yumurtaları aynı sepete koymamak" şeklinde tanımlanmaktadır[25].

Geleneksel portföy yönetimi yaklaşımı temel olarak 4 aşamadan oluşmaktadır. Bunlar yatırımcıya ait bilgilerin toplanması, portföy amacının belirlenmesi, yatırım politikalarının belirlenmesi ve portföye dahil olacak finansal varlığın seçilmesi aşamalarıdır. Yatırımcıya ait bilgilerin toplanması aşamasında yatırımcının gelir düzeyi, tüketim eğilimi, tasarruf hacmi, eğitimi ve gelecek planları detaylı bir şekilde incelenir. Durumuna uygun yatırım yapabileceği finansal varlıklar belirlenerek tavsiyelerde bulunulur. Portföy amaçlarının belirlenmesi için yatırımcının yüklenmesi gereken sınırlamalar vardır. Bunların ortaya çıkarılması gerekmektedir. Geleneksel yönetimindeki bu sınırlamalar; cari fiyatlarla gelir elde etme ihtiyacı, sabit fiyatlarla gelir elde etme ihtiyacı, cari fiyatlarla anaparanın korunması ihtiyacı, sabit fiyatlarla anaparanın korunması ihtiyacı, vergiden muafiyet ihtiyacı ve yatırımcının tavrıdır. Portföye dahil olacak finansal varlıkların belirlenmesi ve portföy içindeki ağırlıklarının hesaplandığı aşama portföye dahil olacak finansal varlığın seçimi aşamasıdır. Geleneksel portföy yönetimi yaklaşımında seçim ilkeleri aşağıda belirtilmiştir.

- ✓ Hisse senetlerine yatırım yapılırken, daha çok sayıda ve farklı endüstrilerin hisse senetlerinin seçilmesi
- ✓ Tahvil yatırımlarında ise, aynı vadeye sahip tahvillerin portföydeki ağırlıklarının azaltılmasıdır.

Geleneksel yaklaşımda farklı vadelere sahip ve farklı endüstrilerde işlem gören çok sayıda finansal varlığın portföye katılmasıyla portföyün riskinin azalacağı savunulmaktadır. Bu seçim sürecinde üstünlük ilkesi kullanılmaktadır. Üstünlük ilkesi verilen herhangi bir getiri düzeyi için yatırımlar arasındaki en az riskli olanının

[25] Ali Ceylan ve Turhan Korkmaz, a.g.e., 1998, s.124.

seçilmesidir. Bir başka ifadeyle verilen bir risk düzeyi içinde en yüksek getiriye sahip olan en iyisidir[26].

### 3.3.2. Modern Portföy Yönetimi Yaklaşımı

Geleneksel portföy yönetimi, portföy içindeki finansal varlıkların sayısını artırarak riskin azaltılabileceğini, ne kadar çok ve farklı sektörlerdeki hisse senetlerine yatırım yapılırsa o kadar iyi bir çeşitlendirme olacağını savunmuştur. Bu düşünce 1950'li yılların başına kadar devam etmektedir. İlk defa 1952 yılında Harry Markowitz, yayınladığı bir makale ile portföyün riskinin ve beklenen getirisinin nasıl ölçülebileceği konusunda bir model önermiştir[27]. Bu makale ile modern portföy yönetiminin temelleri atılmıştır. Bu makalede yalnız portföy riskinin nasıl düşürülebileceği açıklanmamış, etkin bir çeşitlendirmenin nasıl yapılacağı da anlatılmaktadır. Daha sonra Markowitz, 1959'da yayımlanan kitabı ile etkin çeşitlendirme konusuna açıklık getirmiştir[28].

Markowitz'in önerdiği ortalama-varyans modeli karmaşık hesaplamalar içermektedir. Özellikle yatırım yapılacak hisse senedinin artması bu hesaplamaları elle yapılamaz hale getirmektedir. Bu sebeple ortalama-varyans modelini geliştiren öğrenci William Sharpe 1963 yılında tek indeks modelini önermiştir[29]. Bu model, daha çok alternatif hisse senetlerine yapılan yatırımların getirilerini maksimize etmeye çalışmaktadır. Daha sonraları Sharpe, Lintner ve Mossin, bütün tasarruf sahiplerinin modern portföy kuramına uygun olarak, menkul kıymetlere özellikle hisse senetlerine yatırım yapmaları durumunda, fiyatların ne yönde değişeceğini araştıran Sermaye Varlıklarını Fiyatlama modelini ortaya atmışlardır. Aynı yıllarda Steve Ross, SVFM alternatif olarak Arbitraj Fiyatlama Kuramını ortaya atmıştır[30].

---

[26] Ali Ceylan ve Turhan Korkmaz, a.g.e., 1998, s.125-135.
[27] Harry, Markowitz, "Portfolio Selection", **Journal of Finance**, 7(1), 77-91, Mart 1952.
[28] Harry Markowitz, **Portfolio Selection- Efficient Diversification of Investments**, New York, John Wiley&Sons, 1959.
[29] Willian Sharpe, "A Simplified Model for Portfolio Analysis", **Management Science**, 9(2), 277-293, 1963.
[30] Ali Ceylan ve Turhan Korkmaz, a.g.e., 1998, s.144.

Markowitz, çeşitlendirmenin hisse senetlerinin getirileri arasındaki ilişkiye (korelasyon) dayanması gerektiğini ve belli sayıdaki hisse senedinden sonra portföye eklenecek hisse senedinin portföyün riskini azaltmada fayda sağlamadığını savunmaktadır. Bu yüzden geleneksel portföy yönetimin savunduğu "daha çok menkul kıymete yatırım" düşüncesi ortadan kalkmıştır.

Yatırımcılar oluşturdukları portföyün getirisinin yüksek olmasını ve riskinin de düşük olmasını istemektedirler. Ancak getiri ile risk aynı yönde hareket eden kavramlardır. Getiri artarsa riskte artmaktadır. Riskin artışı ise yatırımcılar tarafından arzu edilen bir durum değildir. Markowitz önerdiği modelde bu soruna da çözüm getirmiş ve belirli bir getiri düzeyinde riskin minimizasyonu, belli risk düzeyinde getirinin maksimizasyonu olarak açıklamıştır. Böylece yatırımcının beklenen faydası maksimize edilecektir[31].

Markowitz ortalama-varyans modelini oluştururken bazı varsayımlar yapmıştır[32]. Bunlar:

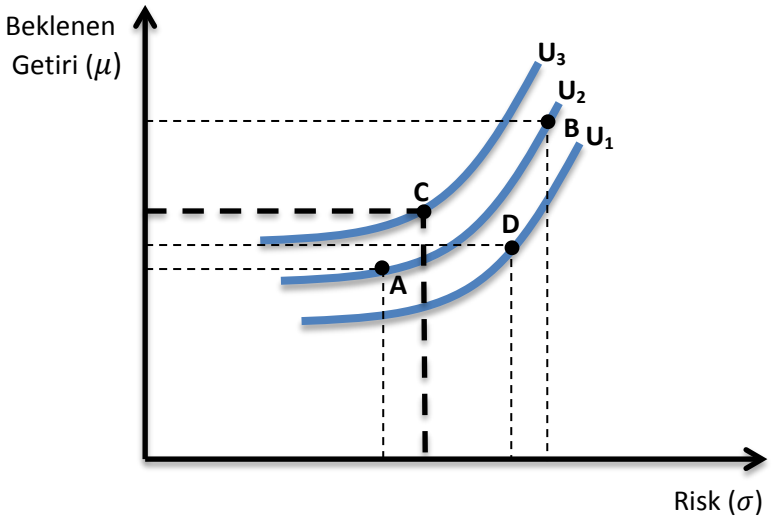
- ✓ Yatırımcıların amacı fayda fonksiyonunu maksimize etmektir.
- ✓ Yatırımcılar yatırım kararlarını yalnızca beklenen getiri ve riske göre almaktadır.
- ✓ Yatırımcıların risk ve getiri hakkındaki beklentileri homojendir. Yani aynı risk düzeyinde getirisi fazla olanı seçerler.
- ✓ Yatırımcılar özdeş zaman ufkuna sahiptirler.
- ✓ Sermaye piyasası oldukça etkindir. Fiyat değişimleri eş zamanlı olarak aktarılmaktadır.

Ortalama-Varyans Modeline geçmeden önce bu konunun temellerini oluşturan kayıtsızlık eğrileri ve etkin sınır kavramlarından bahsetmek gerekmektedir.

[31] Gürel Konuralp, **Sermaye Piyasaları: Analizler, Kuramlar ve Portföy Yönetimi**, 2. Baskı, Alfa Yayınları, 2005, s.313-314.
[32] Ali Ceylan ve Turhan Korkmaz, a.g.e., 1998, s. 149.

Yatırımcıların risk ve getiri tercihleri arasındaki ilişkiyi gösteren eğrilere *kayıtsızlık eğrileri* denir. Bu eğriler sayesinde yatırımcı katlandığı riskler karşılığında talep ettikleri getirileri, bir başka deyişle hangi risk düzeyinde ne kadar getiri beklediğini görebilir. Kayıtsızlık eğrileri her yatırımcıya göre değişir ve sonsuz sayıda kayıtsızlık eğrisi çizilebilir (Şekil 3.5.). Bu eğriler fayda teorisinden yararlanılarak çizilmektedir. Basit anlatımıyla, yatırımcıya çeşitli beklenen getiri ve risk kombinasyonuna sahip çeşitli portföyler önerilerek bunlar arasındaki tercihleri sorulur. Verdikleri cevaplar ışığında o yatırımcıya ait kayıtsızlık eğrilerinin yeri ve eğimi tespit edilir. Eğimin fazla oluşu bu yatırımcının riskten kaçtığını belirtmektedir. Eğim azaldıkça riske olan eğilim artmaktadır[33].

**Şekil 3.5. Kayıtsızlık eğrileri**




**Kaynak:** Gürel Konuralp, a.g.e., 2005, s.316.

Bu eğrilerin temel özellikleri aşağıda verilmiştir[34]:

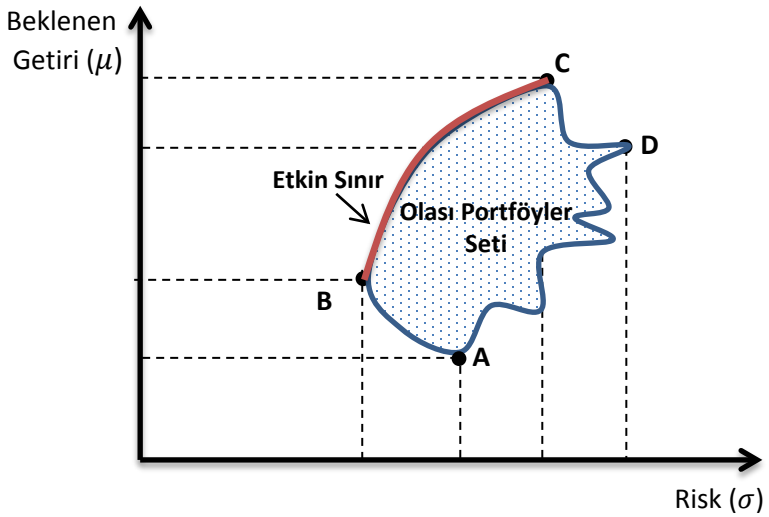
- ✓ Aynı kayıtsızlık eğrisi üzerinde yer alan tüm portföyler yatırımcılarına eşit düzeyde fayda sağlar.
- ✓ Kayıtsızlık eğrileri birbirini asla kesmezler.

[33] Gürel Konuralp, a.g.e., 2005, s.317.
[34] Mehmet B. Karan, a.g.e., 2011, s. 171.

- ✓ Yatırımcılar daha kuzey batıdaki kayıtsızlık eğrisinde yer alan bir portföyü, yeterince kuzey batıda yer almayan portföye göre daha fazla tercih eder.

Yatırımcılar için optimum portföy, en üst düzeydeki kayıtsızlık eğrisinin üzerinde yer alan portföydür. Bu eğri üzerinde sonsuz sayıda mümkün portföy bulunmaktadır. Yatırımcının bu portföylerden hangisinin kendine en uygun olduğunu tespit etmesi mümkün değildir. Bu aşamada Markowitz'in önerdiği etkin set teorisi, yatırımcıya sadece belli sayıda olası portföyler alt setine bakarak kendi optimum portföyünü oluşturma şansını tanımaktadır. Yatırımcı kendi optimum portföyünü belirlerken değerlendirmeye alacağı portföyler, değişen her risk seviyesinde en yüksek getiriyi sağlayan portföyler ve değişen her getiri seviyesinde en düşük riski sunan portföyler olmalıdır. Bu portföylerin oluşturduğu eğriye de *etkin set* ya da *etkin sınır* adı verilmektedir (Şekil 3.6.).

**Şekil 3.6. Etkin set (Etkin sınır)**

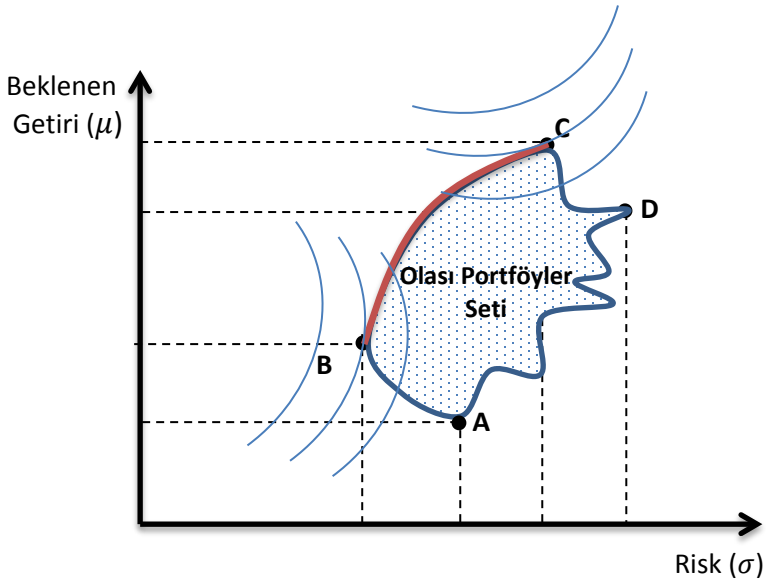



**Kaynak:** Gürel Konuralp, a.g.e., 2005, s. 320.

Etkin sınır üzerinde yer almayan portföyler yatırımcının amacına uygun olmayan portföylerdir. Etkin sınır üzerindeki her portföy yatırımcı için ideal portföydür. Bunlardan hangisinin seçileceği yatırımcının riske karşı duyarlılığına

bağlıdır. Yatırımcının risk üstlenmedeki tutumu kayıtsızlık eğrileri ile belirlenmektedir. Böylece portföye ait etkin sınır grafiği ile yatırımcının kayıtsızlık eğrisi grafiği birleştirilir (Şekil 3.7.). Kayıtsızlık eğrisinin etkin sınır eğrisini teğet geçtiği nokta yatırımcı için en optimal portföydür.

