



**EGE ÜNİVERSİTESİ**

**DOKTORA TEZİ**

**ETMEN TABANLI BENZETİMDE PARAMETRE AYARLAMA**

**Rabia KORKMAZ TAN**

**Tez Danışmanı: Dr. Öğretim Üyesi Şebnem BORA**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Sunuş Tarihi : 30.04.2018**

**Bornova-İZMİR**

**2018**



**EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ**

**(DOKTORA TEZİ)**

**ETMEN TABANLI BENZETİMDE PARAMETRE  
AYARLAMA**

**Rabia KORKMAZ TAN**

**Tez Danışmanı: Dr. Öğretim Üyesi Şebnem BORA**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Bilim Dalı Kodu: 619.01.00**

**Sunuş Tarihi: 30.04.2018**

**Bornova-İZMİR**

**2018**



**Rabia KORKMAZ TAN** tarafından **DOKTORA** tezi olarak sunulan “**Etmen Tabanlı Benzetimde Parametre Ayarlama**” başlıklı bu çalışma E.Ü. Lisansüstü Eğitim ve Öğretim Yönetmeliği ile E.Ü. Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi'nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve **30.04.2018** tarihinde yapılan tez savunma sınavında aday **oybirliği/oyçokluğu** ile başarılı bulunmuştur.

**Jüri Üyeleri:**


**İmza**

**Jüri Başkanı:** Dr. Öğretim Üyesi Şebnem BORA ..... 

**Raportör Üye:** Prof. Dr. Oğuz DİKENELLİ ..... 

**Üye:** Prof. Dr. Yalçın ÇEBİ ..... 

**Üye:** Prof. Dr. Aybars UĞUR ..... 

**Üye:** Dr. Öğretim Üyesi Tolga AYAV ..... 



## EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

### ETİK KURALLARA UYGUNLUK BEYANI

E.Ü. Lisansüstü Eğitim ve Öğretim Yönetmeliğinin ilgili hükümleri uyarınca ~~Yüksek Lisans Tezi~~ / Doktora Tezi olarak sunduğum “Etmem Tabanlı Benzetimlerde Parametre Ayarlama” başlıklı bu tezin kendi çalışmam olduğunu, sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi, tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını, bu tezin herhangi bir bölümünü bu üniversite veya diğer bir üniversitede başka bir tez çalışması içinde sunmadığımı, bu tezin planlanmasından yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

25 /05 / 2018

  
Rabia KORKMAZ TAN



## ÖZET

### ETMEN TABANLI BENZETİMLERDE PARAMETRE AYARLAMA

KORKMAZ TAN, Rabia

Doktora Tezi, Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Dr. Öğretim Üyesi Şebnem BORA

Nisan 2018, 183 Sayfa

Bu çalışma, etmen tabanlı model ve benzetim ortamında modellenen karmaşık sistemlerin parametre ayarlama problemine çözüm getirmek için yapılmıştır. Modellenen karmaşık sistemler, gerçeğine uygun ve istenen davranışı sergileyebilmek için iyi bir parametre setine ihtiyaç duyarlar. Bu işlem, büyük parametre uzaylarına sahip olan karmaşık sistemler için çözülmesi gereken bir optimizasyon problemidir.

Bu tez çalışması, etmen tabanlı model ve benzetimler için 3 aşamalı melez bir parametre ayarlama yaklaşımı sunmaktadır, ilk aşamada problem tanımlanmış. İkinci aşamada, bir meta-sezgisel algoritma olan Genetik Algoritma, Parçacık Sürü Optimizasyonu, Yapay Arı Kolonisi ve Ateş Böceği Algoritmaları kullanılarak parametre ayarlama işlemi gerçekleştirilmiştir. Yine ikinci aşamada kullanılan meta-sezgisel algoritmaların sahip olduğu kritik parametre değerlerinin, algoritmaların performansına doğrudan etki ettiği bu tez çalışmasında gözlemlenmiştir. Ayrıca, çevrimiçi uyarlanabilir parametre ayarlama (ÇUPA) tekniği F-Race algoritması kullanılarak geliştirilmiş ve bahsedilen meta-sezgisel algoritmalara uygulanıp uyarlanabilir MSA'lar geliştirilmiştir. Geliştirilen bu algoritmalar da model parametrelerini ayarlama işlemi için kullanılmıştır. 3. aşama kontrol aşaması olup optimizasyon sonucu elde edilen algoritmaya ait kritik parametre değerleri ve modele ait bulunan parametre değerleri manuel girilerek test edilme işlemi yapılmaktadır. Geliştirilen bu yaklaşım, 3 farklı modelde ve 4 farklı nümerik test fonksiyonunda test edilmiş problemlerdeki algoritma başarıları karşılaştırılmıştır.

**Anahtar Sözcükler:** Etmen tabanlı model ve benzetim, Parametre Ayarlama, Kritik Parametrelerin Ayarlanması, Meta-sezgisel Algoritmalar, Genetik Algoritmalar, Parçacık sürü Optimizasyonu Algoritması, Yapay Arı Kolonisi Algoritması, Ateş Böceği Algoritması, Optimizasyon.



## ABSTRACT

### PARAMETER TUNING IN AGENT BASED SIMULATION

KORKMAZ TAN, Rabia

Ph.D. in Computer Engineering

Supervisor: Asst. Prof. Dr. Şebnem BORA

April 2018, 183 pages

This study was carried out to solve the parameter tuning problem of complex systems that are modelled in an agent-based modelling and simulation environment. Complex systems that are modelled need a good set of parameters to be able to demonstrate the realistic and desired behaviour. This process is an optimization problem that must be solved for complex systems with large parameter spaces.

This study presents a three step hybrid parameter tuning approach for agent-based models and simulations. In the first step, the problem is defined; in the second step, parameter tuning process is carried out using Genetic Algorithms, Particle Swarm Optimization, Artificial Bee Colony and Firefly Algorithms, which are meta-heuristic algorithms. Again, meta-heuristic algorithms used in the second stage have been observed in this thesis that critical parameter values directly affect the performance of the algorithms. Therefore, adaptive meta-heuristic optimization algorithms which are adapted to the problem are developed using the online adaptive parameter control technique. These algorithms are also used to adjust the model parameters. In the third step, which is the control phase, the algorithm parameters obtained by the online adaptive parameter control method and the parameter values of the model from the used meta-heuristic algorithms are manually provided and tested. This approach is compared with the algorithm performance of the problems tested in 3 different models and 4 different numerical test functions.

**Keywords:** Agent Based Modeling and Simulation, Parameter Tuning, Critical Parameters Tuning, Meta-Heuristic Algorithms, Genetic Algorithm, Particle Swarm Optimization Algorithm, Artificial Bee Colonies Algorithm, Firefly Algorithm, Optimization.



## TEŞEKKÜR

Tez çalışması süresince danışmanlığımı yapan, deneyimleriyle ve bilgi birikimiyle bana sürekli yol gösteren, sabrı ve hoşgörüsüyle desteğini hiçbir zaman esirgemeyen Dr. Öğretim Üyesi Şebnem BORA'ya en içten teşekkürlerimi sunarım.

Ayrıca sürekli yanımda olan ve desteklerini esirgemeyen meslektaşlarım Arş. Gör. Sinem ÇINAROĞLU ve Arş. Gör. Sema BODUR'a gönülden teşekkür ederim.

Her zaman yanımda olan, eğitim hayatım boyunca benden hiçbir desteği esirgemeyen aileme, tez çalışmam süresince ve hayatım boyunca beni her konuda destekleyen abim Selami KORKMAZ'a, eşim Metin TAN'a ve manevi varlığı ile bana güç veren sevgili oğlum Alaz Furkan TAN'a teşekkür ederim.



**İÇİNDEKİLER**

	<u>Sayfa</u>
ÖZET .....	vii
ABSTRACT .....	ix
TEŞEKKÜR .....	xi
ŞEKİLLER DİZİNİ .....	xx
ÇİZELGELER DİZİNİ .....	xxv
ALGORİTMALAR DİZİNİ .....	xxviii
SİMGELER VE KISALTMALAR DİZİNİ .....	xxx
1. GİRİŞ .....	1
2. ETMEN TABANLI BENZETİM VE MODELLEME .....	6
2.1. Etmenin Tanımı .....	6
2.2. Etmen Özellikleri .....	9
2.3. Çoklu Etmen Sistemler .....	10
2.3.1. Çoklu Etmen Sistemlerinde Bireysel Etmenler .....	12
2.3.2. Çoklu Etmen Sistemlerde İletişim .....	13
2.4. Benzetim ve Modelleme .....	14
2.4.1. Benzetimlerin Avantajları ve Dezavantajları .....	16
2.5. Etmen Tabanlı Benzetim ve Modelleme .....	16

**İÇİNDEKİLER (Devam)**

	Sayfa
2.6. Karmaşık Sistemler.....	18
2.7. Etmen Tabanlı Sistemlerin Değerlendirilmesi .....	21
3. LİTERATÜRDE YER ALAN ÇALIŞMALAR.....	22
3.1. Parametre Ayarlama .....	23
3.2. Parametre Ayarlama Kullarılan Teknikler .....	25
3.2.1. İlk Parametre Setini Belirleyen Metotları.....	25
3.2.2. Model-tabanlı metotlar .....	26
3.2.3. Gözleme Dayalı Metotları .....	28
3.2.4. Arama Tabanlı Metotlar .....	29
3.2.5. Etmen Tabanlı Metotlar .....	30
3.2.6. Meta-Sezgisel Metotlar.....	32
3.2.7. Modele Özgü Geliştirilen Diğer Parametre Ayarlama Algoritmaları .....	43
3.3. Literatürde Yapılan Çalışmaların Değerlendirilmesi .....	45
4. ÇALIŞMANIN TEORİSİ VE ARAÇLARI .....	49
4.1. Meta-Sezgisel Algoritmalar.....	50
4.1.1. Sezgisel ve Meta-Sezgisel Kavramları .....	50
4.1.2. Meta-Sezgisel Algoritmalar (MSA) .....	51

**İÇİNDEKİLER (Devam)**

	Sayfa
4.1.3. Meta-Sezgisel Algoritmaların Kritik Parametre Değerlerinin Ayarlanması	55
4.2. Etmen Tabanlı Benzetim ve Modelleme Araçları.....	58
4.2.1. SWARM.....	58
4.2.2. MASON .....	59
4.2.3. NETLOGO .....	59
4.2.4. REPAST .....	59
4.2.5. JACK.....	60
4.2.6. DIAS.....	60
5. KULLANILAN META-SEZGİSEL ALGORİTMALAR.....	62
5.1. Kullanılan Meta-Sezgisel Algoritmaların Ortak Kavramları .....	63
5.2. Genetik Algoritma (GA) .....	63
5.2.1. Genetik Algoritma Kavramları ve Operatörleri .....	64
5.2.2. Genetik Algoritmanın Çalışması.....	71
5.3. Parçacık Sürü Optimizasyonu (PSO) Algoritması .....	74
5.3.1. PSO'da Kullanılan Kavramlar .....	75
5.3.2. PSO Algoritmasının Çalışması .....	78
5.4. Yapay Arı Kolonisi (Artificial Bee Colony-ABC) Algoritması .....	80

## İÇİNDEKİLER (Devam)

	Sayfa
5.4.1. Standart ABC Algoritması.....	81
5.4.2. Modifiye Yapay Arı Kolonisi (M-ABC) Algoritması.....	82
5.4.3. Yapay Arı Kolonisi Algoritmasının Çalışması.....	82
5.5. Ateş Böceği Algoritması (Firefly Algorithms-FA) .....	86
5.5.1. Ateş Böceği Algoritmasının Çalışması.....	87
6. GELİŞTİRİLEN ADAPTİF META-SEZGİSEL ALGORİTMALAR .....	90
6.1. Kritik Parametrelerin Uyarlanabilir Parametre Ayarı .....	90
6.2. Çevrim İçi Uyarlanabilir Parametre Ayarlama Algoritması.....	93
6.3. Uyarlanabilir Genetik Algoritma (Adaptive Genetic Algorithm (AGA)) ....	95
6.4. Uyarlanabilir PSO Algoritması (Adaptive Particle Swarm Optimization (APSO)).....	96
6.5. Uyarlanabilir ABC Algoritması (Adaptive Artificial Bee Colony (AABC)) .....	98
6.6. Uyarlanabilir Ateş Böceği Algoritması (Adaptive Firefly Algorithm (AFA)) .....	100
7. GELİŞTİRİLEN PARAMETRE AYARLAMA ARACININ KULLANIMI .....	102
7.1. Parametre Ayarlama Aracının Gereksinimleri .....	102
7.2. Geliştirilen Parametre Ayarlama Aracının Mimarisi.....	105

**İÇİNDEKİLER (Devam)**

	Sayfa
7.2.1. Ortak Paket Sınıfları.....	105
7.2.2. Algoritma Paketi Sınıfları .....	106
7.3. Geliştirilen Aracın Arayüz Detayları .....	108
7.3.1. Test işlemleri için Geliştirilen Ek Arayüz.....	112
8. KULLANILAN MODELLER VE DENEYSEL ÇALIŞMA SONUÇLARI .....	116
8.1. Model ve Kullanılan Kaynak Detayları .....	117
8.2. Av-Avcı (Predator-Prey) Modeli .....	119
8.2.2. ETMB Ortamında Av-Avcı Modeli .....	121
8.2.3. Av-Avcı Problemi için Uygunluk Fonksiyonu .....	126
8.2.4. Av-Avcı Modeli Benzetim Sonuçları.....	128
8.3. Sekiz Vezir (Eight Queens) Modeli .....	141
8.3.1. Sekiz Vezir Problemi İçin Uygunluk Fonksiyonu .....	143
8.3.2. Sekiz Vezir Modeli Benzetim Sonuçları.....	144
8.4. Zombi-İnsan (Human Zombie) Modeli.....	150
8.4.1. Zombi Problemi İçin Uygunluk Fonksiyonu .....	151
8.4.2. Zombi-İnsan (Human Zombie) Modeli Benzetim Sonuçları .....	152
8.5. Optimizasyon İşlemi için Kullanılan Nümerik Test Fonksiyonları .....	155

## İÇİNDEKİLER (Devam)

Sayfa

9. SONUÇ VE TARTIŞMA.....161

KAYNAKLAR DİZİNİ.....165





## ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
2.1 Karmaşık Sistem Modeli (Gündoğmuş, 2014) .....	19
3.1 Parametre Ayarlama Süreci Akış Diyagramı (Korkmaz Tan ve Bora, 2017b)	25
5.1 Rulet Tekerleği Yöntemi.....	67
5.2 Turnuva Yöntemi .....	68
5.3 Mutasyon Operatörü Kullanım Yöntemleri (Bolat et al., 2004) .....	70
5.4 Parçacığın Arama Uzayındaki Hareketi (Hosseini et al., 2014) .....	77
5.5 Ters Kare Kanunu .....	87
7.1 Parametre Ayarlama Arayüzünün Modelle Gösterimi.....	111
7.2 Geliştirilen Parametre Ayarlama Aracı Arayüzü .....	112
7.3 Modele Ait Bulunan En iyi Parametre Kümesinin Test Edildiği Arayüz.....	114
7.4 Modele Ait Bulunan En iyi Parametre Kümesinin Test Edildiği Arayüzün Modelle Gösterimi.....	115
7.5 Modele ait bulunan en iyi parametre değerlerinin testi sonucu elde edilen grafik .....	115
8.1 Av-Avcı Popülasyonunun Zaman İçindeki Değişimi (Kara, 2006).....	120
8.2 Av-Avcı Modelinin Faz Diyagramı (Kara, 2006).....	121
8.3 Av-Avcı Modeli Sınıf Diyagramı .....	123
8.4 Rasgele Oluşturulan İlk Popülasyon .....	125

## ŞEKİLLER DİZİNİ (Devam)

<u>Şekil</u>	<u>Sayfa</u>
8.5 Son İterasyon ve Elde Edilen Parametre Değerleri .....	126
8.6 Random GA ile Av Avcı Modelinde Parametre Ayarlama .....	134
8.7 Random PSO ile Av Avcı Modelinde Parametre Ayarlama .....	134
8.8 Random ABC ile Av Avcı Modelinde Parametre Ayarlama.....	135
8.9 Random FA ile Av Avcı Modelinde Parametre Ayarlama.....	135
8.10 AGA ile Av Avcı Modelinde Parametre Ayarlama.....	136
8.11 APSO ile Av Avcı Modelinde Parametre Ayarlama .....	136
8.12 AABC ile Av Avcı Modelinde Parametre Ayarlama .....	137
8.13 AFA ile Av Avcı Modelinde Parametre Ayarlama .....	137
8.14 Sonra AGA ile Av Avcı Modelinde Parametre Ayarlama .....	138
8.15 Sonra APSO ile Av Avcı Modelinde Parametre Ayarlama.....	138
8.16 Sonra AABC ile Av Avcı Modelinde Parametre Ayarlama.....	139
8.17 Sonra AAFA ile Av Avcı Modelinde Parametre Ayarlama .....	139
8.18 Sonra AGA ile Elde Edilen En İyi Av-Avcı Model Parametre Değerlerinin Av-Avcı Modeline Manuel Girilerek Modelin Çalıştırılması Sonucu Elde edilen Grafik .....	140
8.19 Sonra APSO ile Elde Edilen En İyi Av-Avcı Model Parametre Değerlerinin Av-Avcı Modeline Manuel Girilerek Modelin Çalıştırılması Sonucu Elde edilen Grafik .....	140

## ŞEKİLLER DİZİNİ (Devam)

<u>Şekil</u>	<u>Sayfa</u>
8.20 Sonra AABC ile Elde Edilen En İyi Av-Avcı Model Parametre Değerlerinin Av-Avcı Modeline Manuel Girilerek Modelin Çalıştırılması Sonucu Elde edilen Grafik.....	141
8.21 Sonra AFA ile Elde Edilen En İyi Av-Avcı Model Parametre Değerlerinin Av-Avcı Modeline Manuel Girilerek Modelin Çalıştırılması Sonucu Elde edilen Grafik.....	141
8.22 Sekiz Vezir Problemi İçin Örnek Çözüm Gösterimi .....	142
8.23 Tahtanın x ve y koordinatları ve her bir karenin indis numarası .....	143
8.24 Vezir hareketleri.....	143
8.25 Random GA ile Sekiz Vezir Modelinde Parametre Ayarlama .....	147
8.26 Random PSO ile Sekiz Vezir Modelinde Parametre Ayarlama.....	147
8.27 Random ABC ile Sekiz Vezir Modelinde Parametre Ayarlama.....	147
8.28 Random FA ile Sekiz Vezir Modelinde Parametre Ayarlama.....	147
8.29 AGA ile Sekiz Vezir Modelinde Parametre Ayarlama.....	148
8.30 APSO ile Sekiz Vezir Modelinde Parametre Ayarlama .....	148
8.31 AABC ile Sekiz Vezir Modelinde Parametre Ayarlama .....	148
8.32 AFA ile Sekiz Vezir Modelinde Parametre Ayarlama.....	148
8.33 Sonra AGA ile Sekiz Vezir Modelinde Parametre Ayarlama.....	149
8.34 Sonra APSO ile Sekiz Vezir Modelinde Parametre Ayarlama .....	149

**ŞEKİLLER DİZİNİ (Devam)**

<u>Şekil</u>	<u>Sayfa</u>
8.35 Sonra AABC ile Sekiz Vezir Modelinde Parametre Ayarlama.....	149
8.36 Sonra AFA ile Sekiz Vezir Modelinde Parametre Ayarlama.....	149
8.37 İnsan Zombi Modeli Arayüzü.....	151
8.38 Sonra GA ile İnsan Zombi Modelinde Parametre Ayarlama.....	154
8.39 Sonra PSO ile İnsan Zombi Modelinde Parametre Ayarlama .....	154
8.40 Sonra ABC ile İnsan Zombi Modelinde Parametre Ayarlama .....	155
8.41 Sonra FA ile İnsan Zombi Modelinde Parametre Ayarlama .....	155



## ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
3.1 Parametre Ayarlama Algoritmalarının Özellikleri (Korkmaz Tan ve Bora, 2017a).....	46
6.1 Algoritmalara Ait Kritik Parametre Değişkenleri.....	92
7.1 Parametre Ayarlama Aracının Gereksinimleri .....	103
izelge 8.1 Deneyler İçin Kullanılan PC Detayları .....	117
8.2 Tüm Algoritmalar İçin Kullanılan Ortak Parametreler .....	118
8.3 Algoritmalarda Kullanılan Parametre Değerleri.....	119
8.4 ETMB Ortamında Modellenen Av Avcı Modeline Ait Parametreler .....	123
8.5 Çizelge 8.6'deki Satırların-Kullanılan Algoritmaların Açıklaması.....	129
8.6 Etmen Tabanlı Model parametrelerinin ayarlama işleminde kullanılan algoritmalarından elde edilen sonuçlar.....	132
8.7 Algoritmaların en iyi uygunluk değerleri .....	133
8.8 Etmen Tabanlı Model ve Benzetimde Sekiz Vezir probleminde parametrelerinin ayarlama işleminde kullanılan algoritmalarından elde edilen sonuçlar .....	146
8.9 Etmen Tabanlı Model ve Benzetimde İnsan-Zombi probleminde parametrelerinin ayarlama işleminde kullanılan algoritmalarından elde edilen sonuçlar .....	153
8.10 Kullanılan Nümerik Test Fonksiyonları .....	156
8.11 GA, PSO, ABC, FA Algoritmalarının Nümerik Test Fonksiyonlarındaki Optimizasyon Sonuçları .....	157

**ÇİZELGELER DİZİNİ (Devam)**ÇizelgeSayfa

8.12 Numerik Test Fonksiyonların Optimizasyon İşlemlerinde Kullanılan Algoritmaların Kritik Parametre Değerleri.....	158
8.13 AGA, APSO, AABC, AFA Algoritmalarının Numerik Test Fonksiyonlarındaki Optimizasyon Sonuçları .....	160





**ALGORİTMALAR DİZİNİ**

<u>Algoritma</u>	<u>Sayfa</u>
5.1 Genetik Algoritma.....	74
5.2 Parçacık Sürü Optimizasyonu Algoritması.....	80
5.3 Yapay Arı Kolonisi Algoritması.....	85
5.4 Ateş Böceği Algoritması.....	89
6.1 Çevrimiçi Uyarlanabilir Parametre Ayarlama Algoritması.....	94
6.2 Uyarlanabilir Genetik Algoritma.....	96
6.3 Uyarlanabilir Parçacık Sürü Optimizasyonu Algoritması.....	97
6.4 Uyarlanabilir Yapay Arı Kolonisi Algoritması.....	99
6.5 Uyarlanabilir Ateş Böceği Algoritması.....	101



**SİMGELER VE KISALTMALAR DİZİNİ**

<u>Kısaltmalar</u>	<u>Açıklama</u>
AABC	(tr.) Uyarlanabilir Yapay Arı Kolonisi (ing.) Adaptive Artificial Bee Colony
ABC	(tr.) Yapay Arı Kolonisi (ing.) Artificial Bee Colony
ACO	(tr.) Karınca Kolonisi Algoritması (ing.) Ant Colony Algorithm
ADO	(tr.) Uyarlanabilir Atomik Parçalara Ayırma (ing.) Adaptative Dichotomic Optimization
AFA	(tr.) Uyarlanabilir Ateş Böceği Algoritması (ing.) Adaptive Firefly Algorithm
AGA	(tr.) Uyarlanabilir Genetik Algoritma (ing.) Adaptive Genetic Algorithm
ANN	(tr.) Yapay Sinir Ağları (ing.) Artificial Neural Network
APSO	(tr.) Uyarlanabilir Parçacık Sürü Optimizasyonu (ing.) Adaptive Particle Swarm Optimization
APTM	(tr.) Parametre Ayarlama Modülü (ing.) Automatic Parameter Tuning Module
AVT	(tr.) Uyarlanabilir Değer İzleyici (ing.) Adaptive Value Tracker
BBO	(tr.) Biyocoğrafya Tabanlı Optimizasyon (ing.) Biogeography Based Optimization
CA	Haberleşme etmenleri ayrı bir kümede tutulur

**SİMGELER VE KISALTMALAR DİZİNİ (Devam)**

<u>Kısaltmalar</u>	<u>Açıklama</u>
CH	Sadece insan katılımcıları bulunduğu küme
CM	Etmenler ve insan davranışlarının tutulduğu küme
CMA	(tr.) Kovaryans Matrisi Adaptasyonu
CMA-ES	(tr.) Kovaryans Matrisi Adaptasyonu - Evrim Stratejisi
ÇES	(tr.) Çoklu Etmen Sistem (ing.) Multi Agent System
ÇUPA	(tr.) Çevrimiçi Uyarlanabilir Parametre Ayarlama (ing.) Online Adaptive Parameter Tuning
DOO	(tr.) Doğrusal Olmayan Otoregresif
DoE	(tr.) Deney Tasarımı (ing.) Design of Experiments
DVM	(tr.) Destek Vektör Makineleri
EA	(tr.) Evrim Algoritması (ing.) Evolution Algorithm
ETMB	(tr.) Etmen Tabanlı Modelleme ve Benzeti (ing.) Agent Based Modeling and Simulation
FA	(tr.) Ateş Böceği Algoritması (ing.) Firefly Algorithm
GA	(tr.) Genetik Algoritma (ing.) Genetic Algorithm
GOA	(tr.) Girdap Optimizasyon Algoritması

**SİMGELER VE KISALTMALAR DİZİNİ (Devam)**

<u>Kısaltmalar</u>	<u>Açıklama</u>
HIS	(tr.) Yaşam Form Endeksi
İBGY	(tr.) İleri Beslemeli Geri Yayılımlı
İKGY	(tr.) İleri Kademeli Geri Yayılımlı
LRPCA	(ing.) Local Region Principal Components Analysis
LM	(tr.) Levenberg-Marquardt
M-ABC	(tr.) Modifiye Yapay Arı Kolonisi (ing.) Modified Artificial Bee Colony
MSA	(tr.) Meta Sezgisel Algoritma (ing.) Meta Heuristik Algorithm
PAMAS	(tr.) Çoklu Etmen Sistemde Parametre (ing.) Parameter Multi Agent System
PSO	(tr.) Parçacık Sürü Optimizasyonu (ing.) Particle Swarm Optimization
QoS	(tr.) Hizmet kalitesi (ing.) Quality of Service
REVAC	(tr.) Parametrenin İlişki Tahmini ve Değer Kalibrasyonu (ing.) Relavance Estimation and Value Calibration of Parameter
SPO	(tr.) Sıralı Parametre Optimizasyon
SVM	(tr.) Destek Vektör Makineleri (ing.) Support Vector Machines

**SİMGELER VE KISALTMALAR DİZİNİ (Devam)**

<u>Kısaltmalar</u>	<u>Açıklama</u>
SIV	(tr.) Uygunluk İndeksi Değişkeni
SKM	(tr.) Bilgilerin tutulduğu Resmi Bilgi Modülü (ing.) State Knowledge Module





## 1. GİRİŞ

Etmen tabanlı model ve benzetim (ETMB), karmaşık sistemlerin modellenmesi için son zamanlarda sıklıkla kullanılan bir yöntem olup (Macal, and North, 2009) çok geniş uygulama alanlarına sahiptir (Epstein, 2007; Tesfatsion and Judd, 2006; Cioppa et al., 2004; Grimm et al., 2005; Balmer et al., 2005). Ancak karmaşık sistemlerin genel yapısı ve literatürde yapılan karmaşık sistem benzetim çalışmaları incelendiğinde, gerçek sistemi tam anlamıyla yansıtacak benzetim modelleri oluşturmak oldukça zor olup ETMB'ler içinde önemli bir problemdir (Macal, and North, 2009).

Doğal olarak, bu alandaki araştırmacıların olası gerçek sistemi taklit edebilecek zenginlikte bir model oluşturabilmesi için sistem hakkında varsayımlarda bulunması gereklidir. Gerçek sistemi temsil edebilecek uygun bir modelin aynı zamanda benzetim ortamlarında gerçekleştirilebilir olması gereklidir. Böylece gerçek sistemin davranışlarını benzetim ortamında taklit edebilecek ve beklenen davranışların ortaya çıkmasını sağlayacak düzeyde minimum gerçek sistem özelliklerini içeren bir model oluşturulabilecektir. Ancak bu şekilde bir tasarım süreci beraberinde ciddi problemleri ortaya çıkarır. Bu problemlerden biri de parametre ayarı (tuning) olarak karşımıza çıkmaktadır. Her etmenin gerçek sisteme benzetilmesi için ortama, mevcut duruma ve diğer etmenlerin davranışlarına bağlı olarak sürekli farklı davranışlar sergilemesi ve kendini organize etmesi beklenmektedir. Her bir etmenin bu durumu gerçekleştirmesi için sahip oldukları bir dizi parametre mevcuttur etmen davranışları bu parametrelerle karakterize edilmektedir (Gürcan et al., 2014). Son zamanlardaki bazı çalışmalar ETMB'lerin parametrelerinin ayarlanması problemini ortaya koymaktadır (Bonjean et al., 2009; Montagna and Roli 2009; Gardelliet al., 2009; Terano, 2007; Fehler et al., 2006). Parametrelerin otomatik olarak ayarlanması için bir takım araçlara gereksinim duyulmaktadır. Bundan dolayı bu tez çalışması, ETMB ortamında modellenen karmaşık sistemlerin sahip olduğu parametre uzayından en uygun parametre setini bularak modellenen sistemin gerçek sistemi yansıtmasını sağlayan aracın geliştirilmesi amaçlanmıştır. Bu tez çalışmasında geliştirilen araç, ETMB'lerde büyük öneme sahip parametre ayarlama işlemini gerçekleştirmekle birlikte geliştirilen yöntem probleme özgü çözümler sunmaktadır. Geliştirilen bu araç birçok yöntemin aynı araç içinde bağımsız kullanımına olanak sağlamaktadır. Ayrıca geliştirilen probleme uyarlanabilen yöntemlerin de kullanımı birçok farklı probleme çözüm üretmesi bakımından aracı önemli hale getirmektedir.

Karmaşık sistemlerin modellendiği ETMB'lerin parametre ayarlama problemini çözmek için klasik optimizasyon teknikleri yeterli değildir. Optimizasyon problemlerinin çözümünde kullanılan Meta-sezgisel algoritmaların (MSA) özellikle bu tarz karmaşık problemlerin çözümünde kabul edilebilir doğrulukta sonuçların daha kısa sürede elde edilebildiği incelenen çalışmalarda kanıtlanmıştır (Schwefel, 1995; Ergün ve Aydoğan, 2015; Calvez and Hutzler 2007)

MSA, optimizasyon problemleri için aday çözümleri iteratif bir şekilde iyileştirerek çözüme ulaşan algoritmalarıdır. MSA her zaman optimum çözüme ulaşması imkânsızdır, fakat kısa sürede optimuma yakın çözümü bulabilmektedir. MSA optimizasyon problemlerinde arama işlemine de yön veren teknikleri içermektedir. Bu algoritmalar yerel optimuma takılmayı önleyecek birçok yapıya sahiptir. MSA kullanıldığında her bir algoritmanın kendine özgü bir takım probleme uyarlanması gereken kritik parametre değişkenlerini içinde barındırırlar. Bu kritik parametre değerleri ve algoritmanın optimum ya da optimuma yakın değerleri bulabilmesinde kritik öneme sahiptir. Bu parametre değerlerinin ayarlanması MSA için oldukça önemlidir ve bununla ilgili bazı çalışmalar mevcuttur (Eiben et al., 1999; Eiben and Smith, 2003).

Bu tez çalışmasında MSA'lara ait kritik parametre değişkenlerinin ayarlanması gerekliliği çözülmesi gereken ikinci problem olarak ele alınmıştır. Çünkü problem için en uygun çözümün elde edilmesi yani probleme özgü çözümlerin bulunması, kullanılan algoritmanın kritik parametre değerlerine bağlıdır. MSA'ların problemin çözümüne uygun kritik parametre değerleri alması algoritmanın performansını doğrudan etkilemekte ve en uygun çözüme ulaşmasını sağlamaktadır. MSA'lar problemden bağımsız geliştirilen algoritmalarıdır. Bundan dolayı bazı problemlerde başarılı olan bir MSA algoritması başka problemin çözümünde aynı başarıyı yakalayamadığı bu tez çalışmasında kanıtlanan sonuçlar arasındadır. Bundan dolayı algoritma kullanıldığı probleme uyarlanabilir olmalıdır. Her bir MSA algoritmasının sahip olduğu kritik parametre değişkenleri farklı problemler için farklı değerler almaktadır. Bu değerler yerel en iyilere takılmamak ve optimum çözüme yaklaşma hızını arttırmak için algoritma çalışırken problemden aldığı geri bildirimler doğrultusunda güncellenmelidir. Bundan dolayı bu tez çalışmasında ikinci amaç, ETMB'lerin parametre veri uzaylarını ayarlamak amacı ile kullanılan MSA'ların kritik parametre veri uzaylarında ayarlamaktır. MSA'lara ait kritik parametre değerleri çevrimiçi uyarlanabilir parametre ayarlama (ÇUPA)

yöntemi kullanılarak bu tez çalışmasında ayarlanmış ve kullanılan MSA'lar probleme uyarlanabilir hale getirilmiştir.

Bu tez çalışmasında optimizasyon problemlerinde başarısı kanıtlanmış MSA'dan Genetik Algoritma (GA), Parçacık Sürü Optimizasyonu (PSO), Yapay Arı Kolonisi (Artificial Bee Colony (ABC)) ve Ateş Böceği (Firefly Algorithm (FA)) algoritmaları kullanılmıştır.

Genetik algoritma bir sezgisel arama algoritmasıdır. Doğal seçilim ve genetik konularından esinlenerek mutasyon ve çaprazlama yöntemlerini de kullanarak verilen probleme çözüm sunmayı hedefler (Holland, 1975).

Parçacık sürü algoritması bir sürü tabanlı sezgisel optimizasyon algoritmasıdır. Sürülerin özellikle kuş sürülerinin hareketlerinden esinlenilerek geliştirilmiştir. Kuşların sürü içerisinde besine yakın olan kuşların konumlarına göre hareket etmeleri algoritmanın temelini oluşturur (Kennedy, Eberhart, 1995)

Yapay Arı Kolonisi sürü tabanlı bir sezgisel optimizasyon algoritmasıdır. Yapay Arı Kolonisi Algoritması arıların besin arama davranışlarından esinlenilerek yazılmıştır. İşçi, gözcü ve kâşif arıların besin arama ve işleme davranışları algoritmaya uyarlanmıştır (Karaboga, Basturk, 2007).

Ateş Böceği Algoritması (FA) da Sürü tabanlı sezgisel optimizasyon algoritmasıdır. Ateş böceklerinin parlaklığa duyarlı sosyal davranışlarını ele alarak geliştirilmiştir. Algoritma ateş böceklerini cinsiyet olmadan ele alır. Yani tüm ateş böcekleri bir birlerine yönelebilirler. Daha parlak olan ateş böcekleri daha çekicidir. Daha az parlak olan ateş böcekleri çekici olan ateş böceklerine doğru yönelir. Parlaklık etkisi uzaklık arttıkça azalacağı için daha uzaktaki ateş böcekleri uzaktaki parlak ateş böceklerinden daha az etkilenir. Ayrıca bir ateş böceği eğer kendinden daha parlak bir ateş böceği bulamazsa rasgele hareket gerçekleştirir (Yang, 2008).

Ayrıca bu algoritmalar probleme uyarlanabilir duruma getirilerek farklı modellerde daha iyi sonuçlar üretmesi hedeflenmiştir. Bunun için algoritmaların kritik parametreleri çalışma esnasında arama sürecinden gelen geri bildirimlere göre güncellenmektedir yani uyarlanabilir parametre ayarlama işlemi yapılmıştır. Böylece her algoritmanın parametre değeri probleme bağlı olarak değişmektedir bu da çözüm kalitesini arttırmaktadır. Geliştirilen bu algoritmalar, Uyarlanabilir Genetik Algoritma (Adaptive Genetic Algorithm (AGA)), Uyarlanabilir Parçacık

Sürü Optimizasyonu (Adaptive Particle Swarm Optimization (APSO)), Uyarlanabilir Yapay Arı Kolonisi (Adaptive Artificial Bee Colony (AABC)) ve Uyarlanabilir Ateş Böceği Algoritması (Adaptive Firefly algorithm (AFA)) olup probleme uyarlanabildiği gibi her bir algoritmanın bağımsız olarak farklı problemlerde daha başarılı olabildiği gözlemlenmiştir. Her model üzerinde bağımsız parametre arama işlemi gerçekleştirilmektedir. Her problemde farklı algoritmanın başarılı olduğu elde edilen sonuçlarla gözlemlenmiştir.

Bu çalışmanın devamı şu şekildedir;

2. bölümde, Etmen, Çoklu Etmen Sistemler, Etmen Tabanlı Model ve Benzetim ve Karmaşık Sistemler hakkında bilgi verilmiştir.

3. bölümde, parametre ayarlama işleminde kullanılan yöntemler ve yapılan çalışmaların yer aldığı literatür araştırması verilmiştir.

4. bölümde, çalışmanın gerçekleştirildiği ortam ve kullanılan teknik hakkında detaylı bilgi verilmiştir.

5. bölümde, parametre ayarlama işleminde kullanılan ve hepsi birer MSA olan Genetik Algoritma (GA), Parçacık Sürü Optimizasyonu (Particle Swarm Optimization (PSO)), Yapay Arı Kolonisi (Artificial Bee Colony (ABC)) ve Ateş Böceği Algoritması (Firefly Algorithm (FA)) algoritmaları detaylı bir şekilde açıklanmıştır.

6. bölüm, parametre ayarlama işleminde kullanılan probleme uyarlanabilen ve bu tez kapsamında geliştirilen Uyarlanabilir MSA'lar detaylı açıklanmıştır. Bu algoritmalar, Uyarlanabilir Genetik Algoritma (Adaptive Genetic Algorithm (AGA)), Uyarlanabilir Parçacık Sürü Optimizasyonu (Adaptive Particle Swarm Optimization (APSO)), Uyarlanabilir Yapay Arı Kolonisi (Adaptive Artificial Bee Colony (AABC)) ve Uyarlanabilir Ateş Böceği Algoritması (Adaptive Firefly algorithm (AFA)) algoritmalarıdır.

7. bölüm, geliştirilen parametre ayarlama aracının kullanımı ile ilgili detaylara yer verilmiştir.

8. bölüm, test için kullanılan modeller ve deneysel çalışma sonuçlarına yer verilmiştir.

9. bölüm, bu tez çalışması ile ilgili sonuçları sunmakta ve sonuçlar tartışılmıştır.



## 2. ETMEN TABANLI BENZETİM VE MODELLEME

Bu bölümde tez kapsamında ele alınan etmen, çoklu etmen sistemleri (ÇES), modelleme ve benzetim, etmen tabanlı modelleme ve benzetim (ETMB) son olarak karmaşık sistemler gibi temel alanların üzerinde durulmaktadır. Etmen Tabanlı Modelleme ve Benzetim, modellenmesi zor olan karmaşık sistemlerin modellenmesi için kullanılan bir araçtır; karmaşık bir sistemin herhangi bir amacını ya da herhangi bir davranışını yerine getirmek için bir araya gelmiş, ortamla ve ortamdaki diğer bileşenlerle etkileşimde bulunan etmenlerden oluşmaktadır. Bu nedenle bu bölümde öncelikle, etmenler, ÇES ve ETMB kavramları tanımlanmaktadır. ETMB özellikle karmaşık sistemlerin benzetimi için uygun özellikleri sunması sebebiyle giderek yaygın şekilde kullanılmaktadır. Bu nedenle, karmaşık sistemler bu bölümde kısaca tanıtılmaktadır.

### 2.1. Etmenin Tanımı

Etmenler yapay zekâ tekniklerini işleyen her biri özerk olan sanal ya da fiziksel bir varlıktır. Her bir etmen belirlenen amaç ya da amaçlara ulaşmak için kendi karar mekanizmasını kullanan ve çevreyi algılayarak vereceği kararı işleyen, sınırları belirlenmiş ve oluşturulan arayüzler ile iyi tanımlanmış yapay varlıklardır. Bir ortama konumlandırılırlar, konumlandırıldıkları bu ortamı algırlar, etkileyiciler yardımı ile ortamda hareket edebilir ve gerektiğinde tepki verirler. Etmenlerde insanlar gibi diğer etmenlerle iletişime geçebilen sosyal varlıklardır. Ayrıca sistemden istenenleri sınıflandırma, tartışma ve reddedebilen oto kontrol mekanizmaları vardır. Tüm bu özelliklere sahip etmenler karmaşık problemleri çözmede oldukça başarılıdır.

Etmen (agent), çoğunlukla, bilişsel varlıkların genel alanını tanımlamak için kullanılıyor. Farklı alanlarda yapılan araştırmaların özelliklerine göre farklı araştırmacılar farklı etmen tanımları yapmıştır. Bu tanımların bazıları aşağıda verilmiştir.

“Etmenler, karmaşık sistemlerin somutlaştırılmış verimli problem çözücüleri olarak tanımlanabilir. Etmenler ortamdan aldıkları etkileri bazı algı araçları ile alıp uygun tepkiler yaratan sistemlerdir” (Franklin ve Graesser, 1996).

“Belli bir amaca adanmış, kalıcı yazılım varlıkları” (Smith ve ark., 2002)

“Başka bir insanın sizin için yapabileceği herhangi bir şeyi yaparak insan ilişkilerini taklit eden bilgisayar programı” (Luck ve d’Inverno, 2004).

“Tasarım hedeflerini karşılamak için, bir ortama yerleştirilmiş ve bu ortamda esnek ve otonom hareket etme kapasitesinde, kapsüllü bilgisayar sistemleridir.” (Gandon, 2002).

“Ana özellikleri yerleşmişlik, özerklik, uyarlanabilirlik ve toplumsallık olan bir bilgisayar yazılımı” (Sycara, 1998).

“Belirli bir amaca hizmet etmek için tasarlanmış, kendi, iç durumları ve kendi davranışları üzerinde kontrol sahibi olan, tasarım nesnelerini elde etmeye uğraşırken esnek problem çözme davranışı sergileme kapasitesine sahip, sensörlerle algıladıkları ve dengeleyicilere göre davrandıkları ortamlarda yerleşmiş, iyi tanımlanmış sınırlar ve arayüzlerle, açıkça tanınabilir problem çözücü varlıklardır” (Gandon, 2002).

"Algılayıcıları yardımıyla ortamı algılayan ve etkileyicileri yardımıyla bu ortamı etkileyen bir sistemdir" (Russell et al., 1996).

“Birbirinden bağımsız, özerk karar alan, dağıtık yazılım birimleridir.” (Weiss, 1999).

"Bir işi nasıl yapıldığının anlaşılmasına gerek kalmadan kullanıcıları adına gerçekleştiren makinedir" (Minsky and Riecken, 1994).

"Karmaşık ve dinamik olarak değişebilen bir ortamda bulunan, ortamı algılayan, ortamı etkileyen ve bunları bir takım amaçlar doğrultusunda yapan sistemlerdir" (Maes, 1995).

"Akıllı etmenler sürekli olarak aşağıda belirtilen üç fonksiyonu yerine getiren sistemlerdir:

1. Ortamdaki dinamik değişimleri algılamak.
2. Ortamı etkileyen eylemlerde bulunmak.
3. Algıları yorumlamak ve amaçlar doğrultusunda yapılması gereken eylemlerin belirlenmesi için akıl yürütmek" (Hayes-Roth, 1995).

"Bir sistem ancak ve ancak bir etmen iletişim dili ile iletişimde bulunma yeteneđi varsa etmen olarak adlandırılabilir" (Genesereth and Ketchpel, 1994).

"Etmen, kullanıcısının adına bir takım görevleri yerine getirmek üzere davranma yeteneđi olan yazılım veya donanım bileşenidir" (Nwana, 1995).

"Etmen;

- bir ortamda hareket edebilen,
- diđer etmenlerle iletişim kurabilen,
- hedefleri dođrultusunda varlığını sürdürebilen,
- kendi kaynakları olan,
- belirlenen ölçütlerle bulunduğu ortamı algılayabilen,
- ortamın sadece belli bir bölümünü temsil eden,
- yetenekleri olan ve hizmet öneren,
- kendi kendinin kopyalayarak üreyebilen,

-hedefleri dođrultusunda davranışlarına yön veren, kaynakları kullanabilen, fiziksel ya da sanal bir varlıktır" (Ferber, 1999).

"Kendisinden beklenenleri yerine getirmek için belli bir ortamda belli derecede özerklik çerçevesinde çalışan, algılayıcıları ile ortamdaki dinamik deđişimleri algılayan ve elde ettiđi algılara göre bilgisini, amaçlarını yeniden deđerlendiren, amaçları dođrultusunda planlama yaparak bu planlara ilişkin eylemleri yapan, diđer etmenler ile bir iletişim dili aracılığı ile iletişimde bulunma yeteneđi olan ve bulunduğu ortamda süreklilik gösteren yazılım veya donanım tabanlı sistemdir" (Erdur, 2001).

Verilen etmen tanımları incelendiđinde bir etmende olması gereken temel özellikler hakkında bilgi verdiđi görülmektedir. Her etmenin kendine ait özellikleri vardır ve etmen yazılım sistemlerini diđer yazılım sistemlerinden ayırır. Bu özellikler sonraki bölümde incelenmiştir.

## 2.2. Etmen Özellikleri

Etmenlerin tasarlanması, onların bazı özelliklere göre sınıflandırılmasını gerektirmektedir. Etmenler ihtiyaç doğrultusunda bu özelliklerin bazılarını ya da hepsine sahip olabilirler. Yapılan çalışmaların incelenmesi sonucunda da etmenlerin sınıflandırılması aşağıdaki şekilde yapılmıştır. (Ferber, 1999; Wooldridge, 2002; Weiss, 2000; Gandon, 2002; Luck ve d'Inverno, 2004; Ren ve Anumba, 2003);

• **Özerklik:** bulunduğu ortamda başka etki olmadan kendi hareketlerini kontrol edebilmesi. Etmenlerin davranışları sınırsız özerklik içinde değildir; kullanıcıdan aldığı geri bildirimlere göre şekillenmektedir.

• **Karşıt-eylemlilik:** Etmenler buldukları ortamları algılayabilmeli ve ortamdaki değişimler doğrultusunda davranışlarına yön verebilmelidir. Gerekirse bu değişimi diğer etmenlere bildirebilmelidir. Etmenin bulunduğu ortam, fiziksel algılayıcılarla algılanabilir ya da etmen topluluğunun oluşturduğu ortam olabilir.

• **Amaç-Yönelimlilik:** İnisiyatif kullanma ayrıca hedefler doğrultusunda hareket etme.

• **Sosyal Yetenek:** Bir iletişim dilini kullanarak diğer etmenlerle ya da kullanıcılarla iletişim kurabilmesi. Etmenler birbirleri ile ya da kullanıcılar ile iletişimde bulunarak planladıkları görevleri yerine getirebilir.

• **Kalıcı Süreklilik:** Etmen kendi görevini tamamladıktan sonrada etkin olmalı gerektiğinde yeni görevler üstlenip gerçekleştirebilmelidir.

Yukarıda bahsedilen özellikler etmen tabanlı yazılım sisteminde her bir etmenin sahip olması gereken özelliklerdir. Aşağıda yer alan özellikler, geliştirilen sistemin beklentisine göre etmene yüklenebilen özelliklerdir.

▪ **Göç Edebilme:** Çalıştığı ortamı değiştirmesi, farklı sistem ortamına geçmesi.

▪ **Öğrenme:** Davranışlarının önceden elde edilen tecrübeyle değiştirmesi.

▪ **Akılcılık:** Amaçlarına ulaşmak için planların yapılması.

▪ **Dürüstlük:** Bir etmenin mümkün olduğunca doğru davranışı göstermesi.

▪ **Tepkisellik:** Kendi ortamını algılaması yani bulunduğu ortamdaki değişimlere anlık cevap verebilmesi.

▪ **Bireysellik:** İyi tanımlanmış sınırları belli olan bir ortamda açıkça kimliği belli olan bir birey olması.

▪ **Oto kontrol:** Komutları modifiye ederek gerçekleştirmesi, doğrudan komutları gerçekleştirmemesi.

▪ **Esneklik:** Hareket seçiminin etmene bırakılması, ortamdaki beklenmedik durumlarla karşılaşıldığında uyum sağlayabilmesi.

▪ **Hedef-Yönelimlilik:** Diğer etmenlerin isteklerini dikkate alarak önceliği olan istekleri hangi durumda ve ne zaman gerçekleştireceğine kendisinin karar vermesi.

▪ **Kişisellik:** İnsanlarda olan kişilik ve duygusallık özelliklerini edinilmesi.

▪ **Adaptasyon:** Ortamdaki değişikliklere ya da kullanıcının isteklerine göre kendini uyarlaması.

▪ **Gezicilik:** Ağ üzerinde dolaşabilme.

▪ **Görsel Sunum:** Sanal gerçekliğe sahip arayüzlerden bilinen klasik arayüzlere kadar geniş bir alanı kapsamaması.

▪ **Yardımseverlik:** Etmenlerin hedefleri çakışmadığı sürece diğer etmenlerin her istediğini yaptığının farz edilmesi.

▪ **Rasyonellik:** Etmenlerin amaçlarına ulaşmak için hareket ettiğini varsayarak, etmenlere insanlara ait özelliklerin yüklenmesi ve etmenlerin insanlar gibi davranmasını sağlamak. Etmen tabanlı yazılımlara etmenlere bilgi, niyet ve inanç gibi özellikler kazandırılmaya çalışılmaktadır.

Bir etmen her şeyi yapabilen ve her şeyi bilen bir bileşen olmak zorunda değildir. Etmenler farklı durumlara uyum sağlayarak ortamdan ve diğer etmenlerden aldığı bilgiler doğrultusunda hedeflerine ulaşmaya çalışmalıdır. Bunun için de etmen için akıl yürütme süreci ve planlama yapılmalıdır (Erdur, 2001).

### 2.3. Çoklu Etmen Sistemler

Etmen tabanlı sistemlerle alışılmış yazılım dünyasına alternatif bir bakış getirilmiş, klasik yöntemlerle çözüm üretilemeyen özellikle internet gibi dağıtık ve karmaşık sistemlerdeki problemlere, çözüm sunan yazılım geliştirme ortamı sağlanmıştır. Teknolojinin gelişmesi ile birlikte uygulamaların karmaşıklığı artmıştır bu problemlerin çözümü birbiriyle doğrudan bağlantı kurabilen ve işbirliği yaparak problemlere çözüm üreten çok etmenli sistemler geliştirilmiştir (Sycara, 1998).

Bugüne kadar çoğu Avcı, tek bir etmenin davranışlarını ve özelliklerini inceleyebilmek ve anlayabilmek için kuramlar, yöntemler ve sistemler geliştirme üzerinde durmuşlardır. Yapay Zekâ teknikleri geliştikçe, daha karmaşık problemlerin de bilişim teknolojileri kullanılarak çözülebileceği görülmektedir. Fakat gerçek dünyadaki problemleri, tek bir etmenin çözmesi mümkün değildir.

Akıllı bir etmenin sahip olduğu tüm bilgiler, hesaplama verileri ve bakış açılarıyla sınırlıdır. Bu nedenle, çoklu etmen sistemlere ihtiyaç duyulmuştur. Eğer bir problem, özellikle karmaşık ve büyük veri setine sahip ya da dinamikse, bu problemin çözümü, kendine özgü işlevselliği ve modülerliği olan çok sayıda bileşenin geliştirilmesiyle olur. Her bir etmen, hedef problemi çözmek için en uygun paradigmayı kullanır. (Anumba vd., 1999). Her bir etmen faydalı çıkarım eylemlerini kendi başına yapabilme yeteneğine sahip bireysel yapay zekâlardır. Ancak, birbirlerinden bağımsız problemlerin oluşması durumunda, performanslarını artırmak ve probleme çözüm geliştirmek amacıyla diğer etmenlerle işbirliği yapıp ortak çalıştıkları bir yapay toplum platformu oluştururlar (Ferber,1999).

Son yıllarda yapılan çalışmalar incelendiğinde, çoklu etmen sistem çalışmalarında artış olduğu görülmektedir. Yapılan çalışmalar, birbirleriyle ve çevreleriyle etkileşim içinde olan, özerk etmenlerin bir araya gelmesiyle oluşan çoklu etmen çalışmalarıdır. Çoklu etmen sistemlerle ilgili çalışmalarda, yapay zekânın ötesinde, sosyal hayat problemleri gibi karmaşık sistem problemleri çözülmeye çalışılmaktadır. Bu problemleri çözmek için tek bir etmenin bilgisinin, yeteneğinin veya kaynaklarının yeterli olmadığı durumlarda, birbirleriyle ve ortamla etkileşim içerisine giren çok sayıda etmen işbirliği, yani çoklu etmen sistem kullanılarak problem çözülmeye çalışılır (Shen ve Lesser, 2006).

Çoklu etmen sistemler, gerçek dünyadaki karmaşık problemleri çözmek için geliştirilmiş, gerçek dünyadaki bileşenleri sanal ortamda temsil eden, istenilen hedefleri gerçekleştirmek için birlikte çalışan ya da birbirleri ile yarış içinde olan çok sayıda zeki etmenin bir arada olduğu, temeli organizasyon, koordinasyon ve işbirliği olan, karmaşık sistemle başa çıkmanın en etkili yöntemi sayılan modülerlik ve soyutlamayı sağlayan ve hızlı gelişen bir bilgi teknolojisidir. Çoklu etmen sistemlerde önceden oluşturulmuş bir mimari ya da etmenlerin bir arada çalışmasını sağlayan bir yapılandırma yoktur. En önemlisi, etmenlerin, ortak davranışlar yerine her etmenin kendi amaç ve yetenekleriyle farklı etmenlerden oluşan bir topluluk oluşturmasıdır. Böylece çoklu etmen sistemler, karmaşık sistem problemlerine, dağıtık yapay zekâ tekniğini getirmiş olur (Şeker, 2006)

Çoklu etmen sistemle ilgili yapılmış genel tanımlar;

“Bir çoklu etmen sistem, bazı hedeflerin başarılması için birlikte çalışan, birçok etmeni içermektedir. Bu durumda bir etmen örgütü olarak, ya da başka bir deyişle, yapay toplum ya da teşkilat olarak görülebilir” (Mellouli et al., 2002.)

“Tek bir etmenin yalnız başına kendi bilgi ve bireysel yeteneklerini kullanarak çözemediği veya etkin bir biçimde çözemeyeceğini düşündüğü problemleri birbiriyle iş birliği yaparak eş güdümlü bir biçimde çözmek için bir araya gelen etmenlerin oluşturduğu ağ, iş birlikçi çoklu etmen sistemi olarak adlandırılmaktadır” (Sycara, 1998).

“Tek bir etmenin tek başına çözemediği problemleri birden fazla etmenin bir araya gelerek işbirliği yaparak eşgüdümlü çözen sistemlerdir” (Durfee ve ark, 1989).

Çoklu etmen sistemler etmenlerin yetenek, bilgi ve hedefleri nasıl eşgüdümlü gerçekleştireceği üzerinde durur.

Çok etmenli sistemleri diğer sistemlerden ayıran özellikler;

- Etmenler tek başlarına problem çözümü için gerekli olan tüm bilgiye sahip değildirler.
- Sistemde yer alan her bir etmen tek başlarına problemi çözebilmeye yeterliliğine sahip değildir.
- Sistem kontrolü dağıtıktır.
- Veriler bir merkezde değildir dağıtıktır.
- İşleyiş eş zamanlı değildir.

Çoklu Etmen Sistemi özetleyecek olursak yaygın bir yapay zekâ tekniği olup etmenler arasında işbirliğinin gerekli olduğu, bir problemin çözülmesi gerektiği ve uzman bilgisi içeren etmenlerle iş birliği yapılması gerektiği durumlarda kullanılmaktadır (Ferber, 1999). Geliştirilecek sistemde tek bir etmene sahip bir yapı mı yoksa çoklu etmene sahip bir yapının mı tercih edilmesi gerektiği tamamıyla sistemin gereksinimlerine bağlıdır.

### **2.3.1. Çoklu Etmen Sistemlerinde Bireysel Etmenler**

Çoklu etmen sistemlerde, sistem yapısını oluşturan ilk bileşen bireysel etmenlerdir. Sistemdeki her etmen, yerel hareketlerin küresel etkilerini

değerlendirebilmelidir, diğer etmenlerin davranışlarına yönelik beklentiler oluşturabilmeli ya da olabilecek zararlı etkileşimlere ve çakışmaları düzeltebilmelidir. Bu alanda yapılan birçok çalışma etmenlere mantıklı aksiyom özelliği kazandırmaya yöneliktir (Jennings ve Wooldridge, 1995).

Çoklu etmen sistemlerde, her etmenin;

- İnanç kümeleri; sahip olduğu bilgiyi tutar.
- Hedef Kümeleri; ulaşacağı hedefleri.
- Hareket kümeleri; hedefine ulaşacak hareketlerin davranışların yer aldığı kümedir.
- Plan kümeleri; etmen hareketlerinin hedefe ulaşmasına yardımcı olacak kümedir (Shakeri,1998; Sycara,1998).

Bir olay meydana geldiğinde etmen;

- İlk olarak cevap verebilecek planları arar;
- Bulduğu plan için, etmenin içinde bulunduğu duruma uygunluğunu sorgular;
- Bulduğu en uygun planı işlemeye başlar.

Ayrıca etmen, hangi hedefe yöneleceğini, hangi duruma tepki göstereceğini, hedefe nasıl ulaşacağına ya da gerekirse hedefe ulaşmayı yarıda bırakıp başka hedefe nasıl yönleneceğine bireysel olarak karar verir. Etmenler, tepkisel ve düşünceli davranışları arasındaki dengeyi, bir sonraki seferde ne yapacaklarına karar vermeleri için belirlenen sürenin uzunluğuna veya kısalığına göre değiştirirler. Böylece bir sonraki adımda aynı etmen daha duygusal veya daha tepkisel olabilir (Autonomous Decision Making Software, 2018).