**Şekil 3.6. Yatırımcının etkin sınır eğrisi üzerindeki yeri**




**Kaynak:** Gürel Konuralp, a.g.e., 2005, s. 321.

Şekil 3.7.'de farklı risk tutumuna sahip iki yatırımcı gösterilmektedir. Riskten kaçan yatırımcının kayıtsızlık eğrileri B noktasında etkin sınır eğrisine teğet geçerken, riski seven yatırımcının kayıtsızlık eğrileri etkin sınır eğrisini C noktasında teğet geçmektedir. Buna göre riskten kaçan yatırımcı için optimal portföy B portföyü iken, riski seven yatırımcı için ise C portföyüdür[35].

[35] Gürel Konuralp, a.g.e., 2005, s.321.

### 3.3.2.1. Markowitz Ortalama-Varyans Modeli

Ortalama-Varyans Modeli portföyün beklenen getirisi ve riski ile ilgilenmektedir. Portföyün beklenen getirisi, portföy içindeki hisse senetlerinin beklenen getirilerinin ağırlıklı ortalamasıdır. Portföyün riski ise, portföyün standart sapması ile açıklanmaktadır. Standart sapma ise menkul kıymetler arasındaki kovaryans değerlerinin portföy içindeki ağırlıklarının çarpımının toplamı ile hesaplanmaktadır.

Markowitz, ulaşılması gereken beklenen getiri düzeyi için minimum varyanslı yani riskli portföyü bulmayı amaçlamaktadır. Bu durumda modelin amacı portföy varyansının bir başka deyişle portföy riskinin minimize edilmesidir ve aşağıdaki şekilde ifade edilmektedir.

$$Min. \quad \sum_{i=1}^{M}\sum_{j=1}^{M} w_i w_j \sigma_{i,j} \tag{3.15.}$$

Burada $\sigma_P^2$, portföyün varyansını; $w_i$, $i$. menkul kıymetin portföydeki ağırlığını; $\sigma_{i,j}$, $i$ ve $j$ menkul kıymetleri arasındaki kovaryans değerini ve $M$, portföy içindeki hisse senedi sayısını göstermektedir.

Standart Markowitz Modelinde kısıt söz konusudur. Bunlardan birincisi ulaşılması gereken beklenen getiri seviyesinin karşılanmasıdır. Matematiksel ifadesi Denklem 4.16'da gösterilmektedir.

$$\sum_{i=1}^{M} w_i \mu_i = \mu \tag{3.16.}$$

Burada $\mu_i$, $i$. menkul kıymetin beklenen getirisini ve $\mu$, ulaşılması gereken beklenen getiri düzeyini göstermektedir.

İkinci kısıt ise portföyde bulunan tüm menkul kıymetlerin ağırlıklarının toplamının 1 olmasını sağlayan ifade aşağıda gösterilmektedir.

$$\sum_{i=1}^{M} w_i = 1 \qquad (3.17.)$$

Portföy içindeki menkul kıymetlerin ağırlıklarının sıfırdan küçük olması mümkün değildir. Bu yüzden bu ağırlıklar 0 ile 1 arasında bir değer almaktadır. Bu durum modele kısıt olarak eklendiğinde Markowitz'in önerdiği Ortalama-Varyans Modeline ilişkin genel matematiksel ifade aşağıda gösterilmektedir.

$$\begin{aligned} Min. \quad & \sum_{i=1}^{M}\sum_{j=1}^{M} w_i w_j \sigma_{i,j} \\ s.t. \quad & \sum_{i=1}^{M} w_i \mu_i = \mu \\ & \sum_{i=1}^{M} w_i = 1 \\ 0 \leq\ & w_i \leq 1, \qquad i = 1, \dots, M \end{aligned} \qquad (3.18.)$$

Burada;

$w_i$ : $i$. menkul kıymetin portföydeki ağırlığı

$\sigma_{i,j}$ : $i$ ve $j$ menkul kıymetleri arasındaki kovaryans değeri

$\mu_i$ : $i$. menkul kıymetin beklenen getirisi

$\mu$ : ulaşılması gereken beklenen getiri düzeyi

$M$ : Portföydeki menkul kıymet sayısı

ile gösterilmektedir[36].

[36] Aydın Ulucan, a.g.e., 2004, s.18-19.

### 3.3.2.2. İndeks Modelleri

Optimal bir portföy oluşturabilmek için portföy içindeki hisse senetlerinin beklenen getirilerinin, varyans veya standart sapmalarının ve hisse senetleri arasındaki kovaryans değerlerinin ve korelasyon katsayılarının hesaplanması gerekmektedir. Portföye N tane hisse senedinin dahil olacağı düşünülürse hesaplanması gereken korelasyon katsayısı $(N^2-N)/2$ kadar olacaktır. N sayısının artması korelasyon katsayılarının hesaplanmasını güçleştirmektedir. Bu yüzden Markowitz çeşitlendirmesi büyük zaman ve maliyet unsuru taşımaktadır. Ortalama-Varyans Modelindeki bu eksiklik William Sharpe tarafından geliştirilen Tek İndeks Modeli ile giderilmiştir. Tek indeks modelinde bir tek menkul kıymet getirisi ile bir endeks arasında doğrusal bir ilişkinin olduğu varsayılmaktadır. Bu endeks, gayri safi milli hasıla olabileceği gibi başka bir endeks de olabilir[37].

Tüm endeks modellerinin genel yapısında, menkul kıymetlerin getirilerini açıklayacak bir bağımsız değişken bulunmaktadır. Burada menkul kıymet getirileri ile bağımsız değişken arasında yüksek dereceli korelasyon olması beklenmektedir. Gerçek hayatta menkul kıymet getirilerini açıklayan birçok değişken söz konusudur. Portföy yönetimi konusunda önemli yer teşkil eden indeks modelleri, Markowitz analizleri için gerekli olan beklenen getiri, varyans ve kovaryans bilgilerini içerdiği gibi, portföyün bazı faktörlere olan duyarlılığını da ortaya çıkarmaktadır[38].

#### 3.3.2.2.1. Tek İndeks Modeli

Markowitz çeşitlendirmesinden kaynaklanan sakıncanın ortadan kaldırılması için tek indeks modeli ya da ortalama modeli kullanılmaktadır. Tek indeks modelinde hisse senetleri getirileri arasındaki korelasyon katsayıları yerine, beta katsayıları kullanılmaktadır. Burada beta katsayıları, her bir hisse senedi getirisinin piyasa ortalama getirisi veya piyasa endeksi ile olan ilişkisini ifade eder.

[37] Ali Ceylan ve Turhan Korkmaz, a.g.e., 2000, s.287.
[38] Mehmet B. Karan, a.g.e., 2011, s. 230.

Sharpe, bütün menkul kıymetlerin piyasa ile doğrusal bir ilişkiye sahip olduğunu düşünmektedir. Bu durumun basit regresyon modeli ile ifade edilebileceğini söylemiştir. Ayrıca bu modelde menkul kıymetlerin piyasa ile olan ilişkisi çerçevesinde birbirleriyle de ilişkili olduklarını varsaymaktadır.

Menkul kıymetlerin getirisini, enflasyon oranındaki dalgalanmalar, faiz oranlarındaki değişimler gibi makro olaylar ve grev, yangın, yeni ürün geliştirme gibi mikro olaylar etkilemektedir. Makro olaylardan pazar portföyünün getirileri de etkilenmektedir[39].

Tek indeks modelinin genel ifadesi aşağıda gösterilmektedir[40].

$$R_i = a_i + \beta_i R_m + e_i \qquad (3.19.)$$

Burada;

$R_i$ : $i$ hisse senedinin getirisi

$a_i$ : kesim oranı (sabit sayı)

$\beta_i$ : $i$ hisse senedinin piyasa indeksi ile olan ilişkisini gösteren katsayı

$R_m$ : Pazar indeksinin getirisi

$e_{it}$ : Hata terimi

Tek indeks modelinde $a_i, \beta_i$ ve hata terimleri varyansı olan $\sigma_{ei}^2$ regresyon analizi ya da zaman serileri yardımıyla bulunmaktadır. Bu özelliği sayesinde kovaryans yapısını kullanan diğer modellerden farklıdır.

Menkul kıymetin beklenen getirisi aşağıdaki formülle hesaplanmaktadır.

$$\bar{R_i} = a_i + \beta_i \bar{R}_m \qquad (3.20.)$$

[39] Ali Ceylan Ve Turhan Korkmaz, a.g.e., 2000, s.288.

[40] Edwin J. Elton ve dğr., Modern Portfolio Theory and Investment Analysis, 6. Baskı, USA, John Wiley&Sons, Inc., 2003, s.132.

$\bar{R}_m$ : Pazar indeksinin beklenen değeri

Menkul kıymetin varyansı;

$$\sigma_i^2 = \beta_i^2 \sigma_m^2 + \sigma_{ei}^2 \quad (3.21.)$$

denklemi yardımıyla hesaplanmaktadır.

Burada;

$\sigma_m^2$ : Pazar indeksinin varyansı

$\sigma_{ei}^2$ : Hata terimlerinin varyansı

İki menkul kıymet arasındaki korelasyon;

$$\sigma_{ij} = \beta_i \beta_j \sigma_m^2 \quad (3.22.)$$

denklemi yardımıyla hesaplanmaktadır[41].

Portföyün beklenen getirisini hesaplamak için ise öncelikle $\alpha_P$ ve $\beta_P$ değerlerinin hesaplanması gerekmektedir. $\alpha_P$, her bir menkul kıymete ait $\alpha$ değerlerinin ağırlıklı ortalamasıdır (Denklem 3.23.). $\beta_P$ ise her bir menkul kıymete ait $\beta$ değerlerinin ağırlıklı ortalamasıdır (Denklem 3.24.).

$$\alpha_P = \sum_{i=1}^{M} w_i \alpha_i \quad (3.23.)$$

$$\beta_P = \sum_{i=1}^{M} w_i \beta_i \quad (3.24.)$$

[41] Edwin J. Elton ve dğr., a.g.e., 2003, s.134.

Bu bilgiler ışığında portföyün beklenen getirisi aşağıdaki Denklem 3.25. yardımıyla hesaplanmaktadır.

$$\bar{R}_P = \alpha_P + \beta_P \bar{R}_m \tag{3.25.}$$

Portföyün varyansı ise aşağıda Denklem 3.26. yardımıyla hesaplanmaktadır[42].

$$\sigma_p^2 = \beta_p^2 \sigma_m^2 + \sum_{i=1}^{M} w_i^2 \sigma_{ei}^2 \tag{3.26.}$$

Burada;

$\sigma_p^2$ : Portföyün varyansı

$\beta_p^2$ : Her bir menkul kıymete ait $\beta$ değerlerinin ağırlıklı ortalamasının karesi

$\sigma_m^2$ : Pazar indeksinin varyansı

$w_i^2$ : Her bir menkul kıymetin portföy içindeki ağırlığı

$\sigma_{ei}^2$ : Hata terimlerinin varyansı

$M$ : Menkul kıymet sayısı

[42] Edwin J. Elton ve dğr., a.g.e., 2003, s.137-138.

# 4. UYGULAMA

## 4.1. Kardinalite Kısıtlı Ortalama Varyans Modeli

Bu çalışmada, bilinen Markowitz Ortalama-Varyans Modeli[1] geliştirilerek elde edilen "Kardinalite Kısıtlı Ortalama Varyans Modeli"[2-3] kullanılmıştır. Portföy optimizasyonu problemine Markowitz'in önerdiği Ortalama-Varyans Modelinin matematiksel ifadesi aşağıda tanımlanmıştır.

$$
\begin{aligned}
Min. \quad & \sum_{i=1}^{M}\sum_{j=1}^{M} w_i w_j \sigma_{i,j} \\
s.t. \quad & \sum_{i=1}^{M} w_i \mu_i = \mu \\
& \sum_{i=1}^{M} w_i = 1 \\
0 \leq\ & w_i \leq 1, \qquad i = 1, \dots, M
\end{aligned}
\tag{4.1.}
$$

Burada;

$w_i$ : $i$. hisse senedinin ağırlığı

$\sigma_{i,j}$ : $i$ ve $j$ hisse senetleri arasındaki kovaryans değeri

$\mu_i$ : $i$. hisse senedinin beklenen getirisi

$\mu$ : Ulaşılmak istenen beklenen getiri düzeyi

$M$ : Mevcut hisse senedi sayısı

Çeşitli $\mu$ değerlerine karşın amaç fonksiyonunun aldığı değerleri gözlemleyebilmek için standart modele riski önleme parametresi adı verilen ve 0 ile

[1] H.M. Markowitz, a.g.e., 1952.

[2] T.J. Chang, N. Meade, J.E. Beasley, ve Y.M. Sharaiha, "Heuristics for cardinality constrained portfolio optimization", **Computers and Operation Research**, 27, 1271-1302, 2000.

[3] Alberto Fernández, Sergio Gómez, "Portfolio selection using neural networks", **Computers&Operations Research**, 34, 1177-1191, 2007

1 arasında değer olan λ parametresi ilave edilir. Böylece model aşağıdaki gibi ifade edilir:

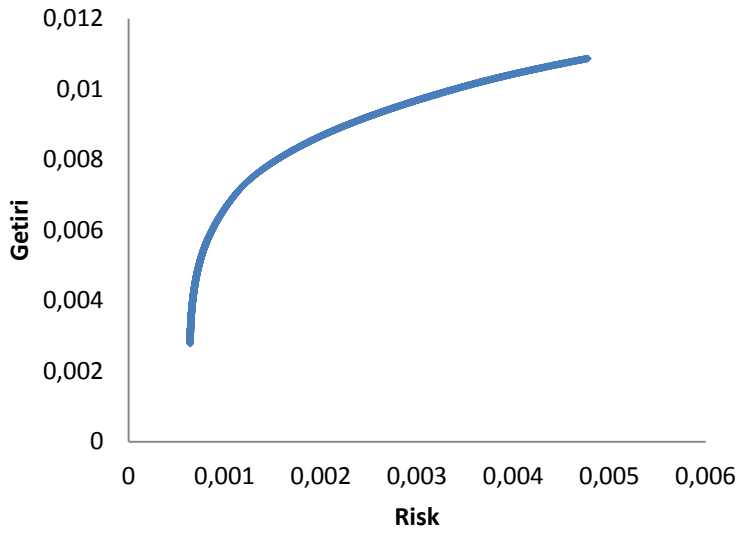
$$Min. \quad \lambda \sum_{i=1}^{M} \sum_{J=1}^{M} w_i w_j \sigma_{ij} - (1-\lambda) \sum_{i=1}^{M} w_i \mu_i$$
$$s.t. \quad \sum_{i=1}^{M} w_i = 1 \qquad (4.2.)$$
$$0 \leq w_i \leq 1, \quad i = 1, \dots, M$$

Eğer λ parametresi 0 değerini alırsa, model riski dikkate almadan portföyün ortalama getirisini maksimize etmeye çalışır. Tam tersine λ parametresi 1 değerini alırsa, model getiriye bakmaksızın portföyün riskini minimize etmeye çalışacaktır. Bu demek oluyor ki λ parametresinin değeri yatırımcının riske olan duyarlılığına göre değişir.