### **2.3.2. Çoklu Etmen Sistemlerde İletişim**

Etmenler arası etkileşim ve işbirliği çok etmenli sistemlerde önemlidir. Çok etmenli sistemler problemi birlikte çözebilmeleri için birbirleri ile iletişimde bulunmalıdır. Basit sunucu/istemci iletişim ağı çok etmenli sistemlerde yeterli değildir bunun nedeni bir etmenin bazı durumlarda istemci bazı durumlarda sunucu olabileme özelliğidir. Ayrıca etmen tabanlı sistemlerde basit istemci/sunucu iletişim biçimi dışında farklı türden etkileşim biçiminin de kullanılıyor olmasıdır.

İletişim ve etkileşim terimleri birbirinden farklıdır. İletişim protokolleri, etmenlerin karşılıklı ileti gönderebilmeleri ve bu iletilerin anlamlarını etmenler tarafından aynı algılanmasına yöneliktir. Etkileşim protokolleri, etmenlerin birbirleri ile konuşabilmeleridir. Konuşma belirli özellikteki birden çok iletinin karşılıklı olarak gönderilmesidir. Etmen iletişim dilleri ve protokoller iletişimi gerçekleştirmektedir (Erdur, 2001).

## 2.4. Benzetim ve Modelleme

Benzetim, gerçek dünya sistemlerinin (Karmaşık sistemlerin) veya süreçlerinin zaman içinde işleyişinin sanal ortamda taklit edilmesidir (Banks 1999; Banks 2004).

Modelleme, gerçek ortamda incelenmesi zor olan karmaşık sistemi incelemek veya bilgi elde etmek için kullanılan ve o sisteme ait bileşenlerin, kuralların ve ortamın temsil edildiği herhangi bir görüntü olarak tanımlanır (Carson, 2003).

Benzetimlerin amacı; Gerçek sistem davranışlarını taklit etmek uygun girdi değerleri ile istenen çıktıları kullanıcıya sunabilmesidir. Gerçek ortamda incelenmesi ve gözlemlenmesi zor olan veya çok uzun zaman alan sistemlerin benzetim yolu ile çok daha kısa zamanda istenen davranış ya da olayın incelenmesi çok daha kolaydır. Gerçek davranışların tahmin edilmesi, varsayımların doğrulanması gibi birçok neden bilimsel araştırmaların yapılmasında benzetim yönteminin tercih edilmesini sağlamıştır (Banks, 2005; Naylor, 1966; Shannon, 1998).

Benzetim aşağıda verilen amaçlardan birisini veya bir kaçını gerçekleştirmek için kullanılır.

- **Değerlendirme:** Belirlenen ölçütler doğrultusunda geliştirilen sistemin beklentileri ne kadar karşıladığının gösterilmesi,

- **Karşılaştırma:** Geliştirilen sistem tasarımlarının veya ilkelerin karşılaştırılması,

- **Tahmin:** Önerilen koşullar altında sistemin performansının tahmin edilmesi,

- **Duyarlılık Analizi:** Sistemin başarısı üzerinde hangi etkenlerin etkili olduğunu belirlenmesi,

• **Optimizasyon:** En iyi performans değerini veren etken düzeylerinin bir kümesinin belirlenmesi,

• **Darboğaz Analizi:** Bir sistemde darboğazların belirlenmesi amacıyla benzetim kullanılır (Pedgen et al., 1995).

Ayrıca benzetim yeni bilgilerin elde edilmesini mümkün kılmıştır. Yapılan araştırmalar incelendiğinde aşağıdaki durumlarda benzetim kullanımına ihtiyaç duyulmuştur. (Naylor, 1966; Shanon, 1998):

- Problemin matematiksel çözümü yoksa.
- Sonuçlar için uzun zaman beklemek gerekiyorsa.
- Gerçek sistemin üzerinde çalışmanın zor ya da mümkün olmadığı durumlarda,
  - Gerçek ortamda davranışları yinelemek mümkün değilse,
  - İlk örnekleme gereken durumlarda,
  - Sistem karmaşık olmadığı ve kontrol edilebilir olduğu durumlarda,
  - Gerçek sistemin anlaşılması zor ve karmaşık olması durumunda,
  - Sistemin tamamı yerine o sisteme ait bir davranış, teori ya da hipotez kontrol edilmek istendiği durumlarda,
    - Gerçek sistemden farklı ya da daha gelişmiş bir sistem modellemek istendiğinde,
    - Genel varsayımlar yapılabildiğinde,
    - Varsayımların sonuçlarının ve etkilerinin gözlemlenebildiği bir arayüze ihtiyaç duyulduğunda benzetimlere başvurulur.

Her sistemin modellenmesi birbirinden farklı olur bu nedenle farklı benzetim yöntemleri kullanılmalıdır (Troitzsch,1996). Ayrıca, benzer uygulamaların farklı benzetim yöntemleriyle modellenmesi de mümkündür.

Benzetimin kullanılması bilimse araştırmalara büyük kolaylık sağlamıştır (Banks, 2005). Benzetimler gerçek sitemi etkilemeden çalışır. Gerçek sistemden çok daha hızlı sonuç üretmesini, sistemin incelenmesini kolaylaştırdığından araştırmacılar tarafından tercih edilmektedir. Benzetimlerdeki değişkenlerinin etkilerinin gözlemlenmesi daha kolaydır. Ayrıca gerçek sistemde hipotez testinin yapılmasının imkânsız olduğu durumlarda kurtarıcı etkiye sahiptir. Bu avantajların yanında benzetimin bir takım tez avantajları da mevcuttur (Banks, 2005). Benzetim modeli sistem karmaşıklıkça oluşturulması da zorlaşmaktadır. İstenen davranışın ya da durumun gözlenebilmesi için gerçek sistemin eksiksiz modellenmesi

gerekmektedir. Bunun gerçekleşmesi için benzetim modelleme evrelerinde gerekli test ve doğrulama işlemlerinin yapılması gerekmektedir. Ayrıca doğru parametre değerlerinin kullanılarak sonuçların gerçek sistemi yansıtması önemlidir. Fakat var olan benzetim araçları parametre ayarı, doğrulama ve test işlemlerini gerçekleştirme yeterliliğine sahip değildir.

#### **2.4.1. Benzetimlerin Avantajları ve Dezavantajları**

Bilimsel çalışmalar için benzetim kullanımı birçok avantaj sağlar (Banks, 2005). Benzetimler gerçek sistemi etkilemeden çalışırlar. Ayrıca, Benzetimler uyarlanan gerçek sistemden çok daha hızlı sonuç üretirler gerçek ortamda gözlemlenmesi ve incelenmesi zaman alan davranışlar kısa sürede gözlemlenebilir. Benzetimler farklı parametrelerin sistemin genelinde ya da diğer parametrelere, davranışlara olan etkilerinin gözlemlenebilmesi bakımından yararlıdır. Bunlarla birlikte benzetimler hipotez testi ya da hipotez testinin imkânsız olduğu durumlar için kurtarıcıdır.

Benzetim kullanımının avantajlarının yanında bir takım dezavantajları da vardır (Banks, 2005). Benzetim modellerinin gerçekleştirilmesi zordur. Oluşturulan modeller sonucu elde edilen davranışların gerçek sistem davranışlarını yansıtabilmesi için benzetim modelinin doğru, tam ve geçerli olarak gerçekleştirilmesi gerekmektedir. Var olan benzetim araçları geliştirim, modelleme, doğrulama ve geçerleme açılarından zayıftır. Benzetim sonuçlarının değerlendirilmesi zaman alıcıdır. Ancak benzetim sonuçlarının değerlendirilmesi gerçek sistemin gözlemlenip elde edilen sonuçların değerlendirmesine kıyasla daha az zaman almaktadır.

#### **2.5. Etmen Tabanlı Benzetim ve Modelleme**

Etmen tabanlı benzetim ve modellemenin (ETMB) temel amacı varlıkların gerçek dünyadaki davranışlarının etmenlerin özerk davranışları ile modellenebilmesidir. ETMB ve benzetiminin başlıca yararı aynı model ile diferansiyel denklemlerin, sayısal değişkenlerin ve sembolik kurallara bağlı olan davranışların modellenmesi ve benzetimine olanak tanımaktadır. Aynı zamanda, modelde kolayca değişiklik yapılabilir. Ayrıca zaten etkileşim halinde olan etmenlerin etkileşimde oldukları etmen sayısını arttırmak diğer bir avantajını oluşturmaktadır. Etmen tabanlı benzetimlerin en önemli özelliği alt seviyedeki davranışlarda meydana gelen değişimin üst seviye değişikliklere neden olması ve

geleneksel benzetimlerle modellenmesi zor karmaşık sistemlerin de ETMB ile modellenmesi ve benzetimi mümkün kılmasıdır. ETMB, Gerçek sistem bileşenlerinin etmenler tarafından temsil edilen bir benzetim modeli olarak tanımlanabilmektedir (Mala, 2009).

ETMB'lerin üç temel bileşeni vardır;

- **Etmenler:** Sistem bileşenleridir. Bu bileşenler buldukları ortamla gerçek sistemdeki bileşenler gibi etkileşim halindedir. Etmenler gerçek sistem bileşenlerinin davranışlarını üstlenirler. Benzetim modeli içinde etmenler etmen tanımında da ve çok etmenli sistem bölümünde de açıklandığı gibi buldukları ortamla ve diğer etmenlerle etkileşim içindedirler.

- **Temsili Ortam:** Gerçek sistemin etmen olmayan, fakat sistemin bir bütün olması için gerekli olan bileşenlerini temsil ederler. Ayrıca etmenlerin etkileşimde bulunduğu, hareketlerinin eyleme dönüştüğü ve iletişim kurdukları gerçek ortamı temsil ederler. Etmen temelli benzetim modelinde (ETMB) en az bir temsili ortam bulunmak zorundadır. Benzetim modelinin bir parçası olan temsili ortamda çoğunlukla tüm etmenler tarafından modelde paylaşılan, bazen bilgi içeren kaynaklar ve durumlar bulunmaktadır. Fakat birçok benzetim ve modellemede ortam, sadece etmenlerin eylemlerini gerçekleştirdikleri yer olarak tanımlanmaktadır. Gerçek dünya da, ortamdaki aktif bileşenlerin birbirleri ve ortamla etkileşim içinde olmaları nedeniyle ETMB için temsili ortamın önemli bir yeri vardır. Benzetim modeli oluşturulurken temsili ortamda bulunan etmenler, etmen davranışları ve diğer bileşenler tanımlanır. Bununla birlikte benzetim bileşenlerinin davranışlarını belirleyen kurallar benzetim modeli içinde bulunabilir. Fakat benzetim modelinde tanımlanan davranışların dışında etmenlerin diğer etmenlerle, ortamla ve etmen dışındaki diğer bileşenlerle etkileşimleri sonucunda yeni davranışlar ortaya çıkabilir. Bu davranışlar bazı değişikliklere neden olabilir (Polack, 2010). Sonuç olarak temsili ortam hazırlanan benzetimin amacına ulaşması için önemli olan diğer bileşendir.

- **Benzetim Ortamı:** Benzetim ortamı etmenlerden ve temsili ortamdaki oluşan benzetim modelinin çalıştırılmasını sağlayan ve benzetim altyapısını temsil eden bileşendir. Benzetim ortamı, benzetimin mesajlaşma, zamanlama ya da dizin hizmetleri gibi temel benzetim işlevlerini sağlar. Bu ortam ETMB'nin bir parçası olarak görülmektedir. Bunun nedeni, benzetim modelinin gerçekte modellenen sistemi temsil edebilmesi için bileşenlerinin gerçek sistemin bir parçası olması gerekliliğidir. Benzetim ortamı her bir benzetim için tektir ve benzetim ait davranışları etkilemez.

ETB modelinde her etmene ait özellikler, davranışlar vardır ve sistemin bir parçası olarak işlev görürler. Etmenlerin davranışları, benzetilen ortamda ki diğer etmenlerden etkilenebilir. Benzetim sırasında etmenlerin birbirleriyle ya da ortamla olan etkileşimleri sonucu sistemde tanımlı olmayan davranışların ya da davranış desenlerinin ortaya çıkması gözlemlenebilir. ETMB'lerde zaman kavramı zaman adımları olarak tanımlanır. Her zaman adımında modeldeki her etmen, bir davranış gerçekleştirme, diğer etmenlerle ya da ortam bileşenleri ile etkileşimde bulunma faaliyetlerini gerçekleştirir.

ETMB birçok uygulama alanı için özellikle modellenmesi ve gerçek ortamda incelenmesi zor olan karmaşık sistemlerin modellenmesi için yenilikçi ve çözüm sağlayan bir benzetim yöntemidir. Fakat çoğu zaman tek başına kullanılması gerçek sistemi tam modellemede yeterlilik sağlamayabilir. Bu nedenle bu tez çalışması özellikle optimizasyon işlemlerinde yetersiz kalan ETMB'ler için yardımcı bir araç geliştirmesi bakımından önemlidir (Çakırlar, 2015).

ETMB, karmaşık sistemlerin modellenerek bu sistemleri çözümlenmek ve değerlendirmek için sıklıkla kullanılmaktadır. ETMB, sosyal, ekonomik, biyolojik (Sun et al., 2012), trafik ve diğer karmaşık sistemleri (Heath et al., 2009; Macal and North, 2006) içeren birçok alanda kullanılmaktadır. Bu tez kapsamında ETMB ortamında modellenen benzetimler incelenmiştir.

## 2.6. Karmaşık Sistemler

Karmaşık sistemler için yapılan tanımlar aşağıda yer almaktadır

“Karmaşık sistem, aniden, kendiliğinden ortaya çıkan ve kendi kendine organize davranışlar sergileyen bir sistemdir” (Mitchell, 2009).

“Karmaşık sistemler, evrenin bütünlük, ama aynı zamanda alışılmış mekanik ya da doğrusal yollardan anlayamayacağımız kadar karmaşık, zengin ve çeşitli olan durumu anlamına gelmektedir. Bu yoldan evrenin birçok olay bileşenini anlayabiliriz, fakat daha büyük ve içsel ilişkileri daha büyük olan olgular - ayrıntılara bakarak değil – ancak ilke ve kalıplarla anlaşılabilir. Karmaşıklık; belirme, buluş, öğrenme ve kendini uyarılmanın doğasıyla ilgilidir” (Battram, 1999).

“Karmaşıklık bir sistemin kendini gerçekleştirebilmesi için gerekli olan olanakların sayısının logaritması ya da sistemin alabileceği durumların sayısının logaritmasıdır” (Cramer, 1998).

Karmaşık sistemler, çoğu zaman, bir sistemin bileşenleri arasındaki ilişkilerin toplu davranışlara neden olduğunu ve sistemin çevreyle etkileşim kurma ve çevreyle olan ilişkilerini nasıl şekillendirdiğini araştıran, bilimsel bir yaklaşım olan karmaşık sistemlerin incelemesine değinmektedir (Bar-yam, 2002). Karmaşık sistemler, birbiri ile etkileşim içinde olan çok sayıda bileşenden meydana gelir. Her bileşenin eylemi diğer bileşenlere bağlıdır, bundan dolayı bu sistemler doğrusal değildir. Sistemde yer alan bileşenlerin davranışları gerçek sistemde gözlemlemek ve algılamak oldukça zordur. Bu nedenle bileşenler ve bileşen davranışlarının anlaşılması ve kontrol edilmesi için modelleme ve benzetim araçlarından yararlanılır (Juuso, 2006).

Sistem düzeyinde sistemdeki bileşenleri kontrol eden merkezi bir eleman yoktur. Karmaşık sistemler için oluşturulmuş referans modeli Şekil 2.1’de gösterilmiştir.

Şekilde (Bkz. Şekil 2.1)’deki daireler bileşenleri ve oklar ise bileşenlerin birbirleri ile ve bileşenle ortam arasındaki etkileşimleri göstermektedir. Belirme isimli ok işareti bileşenlerin etkileşimi sonucu ortaya çıkan küresel davranışı temsil etmektedir. Aşağı yönlü ok işareti ise ortaya çıkan yeni durumun bileşenler üzerindeki etkilerini temsil etmektedir.



Şekil 2.1 Karmaşık Sistem Modeli (Gündoğmuş, 2014)

Bileşen etkileşimleri, sistem tarafından algılanan ve davranışları etkileyen daha karmaşık davranışları ve desen özelliklerini ortaya çıkarır. Ortaya çıkan bu yapılar belirme olarak isimlendirilirler. Beliren yapı, sistemin bileşenleri arasında etkileşim dinamikleri sonucunda oluşur. Belirmiş yapı çoğunlukla görece basit bileşenlerin görece basit etkileşimlerinden çıkan beklenmedik, sıradan olmayan sonuçlardır (Serugendo, 2006).

Karmaşık sistemin üç özelliği; evrim, birleşik davranış ve önceden tahmin etmedir.

- Evrim (evolution): Zaman geçtikçe ortamdaki bileşenlerin etkileşimde bulunan sistem bileşenleri yaşama olasılığını arttırmak için evrimleşirler. Bileşenlerin evrimleşme yeteneği karmaşık sistem özelliğidir.

- Birleşik davranış (aggregate behavior): Karmaşık sistemler basit bileşenlerin tek başlarına gösteremeyeceği ancak tüm sistem içindeki etkileşimler sonucu ortaya çıkan davranıştır. Elde edilen davranış belirme ya da ortaya çıkma olarak da ifade edilir

- Önceden tahmin etme (anticipation): Bu özellik belirli karmaşık sistemlerin anlaşılmasını sağlayan birçok cevabı oluştururken sistemleri daha da karmaşık hale getirir. Mevcut ortama sistem uyum sağlamaya çalışırken bileşenler bazı cevapları tahmin ederek adaptasyon kurallarını uyguluyorlar. Bu durumu karmaşık sistemlere uygulamak için karmaşık bir koşullandırma yapısı kullanılır. Bileşenler arası gerçekleşen etkileşimlerle çeşitli koşullandırmalar oluşmaktadır. Bu koşullanma ortaya çıkan davranış üzerinde etkilidir. Sistemin birleşik davranışı her zaman optimal düzeyde olmayabilir bazen optimale çok uzak olabilir. Bunu standart teorilerle optimize etmek bazen mümkün olmamaktadır. (Hodges and Dewar, 2003).

Karmaşık sistemler çok farklı alanlarda çalışılmış bir konu olup özellikle petrol endüstrisi, metal sanayi, görüntü işleme, işaret işleme, elektronik, elektrik, bilgisayar, biyokimyasal süreçler, hastalık teşhisi, medikal cihazlar, su kaynakları, yenilenebilir enerji kaynakları gibi alanlarda karşılaşılmaktadır (Hanbay, 2007). Gerçek dünyayı ve kapsamını anlamak için araştırmacıların çalışmaları sonucunda ortaya çıkmış bir kavramdır. Gerçek dünya birbirleri ile etkileşimde bulunan bileşenlerle evrimleşmektedir. Küçük ve basit özelliklere sahip olan bileşenlerin etkileşimleri, karmaşık davranışların ya da yapıların ve karmaşık sistemlerin oluşmasına neden olur (Miller and Page, 2007). ETMB, karmaşık yapıyı basitleştirmek, sistemin tümünü anlamak için bileşenlerin nasıl davrandığını,

bileşen etkileşimlerinin daha üst seviyelerde nasıl etkilere neden olduğunu gözlemlemeyi amaçlar.

## **2.7. Etmen Tabanlı Sistemlerin Değerlendirilmesi**

Karmaşık sistemler incelendiğinde, bu sistemlerde bir takım belirsizliklerin olduğu görülmektedir. Gerçek ortamda incelenmesi zor ve maliyetli olan bu sistemlerin karmaşık yapısından dolayı incelenmesini kolaylaştırmak amacıyla modelleme ve benzetim tekniklerinden faydalanılmaktadır (Serugendo et al., 2011; Kaddoum and Georg'e, 2012; Guivarch et al., 2012; Brax et al., 2013; Bernon et al., 2009; Georgé et al., 2003; Gürcan, 2013). ETMB ortamlarının karmaşık sistemleri modellemedeki başarısı kanıtlanmıştır. ETMB ortamı gerçek sistem performans ölçütlerinin tahmini ve analizi için bir tasarım aracı olarak kullanılmaktadır. ETMB var olan girdi setine karşılık bir çıktı seti oluşturmakta ve bunu etmenlerin iş birliği ve karar mekanizmalarını kullanarak gerçekleştirmektedir. Karmaşık sistemlerin boyutunun büyümesi ve geniş parametre uzaylarına sahip olması istenen davranışın ortaya çıkmasını (belirmesini) zorlaştırmaktadır. Bu parametreleri optimize edecek sistemlere ihtiyaç vardır. Optimizasyon için kullanılacak mekanizma ile ETMB ortamında modellenen sistemlerin gerçeğini yansıtması ya da istenen davranışın ortaya çıkması daha kolay olacaktır. Bundan dolayı bu tez kapsamında ETMB ortamında modellenen karmaşık sistemlerin parametrelerinin ayarlanması üzerinde durulmuştur.

Sonraki bölüm, bu gereksinime yönelik olası hesaplamalı yöntemleri, yapılan çalışmaları incelemekte ve tartışmaktadır.

### 3. LİTERATÜRDE YER ALAN ÇALIŞMALAR

Karmaşık sistemlere ait büyük parametre uzaylarına sahip modellerin parametrelerinin ayarlanması, modellenen sistemin gerçek sistemi yansıması için önemlidir. Parametre ayarlama problemlerinde çözüme ulaşmak için farklı yöntemlerden faydalanılır. Fakat bu tekniklerin kullanımındaki hedef, gerçek veriler ile modelden elde edilen veriler arasındaki farkı minimize etmektir. Modelin beklenen davranışı gösterme yeteneği, model parametrelerinin doğru belirlenmesine bağlıdır (Yanıkoglu vd., 2010).

Parametre ayarlama problemi bir optimizasyon problemidir ve yapılan araştırmalar parametre ayarlama işleminin eksikliğini ortaya koymaktadır. Optimizasyon işlemleri, birçok alanda kullanılmakta olup özellikle karmaşık sistem modelleme ve benzetim uygulamalarının en önemli somut problemlerinden biri olan geniş parametre uzaylarının ayarlanma işlemlerinde ihtiyaç duyulmaktadır. Karmaşık sistemleri modellemede ETMB'lerden faydalanılır. Modelleme ve benzetim yöntemi var olan girdi verilerine karşılık bir çıktı veri seti oluşturmakta ve bunu tahminleme yöntemi kullanarak gerçekleştirmektedir. Karmaşık sistemin parametre uzayı arttıkça tahminleme yönteminin istenen parametre değerlerini oluşturması imkânsız hale gelmektedir. Bundan dolayı parametreleri ayarlayacak yeni yapılara ihtiyaç vardır. Son yıllarda yapılan çalışmalar incelendiğinde benzetim ve modelleme ile birlikte optimizasyon işlemleri için sıklıkla optimizasyon algoritmalarının kullanıldığı görülmektedir. Optimizasyon yöntemlerini kullanmanın önemli avantajları olduğu bilinmektedir. Karmaşık sistem tasarımı için, geleneksel yöntemler istenen çözümleri bulamazken ya da çoğu zaman yetersiz kalırken optimizasyon yöntemleri gerçeğine yakın fiziksel modellere dayalı en iyi sonucu verecek parametre setini arar. Modelleme ve benzetim alanında önemli olan optimizasyon yöntemleri karmaşık fonksiyonların farklı çözümleri için çok sayıda işlem yapmayı gerektirir.

Bu bölüm, Parameter Tuning Algorithms in Modeling And Simulation (Korkmaz Tan ve Bora, 2017a), Modelleme Ve Benzetim Ortamında Parametre Optimizasyonu Ve Kullanılan Teknikler (Korkmaz Tan ve Bora, 2017b) isimli çalışmalardan derlenmiş olup bir optimizasyon problemi olan parametre ayarlama işlemlerine çözüm sunmak için geliştirilen tekniklere ve yapılan modelleme çalışmalarının da kullanılan parametre ayarlama yöntemlerine yer vermektedir. Ayrıca bu bölümde yapılan çalışmalar incelendiğinde ETMB tekniği ile modellenen karmaşık sistemlerde parametre ayarlama problemi ile ilgili yeteri kadar çalışmanın

olmadığı görülmektedir. Yapılan çalışmalar ise tek bir problemin çözümüne yöneliktir. Bu nedenle bu tez kapsamında geliştirilen araç farklı karmaşık sistem problemlerine çözüm sunmak için yapılmıştır.

Bu bölümün devamında parametre ayarlama kavramı, literatürde kullanılan parametre ayarlama algoritmaları ve bu algoritmaların kullanıldığı modelleme çalışmaları, parametre ayarlama kullanılan algoritmaların değerlendirilmesi, Son olarak neden karmaşık sistemlerde parametre ayarlama ihtiyacı olduğu ve nasıl bir yöntemin kullanılması gerektiği tartışılmaktadır.

### 3.1. Parametre Ayarlama

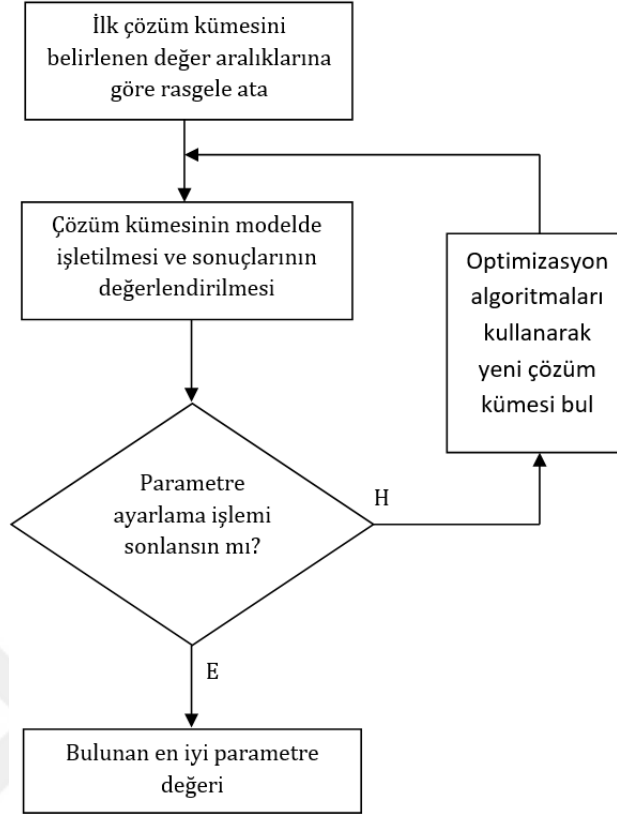
Parametre ayarlama, mühendislik ve temel bilim alanlarında sıklıkla kullanılan, var olan bir sistemdeki parametrelerin en verimli şekilde kullanıldığı ve belirli hedeflere ulaşmayı sağlayan bir teknolojidir. Bir model ve benzetim çok sayıda parametre içermektedir. Bazı parametreler benzetim ortamına ait parametrelerdir ve bu parametreler değiştirilemez. Bundan dolayı ayarlanacak parametre uzayı içinde benzetim ortam parametreleri bulunmaz, sadece modele ait parametreler bulunur. Model parametrelerinden bazıları, ortam bilgisiyle yani sabit değiştirilemez değerler alırlar, bazı modele ait parametrelerin de değerlerinin korunması gerekebilir. İlk aşamada ayarlanması gereken parametrelerin belirlenmesi ve ardından parametre ayarlama sürecine girilmesi gerekmektedir. Parametre ayarlama, uygun parametre setini bulma sürecini hızlandırmakta ve parametre veri seti kalitesini arttırmakta kullanılarak gerçek hayatta karşılaşılan problemleri etkin, doğru ve gerçek zamanlı çözümünde kullanılmaktadır. Ayarlama modelleme ve çözümlenme iki önemli bileşen olarak nitelendirilmektedir. Modelleme gerçek yaşamda karşılaşılan problemin bilgisayar ortamındaki ifadesidir; çözümlenme ise bu modeli sağlayan en iyi çözümün elde edilmesini sağlamaktır. Parametre ayarlamanın gelişiminde araştırmacılar öncelikli olarak modellemeyle ilgilenmişlerdir.

Yıllar boyunca yapılan araştırma ve geliştirme faaliyetleri sonucunda özellikle doğrusal programlama problemlerinin çözümünde büyük ilerleme kaydedilmiş olup halen yaygın olarak kullanılmaktadır. Diğer ayarlama problemlerinin çözümüne yönelik yazılımlar olmakla birlikte, bu problemlerin çözümünü en etkin şekilde elde eden çözüm yöntemleri devamlı geliştirilmektedir. Amaç optimum sonuca yaklaşımdır. Optimizasyon algoritmaları ise sistemin

işleyişini ve özelliklerini yansıtan, sistemin içindeki ve çevresindeki diğer sistemlerle olan etkileşimleri kapsayan ifadelerden oluşmaktadır (Türkay, 2006).

Parametre ayarlama amaç, gerçek sistem verileri ile modelden alınan veriler arasındaki farkı en aza indirmektir. Model parametreleri belirlerken gerçek sistemin incelenmesi ve sistem için önemli parametrelerin belirlenmesi gerekmektedir. Model davranışlarının gerçeğini yansıtabilmesi, süreç gösterim yeteneği büyük ölçüde, parametrelerinin doğru belirlenmiş olması ile orantılıdır (Yüceer vd., 2008).

Parametre ayarlama kullanılan teknikler bölümünde detaylı olarak verilen parametre ayarlama çalışmaları incelendiğinde parametre ayarlama işleminde izlenen adımlar birbirine benzerlik göstermektedir. İlk olarak başlangıç parametre çözüm kümesi oluşturulur. Bu çözüm kümesi oluşturulurken başlangıç parametre setini belirleme algoritmaları kullanılarak oluşturulabileceği gibi rasgele de oluşturulabilir ya da sabit verilebilir. Bu çözüm kümesi kullanılarak model çalıştırılır her çözüm kümesine ait uygunluk değeri modele ait uygunluk fonksiyonunu kullanarak hesaplanır. Uygunluk fonksiyonu, çözüm kümesinin kalitesini belirlenen hedefe ne kadar ulaştığı belirlemektedir. Sonraki adımda daha iyi parametre değerlerini elde edecek optimizasyon ya da arama algoritmaları kullanılarak yeni çözüm kümeleri bulunur ve algoritmada test edilmeye devam edilir. Parametre ayarlama işlemini sonlandıracak bitirme kriteri sağlanınca ya da istenen iterasyon sayısına ulaşıncaya kadar bu işlem devam eder. Model parametrelerinin ayarlama sürecini gösteren akış diyagramı Şekil 3.1'de gösterilmiştir (Korkmaz Tan ve Bora, 2017b).



Şekil 3.1 Parametre Ayarlama Süreci Akış Diyagramı (Korkmaz Tan ve Bora, 2017b)

## 3.2. Parametre Ayarlama Kullanılan Teknikler

### 3.2.1. İlk Parametre Setini Belirleyen Metotları

#### 3.2.1.1. Örnek Alma Metotları

Örnek Alma metotları olarak kullanılan Latin-Kare (Latin-Square), Taguchi Dikey Dizini (Taguchi Orthogonal Arrays) ve Latin Hiperküp yöntemleri, tüm bir faktöriyel tasarıma uygun bir şekilde test edilen parametre vektörleri sayısını azaltarak araştırma zamanını kısaltan metotlar şeklinde tanımlanabilir. Örnekendirme metodu oturumunun ortaya çıkışı hangi parametre değerinin en iyi şekilde çalıştığı ve hangisinin en iyi kuvvete sahip olduğu tahminini yaptıktan sonra analizi yapılır. Bu yüzden çoğu örnekendirme metotları genelde model-temelli metotlar için bir başlangıç olarak kullanılır. Tüm parametre uzayında araştırmanın yapılmaması parametre vektörlerinde düşük niteliğe yol açar ve bu da edinilen bilginin özellikle düşük toleranslı algoritmalar üzerine kısıtlı bir bilgiye sebep olur

(Myers and Hancock, 2001; Taguchi and Yokoyama, 1993; McKay et al., 1979; Akgüç, 2010).

### **3.2.1.2. Calibra**

Örneklendirme metodu her bir ardışıkta yeni noktaların örneklendirildiği bir alanı tanımlar. Bu yüzden, bağımsız birer alıcılar olarak kullanılabilirler. Calibra, her bir parametre alanı arasındaki birinci ve üçüncü nicelik temelli tam bir faktöriyel deneyle başlar. Bu verileri kullanarak sonraki ardışıklık için yeni vektörler üç (daraltılmış) seviyede Taguchi Dikey Dizin temelli yöntemle üretilir ve bu prosedür maksimum test sayısına ulaşılan dek tekrarlanır (Adenso-Diaz ve Laguna, 2006).

### **3.2.1.3. Deney Tasarımı (Design of Experiments (DoE))**

DoE temsili deney davranışlarının stokastik çerçevesidir. Yüksek kaliteli sonuçları koruyan bir analiz için gerekli olan deneylerin miktarını en aza indirmeye çalışır. Deneylerin çıkış değişkenleri (cevaplar) ve giriş değişkenlerine (faktörler) sahip olduğu kabul edilir. DoE uç değerleri ve temsili deneyleri birleştirdiği deneylerle iyi organize edilmiş bir yaklaşım önerir, yani "merkezi noktaları" çağırır. DoE ile ortak bir amaç yanıtların kalitesini değerlendirmek karşılaştırma ile faktörleri optimize etmektir. DoE özellikle sayısal optimizasyon problemleri ve manuel parametre ayarlama için bir araç olarak başarıyla uygulanmıştır (Dobslaw, 2010).

## **3.2.2. Model-tabanlı metotlar**

Parametre ayarı için uygulanan, böyle bir yöntem işe yarar bir model oluşturur ve model tahminlerini kullanarak gerçek testlerden bazılarını değiştirerek test sayısını azaltır. Model test tarafından iletilen programlar parametrelerle ilgili verilere dayanır. Ortak bir yaklaşım, bilinmeyen bir parametre vektörü yararını tahmin etmek için bir regresyon yöntemi kullanmaktır. Model daha sonra yararlı parametre değerlerini tahmin eder (Czarn et al., 2004; Ramos et al., 2005; François and Lavergne, 2001)

### **3.2.2.1. Coy'un Prosedürü**

Parametre değerlerini optimize eden yerel bir araştırma prosedürünün izlendiği standart Model-Temelli Metotların bulunduğu en temel uzantılarından biridir. İki evreli prosedürü tanımlar: ilki bir model bulmak ve ikincisi en iyi parametre vektörü tanımlamada spesifik hedef belirler. İlk evre tüm parametre alanı üzerine kurulan tam bir faktöriyel tasarımı içerir. Veriler dikey bağlanım modeline uyması için ve en dik kökenin yolunu belirlemek için kullanılır. İkinci evrede, bu yol izlenir ve yeni vektörler üretilip spesifik evrelerin sayısında bir değişiklik olmayana kadar bulunan en iyi çözüm test edilir. Bu ikinci evrede model güncellenmediğinden bulunan en iyi parametre vektörünün niteliği ilk evredeki modelin doğruluğuna bağlıdır (Coy et al., 2001).

### **3.2.2.2. Sıralı Parametre Optimizasyonu (SPO)**

Modelin tamamıyla güncellendiği gerçek bir multi-evre prosedürü uygulanır. Her biri ardışık yeni vektör kümesi üreterek başlar ve modeli kullanırken bunların kullanım alanını tahmin eder. Daha sonra en iyi tahmin edilen vektörler 'gerçek' kullanım alanını belirlemek için test edilir ve bu ölçülen program değerleri modeli güncellemek için kullanılır. Maksimum test sayısına ulaşıldıktan sonra en uygun modelin belirlenimiyle süreç sona erer. Görüldüğü gibi hem modelin tutarlılığı hem de en iyi parametre vektörünün niteliği kullanılan modele bağlıdır. Yazarlar genelde utility landscape yaklaştırmak için Kriging modellerini kullanmayı tercih etmiştir çünkü sayısal parametrelerle ayar problemleri üzerinde mükemmel performans sağlamaktadır. Karşılaştırmalı bir çalışmada görülür ki SPO, MSA'lar (Bolme, 2011) tarafından bulunan vektörlere kıyasla yüksek nitelikli parametre vektörleri bulmada daha kullanışlıdır. Sonuç alıcı metotlardan çıkarılabilen harmanlanan bilgiler ışığında bu, yüksek nitelikte parametre alıcıları üretir (Bartz-Beielstein et al., 2004; Lasarczyk, 2007).

### **3.2.2.3. Beyaz Kutu (White Box)**

Çok etmenli benzetim modellerinde kullanılmak üzere beyaz kutu (whitebox) yöntemi geliştirilmiştir. Bu modelde, küçük parametre alanları ayrıştırma yöntemleri sunulmuştur. Böylece ayarlama işlemlerinin karmaşıklığı azaltılmıştır. Kullanılan model hiyerarşik bir şekilde alt modellere ayrıştırılmalıdır. Ayrıştırma için çeşitli zamansal evreler kullanılabilir, görev tabanlı ayrıştırma ya da davranış tabanlı ayrıştırırmalardan faydalanılır. Son olarak alt model oluşturup içsel model

ilişkileri analiz edilir. Alt modellerin her biri için bir amaç fonksiyonu ve kritik durumların tespit edilmesi gerekmektedir. Sadece bu kritik durumlarda alt modelleri ayarlamak tek bir benzetimin çalışması için gerekli zamanı azaltacaktır. Ayarlamamanın son aşamasında elde edilen parametre ayarları ile alt programlar birleştirilir. Bu birleştirme işlemi ek ayarlama işlemleri isteyebilir. Bu teknik büyük dikkatle uygulanmalı, her bir alt model problem seti için ayrıştırma (parçalama) işleminden sonra düşükten yükseğe seviyelendirme işlemine tabi tutulmalıdır. Seviyelerin sonuçlarını ayarlamak için bir tanımlama kümesine ihtiyaç duyulmaktadır. Hızlı hesaplama modeline izin verecek optimizasyon tanımlama uygulaması yapısal değişikliğe engel olacaktır. Fakat ayrıştırma ve birleştirme işlemleri oldukça zor işlemler olup özellikle birleştirme işleminde tekrar ayarlamaya ihtiyaç duymakla birlikte yapısal değişikliğe neden olabilmektedir (Manuel et al., 2006).

### **3.2.3. Gözleme Dayalı Metotları**

Gözlem metotlarının ortaya çıkışının ardındaki fikir minimum test sayısı ile verili vektör kümelerinden en iyi parametre vektörünü tanımlamaktır. Bu yöntemde seçilen vektörler test edilir ve başka bir test ihtiyacı olmayana kadar bu süreç tekrarlanır. Böylece ya en iyi parametre vektörünü örnekendirme metoduna kıyasla daha az tüketim çabasıyla tanımlar ya da aynı tüketim çabasıyla daha geniş bir parametre vektörü kümesi araştırır. İkinci durumda bulunan en iyi parametre vektörü niteliği muhtemelen daha yüksek olur ve parametre değerlerindeki değişimlere dayanıklı olan kuvvetleri tahmin etmede daha fazla bilgi sunar.

Gözlem metotları parametre derecelendirmelerine sunulan en eski yaklaşımlardan biridir ve ağırlıklı olarak sistem seçilimi alanı tarafından etkilenmiştir. Burada gereken az bir stokastik benzetimlerle geniş bir rekabet sistemlerinden en iyi opsiyonu seçme amacı vardır (Goldsmann et al., 1991). Parametre vektörleri rekabetçi sistemler ve stokastik benzetim olarak algoritma alanı olarak bilirse de, sistem seleksiyonu alanındaki metotlar, parametre derecelendirmesi yaklaşımları olarak bilinir.

#### **3.2.3.1.F-Race**

F-Race'teki rekabetçi tekniği kullanarak bu gruptaki vektör sayısı, belirli bir koşul gerçekleşene kadar azaltılır. Yani normal dağılımın çoklu-değişkeni kalan vektörlere uyumludur ki bu daha sonra yeni bir popülasyon noktalarını

örneklendirmek için olası yoğunluk işlevi olarak kullanılır. Yeni noktaların tüm bir gözlem ve yönetim prosedürü maksimum test sayısına ulaşılan dek yeniden tekrarlanabilir. Yinelemeli F-Race metodu bir dağılımı örneklendiren birçok-evreli metot olduğundan, ardışık model-temelli bir metodun basit bir formu olarak da görülebilir (Birattari et al., 2010).

F-Race algoritması, problem örnekleri için rasgele belirlenmiş parametrelerle çalışmaya başlar ve bir yarıştaymış gibi kullandığı istatistik testlere göre en kötü olan değerleri siler. Yinelemeli F-Race algoritmasında ise bu süreç belirlenen iterasyon sayısınınca döngü olarak devam eder ve her döngüde bir önceki adımda oluşturulan en iyi aday parametre değerleri kullanılarak yeni parametre kümeleri kullanılır.

### **3.2.4. Arama Tabanlı Metotlar**

#### **3.2.4.1. Kara Kutu (Black Box)**

Kara kutu arama yöntemi, giriş ve çıkış değerleri arasındaki ilişkiyi tahmin ederek elde etmeye çalışır. Popüler black box yöntemleri; gradyan tabanlı arama yöntemi, stokastik yaklaşım yöntemi, örnek yol (samplepath) optimizasyonu, tepkisel yüzey optimizasyonu (response surface optimization) ve sezgisel (heuristic) arama yöntemleridir. Black box'ın bir avantajı prosedürün bu arama yöntemlerinden hangisini kullandığı ayarlama için önemli değildir. Bu avantaj aynı zamanda eksiklidir. İçyapıyla ilgili bilginin olmaması parametre uzayının büyümesi ile birlikte kısıtlı zamanda yeterli arama yapılamamasına neden olmaktadır. Maliyeti düşürmek için çalışma zamanının da kısıtlamalar yapılması gerekmektedir. Buda istenen sonucun çıkmasını engellemektedir. Sadece giriş parametreleri üzerinde black box yöntemi ile ayarlama uygulanabilmektedir (Manuel et al., 2004).

#### **3.2.4.2. Metropolis (Monte Carlo)**

Bu algoritma rasgele yürüyüş mantığına dayanır. Metropolis algoritması birden çok minimumlu fonksiyonun global minimumu bulabilmesidir. Metropolis Algoritması, N boyutlu uzaydaki bir noktayı argüman kabul eden fonksiyon minimumu bulma problemi için kullanılabilir. Bir önceki algoritmaya göre üstünlüğü, yerel minimumlarda takılmayıp diğer minimumlara doğru aramaya devam edebilmesidir. Minimum arama sırasında, fonksiyonun argümanı olan X, N

boyutlu uzayında hareket ettirilir,  $X_0, X_1, \dots, X_i, X_{i+1}, \dots, X_S$  noktaları boyunca olan hareket sonunda aranan minimum değerini veren nokta olması beklenir.

Bu yöntem kesintisiz parametre uzayında iyi bir performans sergiliyor. Eğer model stokastik (değişken, rastlantısal) ise o zaman model kullanışlı olmamaktadır. Bu modelde parametredeki küçük değişiklik büyük değişikliğe neden olacaktır (Sallans et al., 2011).

### *Metropolis yönteminin kullanıldığı modelleme çalışması;*

İki etkileşimli pazarda sınırlı rasyonel etmen davranışlarını inceleyen bir model geliştirilmiştir. Finansal ve tüketici pazarlarının birleştirildiği ayrık zaman modeli işlenmiştir. Model, heterojen tüketici, mali tüccar ve üretim firmalarından oluşur. Model gerçek piyasaların bilinen deneysel özellikleri ile karşılaştırılarak doğrulanır. Daha iyi davranışlar için parametrelerin ayarlanması gerekmektedir. Bunun için Metropolis algoritması kullanılmıştır (Sallans et al., 2011).

#### ***3.2.4.3. Uyarlanabilir Değer İzleyici (Adaptive Value Tracker (AVT) )***

Bu yöntemde, her parametrenin verilen bir arama uzayında aranan devingen bir değeri (yani zamana bağlı değişen bir değer) bulup izleyebilen bir Uyarlanabilir Değer İzleyicisi (Adaptive Value Tracker - AVT) bulunmaktadır. İzleme işlemi, AVT'nin ortamından gelen muhtemelen aranan değere doğru götüren ardışık geribildirimler sayesinde sağlanmaktadır. Bu işlemler sezgisel ve rasgele gerçekleştirildiği için her zaman istenen değerler elde edilememektedir (Lemouzy et al., 2011; Brax et al., 2013).

### **3.2.5. Etmen Tabanlı Metotlar**

#### ***3.2.5.1.3 Aşamalı Etmen Tabanlı Algoritma***

Ekoloji modelinde parametre ayarlama işlemi gerçekleştirmektedir. Bu çalışmada zeki bir etmen modele ait parametreleri ayarlama görevlendirilir. Etmenin ilk olarak değişkenlerin ilişkilerini belirlendiği bir matris oluşturmak. Sonraki adımda, Farklı parametreler analiz edilir. Son adımda model tekrar tekrar çalıştırılarak mevcut eksikler belirlenerek giderilmeye çalışılır (Pereira et al., 2008).

### **3.2.5.2. Çoklu Etmen Sistemi ve Parametre Ayarlama Yöntemi**

Çoklu etmen sistemi kullanarak parametre ayarlama işlemi yapılmaya çalışılmıştır.

Bu yöntem 5 adımda gerçekleşmektedir:

- Modeldeki etmen davranış verilerinin toplanması.
- Benzer durumdaki insan verilerinin yine model yardımı ile toplanması.
- İnsan katılımcıların verileri değerlendirmesi.
- Verilerin yeniden düzenlenmesi.
- Elde edilen verilerin karşılaştırılması, mevcut davranışların yorumlanması.

Üç küme kullanılmıştır:

- 1) Etmenler ve insan davranışlarının tutulduğu küme CM,
- 2) Haberleşme etmenleri ayrı bir kümede tutulur CA,
- 3) Sadece insan katılımcıları bulunduğu küme CH,

Etmen davranışları insan katılımcıların model verilerindeki davranışlarla kıyaslanarak hesaplanır. Uygun davranışlar o etmene ait parametreler ile karşılaştırılarak parametreler belirlenir. Bu çalışmada etmen davranışları parametre ayarlamada kullanılır. Bu işlemler istenen davranışlar elde edilinceye kadar test edilir. Büyük boyutlu sistemlerde uygulanması zor ve zaman alıcıdır ( Darty et al., 2015).

### **3.2.5.3. Tek Etmenli Parametre Ayarlama**

Sürgülü Puzzle Modelindeki parametrelerin ayarlanması için geliştirilmiştir. En iyi parametre değerini ayarlama işlemini tek bir etmen yapmaktadır. Geliştirilen algorithmada, en iyi parametreyi bulmaya çalışan karar dizileri kullanılmıştır. Karar dizileri içinde parametre değerleri, uygunluk fonksiyonu ve arama algoritması

bulunmaktadır. Karar dizileri içinde birbirinden farklı arama algoritmaları mevcuttur. Tüm arama algoritmaları döngü içinde arama işlemi yapar. Döngü işlemi bir çözüm bulununcaya kadar tekrar eder. Her algoritma örneği tamamen bağımsız bir arama gerçekleştirir. Örnekler arasında bellek paylaşımı mevcut değildir. İletişim bir çözümün sonlandığını, mevcut problemin bulunamadığını belirten mesajlarla sınırlandırılmıştır. Farklı algoritmaların aynı anda kullanılması ve çözümü bulununcaya kadar sınırsız döngü kullanımı programın performansını olumsuz etkilemektedir ( Valenzano et al., 2010).

#### **3.2.5.4. Çoklu Etmen Sistemde Parametreler (Parameter Multi Agent System (PAMAS))**

Çoklu Etmen Sistemde Parametre (Parameter Multi Agent System (PAMAS)) adı verilen yöntem Deniz Gözetleme Sisteminde Parametre optimizasyonunu gerçekleştirmektedir. Bu çalışmada kendini uyumlandıran işbirlikçi var olan parametrelerin yerlerini öğrenmiş etmenlerden oluşan çok etmenli bir yöntem anlatılmaktadır. Bu yöntemde parametre sayısının bilinmiyor olması ve bu sayıların sistemin yürütülmesi esnasında değişebilir olması gibi bir özellik söz konusudur. Bunun sonucu olarak sistemin, anlık olarak parametre sayısını ve parametre değişimini öğrenmesi gerekir. Bir yandan parametre etmenleri Adaptive Value Tracker (AVT) kullanma yetkisine sahiptir ve bu araç bir arama uzayında bir örnek değeri aramak için bir etmen tarafından kullanılır. Böylece her bir parametre etmen yönettiği parametrenin değerini AVT ile arar. Öte yandan parametre etmeni nümerik değer hesaplama yeteneğine sahiptir ve bu değer parametrenin değerini ayarlamak için parametre etmenleri tarafından kullanılır. Bu özellikler ve araçlar göz önüne alındığında parametre etmenlerinin etkileşimde buldukları ortamda ayarlandığı görülür (Brax et al., 2013).

#### **3.2.6. Meta-Sezgisel Metotlar**

Parametre ayarlama işleminde parametre uzayı içinde optimum değerleri bulmak için sezgisel arama algoritmaları ve optimizasyon algoritmaları kullanılabilir. Arama işlemine rehberlik eden yöntemleri kullanan MSA, özellikle son yıllarda yapılan araştırmalar büyük boyutlu ve karmaşık yapıdaki gerçek hayat problemlerinin çözümünde başarılı olduğunu kanıtlamıştır. Bu algoritmaların amacı, veri uzayını etkili bir şekilde aramak ve optimuma en yakın çözümleri hızlı bir şekilde bulmaktır. Kolay, anlaşılır ve farklı problemlere uygulanabilir olması nedeni ile günümüzde yaygın olarak kullanılmaktadır. Meta-sezgisel yöntemler,

esin kaynağı (doğal yaşam veya yapay zekâ), kullandığı başlangıç çözümü (popülasyon veya tek birey), kullanılan uygunluk fonksiyonu (dinamik, statik), komşuluk yapısı (tekli, çoklu) ve hafıza durumu (hafızalı, hafızasız) gibi kriterlere sınıflandırmaya tabi tutulabilir (Şahin, 2014).

### **3.2.6.1. Genetik Algoritmalar**

Geniş çözüm uzaylarının klasik yöntemlerle taranması hesaplama zamanını ve maliyeti artırmakta olup istenen çözümlere erişmek çoğu zaman mümkün olmamaktadır. Genetik algoritmalar (GA) kısa sürede kabul edilebilir doğrulukta optimuma yakın sonucu bulabilir. Genetik algoritmalar, biyolojiden esinlenerek oluşturulmuştur problem fonksiyonlarını optimize eden evrim algoritmalarıdır. GA parametreleri, biyolojideki genleri temsil ederken, parametrelerin toplu kümesi de kromozomu oluşturmaktadır. GA'ların her bir ferdi kromozom olarak temsil edilen popülasyonlardan oluşur. Popülasyonun uygunluğu değeri, belirli kurallar çerçevesinde maksimize veya minimize edilir. Her yeni nesil, rasgele bilgi değişimi yani çaprazlama gibi işlemler ile oluşturulan veri setleri içinde hayatta kalanların yani en iyi uygunluk değerine sahip olanların birleştirilmesi ile oluşur.

Uygulamadaki ilk adım, ilk popülasyonun oluşturulup, uygunluk değerinin bulunmasıdır. Daha sonra mevcut oluşturulan nesile, çoğalma, çaprazlama, mutasyon gibi genetik işlemler uygulanır. Her nesil için uygunluk değeri hesaplanır. Bu durum belirlenen sonlanma şartı sağlanıncaya kadar devam eder (Calvez and Hutzler, 2005; Başak et al., 2009; Salwala et al., 2010; Imbault and Lebart, 2004; Bolme et al., 2011; Saraçoğlu ve Demirören, 2008; Deliktaş ve Türker, 2005).

Genetik algoritma adımları aşağıda açıklanmıştır;

Arama uzayındaki tüm olası çözümler bir grup çözüm seti olarak belirlenir. İlk popülasyon rasgele seçilir ve başlangıç popülasyonu oluşturulur.

- Her bir kromozom için uygunluk değeri hesaplanır. Bulunan uygunluk değeri kromozomun çözüm kalitesini gösterir.
- Bir grup kromozom belirli bir olasılığa göre rasgele olarak seçilir. Seçilen kromozomlar çaprazlama ve mutasyon işlemlerine tabi tutulur.
- Oluşan yeni popülasyon eski popülasyon ile yer değiştirir.

- Durdurma koşulu sağlanana kadar yukarıdaki işlemlere devam edilir. En uygun olan dizi çözüm olarak seçilir (Angeline, 1995).

Genetik Algoritmanın Parametre Ayarlama Kullanımı İçin Yapılan Modelleme Çalışmaları;

GA, hayvanların kalın bağırsağında yaşayan bir Escherichiacoli isimli bakterinin fosfotransferaz ve glikoliz sistemlerine yönelik bir benzetim ve modelleme çalışmasında parametre ayarlama işlemi için uygulanmıştır. Bu çalışmada, enzim ve molekül karmaşık sisteminin modellendiği bir çalışmadır. Bu model hücre davranışlarındaki değişmelerin ve ortama uyum sağlamasını gözlemlemektedir. Bu benzetim modelinde etmenlerle temsil edilen 25 molekül kullanılmıştır. Sistem 2200 etmenden oluşmaktadır. Hücreler içinde bulunan enzim molekül davranışları olası hücre davranışları ve istenen davranışların elde edilmesine yönelik parametre ayarlama işlemi yapılmıştır. Ayrıca yine GA kullanılarak karınca kolonisi modeli üzerinde parametre ayarlama işlemi yapılmıştır. Çalışma sadece bu iki etmen tabanlı model üzerinde sınırlı kalmıştır (Calvez and Hutzler, 2005).

GA, MOS transistörleri benzetim modelinde parametre ayarlama işlemi için GA kullanılmıştır. Parametreler MOS BSIM3V3 modelinde yer alan C35 üretmek için kullanılan 0.35µm teknoloji ile belirlenen SPICE parametre verileri kullanılmıştır. Elde edilen parametre sonuçlar ile gerçek parametre sonuçlar karşılaştırılmıştır ve yöntemin başarısı kanıtlanmıştır (Başak vd., 2009).

GA, Betonarme kiriş tasarımında ilgili parametrelerin ayarlanması işleminde kullanılmıştır. Tasarım için tek açıklıklı dikdörtgen kesitli betonarme kirişin boyutları gibi parametreler minimum maliyet için ayarlanmaktadır. Genetik algoritma ile elde edilen sonuçlar yinelemeli olarak elde edilen grafiksel sonuçlar ile karşılaştırılmaktadır. Her iki çözüm ile elde edilen sonuçların parametre ayarlama işleminde GA'nın başarısını göstermektedir (Deliktaş vd., 2005).

GA, GPU Kullanılan Oyunlarda Parametre Ayarlama yönelik çalışmada kullanılmıştır. Oyun teknolojisinde gerçek ortam tasarımı oldukça önemlidir. Yangının yayılması, suyun akışı gibi doğal olayları gerçeğine uygun bir şekilde tasarlamak gerekiyor. Bu çalışmada iki önemli uygulama vardır; hücresel otomata (CA) benzetim modelinin geliştirilmesi ve GA kullanarak parametre ayarlama işlemi. CA, yangının yayılmasını ve suyun akışını modellemede kullanılan bir

yöntemdir. Benzetimde yangının yayılması için hücre özellikleri, sıcaklık, kütle, hasar, nem gibi özellikler kullanılmaktadır. Akış benzetimindeki hücre özellikleri ise yükseklik ve akışkanlıktır. Etkin bir şekilde optimal erişim desenine göre cihaz hafızasını kullanmak için, bellekler her bir hücre yerine kullanılmış, hücrenin bir özelliğini yönetmiş ve yer ayırmıştır. Her bir özelliğin kendisi ile ilişkili değişkeni var olması nedeni ile her bir değişken için ayrı yerler tahsis edilir. Bu yaklaşım, ayrıca daha az bellek erişimini gerektirmektedir. Dolayısı ile genetik algoritmada mutasyon ve çaprazlama işlemlerini daha hızlı gerçekleştirmesi için de yardımcı olmaktadır. Her bir iş parçacığı CA'da bir hücreye tekabül eder. İlk olarak, her bir iş parçacığı, hücreler arasındaki geçiş kurallarını belirlemek için herhangi bir olay ile ilgili verilerle yüklenir ve bellek paylaşımını gerçekleştirir. Gerekli tüm veriler yüklendikten sonra hücreler arası geçiş kuralları yapılmaktadır. Çekirdek blok boyutu CA boyutundan daha küçükse, bu aşamada çekirdek bloğunun dışında bazı veriler fazladan bellek erişimi sonucuna ulaşır. Her bir blok için maksimum iş parçacığı sayısı 512 olduğundan haliyle hız düşecektir. Her bir işlemcinin kayıt ve paylaşılan belleğin maksimum sayısı da ayrıca sınırlıdır. Bir hücre için geçiş kuralları uygulanmaktadır, her iş parçacığından sonra hücreler arasında geçiş kuralları performansı bitirecektir. Daha sonra, her bir iş parçacığı GPU küresel belleğe, güncellenmiş verileri geri yazar.

GA, belirli bir senaryo ile ilk hücre özelliklerini eşleştirmek için gerekli arama işlemlerini yapmaktadır: yangın yayılması senaryosuna ve su akış senaryosuna çözüm bulmak için yapılan arama. İlk aramadan elde edilen tüm bireyler senaryo ile ilgili parametreleri oluşturur ve bütün CA'nın ilk yapılandırmasını temsil eder. Her iki arama da aynı şekilde yapılır. Her bir birey için, her bir ilgili özellik için ilk veri rasgele başlatılır. Bundan sonra her bir kuşak için, her bireyin uygunluk değeri değerlendirilir, bireyler uygunluk değerine göre sıralanır. Daha sonra seçkin seçim ve yeni nesil oluşturmak için mutasyon ve çaprazlama yapılır (Salwala vd., 2010).