Her bir farklı λ değeri ile bulunan getiri ve risk değerleri, grafik üzerinde işaretlendiğinde ortaya sürekli bir fonksiyon eğrisi çıkmaktadır. Bu eğriye Markowitz teorisinde "Etkin Sınır" adı verilmektedir[4]. Etkin sınır üzerindeki her nokta optimal risk ve getiri birleşimini göstermektedir. Formül 4.2.' de verilen model ile elde edilen etkin sınır eğrisi, risk ve getiri arasındaki en iyi değiş tokuşu göstermektedir. Şekil 4.1.'de yer alan eğri, ileride açıklanacak olan en küçük karşılaştırma problemine (Hang Seng) ait etkin sınırı göstermektedir. Bu eğri 2000 farklı λ değeri için hesaplanmış 2000 farklı amaç fonksiyonu değerine ait getiri ve risklerin kesişimini göstermektedir. Her bir çözüm, etkin sınır üzerindeki bir noktaya denk gelmektedir. Bu eğriye "Standart Etkin Sınır" adı verilmektedir[5].

[4] H.M. Markowitz, a.g.e., 1952, s. 87.
[5] Alberto Fernández, Sergio Gómez, a.g.e., 2007, s. 2.

**Şekil 4.1.** Hang Seng karşılaştırma problemine ait standart etkin sınır




Kardinalite Kısıtlı Ortalama Varyans modelini tanımlayabilmek için var olan standart modele bazı kısıtlar eklenmesi gereklidir. Bunlardan birincisi, portföyde seçilebilecek hisse senedi sayısının K ile ifade edilen bir değere eşit olmasıdır. Bu kısıtın matematiksel ifadesinde 0 ya da 1 değerini alabilen bir $z_i$ karar değişkeni bulunmaktadır. Bu değer 0 değerini alırsa, *i* hisse senedi portföyde yer almamaktadır, eğer 1 değerini alırsa, *i* hisse senedi portföyde yer almaktadır denilir. İkinci kısıt ise tüm hisse senetlerinin portföy içindeki ağırlıklarının belirli bir alt sınır ($\varepsilon_i$) ile üst sınır ($\delta_i$) arasında olma zorunluluğudur. Bu tanımlamalar sonucunda kardinalite kısıtlı ortalama varyans (KKOV) modeli aşağıdaki gibi olur[6].

$$Min. \quad \lambda \sum_{i=1}^{M} \sum_{J=1}^{M} w_i w_j \sigma_{ij} - (1-\lambda) \sum_{i=1}^{M} w_i \mu_i \tag{4.3.}$$

$$s.t. \quad \sum_{i=1}^{M} z_i = K \tag{4.4.}$$

[6] Tunçhan Cura, "Particle swarm optimization approach to portfolio optimization", **Nonlinear Analysis: Real World Applications**, 10(4), 2396-2406, 2009.

$$\sum_{i=1}^{M} w_i = 1 \tag{4.5.}$$

$$\varepsilon_i z_i \leq w_i \leq \delta_i z_i \quad , \qquad i = 1 \dots M \tag{4.6.}$$

$$z_i \in \{0,1\} \quad , \quad i = 1 \dots M \tag{4.7.}$$

$$0 \leq \lambda \leq 1 \tag{4.8.}$$

Burada:

$M$ : Mevcut hisse senedi sayısı

$w_i$ : $i$ hisse senedinin ağırlığı, (karar değişkeni) $(i = 1, \dots, M)$

$\mu_i$ : $i$ hisse senedinin beklenen getirisi $(i = 1, \dots, M)$

$K$ : Portföydeki hisse senedi sayısı

$z_i$ : Hisse senedinin seçilip seçilmediğini gösteren karar değişkeni

$\varepsilon_i$ : $i$ hisse senedinin alabileceği en düşük ağırlık

$\delta_i$ : $i$ hisse senedinin alabileceği en yüksek ağırlık

$\sigma_{ij}$ : $i$ ve $j$ hisse senetleri arasındaki kovaryans değeri $(i, j = 1, \dots, M)$

## 4.2. Kardinalite Kısıtlı Ortalama Varyans Portföy Optimizasyonu Problemine Yapay Arı Kolonisi Algoritması Yaklaşımı

KKOV modeli, hem kuadratik hem tamsayılı programlama probleminin karışımıdır. Bu tür problemler için etkin bir algoritma bulunmamaktadır. Bu yüzden araştırmacılar sezgisel tekniklere yönelmişlerdir. Literatüre bakıldığında genetik algoritma, tabu arama algoritması, tavlama benzetimi, parçacık sürü optimizasyonu gibi başarılı sezgisel teknikler ile yapılmış çalışmalar mevcuttur. Bu çalışma, yapay arı kolonisinin KKOV Portföy Optimizasyonu Problemi karşısındaki başarısını sınamak için yapılmıştır.

YAKA, 2005 yılında Kayseri Erciyes Üniversitesi Bilgisayar Mühendisliği Bölümünde öğretim üyesi olan Derviş Karaboğa tarafından ortaya atılmıştır. YAKA, çok boyutlu ve çok doruklu optimizasyon problemlerinin çözümü için geliştirilmiş popülasyon tabanlı bir algoritmadır. YAKA'da görevli arı, kaşif arı ve gözcü arı

olmak üzere üç tip arı bulunmaktadır. Koloninin yarısı görevli arılardan diğer yarısı ise gözcü arılardan oluşmaktadır. Görevli arı sayısı da besin kaynağı sayısına yani olası çözümlerin sayısına eşittir. Görevli arılar, yeni besin kaynaklarının bulunması ve gözcü arılara bu konuda bilgi verilmesinden sorumludur. Gözcü arı ise yuvada bekleyip görevli arıdan gelen bilgileri değerlendirir ve hangi besin kaynağına gideceğine karar verir. Kaşif arılar ise görevli arıların işleri bitince ortaya çıkmaktadır. Kaşif arılar, yeni besin kaynaklarını ön bilgi olmadan rassal olarak bulduğundan algoritmanın daha önceden keşfedemediği zengin besin kaynaklarını yani yüksek uygunluktaki olası çözümler kümesini bulabilirler.

YAKA'da öncelikle rassal olarak portföy popülasyonu oluşturulur. Bu portföyler, görevli arı ve gözcü arı tarafından manipüle edilerek iyileştirilmeye çalışılır. Çözüm değerinde belirli bir limit değerine kadar iyileşme sağlamayan portföyler kullanılmamaktadır. Kâşif arılar elde edilen portföyleri tamamen bozmakta ve yeni aday portföyler için çözüm değerleri elde etmektedir. Elde edilen çözüm değerleri önceki çözüm değerlerinden çok kötü olsa bile yerel minimuma takılma riskine engel olmak için kabul edilmekte ve işçi arılar ve gözcü arılar bu rotaları değiştirerek yeni çözüm değerleri elde etmektedirler. İlk elde edilen çözüm değerleri ile kâşif arı safhası sonucunda elde edilen çözüm değerleri karşılaştırılmakta ve en iyi çözüm değeri yeniden hafızaya alınmaktadır. Bu durum belirli iterasyon adedince tekrar edilmekte ve böylelikle elde edilen en iyi çözüm değeri için yerel minimuma takılıp kalma riski ortadan kalkmaktadır. YAKA ile yerel maksimumların ve global maksimumların tespit edilmesi nispeten kolaydır ve YAKA bu konuda oldukça etkili sonuçlar üretmektedir.

Bu çalışmada KKOV modelinin çözümünde kullanılan YAKA yaklaşımının anlatımını kolaylaştırmak için tüm algoritma adım adım anlatılacaktır.

Algoritmada 3 temel fonksiyon çalışmaktadır. Bunlardan birincisi KKOV modeli için YAKA yaklaşımının uygulandığı ana program fonksiyonu, ikincisi amaç fonksiyon değerlerinin bulunduğu hesapla fonksiyonu ve üçüncüsü önerilen aday çözümlerin kısıtlara uygun hale getirildiği düzenleme fonksiyonudur.

Algoritma ilk olarak portföy içindeki hisse senetlerinin ağırlıklarını gösteren $w_{ij}$ ve hisse senedinin portföyde bulunup bulunmadığını gösteren $z_{ij}$ değerlerinin rastgele olarak belirlenmesiyle başlamaktadır. Burada $i$ değerleri besin kaynağının indisini ifade etmektedir ve algoritmada besin kaynağı sayısı BS ile gösterilmektedir. $j$ değerleri ise hisse senedini gösterirken toplam sayı M ile ifade edilmektedir. Bunların yanında besin kaynağı sayısı kadar deneme sayacı ($ds_i$) tanımlanır ve değeri sıfırlanır. Her iterasyonda elde edilen aday portföyler kısıtlara uygun hale getirilmek için düzenleme fonksiyonuna gönderilirler. Uygun hale gelmiş çözümler hesapla fonksiyonuna gönderilerek amaç fonksiyonu değerleri ($f_i$) hesaplanır. Bu değerler içinden en iyi çözüm hafızaya alınır. Bu aşamadan sonra belirli bir durdurma kriterine kadar sırasıyla görevli arı, gözcü arı ve kaşif arı aşamalarından geçilmektedir. Algoritma durdurma kriteri sağlanana kadar devam eder. Ayrıca algoritmanın farklı $\lambda$ değerlerine göre hesaplanabilmesi için belirli bir artırma değeri ($\Delta\lambda$) ile 0'dan 1'e kadar çalıştırılmıştır. Burada $\Delta\lambda$=0.02 olarak belirlenmiştir.

Görevli arı aşamasında tüm besin kaynakları yani portföyler kümesi için rassal komşular seçilir ve yine rassal olarak seçilen bir hisse senedinin ağırlığı $w_{ij}$ değiştirilir. Elde edilen yeni ağırlık eğer 0'dan büyükse bu ağırlığa ait $z_{ij}$ değeri 1 yapılıp o hisse senedi portföye dahil edilir. Bu aşamadan sonra düzenleme ve hesapla fonksiyonu çalıştırılır. Daha sonra elde edilen sonuç en iyi $f_i$ değeri ile kıyaslanır. Daha iyi sonuçsa hafızaya alınıp deneme sayacı sıfırlanır, değilse deneme sayacı bir arttırılır. Bu süreç besin kaynağı sayısı kadar tekrarlanır.

Gözcü arı aşamasında ise hangi besin kaynaklarının seçileceğinin belirlenmesinde kullanılan olasılık değerlerinin hesaplanması gerekmektedir. Olasılıkları hesaplamadan önce tüm amaç fonksiyonu değerlerinin pozitif hale getirilmesi gerekmektedir. Çünkü olasılık hesabı yapılırken amaç fonksiyonu değerlerinin toplamı alınacaktır. Riskin getiriden düşük olduğu durumlarda amaç fonksiyonu değeri negatif olacağından olasılık değerleri hesaplanamayacaktır. $f_i$'ler pozitif hale getirilmeden önce $f_{min}$ değeri bulunur. Eğer $f_{min} \leq 0$ ise $f_i$ değerleri Formül 4.9. yardımıyla yeniden hesaplanır. Burada 0,01 gibi küçük bir sayının da

eklenmesi $f_{min}$ değerine sahip portföyün de olasılık hesabında yer almasını sağlamak içindir.

$$f_i^{yeni} = f_i + |f_{min}| + 0{,}01 \quad i = 1, \dots, BS \tag{4.9.}$$

Gözcü arıların hangi besin kaynaklarını seçeceğini belirleyen olasılık değerleri ($p_i$), Formül 4.10. yardımıyla hesaplanır.

$$p_i = \frac{1}{f_i} \Big/ \sum_{i=1}^{BS} \frac{1}{f_i} \tag{4.10.}$$

Burada $f_i$'ler yerine $1/f_i$'lerin kullanılmasının sebebi düşük $f_i$ değerlerine sahip portföylerin seçilmesini arttırmaktır.

Olasılıkların hesabından sonra tüm gözcü arılar için besin kaynaklarının belirlenmesi işlemine geçilir. Rastgele bir sayı ile $p_i$ değerleri karşılaştırılır. Eğer $p_i$ değeri daha büyükse görevli arı aşamasında yapılan işlemler tekrarlanır, tersi durumda ise $i$ değeri 1 arttırılarak besin kaynağı aramaya devam edilir. Bu süreç tüm gözcü arılar (GS) tamamlanana kadar devam eder.

Kaşif arı aşamasında ise tüm olası çözümlerin deneme sayaçları belirli bir limit değerini aşmışsa aşan çözüm tamamen silinip yerine rassal olarak belirlenen yeni çözüm getirilir. Elde edilen çözüm en iyi çözüm ile karşılaştırılır. Daha iyi bir çözüm ise hafızaya alınır.

Yukarıda anlatılan KKOV modeli için YAKA yaklaşımına ilişkin sözde kod Şekil 4.2.'de gösterilmektedir.