GA ve Tavlama Benzetimi Algoritması, Destek Vektör Makine Tasarımında Parametre Ayarlama çalışması için kullanılmıştır. Destek Vektör Makineleri (Support Vector Machines) (SVM) hem matematiksel olarak iyi finanse edilmiştir, hem de çok sayıda gerçek dünya uygulamalarında verimli olarak kullanılmıştır. Fakat sınıflandırma sonuçları modelin parametre değerlerine bağlıdır. Bu çalışmada GA ve tavlama benzetim algoritmaları kullanılmıştır ve değerlendirilmiştir. Sezgisel optimizasyon teknikleri optimum sonuca yakın çözümler üretir. GA ve tavlama benzetim tekniklerinin ikisi de birbirine yakın sonuçlar vermektedir. GA

tavlama benzetime göre biraz daha hızlıdır. GA'ya bir başlangıç popülasyonu, çaprazlama ve mutasyon oranı gibi parametre değerlerinin ayarlanmasında ihtiyaç duyulmaktadır. Tavlama benzetim de daha az parametre vardır. Bu yöntemlerin temel dezavantajı maliyettir (Imbault ve Lebart, 2004).

GA, Yüz tanıma algoritmalarında parametre ayarlama işlemi için kullanılmıştır. Bu çalışmada amaç insan kullanıcıların tecrübelerine dayanarak yapmaya çalıştığı parametre ayarlama işlemi GA'nın yapmasının sağlanmasıdır. GA'nın en önemli avantajı optimizasyon işlemi durmak nedir bilmeksizin yapmaktadır. GA çok farklı çalışma alanında ve en karmaşık parametre yapılarında bile en uygun çözümü bulabilmektedir. Doğrulama, uygunluk fonksiyonuna bağlıdır. Yüz tanımda Local Region Principal Components Analysis (LRPCA) algoritmaları kullanılmaktadır. Yüz tanıma çalışmasında sıklıkla kullanılan LRPCA algoritmasının doğrulanmasının yanı sıra GA'nın daha iyi sonuçlar çıkardığı kanıtlanmıştır (Bolme et al., 2011).

### ***3.2.6.2.Evrin Algoritması***

Evrin Algoritması (EA) etkinliğini yaptığı deneylerle göstermiştir. EA'da kullanılan bireyler tasarım tabakası üzerinde sayısal değer vektörleri olarak temsil edilir. Bu değerlerin her biri Bazal EA ayarlı olan parametrelerden birine karşılık gelir. Bu tür bir vektör yararını değerlendirmek için, verilen parametre değerleri kullanılarak birkaç kez Bazal EA çalıştırılır. (Meta)uygunluk olarak temsili ve yardımcı programını kullanarak, temelde herhangi bir evrim algoritması eğer sadece bireylerin gerçek değerli vektörleri ile başa çıkabilirse bir meta-EA olarak kullanılabilir. Ayrıca Hansen tarafından önerilen Kovaryans Matrisi Adaptasyonu (CMA) ile Evrim Stratejisi (ES) birleştirilerek yeni bir algoritma önerilmiştir. Bu seçim Evrimsel Stratejileri sayısal ve optimize olarak motive edilir. Geliştirilen CMA-ES algoritması standart ES'nin iyileştirilmiş halidir (Grefenstette, 1986).

### ***3.2.6.3.Parametrenin İlişki Tahmini ve Değer Kalibrasyonu (Relevance Estimation and Value Calibration of Parameter-REVAC)***

Parametrenin İlişki Tahmini ve Değer Kalibrasyonu, optimal parametre vektörü bulmak için gelecek vaat eden bir algoritmadır. Özünde, REVAC, bir toplum tabanlı stokastik arama yöntemidir. Nüfus tabanlı EA parametre vektörleri birer bireydir. Algoritmanın sona ermesinden sonra parametre başına tahmini dağılımlar, yardımcı program modelini temsil etmektedir. Özünde, REVAC bir

evrimsel algoritmanın belirli bir türüdür. Kullanıldığı alanlarda mümkün olan yoğunluk fonksiyonunu popülasyona yaklaştırır. Parametre etkileşimleri dolayısıyla kördür ve parametreler koordinatlarına ayrıştırılmadığından bu yoğunluk fonksiyonu oldukça basittir. Ama farklı parametreler ve her parametreyi ayarlayarak maliyetlerinin hassasiyetini ve ilgisini analiz etmek için de kullanılabilir. Ayrıca sırayla daha etkin yarar değerleri istatistiği ile başa çıkmak için REVAC yarış (race) tekniği ile birlikte kullanılmıştır. REVAC için bu eklenti, REVAC algoritmasının parametre değerlerini bulmayı amaçlamaktadır. Böyle gelişmiş bir EA'nin algoritması çok daha iyi sağlam parametre değerlerini bulmanın mümkün olduğunu göstermektedir (Smit and Eiben, 2009; Smit and Eiben, 2010a; Smit and Eiben, 2010b; Nannen and Eiben, 2006; Nannen and Eiben, 2007a; Nannen and Eiben, 2007b).

#### ***3.2.6.4. Karınca Kolonisi Algoritmaları (Ant Colony Optimization Algorithm (ACO))***

Dorigo ve arkadaşları tarafından önerilmiş en yeni MSA'lardan biridir. Algoritma gerçek karınca kolonisinin davranışları üzerine tasarlanmıştır. Günümüze kadar ACO'nun yeni modelleri ortaya çıkmış ve bu modellerin özellikle ayrık optimizasyon problemlerinin çözümüne uygulanması konusunda çeşitli çalışmalar yapılmıştır. Karınca koloni optimizasyon algoritması, yukarıda tanımlanan gerçek karınca kolonisinin yapmış olduğu doğal optimizasyon işleminin yapay bir versiyonudur (Salwala et al., 2010).

#### ***Karınca kolonisi Algoritmalarının Kullanıldığı Modelleme Çalışmaları:***

ACO, gerçek karıncaların yuvaları ile yiyecek noktaları arasındaki en kısa yolu bulma kabiliyetlerinden esinlenilerek geliştirilmiştir. Alternatif yolların söz konusu olduğu durumlarda karıncalar, öncelikle bu yollara eşit olasılıkla dağılırlarken belli bir süre sonra en kısa olan yolda yoğunlaşmaktadır. Zaman geçtikçe tüm karıncaların en kısa olan yolu kullandıkları görülmektedir. Bunu yaparken önceki geçişlerden yollarda kalan feromon izlerinden faydalanmaktadırlar. Temel kural, feromon miktarının yoğun olduğu yolun tercih edilme olasılığının da yüksek olmasıdır. Görme duyuları çok gelişmemiş olan karıncalar yol tercihlerini feromon izlerine göre yapmaktadırlar (Gambardella and Dorigo, 1995; Dorigo and Gambardella, 1997; Stützle and Hoos, 1997; Bullnheimer et al., 1997; Maniezzo et al., 1994; Stützle and Hoos, 1998; White et al., 2000; Calvez and Hutzler, 2007a).

ACO, karesel atama problem çözümünde kullanılmıştır. Karesel Atama ile aday bölge ve atanacak tesisler belirlenir. Bu iki değer bir birine eşit olmalıdır. Toplam maliyeti minimize ederek her bir aday bölgeye bir tesis atanmaktadır. Atama işlemi hiçbir aday bölge veya tesis boşta kalmayacak şekilde yapılmaktadır (Maniezzo et al., 1994; M. Gambardella et al., 1997; Stützle ve Hoos, 1997).

ACO, araç rotalama problemini çözmek için kullanılmıştır. Araç rotalama problemi, perakendecilerin verdiği sipariş miktarı, dağıtımın ne kadar süreceğini ve aracın ne kadar sürede depoya geri dönebileceğini belirlemektedir. Amaç aracın en etkin rotayı izlemesini sağlamaktır. Bir depodan perakendecilere minimum uzunlukta ve zamanda rotayı bulma problemidir (White et al., 1998).

### ***3.2.6.5. Uyarlanabilir Atomik Parçalara Ayırma (Adaptive Dichotomic Optimization (ADO))***

Uyarlanabilir atomik parçalara ayırma (Adaptive Dichotomic Optimization (ADO)) yaklaşımı Calvez ve Hutzler tarafından önerilmiştir. Etmen tabanlı modeller çok sayıda parametre ile karakterize edilirler. Birçoğu gerçek sistem bilgileri ile değerlendirilemezler. Amaç en uygun parametre setinin bulunmasıdır. Bu çalışmada parametre alanları (space) mevcuttur. Parametre alanı içindeki gezinti, karınca koloni sisteminden esinlenerek oluşturulmuştur. Bu algoritmaya göre ya gruplaşma ya da bazı özelliklere göre bölünme vardır. Amaç gerçeği yansıtmak için etmen tabanlı benzetimlerden faydalanmaktır. Bu çalışmada parametrelere farklı ayarlar uygulanarak etmenlerin farklı etkileşimlerde bulunmasını sağlamak önerilmektedir. ADO için karınca kolonisinden ilham alınmış ve karıncaların yiyecek arama yaklaşımı uygulanmıştır. Genetik algoritmalarından elde edilen sonuçlarla karşılaştırıldığında neredeyse onun kadar iyi sonuçlar ürettiği ortaya çıkmıştır. Bu yöntemi uygularken işlemler farklı bilgisayarlara dağıtılarak daha hızlı sonuç alınması sağlanabilir. Geniş parametre uzayında rasgele arama yapan optimizasyon teknikleri ile karşılaştırıldığında ayrıştırma yöntemini kullandığı için daha hızlı ve optimale yakın sonuçların elde edildiği gözlenmiştir. ADO yönteminin genetik algoritmaya göre iki avantajı vardır. Aynı anda birden çok modeli farklı bilgisayarlara dağıtarak çalıştırabilmektedir. Bu nedenle daha hızlıdır. Diğer bir avantajı parametreler görsel haritalandırılmış alanlardan oluşmuştur. Seyrek aralıkların olduğu bölge daha az tercih edilen parametrelerin olduğunu, yoğun aralıkların olduğu bölge daha çok tercih edilen parametrelerin olduğunu göstermektedir. Görselleştirme teknikleri kullanılarak bu

özellikten faydalanmak mümkündür. Fakat bu yöntem büyük modellerde test edilmemiştir (Calvez and Hutzler, 2007b).

### ***3.2.6.6. Yapay Sinir Ağları (Artificial Neural Networks (ANN))***

ANN tahmin, optimizasyon ve sınıflandırma problemlerinde kullanımı son yıllarda yaygınlaşmıştır. Sinir ağları insan beyninin çalışma prensibine uyarlanan bir algoritmadır. İnsan beyninde olduğu gibi sinir ağlarında da sinir hücreleri kendi aralarında farklı şekillerde bağlanarak ağları oluştururlar. Böylece oluşan ağlar, hafızaya alma ve veriler arasında ilişki bulma, öğrenme gibi birçok işlevi yerine getirme kapasitesine sahiptir. Matematiksel algılayıcı, biyolojik sinir hücresi davranışından esinlenerek tasarlanmıştır ( Doğru, 2015; Banjanovic-Mehmedovic and Karic, 2011; Dobslaw, 2010; McCulloch and Pitts, 1990).

#### ***Yapay Sinir Ağları Kullanan Modelleme Çalışmaları:***

ANN, jeofizik modelleme ve benzetimde yaygın olarak kullanılan GA ve Levenberg-Marquardt (LM) optimizasyon tekniklerine ek olarak kullanılmıştır. ANN yöntemle yeraltı boşluklarına ait rezidüel gravite anomalisine ait parametrelerin ayarlanması amaçlanmıştır. İleri Beslemeli Geri Yayılımlı (İBGY) sinir ağı genellikle ters çözüm problemlerinde kullanılan bir yöntemdir. Bu yöntemle birlikte, İleri Kademeli Geri Yayılımlı (İKGY) ve Doğrusal Olmayan Otoresif (DOO) sinir ağı, parametre ayarlama işlemi için denenmiştir. Ayrıca global bir yöntem olan GA ve geleneksel bir yöntem olan LM algoritması ile rezidüel anomaliden derinlik ve yarıçap parametreleri hesaplanmış ve sonuçlar karşılaştırılmıştır.

ANN'nin, LM ve artık jeofizikte sıkça kullanılan GA gibi jeofizik problemlerinin ters çözümünde başarılı sonuçlar verdiği görülmüştür. ANN, GA ve LM yöntemlerinin çalışma mantığı çok farklı olmasına rağmen aynı problemlere uygulanabilirliği ve jeofizik problemlerin çözümünde DOO ve İKGY ağlarının da, İBGY yöntemine ek olarak kullanılabileceği görülmüştür (Doğru, 2015).

ANN, insanların robotları montajlama aşamalarından esinlenilerek gerçekleştirimi amaçlandığı bu çalışmada kullanılmıştır. Montaj işlemi, montaj şeklini öğrenme ve montajlama yapma işlemlerinden oluşmaktadır. İki tür öğrenme vardır; ilk olarak yapay sinir ağı tabanlı öğrenmedir, bu öğrenme, robot davranışlarını iyileştirmeyi ve durum parametrelerini etkin kullanmayı içerir.

Parametre vektörü  $\theta$ 'nın, öğrenme sistemi tarafından ayarlanması gerekmektedir. Genlik ve frekanslarda titreşim veri derlemesi deneyler sırasında toplanır ve öğrenme algoritması için bilgi kaynağı olarak kullanılır. İkincisi; denetimsiz öğrenmedir ve bir hedef noktasına ulaşmak için kullanılır. Sistemde hata sinyali alındığında davranışlar yeniden planlanır. Robot düzeneğini akıllı sistemlerle kontrol etmek mümkündür (Banjanovic-Mehmedovic ve Karic, 2011).

ANN, Gezin Satıcı Probleminde parametre ayarlama işlemi için kullanılmıştır. MSA'lar, karmaşık sistemlerin büyük parametre setlerinin optimizasyon problemleri için uygulanabilmektedir. Her bir MSA çalıştırılmadan önce ilk parametre seti rasgele tanımlanır. Gerekliğinde, meta-sezgiselde başlangıç parametre seti ayarlama yöntemlerinden faydalanmak mümkündür. Bu çalışmada meta-sezgisel yöntemle parametre ayarlamak için yeni bir çerçeve sunulmuştur. Metodoloji 4 aşamadan oluşur, 1. Problem tanımlanır, 2. Veriler eğitilir, 3. Parametre erişimi, 4. İşletme. Bu yöntem etkili parametre ayarlama yöntemi olan Design of Experiments (DoE) ve iyi bir parametre ayarlama tekniği olan ANN tekniğini birleştirmektedir. DoE var olan parametre seti içinde optimum sonuca yakın olduğu düşünülen parametre alanı belirleyerek ANN'nin bu alan içinde en iyi parametre setini bulmasını sağlar. (Dobslaw, 2010).

### ***3.2.6.7. Parçacık Sürü Optimizasyonu (PSO)***

Yerel en iyi çözümler elde eden algoritmaların aksine, global arama özelliğine sahip ve kuş ve balık sürülerinin davranışlarından esinlenerek geliştirilmiş, sürü tabanlı optimizasyon algoritmasıdır. Eberhart ve Kennedy tarafından önerilen bu yöntem, fonksiyon optimizasyonu, bulanık sistem kontrolü, yapay sinir ağı eğitimi gibi birçok alanda uygulanmış ve başarısı kanıtlanmıştır. Algoritma rasgele çözümler içeren bir popülasyonla başlatılır ve her iterasyon da nesilleri güncelleyerek en optimum çözüme ulaşmaya çalışır. PSO'da Parçacıklar olası çözümlerdir. Parçacıklar o anda optimum uygunluk değerine sahip parçacığı izleyerek veri uzayında dolaşır. PSO'nun klasik optimizasyon tekniklerinden en önemli farklılığı türev bilgisine ihtiyaç duymamasıdır. Diğer MSA'lara göre PSO algoritmasının gerçekleştirimi, ayarlanması ve sahip olduğu parametre sayısının az olması sebebiyle daha kolaydır (Çavuşlu vd., 2010).

### 3.2.6.8. Biogeography-Based Optimization (BBO)

BBO, Asenkron Motorların (ASM) elektriksel eşdeğer devre parametrelerinin ayarlanması için kullanılmıştır. Hedef parametrelerin minimum hata oranı ile en hızlı şekilde belirlenmesidir. BBO algoritması GA algoritmasından esinlenilerek geliştirilmiştir. Bu algoritma bir sezgisel algoritma olup arama uzayında birden çok noktada arama gerçekleştirebilmektedir. BBO ve GA algoritmasında uygunluk fonksiyonu olarak, asenkron motorların nominal, kalkınma ve devrilme momenti eşitlikleri kullanılmıştır. Eşdeğer devre parametrelerinin belirlenmesi 1.1kW ve 0.37kW gücünde iki asenkron motor üzerinde gerçekleştirilmiştir. Elde edilen sonuçlar GA sonuçları ile karşılaştırılmıştır.

Biyocoğrafya, türlerin doğadaki dağılımını inceler. Her bir ada bir çözümü temsil eder ve yaşamla ilgili parametrelere sahiptir: her bir çözüm uygunluk indeksi değişkeni (SIV) olarak adlandırılır. Her çözüme kendi yaşam form endeksi (HIS) denir. BBO'da, H yaşam alanı, rasgele başlatılan N (SIV) vektörüdür ve daha sonra, hedef için minimum göç ve mutasyon adımını izler.

BBO da, GA'da çeşitliği arttırmak için kullanılan mutasyon oranı BBO algoritmasında da kullanılmıştır. Mutasyon operatörü E, mutasyon oranı m olarak temsil edilmektedir.

BBO Algoritması;

Parametreler:

P: Popülasyonun büyüklüğü.

G: Maksimum nesil.

Keep: Elitism parametresi.

Pmod: Ada değiştirme imkânı.

1: Her olası canlı türü ve gelişigüzel seçilmiş ilk popülasyonların sayısı.

2: Popülasyondaki her birey için uygunluğun kontrolü.

- 3: Bitiş koşulu sağlanmazsa.
- 4: Dizideki en iyi yaşamın kaydedilmesi.
- 5: Her yaşam için,  $S$ ,  $\lambda$  ve  $\mu$  türlerini HSI ile eşleştirilmesi.
- 6: Göç oranlarına dayalı muhtemel bir göçmen adası seçimi.
- 7: 6. adımda seçilen adadan gelişigüzel seçilen SIV'ların göç ettirilmesi.
- 8: Her mutasyon algoritması için ayrılan popülasyon değerlerinden en kötü yarısının değiştirilmesi.
- 9: Popülasyonda yer alan bireylerin uygunluğunun denetlenmesi.
- 10: Popülasyonun iyiden kötüye sıralanması.
- 11:  $G = G + 1$
- 12: Döngünün sonlandırılması.

İki farklı motor için parametre ayarlama işlemi olan biyocoğrafya tabanlı optimizasyon metodu kullanımı tavsiye edilmiştir.

BBO'dan elde edilen sonuçlar GA'dan elde edilen sonuçlar ile karşılaştırılmıştır. Benzetim sonuçları ile hedeflenen yöntemle geliştirilen benzetim sonuçlarındaki hata değerlerinin genetik algoritma sonuçlarından daha düşük olduğu bulunmuştur ve parametresinin ayarlama BBO yönteminin daha iyi olduğu görülmüştür (Saraçoğlu vd., 2013).

### **3.2.6.9. Girdap Optimizasyon Algoritması (GOA)**

Doğal girdaplardan esinlenilerek geliştirilmiş bir sezgisel optimizasyon yöntemidir. Algoritma, çözüm uzayına ilk aşamada rasgele bırakılan yapay işlem Parçacıklarının optimum çözümü aramasına dayanmaktadır. Girdap değerine ( $v$ ) sahip  $N$  Parçacık, verilen bir uygunluk fonksiyon için belirli ilk değerler ve parametreler eşliğinde optimum çözümü bulmaya çalışır. İteratif ilerleyen çözüm adımlarında, her parçacığın sahip olduğu uygunluk değeri değerlendirilmekte ve  $v$

değeri ortalamaya göre daha iyi değere sahip olan Parçacıkların girdap statüsüne ulaşması beklenmektedir. Yine her iterasyon sonunda girdap statüsüne gelemeyen Parçacıkların sayısı eleme katsayısından ( $e$ ) küçük veya eşit ise, bu Parçacıklar yok olmakta ve yerlerine aynı sayıda yeni Parçacıklar eklenmektedir. Bu süreçler sırasında Parçacıkların  $v$  değerleri, pozisyonları güncellenmektedir. Çözüm arama süreci, belirlenen iterasyon sayısı sağlanıncaya kadar devam etmektedir. Kullanıldığı model çalışması aşağıda açıklanmıştır.

### Girdap Optimizasyon Algoritması (GOA) Kullanan Modelleme Çalışmaları;

Bu çalışmada, GOA tabanlı bir Destek Vektör Makineleri (DVM) ile oluşturulan, hibrit bir Yapay Zekâ sistemi kullanılarak diyabet tespiti gerçekleştirilmiştir. Kurulan sistem yapısına göre, DVM'nin eğitimi esnasında, Gauss (RBF) çekirdek fonksiyonunun parametresi olan, sigma ( $\sigma$ ) parametresinin en iyi değerinin bulunması için GOA kullanılmıştır. Bu çalışma için kullanılan veri seti Pima Yerlileri diyabet veri setidir. Süreç ile elde edilen bulgular GOA'nın başarılı bir yöntem olduğunu göstermiştir (Köse vd., 2015).

### 3.2.7. Modele Özgü Geliştirilen Diğer Parametre Ayarlama Algoritmaları

#### 3.2.7.1. Parametre Ayarlama Modülü (Automatic Parameter Tuning Module (APT))

Video konferans Sisteminde Hizmet kalitesi (Quality of Service (QoS)) parametrelerinin ayarlanması bu çalışmada amaçlanmıştır. İki modül mevcuttur: Bilgilerin tutulduğu Resmi Bilgi Modülü (State Knowledge Module (SKM)) ve parametre ayarlama işlemini SKM'den aldığı bilgilere göre yapan Otomatik Parametre Ayarlama Modülü (Automatic Parameter Tuning Module (APT)). Video konferans esnasında bilgiler sürekli değişmektedir ve parametrelerin alacağı değerler bu değişim göz önüne alınarak gerçekleştirilmelidir. Değişen bilgiler SKM'de sürekli güncellenir. Gradient (eğim) çıkış veya iniş yöntemini kullanarak, yeni parametreler elde edilir.  $d(t)$  çalışma koşullarını ve  $y(t)$  geçerli çalışma durumunu göstermektedir.  $y(t)$  QoS parametreleri kümesi için bir fonksiyon olsun (eşitlik 3.2.1). Eğer parametre değerleri artar ya da azalırsa mevcut kaynak azalır.

$$y(t) = f(A(t))$$

3.2.1

Parametre dizisi  $A(t) = [a_1(t), a_2(t), \dots, a_i(t), \dots, a_n(t)]$  ve  $t$  anında  $i$ . parametre  $a_i(t)$  olarak gösterilir. Hata ise  $e(t)$  olarak gösterilir ve  $T$  zamanında geçerli çalışma koşulları ile istenen çalışma koşulları arasındaki farkı vermektedir.

$$e(t) = |d - y(t)| \quad 3.2.2$$

Eşitlikte görüldüğü gibi  $e(t)$  değeri  $A(t)$  ve  $y(t)$  değerine bağlıdır. Hata değeri sıfır ya da sıfıra yakın değer veren parametrelerin ayarlanması gerekir. İlk iterasyon da ayarlanan parametreler  $A(t+1) = [a_1(t+1), a_2(t+1), \dots, a_i(t+1), \dots, a_n(t+1)]$  olarak ifade edilir ve  $a_i(t+1)$  aşağıdaki gibi bulunur:

$$\Delta a_i(t+1) = a_i(t) + \Delta a_i(t) \quad 3.2.3$$

$\Delta a_i(t)$ ,  $a_i(t)$  parametresinin ayarlanma miktarını temsil eder. Elverişlilik için,  $\Delta a_i(t)$ ,  $a_i(t)$  için uygun oranın önceden belirlenen değer setidir.

$$\Delta a_i(t+1) = \rho_i \cdot a_i(t) \quad 3.2.4$$

$\rho_i$  bir ölçekleme faktörüdür. Ölçekleme faktörü boyutu kullanıcılar tarafından parametre tercihi anlamına gelir. Ayrıca 3.2.3 ve 3.2.4 eşitliklerinde görüldüğü gibi, ölçekleme faktörünün işareti parametre uzayında arama yönünü ifade eder. İlk yineleme yapıldıktan sonra mevcut çalışma durumu ve hatalar açıklanır.

$$y(t+1) = f(A(t+1)) \quad 3.2.5$$

ve

$$e(t+1) = |d(t) - y(t+1)| \quad 3.2.6$$

$e(t+1)$  tatmin edici değere ulaşırsa o zaman ayar işlemi sonlandırılır, diğer bir deyişle hata tolere edilinceye kadar uyum süreci devam eder. 3.2.4, 3.2.6 eşitliğinde görüldüğü gibi ölçekleme faktörünün değeri ayarlama işlemi için önemlidir. Eğer ölçekleme faktörü çok büyükse o zaman ayarlama hızı yüksek olabilir, fakat ayarlama işlemlerinde yineleme ilerledikçe parametrenin optimal değeri izole edilebilir. Ölçeklendirme faktörü çok küçük ise o zaman ayarlama hızı düşük olabilir, fakat parametrelerin optimal değer kümesine yakınsaması gerçekleşebilir. Sorunu önlemek için ayarlama miktarı eşitlik 3.2.7 de görüldüğü gibi her bir iterasyonda giderek azalır.

$$\Delta ai(t + k) = \Delta ai(t + k - 1)/2k - 1 \quad 3.2.7$$

Burada k yineleme sayısıdır (Lee et al., 2006).

### 3.2.7.2. Ters Çözüm Tekniği

Ters çözüm tekniği kullanılarak parametre ayarlama işlemi yapılmıştır. Ters çözüm sürecinde manyetik kütlelerin manyetik anomali değerleri, yatay düzlemdeki yapı sınırları, manyetik duyarlılık bilgisi, yer manyetik alanının eğim ve sapma açıları ve model başlangıç parametre değerleri giriş verisi olarak kullanılmaktadır. Uygulamalarda, çeşitli teorik modellerin yeryüzünde oluşacak manyetik anomalilerin değerleri, prizma şeklinde konstrüksiyonlar kullanılarak hesaplanmıştır. Daha sonra, Gauss-Newton ters çözüm tekniğini uygulayarak, teorik anomalileri veren teorik prizma modellerine ait derinlik parametre değerleri hesaplanmıştır. Modele ait parametreler, her yineleme işleminde yeniden hesaplanır ve teorik anomaliler arasında en iyi uyum oluşuncaya kadar yineleme işlemi devam eder. Uygulamalardan elde edilen sonuçlar incelendiğinde ilk tahmin modeline ait alt derinliklerin seçiminin üst derinliklere göre daha duyarlı olduğu görülmüştür. Yani alt derinliklerin seçimi üst derinliklere göre daha az hata payıyla yapılması önerilmiştir. Öte yandan, model içindeki veri alanının genişliği de yöntemin çözüm gücünü etkilemektedir (Öksüm ve Dolmaz, 2006).

## 3.3. Literatürde Yapılan Çalışmaların Değerlendirilmesi

Optimizasyon işlemlerinde kullanılan algoritmalar, yöntemler Çizelge 3.1 de bazı özelliklerine göre bir araya getirilmiştir. Programların ne ölçüde başarılı olduğu gösterilmiştir. Algoritmaların nitelikleri ++ dan -- ye sembolleri ile derecelendirilmiştir. Dereceler kullanılan yöntemler tarafından verilen sınırlara dayanarak oluşturulmuştur. Bu çizelge literatürde kullanılan parametre ayarlama algoritmalarını incelemiş ve tek bir yerde toplamıştır. Yapılan çalışmalarda incelemeler doğrultusunda algoritmaların özellikleri ve nitelikleri ortaya konulmaya çalışılmıştır. Ayrıca bu çalışma gelecekte yapılacak optimizasyon çalışmalarında en doğru algoritmayı bulma yönünde öncü olacağı düşünülmektedir.

Çizelge 3.1 Parametre Ayarlama Algoritmalarının Özellikleri (Korkmaz Tan ve Bora, 2017a)

Metotlar	Algoritmalar	Kalite	Hız	Hedefe İstenen Zamanda Ulaşma	İstikrar	Hedeflenen Değere Ulaşma	Hatanın Az Olması
İlk Parametre Seti Belirleyen Metotlar	Latin-Kare	--	--	--	O	--	--
	Taguchi Dikey Dizileri	--	--	--	O	--	--
	Latin Hiperküp	--	--	--	O		
	Calibra		O		O	+	--
	Design of Experiments (DoE)	-	-	-	-	O	O
Model Tabanlı Metotlar	Coy'un Prosedürü		+		-	++	+
	Sıralı Parametre Optimizasyonu		++		-	++	++
	Beyaz Kutu	--	--		--	O	--
Gözleme Dayalı Metotlar	Yarış (Racing)				--	+	--
Arama Tabanlı Metotlar	Kara Kutu (Black Box)	--	--	--	--	-	--
	Metropolis		-	-	--	+	O
	Adaptive Value Tracker (AVT)		-	-	--	-	O
Etmen Tabanlı Metotlar	Parameter Multi-Agent System (PAMAS)		--	O	-	+	+
Meta-Sezgisel Metotlar	Genetik Algoritma (GA)		++		-	O	O
	REVAC		++		--	+	O
	EA		++		-	--	--
	CMA-ES		++		-	--	--
	Karınca Kolonisi		++		-	-	O
	ADO		++		-	O	-
	Yapay Sinir Ağları		++		-	O	O
	Parçacık sürü Optimizasyonu		++		-	O	O
Biogeography-Based Optimization (BBO)		++				+	O
++ Çok iyi, + iyi, O Kabul edilebilir, - Kötü, -- Çok kötü							

Bu bölümde, Parametre ayarlama işleminde kullanılan algoritmalar ve kullanılan teknikler sunulmuştur. Kullanıldığı alanlardaki farklı eksikliklerin altları çizilmiştir.

Yapılan çalışmalar incelendiğinde parametre ayarlama işleminin birçok farklı alandaki modelleme çalışmalarında gerekli olduğu ve optimizasyon algoritmalarına ihtiyaç duyulduğu görülmüştür. Özellikle gerçek ortamda incelenmesi mümkün

olmayan karmaşık sistemlerin incelenmesi için modelleme ve benzetim ortamlarının modelleme işlemlerinde kullanımı yaygınlaşmıştır. Karmaşık sistemlere ait büyük parametre uzaylarının verimli kullanılabilmesi, istenen sonuçların veya davranışların elde edilebilmesi için uygun parametre setinin bulunması gerekmektedir. Aksi takdirde gerçek ortamda gözlenen davranışların, sonuçların benzetim ortamında gözlenmesi imkânsız hale gelmektedir.

Bu bölümde yapılan çalışmaların incelediği algoritmalar parametre uzayının özellikleri ya da tasarımcıların beklentileri doğrultusunda farklı sınıflandırmaya tabi tutulmuştur. Bu sınıflandırma problem özelliklerine bağlı olarak uygun algoritmanın seçilmesinde kolaylık sağlayacaktır.

Yapılan çalışmalar incelendiğinde modelin ihtiyacına göre 2 durum gözlenmiştir. Birinci durum parametre ayarlarken performansın önem kazandığı ikinci durum en iyi parametre vektörünün bulunmasının önem kazandığı durumlardır. Literatürdeki çalışmalarda, performansın önem kazandığı durumlarda iki aşamalı parametre ayarlama yöntemine başvurulmuştur. İlk olarak büyük parametre uzayından başlangıç parametre seti elde edilir. Bu parametre seti seçime tabi tutularak çok iyi olmasa da kabul edilebilir parametre vektörü elde edilir. Sonuç olarak parametre ayarlama işlemi daha hızlı gerçekleştirilmiş olur. İkinci durumda amaç problemin çözümünde en iyi sonucu veren parametre vektörlerini bulmaktır. Bu durumda performans bir nevi göz ardı edilerek algoritma seçimi yapılabilir. Hem performansın hem de iyi parametre vektörü seçiminin önemli olduğu durum için herhangi bir çalışma yapılmamıştır. Bu tez çalışması her iki duruma çözüm üretmesi bakımından önemlidir.

Yapılan çalışmalar incelendiğinde elde edilen diğer bir sonuç problemin yapısına bağlı olarak, her problem için daha kaliteli bir parametre ayarı kullanılacak yöntemde farklılık gösterecektir. Çünkü her problemin ihtiyacı farklıdır.

Bu çalışmalardan elde edilen en önemli sonuçlardan biri MSA'ların özellikle matematiksel çözümü olmayan karmaşık sistemlerin parametre ayarlama problem çözümünde sıklıkla kullanıldığı görülmektedir. Kullanıldığı uygulamalar için iyi bir parametre seti bulunduğu gözlenmiştir. Özellikle MSA'ların kullanıldığı karmaşık sistem parametre ayarlama problem çözümünde elde edilen sonuçların uygun çözüme yaklaştığı görülmüştür. Bu bölümde elde edilen diğer bulgular, yapılan çalışmalardan hiç birinde global optimum sonucun bulunmadığı fakat beklentiyi karşılayacak sonuçların elde edildiği görülmektedir. Yapılan çalışmalarda

gözlemlenen diğerk bir eksikliğin parametre ayarlama işleminin tek bir problemi çözmek amaçlı yapılmasıdır.

Bu tez çalışmasında, literatürde çok fazla çalışmanın yapılmadığı ve yapılan çalışmaların yetersiz olduğu ETMB ortamında modellenen karmaşık sistemlerin, parametre ayarlama problemine çözüm geliştirilmiştir. Literatürde yapılan çalışmalar(Manuel et al, 2006; Salas et al., 2011;Pereira et al., 2008;Darty et al., 2015; Brax et al., 2013; Calvez and Hutzler, 2005; Calvez and Hutzler, 2007b) tek bir problemi çözmeye odaklı geliştirilmiş olup farklı modellere uyarlanamamaktadır. Bu tez çalışması ile probleme uyarlanabilen yani farklı etmen tabanlı model parametrelerini en uygun yöntem kullanarak ayarlayabilen araç geliştirilmiştir. Ayrıca literatürde, ya performansın ya da en iyi parametre setinin bulunmasının önemli olduğu durum için çözüm yöntemleri geliştirilmiştir. Bu tez kapsamında geliştirilen araç kullanıcının her iki gereksinimine de cevap vermektedir.

Karmaşık sistemler sıklıkla ETMB ortamları kullanılarak modellenirler ve gerçekleştirilen modelin gerçek sistemi yansıtması beklenir. Karmaşık sistemler büyük veri uzaylarına sahiptir. Gerçek sistemi yansıtacak modelin elde edilmesi uygun parametre setinin bulunmasına bağlıdır. En uygun parametre setini bulmak büyük veri uzaylarında bir problem olup bu tez çalışması bu probleme çözüm geliştirmesi hedeflenmiştir. Geliştirilen yöntem yapılan çalışmaların aksine farklı modellerin parametre ayarlama problemine çözüm üretmektedir. Bu tez çalışmasında farklı MSA algoritmaları kullanılmıştır. Parametre ayarlama işlemindeki performansını arttırmak ve farklı problemlerde çözüm üretmesini sağlamak amacı ile algoritmalar uyarlanabilir hale getirilmiştir. 4 MSA algoritması ve 4 uyarlanabilir MSA algoritması olmak üzere toplam 8 farklı algoritmadan oluşan bu aracın başarısı, farklı modeller üzerinde test edilmiştir.

#### 4. ÇALIŞMANIN TEORİSİ VE ARAÇLARI

Karmaşık sistemler incelendiğinde, birbirleriyle etkileşen bileşenlerden oluştukları, bu etkileşimlerin ileri düzeyde bilgi işleme gerektirdiği ve ortamdaki değişikliklere uyum gösterme davranışları içinde oldukları gözlenmektedir. Gerçek ortamda incelenmesi zor ve maliyetli olan bu sistemlerin karmaşık yapısından dolayı incelenmesini kolaylaştırmak amacıyla çoğunlukla etmen tabanlı modelleme ve benzetim tekniklerinden faydalanılmaktadır. Etmen Tabanlı Modelleme ve Benzetim, gerçek sistem performans ölçütlerinin tahmini için analiz ve yeni kurulacak sistemler için bir tasarım aracı olarak kullanılmaktadır. Karmaşık sistem veri uzayını optimize etme yeterliliğine sahip değildir. Bu yüzden modelleme ve benzetim ile birlikte sıklıkla optimizasyon algoritmaları kullanılmaktadır (Korkmaz Tan ve Bora, 2017a).

Karmaşık sistem modellerinin sahip olduğu parametrelerin ayarlanması bir optimizasyon problemidir. Bu problemlerin çözümünde önceleri matematiksel yöntemler kullanılmaktaydı. Matematiksel yöntemin kullanılabilmesi için problemin matematiksel fonksiyonlarla tanımlanması gerekmektedir. Bu işlem karmaşık sistemlerde neredeyse imkânsızdır. Günümüzde doğadaki olaylardan esinlenilerek oluşturulan sezgisel algoritmalar optimizasyon problemlerin çözümünde sıklıkla kullanılmaktadır. Ayrıca, MSA'ların karmaşık sistem modellerine ait olan parametreleri ayarlama için kullanılacak en uygun yöntem olduğu yapılan çalışmalarla desteklenmiştir (Deliktaş, 2005; Calvez and Hutzler, 2005; Başak, 2009; Salwala, 2010; Imbault and Lebart, 2004; Bolme, 2011; Saraçoğlu ve Demirören, 2008; Maniezzo et al., 1994; White, 1998; Calvez and Hutzler, 2007; Doğru, 2005; Banjanovic-Mehmedovic and Karic, 2011; Dobsław, 2010).

Bu bölümün devamında, sırası ile var olan MSA'lardan ve tez kapsamında kullanılan MSA'lardan bahsedilmiştir. Bir etmen tabanlı modelleme ve benzetim aracı olan ve bu tez çalışmasında kullanılan REPAST aracı ve diğer etmen tabanlı araçlar tanıtılmıştır.

## 4.1. Meta-Sezgisel Algoritmalar

### 4.1.1. Sezgisel ve Meta-Sezgisel Kavramları

Sezgisel kavramı İngilizcede “heuristic” kelimesine karşılık gelen bulma ve keşfetme anlamına gelmektedir. Geliştirilen sezgisel optimizasyon yöntemleri tam çözüm tekniğini kullanan diğer optimizasyon yöntemlerinin aksine büyük boyutlu problemlere daha kısa sürece optimum sonuca yakın ve kabul edilebilir çözümler sunabilmektedir. Sezgisel algoritmalarda tam çözüm tekniklerini kullanan diğer yöntemler gibi optimum değerleri bulabilmekte fakat minimum hesaplama maliyeti ile kabul edilebilir çözüm bulması çoğu zaman tercih edilmektedir. Son yıllarda yapılan araştırmalar incelendiğinde özellikle büyük boyutlu karmaşık sistemlerin optimizasyon problemlerinin çözümünde tercih edildiği görülmektedir (Reeves, 1995). Sezgisel algoritmaların en önemli dezavantajı farklı problemlerde beklenen performansları gösterememesidir. Bir problemin çözümünde başarı sağlarken başka problemde başarıyı yakalayamamaktadır. Bundan dolayı farklı problemlere uyum sağlayabilecek yöntemlere ihtiyaç duyulmuştur. Bu ihtiyaçla birlikte Meta-sezgisel yöntemler geliştirilmiştir. Meta-sezgiseller farklı optimizasyon problemlerinde istenen düzeyde başarı sağlaması yönünden önemlidir. Meta-sezgisel kavramının kelime anlamı “meta” İngilizcede “sonra”, “ötesinde” anlamına gelmektedir. Meta-sezgisel kavramı “sezgisel sonrası”, “sezgisel ötesi” gibi anlamlar taşımaktadır çünkü sezgisel algoritmaları geliştirerek bir adım ileri taşımıştır (Glover ve Laguna, 1997).

Meta-Sezgisel kavramı için literatürde bir takım tanımlamalar yapılmıştır bunlardan bir kaçını aşağıda verilmektedir:

“Bir meta-sezgisel, arama uzayını araştırmak yani çözüm çeşitliliği oluşturmak (exploring) ve bulunan bilgiyi işletmek ve ondan faydalanmak (exploiting) için farklı kavramları zeki bir şekilde birleştirerek alt seviye sezgisellere rehberlik eden yinelemeli üretim sürecidir.” (Osman and Laporte, 1996).

“Bir meta-sezgisel, arama uzayında normalde oluşan yerel en iyilerin ötesinde bir çözüm üretmek için diğer sezgisellere rehberlik eden veya diğer sezgiselleri değiştiren ana stratejilerdir.” (Glover and Laguna, 1997).

“Meta-sezgisel, iyi ve kaliteli çözümleri etkin bir şekilde elde etmek için alt seviye sezgisellere rehberlik eden yinelemeli üst seviye bir süreçtir. Alt seviye sezgiseller, basit yerel arama algoritması veya çözüm üretici bir yöntem olabilir.” (Voss et al., 1999).

“Meta-sezgiseller, arama uzayında yüksek kaliteli çözümlere ulaşmak için probleme özgü sezgisellere rehberlik eden yüksek seviye stratejilerdir. Amaç, yerel en iyi çözümlerden kaçınmaktır.” (Stützle, 1999).

“Bir meta-sezgisel, arama uzayının kaliteli çözümlerini içeren bölümlerinde aramayı yoğunlaştırmak için probleme özgü sezgisellere rehberlik etmek amacıyla tasarlanan genel amaçlı sezgisel yöntemdir.” (Dorigo and Stützle, 2004).

Literatürdeki tanımlar incelendiğinde bir meta-sezgisel, probleme özgü optimizasyon problemlerinin performanslarını arttırmak için geliştirilen üst düzey teknikleridir. Geniş veri uzayında yerel optimumlara takılıp kalmasına engel olmak temel özelliğidir. Aramanın tüm veri uzayında yapılabilmesi için daha kötü çözüm değerlerine geçişine izin verir. Başlangıç çözümleri rassal oluşturabileceği gibi zeki teknikleri kullanarak da oluşturabilir. Çoğu meta-sezgisel yöntem kabul edilebilir doğrulukta ya da optimuma yakın sonuçları oldukça hızlı bulabilmektedir (Kılıç, 2008).

Geliştirilen çok sayıda meta-sezgisel algoritma vardır. Her geçen gün yeni meta-sezgisel algoritma geliştirilmekte ve bu alana eklenmektedir. Burada bu algoritmaların hepsinden bahsetmek mümkün olmayacağı için, bu tez kapsamında kullanılan ve en çok bilinen meta sezgisel algoritmalar üzerinde durulacaktır.

#### **4.1.2. Meta-Sezgisel Algoritmalar (MSA)**

MSA, matematiksel çözümü olmayan, büyük boyutlu karmaşık sistemlerin çözümü için geliştirilen optimizasyon yöntemleridir. MSA’lar sadece karmaşık sistem problemlerine değil tüm problemlere uygulanabilme özelliğine sahiptir. Meta-sezgisel algoritmanın her problemde başarılı olmasını sağlayabilmek için bu algoritmalara ait stratejik parametre değerlerinin probleme özel ayarlanmalı yani algoritma probleme uyarlanmalıdır (Hertz ve Widmer, 2003).

MSA’lar yinelemeli fonksiyonları, gelişmiş arama yöntemleri ile büyük boyutlu karmaşık problemlerinin çözümünü yerel optimumlara takılmadan bulabilmektedir. Global optimuma ulaşmak için geliştirilen bu algoritmalar

özellikle gerçek ortamda incelenmesi zor olan karmaşık sistemlerin optimizasyon problemlerin çözümünde kullanılmaktadır (Laguna, 2001). Birçok probleme kolayca uyarlanabilir olması farklı problemlerde sıklıkla kullanılmasını sağlamaktadır.

MSA'ların temel özellikleri aşağıdaki gibi sıralanabilir (Blum ve Roli, 2003);

- a. MSA'lar arama uzayında global optimumu arayan algoritmalarıdır.
- b. MSA'ların amacı aramayı etkin bir şekilde yapıp optimuma yakın sonucu bulmaktır.
- c. MSA'lar basit yerel arama yöntemlerinden, karmaşık öğrenme prosedürlerine kadar çeşitli yöntemler kullanmaktadır.
- d. MSA'lar deterministik olmayıp yaklaşık çözüm bulan yöntemleridir.
- e. Arama uzayında yerel optimuma takılmayı engelleyecek özellikleri barındırırlar.
- f. MSA'lar problemlerden bağımsız olup genellikle tüm optimizasyon problemlerine uygulanabilir özelliktedir.
- g. MSA'lar arama işlemlerinde hafıza tabanlı süreçleri kullanmaktadır.

Herhangi bir MSA'da keşfetme ve yararlanma olmak üzere iki ana bileşen bulunmaktadır. Keşfetme, arama uzayını araştırıp farklı çözümler üreten bileşendir. Yararlanma, bulunan iyi çözüm bölgesindeki bilginin belirtilen problem çözümü için kullanılmasıdır. MSA'lar yararlanma ve keşfetme arasında denge oluşturup optimum çözüme ulaşmaya çalışırlar. Keşfetme bileşeni MSA'ların yerel optimuma takılmalarını önlerken yararlanma sayesinde global optimuma ulaşırlar (Blum and Roli, 2003).

Bilinen ve optimizasyon problemlerinde sıklıkla kullanılan bazı sezgisel algoritmaların bir listesi aşağıda verilmiştir.

**Evrimsel algoritmalar**, doğayı, doğadaki değişim sürecini inceleyen bilimi temel alarak geliştirilen rassal optimizasyon algoritmasıdır. (Bremermann, 1962; Friedberg, 1958; Box, 1957).

**Genetik Algoritma**, Biyolojik evrimden esinlenerek geliştirilen bu algortmada üreme, mutasyon, çaprazlama ve seçim işlemleri kullanılarak optimum çözüm bulmayı hedefler (Holland, 1962; Holland, 1975).

**Benzetimli Tavlama Algoritması**, Tavlama malzemeyi rahatlatmak, yumuşatmak ve içyapıyı daha kullanılabilir hale getirmek için bir katının sıcaklığının maksimum dereceye kadar artırılarak tekrar azaltılması işlemlerinden esinlenilerek geliştirilmiştir (Kirkpatrick et al., 1983).

**Yapay Bağışıklık Sistemleri**, doğal bağışıklık sisteminden esinlenerek, karmaşık sistem problemlerin çözümünde kullanılan optimizasyon yöntemlerden biridir (Farmer et al., 1986).

**Yapay Sinir Ağları**, insan beynindeki süreçlerden esinlenerek geliştirilmiştir, biyolojik sinir ağlarını taklit eden bilgisayar programlarıdır (McCulloch and Rumelhart, 1986).

**Karınca Kolonisi Algoritmaları**, karıncaların yiyecek arama ve birbirleri ile iletişim kurma gibi davranışlarından esinlenerek geliştirilmiştir (Dorigo, 1992).

**Diferansiyel Gelişim Algoritması**, işleyiş ve operatörleri itibariyle evrimsel algortmaya dayanan popülasyon temelli optimizasyon tekniği olarak geliştirilmiştir (Storn and Price, 1995).

**Parçacık Sürü Optimizasyonu**, Parçacık sürü algoritması bir sürü tabanlı sezgisel optimizasyon algoritmasıdır. Sürülerin özellikle kuş sürülerinin hareketlerinden esinlenilerek geliştirilmiştir. Kuşların sürü içerisinde besine yakın olan kuşların konumlarına göre hareket etmeleri algoritmanın temelini oluşturur (Kennedy and Eberhart, 1995).

**Tabu Arama**, bu teknik arama yapılırken elde edilen bilgiler tabu adında bir listede tutuluyor her yeni arama adımında önceki çözümler kontrol edilmekte yinelemeli çözümden kaçınma prensibine dayanmaktadır (Glover and Laguna, 1997).

**Harmoni Arama**, bir orkestradaki müzisyenlerin çaldıkları notalar ile en iyi melodinin elde edilmesi prensibine dayanmaktadır (Geem, 2001).

**Yapay Arı Kolonisi Algoritması**, Arıların nektar toplama davranışlarından esinlenilerek geliştirilen bu algorithmada 3 tip arı bulunmaktadır işçi, kâşif ve gözlemci arı bu arıların davranışları optimizasyon problemlerine çözüm getirmek için geliştirilmiştir (Karaboğa, 2005).

**Ateş Böceği Algoritması**, Sürü tabanlı sezgisel optimizasyon algoritmasıdır. Ateş böceklerinin parlaklığa duyarlı sosyal davranışlarını ele alarak geliştirilmiştir. Algoritma ateş böceklerini cinsiyet olmadan ele alır. Yani tüm ateş böcekleri bir birlerine yönelebilirler. Daha parlak olan ateş böcekleri daha çekicidir. Daha az parlak olan ateş böcekleri çekici olan ateş böceklerine doğru yönelir. Parlaklık etkisi uzaklık arttıkça azalacağı için daha uzaktaki ateş böcekleri uzaktaki parlak ateş böceklerinden daha az etkilenir. Ayrıca bir ateş böceği eğer kendinden daha parlak bir ateş böceği bulamazsa rasgele hareket gerçekleştirir (Yang, 2008).

**Yerçekimi Arama Algoritması**, yerçekimi kanunundan esinlenilerek geliştirilmiştir (Rashedi et al., 2009).

**Guguk Kuşu Algoritması**, guguk kuşu davranışlarından esinlenilerek geliştirilmiştir (Yang and Deb, 2010).

Yukarıda bahsedilen algoritmalar dışında da geliştirilen birçok MSA vardır sadece en çok bilinen algoritmalar kısaca açıklanmıştır. Bu algoritmaların neredeyse hepsinde kendine ait kritik parametre değerleri vardır. Bu parametre değerleri farklı problemlere uyarlanırken değişebilmekte ve algoritmanın performansını doğrudan etkilemektedir. Böylece MSA'ların farklı alanlardaki problemlere uyarlanması kolaylaşmaktadır. Farklı problemlerin çözümünde kullanılan MSA'ların kritik parametre değerleri genelde deneme yanılma yolu ile bulunmaya çalışılır. En iyi değerlerin bulunması için algoritmanın çok sayıda çalıştırılması gerekmektedir. Bu da uğraştırıcı ve zaman alıcı bir işlemdir. En iyi performansı yakalayacak parametrelerin bulunması neredeyse imkânsızdır. Bundan dolayı kritik parametrelerin probleme uygun değerleri otomatik bulabilecek bir sistemin geliştirilebilmesi gerekmektedir ve bunun üzerine çalışmalar yapılmaktadır (Eiben et al., 1999). Bu tez kapsamında meta-sezgisel algoritmalara ait parametrelerin ayarlanması için uyarlanabilir parametre ayarlama yöntemi kullanılmıştır.

### 4.1.3. Meta-Sezgisel Algoritmaların Kritik Parametre Değerlerinin Ayarlanması

Sezgisel algoritmaların problem bağımlılığını ortadan kaldırmak amacı ile geliştirilen MSA'lar farklı problemlere çözüm sunmaktadır. Fakat sahip oldukları kritik parametre değerleri performansı etkilemektedir. MSA'ların optimizasyon işlemindeki performanslarını arttırmak ve optimuma yakın çözümlere ulaşabilmek için kritik parametre değerlerinin otomatik olarak probleme uyarlanması gerekmektedir.

Bir MSA'nın birden çok kritik parametre değerleri olabilmektedir. En iyi çözümü sunacak değerlerin manuel ayarlanması neredeyse imkânsızdır. Bir problemde başarılı sonuçlar üretebilen değerlere sahip algoritmalar aynı kritik parametre değerleri ile farklı problemlerde aynı başarıyı maalesef yakalayamamaktadır. Bu tez çalışmasında yapılan uygulamalarla bu kanıtlanmıştır. Kritik parametrelerin ayarlanması ihtiyacından doğan gereksinimle literatürde yapılan çalışmalar incelendiğinde çevrimdışı ve çevrimiçi iki farklı parametre ayarlama yöntemi bulunduğu görülmektedir (Eiben et al., 1999).

#### 4.1.3.1. Çevrimdışı Ayarlama

Parametre değerlerinin ayarlanması algoritmanın çalıştırılmadan önce belirlenmesi ve bu değerlerle algoritmanın çalışması esasına dayanır. Bu yöntemden biri deneme yanılma yöntemidir. Bu yöntemde, özellikle çok sayıda parametre değeri olan algoritmalar için tüm parametre kombinasyonlarını denemek neredeyse imkânsızdır. Diğer bir yöntem bir optimizasyon algoritması kullanılarak en iyi parametre değerlerinin bulunması ve bulunan sabit parametre değerleri ile algoritmanın çalıştırılması ilkesine dayanmaktadır. Bu parametrelerin ayarlanması için kullanılacak algoritmanın başarısı iyi parametre değerlerini bulmada önemlidir. Literatürde yapılan çalışmalarda kullanılan bir takım algoritmalar mevcuttur. Bunlar;

CALIBRA (Adenso-Diaz and Laguna, 2006), SPO (Bartz-Beielstein et al., 2004; Lasarczyk, 2007) ve REVAC (Nannen and Eiben, 2006) algoritmalarını kullanan çalışmalar çevrim dışı sadece sayısal değer alan parametre değerlerini bulmuşlardır. Ayrıca F-Race ve yinelemeli F-Race (Birattari et al., 2010) ve genetik programlama (Oltean, 2005; Fukunaga, 2008) algoritmalarını kullanan çalışmalar sayısal değere sahip parametreler dışında değerler alan parametrelerin de çevrim

dışı ayarlaması amaçlanmıştır. Tüm bu çalışmalarda iyi parametre değerleri algoritma çalıştırılmadan önce bulunur ve algoritma parametrelerine sabit olarak atanır ve çalıştırma süresince bu değerler değişmez. Bu durum özellikle karmaşık sistemlerin çalışma esnasında sürekli değişen durumlarına uyum sağlayamaması, Her problem çözümünde parametre değerlerinin yeniden ayarlanması gerekliliği, maliyetin yüksek olması gibi sebeplerden dolayı çevrimiçi parametre ayarlama yöntemi geliştirilmiştir.

#### **4.1.3.2. Çevrimiçi Ayarlama**

Yapılan çalışmalar incelendiğinde özellikle karmaşık sistemlerin dinamik yapısından dolayı sabit parametre değerlerine sahip MSA'ların yetersiz olduğu görülmüştür. Uygulandığı problemde aldığı geri bildirimlerle uygun parametre değerleri kullanan MSA'ların optimizasyon işlemlerinde daha başarılı olduğu kanıtlanmıştır. Bundan dolayı çevrimiçi parametre ayarlama yöntemi geliştirilmiştir (Back, 1992; Hesser and Manner, 1991; Jain and Fogel, 2000). Çevrimiçi yani algoritmanın çalışma esnasında uygun kritik parametre değerlerin bulunması algoritmanın performansını arttırdığı yapılan çalışmalarla kanıtlanmıştır.

Çevrimiçi kritik parametre ayarlama işlemi için farklı teknikler kullanılmıştır (Eiben et al., 1999). Bu tekniklerden biri deterministik parametre ayarlama yöntemi olup belirlenen kurallara ve zaman dilimlerine bağlı olarak parametre değerlerinin güncellenmesidir. İkinci yöntem kendinden uyarlanabilen parametre ayarlama yöntemi olup bu yöntem algoritmanın çalışması esnasında arama uzayına dâhil edilerek en uygun değerlerin bulunması sağlanır yani MSA'ların kritik parametre değerleri yine MSA'lar kullanılarak çevrimiçi parametre değerlerinin ayarlanması sağlanır. Son yöntem ise uyarlanabilir parametre ayarlama yöntemi olup algoritmanın çalışma esnasında yani arama sürecinde çözülmeye çalışılan problemde elde edilen geri bildirimlere göre parametre değerlerinin güncellenmesi sağlanır.

#### **4.1.3.3. Uyarlanabilir Parametre Ayarlama Yönteminde İzlenen Adımlar**

Bu tez kapsamında, MSA'ların kritik parametre değerlerini ayarlamak için uyarlanabilir parametre ayarlama yöntemi kullanılmıştır. Bu yöntemde önemli olan algoritma performansına doğrudan etkisi olan kritik parametrelerin ve bu parametrelerin alabilecekleri değer aralıkları belirlenmesidir. Stratejik

parametrelerin belirlenmesi, her bir MSA bünyesinde birbirinden farklı ve birçok kritik parametre barındırmaktadır. Bu parametreleri ayarlamadaki temel amaç algoritma performansını arttırmaktır. Bu yüzden ilk olarak algoritma performansını doğrudan etkileyen parametreleri belirlemek gerekmektedir. Uyarlanabilir parametre ayarlamasının temel işlevi aşağıda verilmiştir.

- *Kritik parametrelerin belirlenmesi*, algoritmanın farklı parametre değerleri ile çalıştırılması ve elde edilen sonuçların incelenmesi gerekmektedir. Kritik parametre belirlenirken, seçilen parametre dışındaki parametreler sabit tutularak değerleri değiştirilir eğer çalıştırma sonucu elde edilen sonuçlar değişiyorsa bu parametre kritik parametre olarak seçilir. İkinci önemli adım belirlenen bu parametre değişkenlerinin değer aralıklarının belirlenmesidir.
- *Parametre değişkenlerinin değer aralıklarının belirlenmesi*, Algoritma çalıştırılırken belirlenen kritik parametrelerin değerleri güncellenmektedir. İlgili MSA'yı geliştiren yazılımcı her bir parametrenin alacağı değer aralıklarını belirlemiş olmasına rağmen bu değerler bazı problemlerde başarısız sonuçlar üretmektedir. Bu nedenle değer aralıkları probleme bağlı olarak belirlenmelidir. Parametre değer aralıklarının belirlenmesinde uyarlanabilir parametre ayarlama algoritmalarının önemli rolleri vardır. Elde edilen geri bildirimlerle değer aralıklarının değiştirilmesi daraltılması ya da genişletilmesi mümkündür. Değer aralıkları belirlendikten sonra parametre ayarlama işlemi mümkündür.
- *Uyarlanabilir Parametre Ayarlama*, kritik parametre değer aralıkları belirlenirken geçmişte kullanılan başarılı değerler kullanılır. İlk olarak her bir kritik parametre değerleri rasgele üretilerek bir diziye atılır. Algoritma sırayla bu dizideki değerlerle çalıştırılır. Elde edilen başarılı değerler başarılar dizisine atılır. Başarılar dizisinde boş kalan alanlara yine rasgele değerler atanır böylece algoritmanın yerel en iyilere takılması önlenmiş olur. Eğer rasgele üretilen ilk dizideki değerlerden hiç biri başarı gösterilmemiş ve başarılar dizisine eleman eklenmemiş ise algoritma duraksamış kabul edilir dizi için belirlenen değer aralığı değiştirilir ve üretilen yeni değerler ile başlangıç dizisi oluşturulur. Bu yöntemle algoritmanın en iyi parametre değerleri uyarlanabilir parametre ayarlama yöntemi ile bulunmuş olur.