**Şekil 4.2. KKOV modeli için YAKA yaklaşımı**

**Begin**
λ=0
**While** (λ ≤ 1)
Rasgele olarak $w_{ij}$ ve $z_{ij}$ değerleri hesapla
$i = 1,2,\dots,BS\ ;\ j = i,2,\dots,M.$
$ds_i$=0;
**düzenleme** ($w_{ij}, z_{ij}$)
$f_i$ = **hesapla**(λ, $w_{ij}$,$\mu$ ,$\sigma$)
$f_i^{eniyi}$ =En iyi $f_i$
$w_{ij}^{eniyi}$ =En iyi $w_{ij}$
sayac=1
**While**(sayac ≤ durdurmakriteri)
**for** i=1 **to** BS
k ve m değerlerini rasgele olarak ata
$w^{ge} = w_{i,:}$
$z^{ge} = z_{i,:}$
$w_m^{ge} = w_{im} + \phi_{im}(w_{im} - w_{km})$
**if** $w_m^{ge} > 0$ **then**
$z_m^{ge} = 1$
**End if**
**düzenleme** ($w_m^{ge}, z_m^{ge}$)
$f^{ge}$ = **hesapla** (λ, $w_m^{ge}$, $\mu$ , $\sigma$)
**if** ($f^{ge} < f_i$) **then**
$w_{i,:} = w_m^{ge}$
$z_{i,:} = z_m^{ge}$
$f_i = f^{ge}$
$ds_i = 0$
**Else**
$ds_i = ds_i + 1$
**End if**
**End for**

$f_{min} = \min(f_i), \quad i = 1,\dots,BS$
**if** $f_{min} \le 0$ **then**
$f_i = f_i + |f_{min}| + 0{,}01 \quad i = 1,\dots,BS$
**End if**

$p_i = \frac{1}{f_i} / \sum_{i=1}^{BS} \frac{1}{f_i}$
$i = 1$
gozcu= 0
**while** ( gozcu<GS)
**if** (rand< $p_i$) **then**
gozcu=gozcu+1
k ve m değerlerini rasgele olarak ata.
$w^{ge} = w_{i,:}$
$z^{ge} = z_{i,:}$
$w_m^{ge} = w_{im} + \phi_{im}(w_{im} - w_{km})$
**if** $w_m^{ge} > 0$ **then**
$z_j^g = 1$
**End if**
**düzenleme** ($w_m^{ge}, z_m^{ge}$)
$f^{ge}$ = **hesapla** (λ, $w_m^{ge}$, $\mu$ , $\sigma$)
**if** ($f^{ge} < f_i$) **then**
$w_{i,:} = w_m^{ge}$
$z_{i,:} = z_m^{ge}$
$f_i = f^{ge}$
$ds_i = 0$
**Else**
$ds_i = ds_i + 1$
**End if**
**End if**
$i = i + 1$
**if** ($i = BS + 1$) **then**
$i = 1$
**End if**
**End while**
$f_i^{eniyi}$ =En iyi $f_i$
$w_{ij}^{eniyi}$ =En iyi $w_{ij}$
ceza=$ds_i$ > limit
**for** yy=1 **to** BS
**if** (ceza(yy)=1) **then**
$ds_{yy} = 0$
Rasgele olarak $w^*$ ve $z^*$ değerlerini hesapla.
**düzenleme** ($w^*$, $z^*$)
$f^*$ = **hesapla**(λ, $w^*$, $\mu$ , $\sigma$)
$w_{yy,:} = w^*$
$z_{yy,:} = z^*$
$f_{yy} = f^*$
**End if**
**End for**
sayac=sayac+1
**End While**
$\lambda = \lambda + \Delta\lambda$
**End While**
**End Begin**

Aday çözümlere ilişkin amaç fonksiyonu değerlerinin bulunduğu hesapla fonksiyonunda KKOV modelinin amaç fonksiyonu kullanılmaktadır. Algoritmanın kullandığı amaç fonksiyonu ($f_i$), Formül 4.11.'de gösterilmektedir.

$$f_i = \lambda \sum_{j=1}^{M} \sum_{k=1}^{M} z_{ij} w_{ij} z_{ik} w_{ik} \sigma_{jk} - (1-\lambda) \sum_{j=1}^{M} z_{ij} w_{ij} \mu_j \qquad (4.11.)$$

Önerilen portföyler, düzenleme fonksiyonu yardımıyla KKOV modeli kısıtlarını ihlal etmemeyi garantilemektedir. Bu fonksiyon iki aşamadan oluşmaktadır. Birinci aşaması seçilen $K^*$ hisse senedi sayısının, belirlenen $K$ hisse senedi sayısına eşitlenmesidir. İkincisi ise seçilen hisse senetlerinin portföy içindeki ağırlıklarını gösteren $w_{ij}$ değerlerinin önceden belirlenmiş $\varepsilon_i$ ağırlık değeri alt sınırı ile $\delta_i$ ağırlık değeri üst sınırı arasında kalmasını sağlamaktır. Sınama veri setleri için tüm hisse senetlerinin aynı alt ve üst sınır değerine sahip olduğu varsayılmıştır. Algoritmada $K = 10$, $\varepsilon_j = 0{,}01$ ve $\delta_j = 1$ olarak belirlenmiştir. Düzenleme fonksiyonuna ilişkin sözde kod Şekil 4.3.'de gösterilmiştir.

Birinci aşamada $K^* > K$ olduğu sürece en düşük $w_{ij}$ değerine sahip hisse senedi belirlenir ve ilgili hisse senedinin $z_{ij}$ değeri sıfırlanır ve tekrar $K^*$ değeri hesaplanır. $K^* < K$ olduğu sürece rassal olarak bir $w_{ij}$ değeri seçilir. İlgili hisse senedinin $z_{ij}$ değeri sıfır ise bu değer 1 yapılır ve tekrar $K^*$ değeri hesaplanır.

İkinci aşamada öncelikle seçilen hisse senetlerine ait $w_{ij}$ değerleri normalize edilir. Daha sonra üst limitin üstünde kalan ağırlıkların toplamı hesaplanır ve $\eta$ değişkenine atanır. Aynı şekilde alt limitin altında kalan ağırlıkların toplamı hesaplanır ve $\theta$ değişkenine atanır. $\eta$ ve $\theta$ değerlerinin sıfır olması tüm hisse senetlerine ait ağırlıkların alt ve üst sınır içinde olduklarını göstermektedir. Eğer iki değişkenden biri sıfırdan farklı ise ihlal söz konusudur. Bu durumunda $j$ hisse senedinin portföydeki ağırlığı için artırılması mümkün miktar ($t_j$) ile azaltılabilmesi mümkün miktar ($e_j$) ve bunların toplamları hesaplanır ($\delta^* = \sum_{j \in Q} t_j$, $\varepsilon^* = \sum_{j \in Q} e_j$).

Çünkü fazlalık ya da eksik ağırlık değerine sahip olan hisse senetlerinin yeni $w_{ij}$ değerleri, artırılabilecek ya da azaltılabilecek boşluk miktarlarının, toplam üst limitin üstünde kalan ya da toplam alt limitin altında kalan değerler ile oransal çarpımıyla bulunur. Böylelikle daha fazla eksiği olan hisse senedi, diğer eksikliği bulunan hisse senetlerine göre daha fazla pay almaktadır. Aynı şekilde daha az fazlalığı olan senet, fazlalığı daha çok olan hisse senedine göre daha az pay vermektedir.

**Şekil 4.3. Her bir besin kaynağı için düzeltme fonksiyonu**

**düzenleme(w,z)**
$w_{ij}$ : i portföy içindeki j hisse senedinin ağırlığı
$z_{ij}$: j hisse senedinin portföy içindeki karar değişkeni
$Q$: i. portföy için $\forall j: z_{ij} = 1$ olan hisse senetleri kümesi
$K^* = \sum z_{ij}$
**Begin**
**While ($K^*$>K)**
$j$ = En düşük $w_{ij}$ sahip hisse senedinin indeksini seç, $j \in Q$
$z_{ij} = 0$
$K^* = \sum z_{ij}$
$Q = Q - [j]$
**End while**
**While ($K^*$<K)**
$j$ = Rasgele bir $w_{ij}$ sahip hisse senedinin indeksini seç, $j \notin Q$
**İf** $z_{ij} = 0$ **then**
$z_{ij} = 1$
$K^* = \sum z_{ij}$
$Q = Q \cup [j]$
**End if**
**End while**
**while(true)**
$\chi = \sum_{j \in Q} w_{ij}$
$w_{ij} = w_{ij}/\chi \quad \forall j \in Q$
$\eta = \sum_{j \in Q} \max(0, w_{ij} - \delta_j) \; \forall j \in Q$
$\theta = \sum_{j \in Q} \max(0, \varepsilon_j - w_{ij}) \; \forall j \in Q$
**if** $\pi$ ==0 **and** $\theta$ ==0 **then**
düzenleme fonksiyonunu sonlandır
$t_j = \max(0, \delta_j - w_{ij}) \; \forall j \in Q$
$\delta^* = \sum_{j \in Q} t_j$
$e_j = \max(0, \; w_{ij} - \varepsilon_j) \; \forall j \in Q$
$\varepsilon^* = \sum_{j \in Q} e_j$
**for** j=1 **to** M
**if** $z_{ij}$=1 **then**
**if** $t_j$>0 **then**
$w_{ij} = w_{ij} + (t_j/\delta^*) * \eta;$
**else**
$w_{ij} = \delta_j$
**end if**
**if** $e_j$>0 **then**
$w_{ij} = w_{ij} - (e_j/\varepsilon^*) * \theta;$
**else**
$w_{ij} = \varepsilon_j$
**End if**
**End if**
**End For**
**End While**
**End Begin**

Portföy içindeki $j$ hisse senedinin arttırılabilecek boşluğu $t_j$>0 ise bu hisse senedine ait yeni $w_{ij}$ değeri Formül 4.12. yardımıyla hesaplanmaktadır. Aksi durumunda bu hisse senedine ait yeni $w_{ij}$ değeri, üst sınır olan $\delta_j = 1$ değerine eşitlenir. Eğer $j$ hisse senedinin azaltılabilecek boşluğu $e_j$>0 ise bu hisse senedine ait yeni $w_{ij}$ değeri Formül 4.13. yardımıyla hesaplanmaktadır. Aksi durumunda bu hisse senedine ait yeni $w_{ij}$ değeri, alt sınır olan $\varepsilon_j = 0{,}01$ değerine eşitlenir.

$$w_{ij} = w_{ij} + (t_j/\delta^*) * \eta; \quad (4.12.)$$

$$w_{ij} = w_{ij} - (e_j/\varepsilon^*) * \theta; \quad (4.13.)$$

## 4.3. Parametre Seçimi

YAKA ile PO'nun uygulanmasında problemlerde kullanılan parametreler için çeşitli denemeler yapılmış olup bu denemeler sonucunda en iyi sınama sonucunu veren parametre değerleri tercih edilmiştir. Problemlerde yer alan parametreler hisse senedi sayısı, işçi arı sayısı, gözcü arı sayısı, kaşif arı sayısı, besin sayısı, limit ve iterasyon sayısıdır. Hisse senedi sayısı örnek problemlere göre değişmektedir. Burada limit parametresi, bir arının arka arkaya iyileşme olmaksızın en fazla deneme sayısını göstermektedir. Hisse senedi sayısı ise P ile temsil edilmektedir. Parametreler için denenen değerler ve tercih edilen parametre değerleri Tablo 4.1.'de yer almaktadır.

**Tablo 4.1. Denenen parametre değerleri ve seçilen değerler**

| **Parametreler** | **Denenen Değerler** | **Seçilen Değer** |
|---|---|---|
| Kolonideki toplam arı sayısı | 15, 25, 35 | 25 |
| İşçi Arı Sayısı | 10, 15, 20 | 20 |
| Limit | M/2, M, 2xM | M |
| İterasyon Sayısı | Mx500, Mx1000, Mx1500 | Mx1000 |

Kolonideki toplam arı sayısı, işçi arı ve gözcü arı sayısının toplamı olduğundan denenen değerlerde işçi arıdan geriye kalanlar gözcü arı olmaktadır. Yapılan denemeler sonucunda en uygun parametre değerleri toplam arı sayısı 25, işçi arı sayısı 20, gözcü arı sayısı 5, limit M, iterasyon sayısı da Mx1000 olarak belirlenmiştir. Kaşif arı sayısı da limit değerini aşan besin kaynağı sayısı kadardır.

# SONUÇLAR VE ÖNERİLER

Bu kısımda geliştirilen YAKA literatürde yer alan veri setleriyle sınanmış ve sınama sonuçları daha önce farklı sezgisel tekniklerle elde edilmiş sonuçlarla kıyaslanmıştır. Literatürde yer alan kıyaslama veri setlerine ilave olarak BIST 30 verileriyle de sınama yapılmıştır.

Literatürde bulunan 5 tane test veri seti, Hong Kong'daki Hang Seng, Almanya'daki DAX100, İngiltere'deki FTSE 100, Amerika'daki S&P 100 ve Japonya'daki Nikkei 225 indekslerine ait Mart 1992-Eylül 1997 yılları arasındaki haftalık verileri içermektedir. Bu veri setleri, bir çok optimizasyon problemine ilişkin verilerin bulunduğu bir veri seti kütüphanesinden alınmıştır. http://people.brunel.ac.uk/~mastjjb/jeb/orlib/portinfo.html adresinden bu veri setlerine ulaşılabilmektedir. Veri setlerinde farklı sayıda hisse senetleri bulunmaktadır. Hisse senetleri sayısına ilişkin bilgi Tablo 5.1.'de bulunmaktadır.

**Tablo 5.1. Test verilerine ilişkin hisse senedi sayıları**

| Endeks | Hisse Senedi Sayısı |
|---|---|
| Hang Seng | 31 |
| DAX100 | 85 |
| FTSE 100 | 89 |
| S&P 100 | 98 |
| Nikkei | 225 |

BIST30 veri seti ise endekste işlem gören hisse senetlerinin Ocak 2012-Haziran 2014 tarihleri arasındaki haftalık kapanış fiyatlarını içermektedir. Veriler Recep Tayyip Erdoğan Üniversitesi'nin 31.12.2014 tarihine kadar anlaşmasının bulunduğu www.finnet.com adresinden düzeltilmiş veri şekilde elde edilmiştir. Bu tarihler arasında işlem gören hisse senedi sayısı 24 olarak belirlenmiş ve 6 tane hisse senedi elenmiştir. Bu 24 tane hisse senedinin isimleri Tablo 5.2.'de gösterilmektedir.

**Tablo 5.2. BIST 30 endeksinde işlem gören hisse senetleri**

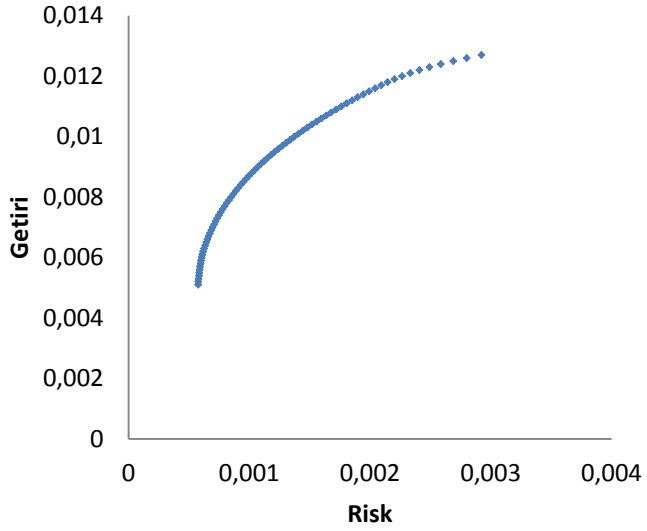
| HİSSE SENETLERİ | | |
|---|---|---|
| Akbank | İş Bankası (C) | T. Halk Bankası |
| Arçelik | Kardemir (D) | Tofaş Oto. Fab. |
| Asya Katılım Bankası | Koç Holding | Turkcell |
| Bim Mağazalar | Koza Altın | Tüpraş |
| Emlak Konut Gmyo | Migros Ticaret | Türk Hava Yolları |
| Enka İnşaat | Petkim | Türk Telekom |
| Ereğli Demir Celik | Sabancı Holding | Vakıflar Bankası |
| Garanti Bankası | Şişe Cam | Yapı ve Kredi Bankası |

Bu 24 hisse senedinin haftalık kapanış fiyatlarından dönemlik getirileri hesaplanmıştır. Elde edilen veriler üzerinden ortalama getiri, standart sapma ve kovaryans matrisi elde edilmiştir. Ayrıca BIST 30 endeksine ait etkin sınır hesaplanmıştır. Elde edilen sonuçlar incelendiğinde 0,0015 ile 0,013 getiri seviyeleri arasında optimal sonuçlar verdiği gözlenmiştir. Bu verilerin sezgisel tekniklerle kıyaslanabilmesi için bu değerler arasındaki nokta çifti sayılarının arttırılması gerekmektedir. Bu sebeple getiri seviyeleri 0,0001 artış yapılarak 113 tane farklı getiri seviyesi ile buna ilişkin risk değerleri bulunmuştur. Fakat bunların ilk 36 tanesi etkin olmadığından uygulamadan çıkarılmıştır. Geriye kalan 77 tane nokta çifti ile etkin sınır çizilmiştir (Şekil 5.1.).

Tüm sonuçlar, K=10 hisse senedi, bu hisse senetlerine ait alt sınır $\varepsilon_j = 0{,}01$ ve üst sınır $\delta_j = 1 \quad (j = 1, \dots, M)$ olarak kabul edilerek bulunmuştur. Etkin sınırın çizilmesinde kullanılacak olan getiri ve riskler, farklı $\lambda$ değerleri ile algoritmanın çalıştırılmasıyla elde edilmiştir. $\Delta\lambda$=0,02 artırılarak 0 ile 1 arasında $\Upsilon$ = 51 tane getiri ve risk değeri bulunmuştur.

Karşılaştırmalar için tüm algoritmalar Windows 7 İşletim sistemi koşturan bir PC'de yürütülmüştür. PC, 2.4 GHz saat vurum sıklığına sahip dört çekirdeklidir ve 4 GB belleği vardır.

**Şekil 5.1. BIST 30 endeksine ilişkin standart etkin sınır**




Bu çalışmada test verilerine ait standart etkin sınır ile sezgisel algoritmalar tarafından bulunan sezgisel etkin sınırlar karşılaştırılmaktadır. Bu karşılaştırma ortalama öklid uzaklığı, yüzde katılım oranı, yüzde getiri hatası, yüzde varyans hatası ve saniye cinsinden hesaplama sürelerine göre yapılmıştır. Tüm veri setleri için 5 sezgisel tekniğin bulduğu sonuçlar Tablo 5.3.'de gösterilmektedir. Tabloda verilen GA, TA, TB ve PSO algoritmalarının sonuçları Cura[1] tarafından elde edilmiştir. Test verilerinin standart etkin sınırı ile tüm sezgisel tekniklerin bulduğu sezgisel etkin sınırların karşılaştırmalı grafikleri Şekil 5.2.-Şekil 5.7.'de gösterilmektedir.

$(r_i^s, v_i^s)(i = 1, \ldots., 2000)$, standart etkin sınırdaki getiri ve varyans çiftlerini temsil ettiğini; $(r_j^h, v_j^h)(j = 1, \ldots., \Upsilon)$, sezgisel etkin sınırdaki getiri ve varyans çiftlerini temsil ettiğini varsayarsak $(r_j^h, v_j^h)$ noktasına en yakın $(r_{i_j}^s, v_{i_j}^s)$ noktası Formül 5.1. yardımıyla bulunur. Burada bulunan $(r_{i_j}^s, v_{i_j}^s)$ noktası ortalama öklid uzaklığının bulunmasında kullanılmaktadır. Ortalama Öklid Uzaklığı $(O\ddot{O}U)$, Formül 5.2. ile hesaplanmaktadır.

[1] Tunçhan Cura, a.g.e., 2008.

**Tablo 5.3. Beş sezgisel tekniğe ait uygulama sonuçları**

| Endeks | H.S.S. | | GA | TA | TB | PSO | YAK |
|---|---|---|---|---|---|---|---|
| Hang Seng | 31 | **Ort. Öklid Uzaklığı** | 7,75E-05 | 7,79E-05 | 7,83E-05 | 9,60E-05 | 1,1491E-04 |
| | | **Katılım Oranı (%)** | 68,62745 | 29,41176 | 13,72549 | 1,96078 | 15,68627 |
| | | **Varyans Hatası (%)** | 1,64626 | 1,65643 | 1,66294 | 2,24212 | 2,70439 |
| | | **Getiri Hatası (%)** | 0,60715 | 0,61066 | 0,62375 | 0,74267 | 0,88867 |
| | | **süre(s)** | 15 | 8 | 9 | 31 | 28 |
| DAX 100 | 85 | **Ort. Öklid Uzaklığı** | 1,49E-04 | 1,62E-04 | 1,53E-04 | 1,76E-04 | 1,8933E-04 |
| | | **KatılımOranı (%)** | 62,7451 | 5,8824 | 9,8039 | 3,9216 | 33,3333 |
| | | **Varyans Hatası (%)** | 7,2180 | 9,0309 | 8,5423 | 6,8588 | 7,2117 |
| | | **Getiri Hatası (%)** | 1,2791 | 1,9078 | 1,2825 | 1,5885 | 1,7164 |
| | | **süre(s)** | 88 | 37 | 46 | 160 | 141 |
| FTSE 100 | 89 | **Ort. Öklid Uzaklığı** | 3,87E-05 | 4,22E-05 | 4,18E-05 | 4,34E-05 | 6,7380E-05 |
| | | **Katılım Oranı (%)** | 60,7843 | 17,6471 | 25,4902 | 17,6471 | 27,4510 |
| | | **Varyans Hatası (%)** | 2,8660 | 4,0123 | 3,8205 | 3,0596 | 4,7625 |
| | | **Getiri Hatası (%)** | 0,3277 | 0,3298 | 0,3304 | 0,3640 | 0,5449 |
| | | **süre(s)** | 93 | 38 | 49 | 169 | 148 |
| S&P 100 | 98 | **Ort. Öklid Uzaklığı** | 8,08E-05 | 7,99E-05 | 8,04E-05 | 1,02E-04 | 1,2754E-04 |
| | | **Katılım Oranı (%)** | 60,7843 | 0,0000 | 5,8824 | 13,7255 | 25,4902 |
| | | **Varyans Hatası (%)** | 3,4802 | 5,7139 | 5,4247 | 3,9099 | 5,1818 |
| | | **Getiri Hatası (%)** | 1,2258 | 0,7125 | 0,8416 | 1,4048 | 1,6434 |
| | | **süre(s)** | 112 | 45 | 58 | 192 | 168 |
| Nikkei | 225 | **Ort. Öklid Uzaklığı** | 2,03E-05 | 1,95E-05 | 1,92E-05 | 3,76E-05 | 4,5621E-05 |
| | | **Katılım Oranı (%)** | 76,4706 | 5,8824 | 9,8039 | 15,6863 | 15,6863 |
| | | **Varyans Hatası (%)** | 1,2056 | 1,2431 | 1,2017 | 2,4274 | 3,0353 |
| | | **Getiri Hatası (%)** | 5,3266 | 0,4207 | 0,4126 | 0,7997 | 0,9693 |
| | | **süre(s)** | 659 | 207 | 256 | 814 | 727 |
| BIST30 | 24 | **Ort. Öklid Uzaklığı** | 6,39E-05 | 6,61E-05 | 6,52E-05 | 7,37E-05 | 7,8094E-05 |
| | | **Katılım Oranı (%)** | 66,6667 | 9,8039 | 13,7255 | 1,9608 | 37,2549 |
| | | **Varyans Hatası (%)** | 2,5498 | 2,6706 | 2,5668 | 3,5190 | 3,8842 |
| | | **Getiri Hatası (%)** | 0,7797 | 0,8026 | 0,7998 | 0,8132 | 0,7538 |
| | | **süre(s)** | 16 | 8 | 9 | 30 | 29 |



122

$$i_j = \underset{i = 1, \dots, 2000}{\arg min} \left( \sqrt{(r_i^s - r_j^h)^2 + (v_i^s - v_j^h)^2} \right) \quad \text{j=1, … , } \Upsilon \tag{5.1.}$$

$$O\ddot{O}U = \left( \sum_{j=1}^{\Upsilon} \sqrt{(r_{i_j}^s - r_j^h)^2 + (v_{i_j}^s - v_j^h)^2} \right) \Big/ \Upsilon \tag{5.2.}$$

Yüzde getiri hatası ($GH$) ve yüzde varyans hatası ($VH$) aşağıdaki formüller yardımıyla bulunmuştur.

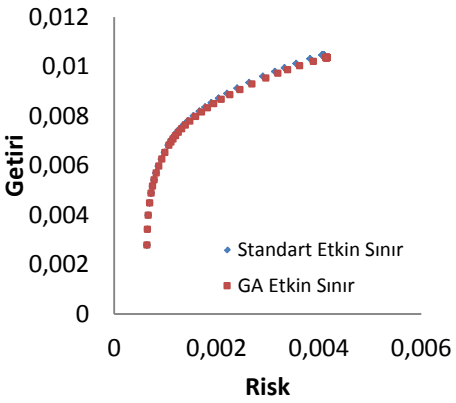
$$GH = \left( \sum_{j=1}^{\Upsilon} 100 \left| r_{i_j}^s - r_j^h \right| / r_j^h \right) x \frac{1}{\Upsilon} \tag{5.3.}$$

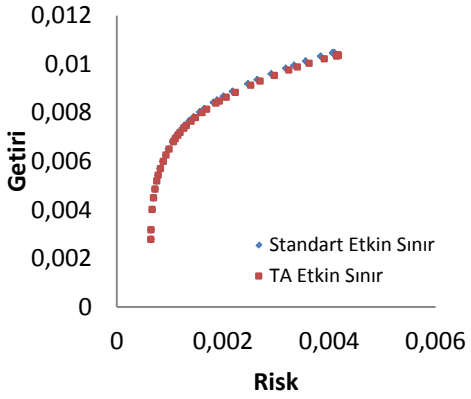
$$VH = \left( \sum_{j=1}^{\Upsilon} 100 \left| v_{i_j}^s - v_j^h \right| / v_j^h \right) x \frac{1}{\Upsilon} \tag{5.4.}$$

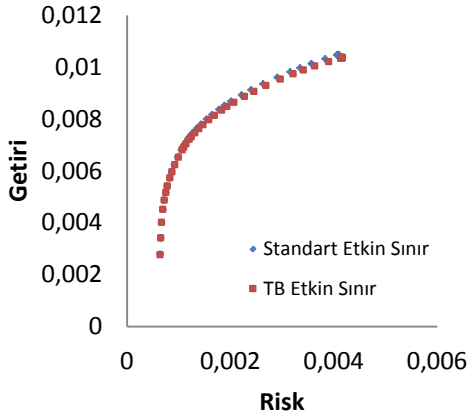
Katılım oranları bulunurken ilk önce tüm sezgisel tekniklerin elde ettikleri sonuçlar ile her bir λ değerine ilişkin KKOV'nin amaç fonksiyonu değeri hesaplanmıştır. Bu 5 sezgisel teknik içinden en küçük amaç fonksiyonu değerine sahip sezgisel teknik seçilmiştir. Söz konusu amaç fonksiyonuna ilişkin getiri ve varyans değerleri bir tabloda saklanmıştır. Elde edilen $\Upsilon$ = 51 tane veri çifti ile birleştirilmiş etkin sınırı çizilmiştir (Şekil 5.8.). Sezgisel tekniğin toplam seçilme sayısı, elde edilen toplam veri çifti sayısına bölünüp 100 ile çarpılarak katılım oranları bulunmuştur. Birleştirilmiş etkin sınır çizme fikri Fernández ve Gómez tarafından ortaya atılmıştır[2].

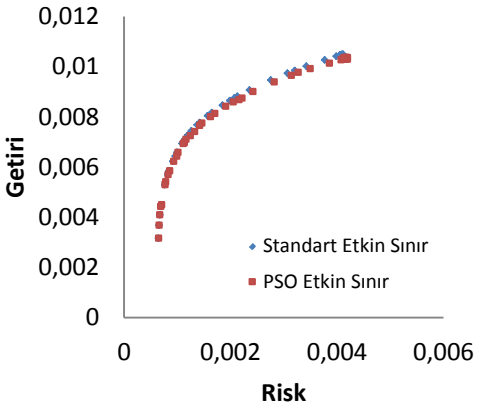
[2] Alberto Fernández, Sergio Gómez, a.g.e., 2007.

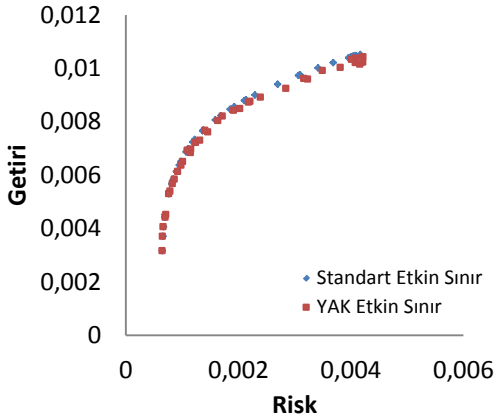
**Şekil 5.2. Hang Seng endeksi için sezgisel etkin sınırlar**



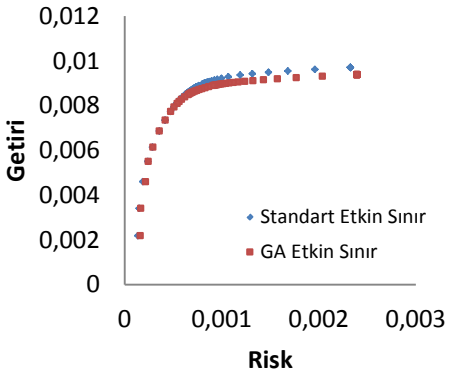



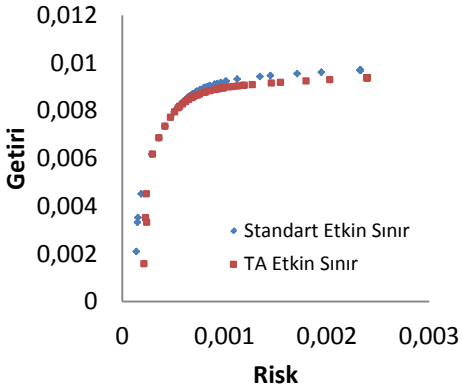



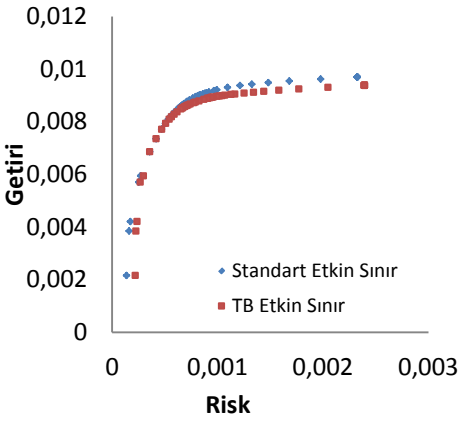



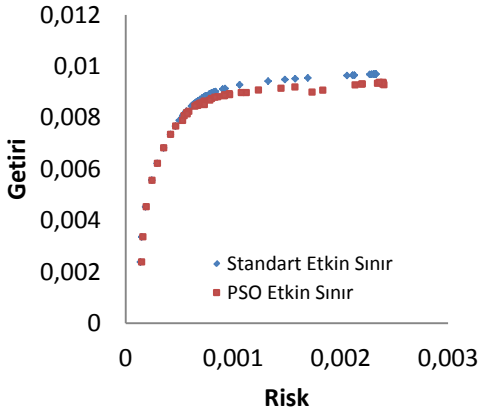



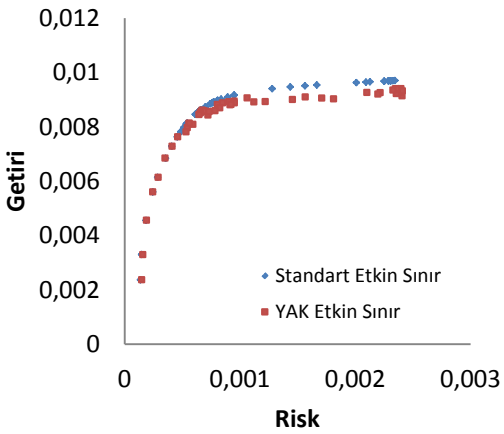
**Şekil 5.3. DAX100 endeksi için sezgisel etkin sınırlar**