Bu tez çalışmasında kullanılan MSA'ların kritik parametrelerini ayarlamak için yukarıda madde başlıkları ile açıklanan çevrimiçi uyarlanabilir parametre ayarlama yöntemi geliştirilmiş ve kullanılmıştır. Amaç bu yöntemi kullanarak kullanılan MSA'nın probleme uyarlanabilir hale getirip problemin çözümündeki performansını arttırmaktır.

## 4.2. Etmen Tabanlı Benzetim ve Modelleme Araçları

Karmaşık Sistemlerin modellenmesinde etmenler oldukça güçlü çözümler sunarlar. Karmaşık sistemlerin temel yapısı tüm sistemin global davranışlarının basit bileşenlerin etkileşimleri ve davranışları sonucu karmaşık davranışların ortaya çıkması esasına dayanır. Etmenler bileşenleri temsil eden otonom zeki yapılardır. Bundan dolayı bileşenler arası etkileşimi incelemek, dinamiklerini belirlemek ve davranışların analizi için ETMB'ler kullanılır.

Karmaşık sistem modellerinin ana yapı taşı genellikle bulunduğu ortama uyum sağlayabilen otonom zeki etmenler oluşturur. Etmenin ve etmenler arasındaki etkileşim, bilgi akışını, kontrol edilecek ve eyleme dönüştürülecek davranışların yapı taşı mantığı ile oluşturulduğu mekanizmalara genel olarak etmen mimarileri denir (Maes, 1991; Kaelbling, 1991).

Bir çoklu etmen yapısını tasarlanması için bilgi teknolojisi hâkimiyetinin iyi olması ve tasarlanacak problem ortamının çok iyi bir şekilde tanımlanabilmesi gerekmektedir. Bundan dolayı tasarım sürecini kısaltmak amacı ile hazır benzetim araçları geliştirilmiştir. Aşağıda verilen benzetim araçları etmen tabanlı modellemede en çok kullanılan araçlardır.

### 4.2.1. SWARM

SWARM geliştirilen ilk reaktif etmen tabanlı modelleme yazılım kütüphanelerinden biridir. SWARM Nesne Tabanlı bir Yazılım dili olarak tasarlanmıştır. Tasarımda grup davranışları önemlidir. Oluşturulan her nesneye ait isim, kural ve veri vardır. İsim nesneyi nitelendirir, veri nesneye ait tüm bilgileri ve kurallar nesnelere arası iletişimi gerçekleştiren mesajlar tutulur. SWARM'ın en basit bileşeni belli davranışlara sahip etmenler topluluğu oluşturmaktadır. Bu topluluklar model etmen ve izleyici etmen topluluklarıdır. Model etmen topluluğu, gerçek sistemin karakteristiklerini modellemek için izleyici etmen topluluğu ortamı ve model topluluğu için girdi veri setini oluşturmaktadır. Bununla birlikte veri analizi

ve arayüz görevi yapar. Toplulukların modülerliği ve birleştirilebilirliği SWARM aracına büyük bir esneklik sağlamaktadır (Minar, 1996).

#### **4.2.2. MASON**

MASON, geniş kapsamlı özel amaçlı Java benzetimler için temel teşkil edecek şekilde tasarlanan ve birçok hafif benzetim ihtiyacı için yeterli sayıda işlevsellik sağlamak üzere tasarlanmış, Java için hızlı, ayrık olaylı birçok temsilcili benzetim kütüphanesi çekirdeğidir (Luke et al., 2004).

#### **4.2.3. NETLOGO**

NetLogo, karmaşık doğal ve sosyal fenomeni simüle etmek için çok ajanlı bir programlama dili ve modelleme ortamıdır. Zamanla gelişen karmaşık sistemlerin modellenmesi için özellikle uygundur. Modelleyiciler, bireylerin mikro düzeydeki davranışları ile etkileşimlerinden çıkan makro düzeyde kalıplar arasındaki bağlantıları keşfetmek için, aynı anda yüzlerce veya binlerce bağımsız "aracı" talimat verebilir. NetLogo, kullanıcıların benzetimlerin açmalarını ve çeşitli koşullar altında davranışlarını keşfederek onlarla "oynayabilmelerini" sağlar. NetLogo, öğrencilerin ve araştırmacıların profesyonel programlar olmasa bile kendi modellerini oluşturmalarını sağlayacak kadar basit bir yazarlık ortamıdır (Tisue, 2004).

#### **4.2.4. REPAST**

Etmen benzetimi için yazılımın oluşturulması, görselleştirilmesi ve veri toplanmasını sağlayan ve üç yazılım dilinin kullanıldığı bir araçtır. Kullanılan yazılım dilleri; JAVA, Python ve Microsoft.NET'dir. Ayrıca SWARM'ın özelliklerini ve bunun dışında birçok özelliği içinde barındırır. RePast bilgiyi çizelgeler, histogramlar ve benzetimi video formatında görselleştirme imkânı sunmaktadır. RePast'ın diğer özellikleri: RePast kesikli olay benzetimi olup sıralı hem de paralel kesikli olay operasyonlarını gerçekleştirir. RePast'daki stokastik süreçler Monte Carlo benzetimine dayanılarak modellenmektedir. Kullanıcı etmen özelliklerini, etmen davranış eşitliklerini ve model özelliklerini benzetim esnasında değiştirebilmektedir. Repast, coğrafi ve ağ işlevleri gibi birçok sınıf içermektedir (Railsback et al., 2006). MASON, bir Java platformu olarak geliştirilmekte olup Repast MASON'a göre daha olgundur ve daha yaygın kullanılan bir benzetim aracıdır. Sosyal bir bilim araştırması olarak tasarlandığından sosyal ağlar ve

bunların bazı ağ değerlerinin ölçülmesine olanak sağlar (Collier et al., 2003; North et al., 2006)

#### **4.2.5. JACK**

JACK, Agent Oriented Software tarafından geliştirilen bir yazılımdır. JACK, etmenler, etmenlerin yetenekleri, inanç kümeleri, eylemleri ve planları gibi bileşenlerden oluşmaktadır. BDI mimarisini kullanan JACK’de Etmenlerin inançları, inançlar kümesinde tutulur ve benzetim esnasında değişebilir. Etmenlere ait hedefler ve bu hedeflere ulaşmak için birden çok planları vardır. Etmenler hedefe ulaşmak için farklı planlar yapar (Kota, 2002; Agent Oriented Software, 2005)

#### **4.2.6. DIAS**

Dinamik Bilgi Mimarisi Sistemi (DIAS), çok disiplinli, dinamik benzetimler geliştirmek için esnek, genişletilebilir, nesne tabanlı bir araçtır. DIAS etki alanı-nötrdür; başka bir deyişle herhangi bir disipline veya konu alanına özgü benzetimler için tasarlanmamıştır; aksine neredeyse her tür uygulama benzetimin geliştirilmesini desteklemektedir. DIAS “NE?” sorusunu “NASIL” sorusundan ayırt etmektedir. Bunun nedeni nesne sınıfının nesnenin davranışı hakkında sadece öz bilgiler içermesi ve uygulama ile ilgili detaylara sahip olmamasıdır. Model ile uygulama ise “NASIL?” sorusuna karşılık gelen nesnelerin tüm davranışlarını uygulayarak bunları uygun nesnelere ile bağlantılı olarak gerçekleştirmektedir (Campbell and Hummel, 2007).

### **4.3. Sonuç ve Tartışma**

Bu tez çalışmasında amaç karmaşık sistemlerin modellendiği ETMB ortamında geliştirilen farklı problemlerin parametre ayarlama işlemine en iyi çözümü bulacak yaklaşımı geliştirmektir. Karmaşık sistemlerin optimizasyon problemlerine çözüm sunacak en uygun tekniğin MSA’ların olduğu literatürde yer alan araştırmalar ile kanıtlanmıştır. Fakat MSA’ların sahip olduğu kritik parametre değerlerinin de ayarlanması gerekliliği farklı problemlere çözüm sunmayı hedefleyen bu çalışmada önemlidir. Mata-sezgisel algoritmaların sahip olduğu parametre değerleri kritik parametreler olarak adlandırılmış olup çevrimiçi ayarlanabilir parametre ayarlama yöntemi kullanılarak ayarlama işlemi gerçekleştirilmiştir. Böylece bu tez çalışmasında iki katmanlı parametre ayarlama problemlerine çözüm getirilmiştir. Karmaşık sistem model parametrelerini ayarlama

işleminde MSA'lar kullanılmış ve kullanılan MSA'ların kritik parametre değerlerini ayarlamak için çevrimiçi uyarlanabilir parametre ayarlama yöntemi kullanılmış olup parametre ayarlama iki boyutlu hibrit bir yaklaşım geliştirilmiştir.

Bu bölümde ayrıca ETMB araçlarına yer verilmiştir. Bu tez çalışmasında kullanılan yazılım aracı Repast olarak seçilmiştir. Repast özellikle karmaşık sistemleri modellemede yaygın kullanılan bir araç olduğundan, var olan modellerin parametrelerini ayarlamak için tercih edilmiştir. Bununla beraber, bütün etmen-tabanlı benzetim araçlarında olduğu gibi, Repast da modelleri için parametre ayarlama araçları açısından eksiktir. Böyle bir destek, özellikle güvenilir gerçekçi modeller inşa edilmek isteniyorsa çok önemlidir. Sonraki alt bölümlerde bu gereksinimin üstesinden nasıl gelindiği üzerinde durulmuştur.

## 5. KULLANILAN META-SEZGİSEL ALGORİTMALAR

Günümüzde doğadaki olaylardan esinlenilerek oluşturulan sezgisel algoritmalar optimizasyon problemlerinde sıklıkla kullanılmaktadır. Ayrıca MSA'ların bu sistemlere ait olan parametreleri ayarlama da kullanılabilir en uygun yöntem olduğu yapılan çalışmalarla desteklenmiştir. Optimizasyon alanında kullanılan en başarılı algoritmalar bu çalışmada kullanılmıştır. Genetik algoritma (GA), Parçacık sürü optimizasyonu (Particle Swarm Optimization-PSO), yapay arı kolonisi (Artificial Bee Colony-ABC) ve ateş böceği (Firefly Algorithm-FA) algoritmalarıdır.

GA, bir sezgisel arama algoritmasıdır. Doğal seçim ve genetik konularından esinlenerek mutasyon ve çaprazlama yöntemlerini de kullanarak verilen probleme çözüm sunmayı hedefler (Holland, 1975). geniş arama uzaylarını klasik yöntemlere göre çok daha hızlı taraması ve özellikle çözüm uzayının geniş, süreksiz ve karmaşık olması durumlarında başarılı sonuçların elde edilmesi nedeni ile tercih edilmiştir.

PSO, Kuş veya balık sürülerinin davranışlarından esinlenilerek geliştirilmiştir (Kennedy and Eberhart, 1995). PSO, GA'ların evrimsel hesaplama teknikleri ile birçok benzerlikler gösterdiği gibi GA'nın aksine, PSO'da çaprazlama ve mutasyon gibi parametreler bulunmamaktadır. PSO'nun avantajı gerçekleştirilmesinin kolay olmasıdır ve ayarlanması gereken çok az parametresi olması bu tez kapsamında kullanılma nedenidir.

ABC, ayarlanmanın kolay olması, az sayıda parametreye sahip olması, birçok problemin çözümünde yaygın kullanılıyor olması, farklı problemler için geliştirilen birçok versiyonunun olması nedeni ile bu tez çalışmasında kullanılan diğer algoritmadır (Karaboğa, 2005).

FA, gerçek ateş böceklerinin biyokimyasal ve sosyal yönlerinden esinlenilerek üretilmiş sürü zekâsı tabanlı bir eniyileme algoritmasıdır. Ayarlanması gereken parametre sayısının az olması, anlaşılması ve ayarlanmasının kolay olması nedeni ile bu tez çalışmasında kullanılması tercih edilen diğer bir algoritmadır. (Yang, 2008).

Bu bölümde kullanılan ve birer MSA olan GA, PSO, ABC, FA algoritmaları açıklanacaktır.

## 5.1. Kullanılan Meta-Sezgisel Algoritmaların Ortak Kavramları

Kullanılan GA, PSO, ABC ve FA algoritmalarına ait kullanılacak ortak kavramlar aşağıda verilmiştir.

**Problem:** Algoritmada çözüm aradığımız problemdir. Bu tez için ETMB’lerde modellenen karmaşık sistemlerin parametre değerlerinin ayarlanması problem olarak tanımlanmıştır. Bu problem çözümü için model hedefleri doğrultusunda belirlenecek uygunluk fonksiyonları kullanılmaktadır.

**Uygunluk Değeri:** Çözümün kalitesini belirlemede etkindir. Bu değeri hesaplamak için probleme özgü fonksiyon kullanılır. Elde edilen her bir uygunluk değeri sahip olduğu çözüm kümesinin yani bireyin çözüm gücünü gösterir.

**Popülasyon Boyutu:** Bir popülasyonda yer alan kromozom sayısı yani birey sayısı(çözüm kümesi). Her bir kromozom mevcut problem için çözüm kümesidir.

**Parametre Boyutu:** Problemden ya da kullanılan fonksiyonda yer alan bilinmeyenli değişkenler ya da modelde ayarlanması gereken parametre değerleri olarak tanımlayabiliriz.

**Parametre Aralığı:** Parametrelerin alabileceği maksimum ve minimum değerlerdir. Bu değerleri en iyi model tasarımcısı belirleyeceği gibi. Optimizasyon algoritmaları veya bir takım fonksiyonlar kullanılarak belirlenebilir. Probleme göre değişiklik gösterir.

**Bitirme Kriteri:** Algoritmanın problem çözümü için çalışacağı maksimum döngüdür. Örneğin; iterasyon değerinin artması çözüme yaklaşmayı sağlayabileceği gibi, zaman ve maliyeti arttıracaktır. En uygun çözüm için en uygun iterasyon sayısı belirlenmeli.

## 5.2. Genetik Algoritma (GA)

1950 ve 1960’larda başlayan evrimsel çözümler üzerinde çalışılmaya başlanmış 1960 yıllarında John Holland optimizasyon işlemleri için doğal genetik yöntemlerin kullanıldığı olası çözüm kümelerinden bir popülasyon oluşturulduğu algoritmayı geliştirmiştir. Amaç algoritma geliştirmenin ötesinde doğal ortamdaki

adaptasyon olgusunu bilgisayar ortamına taşımaktı. Böylece genetik yapının kullanıldığı operatörlerin işlendiği bir yöntem ile optimizasyon problemlerine çözüm bulunacaktır (Mitchell, 1999).

Holland'ın GA canlıların genetik şifre yapısı kullanılarak sezgisel olarak en iyi çözümü veya en iyi çözüme yakın bir sonuç bulmayı hedefler. Geniş arama uzaylarının klasik yöntemler kullanılarak taranması uzun zaman almakta ve maliyeti arttırmaktadır. Genetik algoritma beklentiyi karşılayacak yeterlilikte çözümü kısa sürede bulabilmektedir. Özellikle karmaşık sistemlerin sahip olduğu çok boyutlu büyük veri uzayları için en iyileme işlemlerinde GA başarılı olmaktadır. GA'nın diğer avantajı arama işlemini tek bir noktadan değil farklı noktalarda geniş bir alanda gerçekleştirmesidir. Olasılık kurallarını kullanır ve türev bilgisine ihtiyaç duymaz. Karmaşık, geniş, süreksiz ve çok boyutlu veri uzayında en iyinin hayatta kalması ilkesine göre yerel iyilere takılmadan en iyi çözümü arar (Korkmaz Tan ve Bora, 2017c).

GA parametrelerinin her biri biyolojide bir gen olarak temsil edilirken. Parametre kümesi ise kromozomları temsil eder. GA'ların her bir kromozomu bir çözüm kümesidir. Bu çözüm kümeleri bir araya gelerek popülasyonları oluştururlar. Kromozomların sahip olduğu uygunluk değeri probleme bağlı olarak ve belli kurallar çerçevesinde maksimize ya da minimize edilir. Yeni nesiller ilk popülasyonda en iyi uygunluğa sahip çözüm kümelerinin gen değişimi ile elde edilir.

Genetik algoritmalarda seçilme, Çaprazlama ve Mutasyon operatörleri yer almaktadır. Seçilme için, yeni nesillerin oluşması için en iyi uyum değerine sahip bireylerin seçilmesi işlemidir. Çaprazlama için popülasyondan iki adet birey seçilir. Bu bireylerden uygun kurallar çerçevesi ile karşılıklı belirli sayıda gen değişimi yapılır ve yeni birey oluşturulur. Bu işlem sonucunda iki adet yeni birey yani çocuk elde edilmiş olur. Mutasyon operatörü kullanılarak bireylerin genleri değiştirilir. Bu değişim popülasyonun genel olarak %1-%5 ini kapsamaktadır. Mutasyon popülasyonda ki çeşitliliği arttırarak arama işleminde yerel en iyilere takılmayı önler.

### **5.2.1. Genetik Algoritma Kavramları ve Operatörleri**

Genetik algoritma ile ilgili çalışmada kullanılacak temel kavramlar aşağıda verilmiştir;

**Gen:** Ebeveynlerden çocuklarına aktarılan bireyin en küçük kalıtsal biyolojik birimi olarak adlandırılır. Bu tez çalışmasında her bir gen bir parametreyi temsil etmektedir. Bu çalışmada GA'daki gen terimi yerine parametre kavramı da kullanılmaktadır.

**Alel Gen:** Genlerde aynı karakteristik özelliği kodlayan fakat farklı kodlar taşıdığı için farklı özelliklerin ortaya çıkmasını sağlayan genlerden her biri alel genidir.

**Genotip:** Bir canlının sahip olduğu genlerin toplamıdır.

**Fenotip:** Genotip özelliklerin bireyin fiziksel görünüşündeki yansımasıdır.

**Genom:** Organizmanın genetik yapısı yani kalıtım materyalinde bulunan genetik kodların tamamını temsil eder.

**Kromozom:** Kromozomlar anne ve babadan alınan genlerin oluşturduğu kalıtım yapılarıdır. Kromozomlarda anne babaya ait fiziki kimyasal özellikler bulunmaktadır. Döllenme sonucu oluşan yeni bireylerin özellikleri anne babadan alınan özellikler doğrultusunda farklılaşmaktadır. Genlerden oluşan her bir kromozom birey olarak da ifade edilmektedir. Bu çalışmada kullanılan parametre ayarlama problemi için kromozom ya da birey ifadeleri yerine çözüm kümesi ifadesi de kullanılmaktadır. Kromozom, birey ve çözüm kümesi aynı anlamda kullanılmaktadır.

#### **5.2.1.1.Çözümlerin Kodlanması**

Probleme ait bilginin genetik algoritmanın kullanma biçimidir. Parametrelerin kodlanması, probleme ait parametrelerin genetik algoritmanın kullanacağı şekle çevrilmesine olanak tanır (Goldberg, 1989). 4 farklı kodlama tekniği mevcuttur;

- **İkili Kodlama:** Yapılan çalışmalarda en çok kullanılan tekniktir. Çözümlerin hesaplanmasında 0-1 değerleri kullanılır.
- **Permütasyon Kodlama:** Sıralama problemlerinde kullanılır. Dizinin uygunluk değeri genlerin değerlerine ve sıralamasına

bağlıdır. Gezin satıcı gibi problemlerin optimizasyon işlemlerinde kullanımı uygundur.

- **Değer Kodlaması:** Karmaşık sayıların yer aldığı problemlerin optimizasyonu işlemlerinde kullanılır. Değerler probleme bağlı olarak herhangi bir sayıdan ya da karakterlerden oluşabilir. Bu tez çalışmasında bu kodlama tekniği kullanılmıştır.
- **Ağaç Kodlama:** Değişen gelişen programlar, fonksiyonlar, nesnelere ya da komutlar bu ağaç yapısını oluşturmaktadır. Programa ait istenen bir fonksiyonu oluşturmak için bu kodlama tekniği kullanılabilir.

### ***5.2.1.2. Başlangıç Popülasyonu***

Olası çözümler kodlandıktan sonra her bir çözüm kümesi yani kromozom oluşturulur. Oluşturulan çözüm kümelerinin tamamı popülasyonu oluşturur. İlk popülasyonun oluşturulması için rassal değerler üretilebilir. Rassal sayı üretilirken kullanılan kodlama yöntemine göre üretilen değer belirli aralıklar sınırında her bir gen üretilir ve çözüm kümesi elde edilir (Yeo ve Agyel, 1996).

### ***5.2.1.3. Uygunluk Değerinin Hesaplanması***

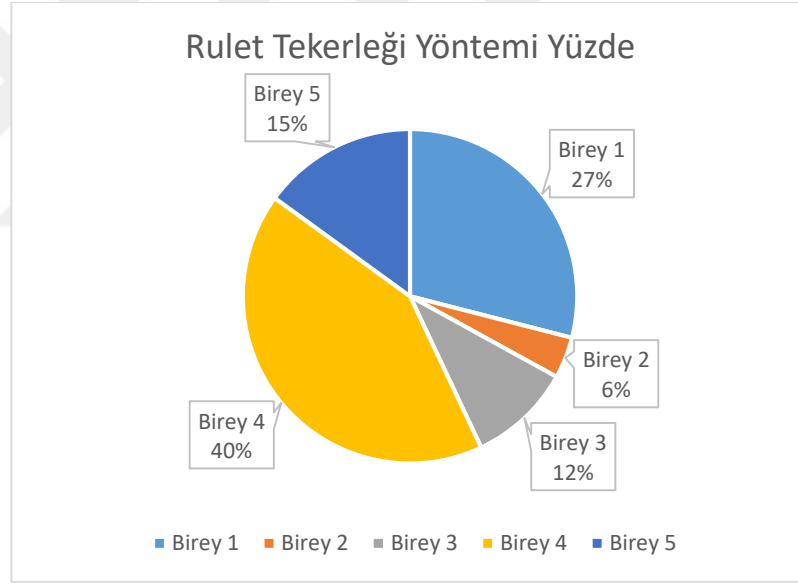
İlk Popülasyon oluşturulduktan sonra her çözüm kümesine ait uygunluk değeri hesaplanır. Uygunluk değeri genelde kullanılan probleme bağlı oluşturulan matematiksel fonksiyonunun değeridir (Jang, 1997). Kullanılan problemde minimizasyon işlemi yapıyorsa en küçük uygunluk değeri eğer maksimizasyon işlemi yapıyorsa en büyük uygunluk değerine sahip çözüm kümesi en iyi çözüm kümesi olarak belirlenir. En iyi uygunluk değerine sahip çözüm kümelerinin yaşama ve üreme şansı o kadar fazladır. Böylece sonraki kuşaklara aktarılan iyi çözümler sayesinde optimum çözüme ulaşılmaya çalışılır (Yeniay, 2001).

### ***5.2.1.4. Elit Bireylerin Seçilmesi-Seçim Operatörü***

İlk olarak uygunluk değeri iyi olan çözüm kümeleri seçilir. Böylece yeni nesil oluşurken önceki popülasyon da yer alan iyi çözümlerin kullanılması uygunluk fonksiyonuna göre seçilir ve iyi bireylerin gelecek nesile aktarılması sağlanmış olur. Seçilme kısaca, seçilen çözüm kümelerini çiftler halinde gruplayarak çaprazlama teknikleri ile yeni çözüm kümelerinin

yani bireylerin oluşmasını sağlamaktadır (Fıçlalı, 2002). Seçim yöntemleri uygunluk değeri yüksek çözüm kümelerinin daha yüksek olasılıkla seçilmesini sağlar böylece bu çözümlerin bir sonraki nesile aktarılma olasılığı arttırır. Seçme işleminde kullanılan farklı yöntemler mevcuttur. En başarılı ve en yaygın olarak kullanılan rulet tekerleği ve turnuva yöntemi bu tez çalışmasında kullanılmıştır.

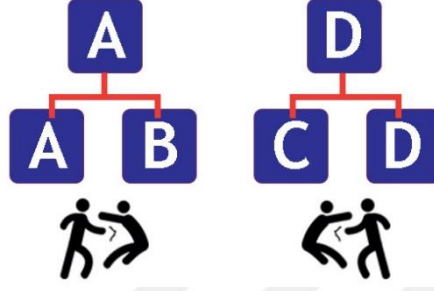
**Rulet Tekereği Yöntemi:** Bu yöntemde her çözüm kümesi uygunluk değeri ile orantılı bir olasılık değeri almaktadır. Rulet tekerleğinin yüzeyi dizilerin uygunluk değerleri ile orantılı olarak atanır. Tekerlek her döndürüldüğünde bir çözüm kümesi eşleme dizisine kopyalanır. Daha iyi uygunluk değerine sahip çözüm kümeleri tekerlekte daha fazla yer kapladıklarından seçilme şansları daha yüksektir. Rulet tekerleği yöntemine ait bir örnek Şekil 5.1’de verilmiştir.



Şekil 5.1 Rulet Tekereği Yöntemi

**Turnuva Yöntemi:** Tüm popülasyondan her adımda rasgele bir çözüm kümesi seçilir. Seçilen çözüm kümeleri turnuvaya dâhil edilir yapılan turnuva iki çözüm kümesi arasında yapılır hangisi daha iyi uygunluk değerine sahipse kazanır. Her çözüm kümesi turnuvaya iki kez katılır ve rakip rasgele seçilir. Böylece her iki turnuvada da başarılı olan çözüm kümesi en iyi çözüm kümesi olarak seçilir. En kötü çözüm kümeleri de iki turnuvada da başarısız olan çözüm kümeleridir. Tek turnuvayı kazanan çözüm kümeleri orta seviyeli çözüm kümeleri olarak belirlenir. En iyi seçilen çözüm kümeleri listeye

isimlerini iki kez orta seviyeli olarak seçilen çözüm kümeleri listeye isimlerini bir kez yazdırır. Şekil 5.2 incelendiğinde popülasyondan seçilen A ve D çözüm kümeleri seçilir daha sonra rasgele B ve C çözüm kümeleri seçilir. Turnuva A ve B ile C ve D arasında gerçekleştirilmektedir. En iyi uygunluğa sahip çözüm kümeleri seçilir. Her çözüm kümesi farklı çözüm kümeleri ile toplamda 2 kez turnuvaya girer.



Şekil 5.2 Turnuva Yöntemi

#### 5.2.1.5.Çaprazlama Operatörleri

Çaprazlama operatörü, en iyi uygunluğa sahip çözüm kümelerinden daha iyi çözümler üretilmesi amacıyla kullanılan operatördür. Çaprazlama eşleme havuzuna atılan çözüm kümelerine uygulanmaktadır (Jang, 1997: 176). Bu operatörün genetik algoritmanın performansına katkısı büyüktür. Bundan dolayı çeşitli çaprazlama teknikleri üretilmiştir. Tek noktalı ve iki noktalı çaprazlama yöntemleri yaygın olarak kullanılmakta olup bu tez çalışmasında da bu iki yöntem kullanılmıştır. Tek noktalı çaprazlama işlemi için seçilen her iki kromozomun da aynı gen sayısına sahip olması gerekmektedir. İki noktalı çaprazlama işlemi için kromozomun iki noktası seçilir kesilir ve karşılıklı olarak pozisyonlar yer değiştirir (Fırlı, 2002).

Çaprazlama İşlemi; Parça değişim örnekleri için parametre boyutu 8 olan  $X_1$  ve  $X_2$  olmak üzere iki örnek çözüm kümesi aşağıda verilmiştir. Bu iki örnek çözüm kümesi üzerinden bu tez kapsamında kullanılan tek ve çift noktalı çaprazlama yöntemleri anlatılmıştır.

$$X_1 = [1.1, -2.1, 3.0, 4.2, -2.5]$$

$$X_2 = [2.2, 0.5, -4.4, -1.5, 1.2]$$

**Tek Noktalı Çaprazlama Yöntemi:** Popülasyondaki çözüm kümeleri rasgele eşleşirler. Seçilen her çözüm kümesi çifti için ilk ve son gen dışındaki genlerden biri rasgele seçilir. Seçilen gen çaprazlama noktasını gösterir. Bu noktadan sonra gelen genler her iki çözüm kümesinde karşılıklı olarak yer değiştirir. Bu işlem için çözüm kümeleri aynı uzunlukta olmalıdır.

Aşağıdaki çözüm kümesinde 2. konum çaprazlama noktası olsun. Bu noktadan sonraki genler çaprazlandığında yeni çözüm kümeleri oluşur.

$$X_1 = [1.1, -2.1, 3.0, 4.2, -2.5] \text{ İkinci gen dâhil başlangıçtan itibaren}$$

$$X_2 = [2.2, 0.5, -4.4, -1.5, 1.2] \text{ İkinci genden sonra}$$

$$1.\text{Yeni birey} = [1.1, -2.1, -4.4, -1.5, 1.2]$$

$$2.\text{Yeni birey} = [2.2, 0.5, 3.0, 4.2, -2.5]$$

**Çift Noktalı Çaprazlama Yöntemi:** Bu yöntemde çözüm kümesi üzerinde ilk ve son genler hariç iki tane rasgele nokta seçilir. Çaprazlama işlemi seçilen bu iki gen arasındaki genlerin yer değiştirmesidir. Aşağıdaki çözüm kümeden seçilen iki nokta arasındaki genler birinci çözüm kümesinden kalan genler 2. çözüm kümesinden alınarak yeni çözüm kümeleri oluşturulur. Örnek noktalarımız 2 ve 4 olsun:

$$X_1 = [1.1, -2.1, 3.0, 4.2, -2.5] \text{ İkinci gen sonundan dördüncü gen sonuna}$$

$$X_2 = [2.2, 0.5, -4.4, -1.5, 1.2] \text{ Kalan kısım}$$

$$1.\text{Yeni birey} = [2.2, 0.5, 3.0, 4.2, 1.2]$$

$$2.\text{Yeni birey} = [1.1, -2.1, -4.4, -1.5, -2.5]$$

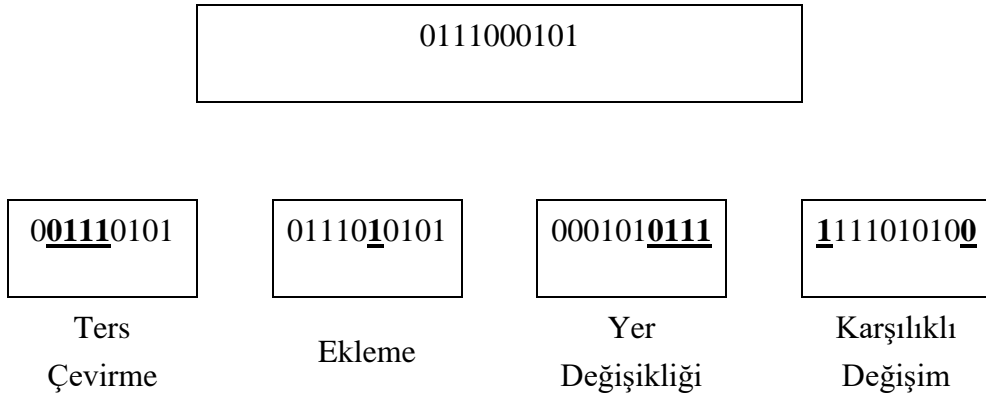
#### 5.2.1.6. Mutasyon Operatörü

GA da zamanla iyi bireylerin korunması esnasından dolayı gen dizilimi birbirine benzemeye başlar bu da çeşitliliğin yani daha geniş veri uzayında arama işlemini olumsuz etkiler. Çeşitliliğin azalması ile yeni bireyler yani

yeni çözüm kümelerinin oluşturulması durabilir (Öztürk, 2007). Çeşitliliği arttırmak amacı ile mutasyon operatörü geliştirilmiştir.

Probleme bağlı olarak geliştirilen farklı mutasyon tekniklerinden faydalanılır. Bu teknikler; ekleme, ters çevirme, yer değiştirme, karşılıklı değişim yöntemlerinden biri olarak tercih edilebilir.

Mutasyon operatörünün kullanım yöntemleri Şekil 5.3’de verilmiştir. Şekilde altı çizili değerler mutasyona uğramış genleri göstermektedir. Ters çevirme, kromozom üzerinde belirlenen rasgele iki nokta arasındaki genlerin tersten yazılmasıdır. Ekleme, kromozom üzerinde rasgele seçilen bir genin rasgele seçilen bir pozisyona yerleştirilmesidir. Yer değişikliği, yine rasgele seçilen iki nokta arasında yer alan genlerin rasgele seçilen bir pozisyondan itibaren yerleştirilmesidir. Karşılıklı değişim, kromozom üzerinde rasgele seçilen iki genin karşılıklı olarak yer değiştirmesi işlemidir (Bolat et al., 2004). Bu tez kapsamında kromozom verilerinin ikili kodlama yönteminin kullanılması ve her pozisyonun anlamı ve değer aralığının birbirinden farklı olması nedeni ile ekleme yöntemi kullanılmış tek bir gen üzerinde belirlenen değer aralıkları temel alınarak mutasyon işlemi gerçekleştirilmiştir.



Şekil 5.3 Mutasyon Operatörü Kullanım Yöntemleri (Bolat et al., 2004)

Çaprazlamanın amacı çözüm kümesinde yer alan her bir genin potansiyelini araştırmaktır. Fakat mevcut arama uzayında tam taranma sağlanamaz ise istenen çözüme gitmek imkânsız olmaktadır. Mutasyon var olan kromozomlar dışında yeni kromozomlar üreterek çeşitliliği arttırmakta ve daha geniş alanda arama gerçekleştirmektedir. GA yapısında iyi çözüme sahip çözüm kümesinin kaybını engellemek için mutasyon oranı genelde küçük tutulmaktadır (Goldberg, 1989).

### **5.2.1.7. Göç Operatörü**

Dağıtık problemlere çözüm geliştirmek amacı ile geliştirilen paralel genetik algoritmaların genetik algoritmadan çok daha başarılı olduğu yapılan çalışmalarla desteklenmiştir. Bu başarıya katkı sağlayan en önemli unsur göç operatörüdür. Göç, belli aralıklarla belirlenen sayıda bireylerin bir alt popülasyondan diğerine taşınması olayıdır (Hiroyasu et al., 1999). Böylece optimizasyonda önemli problem olan yerel en iyilere takılma ihtimalini azaltmaktadır (Rebaudengo and Reorda, 1992). Buradan yola çıkarak karmaşık sistemlerin büyük arama uzayında GA'nın daha geniş alanda arama yapmasını sağlamak amacı ile bu tez çalışmasında göç operatörü kullanılmıştır.

Orijinal GA kullanıldığında ETMB model parametrelerini ayarlama işleminde belirli bir süre içinde yerel en iyilere sıklıkla takıldığı gözlemlenmiştir. Bunun nedeni limitli iterasyon sayısında çok fazla mutasyon gerçekleştirilememesi, birey sayısının sınırlı olması yani başlangıçta üretilen 10 çözüm kümesi üzerinde işlem yapılmakta ve kısa sürede popülasyon aynı çözüm kümesi ile dolmaktadır. Bu da yerel en iyiye takıldığını gösteren bir problemdir. Bu problemi aşmak amacı ile göç operatörü genetik algoritmaya eklenmiştir.

Göç yönteminin kullanımı paralel genetik algoritmadan biraz farklı olmakla birlikte daha geniş alanda arama yapıp daha iyi çözümlere ulaştığını bu tez çalışmasında kanıtlamıştır. Göç operatörü, ters rulet tekerleği yöntemi ile (en kötü bireylerin seçilme ihtimalinin yüksek olması yöntemi) bulunan en kötü uygunluğa sahip birey popülasyondan çıkarılır yerine rassal üretilen çözüm kümesi üretilerek popülasyona dâhil edilir. Göç sayısı her bir iterasyonda 1 kere gerçekleştirilecek şekilde ayarlanmıştır.

### **5.2.1.8. Yeni Kuşağın Oluşması ve Döngünün Durdurulması**

Çaprazlama ve mutasyon işlemlerinden sonra yeni nesil oluşmaktadır. Oluşan yeni neslin uygunluk değeri bulunarak algoritmanın işleyişi devam eder. Seçilme, çaprazlama mutasyon gibi operatörler yeni nesillere uygulanır ve bu döngü belirlenen iterasyon sayısına ulaşıncaya ya da bitirme kriteri sağlanıncaya kadar devam eder (Yeo ve Agyel, 1996).

## **5.2.2. Genetik Algoritmanın Çalışması**

Genetik Algoritmanın çalışma adımları aşağıda verilmektedir;

- i. Olası çözümlerin kodlandığı bir başlangıç popülasyonu oluşturulur. Popülasyonda bulunacak birey sayısı için bir standart yoktur, problemin türüne göre bu sayı değişebilir. Her bireyin sahip olduğu gen değerleri 5.1.1'deki eşitliğe göre atanabilir. Gen sayısı çözülecek problemin parametre sayısıdır.

$$P_{ik} = \min P_j + \text{rand}(0,1) \times (\max P_j - \min P_j) \quad 5.1.1$$

Seçilen çözüm kümesinin,  $i$  indeks numarasını,  $k$  seçilen parametrenin indeks numarasını,  $P_{ik}$  geçerli parametre numarasını,  $\text{rand}(0,1)$  0 ile 1 arasında rasgele sayıyı,  $\min P_j$  Parametrenin alabileceği minimum sayıyı,  $\max P_j$  Parametrenin alabileceği maksimum sayıyı ifade eder.

- ii. Her kromozomun (bireyin) uygunluk değeri, uygunluk fonksiyonu (probleme ait fonksiyon kullanıcı tarafından oluşturulur kritik parametre değişkenleri dikkate alınır) kullanılarak hesaplanır. Genetik algoritmanın başarısı çoğu zaman bu fonksiyonun iyi tespit edilmesine bağlıdır. Bu adımda en iyi birey yerel çözüm olarak kaydedilir.
- iii. Seçilim yüzdesi kadar en iyi uygunluğa sahip bireylerin seçilmesi gerekmektedir. GA'da en çok tercih edilen Turnuva ve Rulet Tekerleği metodu kullanılarak seçim yapılır.
- Turnuva Metodu: Bu yöntemde popülasyondan her seferinde seçilen rasgele dizilerden ikişer uygunluk değerlerinin karşılaştırılması ve yeni diziyeye atılması yöntemidir. Her birey 2 kez turnuvaya katılır.
  - Rulet Tekerleği Metodu: Rulet tekerleğinde daha iyi uygunluk değerine sahip bireylerin tekerlek üzerinde kapladığı alan yüzdesi daha büyüktür böylece üretilen rasgele sayıların büyük alanda çıkma ihtimali yüksek olur. Böylece daha iyi uygunluğa sahip bireylerin seçilme ihtimali yüksek olur.
- iv. Seçim işleminden sonra elde edilen bireyler rasgele eşleştirilir ve seçilen iki bireyin kromozomları arasında parça değişimi (Çaprazlama) işlemi gerçekleştirilir. Çaprazlama işlemi için Tek Noktalı Çaprazlama Yöntemi ya da Çift Noktalı Çaprazlama Yöntemi kullanılabilir.

- a. Tek Noktalı Çaprazlama Yöntemi: Parametre noktalarından biri rasgele seçilir. O noktaya kadar 1. bireyden o noktadan sonrası ikinci bireyden parametre değerleri alınarak yeni birey oluşturulur.
  - b. Çift Noktalı Çaprazlama Yöntemi: Bu yöntemde ilk ve son parametreler dâhil edilmeden aralarında iki nokta seçilir. Bu iki nokta arasındaki genler birinci bireyden kalan genler 2. bireyden alınarak yeni birey oluşturulur.
- v. Seçilme aşamasında göç etkisi kullanılabilir (Adar ve Kuvat, 2012). Böylece algoritmanın genel en iyiye takılması engellenerek daha büyük alanda arama yapmasına katkı sağlamaktadır. Bu özellik ters rulet yöntemiyle seçilen (yani en kötü bireylerin seçilme özelliği daha yüksek) birey popülasyondan çıkarılır yerine dışarıdan göç yani rasgele birey üretilerek popülasyona dâhil edilir. Her jenerasyonda(iterasyon) 1 kere bu olay gerçekleştirilebilir. İsteğe bağlı olarak bu oran arttırılabilir.
- vi. Bir sonraki aşama mutasyondur, oluşturulan yeni bireyin mutasyon yüzdesine bağlı olarak 1 veya daha fazla parametresinde değişiklik yapılır.
- vii. 1. adımda parametre değer aralığı belirlenmiş ve parametre değerleri atanmıştı. Bu adımda isteğe bağlı olarak belirlenen aralığın aşılıp aşılmadığı kontrol edilebilir.
- viii. Durdurma kriteri olarak daha önceden belirlenmiş jenerasyon(iterasyon) sayısı veya herhangi bir ideal küresel çözüm verilebilir. Durdurma kriteri test edilir. Eğer kriter sağlanıyorsa algoritma durdurulur ve son olarak elde tutulan yerel çözüm, küresel çözüm olarak alınır. Kriter sağlanmıyorsa sonraki adıma geçilerek yeni çözümler aramaya devam edilir. Eğer kriter sağlanıyorsa 2. Adıma dönülerek işlemler tekrar eder.

---

### Algoritma 5.1 Genetik Algoritma

---

*Popülasyon Boyutunu Belirle;*

*Seçim Yüzdesini Belirle;*

*Seçim Yöntemini Belirle;*

*Çaprazlama Yöntemini Belirle;*

*Mutasyon Oranını belirle;*

**For** *Popülasyondaki tüm bireyler için*

*Bireylerin kromozomlarındaki genleri rasgele belirle (Eşitlik 5.1.1)*

**End for**

**Do**

**While** *Seçim yüzdesi kadar*

*Seçim yöntemiyle iki birey seçimi yap*

*2 birey arasında parça değişimi yap*

*Mutasyon koşulunu uygula*

*Göç Koşulunu uygula*

**End While**

*Sonuçları karşılaştır en iyi çözüm kümesini bul*

**While** *Maksimum iterasyon sayısına veya optimum çözüme ulaşıncaya kadar devam et.*

---

### 5.3. Parçacık Sürü Optimizasyonu (PSO) Algoritması

1995 yılında Russell Eberhart ve James Kennedy tarafından geliştirilen PSO algoritmasının temeli kuş ve balık gibi toplu hareket eden hayvanların sürü zekâsına ve sürü davranışına dayanmaktadır (Kennedy and Eberhart, 1995). Bu algoritmanın mantığı evrimsel ve genetik algoritma gibi evrimsel yapıya dayanmaktadır. Sürüdeki her bir bireyin birbirleri ile olan iletişimi önemli olup yiyecek ararken, tehlikeden kaçarken ya da ani yön değişimlerinde birbirleri ile olan uyumlarının mükemmel olması bu algoritmanın esin kaynağı olmuştur. Böylece Parçacık sürü optimizasyonu (PSO) algoritması, karmaşık problemlerin çözümü için basit davranışa sahip bireylerin bir araya gelmesi ile çözüme ulaşılabileceği varsayımı üzerinde durulmuştur. Geliştirilme amacı sürü davranışlarını incelemek olan bu algoritma son yıllarda sıklıkla lineer olmayan çok boyutlu ve çok değişkenli optimizasyon problemlerin çözümünde kullanılmaktadır (Küçükdeniz, 2009).

Sürü (Swarm), birbirleriyle dolaylı ya da doğrudan iletişim ve etkileşim içinde olan, var olan problemi çözmek amaçlı bir araya gelmiş, çevreleriyle

uyumlu, belirli ya da belirsiz sayıda topluluklara verilen addır (Khodier and Al-Aqeel, 2009; Kennedy et al., 2001). Her birey Parçacık ya da çözüm kümesi her birey topluluğu da popülasyon olarak nitelendirilir (Kennedy and Eberhart, 1995). Amaç var olan problemi çözmektir. Bu problem bir fonksiyon olabileceği gibi bu tez çalışmasında optimize edilmeye çalışılan karmaşık sistem modellerinin parametre ayarlama işleminde olabilir. PSO amacı, bireyin arama uzayında çözüm aramasıdır denilebilir. Bu algorithmada her bir bireyin iki tür bilgisi mevcuttur. Bilgilerden biri bireyin o an sahip olduğu bilgi değeridir sürünün sahip olduğu en iyi bilgidir (Küçükdeniz, 2009). Birey kendi bilgisi ile sürünün en iyi bilgisini kullanarak her adımda daha iyi çözüme ulaşmaya çalışır.

### 5.3.1. PSO'da Kullanılan Kavramlar

Parçacık sürü optimizasyonunun, kolay anlaşılır olması nedeni ile günümüzde oldukça sık kullanılmaktadır. PSO algoritmasını diğer MSA'lardan ayıran önemli diğer bir özelliği ayarlanması gereken parametre sayısının az olmasıdır.

Temel Kavramlar ve Parametreler;

**Konum vektörü ( $X_{ik}$ ):** Bu değer her bir parçacığı yani çözüm kümesini temsil etmektedir. Bu tez çalışmasında çözüm kümesinde yer alan her bir eleman bir model parametresine karşılık gelmektedir. Konum vektörünün popülasyon içindeki ifadesi  $X_{ik}$  [ $X_{i1}, X_{i2}, \dots, X_{ij}$ ] şeklindedir. k. iterasyonda i. parçacığın konum vektöründeki 1. eleman  $X_{i1}$  ile ifade edilir.

**Hız vektörü ( $V_{ik}$ ):** Hız vektörü formülü kullanılarak rasgele üretilen çözüm kümelerinin iyileştirilmesi hedeflenir. Böylece yeni çözüm kümeleri yani parçacığın bir sonraki konumu belirlenmiş olur. Hız vektörünün popülasyon içindeki ifadesi  $V_{ik}$  ( $V_{i1}, V_{i2}, \dots, V_{ij}$ ) şeklindedir. k. iterasyonda i. parçacığın hız vektöründeki 1. eleman  $V_{i1}$  ile ifade edilir. Hız vektöründe dikkat edilmesi gereken diğer bir husus vektör bileşenlerinin  $[-vmax, vmax]$  aralığında olması gerekliliğidir.

$Vmax$  sınırı herhangi bir zaman diliminde hareket eden parçacığın hızını belirli sınırlar arasında tutmak amaçlı kullanılır. Hız vektörlerinin çok büyük adımlara sahip olması çözüm kümesinin iki çözümü arasındaki mesafeyi arttırmakta ve eğer varsa arada daha iyi çözümlerin işleme alınmamasına neden olmaktadır. Eğer adımlar çok küçük ise bu sefer aynı noktanın çevresinde aramanın

uzun sürmesine ve aynı çözüm kümelerinin etrafında dolaşmasına sebep olmaktadır. Sınırlı iterasyonda çözüm bulunması imkânsız hale gelir.  $V_{max}$  iki durumu dengeleyecek şekilde belirlenmelidir (Eberhart ve Shi, 2001).

Eğer  $V_{ij} > V_{max}$  ise  $V_{ij} = V_{max}$

Eğer  $v_{ij} < -v_{max}$  ise  $v_{ij} = -v_{max}$  kuralına göre atama yapılır.

**Popülasyon Boyutu (Çözüm Kümesi Sayısı):** Sürüde yer alan birey sayısı yani popülasyonun büyüklüğüdür. Problemden probleme farklılık göstermektedir. Genel olarak yapılan çalışmalar incelendiğinde popülasyon boyutu 10-50 arasında seçilmektedir. Popülasyon ne kadar büyük ise çözüm uzayındaki arama o kadar fazla ve farklı noktalardan arama yapabilmektedir. Ancak algoritmanın işleyiş süresini uzatması maliyeti arttırmaktadır. Probleme göre en uygun değer seçilmelidir.

**Parametre boyutu (Çözüm Kümesinin Eleman Sayısı):** Problemin bilinmeyen parametre değerleri olarak ifade edilmektedir. Bu tez çalışmasında modele ait parametre sayısı parametre boyutunu belirleyecektir.

**Parametre Aralığı:** Parametrelerin alabileceği maksimum ve minimum değerlerdir. Bu değerleri en iyi model tasarımcısı belirleyeceği gibi. Optimizasyon algoritmaları veya bir takım fonksiyonlar kullanılarak da belirlenebilir. Probleme göre değişiklik gösterir.

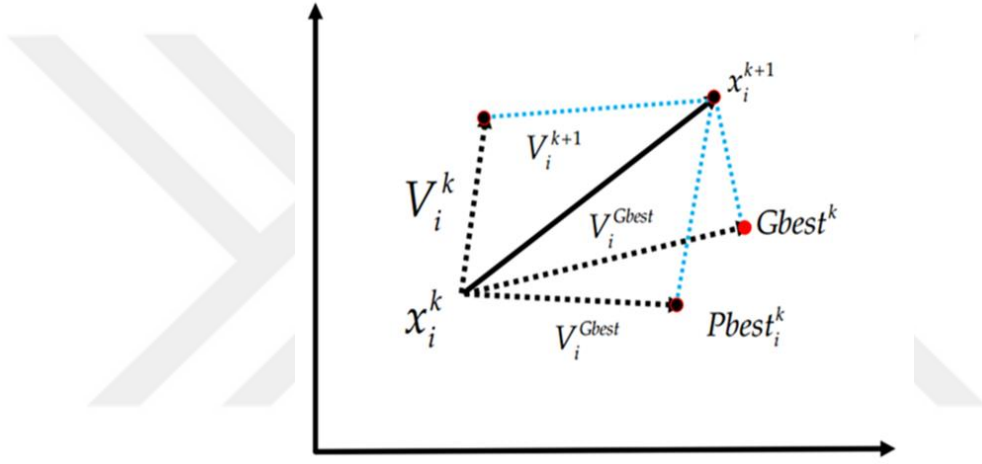
**Öğrenme faktörleri ( $c_1, c_2, \dots, c_n$ ):** Sosyal ve kavramsal değerleri ifade eder. Var olan bilgi oranlarını belirlerler. Sıklıkla kullanılan  $c_1$  ve  $c_2$  hızlandırma ya da öğrenme sabitleri olarak nitelendirilir. PSO algoritmasında oldukça önemli olan bu sabitler bireyin en iyi pozisyonu ( $p_{best}$ ) ve global en iyi pozisyon ( $g_{best}$ ) üzerinde doğrudan etkileri vardır (Ricardo et al., 2007).

$$\text{Hız vektörü; } \mathbf{v}_i^{k+1} = w\mathbf{v}_i^k + c_1 \text{rand}_{1k} (\mathbf{p}_{best}^k - \mathbf{x}_i^k) + c_2 \text{rand}_{2k} (\mathbf{g}_{best}^k - \mathbf{x}_i^k)$$

Hız vektörü eşitliğinde de görüldüğü gibi, öğrenme faktörleri parçacığın kendi en iyi konumuna uzaklığı ile global en iyi konuma uzaklığına ne kadar ağırlık vereceği belirler. Bu durumda  $c_1$ , kendi geçmiş en iyi konuma yaklaşmayı sağlayan oranı,  $c_2$  ise global en iyi konuma yaklaşmayı sağlayan oranı belirlerler. PSO geliştiriciler bu iki sabitin değerini  $c_1=c_2=2$  olarak belirlemişlerdir (Eberhart ve Shi

2001). Bu değerler probleme göre farklı kullanılabilir. Problem için en uygun değerlerin seçilmesi zordur. Eğer bu değerler daha küçük alınırsa kendi ve popülasyonun en iyi konumlarına ulaşması yavaşlayacaktır. Eğer daha büyük değerler alınırsa kendi ve popülasyonun en iyi konumuna yaklaşma hızları artacak bu da yerel en iyilere takılma sebebi olacaktır. En uygun değerlerin seçilmesi PSO algoritması için kritik öneme sahiptir.

Şekil 5.4 de görüldüğü gibi,  $X_i^k$  parçacığın en iyi konumu geçmiş en iyi konum vektörü  $V_i^{Pbest}$  ve popülasyonun en iyi hız vektörü  $V_i^{Gbest}$  ve mevcut hız vektörünün  $V_i^k$  sonucu olarak mevcut konumu güncellenir  $X_i^{k+1}$ .



Şekil 5.4 Parçacığın Arama Uzayındaki Hareketi (Hosseini et al., 2014)

**Atalet ağırlığı (w):** Önceki hız vektörünün, mevcut hız vektörü üzerindeki etkiyi kontrol eder. Bu değer global yayılmayı ya da yerel en iyeye takılmayı kontrol etmek amacı geliştirilmiştir (Shi ve Eberhart 1998a). Bu değer kullanılması yerel ve global aramalar arasında denge kurmakta ve optimum değere ulaşmak için mevcut iterasyon sayısını azalttığı görülmüştür. Bu değer önemi aşağıdaki eşitlik üzerinde inceleyelim;

$$\Delta \mathbf{v}_i = \mathbf{v}_i^{k+1} - \mathbf{v}_i^k = w\mathbf{v}_i^k + c_1 \mathbf{rand}_1^k \cdot (\mathbf{pbest}_i^k - \mathbf{x}_i^k) + c_2 \mathbf{rand}_2^k \cdot (\mathbf{gbest}^k - \mathbf{x}_i^k) - \mathbf{v}_i^k$$

$$= w\mathbf{v}_i^k + \mathbf{f}_i - \mathbf{v}_i^k = \mathbf{f}_i - (1 - w)\mathbf{v}_i^k$$

Bir parçacığın hız vektöründeki değişim,  $\Delta \mathbf{v}_i$  hız değişimini göstermektedir.  $\mathbf{f}_i = c_1 \mathbf{rand}_1 (\mathbf{pbest}_i^k - \mathbf{x}_i^k) + c_2 \mathbf{rand}_2 (\mathbf{gbest}^k - \mathbf{x}_i^k)$  olarak atanmıştır. Eşitlik

sadeleştirildiğinde görülüyor ki atalet ağırlığının ( $w$ ) değerinin büyümesi hız vektöründeki değişimi arttırmaktadır. Bu durumda  $w$  değerinin büyümesi global aramayı küçük olması yerel aramayı desteklemektedir.  $w$  değeri belirlenirken ilk değer arama uzayında büyük adımlarla ilerlemeyi sonlara doğru bu adımların küçülmesi yani detaylı arama yapılmasına yönelik verilmesi PSO algoritmasının kalitesini arttıracaktır (Riccardo et al., 2007).

**Random değerler ( $r_1, r_2, \dots, r_n$ ):** belirlenen aralıkta rasgele değerler olarak düzgün dağılım sağlar.

**Yerel en iyi değer (pBest) :** i. Çözüm kümesinin o ana kadar elde etmiş olduğu en iyi çözümünü temsil eder.

**Küresel en iyi değer (gBest):** O ana kadar bulunan en iyi çözüm değeridir.

**Bitirme kriteri:** Maksimum iterasyon ulaşıldığında ya da istenen optimum sonuç bulunduğu algoritmanın çalışmasını bitirecek olan kriterdir (Gencer, 2010).

### 5.3.2. PSO Algoritmasının Çalışması

PSO'nun çalışma adımları aşağıda verilmektedir.

- i. PSO ilk olarak rasgele üretilen başlangıç pozisyonları ve hızları ile çözüm kümesini oluşturur. Çözüm kümesi (Parçacık)  $N$  adet elemandan oluşur. i. Parçacık  $x = [x_{i1}, x_{i2}, \dots, x_{iN}]$  şeklinde gösterilir. Çözüm kümesindeki her bir eleman problem boyutuna ya da parametrelerine karşılık gelir. Parçacığın her bir elemanı genetik algoritmadaki 5.1.1 eşitliği kullanılarak atanabilir.
- ii. Elemanların değerleri atandıktan sonra modele ya da uyarlanacak başka probleme bağlı olan uygunluk fonksiyonu kullanılarak her bir parçacığın uygunluk değeri hesaplanır (bu fonksiyon modele, probleme ya da kullanıcının beklentisine bağlı olarak değişkenlik gösterir).
- iii. Parçacıkların sürekli hız ve konumlarının güncellenmesi ile optimum çözüme ulaşmaya çalışılır. Her iterasyonda Parçacık konumları, kendi en iyi konumu (pbest) ile sürünün en iyi konumuna (gbest) göre güncellenir ve hafızada

saklanır. Her iterasyon için pbest ve gbest konum değerleri bulunduktan sonra parçacığın hızı ve konumu sırası ile 5.2.1 ve 5.2.2 eşitliğine göre güncellenir.

$$\mathbf{v}_i^{k+1} = w\mathbf{v}_i^k + c_1\text{rand}_{1k} (\mathbf{pbest}_i^k - \mathbf{x}_i^k) + c_2\text{rand}_{2k} (\mathbf{gbest}^k - \mathbf{x}_i^k) \quad 5.2.1$$

$$\mathbf{x}_i^k = \mathbf{x}_i^k + \mathbf{v}_i^k \quad 5.2.2$$

Görüldüğü üzere Parçacıkların konum değerini güncellemek için kullanılan eşitlikler basit toplam ve çarpımlardan oluşmakta olup türev bilgisi gerektirmemektedir. pbest<sub>ik</sub>, i. parçacığa ait o ana kadarki en iyi çözümü sağlayan koordinattır. gbest<sub>k</sub>, o ana kadar bulunan en iyi çözümü sağlayan koordinat değeridir. Öğrenme faktörleri olan c1 ve c2 değerleri sırayla pbest ve gbest değerlerine yaklaşma oranını belirleyen sabitlerdir. Denklemdeki rand1 ve rand2 ise 0-1 arasında rasgele atanan sayıları, k iterasyon değerini, i seçilen çözüm kümesinin indeks numarasını, l<sub>ik</sub> parçacığın konumunu, v<sub>ik</sub> oluşan yeni parçacığın hız değerini, w atalet (intertia) ağırlığı temsil etmektedir (Tamer and Karakuzu, 2006).