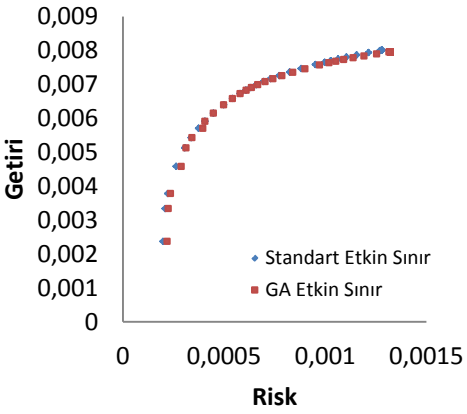



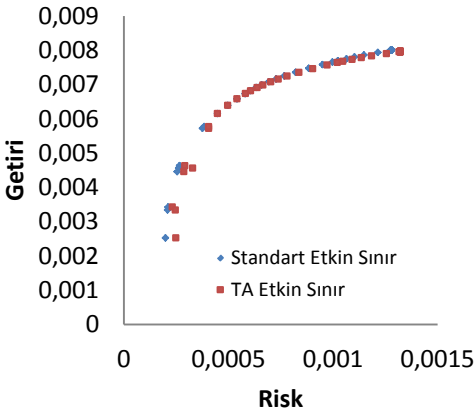



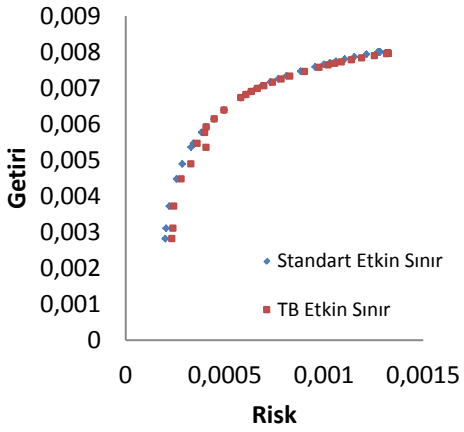



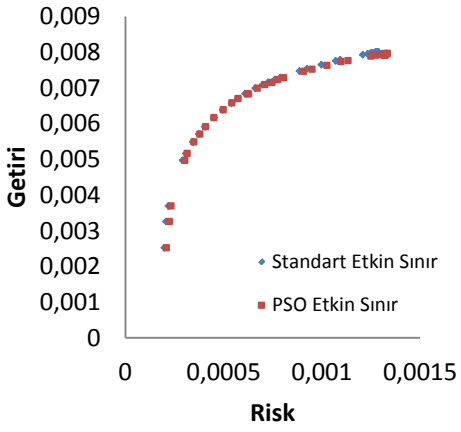



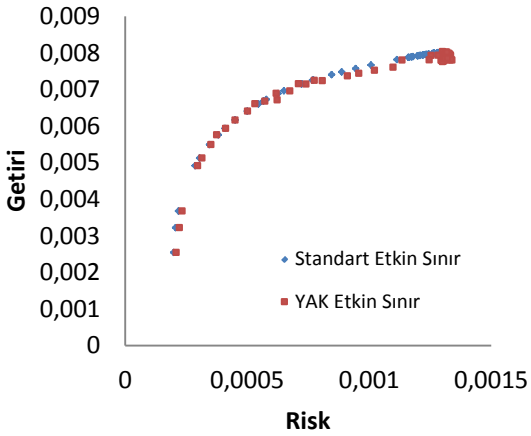
**Şekil 5.4. FTSE 100 endeksi için sezgisel etkin sınırlar**



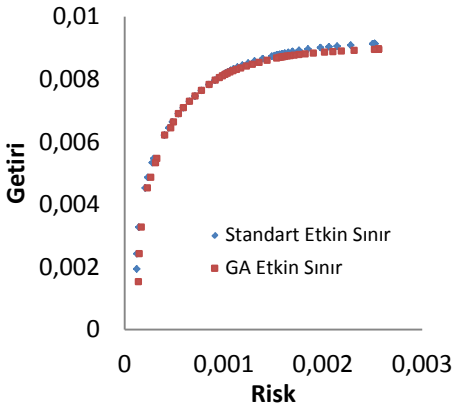



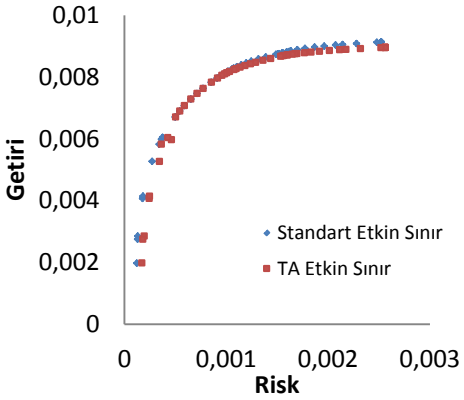



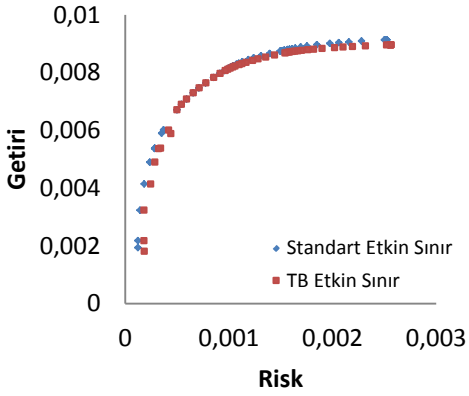



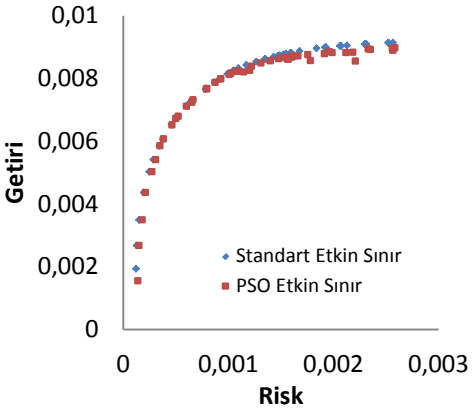



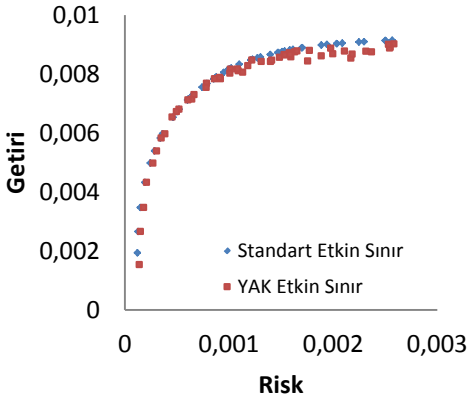
**Şekil 5.5. S&P 100 endeksi için sezgisel etkin sınırlar**



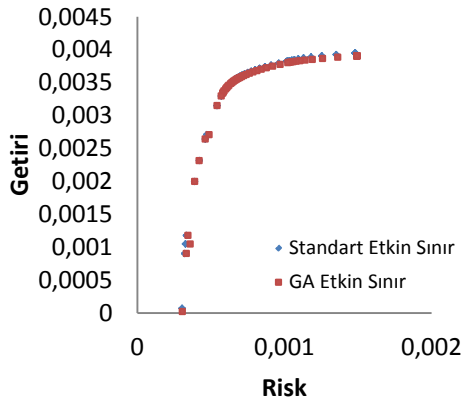



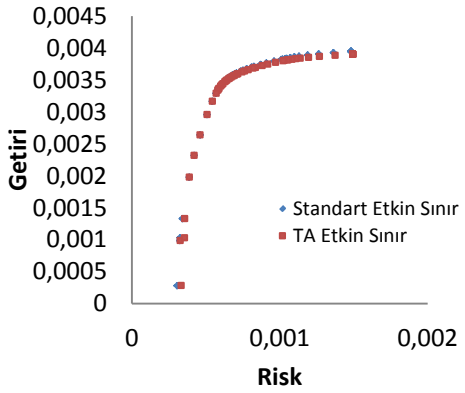



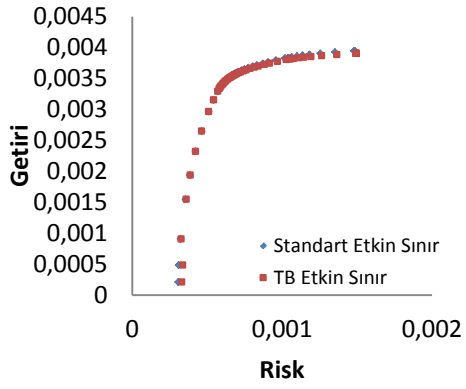



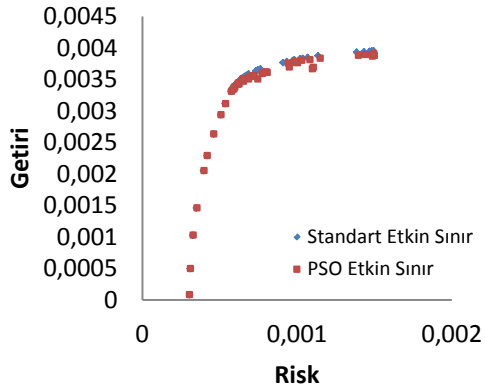



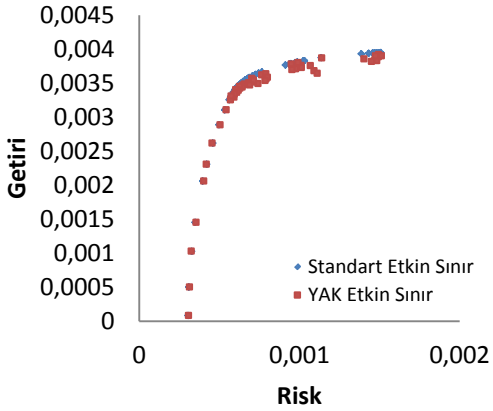
**Şekil 5.6. Nikkei endeksi için sezgisel etkin sınırlar**



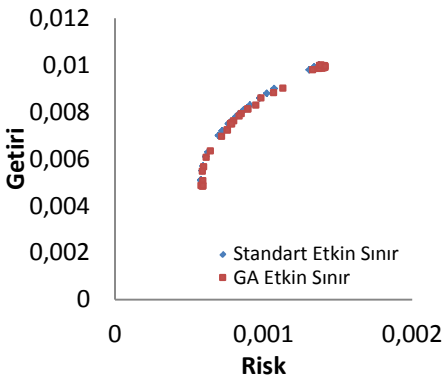



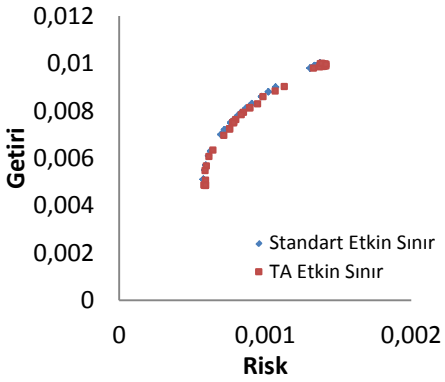



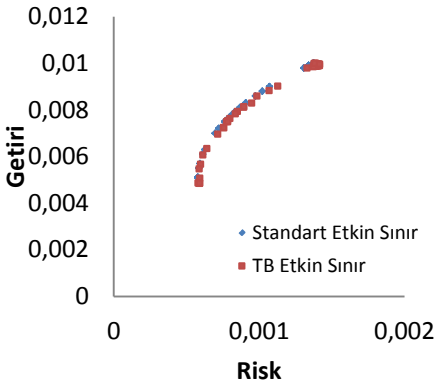



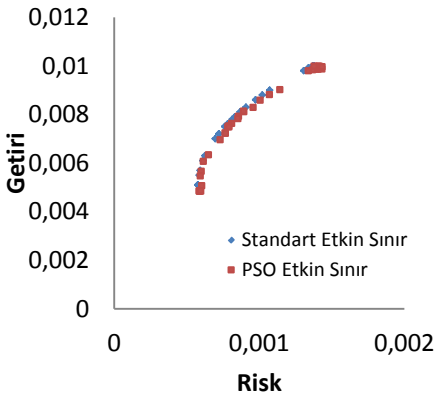



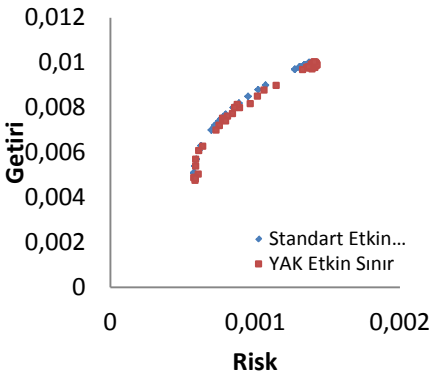
**Şekil 5.7. BİST 30 endeksi için sezgisel etkin sınırlar**



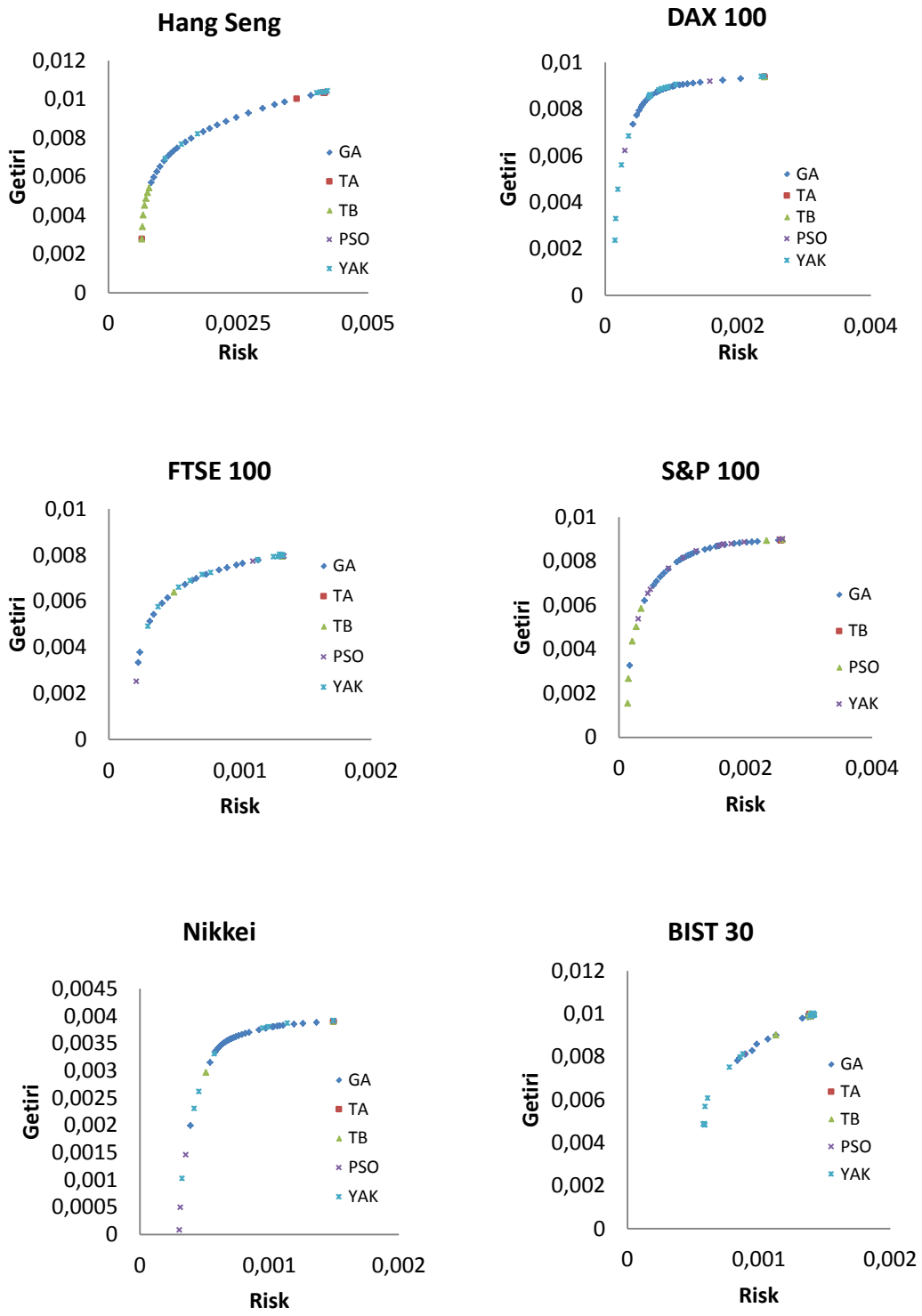



**Şekil 5.8. Problemlere ilişkin birleştirilmiş etkin sınırlar**



Tablo 5.3. incelendiğinde görülüyor ki YAKA, KKOV modeli için diğer sezgisel yöntemlerle rekabet edebilecek düzeyde sonuçlar üretebilmektedir. Bu konuda en iyi olan GA yöntemine oldukça yakın sonuçlar üretmiştir. Özellikle katılım oranları açısından değerlendirildiğinde YAK, GA'dan sonra en iyi yöntem olarak karşımıza çıkmaktadır. Özellikle PSO' nun önemli ölçüde katılım oranını düştüğü ve bunun ağırlıklı olarak YAKA' ya kaydığı görülmektedir. Bunun sebebi YAKA' nın PSO yöntemiyle aynı düzenleme yöntemini kullandığı için benzer fakat daha iyi sonuçlar bulması olarak yorumlanabilir.

YAKA'nın elde ettiği sezgisel etkin sınırlar incelendiğinde FTSE 100 endeksinin etkin sınırı için her orandaki getiri seviyelerinde standart etkin sınıra yakın sonuçlar ürettiği gözlenmiştir. Diğer problem türlerinde ise riskin düşük olduğu durumlarda standart etkin sınıra yakın değerler üretebilirken risk arttıkça standart etkin sınırdan uzaklaştığı gözlenmiştir.

Sonuç olarak YAKA, KKOV modeli için ideal sayılabilecek bir yöntem olup bu konuda çalışacak olan araştırmacılar ve yöneticiler için uygulanması kolay bir yöntem olacaktır.

Gelecek çalışmalar için düzenleme fonksiyonu daha da geliştirilebilir, çeşitli sezgisel teknikler ve arama algoritmaları ile melez sistemler oluşturularak problem tekrar çözülebilir.

# KAYNAKÇA


Ajith, Abraham
Grosan, Crina
Ramos, Vitorino
"Swarm Intelligence in Data Mining", Grosan,Crina, Abraham, Ajith and Chis, Monica, **Swarm Intelligence in Data Mining**, Springer, 2006.

Akay, Bahriye
"Nümerik Optimizasyon Problemlerinde Yapay Arı Kolonisi (Artificial Bee Colony) Algoritmasının Performans Analizi", Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Yayımlanmamış Doktora Tezi, 2009.

Akay, Bahriye
"A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding", **Applied Soft Computing**, 13, 3066-3091, 2013.

Alba, Enrique
Cotta, Carlos
"Evolutionary Algorithms", In Olariu, S., Zomaya, A.Y. (Ed.) **Handbook of Bioinspired Algorithms and Applications**, Taylor&Francis Group, LLC, 2006.

Alizadegan, Amir
Asady, Babak
Ahmadpour, Mohammad
"Two Modified versions of artificial bee colony algorithm", **Applied Mathematics and Computation**, 225, 601-609, 2013

Alvarado-Iniesta, A.
Garcia-Alcaraz, J.L.
Rodriguez-Borbon, M.I.
Maldonado, A.
"Optimization of the material flow in a manufacturing plant by use of artificiai bee colony algorithm", **Expert Systems with Applications**, 40, 4785-4790, 2013.

Antoniou, Andreas
Lu, Wu-Sheng
**Practical Optimization Algorithms and Engineering Application**, SpringerScience+BusinessMedia, New York, 2007.

Arora, Jasbir S.
**Guide to Structural Optimization**, American Society of Civil Engineers, 1997.

Aydın, İlhan
Karaköse, Mehmet
Akın, Erhan
"Genetik Algoritma Kullanan Yapay Bağışık Sistem Tabanlı Arıza Teşhis Modeli", **DEÜ Mühendislik fakültesi Fen ve Mühendislik Dergisi,** 11(31), 57-72, 2009.

Bäck, Thomas,
**Evolutionary Algorithms in Theory and Practice**, Oxford University Press, New York, First Edition, 1996.

Bäck, Thomas
Fogel, David B.
Michalewicz, Zbigniew

**Evolutionary Computation 1 Basic Algorithms and Operators**, Bristol and Philadelphia: Institute of Physics Publishing, 2000a.

Bäck, Thomas.,
Fogel, David B.,
Michalewicz, Zbigniew

**Evolutionary Computation 2 Advanced Algorithms and Operators**, Bristol and Philadelphia: Institute of Physics Publishing, 2000b.

Bagley, J. D.

**The behavior of adaptive systems which employ genetic and correlation algorithms,** Ph.D. dissertation, University of Michigan, Ann Arbor, 1967.

Banzhaf, W.,
Nordin, P.,
Keller, R.E.,
Francone, F.D

**Genetic Programming: An Introduction,** Morgan Kaufman Publishers, Inc, San Francisco, California, 1998.

Baştürk, Alper
Akay, Rüştü

"Performance analysis of the coarse-grained parallel model of the artificial bee colony algorithm", **Information Sciences**,253, 34-55, 2013.

Baştürk, Bahriye
Karaboğa, Derviş

"An artificial bee colony (abc) algorithm for numeric function optimization", **In: IEEE swarm intelligence symposium** Indianapolis,IN,USA, s. 49-53, 2006.

Bayğın, Mehmet
Karaköse, Mehmet

**"**Adaptif Yapay Bağışık Sistem Tabanlı Grup Asansör Kontrol Algoritması"**, Elektrik- Elektronik ve Bilgisayar Sempozyumu,** 205-210, 2011.

Beekman, Madeleine,
Gilchrist, A.L,
Duncan, M.,
Sumpter David J.T.,

"What makes a honeybee scout?", **Behav Ecol Sociobiol,** 61, 985–995, 2007

Belal, M.,
Gaber, J.,
El-Sayed, H.,
Almojel, A.

"Swarm Intelligence", **In Olariu, S., Zomaya, A.Y. (Ed.) Handbook of Bioinspired Algorithms and Applications**, Taylor&Francis Group, LLC, 2006.

Bell, John E.
McMullen, Patrick R.

"Ant Colony Optimization Techniques for The Vehicle Routing Problem", **Advanced Engineering Informatics**, 18, 41-48, 2004.

Beni, G. "The concept of Cellular Robotic Systems", **In Proceedings of the IEEE International Symposiumon Inteligent Systems**, 57-62, IEEE Press, Piscataway,NJ,1988.

Blum, C., Roli, A., Dorigo, M. "HC–ACO: The hyper-cube framework for Ant Colony Optimization," **in Proceedings of MIC'2001 –Metaheuristics International Conference**, vol. 2, Porto, Portugal, pp. 399–403, 2001.

Blum, Christian "Ant Colony Optimization: Introduction and Recent Trends", **Physics of Life Reviews**, 2, 353-373, 2005.

Blum, Christian, Merkle, Daniel **Swarm Intelligence Introduction and Application**, Springer Verlang, Berlin, 2008.

Bolak, Mehmet **Sermaye Piyasası: Menkul Kıymetler ve Portföy Analizi**, 4. Baskı, Beta Yayıncılık, 2001,

Bonabeau, Eric Dorigo, Marco Theraulaz, Guy **Swarm Intelligence: From Natural to Artifical Systems**, Oxford University Press, 1999.

Bullnheimer, B. Hartl, R.F. Strauss, C. **"A new rank based version of the Ant System — a computational study,"** Institute of Management Science, University of Viena, Tech. Rep., 1997.

Bullnheimer, B. Hartl, R.F., Strauß, C. "Applying The Ant System to The Vehicle Routing Problem", **2nd International Conference on Metaheuristics – MIC97**, 1997. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.7946 (08.08.2014)

Canbaş, Serpil Doğukanlı, Hatice **Finansal Pazarlar, Finansal Kurumlar ve Sermaye Pazarı Analizleri**, Genişletilmiş 3. Baskı, Beta Yayıncılık, 2001.

Ceylan, Ali Korkmaz, Turhan **Borsada Uygulamalı Portföy Yönetimi**, 3. Baskı, Ekin Kitabevi, Bursa, 1998

Ceylan, Ali Korkmaz, Turhan **Sermaye Piyasası ve Menkul Değer Analizi**, Ekin Kitabevi, Bursa, 2000.

Chang, T.J. Meade, N. Beasley, J.E. Sharaiha, Y.M. "Heuristics for cardinality constrained portfolio optimization", **Computers and Operation Research**, 27, 1271-1302, 2000.

Chen, Wei — "An Artificial Bee Colony Algorithm for Uncertain Portfolio Selection", **The Scientific World Journal**, Volume 2014, Article ID 578182, 12 pages, 2014.

Chen, Wei
Ma, Hui
Yang, Yiping
Sun, Mengrong — "Application of Artificial Bee Colony Algorithm to Portfolio Adjustment Problem with Transaction Costs", **Journal of Applied Mathematics**, Volume 2014, Article ID 192868, 12 pages, 2014.

Chun-Feng, Wang
Kui, Liu
Pei-Ping, Shen — "Hybrid Artificial Bee Colony Algorithm and Particle Swarm Search for Global Optimization", **Mathematical Problem in Engineering**, Vol. 2014, Article ID: 832949, 8 pages, 2014.

Coley, David. A. — **An Introduction to Genetic Algorithms for Scientists and Engineers**, London: World Scientific Publishing, 1999.

Cordon, O.
De Viana, I. F.,
Herrera, F.,
Moreno, L., — "A new ACO model integrating evolutionary computation concepts: The best-worst Ant System," in Abstract proceedings of ANTS 2000 – **From Ant Colonies to Artificial Ants: Second International Workshop on Ant Algorithms,** M. Dorigo et al., Eds. IRIDIA, Universit´e Libre de Bruxelles, Belgium, pp. 22–29, 2000.

Cura, Tunçhan — **Uygulamalı Modern Sezgisel Teknikler ve Uygulamalar**, Papatya Yayıncılık, İstanbul, 2008.

Cura, Tunçhan — "Particle swarm optimization approach to portfolio optimization", **Nonlinear Analysis: Real World Applications**, 10(4), 2396-2406, 2009.

Cura, Tunçhan — "An artificial bee colony algorithm approach for the team orienteering problem with time Windows", **Computers&Industrial Engineering**, 74, 270-290, 2014.

Çakar, Kemal — **Genetik Algoritmalar Yardımıyla Acil Servis İstasyonu Yerleşiminin Optimizasyonu**, Atatürk Üniversitesi, Sosyal Bilimler Fakültesi, Yayınlanmamış Doktora Tezi, 2009.

Dalkılıç, Gökhan
Türkmen, Fatih — Karınca Kolonisi Optimizasyonu, **1. Ulusal Yüksek Performanslı Bilişim Sempozyumu**, Kocaeli, 2003.

Davis, L. — **Handbook of Genetic Algorithms,** New York: Van Nostrand Reinhold, 1991.

De Castro, L.N. Timmis, J. — "A Novel approach to pattern recognition". In L.Alonso, J.Corchado and C.Fyfe, editors, **Artificial Neural Networks in pattern Recognition**, University of Paisley: 67-84, 2002.

De Castro, L.N. Von Zuben, F. J. — **Artificial Immune Systems: Part 1- Basic Theory and Applications,** Technical Report, TR-DCA 01/99, 1999.

De Castro, L.N. Von Zuben F.J., — "The Clonal Selection Algorithm with Engineering Applications", **GECCO 2000**, Las Vegas, Nevada, USA, July 8, 36-42, 2000.

De Castro, L.N. Von Zuben F.J., — "Learning and optimization Using the Clonal Selection Principle", **In the Special Issue on Artificial Immune Systems of the Journal IEEE Transactions on Evolutionary Computation**, June 6(3), 239-251, 2002.

Dengiz, Berna Altıparmak, Fulya — "Genetik Algoritmalara Genel Bir Bakış", **Endüstri Mühendisliği Dergisi**, 9(3), 3-14, 1998.

Dorigo, Marco — "Optimization, learning and natural algorithms (in italian)," Ph.D. dissertation, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1992.

Dorigo, Marco Birattari, Mauro Stützle, Thomas — **"Ant Colony Optmization: Artificial Ants as a Computational Intelligence Technique"**, IRIDIA-Technical Report Series, TR/IRIDIA/2006-023, September 2006.

Dorigo, Marco Di Caro, Gianni — "The Ant Colony Optimization metaheuristic," **in New Ideas in Optimization**, D. Corne et al., Eds. McGrawHill, London, UK, pp. 11–32, 1999.

Dorigo, Marco Di Caro, Gianni Gambardella, Luca M. — Ant Algorithms for Discrete Optimization, **Artificial Life**, 5, 137-172, 1999.

Dorigo, Marco, Gambardella, L.M. — "Ant colonies for the traveling salesman problem," **BioSystems**, vol. 43, no. 2, pp. 73–81, 1997.