**Intertia Ağırlığı (Atalet Değeri) (w):** Atalet ağırlıkları önemli bir PSO parametresidir. PSO sürecinde optimum değere yakınsamada büyük etkisi olduğu yapılan araştırmalarda kanıtlanmıştır. Atalet ağırlık stratejisinin çok sayıda stratejisi mevcuttur. “Linear Decreasing Inertia Weight stratejisi” diğer stratejilere kıyasla optimum sonuçlara daha çok yaklaşmaktadır (Bansal et al., 2011). “Linear Decreasing Inertia Weight” eşitliği 5.2.3’deki gibi tanımlanmıştır.

$$w = w_{\max} - (iter) / \text{maksIte} * (w_{\max} - w_{\min}); \quad 5.2.3$$

PSO algoritmasının en önemli özelliği hızlı yakınsaması ve kolay uygulanabilir olmasıdır. PSO algoritmasının işleyiş adımları Algoritma 5.2’de verilmektedir.

---

**Algoritma 5.2 Parçacık Sürü Optimizasyonu Algoritması**


---

*Popülasyon Boyutunun belirlenmesi pop;*

*C1, C2 parametre değerlerini ata;*

**For** Her Bir Çözüm Kümesi İçin

*Parçacıkların konumlarını ve hızlarını yükle (Eşitlik- 5.2.1)*

**End for**

**Do**

**For** Her Bir Çözüm Kümesi İçin

*Uygunluk değerini hesapla*

**If** uygunluk değeri *pBest*'den iyi ise

*Geçerli değeri pBest olarak ata*

**End if**

**If** uygunluk değeri *gBest*'den iyi ise

*Geçerli değeri gBest olarak ata*

**End if**

**End for**

**For** Her Bir Çözüm Kümesi İçin

*Hız hesapla (Eşitlik 5.2.2)*

*Konum güncelle (Eşitlik 5.2.3)*

*Vmax sınırını kontrol et*

**End for**

**While** Maksimum iterasyon sayısına veya optimum çözüme ulaşıncaya kadar devam et

---

#### 5.4. Yapay Arı Kolonisi (Artificial Bee Colony-ABC) Algoritması

Yapay arı kolonisi en iyileme algoritması 2005 yılında Karaboğa tarafından geliştirilen sürü tabanlı bir optimizasyon algoritmasıdır (Karaboğa, 2005). Az sayıda parametre içermesi, kolay uygulanabilir olması ve farklı problemler için farklı versiyonlarının bulunması nedeni ile ABC algoritması çok farklı optimizasyon problemlerinin çözümünde kullanılmıştır (Karaboğa ve Baştürk, 2008; Karaboğa ve Akay, 2009; Karaboğa, 2009; Ma et al., 2011; Kashan et al., 2012; Zhu ve Kwong, 2010; Li et al., 2012).

Bu tez çalışmasında standart ABC algoritmasının her optimizasyon probleminin çözümünde başarılı olmadığı görülmüş bundan dolayı ABC algoritmasının farklı versiyonları incelenmiştir. Modifiye yapay arı kolonisi (M-ABC) algoritmasının ayarlanması gereken az sayıda parametresi olması nedeni ile kullanılmış. ETMB ortamında yer alan modellerin parametre ayarlama problem çözümünde başarı sağladığı kanıtlanmıştır. Bu bölümde Standart ve Modifiyeli ABC algoritmaları tanıtılacaktır.

#### 5.4.1. Standart ABC Algoritması

Sürü tabanlı sezgisel algoritma olan yapay arı kolonisi (*artificial bee colony* - ABC) algoritması (Karaboga and Basturk, 2008), arıların besin arama davranışlarından esinlenilerek geliştirilmiştir. İşçi, gözlemci ve kâşif arıların besin arama ve işleme davranışları algoritmaya uyarlanmıştır. ABC algoritması, az sayıda kontrol parametresine sahip olması, basitliği ve kolay geliştirilebilir olması nedeni ile birçok optimizasyon problemlerinin çözümünde kullanımı yaygınlaşmıştır (Singh, 2009; Kang et al., 2009; Samrat et al., 2010).

ABC algoritmasında, işçi (*employed*), gözlemci (*onlooker*) ve kâşif (*scout*) arı olmak üzere üç tip arı bulunmaktadır. İşçi arılar her bir yiyecek kaynağının nektar miktarını hesaplamak ile sorumludur ve popülasyondaki işçi arı sayısı beslenme alanında bulunan yiyecek kaynağı sayısına eşittir. Gözlemci arılar iyi nektar miktarına sahip yiyecek kaynağını seçmekle sorumludur ve popülasyondaki gözlemci arı sayısı işçi arı sayısına eşittir. Kâşif arılar ise yeni yiyecek kaynaklarını keşfetmek ile sorumludur. Yiyecek kaynağı tükenmiş olan işçi arılar kâşif arılara dönüşmektedir ve eski yiyecek kaynağı, kâşif arı tarafından bulunan yeni yiyecek kaynağı ile değiştirilmektedir.

Yiyecek kaynağının pozisyonu, çözülmesi hedeflenen eniyileme probleminin olası çözümünü temsil etmektedir. Yiyecek kaynağının nektar miktarının yüksek olması eniyileme probleminin olası çözümünün iyi olması anlamına gelmektedir. Dolayısı ile olası çözümün kalitesi nektar miktarı ile temsil edilmektedir ve bu değer ABC algoritmasında uygunluk değeri (*fitness value*) olarak adlandırılmaktadır.

### 5.4.2. Modifiye Yapay Arı Kolonisi (M-ABC) Algoritması

Standart ABC algoritmasının basit problemlerde sağladığı başarıyı karmaşık problemlerde gösterememesi farklı sürümlerinin geliştirilmesine neden olmuştur. ABC algoritmasının çözüme ulaşma ve hızını arttırmak adına geliştirilen MABC algoritması bu çalışmada kullanılan karmaşık sistem problemlerinde başarılı olmuştur. MABC algoritması orijinal ABC algoritmasından farklı iki parametresi vardır (Korkmaz Tan ve Bora, 2017c). Birinci parametresi *modification rate* (MR) yani besin kaynağı üretilirken tek bir besin kaynağı güncellenmesi yerine kaç tane besin kaynağının değiştirileceğini belirleyen oranı temsil eder. Eşitlik 5.3.1’de gösterilen şekilde rasgele üretilen  $R_{ik}$  değeri ile MR değerinin karşılaştırılması ile kontrol yapılır.

$$\begin{aligned} &\text{Eğer} \\ &R_{ik} < MR \text{ ise} \quad V_{ik} = X_i + \varphi_{ik} \times (X_{ik} - X_{jk}) \\ & \\ &\text{Değilse} \quad V_{ik} = X_{ik} \end{aligned} \quad 5.3.1$$

Diğer değişiklik ise ölçekleme faktörü (*scaling factor* - SF) parametresidir. ABC’de çözüm arama denkleminde rasgele üretilen sayı  $\varphi_{ik}$  değişkeni  $[-1,1]$  aralığında üretilirken, M-ABC’de  $[-SF, SF]$  aralığında rasgele üretilen değer aralığı kullanılmaktadır. SF parametresinin büyük değerler alması durumunda algoritma daha hızlı çözüme ulaşırken optimum değeri bulma ihtimali düşmektedir. Bu nedenle bu çalışmada bu aralık  $[-1,1]$  olarak kullanılmıştır. MR parametresinin değerlerindeki değişimler gözlemlenmiştir (Akay and Karaboga, 2010).

### 5.4.3. Yapay Arı Kolonisi Algoritmasının Çalışması

ABC’nin çalışma adımları aşağıda verilmektedir;

1. İlk olarak rasgele besin kaynakları yani çözüm kümeleri oluşturulur. Her kümenin eleman sayısı problemin boyutuna bağlı olarak değişmektedir. Bu elemanların değerleri rasgele oluşturulabileceği gibi isteğe bağlı olarak minimum maksimum sınırlar belirlenerek 5.3.2 eşitliği kullanılarak da atanabilir.

$$P_{ik} = \min P_j + \text{rand}(0,1) \times (\max P_j - \min P_j) \quad 5.3.2$$

Seçilen çözüm kümesinin,  $i$  indeks numarasını,  $k$  seçilen parametrenin indeks numarasını,  $P_{ik}$  geçerli parametre numarasını,  $rand(0,1)$  0 ile 1 arasında rasgele sayıyı,  $minP_j$  Parametrenin alabileceği minimum sayıyı,  $maxP_j$  Parametrenin alabileceği maksimum sayıyı ifade eder.

2. Çözüm kümeleri oluşturulduktan sonra her bir çözüm kümesinin uygunluk değeri modelin uygunluk ya da problemin uygunluk fonksiyonu kullanılarak hesaplanır. Bu fonksiyon kullanıcının modelden beklentisine bağlı olarak değişkenlik gösterebilmektedir. Bulunan Uygunluk değeri ABC algoritmasına ait uygunluk fonksiyonuna 5.3.3 uygulanarak nihai uygunluk değeri hesaplanmış olur.

$$\text{Eğer } f_i \geq 0 \rightarrow \frac{1}{1+f_i}$$

$$\text{Eğer } f_i < 0 \rightarrow 1 + |f_i| \quad 5.3.3$$

$f_i$ : Modele ya da kullanılan en iyileme problemin  $i$ . çözüm kümesinin uygunluk değeridir.

3. Oluşturulan çözüm kümelerinin her birine standart ABC algoritmasında 5.3.4 M-ABC algoritmasında 5.3.5 eşitliği uygulanır ve yeni aday çözümler elde edilir. Elde edilen çözüm kümesinin uygunluk değeri hesaplanarak eski çözüm kümesinin uygunluk değeri ile karşılaştırılır. Eğer uygunluk değerinde iyileşme varsa çözüm kümesi eskisi ile değiştirilir. Bu adımda amaç daha iyi besin kaynakları oluşturma yani çözüm kümesini iyileştirmektir.

$$\text{Standart ABC için;} \quad V_{i_k} = X_{i_k} + \varphi_{i_k} \times (X_{i_k} - X_{j_k}) \quad 5.3.4$$

$$\text{Eğer } \quad V_{i_k} = X_{i_k} + \varphi_{i_k} \times (X_{i_k} - X_{j_k}) \quad 5.3.5$$

$$\text{M-ABC için; } R_{ik} < MR \text{ ise}$$

$$\text{Değilse} \quad V_{i_k} = X_{i_k}$$

Eşitlikteki  $V_{ik}$  bulunacak yeni çözümü,  $X_{ik}$  eski aday çözümü,  $\varphi_{i_k}$   $[-1,1]$  aralığında rasgele sayıyı,  $X_{kj}$  seçilen komşu çözümü,  $R_{ik}$ , rasgele üretilen bir değeri,  $MR$ , komşu çözümün üretilmesinde tek bir elemanın değiştirilmesi yerine kaç tane elemanın değiştirileceğini belirleyen oranı temsil etmektedir.

4. Oluşan yeni çözüm kümelerinin her bir elemanı daha öncede belirtildiği gibi modele ait parametreyi içerir. Her parametre değeri belirlenen sınır değerleri içinde olmalıdır. Parametre değerlerinin belirlenen sınırlar içinde olup olmadığını kontrol etmek için eşitlik 5.3.6'ı kullanıyoruz. Böylece parametre değerlerinin belirlenen sınır değerleri içinde kalması sağlanmış olur.

$$\text{Eğer } X_{it} > ub_i \rightarrow X_{it} = ub_i$$

5.3.6

$$\text{Eğer } X_{it} < lb_i \rightarrow X_{it} = lb_i$$

$i$ , Seçilen çözüm kümesinin indeks numarasını,  $k$ , seçilen parametrenin indeks numarası,  $X_{ik}$ , Seçilen çözüm kümesinin geçerli parametre numarasını,  $lb_j$ , Parametrenin alabileceği minimum sayıyı,  $ub_j$ , Parametrenin alabileceği maksimum sayıyı temsil eder.

5. Daha sonra, gözlemci arılar, işçi arılara benzer bir şekilde yeni iyi yiyecek kaynakları bulmaya çalışırlar. Fakat gözlemci arılar işçi arılardan farklı olarak uygunluk değerlerine göre çözüm kümelerinin seçilme olasılığı belirlenir. Daha sonra rulet tekerleği yöntemi kullanılarak en iyi uygunluk değerine sahip çözüm kümeleri seçilmiş olur. Daha sonra seçilen çözüm kümelerine 5.3.7'deki formül uygulanarak daha iyi çözüm kümesi oluşturulmaya çalışılır. Eğer uygunluk değerinde iyileşme görülürse eski çözüm kümesi ile yeni çözüm kümesi değiştirilir. Eğer iyileşme olmaz ise o çözüm kümesinin limit değeri bir atılır.

$$O_i = \frac{F_i}{\sum_j F_j} \quad 5.3.7$$

$O_i$  Seçili çözüm kümesinin gözlemci arı için seçilme olasılığını,  $F_i$  Seçili çözüm kümesinin uygunluk değerini,  $\sum_j F_j$  Tüm çözüm kümelerinin uygunluk değerlerinin toplamını temsil eder.

6. Son adımda besin kaynağı tükenirse işçi arı kâşif arıya dönüşerek besin kaynağı arayışına girer. Bir besin kaynağı belirlenen limit değeri boyunca iyileştirilemezse silinir. Yerine rasgele yeni bir çözüm kümesi oluşturulur. Böylece algoritma yerel en iyilerden kaçınmış olur.

---

**Algoritma 5.3 Yapay Arı Kolonisi Algoritması**

---

*Popülasyon boyutunu belirle;*

*M parametre değerini ata.*

**For** tüm besin kaynakları için

*Besin kaynaklarını yükle (Eşitlik 5.3.1)*

*Uygunluk hesapla (Eşitlik 5.3.2)*

**End for**

**Do**

**For** tüm işçi arılar için

*Besin kaynağını işle yeni uygunluk değeri bul (Eşitlik 5.3.5)*

*Parametre aralıklarını kontrol et (Eşitlik 5.3.6)*

**If** yeni uygunluk daha iyiyse

*Çözüm kümesini değiştir*

**End if**

**End for**

**For** tüm Parçacıkları için

*Olasılıkları belirle (Eşitlik 5.3.7)*

**End for**

**For** tüm gözcü arılar için

**If** rasgele oluşturulan sayı olasılıktan büyük ise

*Besin kaynağını işle yeni uygunluk değeri bul (Eşitlik 5.3.5)*

*Parametre aralıklarını kontrol et (Eşitlik 5.3.6)*

**If** yeni uygunluk daha iyiyse

*Çözüm kümesini değiştir*

**End if**

**End if**

**End for**

**For** tüm işçi arılar için:

**If** limit sayısına ulaşıtıysa (çözüm kümesi belirlenen limit sayısınca

*İyileştirilemedi ise)*

*Kâşif arıya çevir*

**End if**

**End for**

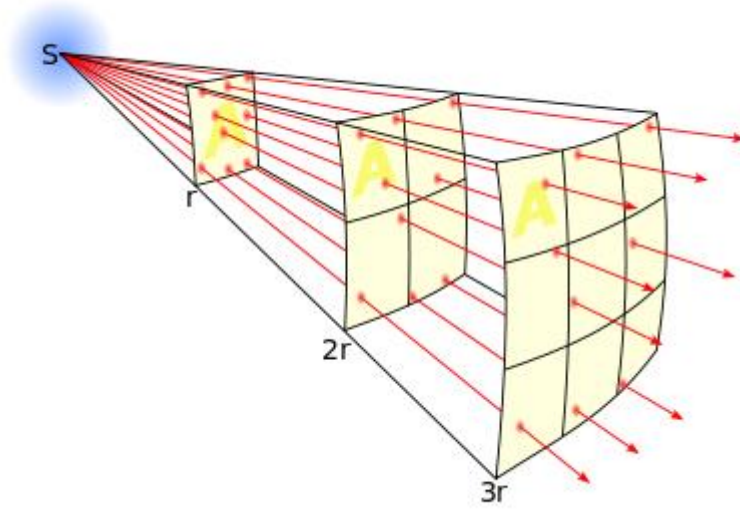
**While** Maksimum iterasyon sayısına veya optimum çözüme ulaşıncaya kadar devam et

---

### 5.5. Ateş Böceği Algoritması (Firefly Algorithms-FA)

Sezgisel optimizasyon algoritmalarından biri olan ateş böceği algoritması (Firefly Algorithms-FA) Krishnanand ve Ghose tarafından 2005 yılında geliştirilmiştir (Krishnanand ve Ghose, 2005). Bu algoritma da genetik algoritma gibi popülasyon temelli bir algoritmadır. Ateş böceklerinin davranışları gözlemlenerek algoritmaya uyarlanmıştır. Bu algoritmayı diğer algoritmalarından avantajlı kılan en önemli özelliği birden çok farklı alana yerleştirilen bireylerin kullandıkları dinamik karar yapılarıdır. Her bir birey karar yapılarını kullanarak diğer bireyin parlaklığı temel alınarak yönelme gerçekleşmektedir. Ateş böcekleri eşlerini ve avlarını çekmek için çevreye yanıp sönen ışık yayarlar. Ateş böceğinin çekiciliği parlaklığı ile doğru orantılıdır. Ateş böceği algoritması, ateş böceklerinin sosyal davranışlarının incelenmesi sonucu geliştirilen meta-sezgisel optimizasyon algoritması olarak da tanımlanabilir (Yang, 2009). Diğer sürü tabanlı algoritmalara benzeyen bu algoritmanın probleme uyarlanması daha kolaydır. Ateş böceklerinin davranışları bu algoritmaya esin kaynağı olmuştur. Ateş böceklerinde cinsiyet ayırımının olmaması ve yanıp sönen ışıkları ile birbirlerini etkileyip çekmeleri algoritmanın temelini oluşturmaktadır. Ateş böceğinin sahip olduğu ışık şiddetini probleme ait uygunluk fonksiyonun üretmiş olduğu uygunluk değeri temsil etmektedir. Parlaklık ile çekicilik doğru orantılı olup böcekleri en parlak ateş böceğine doğru yönlendirir. Eğer ortamda kendisinden daha parlak ateş böceği göremezlerse rasgele hareket ederler parlak ateş böceği bulma ihtimallerini arttırırlar (Apostolopoulos and Vlachos, 2011; Yang, 2010).

Parlaklıkta önemli olan diğer bir etken Şekil 5.5'de görüldüğü gibi ateş böcekleri arasındaki mesafedir. Ateş böcekleri arasındaki mesafe arttıkça ışık şiddeti de azalmaktadır. Fizik'de fiziksel şiddet kaynağından uzaklaştıkça kaynağından uzaklığının karesi kadar şiddetini kaybeder. Işık için aynı kanun geçerlidir mesafe arttıkça ışık şiddetini kaybeder. Denilebilir ki ateş böceğinin parlaklığını daha uzaktaki ateş böceği daha az görebilir. Ateş böceği algoritmasında ışık şiddeti eşitlik 5.4.2'deki gibi hesaplanır. Formülden de anlaşılacağı üzere ışık şiddetini mesafenin karesi ile ters orantılıdır. Aşağıdaki şekil bu fiziksel olayı temsil etmektedir yani fizikte ters kare kanununun gösterimidir.



Şekil 5.5 Ters Kare Kanunu

### 5.5.1. Ateş Böceği Algoritmasının Çalışması

Bir optimizasyon probleminde  $N$  tane ateş böceği yani çözüm kümesi kullanılmaktadır. Her çözüm kümesi problem boyutu kadar eleman içerir. Belirlenen aralıklarda ya da aralık belirtmeksizin rasgele değerlerle iliklenmektedir. Elde edilen çözüm kümesinin uygunluk değeri problemin uygunluk fonksiyonunda hesaplanır. Uygunluk değeri parlaklığı temsil eder. En iyi uygunluğa sahip olan ateş böceği en parlak ateş böceğidir.

FA Algoritmasının çalışma adımları aşağıda verilmektedir.

- i. İlk aşama olarak her çözüm kümesinin her bir elemanı için (eleman sayısı problem boyutuna bağlı) eleman sayısı kadar belirlenen aralıklarda rasgele çözüm kümesi oluşturulur. Bu değerler isteğe bağlı olarak 5.4.1 formülü kullanılarak atanabilir.

$$X_{i_k} = lb_j + rand(0,1) \times (ub_j - lb_j) \quad 5.4.1$$

$i$ , seçilen çözüm kümesinin indeks numarasını,  $k$ , seçilen parametrenin indeks numarasını,  $X_{i_k}$ , seçilen çözüm kümesinin geçerli parametre numarasını,  $rand(0,1)$ , 0 ile 1 arasında rasgele sayıyı,  $lb_j$ , parametrenin alabileceği minimum sayıyı,  $ub_j$ , parametrenin alabileceği maksimum sayıyı temsil eder.

- ii. Bu adımda ateş böceklerinin ışık şiddeti, ateş böcekleri arasındaki uzaklık ve ışık yoğunlukları sırası ile 5.4.2, 5.4.3, 5.4.4 eşitlikleri kullanılarak hesaplanır.

$$I = 1/r^2 \quad 5.4.2$$

I, Işığın şiddetini, r, iki ateş böceği arasındaki uzaklığı temsil eder.

$$r_{ij} = |x_i - x_j|^2 = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \quad 5.4.3$$

$r_{ij}$  = i. ve j. ateşböceği arasındaki uzaklığı, d = eleman sayısını,  $x_{i,k}$  = i. ateş böceğinin çözüm kümesinin k. elemanını,  $x_{j,k}$  = j. ateş böceğinin çözüm kümesinin k. elemanını temsil eder.

$$I(r) = I_0 e^{-\gamma r^2} \quad 5.4.4$$

$I_0$ , başlangıç ışık yoğunluğunu (*light intensity*),

$\gamma$ , sabit emilim katsayısı (*light absorption coefficient*), r ise iki ateş böceği arasındaki uzaklık değerini temsil eder.

- iii. Bir ateş böceğinin çekiciliği (*attractiveness*), diğer ateş böceğinin parlaklığına ve uzaklığına bağlıdır. Bu nedenle çekicilik 5.4'deki eşitlikle hesaplanır.

$$\beta = \beta_0 e^{-\gamma r^2} \quad 5.4.5$$

$\beta$ , ateş böceğinin çekicilik değeridir.  $\beta_0$ , iki ateş böceği arasındaki uzaklığın 0 olduğundaki çekicilik değeridir.  $\beta_0$ , 0 ile 1 arası değer alabilmektedir.

- iv. Ateş böceklerinin daha parlak ateş böceklerine doğru hareket etmesi, her bir ateş böceği diğer tüm ateş böceklerini kontrol eder ve kendinden parlak olan ateş böceklerine 5.5 eşitliğini kullanarak hareket ederler.

$$x_{i,p} = x_{i,p} + \beta (x_{j,p} - x_{i,p}) + \alpha \epsilon_{i,p} \quad 5.4.6$$

$x_{i,p}$ , i. ateş böceğinin p. eleman değerini,  $x_{j,p}$ , j. ateş böceğinin p. eleman değerini,  $\alpha$  = rastlantı değişkenini genelde [0,1] arasında bir değer alır,  $\epsilon_i$ , gauss

(gaussion) dağılımı ile belirlenmektedir fakat genel olarak  $[-0.5,0.5]$  aralığında rasgele bir değeri temsil eder.

- v. Eğer ateş böceği kendinden daha parlak ateş böceği bulamazsa 5.6 eşitliğine göre rasgele hareket ederler.

$$x_{i,p} = x_{i,p} + \alpha \epsilon_{i,p} \quad 5.4.7$$

Maksimum iterasyona ulaşıncaya kadar döngü 2. Adımdan itibaren programı tekrar tekrar işletilir. Elde edilen sonuçlar karşılaştırılarak en iyi çözüm kümesi bulunmuş olur.

---

#### **Algoritma 5.4 Ateş Böceği Algoritması**

---

*Popülasyon boyutunu belirle;  
Y,α değerlerini ata;*

**For** tüm ateş böcekleri için

*Ateş böceklerinin rasgele yükle (Eş. 5.4.1)*

**End for**

**Do**

**For**  $i = 1:n$  (tüm ateş böcekleri için)

**For**  $j = 1:n$  (tüm ateş böcekleri için)

**If**  $I_i < I_j$  ise (j ateş böceği daha parlak ise)

*j ateş böceğinin i ateş böceğine uzaklığını bul*

(Eş.-5.4.3)

*j'nin çekiciliğini bul (Eş.5.4.5)*

*i ateş böceğini j ateş böceğine doğru hareket ettir*

(Eş. 5.4.6)

**Else**

*i ateş böceğini rasgele hareket ettir (Eş. 5.4.7)*

**End if**

**End for**

**End for**

*Sonuçları karşılaştır en iyi çözüm kümesini bul*

**While** iterasyon sayısına ulaşıncaya kadar veya istenilen çözüme ulaşıncaya kadar

---

## 6. GELİŞTİRİLEN ADAPTİF META-SEZGİSEL ALGORİTMALAR

Bu tez çalışmasında 5. Bölümde açıklanan GA, PSO, ABC ve FA algoritmaları ETMB ortamında modellenen karmaşık sistem parametrelerinin ayarlanması problemine çözüm sunmak amacı ile kullanılmıştır. Ancak karşılaşılan ikinci problem kullanılan MSA'ların kritik parametre değerlerinin de ayarlanması gerekliliğidir. MSA'ların kritik parametre değerleri farklı modellerde farklı değerler almaktadır.

Kullanılan MSA'lar incelendiğinde kendilerine ait parametrelerin olduğu ve bu parametre değerlerinin manuel olarak ayarlandığı 5. Bölümde sözde algoritma kodlarında açıkça görülmektedir. Bu parametre değerleri her problem için farklı değerler almakta olup manuel ayarlanmaya çalışılmaktadır. MSA'larda parametre değerlerini otomatik bulabilecek bir mekanizma MSA'ların optimuma yaklaşma hızını arttıracaktır. Geliştirilen MSA'lar için önerilen bir takım sabit kritik parametre değerleri mevcut olmasına rağmen farklı problemlerde istenen sonuca varmayı ve algoritmanın performansını maalesef olumsuz etkilemektedir. En uygun parametre değerini bulmanın birinci yolu olası bütün değerlerin kullanılarak algoritmanın çalıştırılması ve sonuçların incelenmesi ile olabilecek en iyi değerlerin bulunmasıdır. Fakat algoritmaya ait parametre sayısı ve her bir parametrenin alabileceği değer aralığı düşünüldüğünde ve her seferinde algoritmanın çalıştırılıp sonuçların incelenmesi göz önüne alınırsa bu işlem oldukça uzun ve hata oranı yüksek bir yöntemdir. Bundan dolayı yeni yaklaşımlara ihtiyaç duyulmaktadır. Probleme özgü kritik parametre değerlerinin bulunduğu ve çalışma esnasında güncellendiği yöntem optimum çözüme yaklaşma hızını arttırmakta ve yerel en iyi değerlere takılma ihtimalini azaltmaktadır. Algoritmanın probleme uyarlanabilir olması sadece ETMB'ler için değil farklı alanlardaki en iyileme problemlere de uygulanabilmesi açısından önem arz etmektedir.

Kritik parametre değerlerinin çalışma esnasında probleme uyarlanması algoritmanın performansını arttırabileceği gibi optimuma yakın model parametrelerinin bulunmasını hızlandırmaktadır. Bu bölümde geliştirilen çevrimiçi probleme uyarlanabilir MSA'lar ve kullanılan yöntem anlatılmaktadır.

### 6.1. Kritik Parametrelerin Uyarlanabilir Parametre Ayarı

Matasezgisel algoritmalar çeşitli kritik parametrelere sahiptir ve algoritmadan iyi performans elde edebilmek için bu parametrelere uygun değerler verilmelidir.

Bu deęerlerin probleme baęlı olarak deęiřtięi bu tez alıřmasında gzlemlenmiřtir. Her problem iin uygun parametre deęerinin manuel ayarlanması en iyi kritik parametre deęerinin bulunması neredeyse imknsızdır. Bundan dolayı probleme uygun deęerleri bulan parametre ayarlama teknikleri geliřtirilmiřtir (Eiben et al., 1999), 3. Blmde bu tekniklere yer verilmiřtir. En iyi yntemin MSA'lar olduęundan bahsedilmiřti. Kendinden uyarlanabilir parametre ayarlama yntemi iin MSA'lar kullanılmaktadır. Bylece kritik parametre deęerleri de problem parametrelerinin bir parası olmakta ve ayarlanma srecine dhil edilmektedir. Fakat bu tez alıřmasında kendinden uyarlanır parametre ayarlama yntemi tercih edilmemiřtir. nk karmařık sistemlerin karmařık parametre uzayını ayarlama iřlemi yaparken bu parametre deęerlerine eklenecek kritik parametre deęerleri benzetim sresini uzatacaęı gibi benzetim ve algoritma arasında dnen kod karmařıklıęını arttıracaktır. Ayrıca algoritmanın bulduęu zm kmeleri dıřarıda parametreler.java sınıfına evrilerek iřletilmek zere benzetime verilmektedir. Kritik parametre deęerlerini ayırmak ek iřlem ve zaman kaybına neden olmaktadır. Bundan dolayı kendinde uyarlanabilir parametre ayarlama yntemi kullanılmamıřtır.

Bu tez alıřmasında kritik parametre deęerleri ayarlama iřlemi iin 4. Blmde detaylı aıklanan evrimii uyarlanabilir parametre ayarlama yntemi geliřtirilmiřtir yinelemeli F-Race algoritmasından (Eiben et al., 1999) esinlenilerek geliřtirilmiř. Yinelemeli F-Race Algoritması, her tr veri seti zerinde iřlem yapan ve kullanıldıęı algoritmadan aldıęı geri bildirimler doęrultusunda kullanıldıęı MSA iin uygun parametre deęeri bulan bir algoritmadır. Yinelemeli F-Race Algoritması, Eiben ve arkadařları tarafından yapılan alıřmalarda yeterlilięi kanıtlanması ynnden nemlidir. Geliřtirilen F-Race'den esinlenen evrimii uyarlanabilir parametre ayarlama (UPA) algoritması bulacaęı kritik parametre deęerlerini algoritmanın alıřması esnasında belirlemektedir. Kritik parametre deęerleri algoritmadan gelen geri bildirimlerle aranır gerekirse parametre deęer aralıklarında deęiřim yapılabilir. Arama uzayını geniřletip daraltma zellięine sahiptir. Bylece algoritma probleme kendini uyarlar ve algoritma ayarlama iřlemi programa verilerek zaman kazancı saęlanmış olur. MSA'nın bazı problemlerde gstermiř olduęu bařarıyı dięer tm problemlerde de gsterebilmesi iin probleme uyarlanabilir olması gerekmektedir.

Bu tez alıřmasında UPA kullanılarak 4 farklı uyarlanabilir MSA geliřtirilmiřtir. Geliřtirilen Uyarlanabilir MSA'lar;

- Uyarlanabilir Genetik Algoritma (Adaptive Genetic Algorithm (AGA)),
- Uyarlanabilir Parçacık Sürü Optimizasyonu (Adaptive Particle Swarm Optimization (APSO)),
- Uyarlanabilir Yapay Arı Kolonisi (Adaptive Artificial Bee Colony (AABC)),
- Uyarlanabilir Ateş Böceği (Adaptive Firefly Algorithm (AFA)) Algoritmasıdır.

Kısaca çevrimiçi uyarlanabilir parametre ayarlama algoritması, problem örnekleri için rasgele belirlenmiş aday parametre değerleri veya belirlenen aralıklar ya da seçenekleri dikkate alarak rasgele çözüm kümesi oluşturarak çalışmaya başlar ve F-Race algoritması gibi istatistik testlerine göre en kötü olan değerleri elimine eder. Temel amacı algoritmalarındaki kritik parametre değerlerini problemin ihtiyacı doğrultusunda iyileştirmektir. Bu tez çalışmasında kullanılan GA, PSO, ABC ve FA algoritmalarının ayarlanması gereken kritik parametre değerleri Çizelge 6.1’ de verilmiştir.

Çizelge 6.1 Algoritmalara Ait Kritik Parametre Değişkenleri

GA	PSO	ABC	FA
Seçim Yöntemi	C1	M	$\alpha$ (alpha)
Çaprazlama Yöntemi	C2		$\gamma$ (gama)
Ebeveyn Yüzdesi			
Mutasyon Yüzdesi			

**Seçilme Yöntemi:** İlk olarak uygunluk değeri iyi olan çözüm kümelerinin seçildiği parametre.

**Çaprazlama:** En iyi uygunluğa sahip çözüm kümelerinden daha iyi çözümler üretilmesi amacıyla kullanılan parametre.

**Ebeveyn Yüzdesi:** Popülasyon içinde kaç çözüm kümesinin seçileceğini belirleyen parametre.

**Mutasyon Yüzdesi:** Hangi sıklıkla çözüm kümelerinin mutasyona uğraması gerektiğini belirten parametre.

**C<sub>1</sub>:** Çözüm kümesinin geçmiş en iyi konumuna yaklaşma oranını belirleyen parametre.

**C<sub>2</sub>:** Geçmiş en iyi konuma yaklaşma oranını belirleyen parametre.

**M (modification rate-MR):** Çözüm kümesinde yer alan parametrelerden kaç tanesinin değiştirileceğini belirleyen parametredir.

**$\alpha$  (alpha):** Rastlantı değişkenidir,  $\alpha$  değerinin büyük olması rasgele üretilen  $\epsilon_i$  değerinin ateş böceğinin hareketine etkisini artırır.

**$\gamma$  (gama):** Sabit emilim katsayısı ateş böceğinin uzaklığının parlaklığı üzerindeki etkisini ayarlamaktadır.

## 6.2.Çevrim İçi Uyarlanabilir Parametre Ayarlama Algoritması

Kritik parametrelerin çevrimiçi değer alması için geliştirilen yöntem algoritma çalışması esnasında kritik parametre değerlerinin problem çözümüne etkisi sonucu daha iyi değerlere doğru gitmesi şeklindedir.

Kritik parametre değerleri için rasgele üretilen aday parametre değerleri dDizisi'nde tutulur dDizisi'nde yer alan değerler ile F-Race algoritması çalıştırılır ve istatistiksel testlere göre en kötü olan parametre değerleri diziden silinir. Böylece kötü çözüm kümelerinin model üzerinde işletilmesi engellenmiş olur. Kalan değerler algoritmaya gönderilir çevrim içi çalıştırılır elde edilen uygunluk değerleri karşılaştırılır. Eğer global uygunluk değeri iyileştirildiyse o döngüde kullanılan kritik parametre değerleri başarılar dizisine (bdizisi) atılır. Bu işlem dizideki elemanlar bitinceye kadar devam etmektedir. Dizideki elemanlar bittikten sonra dDizisi tekrar ilklenir sonra F-Race algoritmasına gönderilerek başlangıç dizisi oluşturulur ve bu diziyeye başarılar dizisinden belirlenen yüzde oranı (bYuzde) sayısınca değerler dDizisi'ne atanır. bYuzde değeri çalışma esnası boyunca artırılır böylece algoritma ilerledikçe daha çok bDizisi değeri kullanılır. Başarılar dizisindeki tüm değerlerin dDizisi'ne aktarılmama sebebi arama uzayını daraltmamaktır. dDizi elemanları bittiği halde başarılar dizisinde (bDizisi) eleman olmaması durumunda değer aralıkları güncellenerek yani genişletilip yada daraltılarak yeniden aynı işlemler yapılır. Anlatılan bu algoritmanın sözde kodu aşağıda verilmiştir.

---

**Algoritma 6.1 Çevrimiçi Uyarlanabilir Parametre Ayarlama Algoritması**


---

**Do**

*Dongu=denemeBoyutu/100\*bYuzde*

**For** *I=1 to dongu do* /\*deneme dizisi boyutunun başarı dizi yüzdesine  
denk gelen eleman sayısı kadar\*/

**IF** *bDizi.BosMu then* /\*başarı dizisinde eklenmeyen eleman  
kalmadıysa\*/

*Döngüden çık*

**End if**

*Başarı dizisinden rasgele bir elemanı deneme boyuntunda rasgele  
bir indekse ata*

**End for**

**For** *I=1 to (denemeBoyutu-dongu) do* //deneme dizisinde boş yer kadar

**For** *j=1 to kritikDegisken do* /\*kritik değişken sayısı kadar  
kullanılan algoritmaya bağlı  
olarak değişiklik gösterecektir\*/

*Kritik değişkene belirlenen aralıklarda rasgele değer  
belirle ve deneme dizisine at*

**End for**

**End for**

**While** *deneme dizisinin elemanları bitinceye kadar*

*Algoritmaya deneme dizisinden kritik değişken değerlerini ver  
algoritmadan iterasyon sonunda uygunluk değerini al*

**If** *iterasyon!= 1 then*

**If** *uygunluk < genelUygunluk then* /\*uygunluk değeri  
geliştirilmişse\*/

*bSayi=bSayi+1 /\*Uygunluk değerini başarılı  
dizisine ekle\*/*

**Else**

*Bir sonraki deneme boyutuna geç*

*Geliştirilmeme sayısını arttır*

**End if**

**End if**

**If** *gelistirmeSayi=limitGelistirme then* /\*Geliştirme sayısı  
limitine ulaşıldı ise\*/

*Kritik değişkenlerin değer aralıklarını genişlet*

**End if**

**End While**

**While** *iterasyon=MaxItr* //maksimum iterasyona ulaşıncaya kadar

---

### **6.3.Uyarlanabilir Genetik Algoritma (Adaptive Genetic Algorithm (AGA))**

GA canlıların genetik şifre yapısı kullanılarak sezgisel olarak en iyi çözümü veya en iyi çözüme yakın bir sonuç bulmayı hedefler. Karmaşık, çok boyutlu arama uzayında en iyinin hayatta kalması ilkesine göre küresel çözümü arar (Man et al., 2005).

GA parametreleri, biyolojideki genleri temsil ederken, parametrelerin toplu kümesi de kromozomu oluşturmaktadır. GA'ların her bir ferdi, yani her bir olası çözüm, kromozom şeklinde temsil edilir. Bu aday çözümler kümesi de popülasyon olarak adlandırılır. Popülasyonun uygunluğu, belirli kurallar dâhilinde maksimize veya minimize edilir. Her yeni nesil, rasgele bilgi değişimi ile oluşturulan diziler içinde hayatta kalanların birleştirilmesi ile elde edilmektedir.

Genetik algoritmalarda Çaprazlama ve Mutasyon olarak bilinen iki adet temel genetik işlemci vardır. Çaprazlama için popülasyondan iki adet birey seçilir. Bu bireylerde çaprazlanacak nokta belirlenir ve bu noktadan itibaren bireylerin elemanları karşılıklı olarak yer değiştirilir. Böylece iki adet yeni birey elde edilir. Mutasyon işlemcisi ile bireylerin genleri değiştirilir. Bu değişim popülasyonun genel olarak %1-%5 ini kapsamaktadır. Mutasyon popülasyonda çeşitliliğe neden olur ve problem sonucunun yerel çözümlere takılmasını önler.

Genetik çaprazlama yöntemi, ebeveyn seçim yüzdesi, mutasyon oranı ve ebeveyn seçim yöntemi genetik algoritmada belirlenmesi gereken parametrelerdir. Bu parametre değişkenleri her problemde farklı değerler alabilmektedir. Her problem için en iyi değerleri manuel belirlemek oldukça zordur. Uyarlanabilir genetik algoritma genetik algoritmaya ait bu parametreleri problemde aldığı geri bildirimlerle çevrimiçi ayarlama işlemini gerçekleştirmektedir. Probleme özgü geliştirilen AGA'nın sözde kodu aşağıda verilmiştir.

---

**Algoritma 6.2 Uyarlanabilir Genetik Algoritma**


---

```

For pop=1 to maxPop do //popülasyondaki tüm bireyler için
  for par=1 to maxPar do //Sırayla tüm parametre değerleri ata
    Bireylerin kromozomlarındaki genleri rasgele belirle //Eş. 5.1.1
  End for
  Çözüm kümelerinin uygunluk değerlerini hesapla
End For
Do
  If cupaDurum=False Then
    Caprazlama=deger1; /*tüm kritik değerler manuel olarak
                        belirlenen parameter.xml dosyasından alınır*/
    ebeveynSecim=deger2
    ebeveynSecimY=ob
    mutasyon=ob2
  Else
    cupaAlgoritması () //kritik değişkenler bulunur ve atanır
    Caprazlama=deger1 /*tüm kritik değerler manuel olarak
                      belirlenen parameter.xml dosyasından
                      alınır*/
    ebeveynSecim=deger2
    ebeveynSecimY=ob
    mutasyon=ob2
  End if
  For i=0 to ebeveynSecimY do
    Seçim yöntemiyle 2 birey seçimi yap
    2 birey arasında parça değişimi yap
    Mutasyon koşulunu uygula
    Göç işlemlerini uygula
  End for
  Sonuçları karşılaştır en iyi çözüm kümesini bul
While iterasyon=MaxItr //maksimum İterasyona ulaşıncaya kadar

```

---

#### 6.4.Uyarlanabilir PSO Algoritması (Adaptive Particle Swarm Optimization (APSO))

Sürü halinde hareket eden kuşların yiyecek kaynaklarına, çoğu zaman rasgele yaptıkları konum değişikliği ile kolaylıkla ulaşabilmeleri bu algoritmanın esin kaynağını oluşturmuştur. PSO bir grup rasgele çözüm kümesi (Parçacık sürüsü) ile başlar ve güncellemelerle optimum çözüm bulunmaya çalışılır. Her iterasyonda Parçacık konumları, hafızada saklanacak iki en iyi değere (pbest ve gbest) göre güncellenir. Parçacığın elde ettiği en iyi çözümü sağlayan koordinatlar *pbest*, popülasyonda tüm Parçacıklar için o ana kadar elde edilen en iyi çözümü sağlayan koordinatlar ise *gbest* olarak adlandırılır. Öğrenme faktörleri olan *c1* ve *c2* her parçacığı *pbest* ve *gbest* değerlerine doğru çeker. *c1* ve *c2* değerleri optimum

çözümüne ulaşmada önemli olup alabileceği değerler değişebilmektedir. Hangi değerlerin mevcut problemde daha iyi sonuca ulaşacağını bilmek imkânsızdır. Uyarlanabilir Parçacık sürü optimizasyonu algoritması bu değişkenleri çevrimiçi olarak probleme uyarlamaktadır. Problemlerden aldığı geri bildirimlerle sürekli c1 ve c2 değerleri güncellenmektedir. Algoritmaya ait bu parametreleri problemden aldığı geri bildirimlerle çevrimiçi ayarlama işlemini gerçekleştirmektedir. Uyarlanabilir Parçacık sürü optimizasyonu algoritması sözde kodu aşağıda verilmiştir.

---

### Algoritma 6.3 Uyarlanabilir Parçacık Sürü Optimizasyonu Algoritması

---

```

For i=1 to p do //P Parçacık sayısı
  For j=1 to N do //N parçacığın eleman sayısı
    Parçacıkların konumlarını ve hızlarını ata //Eşitlik 5.1.1
  End for
End for
Do
  For i=1 to p do //P Parçacık sayısı
    Uygunluk hesapla
    If uygunluk < gBest then
      geçerli değeri gBest olarak ata
    End if
    If uygunluk < pBest then
      geçerli değeri pBest olarak ata
    End if
  End for
  If cupaDurum=False then /*kritik değerler parameter.xml
    dosyasındaki manuel girilen değerler
    atanır.*/
    C1=deger1
    C2=deger2
  Else
    cupaAlgorithm () //cupa algoritmasını işlet
    C1=deger1
    C2=deger2 //cupa'dan gelen değerler çekilir.
  End if
  For i=1 to p do
    For j=1 to n do
      Eylemsizlik ağırlığının hesaplanması (w) (Eşitlik 5.2.3)
      Hız hesapla (Eşitlik 5.2.1)
      Konum güncelle (Eşitlik 5.2.2)
      Vmax sınırını kontrol et
    End for
  End for
While iterasyon=MaxItr //maksimum iterasyona ulaşincaya kadar

```

---

### 6.5. Uyarlanabilir ABC Algoritması (Adaptive Artificial Bee Colony (AABC))

Arıların besin arama davranışlarından esinlenilerek geliştirilmiştir (yazılmıştır). İşçi, gözlemci ve kâşif arıların besin arama ve işleme davranışları algoritmaya uyarlanmıştır (Karaboğa and Basturk, 2007). ABC algoritmasında, işçi (*employed*), gözlemci (*onlooker*) ve kâşif (*scout*) arı olmak üzere üç tip arı bulunmaktadır. İşçi arılar her bir yiyecek kaynağının nektar miktarını hesaplamak ile sorumludur ve popülasyondaki işçi arı sayısı beslenme alanında bulunan yiyecek kaynağı sayısına eşittir. Gözlemci arılar iyi nektar miktarına sahip yiyecek kaynağını seçmekle sorumludur ve popülasyondaki gözlemci arı sayısı işçi arı sayısına eşittir. Kâşif arılar ise yeni yiyecek kaynaklarını keşfetmek ile sorumludur. Yiyecek kaynağı tükenmiş olan işçi arılar kâşif arılara dönüşmektedir ve eski yiyecek kaynağı, kâşif arı tarafından bulunan yeni yiyecek kaynağı ile değiştirilmektedir. Yiyecek kaynağının pozisyonu, çözülmesi hedeflenen optimizasyon probleminin olası çözümünü temsil etmektedir. Yiyecek kaynağının nektar miktarının yüksek olması optimizasyon probleminin olası çözümünün iyi olması anlamına gelmektedir. Dolayısı ile olası çözümün kalitesi nektar miktarı ile temsil edilmektedir ve bu değer ABC algoritmasında uygunluk değeri (*fitness value*) olarak adlandırılmaktadır.

ABC algoritmasının çözüme yaklaşma hızını artırma için bu çalışmada kullanılan ve komşu çözümün üretilmesinde tek bir parametre güncellenmesi yerine kaç tane parametrenin değiştirileceğini belirleyen MR (*modification rate* - MR) parametresi kullanılmıştır. Ayrıca bu parametre değeri problemden aldığı geri bildirimler sayesinde çevrimiçi güncellenmektedir. Böylece algoritma probleme uyarlanabilmektedir. Geliştirilen uyarlanabilir ABC algoritmasının sözde kodu aşağıdaki gibidir.

---

**Algoritma 6.4 Uyarlanabilir Yapay Arı Kolonisi Algoritması**


---

```

For  $i=1$  to SN do //tüm besin kaynakları için
    Besin kaynaklarını yükle //Eşitlik 5.3.1
    Uygunluk hesapla //Eşitlik 5.3.2
End for
Do
    If cupaDurum=false then
        M=deger1 /*kritik değerler parameter.xml dosyasındaki
            manuel girilen değerler atanır.*
    Else
        cupaAlgorithm ()
        M= fDeger1 //cupa algoritmasının ürettiği değeri ata
    End if
    For  $i=1$  to SN do //tüm işçi arılar için
        Besin kaynağını işle yeni uygunluk değeri bul //Eşitlik 5.3.5
        Parametre aralıklarını kontrol et //Eşitlik 5.3.6
        If yeni uygunluk daha iyiyse
            Çözüm kümesini değiştir
        End if
    For  $i=1$  to SN do // tüm Parçacıkları için
        Olasılıkları belirle //Eşitlik 5.3.7
    End for

    For  $i=1$  to SN do // tüm gözcü arılar için
        If rasgele oluşturulan sayı olasılıktan büyük ise
            Besin kaynağını işle yeni uygunluk değeri bul //Eş. 5.3.5
            Parametre aralıklarını kontrol et //Eşitlik 5.3.6
            If yeni uygunluk daha iyiyse
                Çözüm kümesini değiştir
            End if
        End if
    End for
    For  $i=1$  to SN do // tüm işçi arılar için:
        If limit sayısına ulaşıtıysa (çözüm kümesi belirlenen limit
        sayısınca
            iyileştirilemedi ise)
                Kâşif arıya çevir
            End if
    End for
While (iterasyon=maxIterasyon) // iterasyon sayısına ulaşınca kadar

```

---

## 6.6. Uyarlanabilir Ateş Böceği Algoritması (Adaptive Firefly Algorithm (AFA))

Ateş böceği algoritması, Sürü tabanlı sezgisel optimizasyon algoritmalarından biridir. Ateş böceği algoritması, ateş böceklerinin parlaklığa duyarlı davranışları üzerine kurgulanmış bir algoritmadır. Algoritmada önemli kriterler; tüm ateş böceklerinin cinsiyet göz ardı edilerek birbirlerini etkileyebilmesi, cazibeleri (*attractiveness*) parlaklıkları ile orantılıdır. Az parlak olan ateş böceği daha parlak olan ateş böceğine yaklaşır. Uzaklığın artması parlaklığın etkisini azaltmaktadır. Bu nedenle kendinden daha parlak ateş böceği bulamayan ateş böceği rasgele hareket ederek daha parlak ateş böceği bulma ihtimalini artırır. Ateş böceğinin optimum çözüme ulaşmada önemli etkisi olan Işık soğurma katsayısı  $\gamma$  ve yer değişimi için adım büyüklüğünü temsil eden  $\alpha$  değerleri her problem için farklı değerler alan parametrelerdir. Bu çalışmada FA algoritması için bu değerler çevrimiçi uyarlanabilir kontrol mekanizması kullanılarak probleme uyarlanmış böylece optimuma daha yakın sonuçlar elde eden uyarlanabilir ateş böceği algoritması geliştirilmiştir. Bu algoritmanın sözde kodu aşağıda görülmektedir.

---

**Algoritma 6.5 Uyarlanabilir Ateş Böceği Algoritması**


---

```

For  $i=1$  to  $n$  do                                //tüm ateş böcekleri için. Başlangıç
                                                    //Popülasyonu oluşturuluyor
    Ateş böceklerinin rasgele yükle //Eş. 5.4.1
End for
Do
    If  $cupaDurum=False$  then                        /*kritik değerler parameter.xml
                                                    dosyasındaki manuel girilen değerler
                                                    atanır.*/
         $Y = deger1$ 
         $\alpha = deger2$ 
    Else
         $cupaAlgorithm()$  //cupa algoritmasını işlet
         $Y = deger1$ 
         $\alpha = deger2$  //cupa'dan gelen değerler çekilir.
    End if
    For  $i = 1$  to  $n$  do                            //tüm ateş böcekleri için
        For  $j = 1$  to  $n$  do                        //tüm ateş böcekleri için
            If  $I_i < I_j$  then //j ateş böceği daha parlak ise
                 $j$  ateş böceğinin  $i$  ateş böceğine uzaklığını bul
                (Eş.-5.4.3)
                 $j$ 'nin çekiciliğini bul //Eşitlik5.4.5
                 $i$  ateş böceğini  $j$  ateş böceğine doğru hareket ettir
                //Eş. 5.4.6
            Else
                 $i$  ateş böceğini rasgele hareket ettir //Eş. 5.4.7
            End if
        End for  $j$ 
    End for  $i$ 
    Sonuçları karşılaştır en iyi çözüm kümesini bul
While ( $t < maxIteration$ )                          //Maksimum iterasyon sayısına ulaşıncaya
    kadar

```

---

**Not:** Geliştirilen AGA, APSO, AABC ve AFA algoritmalarına ait Akış diyagramları EK 1'de verilmiştir.

## **7. GELİŞTİRİLEN PARAMETRE AYARLAMA ARACININ KULLANIMI**

Önerilen parametre ayarlama yönteminde, çok sayıda MSA (GA, PSO, ABC, FA) kullanılmıştır. Ayrıca kullanılan bu MSA'lar iyileştirilerek probleme uyarlanan MSA'lar (AGA, APSO, AABC, AFA) geliştirilmiştir. Kullanılan MSA ve geliştirilen uyarlanabilir MSA algoritmaları bu araçta bir araya getirilmiştir. Böylece bu araç, ETMB ortamında yer alan çok sayıda modele ait parametre ayarlama problemine farklı yöntemleri kullanma imkânı sunmaktadır. Araçta birçok parametre ayarlama yönteminin bir arada olması model hedeflerine ulaşacak uygun yöntemin daha kolay bulunmasına olanak tanımaktadır.

Aracın model beklentilerini karşılamaında önemli olan diğer bir kriter parametre ayarlama işlemi için, benzetim hedeflerinin tam ve doğru belirlenmesi gerekliliğidir. Bir ETMB çalışmasına başlanırken benzetim hedeflerinin iyi tanımlanması gerekmektedir. ETMB çalışmasının ilk aşaması olan hedeflerin belirlenmesi, parametre ayarlama süreci için de belirleyici aşama haline gelmektedir. Bu tez kapsamında geliştirilen araç ile parametre ayarlama işlemi yapılırken benzetim hedefleri temel alınır. Parametre ayarlama ile benzetim hedefleri arasında doğrudan birebir ilişki vardır ve bunu tanımlamak kritik öneme sahiptir. Geliştirilen parametre ayarlama aracının kullanım aşaması Çizelge 7.1'de verilmiştir.

### **7.1. Parametre Ayarlama Aracının Gereksinimleri**

Geliştirilen parametre ayarlama araç gereksinimleri Çizelge 7.1'de verilmiş olup geliştirilen aracın kullanılacak modele uyarlanmasını yönlendiren aşama ve gereksinimleri içermektedir.

Çizelge 7.1 Parametre Ayarlama Aracının Gereksinimleri

Aşama	Gereksinimler
Modelin Tanımlanması	<ul style="list-style-type: none"> <li>• Model ile aracın ilişkilendirilmesi</li> </ul>
Yöntem Arayüzünün Eklenmesi	<ul style="list-style-type: none"> <li>• Kullanılacak yöntemin belirlenmesi</li> <li>• Modele ait süreç verilerinin belirlenmesi</li> <li>• Parametre değerlerinin ilklendirilmesi</li> </ul>
Parametrelerin Tanımlanması	<ul style="list-style-type: none"> <li>• Model verisinin hazırlanması</li> <li>• Model parametrelerinin bileşenlerinin tanımlanması (başlangıç, bitiş, aralık işletim zamanı; olayla ilişkilendirme)</li> <li>• Benzetim ve ortam parametre değerleri/değer aralıklarının tanımlanması</li> <li>• Parametre ayarlama sonucu, beklenen çıktılarının tanımlanması</li> </ul>
Model Hedeflerinin Belirlenmesi	<ul style="list-style-type: none"> <li>• Beklenen davranışa uygun kriterlerin belirlenmesi</li> <li>• Her kriter için gerekli ve model hedefine hizmet edecek uygunluk fonksiyonlarının belirlenmesi</li> </ul>
MSA Algoritmasına Ait Kritik Parametrelerin Tanımlanması	<ul style="list-style-type: none"> <li>• Kullanılan parametre ayarlama algoritmasına ait parametrelerin tanımlanması (başlangıç, bitiş, aralık işletim zamanı; olayla ilişkilendirme)</li> <li>• Kritik parametre değerleri/değer aralıklarının tanımlanması</li> </ul>
Parametre Ayarlama Süreci	<ul style="list-style-type: none"> <li>• Belirlenen parametre kümelerinin benzetimde işletilmesi</li> <li>• Elde edilen sonuçların veri dosyalarında tutulması</li> </ul>
Elde Edilen Sonuçlar	<ul style="list-style-type: none"> <li>• Elde edilen parametre değerlerinin test edilmesi.</li> <li>• Bulunan kritik parametre değerlerinin test edilmesi.</li> <li>• Test sonuçlarının değerlendirilmesi ile parametre tekniğinin ilişkilendirilmesi</li> </ul>

**Modelin Tanımlanması:** Parametre ayarlama, uygulanacağı modelin işletilmesi esnasında gerçekleştirilmektedir. Bundan dolayı model, geliştirilen araca tanıtılmalı ve gerekli bağlantıların kurulması sağlanmalıdır.

**Yöntem Arayüzünün Eklenmesi:** Şekil 7.2’de yer alan arayüzün modele eklenmesi işlemini içermektedir. Şekilde de görüldüğü gibi arayüz, parametreleri ayarlanacak modelin parametre ayarlama sürecinde hangi yöntemin kullanılacağı, benzetim süresi, iterasyon sayısı ve çözüm kümesi (birey sayısı) gibi değerlerin seçilmesini sağlamaktadır.

**Parametrelerin Tanımlanması:** Modele ait sabit olmayan model sonuçlarını doğrudan etkileyen model hedefine ulaşmada kritik öneme sahip olan model parametrelerinin ve bileşenlerinin belirlenmesi, parametrelere ait uygun parametre değer aralıklarının belirlenmesi ve elde edilecek çıktı sonuçlarının belirlenmesi gibi işlemlerin yapıldığı aşamadır.

**Model Hedeflerinin Belirlenmesi:** Her model hedefi, gerçek sistemi temsil edecek şekilde tasarlanır. Bu nedenle model hedefleri, model varsayımları, etmen davranışları, etmen etkileşimleri göz önünde bulundurularak belirlenmelidir. Bu hedefler farklı koşullar dikkate alınarak uygunluk fonksiyonları olarak tasarlanır. Hangi koşullar altında hangi uygunluk fonksiyonunun dikkate alınacağı parametre ayarlama aşamasında büyük öneme sahiptir. Modelin anlık etmen davranışları ve etmen parametre değerleri uygunluk fonksiyonunun seçimini belirlemede etkindir. ETMB’lerde farklı durum ve koşullarda farklı yapıların ortaya çıktığı bilinmektedir bu özelliğin dikkate alınarak parametre ayarlama işleminin yapılması gerekmekte olup bu doğrultuda benzetim hedeflerinin doğru belirlenmesi önemli hale gelmektedir.

**MSA Algoritmasına Ait Kritik Parametrelerin Tanımlanması:** Kullanılan MSA algoritmalarını ETMB modellerine uyarlanabilir hale getirmek amacı ile Uyarlanabilir MSA algoritmaları geliştirilmiştir. Uyarlanabilir MSA algoritmaları geliştirilirken bu tez kapsamında ikinci seviye olarak adlandırılan kritik parametre ayarlama işlemi yapılmıştır. İlk olarak kullanılan MSA algoritmasına ait kritik parametre değerleri belirlenmektedir. Belirlenen bu parametrelerin parametre değer aralıkları ve kritik parametre ayarlama işleminde kullanılan çevrimiçi uyarlanabilir parametre ayarlama algoritmasının bir özelliği olan aşamalı aralık daraltma, aralık genişletme değerleri bu aşamada belirlenmektedir.

**Parametre Ayarlama Süreci:** Parametre ayarlama için uygun hedefler belirlendikten sonra, ilgili hedefe ulaşılmasını sağlayacak olan parametre değerlerinin elde edilmesi için parametre ayarlama süreci başlatılır. Hedefe ulaşma

doğrultusunda üretilen her bir parametre kümesi kullanılarak model işletilir. Model işletimi sonlandığında elde edilen sonuçlar sonuç dosyalarında saklanır. Hedefe en yakın parametre seti (Çözüm kümesi) en iyi parametre seti olarak belirlenir.

**Elde Edilen Sonuçlar:** Parametre ayarlama işlemi sonucunda elde edilen sonuçlar incelenir bulunan en iyi parametre değerlerinin model hedeflerine ulaşım ulaşmadığı bu aşamada test edilmektedir. Parametre ayarlama süreci sonunda elde edilen en iyi model parametre değerleri modele manuel verilerek bu aşamada model hedeflerine ulaşım ulaşmadığı gözlemlenir.

## 7.2.Geliştirilen Parametre Ayarlama Aracının Mimarisi

Geliştirilen Parametre Ayarlama'ya ait sınıf diyagramları Ek 3 ve Ek 4'deki şekillerde verilmiştir. Bu şekillerde verilen sınıflar, gerçekleştirim açısından önemli olan sınıfları içermektedir. Parametre ayarlama işlemi ETMB ortamında geliştirilen model parametrelerine yönelik yapılmaktadır ve RePast aracı kullanılmıştır. Ek 3 ve Ek 4 de yer alan şekillerde sadece geliştirilen araca ait sınıflar bulunmaktadır. Parametresi ayarlanacak model paketi içine, araç ve model arasında köprü görevini gören *Initializer* sınıfı eklenmiştir. Ek 3'de yer alan Ortak paket içinde Excel, Sayac, İşlemler, Fonksiyonlar, Parametreler ve Benzetim sınıfları bulunmaktadır. Ek 4'de yer alan Algoritma Paketi içinde Çatı, GA, PSO, ABC, FA ve CUPA sınıfları bulunmaktadır.

### 7.2.1. Ortak Paket Sınıfları

Geliştirilen araca ait Ek 3'de yer alan Ortak Paket altında geliştirilen sınıflar bu bölümde açıklanmıştır.

**Excel Sınıfı:** Excel dosya işlemlerinin toplandığı sınıftır. Belirlenen kriterler doğrultusunda excel dosyası oluşturulur. Modele ait benzetim öncesi ve sonrası parametre değerleri ve uygunluk değeri gibi belirlenen değerlerin Excel dosyasına yazılmasını sağlayan işlemleri içerir.

**Sayaç Sınıfı:** Benzetim işletilirken başlama ve bitiş adım sayılarının tutulduğu ve saniye türünden benzetimin toplam işletildiği sürenin hesaplandığı sınıftır.

**İşlemler Sınıfı:** Benzetime ait özel işlemlerin yapıldığı sınıf olup parametrelerin değer aldığı, benzetimdeki her adım sayısında parametrelerin bileşenlerin kontrol edildiği, uygunluk fonksiyonunun oluşturulduğu parametreler sınıfına bağlanarak parametre değerlerinin alınıp benzetime eklendiği işlemler bu sınıfta gerçekleştirilmektedir. Ayrıca benzetim alanının yüklenmesi ve her bir çözüm kümesinin benzetim işletimi bittikten sonra yeni çözüm kümesi ile yeni benzetimin yüklenmesi işlemi yine bu sınıf içinde gerçekleştirilir. Modelin geliştirilen parametre ayarlama aracına entegre edilmesi için ayrıca Initializer sınıfı yaratılmıştır. Bu sınıf modelle geliştirilen araç arasındaki bağlantıların yapıldığı sınıftır.

**Fonksiyonlar Sınıfı:** tez için gerekli olan matematiksel işlemlerin bulunduğu sınıftır.

**Simülasyon Sınıfı:** MSA'larla benzetim arasında tampon görevi gören sınıftır. Algoritmadan aldığı verileri benzetime, benzetimden aldığı verileri algoritmalara servis eder. Çözüm kümesinin benzetimi gerçekleştirilirken belirlenen adım sayısına ulaşmış ve ulaşmadığı bu sınıfta kontrol edilir. Hangi algoritmanın seçileceğine bu sınıfta karar verilir. Çözüm kümesinin benzetimi bitince bulunan uygunluk değeri bu sınıf üzerinden kullanılacak MSA algoritmasına değer olarak gönderilir. Kullanılan algoritmadan bir sonraki çözüm kümesi alınarak benzetimin parametre değerleri atanır. Çözüm kümesinin daha önce uygunluk değerinin bulunup bulunmadığı kümeler dizisinden kontrol edilir. Uygunluk değeri daha önceden bulunmuş çözüm kümelerinin verileri işlem sınıfına gitmeden excel dosyasına yazılma işlemi burada gerçekleştirilir. Ayrıca benzetim beklenen süreden önce sonlanırsa ya da kriter sağlanırsa sayacı durdurma işlemi bu sınıfta gerçekleştirilir.

**Parametreler Sınıfı:** yeni parametre ekleme ya da gönderilen parametreye değer üretme işleminin yapıldığı sınıftır. Parametreye ait değer aralıkları bu sınıfta belirlenir. Bu değer aralıklarına ve değişken tiplerine bağlı olarak parametrelere değer üretildiği sınıftır.

### 7.2.2. Algoritma Paketi Sınıfları

Geliştirilen araca ait Ek 4'de yer alan Algoritma Paketi adı altında geliştirilen sınıflar bu bölümde açıklanmıştır.

**Çatı Sınıfı:** GA, PSO, ABC ve FA algoritmalarının ortak bulunduğu değişkenleri ve fonksiyonları barındıran sınıftır; yani sezgisel optimizasyon algoritmalarının temel ortak özellikleri çatı sınıfında yer alır. Parametre boyutu, popülasyon boyutu, maksimum iterasyon değişkenlerinin değerleri bu sınıf içinde atanır. Parametre yükleme işlemi, çözüm kümesinin uygunluk değeri geldiğinde çözüm kümesine uygunluk değerinin eklenmesi ve çözüm kümesinin uygunluk değeri bekleyen çözüm kümeleri listesinden çıkarılması, parametre değerlerinin limit kontrolü, genel en iyi uygunluk değeri ile yerel en iyi uygunluk değeri kontrollerinin yapılarak gerekli güncellemelerin yapılması, CUPA durumunun belirlendiği ve mevcut çözüm kümesi ile genel ve yerel en iyi uygunluk değerlerinin ekrana yazdırıldığı sınıftır.

**GA Sınıfı:** Uygunluk değerleri hesaplanan çözüm kümeleri GA algoritmasına gönderilir. Bitirme kriteri sağlanmadığı takdirde CUPA durumunun true olduğu durumda GA'nın sahip olduğu kritik parametre değerleri ayarlanmak üzere çevrimiçi uyarlanabilir parametre ayarlama algoritmasına gönderilir, ayarlanan kritik parametre değerleri ile mevcut çözüm kümeleri (popülasyon) işletilerek GA evrim teknikleri ile en iyi çözüm kümesi (parametre seti) bulunmaya çalışılır. GA algoritmasına ait ebeveyn seçim işlemi, çaprazlama işlemi, mutasyon ve göç gibi işlemlerin gerçekleştirdiği sınıftır.

**PSO Sınıfı:** Uygunluk değerleri hesaplanan çözüm kümeleri yani popülasyon PSO algoritmasına gönderilir. Bitirme kriteri sağlanmadığı takdirde CUPA durumunun true olduğu durumda PSO'nun sahip olduğu kritik parametre değerleri ayarlanmak üzere çevrimiçi uyarlanabilir parametre ayarlama algoritmasına gönderilir, ayarlanan kritik parametre değerleri ile mevcut popülasyon işletilerek evrim teknikleri ile en iyi çözüm kümesi (parametre seti) değeri bulunmaya çalışılır. Çözüm kümelerine ait rasgele hız değerleri ve konum güncelleme işlemleri bu sınıfta gerçekleştirilir.

**ABC Sınıfı:** Uygunluk değerleri hesaplanan çözüm kümeleri yani popülasyon ABC algoritmasına gönderilir. Bitirme kriteri sağlanmadığı takdirde CUPA durumunun true olduğu durumda ABC'nin sahip olduğu kritik parametre değerleri ayarlanmak üzere çevrimiçi uyarlanabilir parametre ayarlama algoritmasına gönderilir, ayarlanan kritik parametre değerleri ile mevcut popülasyon işletilerek evrim teknikleri ile çözüm kümesi (parametre seti) bulunmaya çalışılır. Yiyecek kaynaklarının (Çözüm kümelerinin) iyileştirilmesi ve

güncelleme işlemleri, işçi, gözcü ve kâşif arıların yapacağı işler bu sınıf içinde gerçekleştirilir.

**FA Sınıfı:** Uygunluk değerleri hesaplanan çözüm kümeleri yani popülasyon FA algoritmasına gönderilir. Bitirme kriteri sağlanmadığı takdirde CUPA durumunun true olduğu durumda FA'nın sahip olduğu kritik parametre değerleri ayarlanmak üzere çevrimiçi uyarlanabilir parametre ayarlama algoritmasına gönderilir, ayarlanan kritik parametre değerleri ile mevcut popülasyon (çözüm kümeleri) işletilerek algoritmaya ait teknikler kullanılarak en iyi çözüm kümesi (parametre seti) bulunmaya çalışılır. Ateş böceklerinin sahip olduğu uygunluk değerlerinin (parlaklığı) mevcut en iyi uygunluk değerinden daha iyi olup olmadığı kontrol edilir. Ateş böceklerinin hareket işlemleri, hareket sonrası uygunluk değerinde iyileşme olup olmadığının kontrolü, iterasyondaki en iyi ateş böceğinin rasgele hareketin gerçekleşmesi, ateş böcekleri arasındaki çekicilik kurallarının uygulanması, ateş böcekleri arasındaki mesafelerin ölçülmesi, gibi işlemler bu sınıfta gerçekleştirilir.

**CUPA Sınıfı:** Algoritmalara ait kritik parametre değerlerinin ayarlandığı sınıftır. Her bir iterasyona ait en iyi uygunluk değeri alınır ve sonraki adımda bulunan uygunluk değerleri ile karşılaştırılarak kritik parametre değerleri iyileştirilir. Eğer mevcut kritik parametre değerlerinde iyileşme görülmedi ise aralık genişletme ve daraltma işlemleri bu sınıf içinde gerçekleştirilir.

Bu tez kapsamında kullanılan paketler ve paketlere ait sınıflar yukarıda açıklanmıştır. Ek 2'de mevcut çalışmanın akış diyagramı bulunmaktadır. Ek 1'de Ek 2'de yer alan ve geliştirilen AGA, APSO, AABC, AFA ve Çevrim İçi Uyarlanabilir Parametre Ayarlama algoritmalarına ait akış diyagramları daha detaylı verilmektedir. Bu akış diyagramları çalışmanın işleyişi hakkında bilgi vermektedir. Ayrıca Ek 2'de yer alan akış diyagramı model ve kullanılan algoritmalar arasındaki akışı göstermektedir.

### 7.3. Geliştirilen Aracın Arayüz Detayları

Bu tez çalışmasına ait geliştirilen aracın arayüzü Şekil 7.1 ve Şekil 7.2'de verilmiştir. Şekil 7.2'de yer alan arayüz parametreleri Şekil 7.1'in sol bölümünde yer alan parametre alanı görsel olarak daha büyük verilmiş olup ayrıntılar daha net görülmektedir. Arayüz incelendiğinde kullanıcılar her bir model için kullanılacak MSA'yı (GA, PSO, ABC, FA) A-1 bölümünde seçebilmekte ve seçilen algoritmaya

ait kritik parametre değerlerini B, C, D, E bölümlerinde manuel olarak girebilmektedir. Ayrıca eğer KPA kutucuğu seçili ise algoritmaların kritik parametre değerleri benzetimin çalışması esnasında benzetimden alınan geri bildirimlerle yani bulunan çözüm kümelerinin kalitelerine bağlı olarak kritik parametre değerleri güncellenmektedir. Böylece sabit kritik parametreye sahip MSA'ların parametre ayarlamadaki başarımı, geliştirilen Uyarlanabilir MSA'lar ile karşılaştırma imkânı aynı arayüz üzerinde verilmektedir. Şekil 7.1'de yer alan arayüzde eğer KPA kutucuğu seçilmezse F bölümünde yer alan model parametreleri dışındaki B, C, D ve E bölümünde yer alan algoritmaya ait parametreler manuel girilir ve algoritmalar bu parametre değerleri ile model parametre ayarlama işlemi yapılır. Ayarlama sonucu elde edilen parametre değerlerini içeren çözüm kümeleri (parametre setleri) ve bulunan en iyi parametre değerlerini içeren çözüm kümesi (parametre seti) Excel dosyalarında tutulur ayrıca Şekil 8.4 ve Şekil 8.5'de de görüldüğü gibi ekrana da yazdırılır. Manuel Parametreler kutucuğu işaretlenir ise ayarlama sonucu elde edilen en iyi çözüm kümesindeki (modele ait bulunan en iyi parametre seti) her bir eleman (parametre) model parametrelerinin bulunduğu F bölümünde ilgili alana girilir. Model bu parametre değerleri ile A2- Simülasyon Süresi (Adım sayısı) bölümünde belirlenen adım sayısı kadar çalıştırılır. Manuel Parametreler kutucuğu seçili iken parametre ayarlama sürecine girmez sadece bulunan model parametrelerinin, beklentiyi karşılayıp karşılamadığı yani modelin hedeflerine ulaşıp ulaşmadığı gözlemlenir.

Şekil 7.1 ve Şekil 7.2'de yer alan arayüz, geliştirilen aracın arayüz detaylarını vermektedir. Şekil 7.2'de detayları verilen arayüzde yer alan parametrelere ilişkin açıklamalar aşağıda verilmiştir:

**A.1-Algoritma Seçiniz:** Parametre ayarlama işleminde kullanılacak algoritmanın seçildiği alan.

**A.2-Simülasyon Süresi:** Benzetimin (simülasyonun) çalıştırılacağı maksimum adım sayısını verir.

**A.3-CUPA Algoritması:** Bu alan seçili olduğunda kullanılan algoritmanın sahip olduğu kritik parametre değerleri geliştirilen çevrimiçi uyarlanabilir parametre ayarlama yöntemi F-Race ile birlikte kullanılarak ayarlanır. Seçili değilse algoritmalara ait parametre değerlerine B, C, D ve E bölümünde manuel olarak girilen kritik parametre değerleri atanır.

**A.4-Birey (Çözüm Kümesi) Sayısı:** Popülasyon tabanlı kullanılan algoritmanın popülasyon boyutu belirlenir yani birey (çözüm kümesi) sayısının belirlendiği alandır.

**A.5-Maksimum İterasyon:** Uygun çözümün bulunması için algoritmanın toplam kaç iterasyon çalıştırılacağını belirlendiği alanı oluşturmaktadır.

**A.6-Manuel Parametreler:** Benzetim F bölümünde yer alan modele ait parametrelerin manuel girilmesi durumunda manuel parametreler kutucuğu seçilir.

**B.1- Genetik Ebeveyn Seçimi:** GA'ya ait kritik parametre değişkeni olup Genetik Ebeveyn Seçim yönteminin belirlendiği bölümdür. KPA kutucuğu seçili olmadığı durumda GA algoritması B bölümünde yer alan parametre değerleri ile çalışır.

**B.2-Genetik Çaprazlama:** GA'nın Genetik Çaprazlama işlemi için kullanılacak yöntemin seçildiği alanı temsil eder.

**B.3-Genetik Mutasyon Yüzdesi:** Mutasyon yüzdesinin belirlendiği bölümdür.

**B.4-Genetik Ebeveyn Seçim Yüzdesi:** GA'nın Genetik Ebeveyn Seçim Yüzdesi parametresinin alacağı değer girilir.

**C.1-PSO-C1:** PSO'ya ait kritik parametre değişkeni olan C1, Çözüm kümesinin geçmiş en iyi konumuna yaklaşma oranını belirleyen değişkene ait değer girildiği bölümdür.

**C.2-PSO-C2:** PSO'ya ait kritik parametre değişkeni olan C2, geçmiş en iyi konuma yaklaşma oranını belirleyen değişken değerinin girildiği bölümdür.

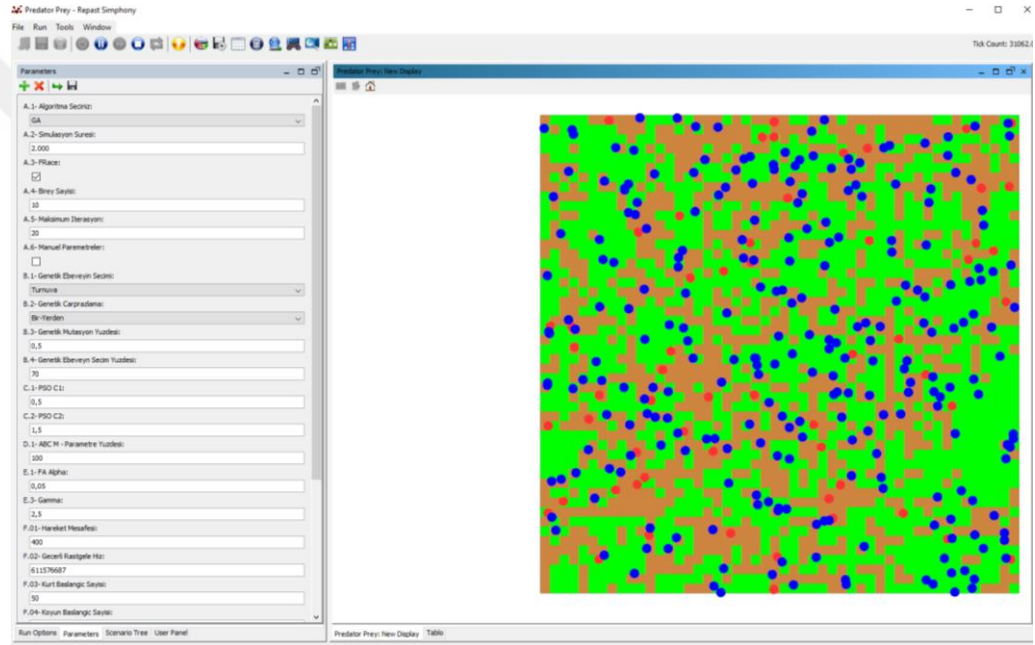
**D.1-ABC-M: M:** modification rate → Komşu çözümün üretilmesinde tek bir parametre güncellenmesi yerine kaç tane parametrenin değiştirileceğini belirleyen değişiklik oranının belirlendiği bölümdür.

**E.1-FA-Alpha:  $\alpha$  (alpha):** Rastlantı değişkeni olup ateş böceği hareketlerini etkileyen bu değer girildiği alanı temsil eder.

**E.2-FA-Gamma:  $\gamma$  (gama):** Sabit emilim katsayısı olup FA'ya ait parametredir.

**F. Bölümü:** Kullanılan modele ait parametreleri içermektedir. Bu bölüm farklı modellerde değişkenlik göstermektedir.

Bölüm B, C, D, E çevrimiçi uyarlanabilir parametre ayarlama algoritması ile ayarlanan ve bulunan kritik parametre değerlerinin girilip başarımının test edilmesinde de kullanılmaktadır. Şekil 7.2'de yer alan F bölümü kullanılan modele ait parametre alanını içermektedir.



Şekil 7.1 Parametre Ayarlama Arayüzünün Modelle Gösterimi

Parameters

A. 1- Algoritma Seciniz:  
GA

A. 2- Simulasyon Suresi:  
2.000

A. 3- CUPA Algoritmasi:

A. 4- Birey Sayisi:  
10

A. 5- Maksimum Iterasyon:  
20

A. 6- Manuel Parametreler:

B. 1- Genetik Ebeveyn Secimi:  
Turnuva

B. 2- Genetik Carprazlama:  
Bir-Yerden

B. 3- Genetik Mutasyon Yuzdesi:  
0,5

B. 4- Genetik Ebeveyn Secim Yuzdesi:  
70

C. 1- PSO C1:  
0,5

C. 2- PSO C2:  
1,5

D. 1- ABC M - Parametre Yuzdesi:  
100

E. 1- FA Alpha:  
0,05

E. 3- Gamma:  
2,5

F. 01- Hareket Mesafesi:  
400

F. 02- Gecerli Rastgele Hiz:  
1.424.812.489

F. 03- Kurt Baslangic Sayisi:  
50

F. 04- Koyun Baslangic Sayisi:

Run Options Parameters Scenario Tree User Panel

Şekil 7.2 Geliştirilen Parametre Ayarlama Aracı Arayüzü

### 7.3.1. Test işlemleri için Geliştirilen Ek Arayüz

Şekil 7.3, Şekil 7.4 ve Şekil 7.5’de arayüzü görünen araç ayarlama sonucu bulunan en iyi parametrelerin test edilmesini hızlandırmak ve kolaylaştırmak amacı

ile geliştirilmiştir. Şekil 7.3’de yer alan arayüz parametreleri Şekil 7.4 ve Şekil 7.5’in sol bölümünde yer alan parametre alanında daha büyük verilmiş, ayrıntılar daha net görülmektedir. Verilen bu arayüzler parametre ayarlama işleminde modele ait bulunan en iyi parametre setinin manuel girme ya da dosyadan okunma alternatifleri ile model üzerinde test edilme işlemini gerçekleştirmektedir. Şekil 7.5’de verilen arayüzün Şekil 7.1’de yer alan arayüzden farkı, parametreleri tek tek elle girmek yerine dosyadan tüm parametreleri okuyarak modeli çalıştırma ya da isteğe bağlı olarak birden fazla parametre setini sırayla dosyadan okuyup model üstündeki başarısını sırayla test etme alternatifini sunmasıdır. Bu da test sürecinin daha hızlı ve kolay gerçekleştirimini sağlamaktadır.

Şekil 7.3’de yer alan ifadelerin kullanım amaçları aşağıda yer alan başlıklarla açıklanmıştır.

**1-Dosyadan Parametreler:** Bu seçenek seçildiğinde Parametreler.txt isimli dosyaya kopyalanan çözüm kümesini (parametre setini) ya da kümelerini işletecektir.

**2-Sıralı Aynı Parametreleri Simüle Et:** Seçeneği seçildiğinde Parametreler.txt dosyasından çekilen çözüm kümesini ya da E bölümünde manuel girilen parametre değerleri kullanılarak. 4. Bölümde belirlenen süre bittiğinde aynı değerlerle tekrar benzetimin başlatılması işlemini yapar.

**3- Otomatik Parametre Değiştir:** Seçeneği Seçili olduğunda Parametre.txt dosyasında yer alan çözüm kümelerini sırasıyla her biri için 4- Adım Sayısı bölümünde belirlenen adım sayısı boyunca işletir.

**4- Otomatik Parametreler:** Bu seçenek seçili olduğunda E bölümünde manuel olarak girilen parametre değerleri işletilmez.

**5- Adım Sayısı:** Bu alan bir çözüm kümesinin benzetimi işleteceği adım sayısının belirlendiği yerdir.

**6- Durma Adım Sayısı:** Benzetim işletiminin biteceği adım sayısıdır.

**E Bölümü:** Simüle edilen modele ait parametrelerin girildiği alan 3- Otomatik Parametreler seçeneği seçilerek bu alan ihmal edilebilir.

Parameters

1 - Dosyadan Parametreler:

2 - Sırayla aynı parametreleri simüle et:

3 - Parametreleri otomatik değiştir:

4 - Otomatik Parametreler:

5 - Adım Sayısı:

6 - Durma Adım Sayısı:

E.01- Hareket Mesafesi:

E.02- Geçerli Rastgele Hiz:

E.03- Kurt Başlangıç Sayısı:

E.04- Koyun Başlangıç Sayısı:

E.05- Cimenin Başlangıç Sayısı:

E.06- Kurtun Yemekten Enerjisi:

E.07- Koyunun Yemekten Enerjisi:

E.08- Kurtun Cogalması:

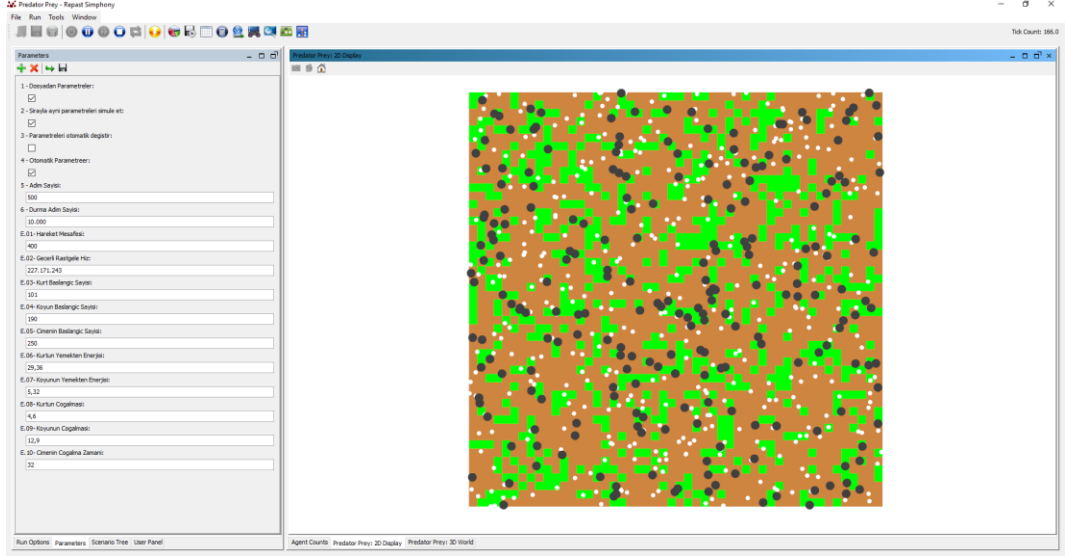
E.09- Koyunun Cogalması:

E.10- Cimenin Cogalma Zamanı:

Run Options Parameters Scenario Tree User Panel

Şekil 7.3 Modele Ait Bulunan En İyi Parametre Kümesinin Test Edildiği Arayüz

Model parametreleri, geliştirilen hibrit yaklaşımdaki en uygun algoritma kullanılarak bulunup manuel olarak girilmektedir ve bulunan uygun parametre değerlerinin modeldeki başarısı gözlemlenebilmektedir. Bu işlem için Şekil 7.4’de test işleminin model gösterimi yer almaktadır. Şekil 7.5’da ise test işleminin grafiksel gösterimi yer almaktadır.



Şekil 7.4 Modele Ait Bulunan En iyi Parametre Kümesinin Test Edildiği Arayüzün Modelle Gösterimi

Şekil 7.5’de görüldüğü gibi model parametrelerine ait bulunan en iyi çözüm kümesinin çalıştırılması sonucu elde edilen grafik için bir örnektir. 3000 adım sayısı boyunca çalıştırılan benzetimde hedeflenen popülasyon devamlılığı sağlanmıştır. Doğal yaşamda yani ekolojik sistemde gözlemlenen popülasyon değişim grafiği Şekil 7.5’de gözlemlenmekte olup elde edilen grafik Şekil 8.1’deki grafik ile uyumludur.



Şekil 7.5 Modele ait bulunan en iyi parametre değerlerinin testi sonucu elde edilen grafik

## 8. KULLANILAN MODELLER VE DENEYSEL ÇALIŞMA SONUÇLARI

Kullanılan GA, PSO, ABC ve FA algoritmalarına ait kritik parametre değerlerinin, değer belirleme tekniğine (rastgele atanarak, CUPA algoritması ile uyarlanarak veya manuel girilerek) göre her bir meta-sezgisel algoritma için 3 farklı yapıda algoritma oluşturulmuş ve parametre ayarlama sürecine etkisi incelenmiştir. İlk olarak kullanılan MSA algoritmalarının sahip olduğu kritik parametre değerleri rasgele belirlenmiş ve bu algoritmalar Random GA, Random PSO, Random ABC ve Random FA olarak isimlendirilmiştir. İkinci adımda kullanılan MSA algoritmalarının kritik parametre değerleri CUPA algoritması kullanılarak çevrimiçi ayarlanmış bu algoritmalar, Uyarlanabilir Genetik Algoritma (Adaptive Genetic Algorithm (AGA)), Uyarlanabilir Parçacık Sürü Optimizasyonu (Adaptive Particle Swarm Optimization (APSO)), Uyarlanabilir Yapay Arı Kolonisi (Adaptive Artificial Bee Colony (AABC)) ve Uyarlanabilir Ateş Böceği Algoritması (Adaptive Firefly algorithm (AFA)) olarak isimlendirilmiştir. Son olarak uyarlanabilir MSA algoritmalarının parametre ayarlama sürecinde CUPA tarafından bulunan en iyi kritik parametre değerleri MSA algoritmasına manuel girilerek Sonra AGA, Sonra APSO, Sonra AABC ve Sonra AFA algoritmaları oluşturulmuştur. Elde edilen bu 12 algoritmanın parametre ayarlama sürecindeki başarısı test edilmiştir. Her bir algoritma ayrı ayrı model parametre ayarlama işleminde kullanılmış ve 10 kez kullanıldığı model üzerinde çalıştırılmıştır. Elde edilen sonuçların ortalaması alınmış ve farklı modellere ait Çizelge 8.6, Çizelge 8.8 ve Çizelge 8.9'da ayrıntılı bir şekilde gösterilmiştir. Her bir algoritmadan elde edilen sonuçlar her bir model için karşılaştırılmıştır.

Test amaçlı kullanılan modeller, etmen tabanlı Av-Avcı (Predator-Prey), Sekiz-Vezir (Eight-Queen) ve İnsan-Zombi (Human-Zombie) modelleridir. Bu modellerin parametre ayarlama sürecinde elde edilen sonuçlar sırası ile Çizelge 8.6, Çizelge 8.8 ve Çizelge 8.9'da verilmiştir. Bu tez çalışmasının en önemli katkılarından biri kullanılan MSA algoritmalarının sahip olduğu kritik parametre değerlerini kullanıldığı modele ya da probleme uyarlayabilen AGA, AABC, APSO ve AFA algoritmalarının geliştirilmesidir.

Parametre ayarlama problemini çözmek için geliştirilen aracın etkinliğini test etmek amacı ile kullanılan model detayları ve model parametre ayarlama işleminden elde edilen sonuçlara ait detaylar bu bölümün devamında verilmiştir.

### 8.1. Model ve Kullanılan Kaynak Detayları

Çizelge 8.1 deneyler için kullanılan PC'nin özellikleri ve aracın geliştirildiği araç ve programlama dili hakkında bilgi vermektedir.

Çizelge 8.1 Deneyler İçin Kullanılan PC Detayları

Sistem Özellikleri	-Windows 7 Professional 64 bit
RAM	-8 GB
CPU	-Intel (R) Core (TM) i7 2.10 GHz
Kullanılan Algoritmalar	-Random GA, Random PSO, Random ABC ve Random FA -GA, PSO, ABC, FA -AGA, APSO, AABC, AFA
Kullanılan Araç ve Programlama Dili	-Repast Symphony 2.5-Eclipse-Java

Çizelge 8.2'de tüm algoritmalar için kullanılan modelin işletilmesinde, parametre ayarlama önemli etkiye sahip sabit olarak verilen ve kullanıcı tarafından manuel olarak değiştirilebilen tüm algoritmalar için ortak olan parametre değerleri mevcuttur. Benzetim süresi her bir çözüm kümesi için modelin işletileceği adım sayısını verir. Popülasyon, her bir MSA algoritmasının probleme bağlı olarak oluşturduğu çözüm kümeleri topluluğunu temsil eder. Popülasyonda yer alan her bir birey bir çözüm kümesini temsil etmektedir. İterasyon sayısı toplam kaç jenerasyonun (iterasyonun) oluşturulup simüle edileceğini gösterir.

Bu çalışmada av-avcı ve insan-zombi modeli için kullanılan 2000 adım sayısı, sistem kısıtları göz önünde bulundurularak belirlenmiştir. Av-Avcı modelinde ve İnsan-Zombi modelinde amaç mevcut türler tükenmeden benzetimi en uzun çalıştırmaktır. Bundan dolayı bu iki modelde adım sayısının artması çözüm kümelerinin optimuma yaklaşmasında etkindir. 2000 adım sayısı üzerinde değer seçilmesi durumunda kullanılan sistem bu modeller için yetersiz kalmaktadır. Aynı durum iterasyon sayısı ve popülasyon boyutu içinde geçerlidir. İterasyon sayısının artırılması arama uzayında daha fazla arama yapmayı sağlayacaktır böylece daha detaylı yapılan arama ile daha iyi çözüm kümelerinin bulunma ihtimali artacaktır. Fakat kullanılan sistem donanımı 20 iterasyon sayısının üzerinde yetersiz kalmaktadır. Çünkü iterasyon sayısının artması yeni çözüm kümelerinin üretilmesine ve benzetimin her yeni çözüm kümesi için tekrar tekrar çalıştırılmasına neden olmaktadır. Popülasyon boyutunun artırılması daha çok çözüm kümesinin oluşturulmasına yine daha geniş alanda arama imkânı sunmakta olup daha fazla

sayıda benzetimin çalışmasına neden olmaktadır buda sisteme ek yük getirecektir. Kullanılan sistem için yeterliliği kabul edilen iterasyon sayısı, popülasyon boyutu ve adım sayısı manuel optimize edilmiştir. Bu koşullar altında belirlenen adım sayıları ve iterasyon sayıları av-avcı ve insan-zombi modeli için hedefe ulaşmada yeterli olmuştur. Sekiz-vezir modelinde her bir zaman adımında bir çözüm üretildiği için adım sayısı 1 olarak belirlenmiştir. Maksimum iterasyon sayısının 1000 olarak belirlenmesindeki amaç tüm algoritmaların optimizasyon işleminde başarılı olup olmadığını gözlemleyebilmektir. 1000 adım sayısını küçültmek daha kısa sürede çözümlerin elde edilmesini sağlayacaktır. Özellikle sekiz vezir probleminde iyi parametre setleri bulan Sonra AGA ve Sonra AABC algoritmalarına ait sonuçlar Çizelge 8.8’de yer almaktadır. Sekiz vezir model parametrelerini ayarlama başarılı olan bu iki algoritmanın popülasyon boyutu 100 olarak ayarlandığında Sonra AGA algoritmasının 3 iterasyonda Sonra AABC algoritmasının ise 10 iterasyonda sonuca ulaştığı gözlemlenmiştir. Hedefler doğrultusunda araç üzerinde gerekli ayarlamalar yapmak mümkündür.

Çizelge 8.2 Tüm Algoritmalar İçin Kullanılan Ortak Parametreler

Parametre Değişken	Av-Avcı Modeli	8 Vezir Modeli	İnsan-Zombi Modeli
Benzetim İşletilme Zamanı	2000 Adım Sayısı	1 Adım Sayısı	2000 Adım Sayısı
Popülasyon Boyutu	10	10	10
Maksimum İterasyon	20	1000	20

Çizelge 8.3’de Algoritmalara ait parametre değerleri mevcuttur. Bu değerler her problemde problemin ihtiyacına bağlı olarak değişmekte olup farklı modellerde farklı değerler alabilmektedir. Çizelge 8.3’deki değerler uyarlanabilir çevrimiçi ayarlama mekanizması kullanılarak problemde alınan geri bildirimler sonucu elde edilen değerlerdir. Görüldüğü üzere farklı problemler için farklı değerler elde edilmiştir.

Çizelge 8.3 Algoritmelerde Kullanılan Parametre Değerleri

Algoritma	Kritik Parametreler	Av-Avcı Modeli	Sekiz-Vezir Modeli	İnsan-Zombi Modeli
GA	Seçim Yöntemi	Turnuva	Turnuva	Turnuva
	Çaprazlama Yöntemi	Bir Yerden	Bir Yerden	İki noktalı
	Ebeveyn Seçim Yüzdesi	27	30	29
	Mutasyon Yüzdesi	0.8	0.5	0.7
PSO	C1	0.8	0.5	0.5
	C2	1.3	1.5	1.4
ABC	MR	100	10	70
FA	Alpha	0.08	0.05	1.15
	Gama	2.5	2.5	4.7

## 8.2. Av-Avcı (Predator-Prey) Modeli

### 8.2.1. Av-Avcı İlişkisi

Biyolojik modeller son yıllarda oldukça fazla çalışma alanı bulmuştur. Popülasyon temelli olan bu modellerden biri olan Av-Avcı modeli bu tez çalışmasında incelenmiştir. Bu modelde iki popülasyon vardır. Bu modelde yer alan bu popülasyonlardan av türü avcı türünü yiyerek beslenen bir mekanizma mevcuttur. Avlar ise ortamda ki başka yiyecekler ile beslenir. Kurt ve koyun örneğini incelersek kurt koyunları koyunlarda çimenleri yiyerek beslenen bir av-avcı modelidir. Ekolojiden buna benzer birçok av-avcı örneği verilebilir. İlk olarak av-avcı modeli köpek balığı (avcı) ve balık (av) popülasyonundaki değişimin analiz edilmesi için geliştirilmiştir (Voltera, 1978). Bu model için popülasyon yoğunluklarını bulan bir takım hesaplamalar yapılmıştır.

Ortamda avcı popülasyonu yoksa av popülasyonundaki değişim; Popülasyon  $ds/dt=as$   $a>0$  oranında artacaktır

Ortamda av popülasyonu yoksa avcı popülasyonu,  $dw/dt=-bw$   $b>0$  oranında azalacaktır.

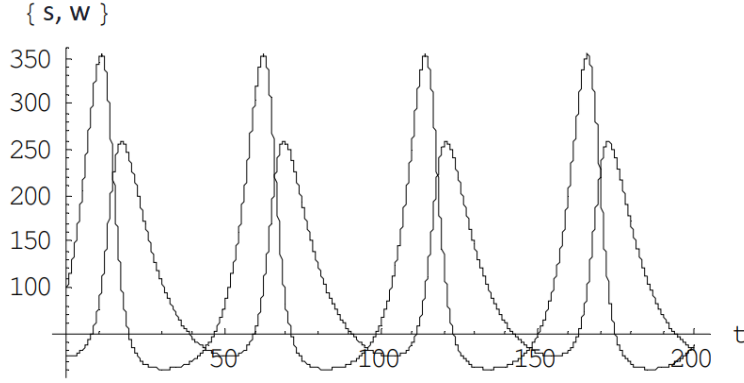
Büyüme ve azalma oranı her iki türün aynı ortamda bulunması durumunda türlerin karşılaşması yani etkileşmesi oranı sıklığında değişecektir. Bu değer  $s.w$  değeri ile orantılı olduğu kabul edilir.

Bu durumda av oranı  $-hsw$   $h>0$  oranında azalmaktadır, Avcı popülasyonu  $gsw$   $g>0$  oranında artmaktadır. Bu değerler tek bir denklemde birleştirildiğinde av ve avcı yoğunluğunu bulan denklemler aşağıdaki gibidir.

$$ds/dt = as - gsw \quad 8.2.1$$

$$dw/dt = -bw + hws \quad 8.2.2$$

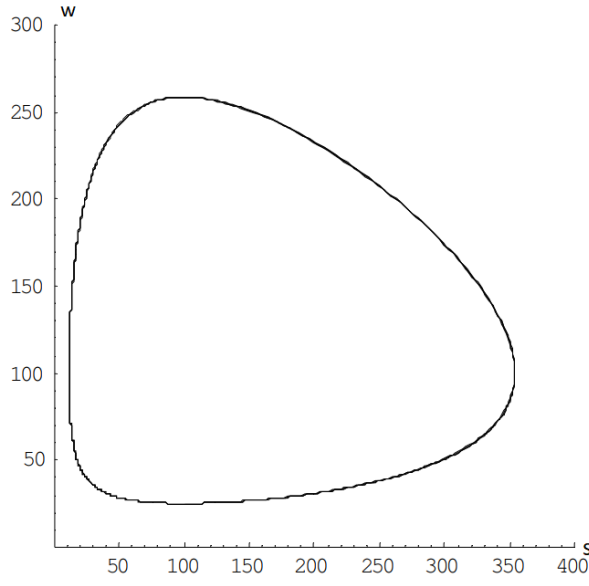
Şekil 8.1 incelendiğinde, koyun ve kurt türlerinin sayıları süreç içerisinde dalgalandığı görülür. Popülasyon hareketleri yaklaşık her 50 adımda periyodik olarak model tekrarlanır. Kurt popülasyonu azaldığında, koyun popülasyonu artmaktadır. Koyun popülasyondaki artış kurt popülasyonun besin sağlama olasılığını arttıracığından kurt popülasyonda artmaya başlayacaktır. Fakat kurt popülasyondaki artış koyun avlanma miktarında artışa neden olacağından zamanla koyun sayısı azalmaya başlayacaktır. Koyundaki azalma kurt besin miktarında azalma kurt sayısını etkileyip eğri azalma yönünde verilecektir. Sonuç olarak, kurt sayısı koyun sayısı yeniden ilk haline gelecektir. Böylece döngü yeniden başlayacaktır.



Şekil 8.1 Av-Avcı Popülasyonunun Zaman İçindeki Değişimi (Kara, 2006)

$a = 0.2$ ,  $b = 0.1$ ,  $g = 0.002$ ,  $h = 0.001$ ,  $s = 100$ ,  $w = 25$  koşulları altında yukarıdaki diferansiyel denklem sistemi sayısal yaklaşım ile çözülebilir (Kara, 2006).

Şekil 8.2 de x eksenini koyun popülasyon sayısını, y eksenini kurt popülasyon sayısının yerleştirildiği bir grafik görülmektedir. Faz diyagramı olarak adlandırılan bu grafik üzerindeki her bir nokta av-avcı sisteminde türlerin popülasyon sayılarını göstermektedir. Bu sistem mevcut eğri etrafında saatin ters istikametinde bir hareket gerçekleştiren bir noktadır (Kara, 2006).



Şekil 8.2 Av-Avcı Modelinin Faz Diyagramı (Kara, 2006)

### 8.2.2. ETMB Ortamında Av-Avcı Modeli

Modelin hedefi, bir av-avcı ekosisteminin devamlılığını sağlamak bunun için gerekli koşulları gözlemlemektir. Geliştirilen yaklaşımı test etmek için kullanılan bu benzetim modelinde, kurt-koyun ekosistemi incelenmektedir. Bu ekosistemin devamlılığına etki eden parametreler tespit edilmiş, bu parametreler ekosistemin hedefleri doğrultusunda ayarlanmıştır. Avcı-av benzetim modelinde üç etmen bulunmaktadır: Kurt, koyun ve Çimen. Kurt ve koyun etmenleri (agent) benzetim ortamında rasgele hareket etmektedirler. Ancak, hem kurt etmenlerinin hem de koyun etmenlerinin ortamda gezinebilmeleri için enerjiye ihtiyaçları vardır. Bu yüzden kurt ve koyun etmenleri belirli bir enerji ile oluşturulmuşlardır. Ortamda hareket eden kurt kuzu beraberinde enerji tüketirler. Enerjileri tükendiğinde yani 0 olduğunda yok olurlar bunu önlemek için ortamda kurt koyunu avlayarak koyun çimeni yiyerek enerjilerini arttırlar.

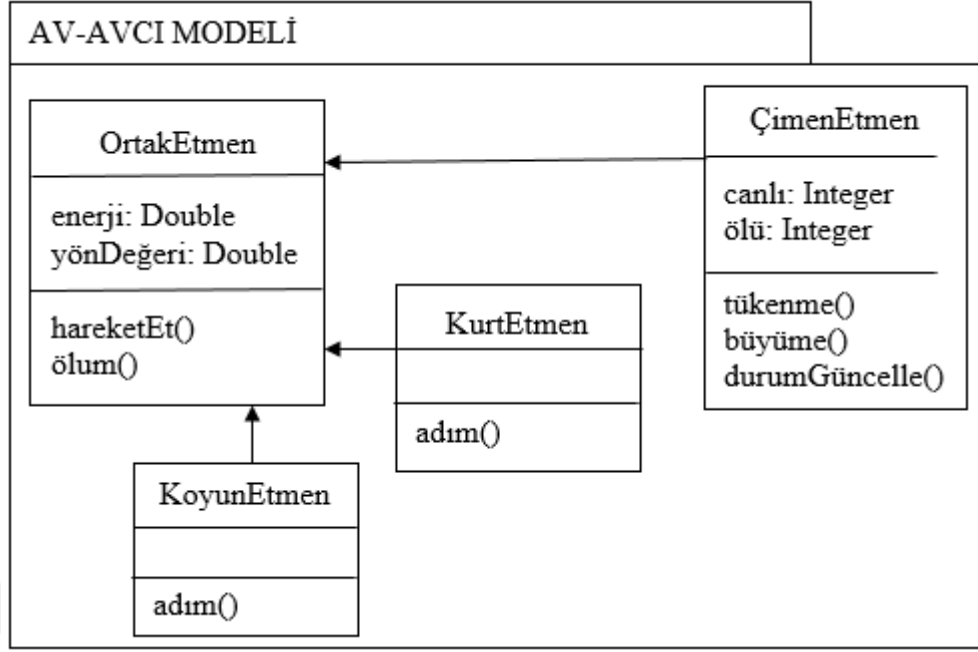
Eğer kurt etmeni bulunduğu koordinatta bir koyun etmeniyle karşılaşır, karşılaştığı koyunu avlar ve enerjisini belirlenen parametre değerine göre (*wolfgainfromfood*) arttırır. Benzer şekilde, koyun etmeni de bulunduğu koordinatta yenebilecek, canlı bir çimen etmeni ile karşılaşır onu yer ve belirlenen parametre değeri kadar (*sheepgainfromfood*) enerjisini arttırır. Çimen etmenleri, ilk oluşturulduğunda tüm alanda oluşturulur. Çimen etmenleri yaratılırken belirlenen parametre değerine göre (*grassregrowth*) büyürler. Çimen, bir kuzu tarafından

yenirse o çimen belirlenen süreye bağlı olarak tekrar oluşturulur. Çimenin yaşam süresi (*countdown*) rasgele atanır yenilmeyen çimende belli bir süre sonra yok olur sonra tekrar büyür (Korkmaz Tan ve Bora, 2018).

Av-avcı modeli kurtların ve kuzuların bulunduğu katman ve çimenlerin bulunduğu katman olmak üzere iki katmanlı olarak tasarlanmışlardır. Çimenlerin yaşam döngüleri kurt ve kuzulardan farklı işlendiği ve görselleştirildiği için farklı katmanda tanıtılmıştır. Bu yapı etmen etkileşimini etkilememektedir. Çimen etmenleri belirlenen sınırlar içinde her koordinatta rasgele ve yaşam süresi boyunca yerleştirilir. Yaşam süresi biten ya da avlanan çimen etmeni belli bir süre sonunda tekrar oluşur. Ayrıca kurt ve koyun etmenleri içinde üreme davranışı bulunmaktadır. Belirli bir olasılık çerçevesinde kurt ve koyun etmenleri üreme davranışı gösterir ve ortama yeni etmenler ekleyerek popülasyonun devamlılığını sağlarlar. Üreme davranışını yeterli enerjiye sahip etmenler arasından yapılan rasgele seçimle yapılmaktadır.

Av avcı modelinde 4 farklı sınıf mevcuttur. Bunlar *OrtakEtmen*, *KurtEtmen*, *KoyunEtmen* ve *ÇimenEtmen* olarak isimlendirilmiştir. *KurtEtmen*, *KoyunEtmen* ve *ÇimenEtmen* sınıfın ortak özelliklerinin tutulduğu *OrtakEtmen* sınıfıdır. Bu sınıfta yer alan *ölüm()* ölüm davranışını *hareketEt()* hareket etme davranışını *yönDeğiştir()* etmenin yönünü belirleyerek koordinat değiştirmesini temsil etmektedir. *hareketEt()* fonksiyonunun gerçekleştirimi her etmen için farklı olan soyut bir fonksiyondur. Şekil 8.3'de Av-Avcı modeline ait sınıf diyagramı gösterilmiştir. *KurtEtmen* sınıfı kurt etmenini temsil etmektedir. *adım()* fonksiyonu kurt ve koyunların üreme ve avlanma davranışlarının gerçekleştiği fonksiyondur.

ÇimenEtmen sınıfı içinde her etmen için rasgele belirlenen yaşam süreleri, sürekli azaltılarak çimen yaşam döngüsü temsil edilmektedir. *tükenme()* fonksiyonu çimenin koyun tarafından yenilme işlemi temsil etmektedir. *büyüme()* fonksiyonu içinde yeniden büyüme işlemi gerçekleşir. *DurumGüncelle()* çimenin mevcut durumunu karman içinde güncelleme işlemi yapar.



Şekil 8.3 Av-Avcı Modeli Sınıf Diyagramı

Bu modelde amaç ekolojik sistemin devamlılığını sağlamaktır. Kurt, koyun ve çimen etmenlerinin tükenmediği gerçek yaşamı temsil eden modeli oluşturmak amaçlanmaktadır. Bunun için bu modele ait parametre değerlerinin ayarlanması gerekmektedir. Ayarlama tabii tutulan parametre değişkenleri Çizelge 8.4’de verilmiştir.

Çizelge 8.4 ETMB Ortamında Modellenen Av Avcı Modeline Ait Parametreler

Model Parametreleri	Parametre Açıklaması	Değer Aralıkları
initialnumberofwolves	Kurt etmeninin popülasyondaki başlangıç sayısı	40-250
initialnumberofsheep	Koyun etmeninin popülasyondaki başlangıç sayısı	40-450
initialnumberofgrass	Çimen etmeninin başlangıç sayısı	
wolfgainfromfood	Kurt etmeninin avlanma sonrası kazandığı enerji	10-40
sheepgainfromfood	Koyun etmeninin avlanma sonrası kazandığı enerji	1-12
wolfreproductionrate	Kurt etmeninin üreme olasılığı	3-15
sheepreproductionrate	Koyun etmeninin üreme olasılığı	3-15
grassregrowtime	Çimen etmeninin yeniden büyüme süresi	10-60

Çizelge 8.4 ayarlanmaya çalışılan av-avcı modeline ait parametre değişkenleri olup en iyi parametre seti ile popülasyonun devamlılığını sağlamak bu

modelin hedefleri arasındadır. Başlangıç çözüm kümesi bu değerlerin alabileceği değer aralıklarına göre rasgele oluşturulmaktadır. Çözüm kümesi, *[initialnumberofwolves, initialnumberofsheep, wolfgainfromfood, sheepgainfromfood, wolfreproductionrate, sheepreproductionrate, grassregrowthtime]* değerlerinden oluşmaktadır. Toplam 10 bireyden oluşan yani 10 çözüm kümesinden oluşan başlangıç popülasyon oluşturulur ve sonrasında kullanılan algoritmalar ile arama uzayında arama yapılarak en iyi çözüm kümesi (parametre seti) bulunmaya çalışılır. Çözüm kümelerine ait birer örnek Şekil 8.4 ve Şekil 8.5’de verilmiştir.

Şekil 8.4’de görünen her bir çözüm kümesi popülasyona ait kromozomları yani parametre setlerini temsil etmektedir. İlk popülasyon Çizelge 8.4’de belirtilen parametre aralıkları dikkate alınarak rasgele atama işlemi yapılır. Çözüm kümesindeki her bir değer Çizelge 8.4’de belirtilen bir parametre değerine karşılık gelmektedir.





Çok sayıda parametreye sahip av-avcı modelinin gerçel sistemi yansıtılabilmeleri ancak uygun parametre değerlerini kullanarak çalıştırılması ile mümkündür. Bundan dolayı modellenen her karmaşık sistem için parametre ayarlama işlemi yapılmalıdır.

Her problemin (modelin) kendine ait hedefleri yani uygunluk fonksiyonları bulunmaktadır MSA'lar bu uygunluk fonksiyonlarını en iyilemeye çalışır. Av-avcı modelinde hedef parametre değerlerinin yer aldığı çözüm kümelerinin kalitesini av, avcı popülasyonlarının tükenmeden sürekli bir denge halinde olmasını sağlamaktır. Bundan dolayı zaman ve av, avcı popülasyonu uygunluk fonksiyonu için önemli değerlerdir. Ayrıca Şekil 8.1 incelendiğinde kurt ve koyun sayıları her zaman diliminde birbirini takip eden değerler olarak ilerlemektedir ve koyun sayısı kurt sayısından belli oranda fazladır. Ayrıca eşitlik 8.2.1 ve 8.2.2 incelendiğinde koyun sayısının tükenmesi, kurt sayısının artması ws oranında değişmektedir. Eğer koyun sayısı 2 katına çıkarsa tükenme oranında 2 katına çıkacaktır. Kurt ve koyun sayısı ekolojide denge halinde olmalıdır.

Amaç kurt ve koyun sayılarının tükenmediği benzetim modelini elde etmektir. Belirlenen adım sayısı sonunda benzetimde yer alan kurt ve koyun sayısı uygunluk değerini belirleyen en önemli değerlerdir. Bu durumda Eşitlik 8.2.3 de yer alan formül uygunluk değeri hesabında kullanılmıştır. Eğer türlerden biri tükenirse benzetim ne kadar uzun çalışırsa o kadar iyi uygunluk değerine sahip olacaktır.

Eğer  $w > 0$  ve  $s > 0$  ve  $w < s$  ise

$$\text{O zaman } f = 1 - ((w / s) + (t / \max T)) / 2 \quad 8.2.3$$

$$\text{Değilse } f = 1 - (t / \max T) / 2$$

$f$ , uygunluk fonksiyonunu,  $w$ , benzetimdeki kurt sayısını,  $s$ , benzetimdeki koyun sayısını,  $t$  benzetimin sonlandığı adım sayısını,  $\max T$ , benzetimin çalışabileceği ve sabit belirlenen maksimum sayıyı temsil eder. Uygunluk değeri 0-1 arasında değer almaktadır 0 en iyi değer 1 en kötü değer olarak kabul edilmektedir.

### 8.2.4. Av-Avcı Modeli Benzetim Sonuçları

#### Sonuçların Elde Edilmesi:

Kullanılan Random GA, Random PSO, Random ABC, Random FA, AGA, APSO, AABC, AFA, Sonra AGA, Sonra APSO, Sonra AABC ve Sonra AFA (Bkz. Çizelge 8.5) algoritmaları av-avcı modeline ait parametreleri ayarlamak üzere av-avcı modelini 10 kez çalıştırır. Elde edilen sonuçların ortalaması hesaplanarak algoritmalar birbirleri ile karşılaştırılır. Av-avcı modeline ait parametreleri ayarlayan en başarılı algoritma bulunur. Elde edilen sonuçların ortalamaları Çizelge 8.6'de görülmektedir.

#### Çizelge 8.6'daki Sütunların Açıklaması:

**Adım Sayısı:** ETMB'lerde etmenlerin gerçekleştirdiği her bir davranışı ya da ortamla veya başka bir etmenle etkileşimi bir zaman adımı olarak tanımlanır. Tüm benzetimlerin çalıştığı toplam adım sayısının ortalamasını verir.

**Ortalama Uygunluk:** Benzetimler çalışırken elde edilen uygunluk değerlerinin ortalamasıdır. Benzetimi çalıştıran tüm çözüm kümesinin uygunluk değerlerinin ortalamasıdır.

**Başlangıç En İyi Uygunluk:** Benzetim çalıştırıldığında ilk iterasyonda rasgele oluşturulan çözüm kümelerinin ortalama en iyi uygunluk değeridir. Bu değer algoritmanın uygunluk değerini ne kadar geliştirebildiğinin kıyaslanması için önemlidir. Örneğin algoritma en iyi uygunluk değerini 0.10 buldu ise ve zaten ilk iterasyonda rasgele oluşturulan kümelere biri 0.10 uygunluk değerine sahipse algoritma bu değere kendi ulaşmadığı yorumu yapılabilmektedir.

**En İyi Uygunluk:** Her bir çalıştırma sonucu bulunan en iyi çözüm kümelerinin uygunluk değerlerinin ortalamasıdır.

**İyileştirme Farkı:** Bu sütun değeri, "Başlangıç En İyi" sütununun "En İyi Uygunluk" sütunundan çıkartılmasıyla bulunur. Başlangıçta bulunan rasgele en iyi çözüm kümesinin ne kadar iyileştiğini gösterir.

**Ortalama Yerel Uygunluk:** Her bir iterasyonda bulunan en iyi yerel uygunluk değerlerinin ortalamasını verir.

**Ortalama Genel Uygunluk:** Tüm benzetimlerde her bir iterasyondan elde edilen genel en iyi uygunluk değerlerinin ortalamasını verir.

**İyileştirme Sayısı:** Uygunluk değerinin ortalama kaç iterasyonda iyileştirildiğinin değeridir.

Çizelge 8.5’da parametre ayarlama sürecinde parametre ayarlama işlemi için kullanılan algoritmalar ve algoritmalara ait açıklama yer almaktadır.

Çizelge 8.5 Çizelge 8.6’deki Satırların-Kullanılan Algoritmaların Açıklaması.

Kullanılan Algoritmalar	Kritik Parametrelere Değer Atama Yöntemi	Açıklama
Random GA Random PSO Random ABC Random FA	Rasgele atanır	Rasgele Atanan kritik parametre değerleri ile algoritmalar çalıştırılır.
AGA APSO AABC AFA	Çevrimiçi uyarlanır parametre ayarlama yöntemi kullanılarak atanır.	Çevrimiçi uyarlanır parametre ayarlama yöntemi ile probleme uygun kritik parametre değerleri alır ve algoritmalar problemden(benzetimden) aldığı geri bildirimlerle sürekli güncellenen bu kritik parametre değerleri ile çalıştırılır.
Sonra AGA Sonra APSO Sonra AABC Sonra AFA	AGA, APSO, AABC ve AFA algoritmalarının çalıştırılması sonucunda elde edilen en iyi kritik parametre değerleri GA, PSO, ABC, FA algoritmalarına manuel atanır.	Manuel girilen kritik parametre değerleri ile algoritma çalıştırılır.