Dorigo, Marco
Maniezzo, Vittorio
Colorni, Alberto
**"Positive feedback as a search strategy"**, Technical Report, 91-016, June 1991.

Dornhaus, Anna
Klügl, F.
Puppe, F.
Tautz, J.
"Task selection in honeybees—Experiments using multi-agent simulation" **In Proceedings of GWAL'98**, Bochum, September 18-19, 1998

Elmas, Çetin
**Yapay Zeka Uygulamaları**, Ankara: Seçkin Yayıncılık, 2007.

Elton, Edwin J.
Gruber, Martin J.
Brown, Stephen J.
Goetzmann, W. N.
**Modern Portfolio Theory and Investment Analysis**, 6. Baskı, USA, John Wiley&Sons, Inc., 2003.

Emel, Gül Gökay
Taşkın, Çağatan
"Genetik Algoritmalar ve Uygulama Alanları", **Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi**, XXI(1), 129-152, 2002.

Engelbrecht, Andries
**Computational Intelligence**, John Wiley & Sons Inc., Second Edition, 2007.

Engin, Orhan
Döyen, Alper
"Artificial Immune Systems and Applications in Industrial Problems", **G.U. Journal of Science,** 17(1), 71-84, 2004**.**

Fernández,Alberto
Gómez, Sergio
"Portfolio selection using neural networks", **Computers&Operations Research**, 34, 1177-1191, 2007

Flecther, Roger
**Practical Methods of Optimization**, Second Edition, John Wiley&Sons, 2000.

Fogel, L.J.
Owens, A.J.
Walsh M.J.
**Artificial Intelligence through Simulated Evolution**, Wiley, New York, 1966.

Fraser, A. S.
"Simulation of genetic systems by automatic digital computers", **Australian Journal of Biological Science,** 10,484–499, 1957.

Gambardella L. M. Dorigo M., "Ant-Q: A reinforcement learning approach to the traveling salesman problem," **in Proceedings of the Twelfth International Conference on Machine Learning (ML-95)**, A. Prieditis and S. Russell, Eds. Morgan Kaufmann Publishers, pp. 252–260, 1995.

Gen, Mitsuo Cheng, Runwei **Genetic Algorithms and Engineering Optimization**, John Wiley & Sons, Inc., 2000.

Goldberg, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning.Reading,** MA: Addison-Wesley**,** 1989.

Haupt, Randy L. Haupt, Sue A. **Practical Genetic Algorithms**, John Wiley & Sons Inc., Second Edition, 2004

Holland J.H. **Nonlinear Enviroment Permitting Efficient Adaptation, Computer and Information Science II**, New York, 1967.

Holland, J.H. **Adaptation in Natural and Artificial Systems,** Cambridge, MA: The MIT Press, 1975.

Imanian, Nafiseh Shiri, Mohammad Ebrahim Moradi, Parham "Velocity based bee colony algorithm for high dimensional continuous optimization problems" **Engineering Applications of Artificial Intelligence**, 36, 148-163, 2014.

Kalaycı, Can B. Gupta, Surendra M. "Artificial bee colony algorithm for solving sequence-dependent disassembly line balancing problem", **Expert Systems with Applications**, 40, 7231-7241, 2013.

Kang Fei, Li, Junjie Li, Haojin "Artificial bee colony algorithm and pattern search hybridized for global optimization", **Applied Soft Computing**, 13, 1781-1791, 2013.

Karaboğa, Derviş **Yapay Zeka Optimizasyon Algoritmaları: Tabu Araştırma, Işıl İşlem, Genetik, Karınca Koloni, Bağışıklık ve Diferansiyel Gelişim Algoritmaları**, 1. Baskı, İstanbul: Atlas Yayın Dağıtım, 2004.

Karaboğa, Derviş "**An Idea Based On Honey Bee Swarm For Numerical Optimization"**, TR-06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

Karaboğa, Derviş — **Yapay Zeka Optimizasyon Algoritmaları**, Genişletilmiş 2. Baskı, Nobel Yayın Dağıtım, 2011.

Karaboğa, Derviş
Akay, Bahriye — "A survey: algorithms simulating bee swarm intelligence", **Artif Intell Rev**, 31, 61-85, 2009.

Karaboğa, Derviş
Baştürk, Bahriye — "A Powerful and Efficent Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm", **J. Glob. Optim**., 39, 459-471, 2007a.

Karaboğa, Derviş
Baştürk, Bahriye — "Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems", **Proceedings of the 12th International Fuzzy Systems Association World Congress on Foundations of Fuzzy Logic and Soft computing**, Springer, Berlin, 789–798, 2007b.

Karaboğa, Derviş
Baştürk, Bahriye — "On the Performance of Artificial Bee Colony (ABC) Algorithm", **Applied Soft Computing**, 8, 687-697, 2008.

Karaboğa, Derviş
Görkemli, Beyza — "A quick artificial bee colony (qABC) algorithm and its performance on optimization problems", **Applied Soft Computing**, 23, 227-238, 2014.

Karaboğa, D.,
Görkemli, B.,
Öztürk, C.,
Karaboğa, N., — **A comprehensive survey: artificial bee colony (ABC)algorithm and applications**, Artificial Intelligence Review, Springer Science+Business Media B.V., 2012

Karan, Mehmet B. — **Yatırım Analizi ve Portföy Yönetimi**, 3. Baskı, Gazi Kitabevi, Ankara, 2011

Karslı, Muharrem — **Sermaye Piyasası, Borsa, Menkul Kıymetler**, 4. Baskı, İrfan Yayımcılık, İstanbul, 1989

Kennedy, J.,
Eberhart, R.C., — "Particle Swarm Optimization", **Proc. IEEE int'l Conf. On Neural Networks**, vol IV, 1942-1948, 1995.

Kennedy, James
Eberhart, Russel, C.
Shi, Yuhui — **Swarm Intelligence**, Morgan Kaufmann Publishers,2001.

Kinnear, Kenneth E. — **Advances in Genetic Programming**, MIT Press, 1994.

Konuralp, Gürel — **Sermaye Piyasaları: Analizler, Kuramlar ve Portföy Yönetimi**, 2. Baskı, Alfa Yayınları, 2005.

Koza, J.R. — **Genetic Programming: On the Programming of Computers by Means of Natural Selection,** MA:MIT Press, Cambridge, 1992.

Liu Wen — "A multistrategy optimization improved artificial bee colony algorithm", **The Scientific World Journal**, vol. 2014, Article ID 129483, 10 p., http://dx.doi.org/10.1155/2014/129483, 2014.

Maniezzo, V. — "Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem," INFORMS **Journal on Computing**, 11(4), 358–369, 1999.

Marinaki, M., Marinakis, Y., Zopounidis, C. — "Honey Bees Mating Optimization algorithm for financial classification problems", **Applied Soft Computing**, 10, 806-812, 2010.

Markowitz, Harry — "Portfolio Selection", **Journal of Finance**, 7(1),77-91, Mart 1952.

Markowitz, Harry — **Portfolio Selection- Efficient Diversification of Investments**, New York, John Wiley&Sons, 1959

Milan, Tuba Nebojsa, Bacanin — "Artificial Bee Colony Algorithm Hybridized with Firefly Algorithm for Cardinality Constrained Mean-Variance Portfolio Selection Problem", **Applied Mathematics&Information Sciences**, 8(6), 2831-2844, 2014.

Mitchell, Melanie — **An Introduction to Genetic Algorithms**, 5. Baskı, Massachusetts: A Bradford Book The MIT Press, 1999.

Mullen, R.J. Monekosso, D. Barman, S. Remagnino, P. — A Review of Ant Algorithms, **Expert Systems with Applications**, 36, 9608-9617, 2009.

Nabiyev, Vasif Vagifoğlu — **Yapay Zeka: Problemler Yöntemler Algoritmalar**, 1. Baskı, Ankara: Seçkin Yayınevi, 2003.

Negnevitsky, Michael — **Artificial Intelligence A guide to Intelligent Systems**, Second Edition, Addison-Wesley, 2005.

Nikolić, Miloš
Teodorović, Dušan
"Empirical study of the Bee Colony Optimization (BCO) algorithm", **Expert Systems with Applications**, 40, 4609-4620, 2013.

Nocedal, Jorge
Wright, Stephen J.
**Numerical Optimization**, Springer-Verlag, New York, 1999.

Özdağlar, Davut
Benzeden, Ertuğrul,
Kahraman, Murat A.
"Kompleks Su Dağıtım Şebekelerinin Genetik Algoritma ile Optimizasyonu", **İMO Teknik Dergi**, 253, 3851-3867, 2006.

Pelikan, M.,
Sastry, K.,
Cantu-Paz, E.,
**Scalable Optimization via Probabilistic Modeling,** Springer-Verlag, Berlin, 2006

Rechenberg, I.
**Cybernetic Solution Path of an Experimental Problem,** Royal Aircraft Establishment Library Translation 1122, 1965.

Rutkowski, Leszek
**Computational Intelligence: Methods and Techniques,** SpringerVerlag, Heidelberg**,** 2005.

Sakawa, Masatoshi
**Genetic Algorithms and Fuzzy Multiobjective Optimization,** Kluwer Academic Publisher, Massachusetts, 2002.

Saif, Ullah
Guan, Zailin
Liu, Weiqi
Wang, Baoxi
Zhang, Chaoyong
"Multi-objective artificial bee colony algorithm for simultaneous Sequencing and balancing of mixed model assembly line", **Int. J. Manuf. Technol**., 75, 1809-1827, 2014.

Sarker, R.,
Mohammadian, M.,
Yao, X.,
**Evolutionary Optimization,** Kluwer Academic Publishers, Dordrecht, 2003.

Seeley, T.
**Honeybee ecology: a study of adaptation in social life**, Princeton University Press, Princeton, 1985

Sharpe, William
"A Simplified Model for Portfolio Analysis", **Management Science**, 9(2), 277-293, 1963.

Sivanandam, S.N.
Deepa, S.N
**Introduction to Genetic Algorithm**, Springer-Verlag, Berlin, 2008.

Spall, James C.
**Introduction to Stochastic Search and Optimization,** John Wiley & Sons Inc., New Jersey, 2003.

Storn, Rainer, Price, Kenneth — "Differential Evolution: A Simple and Efficent Heuristic Strategy for Global Optimization over Continuous Spaces", **Journal of Global Optimization,** 11: 341-359, 1997.

Stützle, T. Hoos, H.H. — "**Improving the Ant System: A detailed report on the MAX–MIN Ant System,"** FG Intellektik, FB Informatik, TU Darmstadt, Germany, Tech. Rep. AIDA–96–12, Aug. 1996.

Sumathi, S., Hamsapriya, T., Surekha, P., — **Evolutionary Intelligence**, Springer-Verlag, Coimbatore, India, 2008.

Taha, Hamdy A., — **Yöneylem Araştırması**, Çev. Baray,Alp Ş. ve Esnaf,Şakir, 6. Baskıdan Çeviri, Literatür Yayıncılık, İstanbul, 2000.

Taşkın, Çağatan Emel, Gül Gökay — **Sayısal Yöntemlerde Genetik Algoritmalar**, Bursa: Alfa Aktüel, 2009.

Tereshko, V. Loengarov, A. — "Collective decision making in honey-bee foraging Dynamics". **Computing and Informaton Systems**, 9(3), 1-7, 2005.

Tereshko, Valery — "Reaction-diffusion model of a honeybee colony's foraging behaviour", M. Schoenauer, et al, Eds., **Parallel Problem Solving from Nature VI** (Lecture Notes in Computer Science,Vol. 1917) Springer-Verlag: Berlin, p. 807-816, 2000.

Ulucan, Aydın — **Portföy Optimizasyonu: Kuadratik Programlama Tabanlı Modelleme**, Siyasal Kitabevi, 2004

Von Frisch, Karl — **The dancing bees: an account of the life and senses of honey bee**, Harcourt, Brace&World, 1954.

Von Frisch, Karl Lindauer, M. — "The "language" and orientation of the honey bee" **Annual Review of Entomology**, 1, 45–58, 1956

Weise, T. — **Global Optimization Algorithms: Theory and Application**, www.it-weise.de, Second Edition, 2009.

Winston, Wayne L. — **Operations Research: Applications and Algorithms,** Forth Edition, Brooks/Cole, 2004.

Wismer, David A.
Chattergy, R.,
**Introduction to Nonlinear Optimization: A Problem Solving Approach**, North-Holland Series, New York, 1978.

Xiang, Wan-li
An, Mei-qing
"An efficient and robust artifical bee colony algorithm for numerical optimization", **Computers&Operation Research**, 40, 1256-1265, 2013.

Xiang, Wanli
Ma, Shoufeng
An, Meiqing
"hABCDE: A hybrid evolutionary algorithm based on artificial bee colony algorithm and differential evolution", **Applied Mathematics and Computation**, 238, 370-386, 2014.

Yao, Baozhen
Hu, Ping
Zhang,Mingheng
Wang, Shuang
"Artificial bee colony algorithm with scanning strateji for the periodic vehicle routing problem", **Simulation: Transactions of the Society for Modeling and Simulation International**, 89(6), 762-770, 2013.

Zhang, Rui
Song, Shiji
Wu, Cheng
"A hybrid artificial bee colony algorithm for the job shop scheduling problem", **Int. J. Production Economics**, 141, 167-178, 2013.

Zhang, Shuzhu
Lee, C.K.M.
Choy, K.L.
Ho, William
Ip, W.H.
"Design and development of a hybrid artificial bee colony algorithm for the enviromental vehicle routing problem", **Transportation Research Part D**, 31, 85-99, 2014.

Zhang, Xiaoxia
Tang, Lixin
"A New Hybrid Ant Colony Optimization Algorithm for The Vehicle Routing Problem", **Pattern Recognition Letters**, 30, 848-855, 2009.

# ÖZGEÇMİŞ

Burcu KARTAL, 1984 yılında Trabzon'da doğdu. İlk ve orta öğrenimini Trabzon'da tamamladı. 2006 yılında Karadeniz Teknik Üniversitesi Fen Edebiyat Fakültesi İstatistik ve Bilgisayar Bilimleri ve çift ana dal yaptığı İktisadi ve İdari Bilimleri Fakültesi İşletme Bölümlerinden mezun oldu. Aynı yıl Karadeniz Teknik Üniversitesi Sosyal Bilimler Enstitüsü İşletme Anabilim Dalında yüksek lisans eğitimine başladı. 2010 yılında yüksek lisans eğitimini tamamladı. Aynı yıl İstanbul Üniversitesi Sosyal Bilimler Enstitüsü İşletme Anabilim Dalı Sayısal Yöntemler Bilim Dalında doktora eğitime başladı. KARTAL, 2009 yılından beri Recep Tayyip Erdoğan Üniversitesi İktisadi ve İdari Bilimler Fakültesi İşletme Bölümü Sayısal Yöntemler Anabilim Dalında araştırma görevlisi olarak çalışmaktadır. İngilizce bilmektedir. Evli ve bir çocuk annesidir.