### Sonuçların Analizi:

Çizelge 8.6 incelendiğinde GA algoritmasının iyileşme oranının çok iyi olduğu görülmektedir. Rasgele alınan parametre değerleri ile elde edilen Random GA, AGA ve Sonra AGA sonuçları karşılaştırıldığında Sonra AGA’nın çok iyi uygunluk değerine sahip olduğu görülmektedir. Bu da ABC, FA VE PSO

algoritmalarından daha iyi çözüm ürettiği anlamına gelmektedir. Ayrıca adım sayısına bakıldığında GA algoritmasının diğer algoritmalarından çok kısa sürede çok iyi çözüm ürettiği görülmektedir. GA algoritmasının daha hızlı olmasının iki nedeni vardır:

1. Her iterasyonda ebeveyn seçim yüzdesi kadar yeni çözüm kümeleri meydana getirildiği için seçim yüzdesine bağlı olarak her iterasyonda diğer algoritmaların aksine daha az çözüm kümesi oluşturur. Böylece diğer algoritmalarından daha az sayıda çözüm kümesinin uygunluk değeri hesaplanır. AGA algoritması çalıştırıldığında ebeveyn yüzdesi 20-80 arasında belirlenmiştir. Diğer algoritmalarda 10 olan çözüm kümesi sayısı bu algortmada maksimum 8 olmaktadır.

2. GA'da oluşturulan çocuklar ebeveynlerinin genlerini aldığı için yani çözüm kümeleri çaprazlandığı için bir birine benzer çözüm kümeleri diğer algoritmalara göre daha fazla oluşmaktadır. Bundan dolayı aynı çözüm kümesinin uygunluk değerleri kaydedilmektedir. Uygunluk değeri bilinen çözüm kümeleri tekrar simüle edilmemektedir bu da süreyi kısaltmaktadır.

Adım sayısının Random GA<AGA< Sonra AGA olmasının nedeni kullanılan av-avcı modelinde çözüm üretim süresini kısaltmak amacı ile algoritmalarda türlerin tükendiği benzetimin belirlenen adım sayısı (2000) kadar devam etmesini engellemek için benzetim sonlandırılmaktadır. Av-Avcı modelinde amaç türlerin tükenmesini engelleyen ekosistemin devamlılığını sağlayan çözüm kümelerinin (parametre setlerini) bulmaktır. Türlerin tükendiği çözüm kümeleri kötü olarak nitelendirilmektedir. Bundan dolayı Random GA daha çok kötü çözüm ürettiği için AGA ve Sonra AGA'dan daha kısa sürmüştür. Aynı şekilde AGA algoritmasındaki kötü çözüm sayısının Sonra AGA'dan fazla olması daha kısa sürmesine neden olmuştur. Genel uygunluk Ortalaması ve İyileştirme Sayısı ve En İyi Uygunluk değerleri de bu sıralamayı göstermektedir. Bu durum PSO, ABC ve FA algoritmasında da gözlemlenmektedir.

PSO algoritması incelendiğinde diğer algoritmalara göre iyileştirme oranının çok yüksek olduğu görülmektedir. Bu algoritmanın önemli avantajlarından biridir. Fakat ürettiği çözüm kümesinin en iyi uygunluk değeri diğer algoritmalara göre daha kötü olduğu görülmektedir. Adım sayısına bakıldığında Sonra APSO algoritmanın Sonra AGA ve Sonra ABC algoritmasından daha yavaş olduğu görülmektedir üstelik bu algoritma daha kötü uygunluk değerlerini simüle etmesine

rağmen daha uzun sürede çözüme ulaştığı gözlemlenen diğer dezavantajdır. Bu durum algoritmanın yapısından kaynaklı olduğunu yani yavaş çalıştığını göstermektedir.

FA algoritmasına bakıldığında Sonra AFA'nın uygunluk değeri Random FA ve AFA değerlerine göre çok daha iyi olduğu görülmektedir. Adım sayısının giderek artması Sonra AFA algoritmasının daha iyi uygunluk değerleri ile benzetimin çalıştırdığı genel uygunluk değerlerine bakıldığında da görülmektedir. Ayrıca iyileştirme sayısı Random FA ve AFA algoritmaların 2 katı kadar bir gelişme göstermiştir. Diğer algoritmalarla karşılaştırıldığında Sonra AGA algoritmasından sonra en iyi çözümü bulan algoritma olduğu görülmektedir. Sonra AGA algoritmasının çözüm kümesinin daha iyi olmasının diğer bir nedeni başlangıç uygunluk değerinin Sonra AFA'ya göre daha iyi olmasıdır. İyileştirme farkına bakıldığında Sonra AFA'nın Sonra AGA'dan daha iyi olduğu bu nedenle iyi çözüm üretmede Sonra AGA'dan daha iyi olabileceği söylenebilir. Fakat adım sayılarına bakıldığında Sonra AFA diğer algoritmalara göre çok uzun sürede çözüme ulaştığı gözlemlenmektedir. Bununda Sonra AFA'nın daha çok işlemci kullandığı buda FA algoritmasının daha fazla döngü ve işlem içerdiği anlamına gelmektedir.

ABC algoritması, AABC ve Sonra AABC ile çalıştırılan benzetimden elde edilen sonuçlara bakıldığında Sonra ABC'nin daha iyi uygunluk değeri ürettiği gözlemlenmektedir. Fakat AABC ile Sonra AABC algoritmasının en iyi uygunluk değerleri karşılaştırıldığında AABC algoritmasında yaklaşık 0,13 olan değer Sonra AABC algoritmasında yaklaşık 0,10 olduğu görülmektedir. Bu da bu iki algoritmanın iyileşme boyutunda çok büyük farkın olmadığını göstermektedir. Bunun nedeni ABC algoritmasının sahip olduğu M (modification rate-MR) parametresinin almış olduğu değerlerdir. AABC algoritmasından elde edilen sonuçlar incelendiğinde M parametresinin çoğunlukla 50'nin üzerinde değerler aldığı ve Sonra ABC algoritmasının kullandığı en iyi değer ise 100 olarak bulunduğu bu nedenle AABC ve Sonra AABC algoritmalarından elde edilen sonuçların birbirine yakın çıktığı sonucuna varılmıştır. Bu yüzden Random ABC algoritmasına kritik parametre değeri olan M değişkenine en kötü kritik değer olan 10 değeri verilerek algoritma çalıştırılmıştır. Random olarak atanacak değer 50'nin üzerinde olma ihtimalini önlemek amaçlanmıştır. Random ABC algoritmasının sonuçlarına bakıldığında bulunan en iyi uygunluk değeri yaklaşık 0,31 olup AABC ve Sonra AABC algoritmalarından çok daha kötü çözüm kümeleri ürettiği elde edilen sonuçlardandır. Bu değerler ABC algoritmasının sahip olduğu

M parametresinin almış olduğu değerlerin algoritmanın performansı üzerindeki etkisini göstermektedir.

Özetlenecek olursa, uyarlanabilir algoritmalar kullanılarak bulunan kritik parametre değerleri önemli ölçüde algoritmaları iyileştirmiş ve algoritmanın daha iyi çözümlere yaklaşmasını sağlamıştır. İyi olan çözüm kümeleri av-avcı model hedefine bağlı olarak modelin daha uzun çalışmasını sağlamaktadır. Modelin hedefi türlerin devamlılığını sağlayacak çözüm kümeleri üretmek olup türlerin tükenmesi ile benzetim çalışmasında sonlandırmaktır. Bu özelliğinden dolayı çözüm kümelerinin iyileşmesi benzetimin çalışma süresini de arttıracaktır. Algoritmalarda zaman adımı olarak tanımlanan adım sayılarının her algoritma grubunda(GA, PSO, ABC veya FA) gözlemlenen artış, algoritmanın performansı ile doğru orantılıdır. Eğer av-avcı benzetiminde tür tükendiğinde benzetim sonlandırılmasaydı çevrimiçi uyarlanabilir (Adaptive) algoritmalarla uyarlama sonucu elde edilen Sonra kelimesi ile nitelendirilen algoritmaların çalışma süreleri birbirine yakın olacaktı.

Çizelge 8.6 Etmen Tabanlı Model parametrelerinin ayarlama işleminde kullanılan algoritmalarından elde edilen sonuçlar

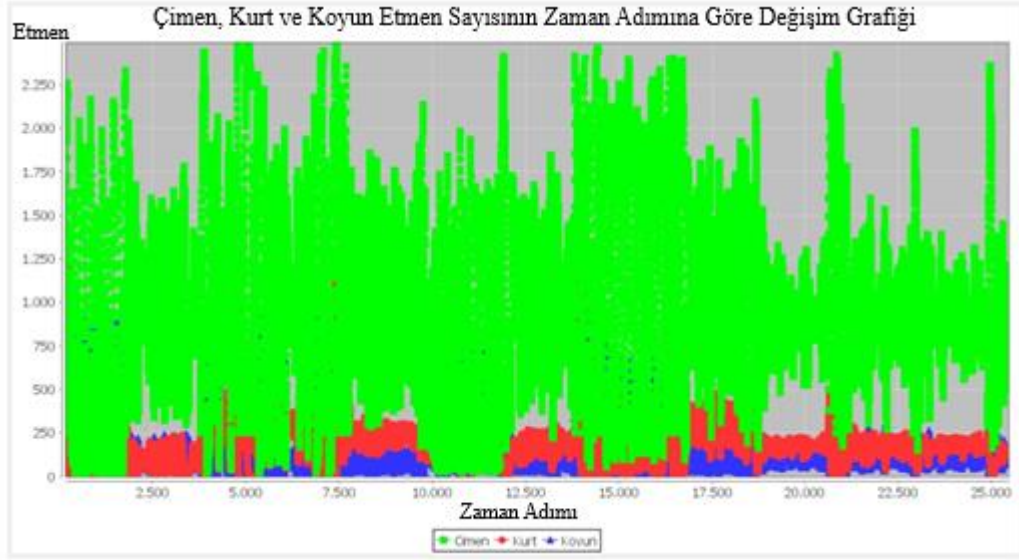
Algoritma	Adım Sayısı	Ortalama Uygunluk	Başlangıç En İyi Uygunluk	En İyi Uygunluk	İyileştirme Farkı	Ortalama Yerel Uygunluk	Ortalama Genel Uygunluk	İyileştirme Sayısı
Random GA	12.675,000	0,84535294	0,90000000	0,22941176	0,67058824	0,35714286	0,35714286	2,00
AGA	96.530,750	0,62217563	0,51184535	0,23496267	0,27688268	0,35871368	0,33094627	3,10
Sonra AGA	156.468,222	0,40666325	0,32946697	0,07659524	0,25287173	0,14464858	0,13753509	4,80
Random PSO	95.592,000	0,86200000	0,90000000	0,58823529	0,31176471	0,71382353	0,64823529	3,00
APSO	193.832,250	0,77432176	0,42269499	0,22610476	0,19659023	0,44587879	0,26873811	2,00
Sonra APSO	270.397,125	0,57842165	0,61294118	0,12971985	0,48322132	0,31676982	0,23409716	4,40
Random ABC	107.464000	0,84126697	0,90000000	0,31176471	0,58823529	0,64735294	0,56058824	3,00
AABC	172.250,250	0,71320468	0,61176471	0,13112948	0,48063523	0,27787187	0,27787187	4,80
Sonra AABC	211.718,625	0,73226898	0,45238971	0,10053188	0,35185783	0,39514715	0,24317515	4,80
Random FA	84.006,000	0,86284363	0,54705882	0,21000000	0,33705882	0,66911765	0,33794118	2,000
AFA	128.430,444	0,78739367	0,45218335	0,11294583	0,33923752	0,46062434	0,16894793	2,400
Sonra AFA	373.799,500	0,42965654	0,48823529	0,08884096	0,39939434	0,23300848	0,19495409	5,600

Rasgele atanan kritik parametre değerlerinin kullanıldığı algoritmaların ürettiği çözüm kümelerine ait en iyi uygunluk değerleri Çizelge 8.7’de görülmektedir. Sonuçlar incelendiğinde Av avcı modeli için en iyi uygunluk değerine sahip algoritmalar Random, Uyarlanabilir ve Sonra olarak gruplandırıldığında her grup içinde GA ve FA algoritmalarının başarılı olduğu elde edilen sonuçlardandır.

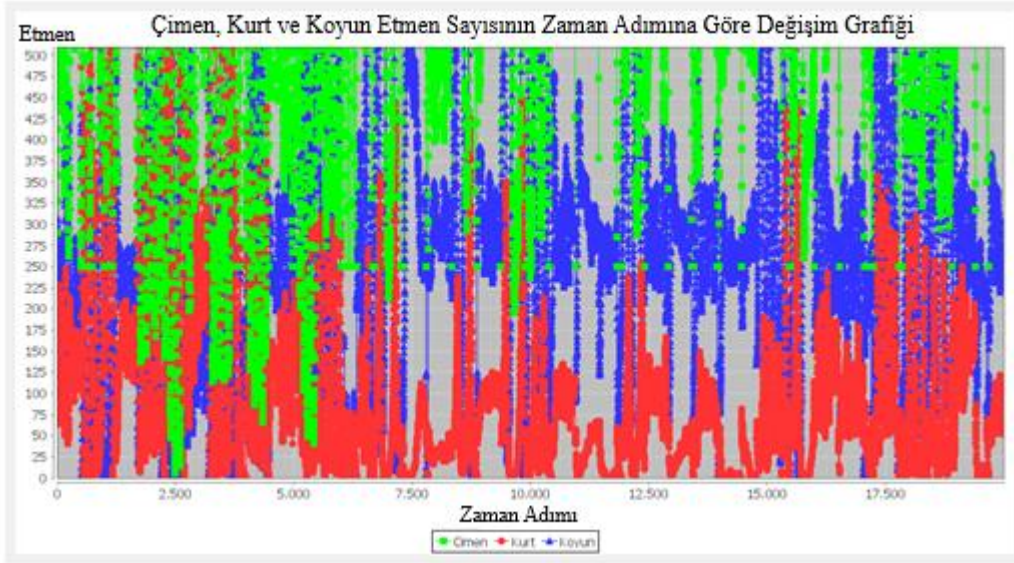
Çizelge 8.7 Algoritmaların en iyi uygunluk değerleri

Kullanılan Algoritmalar	Bulunan En İyi Uygunluk değerleri
<b>Random GA</b>	0,22941176
<b>Random PSO</b>	0,58823529
<b>Random ABC</b>	0,31176471
<b>Random FA</b>	0,21000000
<b>AGA</b>	0,23496267
<b>APSO</b>	0,22610476
<b>AABC</b>	0,13112948
<b>AFA</b>	0,11294583
<b>Sonra AGA</b>	0,07659524
<b>Sonra APSO</b>	0,12971985
<b>Sonra AABC</b>	0,10053188
<b>Sonra AFA</b>	0,08884096

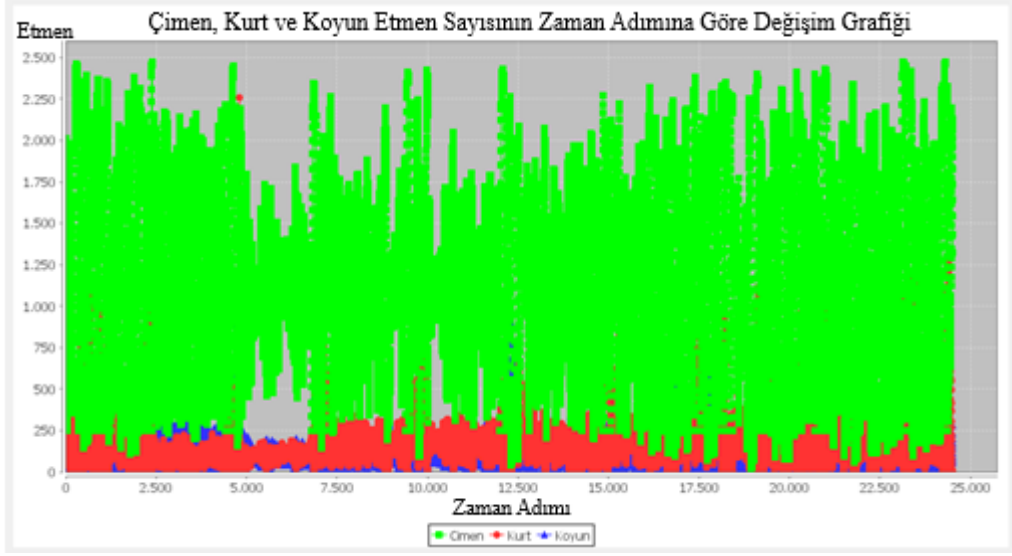
Random GA, Random PSO, Random ABC, Random FA algoritması için Şekil 8.6, Şekil 8.7, Şekil 8.8 ve Şekil 8.9’deki grafiklerde av-avcı modelindeki parametreleri ayarlamak üzere kurt, koyun ve çimen etmen sayılarının aldıkları parametre değerlerine göre değişimleri gözlemlenmektedir. Şekil 8.6, Şekil 8.7, Şekil 8.8 ve Şekil 8.9’deki sırayla Random GA, Random PSO, Random ABC ve Random FA’ya ait kritik parametre değişkenlerinin rasgele atanması sonucu bu algoritmaların yaptığı av avcı benzetimdeki parametrelerini ayarlama sürecini göstermektedir. Bu grafiklerde görüldüğü gibi sık aralıklarla türlerin sayısı sıfır olmaktadır. Bu algoritmaların kabul edilebilir sonucu bulabilmesi için çok daha fazla iterasyonda çalıştırılması gerekmektedir. Bu daha fazla zaman ve kaynağa ihtiyaç duyması anlamına gelmektedir.



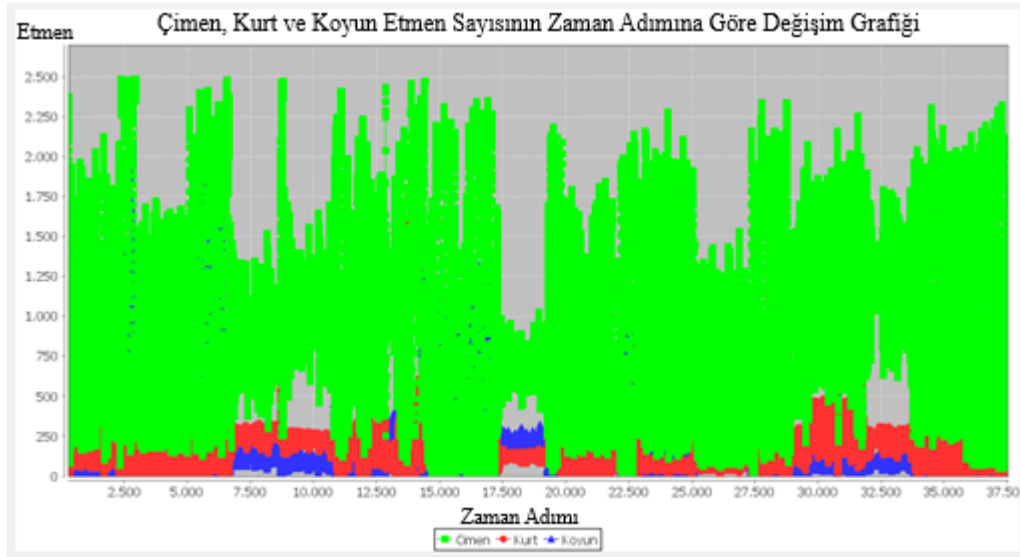
Şekil 8.6 Random GA ile Av Avcı Modelinde Parametre Ayarlama



Şekil 8.7 Random PSO ile Av Avcı Modelinde Parametre Ayarlama



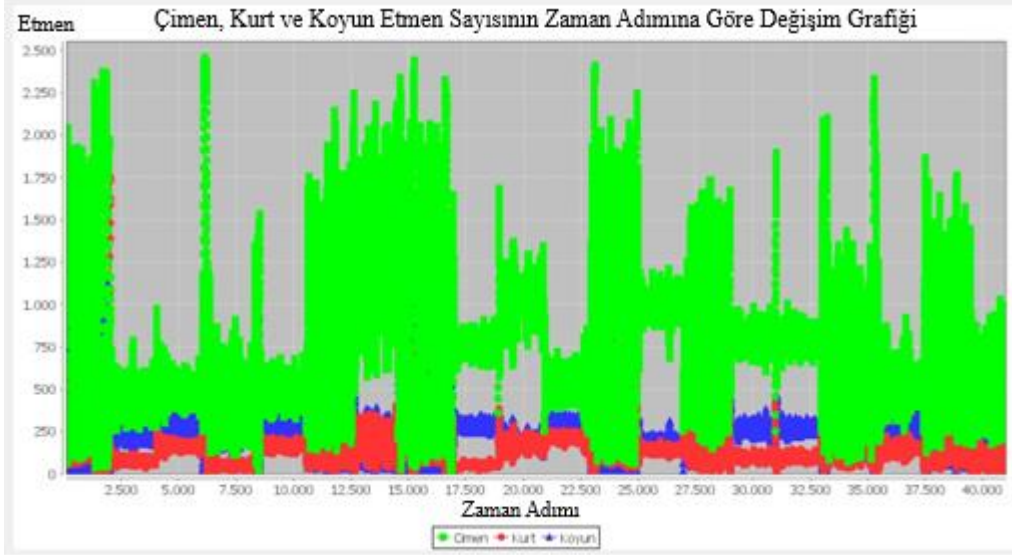
Şekil 8.8 Random ABC ile Av Avcı Modelinde Parametre Ayarlama



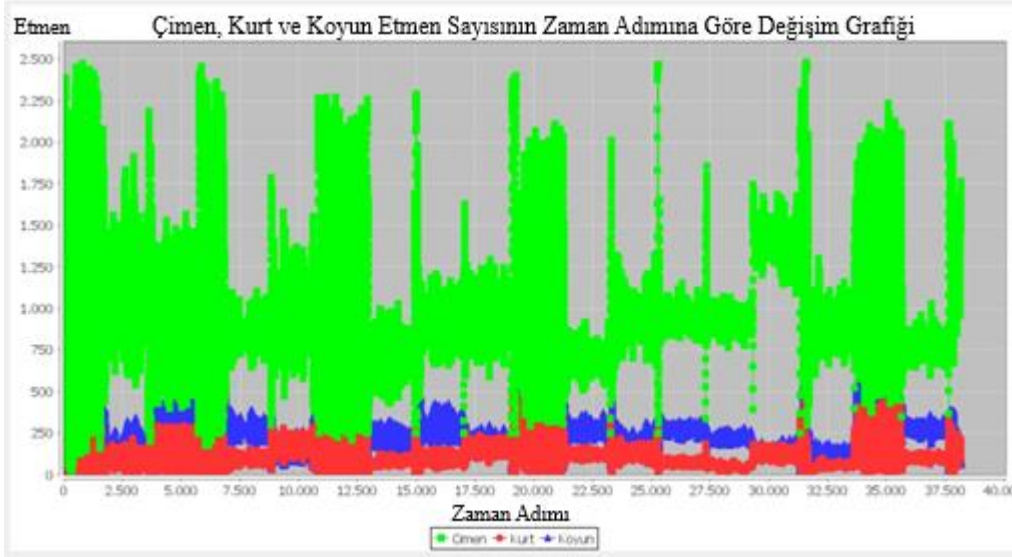
Şekil 8.9 Random FA ile Av Avcı Modelinde Parametre Ayarlama

Şekil 8.10, Şekil 8.11, Şekil 8.12 ve Şekil 8.13'daki grafikler geliştirilen AGA, APSO, AABC ve AFA algoritmalarının parametre ayarlama sürecinde yine av avcı benzetimindeki etmenlerin değişimini göstermektedir. Bu grafikte algoritmaya ait olan kritik parametre değişkenleri her benzetimden alınan geri dönütlerle iyileştirilip tekrar parametre ayarlama işlemi için benzetim çalıştırılmaktadır. Grafik Random ABC algoritmasındaki grafiğe göre ilk birkaç iterasyon benzerlik gösterse de daha sonraki iterasyonlarda yeni kritik değişken değerleri ile bazı aralıklarda benzetimin daha uzun çalıştığı türlerin tükenmediği gözlemlenmektedir. Buda her yeni iterasyonda güncellenen yeni kritik parametre

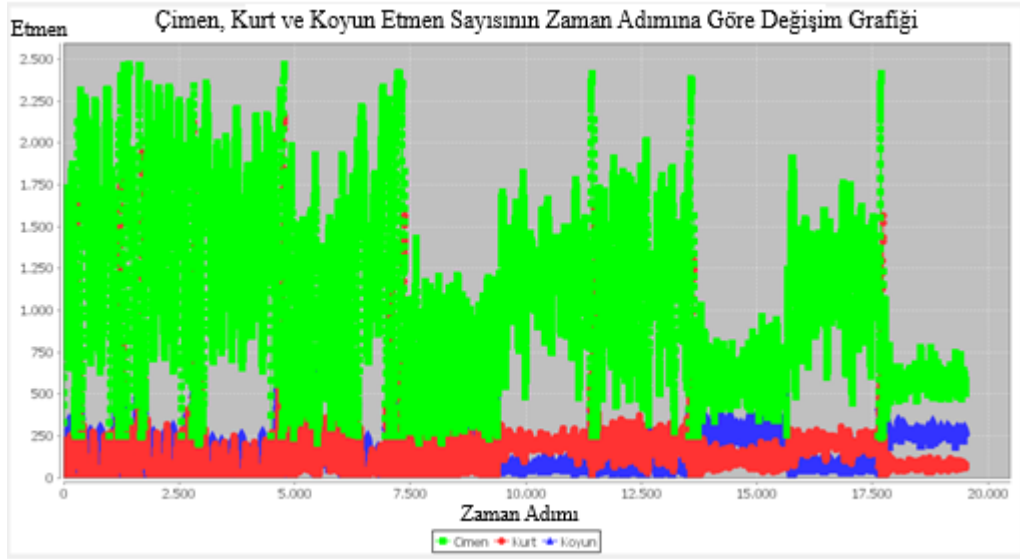
değerlerinin algoritmanın çözüm gücünü arttırdığını ve iyi parametre değerlerinin yani iyi çözüm kümelerinin bulunduğunu göstermektedir.



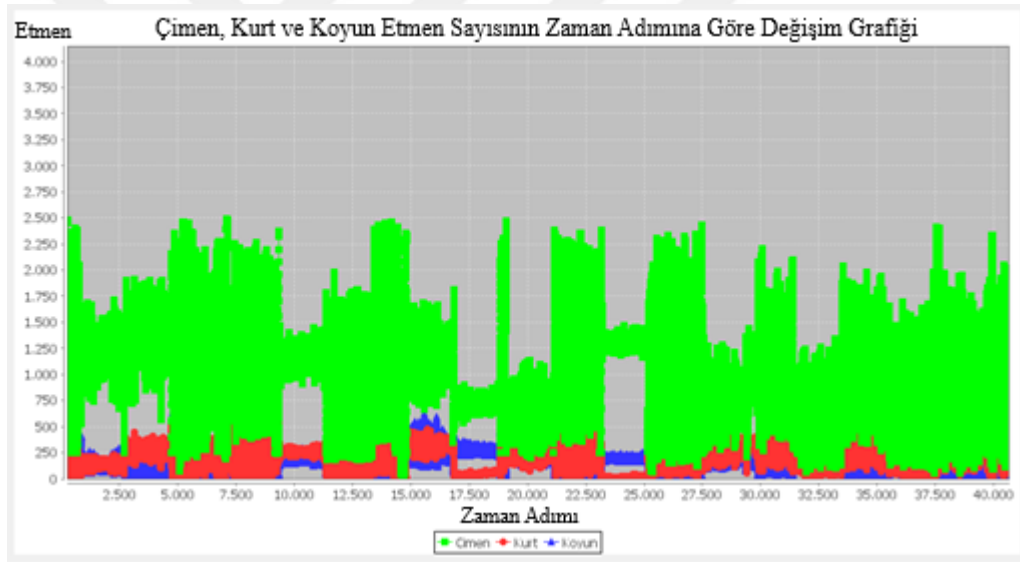
Şekil 8.10 AGA ile Av Avcı Modelinde Parametre Ayarlama



Şekil 8.11 APSO ile Av Avcı Modelinde Parametre Ayarlama



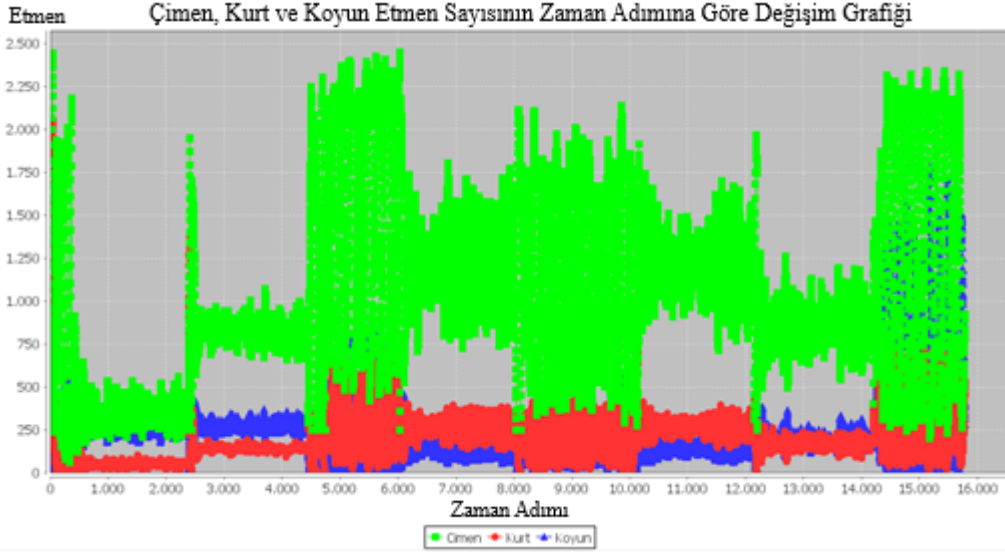
Şekil 8.12 AABC ile Av Avcı Modelinde Parametre Ayarlama



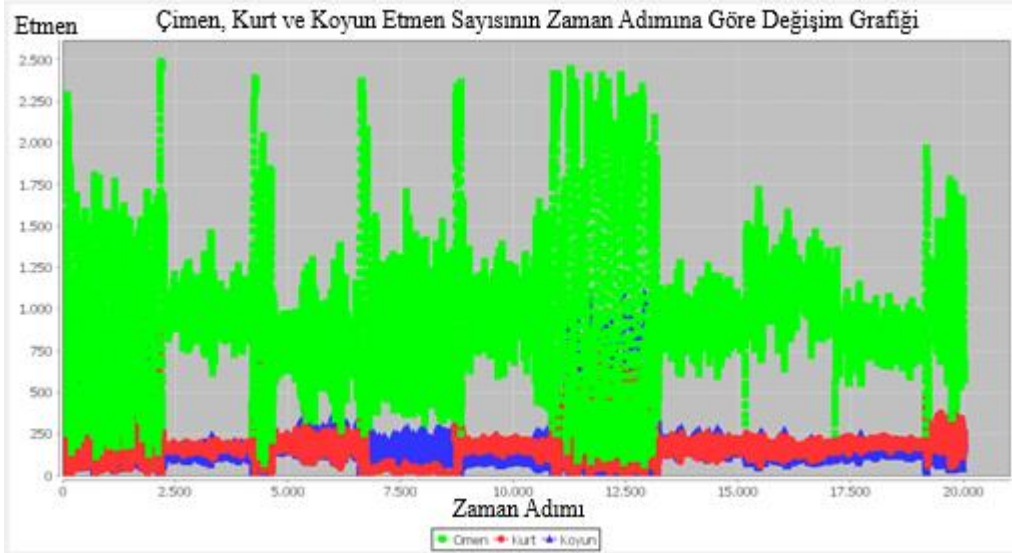
Şekil 8.13 AFA ile Av Avcı Modelinde Parametre Ayarlama

AGA, APSO, AABC ve AFA algoritmalarından elde edilen kritik parametre değerlerinin analizi sonucu en iyi çözüme ulaştıran kritik parametre değerleri bulunmuştur. Bu değerlerin GA, PSO, ABC ve FA algoritmalarına manuel atanması ile elde edilen Sonra AGA, Sonra APSO, Sonra AABC ve Sonra AFA algoritmaları kullanılarak yine av-avcı model parametre ayarlama işlemi yapılmıştır. Parametre ayarlaması yapıldığında elde edilen grafikler sırası ile Şekil 8.14, Şekil 8.15, Şekil 8.16 ve Şekil 8.17'de gösterilmiştir. Grafikler incelendiğinde ilk iterasyonlardan itibaren türlerin tükenmediği aralıkları daha sık görülmektedir. Her iterasyonun yani her bir benzetimin çalışma süresi 2000 adım sayısıdır. Grafikler incelendiğinde

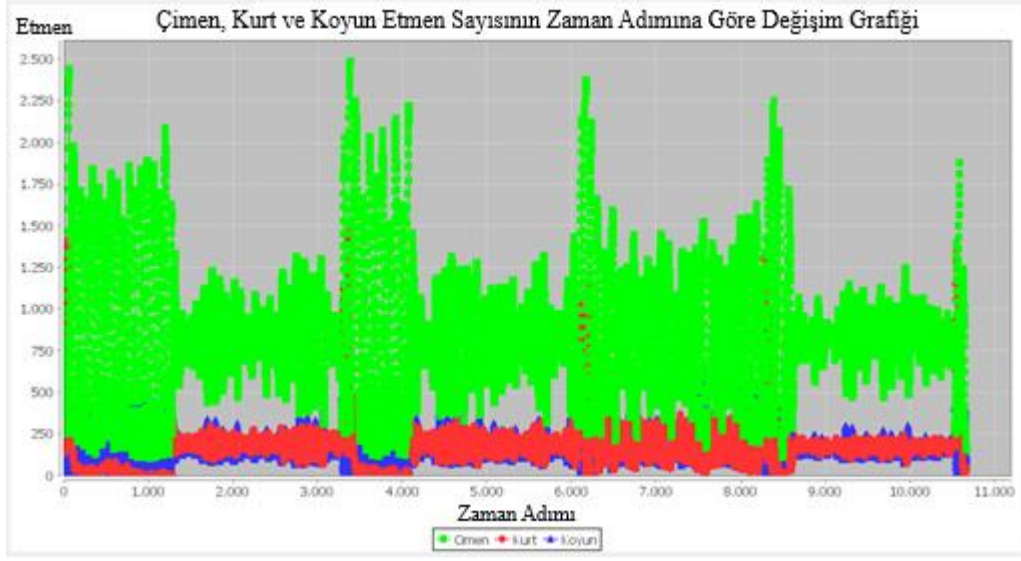
2000 adım sayısı boyunca tükenmeyen benzetimlerin olduğu görülmektedir. Bu da ilk iterasyonlardan itibaren iyi parametre değerlerinin bulunduğunu göstermektedir. Buda istenen çözümlere daha hızlı ulaşıldığını ve daha az iterasyon sayısı ile istenilen sonucun elde edilebileceğini göstermektedir.



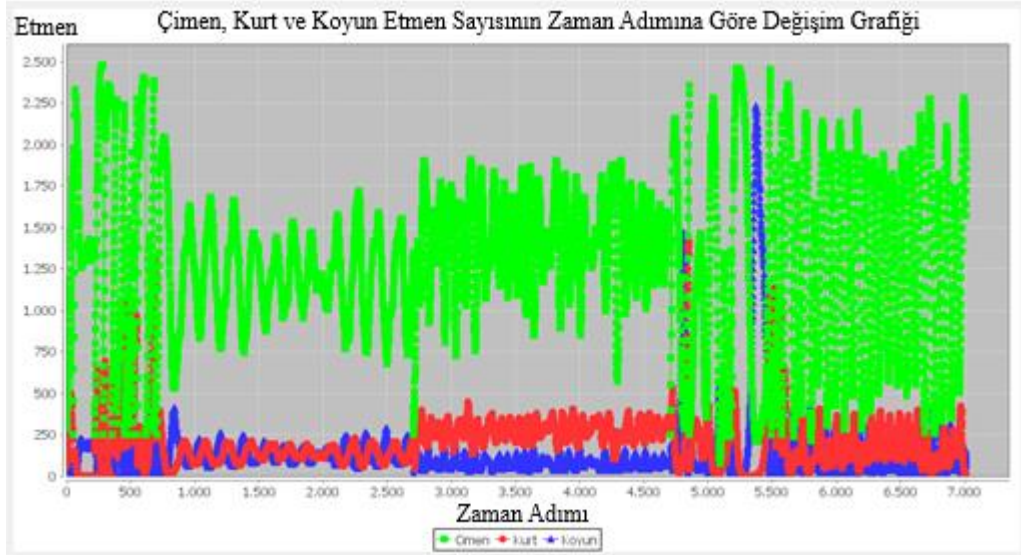
Şekil 8.14 Sonra AGA ile Av Avcı Modelinde Parametre Ayarlama



Şekil 8.15 Sonra APSO ile Av Avcı Modelinde Parametre Ayarlama



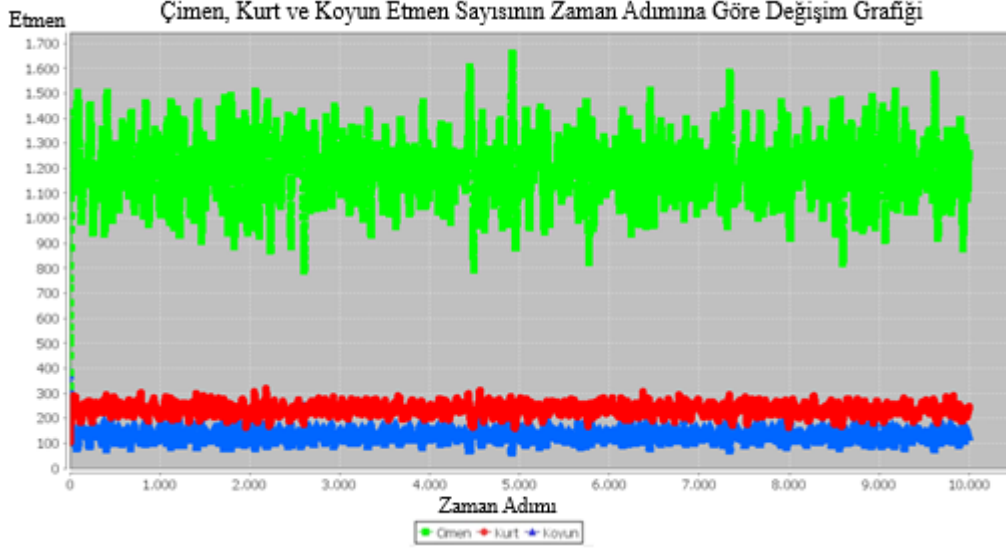
Şekil 8.16 Sonra AABC ile Av Avcı Modelinde Parametre Ayarlama



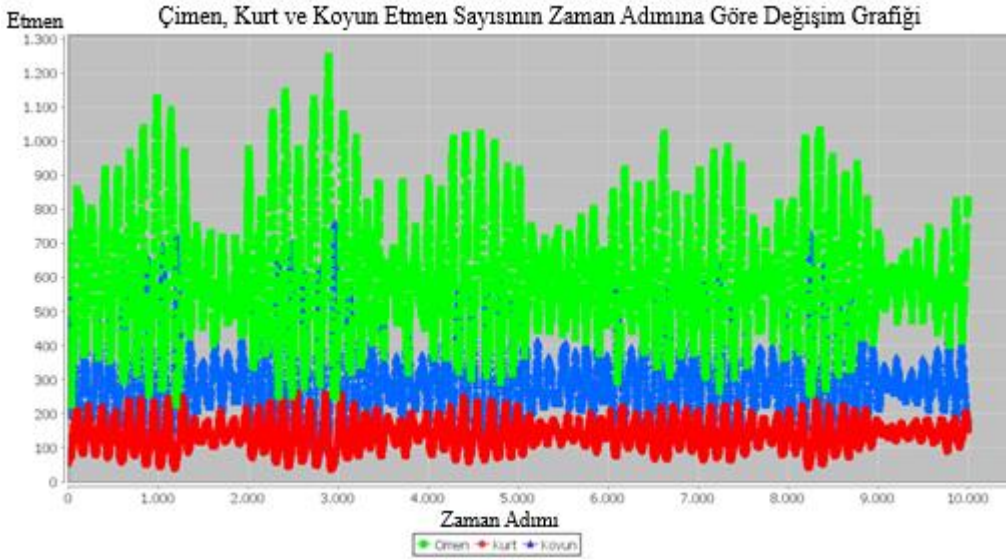
Şekil 8.17 Sonra AAFA ile Av Avcı Modelinde Parametre Ayarlama

Tüm algoritmaların başarısı birbirinden farklı olmasına rağmen tüm AGA, APSO, AABC ve AFA algoritmaları ile bulunan en uygun kritik parametre değerlerini kullanan Sonra AGA, Sonra APSO, Sonra AABC ve Sonra AFA algoritmaları av avcı problemi için diğer algoritmalarından daha iyi çözüm kümeleri (parametre setleri) bulmuştur. Yapılan analizler sonucu bu 4 algoritmanın buldukları en iyi parametre değerleri kullanılarak benzetim 10000 ve 20000 adım sayısı boyunca çalıştırıldığında Şekil 8.18, Şekil 8.19, Şekil 8.20 ve Şekil 8.21'de görünen grafikler elde edilmiştir. Grafik sonuçlarından 4 algoritmanın bulunduğu çözüm kümeleri ile çalıştırılan benzetimde türlerin tükenmediği

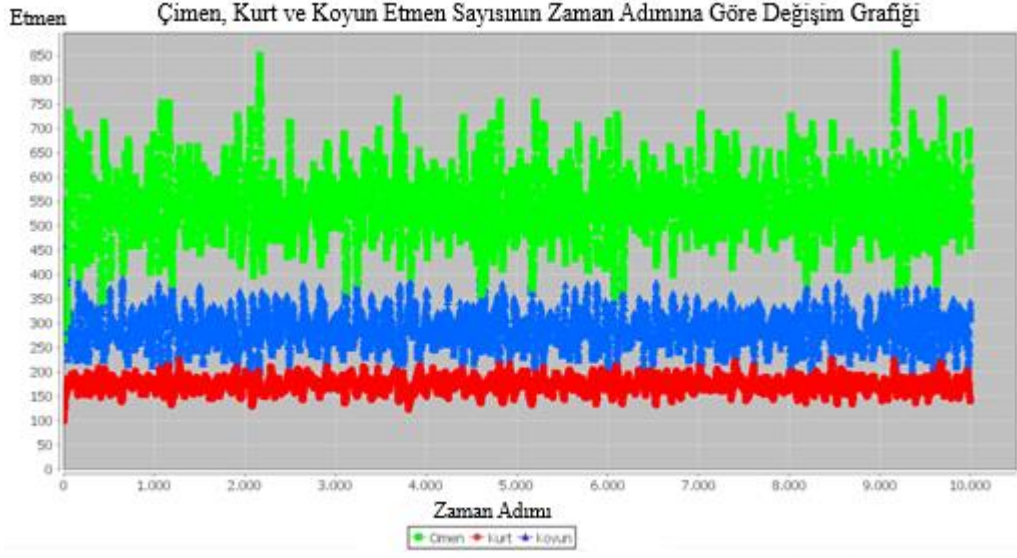
popülasyonun devamlılığının sağlandığı gözlemlenmektedir. Böylece hedeflenen parametre değerleri 4 algoritmanında bulunduğu grafik sonuçlarından görülmektedir.



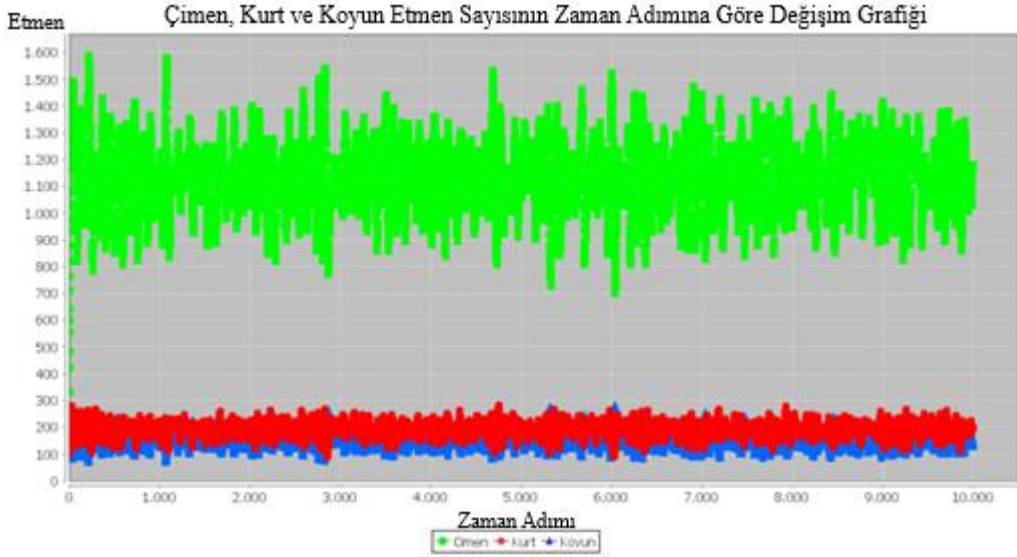
Şekil 8.18 Sonra AGA ile Elde Edilen En İyi Av-Avcı Model Parametre Değerlerinin Av-Avcı Modeline Manuel Girilerek Modelin Çalıştırılması Sonucu Elde edilen Grafik



Şekil 8.19 Sonra APSO ile Elde Edilen En İyi Av-Avcı Model Parametre Değerlerinin Av-Avcı Modeline Manuel Girilerek Modelin Çalıştırılması Sonucu Elde edilen Grafik



Şekil 8.20 Sonra AABC ile Elde Edilen En İyi Av-Avcı Model Parametre Değerlerinin Av-Avcı Modeline Manuel Girilerek Modelin Çalıştırılması Sonucu Elde edilen Grafik

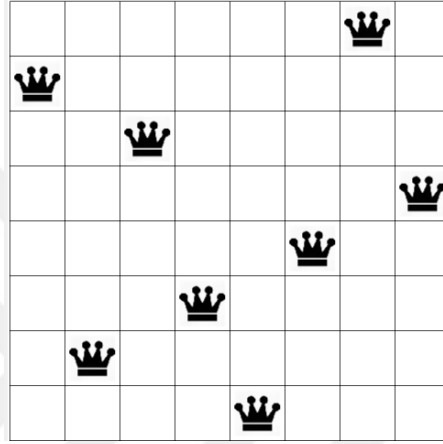


Şekil 8.21 Sonra AFA ile Elde Edilen En İyi Av-Avcı Model Parametre Değerlerinin Av-Avcı Modeline Manuel Girilerek Modelin Çalıştırılması Sonucu Elde edilen Grafik

### 8.3. Sekiz Vezir (Eight Queens) Modeli

Şekil 8.22'de de görüldüğü gibi bu model 8 x 8'lik bir satranç tahtasına birbirini almayacak şekilde sekiz vezirin yerleştirilmesi problemi. Her vezir konumunu diğer vezirler tarafından tehdit altında olmayacak şekilde belirlemelidir. Bunun için bir vezir diğer bir vezirle aynı satırda, aynı sütunda ya da aynı köşegende olmamalıdır. Bu problem ilk olarak 1848 yılında satranç oyuncusu Max Bezzel tarafından bulunmuştur. Sonraki yıllarda Gauss ve Georg Cantor gibi birçok

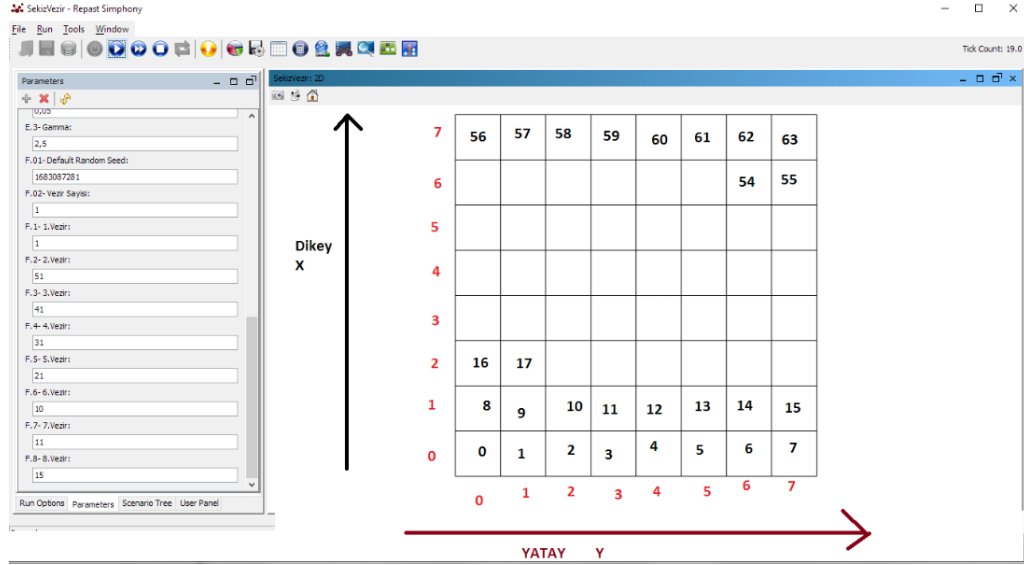
matematikçi çözümü üzerinde çalışmışlardır. Franz Nauck 1950 yılında ilk çözümü yapmıştır. Franz Nauck aynı bu problemi NxN boyutundaki bir tahta üzerinde N vezirin yerleştirilmesi olarak problemi değiştirmiştir. Problemin çözümü toplamda  $(8 \times 8)! / ((8 \times 8 - 8)! \times 8!) = 4.426.165.368$  olasılıkta sadece 92 çözüm mevcuttur. Bundan dolayı bu problemin çözümü çok sayıda hesaplamayı gerektirir. Bazı kısıtlamalar veya kısa yolların kullanılması hesaplama sayısını azaltacaktır. Her satırda veya sütunda bir vezirin bulunması koşulunda çözüm sayısı 88'e düşmektedir.



Şekil 8.22 Sekiz Vezir Problemi İçin Örnek Çözüm Gösterimi

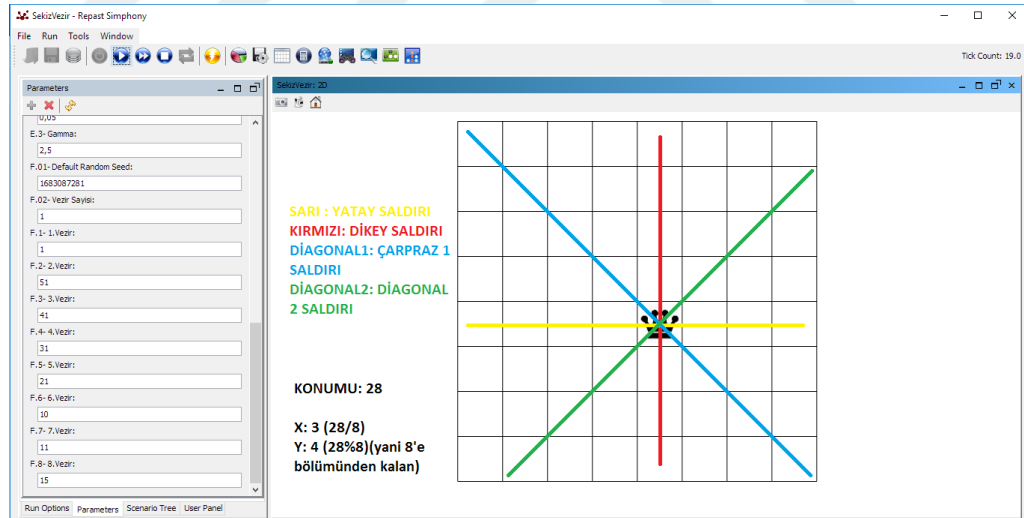
Bu modelin toplam 8 tane parametresi vardır. Bu parametrelerin her biri vezirin satranç tahtası üzerindeki konum değerini tutar. Her bir vezir, satranç tahtası üzerinde 64 farklı şekilde konumlanabilir. Geliştirilen bu yaklaşımla modeldeki 8 vezirin çakışmadan satranç tahtası üzerinde konumlanması sağlanmıştır.

Şekil 8.23'de SekizVezir probleminde yer alan tahta ve tahtanın her bir hücrenin programda aldığı indis değerleri gösterilmiştir.



Şekil 8.23 Tahtanın x ve y koordinatları ve her bir karenin indis numarası

Şekil 8.24’de tahta üzerindeki vezir hareketleri gösterilmiştir yukarı, aşağı, sağa, sola hareket ettiği gibi çarpaz (diagonal) olarak toplam 15 adım hareket edebilmektedir.



Şekil 8.24 Vezir hareketleri

### 8.3.1. Sekiz Vezir Problemi İçin Uygunluk Fonksiyonu

Eşitlik 8.3.1 uygunluk fonksiyonu olarak tasarlanmıştır. Çarpışma sayısı tasarlanan uygunluk fonksiyonu için kritik değer olarak belirlenmiştir. Çünkü amaç çarpışma olmayacak şekilde tüm vezirlerin tahtaya yerleştirilmesidir. Her bir vezirin 4 çarpışma ihtimali var yatay, dikey, çarpraz\_1, çarpraz\_2. Her bir vezir için

4 çarpışma 8 vezir için toplam  $4*8=32$  çarpışma ihtimali vardır. Uygunluk değeri çarpışma sayısı ile ters orantılıdır çarpışma sayısının artması mevcut çözüm kümesinin uygunluk değerinin daha kötü olduğunu göstermektedir. Minimizasyon işlemi yapıldığı için 0-1 arasında değer alan uygunluk değeri için 0 en iyi değer 1 en kötü değer olarak Kabul edilmiştir.

$$\text{uygunluk} = \text{carpisma} / (\text{vezirSayi} * 4) \quad 8.3.1$$

32 çarpışma ihtimali olan bir çözüm kümesine uygunluk fonksiyonu uygulanırsa  $32/32=1$  sonucu elde edilir bu en kötü durumu ifade eder. Eğer mevcut çözüm kümesi için hiç çarpışma ihtimali yoksa  $0/32 = 0$  bu da en iyi çözüm olarak bulunur.

### 8.3.2. Sekiz Vezir Modeli Benzetim Sonuçları

#### Sonuçların Elde Edilmesi:

Kullanılan Random GA, Random PSO, Random ABC, Random FA, AGA, APSO, AABC, AFA, Sonra AGA, Sonra APSO, Sonra AABC ve Sonra AFA (Bkz. Çizelge 8.4) algoritmaları sekiz-vezir modeline ait parametreleri ayarlamak üzere ayrı ayrı sekiz-vezir modelini 10 kez çalıştırır. Elde edilen sonuçların ortalaması hesaplanarak algoritmalar birbirleri ile karşılaştırılır. Sekiz-vezir modeline ait parametreleri ayarlayan en başarılı algoritma bulunur. Elde edilen sonuçların ortalamaları Çizelge 8.8'de görülmektedir.

#### Sonuçların Analizi;

Çizelge 8.8'de GA algoritması incelendiğinde kritik parametreleri ayarlanmış Sonra AGA'nın(Bkz. Çizelge 8.4) başarılı olduğu çizelgedeki en iyi uygunluk değeri ve genel uygunluk değerlerine bakıldığında görülmektedir. Yaklaşık en iyi uygunluk değerleri kıyaslandığında en iyi uygunluk değerini bulan algoritmanın Sonra AGA olduğu görülmektedir  $GA(0,218) > AGA(0,125) > \text{Sonra AGA}(0,062)$ . İyileşme oranı oldukça büyüktür. Bitiş adım sayıları gözlemlendiğinde hepsinde aynı adım sayısında sonuçlandırıldığı görülmektedir. Bunun nedeni her bir benzetim çözümünün tek adım sayısı ile sonuçlanmasıdır. Her iterasyonda 10 çözüm kümesi işletildiği için toplam 1000 iterasyon çalıştırıldığında 10000 adım sayısı çalıştırılmış olacaktır. Bu nedenle 3 algortmada bitiş adım sayıları aynı olmasına rağmen bitiş süreleri birbirinden farklıdır. En uzun süren Random GA

algoritmasının üretmiş olduğu ebeveyn seçim yüzdesi %1 ya da %10 gibi değerlerin çıkması bu algortmada daha çok çözüm kümesinin üretilip işletilmesi anlamına gelmektedir. AGA algoritmasında uyarlama işlemi yani kritik parametre değerlerinin ayarlanması işlemi yapıldığından işlem yükünün fazla olması süreyi arttırmıştır. %30 ebeveyn seçim yüzdesine sahip olan ve daha iyi kritik parametre değerleri ile çalıştırılan Sonra GA algoritması daha iyi performansa sahiptir. Genel olarak tüm algoritmaların parametre ayarlama sürelerine bakıldığında en iyi çözümü ABC algoritmasından sonra GA verdiği gözlemlenmiştir. ABC algoritmasının daha hızlı olmasının nedenlerinden biri en iyi çözümü maksimum iterasyon sayısına ulaşmadan vermesidir.

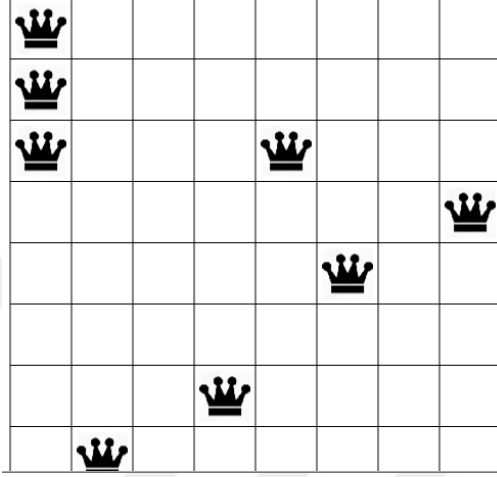
Bu problemin çözümünde başarılı olan hatta optimum çözüme ulaşmayı başaran algoritma ABC algoritmasıdır. Random ABC algoritmasının en iyi uygunluk değerine bakıldığında AABC'nin ve Sonra AABC'nin çok büyük bir iyileşme gösterdiği hatta belirlenen iterasyon sayısına ulaşmadan optimum sonucu bulduğu görülmektedir. Kısa sürede çözüme ulaşması diğer önemli avantajıdır. Av Avcı probleminde çok iyi çözüm üretemeyen ABC algoritmasının bu problemde çok daha başarılı olması her model için farklı çözümlerin olması gerekliliğini göstermektedir.

Bu problemin çözümünde başarılı olmayan PSO ve FA algoritmasına baktığımızda iyileşmenin olduğu gözlemlenmektedir fakat bu iyileşmenin bu problemin sonucunu bulmada yeterli olmadığı elde edilen sonuçlar ve şekillerden de anlaşılmaktadır. FA algoritması av avcı modelinde en iyi sonucu üreten algoritma olarak bulunmuştu. Bu durum problem bazlı çözümün önemini ortaya koymaktadır.

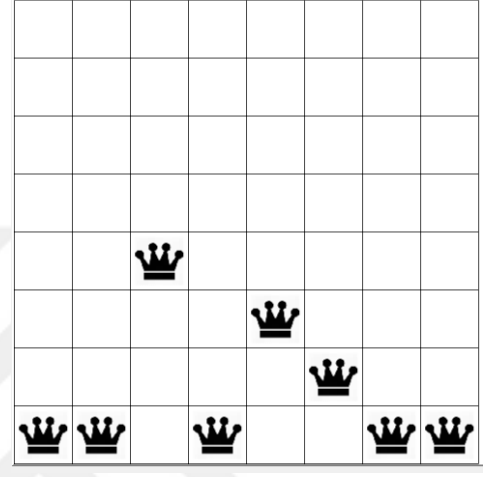
Çizelge 8.8 Etmen Tabanlı Model ve Benzetimde Sekiz Vezir probleminde parametrelerinin ayarlama işleminde kullanılan algoritmalarından elde edilen sonuçlar

Algoritma	Adım Sayısı	Geçen Saniye	Başlangıç En İyi Uygunluk	En İyi Uygunluk	İyileştirme Farkı	Ortalama Uygunluk	Ortalama Yerel Uygunluk	Ortalama Genel Uygunluk	İyileştirme Sayısı
<b>Random GA</b>	10000	2402.558	0,43750000	0,21875000	0,21875000	0,50760549	0,40319337	0,25503125	7,00
<b>AGA</b>	10000	2378.669	0,37500000	0,12500000	0,25000000	0,46661271	0,57147773	0,25190816	3,00
<b>Sonra AGA</b>	10000	2266.425	0,46875000	0,06250000	0,40625000	0,34515923	0,30334596	0,12769019	8,00
<b>Random PSO</b>	10000	3445.72	0,34375000	0,34375000	0,00000000	0,78354352	0,56364676	0,34375000	1,00
<b>APSO</b>	10000	3082.379	0,43750000	0,25000000	0,18750000	0,71573468	0,57147773	0,25190816	3,00
<b>Sonra APSO</b>	10000	2396.854	0,31250000	0,28125000	0,03125000	0,60929532	0,45021197	0,28487863	2,00
<b>Random ABC</b>	9995	2503.299	0,37500000	0,12500000	0,25000000	0,40526836	0,30954043	0,16381250	8,00
<b>AABC</b>	8825	2152.976	0,37500000	0,00000000	0,37500000	0,41989576	0,17300879	0,13265306	8,00
<b>Sonra AABC</b>	7965	1529.984	0,31250000	0,00000000	0,31250000	0,31036593	0,21779245	0,06271897	5,00
<b>Random FA</b>	10000	3221.986	0,34375000	0,28125000	0,06250000	0,73477035	0,54297893	0,28243750	2,00
<b>AFA</b>	10000	3528.346	0,40625000	0,25000000	0,15625000	0,81537784	0,65881507	0,25043794	2,00
<b>Sonra AFA</b>	10000	2693.19	0,34375000	0,12500000	0,21875000	0,47889274	0,33509135	0,14753996	5,00

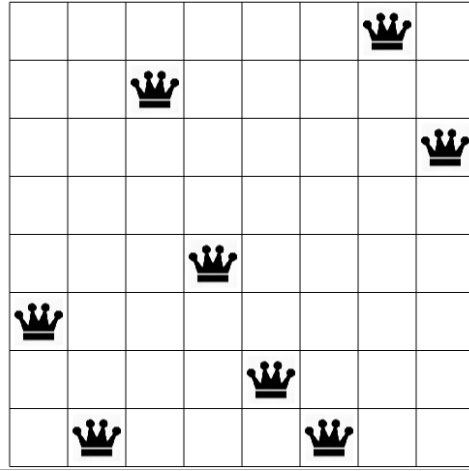
Şekil 8.25, Şekil 8.26, Şekil 8.27 ve Şekil 8.28 Sırası ile Random GA, Random PSO, Random ABC ve Random FA algoritmalarının rasgele atanan kritik parametre değerlerini kullanarak çalıştırılan bu algoritmaların ekran çıktısını göstermektedir. Amaç hiçbir vezirin birbirini tehdit etmeyecek şekilde tahtaya sıralanmasıdır. Bu şekiller incelendiğinde hiç birinde çözümün bulunmadığı görülmektedir.



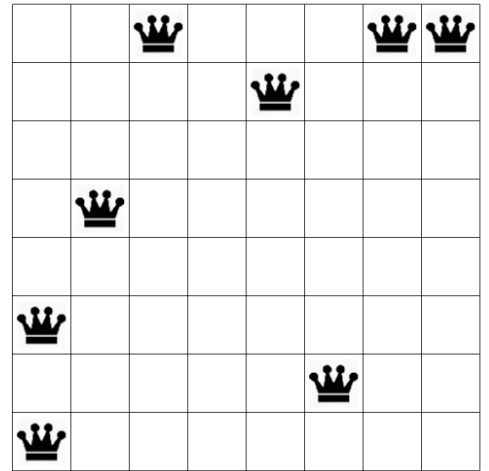
Şekil 8.25 Random GA ile Sekiz Vezir Modelinde Parametre Ayarlama



Şekil 8.26 Random PSO ile Sekiz Vezir Modelinde Parametre Ayarlama



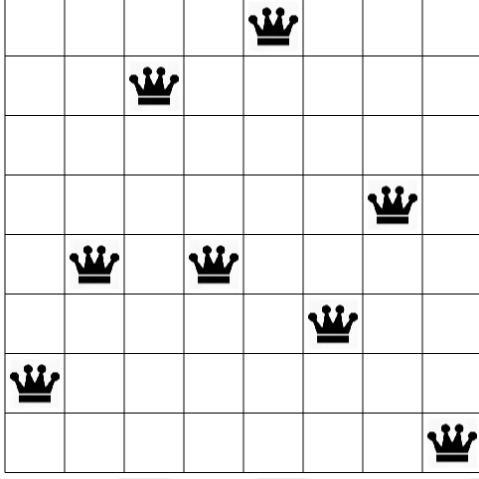
Şekil 8.27 Random ABC ile Sekiz Vezir Modelinde Parametre Ayarlama



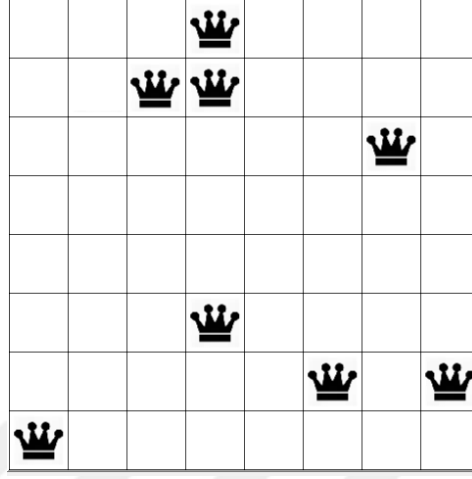
Şekil 8.28 Random FA ile Sekiz Vezir Modelinde Parametre Ayarlama

Şekil 8.29, Şekil 8.30, Şekil 8.31 ve Şekil 8.32 sırası ile AGA, APSO, AABC ve AFA algoritmalarının çalıştırılması sonucu elde edilen şekillerdir. Şekiller incelendiğinde uyarlanabilir çevrimiçi kritik parametre ayarlama sürecinde iyi çözümlerin bulunduğu sonucu çıkarılabilir. Bunun Sonucunda AABC ve AGA

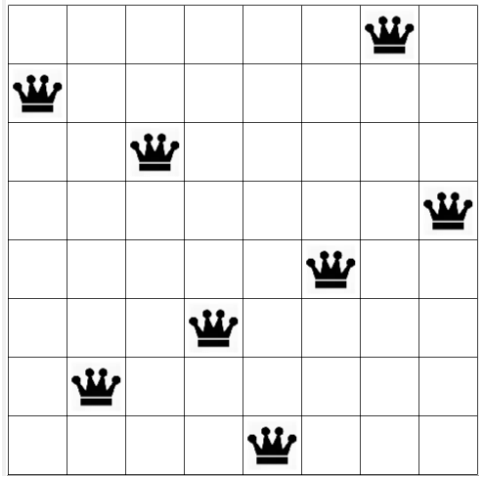
algoritmanın istenen amaç doğrultusunda sekiz vezirin dizilimini sağlarken APSO ve AFA algoritmalarının yine çözümü bulamadığı görülmektedir.



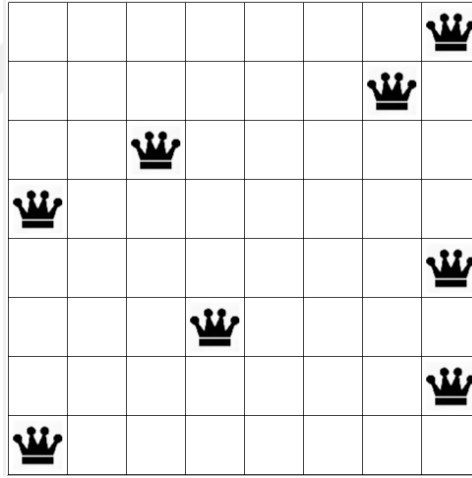
Şekil 8.29 AGA ile Sekiz Vezir Modelinde Parametre Ayarlama



Şekil 8.30 APSO ile Sekiz Vezir Modelinde Parametre Ayarlama

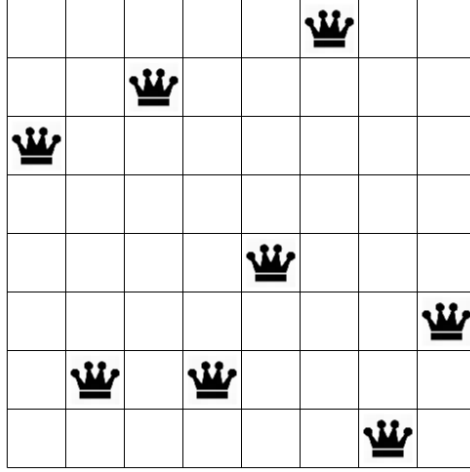


Şekil 8.31 AABC ile Sekiz Vezir Modelinde Parametre Ayarlama

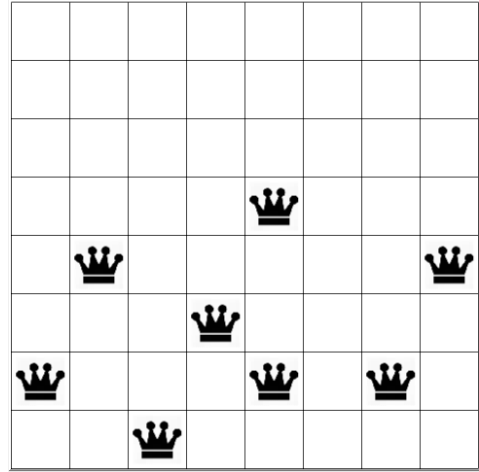


Şekil 8.32 AFA ile Sekiz Vezir Modelinde Parametre Ayarlama

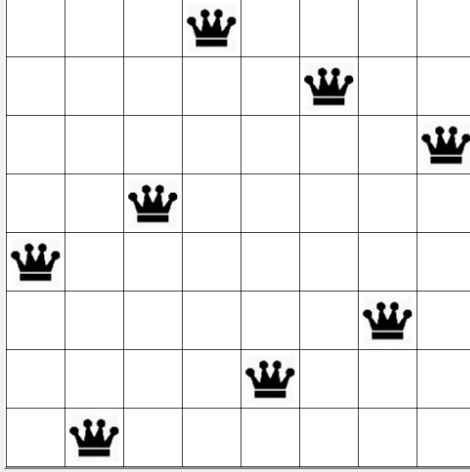
Şekil 8.33, Şekil 8.34, Şekil 8.35 ve Şekil 8.36 sırası ile Sonra AGA, Sonra APSO, Sonra AABC ve Sonra AFA algoritmalarının optimizasyon sonucunu göstermektedir. En iyi kritik parametre değerlerine sahip bu algoritmanın en iyi uygunluk değerlerine yani en iyi çözüm kümelerine sahip olduğu Çizelge 8.8’de incelenmişti. Bu şekillerde de görüldüğü gibi çözüm Sonra GA ve Sonra ABC tarafından bulunmuştur.



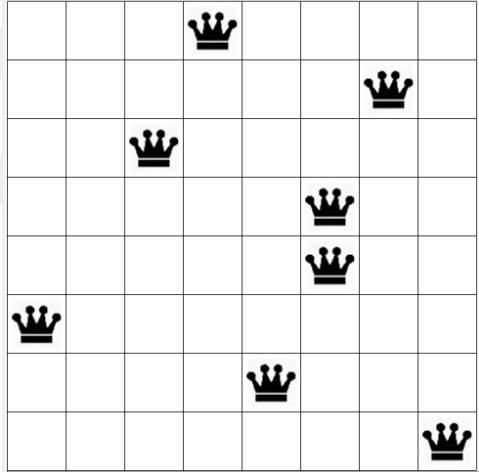
Şekil 8.33 Sonra AGA ile Sekiz Vezir Modelinde Parametre Ayarlama



Şekil 8.34 Sonra APSO ile Sekiz Vezir Modelinde Parametre Ayarlama



Şekil 8.35 Sonra AABC ile Sekiz Vezir Modelinde Parametre Ayarlama



Şekil 8.36 Sonra AFA ile Sekiz Vezir Modelinde Parametre Ayarlama

### Literatürdeki Sekiz Vezir Problem Çözümlerinde Kullanılan Algoritmaların Karşılaştırılması;

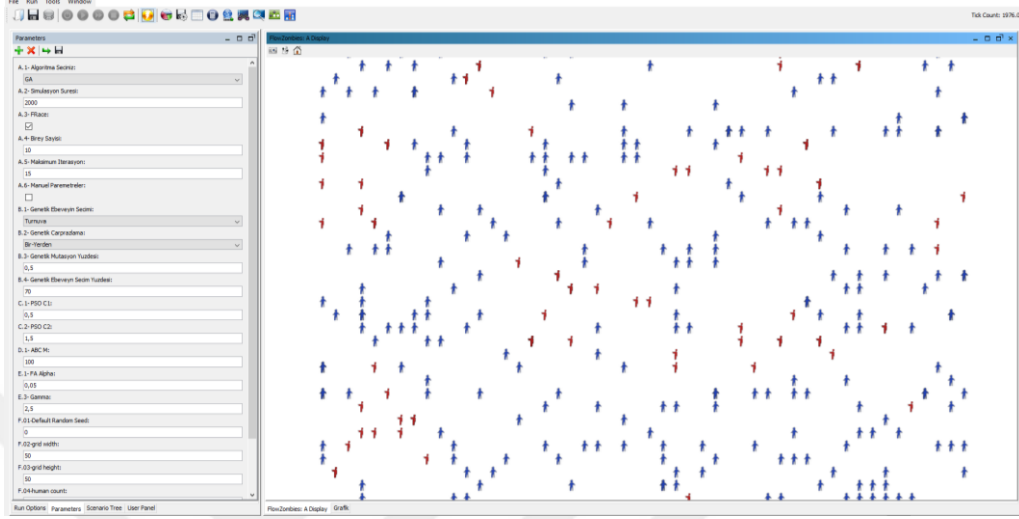
Sekiz Vezir Modeli RePast ortamında modellenen etmen tabanlı bir model olarak bu çalışmada ele alınmıştır. Her bir vezir bir etmen olarak oluşturulur ve her bir vezirin konumu bir parametre olarak belirlenir 8 vezirin konumu 8 elemanlı çözüm kümesini oluşturmaktadır. Bu model bu tez çalışmasında kullanılmış olup vezir etmenlerin 64'lük tahta üzerinde birbirlerini tehdit etmeyecek şekilde konumlanmaları amaçlanmaktadır. Bu çalışmada amaç kullanılan MSA algoritmaları ile geliştirilen Uyarlanabilir MSA algoritmalarının bir etmen tabanlı model olarak oluşturulan Sekiz Vezir problemine çözüm sunmaktır.

Literatürde sekiz vezir problemine çözüm üretmek amacı ile kullanılan birçok algoritma bulunmaktadır (Martinjak, 2007; Masehian, 2014; Masehian, 2013). Bu tez çalışmasında kullanılan MSA'lerden en başarılı olan Sonra AABC ve Sonra AGA algoritmaları literatürde yer alan ve sekiz vezir problemini çözmek için kullanılan Benzetilmiş Tavlama (Simulation Annealing), Yasaklı Arama (Tabu Search) (Martinjak, 2007), ICA (Imperialist Competitive Algorithm) (Masehian, 2014), Yerel Arama (Local Search), Dağıtık Arama (Scatter Search) (Masehian, 2013) algoritmaları ile karşılaştırılmıştır. Tez çalışmasında geliştirilen Sonra AGA ve Sonra ABC algoritmaları yukarıda adı belirtilen algoritmalarından çok daha az iterasyon sayısı ile çözüme ulaştığı elde edilen sonuçlardandır. Yukarıda adı belirtilen algoritmalar 100 popülasyon boyutu kullanılarak gerçekleştirilmiştir. Bu tez çalışmasında MSA'lar için popülasyon boyutu 10 olarak seçilmesi iterasyon sayısını arttırmıştır. Popülasyon boyutu 100 seçildiğinde daha küçük iterasyon sayıları ile çözüme ulaşılmıştır. Sonra GA algoritması popülasyon boyutu 100 olarak alındığında minimum 3 iterasyonda en uygun sonucu bulmuştur. Sonra AABC algoritması yine 100 popülasyon boyutu ile minimum 10 iterasyonda sonuca ulaşmıştır. Literatürde yer alan Benzetimli Tavlama, minimum 66, ICA minimum 17, Yerel Arama algoritması ise minimum 18, Dağıtık Arama 44 iterasyon ile en uygun sonucu bulmuşlardır. Çalışma süresi bakımından yapılan karşılaştırmada Sonra AGA ve Sonra AABC'nin çok daha az iterasyon sayısı ile çözüme ulaşmasına rağmen daha uzun sürede sonuca ulaştığı gözlemlenmiştir. Bunun nedeni sekiz vezir modelinin etmen tabanlı ortam kullanılarak modellenmesi ve çok sayıda etmenin işleyişi diğer benzetim ve modellere göre benzetim süresini uzatmaktadır. Ayrıca kullanılan sistem özellikleri ve kullanılan araç çalışma süresinde farklılıklara neden olmaktadır.

#### **8.4. Zombi-İnsan (Human Zombie) Modeli**

Flow Zombies benzetim modelinde üç parametre bulunmaktadır; zombi sayısı, insan sayısı ve insan enerjisi. Bu modelde bilindiği üzere ortamda var olan zombiler karşılaştığı her insan etmenini zombi etmene dönüştürür. Model içinde insan etmen zombilerin bulunmadığı yerlere doğru hareket ederler, enerjileri tükenince durup dinlenirler ve enerji toplarlar. Bu bekleme süresinde zombi ile karşılaşma ihtimali artar. Zombiler insanların yoğun olduğu alanlara doğru hareket ederler aynı kordinatta bulunduğu insanları zombiye dönüştürürler. Şekil 8.37, insan zombi modeline ait arayüzü göstermektedir. Şekilde yer alan kırmızı renkli etmenler zombi etmen mavi renkli etmenler insan etmen olarak sembolize edilmiştir. Bu modelde amaç, parametre değerlerini insan etmenlerin yaşam

süresini arttırmayı hedefleyerek ayarlamaya çalışmaktır. Modelde yer alan insan, zombi ve insan enerjileri optimize edilerek insan neslinin mümkün olduğunca devam etmesi hedeflenmiştir. Sonuçlarda sürenin uzadığı gözlemlenmiştir.



Şekil 8.37 İnsan Zombi Modeli Arayüzü

#### 8.4.1. Zombi Problemi İçin Uygunluk Fonksiyonu

İnsan Zombi modelinde amaç insan neslinin mümkün olduğunca tükenmesini önlemektir. Bunun için çözüm kümesine ait uygunluk değeri benzetimin çalışma süresi ile orantılıdır. Çalışma süresini maksimize eden çözüm kümeleri bu problem için en iyi çözüm kümeleridir. Bundan yola çıkarak 0-1 arasında belirlediğimiz uygunluk değeri 0'a yaklaştıkça daha iyi çözüm kümesine ait uygunluk değeri olduğu anlamına gelir. İnsan neslinin tükenmesi benzetimi sonlandırmaktadır. Amaç insan neslini mümkün olduğunca uzun yaşamasını sağlamaktır. Bundan dolayı benzetimin zaman adımı( $gencenT$ ) dikkate alınarak uygunluk hesaplanmıştır. Uygunluk değeri 0-1 arasında alınmış olup 0 en iyi uygunluk değerini temsil etmektedir. Minimizasyon işlemi yapılmıştır. Sürenin uzaması uygunluk değerinin( $F$ ) küçülmesi ve daha iyi çözüm kümesinin bulunduğu anlamına gelmektedir. Eğer ilk zaman adımında benzetim sonlanırsa  $gencenT=1$  olacaktır ve  $F=1/1=1$  olacağından en kötü uygunluk değeri elde edilmiş olacaktır. Çalışma süresinin artması insan neslinin devam ettiği anlamına gelmekte ve  $F$  değerinin 0'a yaklaştığını göstermektedir. Bundan dolayı uygunluk değerini hesaplayan uygunluk fonksiyonu Eşitlik 8.4.1 olarak belirlenmiştir.

$$F = 1 / \text{gecenT}$$

8.4.1

F değeri Uygunluk fonksiyonunu, gecenT, geçen süreyi temsil etmektedir. F değeri 0-1 arasında değerler almakta olup 0 en iyi çözüm kümesinin bulunduğu uygunluk değeri anlamına gelmektedir. Sürenin uzaması daha küçük uygunluk değerinin bulunmasına buda daha iyi parametre değerlerinin elde edildiği anlamına gelmektedir.

### 8.4.2. Zombi-İnsan (Human Zombie) Modeli Benzetim Sonuçları

#### Sonuçların Toplanması;

Kullanılan Random GA, Random PSO, Random ABC, Random FA, AGA, APSO, AABC, AFA, Sonra AGA, Sonra APSO, Sonra AABC ve Sonra AFA algoritmaları insan-zombi modeline ait parametreleri ayarlamak üzere insan-zombi modelini 10 kez çalıştırır. Elde edilen sonuçların ortalaması hesaplanarak algoritmalar birbirleri ile karşılaştırılır. İnsan-zombi modeline ait parametreleri ayarlayan en başarılı algoritma bulunur. Elde edilen sonuçların ortalamaları Çizelge 8.9'da görülmektedir.

#### Sonuçların Analizi;

Çizelge 8.9 İncelendiğinde Sonra GA algoritmasının yani kritik parametre değerleri ayarlanan algoritmanın Random GA yani rasgele kritik parametre değeri alan algoritmaya göre en iyi uygunluk değerlerine bakıldığında daha başarılı olduğu görülmektedir. Geçen süre ve adım sayılarına bakıldığında Sonra GA'nın daha uzun sürmesinin nedeni hedef benzetimde insan neslinin uzun yaşamasını sağlayan parametre değerlerini bulmaya çalışmaktır. Bu durumda Sonra GA algoritması daha iyi uygunluk değerleri ürettiği için benzetim süresini uzatmıştır. Geliştirme sayısı da Sonra GA'nın AGA ve Random GA'dan daha iyi olduğunu göstermektedir.

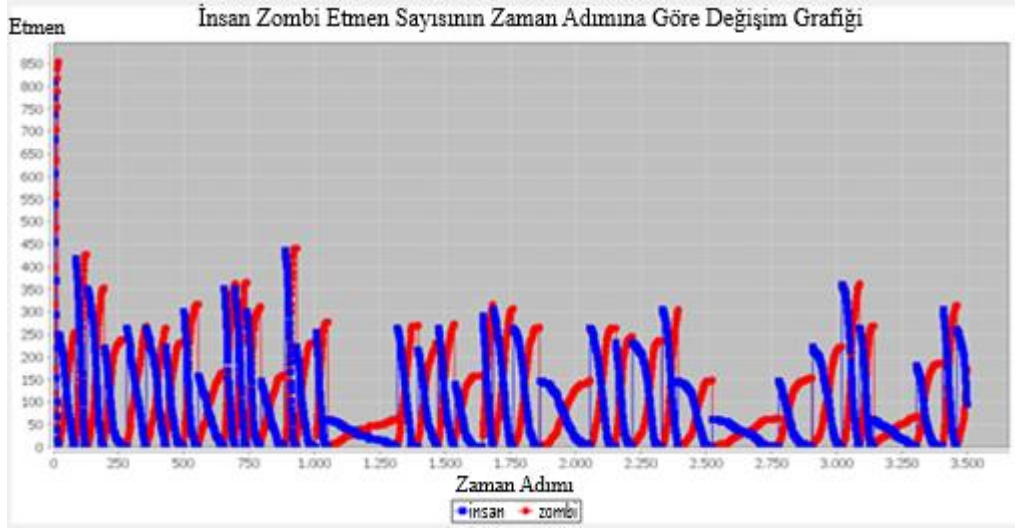
Bu algoritma için PSO, ABC ve FA algoritmaları incelendiğinde GA algoritmasına benzer durumlar gözlemlenmektedir. Tüm algoritmaların bu problem için optimizasyon işlemlerinin yapıldığı fakat PSO algoritmasının iyileştirme oranının çok daha iyi olduğu gözlemlenmektedir. ABC algoritmasının ürettiği uygunluk değeri PSO değerinden daha iyi olmasının nedeni başlangıç uygunluk değerinin de daha iyi olmasıdır. Bu problemde Av-avcı modelinin aksine

FA algoritmasının ürettiği sonuçlar bu problemi çözmeye daha başarısız olduğu gözlemlenmiştir. Bu durum, yapılan bu yaklaşımın problem bazlı çözümler ürettiğini kanıtlamaktadır.

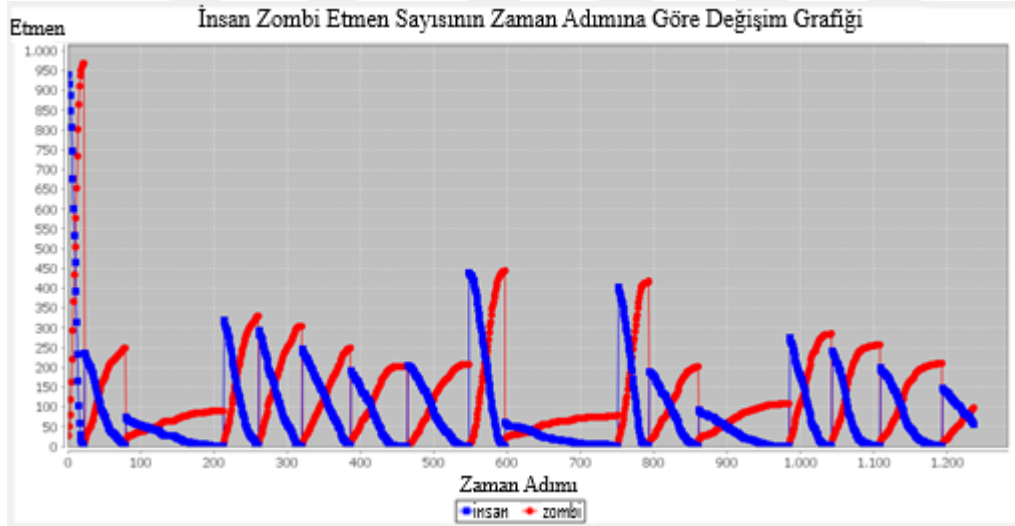
Çizelge 8.9 Etmen Tabanlı Model ve Benzetimde İnsan-Zombi probleminde parametrelerinin ayarlama işleminde kullanılan algoritmalarından elde edilen sonuçlar

Algoritmalar	Adım Sayısı	Geyen Saniye	Başlangıç En İyi Uygunluk	En İyi Uygunluk	İyileştirme Farkı	Ortalama Uygunluk	Ortalama Yerel Uygunluk	Ortalama Genel Uygunluk	İyileştirme Sayısı
Random GA	2200.0	59.571	0,12345679	0,05434783	0,06910896	0,17457331	0,13691953	0,06224638	2,00
AGA	11010.0	140.192	0,07874016	0,02695418	0,05178598	0,10207962	0,04298610	0,03509676	4,00
Sonra GA	17488.0	203.832	0,10101010	0,02369668	0,07731342	0,10324921	0,04536725	0,03859665	4,00
Random PSO	37789.0	126.586	0,07246377	0,03906250	0,03340127	0,21868024	0,07425079	0,06399457	2,00
APSO	48931.0	157.761	0,06493506	0,01724138	0,05974115	0,05974115	0,02943538	0,02550978	6,00
Sonra PSO	51357.0	178.471	0,11111111	0,01569859	0,09541252	0,07137106	0,03445588	0,03282884	11,00
Random ABC	23794.0	139.554	0,02777778	0,02096436	0,00681342	0,06359051	0,02680890	0,02412526	5,00
AABC	40488.0	193.304	0,13513514	0,02331002	0,11182511	0,09564868	0,04753137	0,04753137	7,00
Sonra ABC	44021.0	171.499	0,07092199	0,01492537	0,05599661	0,06441630	0,02815037	0,02607466	8,00
Random FA	10949.0	329.834	0,09708738	0,09009009	0,00699729	0,19464096	0,11983896	0,09708738	1,00
AFA	20705.0	255.681	0,11494253	0,02277904	0,09216349	0,15670723	0,04795208	0,03360096	5,00
Sonra FA	24006.0	304.526	0,06211180	0,01639344	0,04571836	0,14303479	0,03738350	0,02116642	4,00

Şekil 8.38, Şekil 8.39, Şekil 8.40, Şekil 8.41 de İnsan Zombi modeli için Sonra GA, Sonra PSO, Sonra FA algoritmalarının parametre ayarlama işlemleri gözlemlenmektedir. Tüm algoritmaların iyi parametre değerleri ürettiğini benzetim süresinin uzadığı aralıklardan anlaşılmaktadır. Zombi etmen Şekil 8.38, 8.39, 8.40, 8.41'deki grafiklerde kırmızı insan etmen, zombi etmen ise mavi renkle temsil edilmektedir. Etmen sayıları ve zaman adımlarına grafik üzerinden bakıldığında bazı aralıklarda insan etmen sayısının tükenme süresinin uzadığı görülmektedir. Grafik üzerinde görünen uzun zaman adımına sahip aralıklar daha iyi çözüm kümeleri ile modelin çalıştırıldığını göstermektedir. İnsan-zombi modelinin yapısından kaynaklı ortamda zombinin olması insan neslinin muhakkak tükeneceği anlamına gelmektedir. Çizelge 8.9 incelendiğinde tüm algoritmaların iyi parametre değerlerine ulaşabildiği gözlemlenmiş olup bu algoritmalar arasındaki farkları modelin yapısından kaynaklı grafikte gözlemlenmek çok daha zordur.



Şekil 8.38 Sonra GA ile İnsan Zombi Modelinde Parametre Ayarlama



Şekil 8.39 Sonra PSO ile İnsan Zombi Modelinde Parametre Ayarlama



fonksiyonlarıdır. Bu iki farklı özelliğe sahip test fonksiyonlarının optimizasyon işlemi için bu tez çalışmasında kullanılan MSA'ların ve geliştirilen MSA'ların optimizasyondaki başarısını tekrar test etmek için kullanılmıştır. Nümerik test fonksiyonlarının optimizasyon işlemi için kullanılan sistem özellikleri Çizelge 8.1'de verilmiştir. Model parametre ayarlamada başarılı olan algoritmalar test fonksiyonların optimizasyon işlemlerinde de başarılı olmuştur. Sonuçlar Çizelge 8.11 ve Çizelge 8.13'da verilmiştir. Çizelge 8.10'da kullanılan test fonksiyonlarına ait Ad, Formül, en iyi çözüm değeri olarak bilinen En Küçük Nokta ve parametrelerin alabilecekleri değer aralığı yani Arama Aralığı verilmiştir.

Çizelge 8.10 Kullanılan Nümerik Test Fonksiyonları

Adı	Formül	En Küçük Nokta	Arama Aralığı
<b>Rastrigin</b>	$f(x) = An + \sum_{i=1}^n [x_i^2 - A \cos(2\pi x_i)]$ $A = 10$	$f(0,0) = 0$	$-5,12 \leq x,y \leq 5,12$
<b>Sphere</b>	$f(x) = \sum_{i=1}^n x_i^2$	$f(0,\dots,0) = 0$	$-\infty \leq x_i \leq \infty,$ $1 \leq i \leq n$
<b>Rosenbrock</b>	$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$f(1,\dots,1) = 0$	$-\infty \leq x_i \leq \infty,$ $1 \leq i \leq n$
<b>Griewank</b>	$f(x) = 1 + 1/400 \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right)$	$f(0,\dots,0) = 0$	$-600 \leq x_i \leq 600,$ $1 \leq i \leq n$

Nümerik test fonksiyonları optimizasyon işlemlerinde minimuma yani Çizelge 8.11'de yer alan en küçük noktaya yaklaşmayı hedeflemektedir. Bunun için ilk olarak MSA'lardan GA, PSO, ABC ve FA algoritmaları kullanılarak optimizasyon işlemi yapılmıştır. Problem boyutu 6 popülasyon boyutu 20 ve iterasyon sayısı 20 olarak alınmıştır. Çizelge 8.11 incelendiğinde her bir fonksiyon üzerinde MSA'ların 5'er kez çalıştırılması sonucu elde edilen veriler görülmektedir. Her bir çalıştırmada problem (parametre) boyutu 6, popülasyon boyutu (çözüm kümesi sayısı) 20 ve iterasyon sayısı 20 olarak belirlenmiştir. Amaç Uyarlanabilir MAS algoritmaları ile MAS algoritmalarını karşılaştırmak olup problem boyutu popülasyon boyutu ve iterasyon sayısı çok büyük değerler seçilmemiştir. Her 4 fonksiyon optimizasyonu için MAS algoritmalarından PSO algoritması iyileştirme de en başarılı algoritma olduğu Çizelge 8.11'de elde edilen sonuçlarda görülmektedir. PSO'dan sonra GA algoritması optimizasyon işleminde iyi sonuçların elde edildiği 2. Algoritma olduğu elde edilen diğer bir sonuçtur. ABC ve FA algoritması optimizasyonda beklenen iyileşmeyi gösterememiştir.

Çizelge 8.11 GA, PSO, ABC, FA Algoritmalarının Numerik Test Fonksiyonlarındaki Optimizasyon Sonuçları

Fonk.	Algoritma	Çalışma 1	Çalışma 2	Çalışma 3	Çalışma 4	Çalışma 5	Ortalama	
Rastrigin	GA	51,441694	26,240791	42,912781	37,892368	46,646883	41,0269	PSO<GA<ABC<FA
	PSO	11,704209	7,9366828	10,187712	11,287594	19,158632	12,05497	
	ABC	51,098063	65,145178	37,602278	50,545854	68,904444	54,65916	
	FA	2,1720481	92,221849	59,820335	66,337483	76,339845	59,37831	
Sphere	GA	9,251076	15,26091	4,763601	9,72903	13,11957	10,42484	PSO<FA<GA<ABC
	PSO	0,09103	0,147775	0,137739	0,288322	0,172281	0,16743	
	ABC	15,9802	10,3251	19,04103	9,302802	11,07244	13,14431	
	FA	9,790489	4,124033	16,95013	8,960395	7,2633	9,41767	
Rosenbrock	GA	194,2464	82,97482	335,1893	402,0784	47,45999	212,3898	PSO<GA<ABC<FA
	PSO	11,47511	8,4054	7,61424	8,22846	12,0354	9,551723	
	ABC	177,0827	368,4447	88,56046	765,1089	53,08777	290,4569	
	FA	452,1605	621,6795	1398,081	454,0912	1707,094	926,6212	
Griewank	GA	0,010298	0,003482	0,003722	0,020253	0,010115	0,009574	PSO<GA<ABC<FA
	PSO	3,94E-05	8,20E-05	3,11E-05	3,11E-05	7,22E-05	5,12E-05	
	ABC	0,020238	0,018766	0,015367	0,014199	0,019891	0,017692	
	FA	0,025814	0,044912	0,032687	0,050713	0,039837	0,038793	

Çizelge 8.11’de test fonksiyonlarının GA, PSO, ABC ve FA algoritmaları ile optimizasyon sonuçları yer almaktadır. Bu sonuçların elde edilmesinde kullanılan GA, PSO, ABC ve FA algoritmalarına ait kritik parametre değerleri manuel girilmiştir. Bu değerler Çizelge 8.12’de verilmiştir.

Çizelge 8.12 Numerik Test Fonksiyonların Optimizasyon İşlemlerinde Kullanılan Algoritmaların Kritik Parametre Değerleri

Algoritma	Kritik Parametreler	Değer
	İterasyon Sayısı	20
	Problem (Parametre) Boyutu	6
	Popülasyon (Çözüm Kümesi Sayısı) Boyutu	20
GA	Ebeveyn Secim Yüzdesi	40
	Mutasyon Yüzdesi	0,1
	Ebeveyn Seçim Yöntemi	Turnuva Yöntem
	Çaprazlama Yöntemi	Tek Noktalı
	Göç Sayısı	2
PSO	C1	1,0
	C2	1,0
ABC	Numerik test fonksiyonu optimizasyonu işlemi için orijinal ABC algoritması kullanılmıştır. Orijinal ABC algoritması kritik parametre değişkenine sahip değildir.	-
Modifiye ABC	MR	CUPA Algoritması ile ayarlandı.
FA	Gama	1,0
	Alpha	0,1

Çizelge 8.13 incelendiğinde her bir fonksiyon üzerinde Uyarlanabilir MSA'ların (AGA, APSO, AABC, AFA) 5'er kez çalıştırılması sonucu elde edilen veriler görülmektedir. Her bir çalıştırmada problem (parametre) boyutu 6, popülasyon boyutu (çözüm kümesi sayısı) 20 ve iterasyon sayısı 20 olarak belirlenmiştir. Numerik test fonksiyonların optimizasyon işleminde amacı, Çizelge 8.10'da yer alan fonksiyonların yaklaşabileceği en küçük noktaya yaklaşmaktır. Çizelge 8.13'de verilen sonuçlar incelendiğinde en başarılı algoritmanın yani diğer algoritmalara göre daha küçük sonuçlar bulan algoritmanın APSO olduğu görülmektedir. Çizelge 8.13, Çizelge 8.11 ile karşılaştırıldığında Çizelge 8.13'de kullanılan her algoritmanın optimizasyon işleminde daha küçük değerler bulunduğu sonucuna varılmaktadır. Amaç minimumu bulmak olduğu için Uyarlanabilir MSA algoritmaları (AGA, APSO, AABC ve AFA) orijinal MSA algoritmalarına (GA, PSO, ABC ve FA) göre numerik test fonksiyonlarında minimum noktaya ulaşma işleminde daha başarılı olduğu sonucuna varılmıştır. Çizelge 8.11'de optimizasyon

işleminde başarılı olamayan ABC ve FA algoritması probleme uyarlanabilir hale getirildikten sonra aynı test fonksiyonlarının optimizasyon işlemlerinden elde edilen sonuçlara göre minimum değerlere (Bkz. Çizelge 8.10) yaklaştığı yine Çizelge 8.13’de yer alan sonuçlar incelendiğinde görülmektedir. Bu durumda geliştirilen AGA, APSO, AABC ve AFA algoritmaları GA, PSO, ABC ve FA algoritmalarına göre optimizasyon işlemlerinde daha başarılı olduğu elde edilen sonuçlarda görülmektedir. Geliştirilen algoritmalarından AABC algoritmasının optimizasyon başarısının arttığı görülmektedir.

Karmaşık sistem problemlerinde performansı arttırmak amacı ile geliştirilen Modifiye-ABC algoritması bu çalışmada kullanılmış ve parametreleri ayarlanarak AABC algoritması geliştirilmiştir. Orijinal ABC’nin Çizelge 8.11’de numerik test fonksiyonlarının optimizasyonu işlemindeki sonuçları, AABC algoritmasının kullanıldığı Çizelge 8.13’deki sonuçlar ile karşılaştırıldığında AABC algoritmasının daha küçük değerler bulduğu görülmektedir.

MSA algoritmalarında iterasyon sayısının artması optimuma yaklaşma ihtimalini artırır(Yeo ve Agyel, 1996). Bu tez çalışmasında nümerik test fonksiyonu optimizasyonu işleminde tüm algoritmaların 20 iterasyon çalıştırılma sonuçları Çizelge 8.11 ve Çizelge 8.13’de verilmiştir. İterasyon sayısının çok büyük olmamasına rağmen Uyarlanabilir MSA algoritmaların buldukları sonuçlar Çizelge 8.10’da verilen en küçük değerlere çok yakın değerlerdir. Dolayısıyla kısa sürede çok daha iyi çözümlerin uyarlanabilir algoritmalarla elde edildiği gözlemlenen diğer bir sonuçtur.

Çizelge 8.13 AGA, APSO, AABC, AFA Algoritmalarının Numerik Test Fonksiyonlarındaki Optimizasyon Sonuçları

Fonk.	Algoritma	Çalışma 1	Çalışma 2	Çalışma 3	Çalışma 4	Çalışma 5	Ortalama	
Rastrigin	AGA	0,0911682	0,1656174	0,1342727	0,0634769	0,1370721	0,1183215	PSO<ABC<GA<FA
	APSO	8,2E-07	2,5E-07	3,695E-05	1E-08	7,36E-06	9,078E-06	
	AABC	0,0400609	0,0700237	0,0584063	0,0726403	0,0809132	0,0644089	
	AFA	0,3305287	0,0207023	0,0155434	0,2738459	0,0206298	0,13225	
Sphere	AGA	0,0372451	0,0172696	0,1120942	0,0681774	0,0188995	0,0507372	PSO<GA<ABC<FA
	APSO	2E-08	8,29E-06	7,81E-06	0,0000015	3,89E-06	4,302E-06	
	AABC	0,1721976	0,0597481	0,1109223	0,158777	0,1721455	0,1347581	
	AFA	0,1929143	0,157503	0,1425771	0,0228694	2,3919588	0,5815645	
Rosenbrock	AGA	0,1340479	0,0948623	0,033258	0,0348619	0,0541642	0,0702389	PSO<FA<ABC<GA
	APSO	1E-08	4E-08	2E-08	5E-08	0	2,4E-08	
	AABC	0,0191021	0,0129505	0,022575	0,0153738	0,0325735	0,020515	
	AFA	0,029229	0,0114477	0,0364245	0,0102042	0,0127767	0,0200164	
Griewank	AGA	0,1293177	0,1000618	0,1324447	0,2947363	0,0969213	0,1506963	GA<ABC<FA<PSO
	APSO	0,0023651	1,21739	7,4078095	1,300044	12,119873	4,4094964	
	AABC	0,3612823	0,1465825	0,7043986	0,7648852	0,1113145	0,4176926	
	AFA	3,7580979	0,0400757	12,036084	1,434788	3,9278392	4,2393769	

## 9.SONUÇ VE TARTIŞMA

Bu tez çalışmasında Karmaşık sistemlerin modellendiği etmen tabanlı model ve benzetimlerin gerçek sistemi yansıtabilmeleri sahip oldukları geniş parametre uzaylarının ayarlanması problemi üzerinde durulmuştur. ETMB ortamında modellenen karmaşık sistem modellerinin parametre ayarlama problem çözümü için 3. Bölümde yer alan literatür çalışmaları incelendiğinde çok fazla çalışmanın yapılmadığı yapılan çalışmaların ise tek bir problemi çözmeye odaklı yapılmış olduğu görülmektedir. Bu tez çalışmasında geliştirilen parametre ayarlama aracı farklı parametre ayarlama problemlerine çözüm üretmesi, probleme uyarlanabilmesi bakımından önemlidir.

Bu tez çalışmasında parametre ayarlama problem çözümü için MSA'lar kullanılmıştır. MSA'ların kullanılması ile birlikte MSA'lara ait kritik parametre değerlerinin var olduğu ve bu değerlerin probleme göre manuel ayarlanma problemi ile karşılaşmıştır. Böylece MSA'lara ait kritik parametre değerlerinin de sistemin bir parçası haline getirilerek ayarlanması bu tez çalışmasının diğer hedefi olarak ele alınmıştır. Bununla birlikte MSA parametre ayarlama yöntemleri incelenmiş ve bu tez çalışmasına kolay uyarlanabilen, çevrimiçi uyarlanabilir parametre ayarlama yöntemi F-Race algoritması (Eiben et al.,1999) ile birlikte kullanılmıştır.

MSA'ların kritik parametre değerlerinin ayarlanmasında ki amaçlar;

- Bağımsız geliştirilen bu algoritmaları, parametrelerinin uyarlanabilir olması ile probleme uyarlanabilir algoritmalar haline getirmek.
- Bazı problemlerin çözümünde göstermiş oldukları başarıyı başarısız oldukları problemlerin çözümünde de gösterebilmesi.
- Probleme bağlı olarak değişmesi gereken MSA'ların kritik parametre değerlerinin algoritma kullanıcılarından bağımsız otomatik olarak ayarlanabilir olması.
- MSA'ların çalışma esnasında kritik parametre değerlerini güncelleyebilmesi yerel en iyilerden kaçınarak yakınsama hızını arttırmak.
- MSA'lar kullanıcılarından parametre değerlerini ayarlama yükünü alıp algoritmanın kendisine vermek.
- MSA'lar Kullanıldıkları problemlerden aldıkları geri bildirimler doğrultusunda kritik parametre değerlerini güncelleyerek probleme uyarlanabilir yapıya sahip olmaları.

MSA'ların kritik parametre değerlerini ayarlamak için, bu tez kapsamında çevrimiçi uyarlanabilir parametre ayarlama yöntemi kullanılmıştır. Bu yöntem algoritmadan alınan geri bildirimler doğrultusunda en uygun parametre setini bulmayı hedeflemektedir. Kullanılan bu yöntem, aldığı geri bildirimler doğrultusunda bulunan başarılı kritik parametre değerlerini tekrar kullanmakta olup algoritmanın çözüme yaklaşma hızını arttırmayı hedeflemektedir. Başarılı değerlerin bulunamaması durumunda yine aldığı geri bildirimler ile kritik parametre değer aralıklarını daraltma ya da genişletme tekniklerini kullanarak parametre seti güncellenmektedir.

Kullanılan çevrimiçi uyarlanabilir parametre ayarlama yöntemi ile ETMB ortamında modellenen karmaşık sistem model parametrelerini ayarlama sürecinde kullanılan MSA'lar bir adım ileri götürülerek probleme uyarlanabilir algoritmalar haline getirilmiştir.

Karmaşık problemlerin çözümünde sıklıkla kullanılan MSA'lardan GA, PSO, ABC ve FA algoritmaları bu tez çalışmasında kullanılmış ve kullanılan algoritmaların kritik parametre değerleri çevrimiçi uyarlanabilir parametre ayarlama yöntemi kullanılarak probleme uyarlanabilen AGA, APSO, AABC ve AFA algoritmalar geliştirilmiştir. Bulunan en iyi kritik parametre değerleri GA, PSO, ABC ve FA algoritmalarına manuel girilerek Sonra GA, Sonra PSO, Sonra ABC ve Sonra FA algoritmaları elde edilmiştir. Bu 12 algoritma ayrı ayrı av-avcı, sekiz-vezir ve insan-zombi modellerinin parametre ayarlama işleminde kullanılmıştır. Modellerden elde edilen sonuçlar algoritmaların başarımını belirlemek amacı ile karşılaştırılmıştır. Ayrıca GA, PSO, ABC ve FA algoritmaların kritik parametre değerleri rasgele atanarak Random GA, Random PSO, Random ABC ve Random FA algoritmaları da kritik parametre değerinin önemini vurgulamak amacı ile karşılaştırmada kullanılmıştır. GA, PSO, ABC ve FA algoritmalarının parametre ayarlama başarılı olduğu kanıtlanmış en az biri kullanılan bir modelin parametrelerini istenen düzeyde ayarlayabilmiştir. Geliştirilen AGA, APSO, AABC ve AFA algoritmalarının parametre ayarlama sürecinde daha başarılı olduğu av-avcı, sekiz-vezir ve insan-zombi modellerinden elde edilen sonuçlarla kanıtlanmıştır. Buda probleme uyarlanabilir özelliğinin algoritmanın performansına doğrudan etki ettiğinin göstergesidir. Son olarak AGA, APSO, AABC ve AFA algoritmalarının bulduğu kritik parametre değerlerinin GA, PSO, ABC ve FA algoritmalarına manuel girildiğinde elde edilen Sonra AGA, Sonra APSO, Sonra AABC ve Sonra AFA algoritmalarının başarısının ve

performansının arttığı av-avcı, sekiz-vezir ve insan-zombi modellerinden elde edilen sonuçlarda görülmektedir.

Kullanılan farklı MSA'ların farklı ETMB model parametre ayarlama problemleri üzerindeki başarımları test edildiğinde, geliştirilen her algoritmanın her problem üzerinde kabul edilebilir doğrulukta parametre ayarlama yaptığı gözlemlenmiştir. Her problem için farklı algoritmanın daha başarılı olduğu elde edilen sonuçlardandır. Ayrıca geliştirilen bu araç ETMB kullanıcılarından parametre değerlerinin manuel belirlenmesi yükünü alıp modelin kendisine vererek zaman kazancı sağlamaktadır. Bazı problemlerde hızlı ve kabul edilebilir parametre değerlerinin bulunması önemli iken bazı problemler de zaman kısıtlı olmaksızın optimuma en yakın parametre değerlerine ihtiyaç duyulmaktadır. Bu tez çalışmasında geliştirilen bu araç kullanıcının beklentisine cevap verecek şekilde hızın ya da optimuma en yakın parametre değerinin önem kazandığı durumlarda kullanıcının gerekli ayarlamaları manuel yapmasına olanak sağlamaktadır.

Her bir algoritma model üzerinde bağımsız çalıştırılabildiği için hangi modelde hangi algoritmanın daha iyi sonuç verdiği gözlemlenebilmektedir. Her bir model için hayati öneme sahip olan uygunluk fonksiyonu kullanıcının modelden beklentisine göre ve her model için farklılık göstereceğinden bu çalışmada uygunluk fonksiyonu oluşturma üzerinde durulmamıştır.

Bu tezde, karmaşık sistem parametre veri uzayında yer alan parametrelerin ayarlanması ve en uygun parametre setinin bulunması işlemi gerçekleştirilmiştir. Bu işlem için MSA algoritmaları kullanılmıştır. MSA algoritmalarını parametre ayarlama probleminde kullanırken karşılaşılan diğer bir problem MSA algoritmalarının sahip olduğu kritik parametre değerlerinin her problemde değişkenlik göstermesidir. Bu problemi çözmek için MSA algoritmaların sahip olduğu kritik parametre değerleri çevrimiçi uyarlanabilir parametre ayarlama yöntemi kullanılarak ayarlanmıştır. Böylece probleme uyarlanabilen uyarlanabilir MSA algoritmaları geliştirilmiş olup model parametre ayarlama problem çözümünde kullanılmıştır. Geliştirilen bu algoritmaların bulunduğu en uygun parametre setleri model üzerinde test edilmiş olup belirlenen hedeflere ulaşılmıştır.

Bu tez çalışmasında elde edilen önemli sonuçlardan bir diğeri, problemin yapısının, kullanılan algoritmaların başarımı üzerinde doğrudan etkili olmasıdır. Yapılan testlerden elde edilen sonuçlar doğrultusunda GA algoritması parametre ayarlama probleminin çözümünde farklı problemlerin çözümünde başarılı

olmuştur. GA algoritması çözüm kümelerinde yer alan iyi parametreleri sonraki nesle aktarması bu başarının temelini oluşturmaktadır. Ayrıca parametre ayarlama işleminde birçok modelde başarı elde edebileceği elde edilen sonuçlardandır. Parametrelerin yer aldığı çözüm kümesinde tüm parametrelerin güncellenmesi ile optimum sonuca ulaşıldığı problemlerde ya da global optimumun yani en iyi tek bir çözümün bulunduğu arama uzaylarında PSO, ABC ve FA algoritmaları başarı göstermektedir. Ayrıca eğer çözüm kümesinde yer alan parametrelerin bağımsız olarak değerlendirilmesi ve güncellenmesi gibi bir yapı söz konusu ise o zaman ABC algoritması bu problem çözümünde başarı sağlamaktadır.

Bu tez çalışması yapılırken MSA'ların kritik parametre değerlerinin ayarlanmasının dışında karşılaşılan ikinci problem modele ait uygunluk fonksiyonunun tasarımı problemidir. Bu fonksiyon kullanıcının beklentileri doğrultusunda farklılık gösterebilmektedir. Uygunluk fonksiyonu tasarımı kullanıcıya bırakacak bir arayüz desteğine ya da otomatik bulabilecek bir algoritmaya ihtiyaç vardır. Gelecekte yapılacak çalışmalar arasında uygunluk fonksiyonu tasarımı problemine çözüm üretmek hedeflenmektedir. Ayrıca geliştirilen parametre ayarlama aracının ETMB'ler için önemli olan Geçerleme Doğrulama ve Test güdümlü bir yöntem ile birleştirilerek istenen modellerin oluşturulmasını kolaylaştıracak araç haline getirmek yapılması planlanan diğer bir çalışmadır. Yapılması planlanan diğer bir çalışma, bu tez çalışmasında F-Race algoritması kullanılarak geliştirilen çevrimiçi uyarlanabilir parametre ayarlama algoritması yerine literatürde yer alan ve içinde yinelemeli F-Race algoritması ve farklı otomatik yapılandırma algoritmalarını barındıran I-Race yöntemi MSA'lara ait kritik parametreleri ayarlama kullanılması planlanmaktadır. Böylece I-Race yöntemi ile kritik parametre değerleri ayarlanan MSA'ların ETMB ortamında modellenen karmaşık sistemlerin parametre ayarlama işlemindeki başarısının incelenmesi hedeflenmektedir.

## KAYNAKLAR DİZİNİ

- Adar, N., Kuvat, G.**, 2012, Paralel Genetik Algoritmelerde Farklılık Ve Geçirgenlik, Dumlupınar Üniversitesi, Fen Bilimleri Enstitüsü Dergisi, 27, 55-66p
- Adenso-Diaz, B. and Laguna, M.**, 2006, Fine-tuning of algorithms using fractional experimental designs and local search, Operations Research, 54, 99–114p.
- Agent Oriented Software (AOS) Pty.Ltd**, 2005, JACK Intelligent Agents-Agent Manual, Australia, [http://aosgrp.com/documentation/jack/Agent\\_Manual.pdf](http://aosgrp.com/documentation/jack/Agent_Manual.pdf) (Erişim Tarihi: 02 Şubat 2018).
- Akay, B. and Karaboga, D.**, 2010, A modified artificial bee colony algorithm for real-parameter optimization, Information Sciences, 192 (2012), 120-142p.
- Akgüç, A.**, 2010, Sayısal Akışkanlar Dinamiği Problemlerinin Optimizasyon Analizlerinde Kriging Yönteminin Kullanılması, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü.
- Angeline, P.J.**, 1995, Evolution revolution: An introduction to the special track on genetic and evolutionary programming, IEEE Expert Intelligent Systems and their Applications, 10, 6-10p.
- Apostolopoulos, T., and Vlachos, A.**, 2011, Application of the firefly algorithm for solving the economic emissions load dispatch problem, Hindawi Publishing Corporation International Conference Journal of Combinatorics, 2011, 23p.
- Autonomous Decision Making Software (AOS)**, Agent Technology [http://www.agent-software.com/featured-research/autonomy\\_and\\_agents/technology/](http://www.agent-software.com/featured-research/autonomy_and_agents/technology/), (Erişim Tarihi : 12 Ocak 2018).
- Back, T.**, 1992, Self adaptation in genetic algorithms, Toward a Practice of Autonomous Systems: Proceedings of the 1st European Conference on Artificial Life, MIT Press, Cambridge, MA, 263–271p.
- Banks, J.**, 1999, Introduction to simulation. In WSC '99: Proceedings of the 31st Conference on Winter Simulation, New York, NY, USA, 7–13p.
- Banks, J., Carson II, J. S., Nelson B. L., and Nicol D. M.**, 2005, Discrete-event system simulation. pearson education, Inc., Upper Saddle River, NJ, fourth edition.

### KAYNAKLAR DİZİNİ (Devam)

- Bansal, J. C., Singh, P. K., Saraswat, M.,** 2011, Inertia weight strategies in particle swarm optimization, 2011 Third World Congress on Nature and Biologically Inspired Computing, Salamanca, 633-640p.
- Bar-Yam, Y.,** 2002, General features of complex systems. Encyclopedia of Life Support Systems, Eolssunesco Publishers, 1, Oxford, UK. Retrieved 16,
- Bartz-Beielstein, T., Parsopoulos, K. and Vrahatis, M.,** 2004, Analysis of particle swarm optimization using computational statistics, Proceedings of the International Conference of Numerical Analysis and Applied Mathematics, Icaam, Ed: Chalkis, Wiley, 34–37p,
- Balmer, M., Raney, B., and Nagel, K.,** 2005, Adjustment of activity timing and duration in an agent-based traffic flow simulation. Progress in Activity-Based Analysis, 91-114p.
- Banjanovic-Mehmedovic, L., Karic, S.,** 2011, Robotic assembly replanning agent based on neural network adjusted vibration parameters. Prof. Abdelhamid Mellouk (ed), Advances in Reinforcement Learning, InTech, Rijeka, 297-312p.
- Başak, M.E., Kuntman, A. And Kuntman, H.,** MOS parameter extraction and optimization with genetic algorithm, Journal of Electrical and Electronics Engineering, Engineering Faculty, Istanbul University, 9 (2), 1101-1107p.
- Battram A.,** 1999, Karmaşıklıkta yol almak, Türk Henkel Dergisi Yayınları, İstanbul,1, 127p.
- Benoît, C., Guillaume H.,** 2005, Automatic tuning of agent-based models using genetic algorithms, International Workshop on Multi-Agent-Based simulation, Utrecht, 41-57p.
- Bernon, C., Capera, D.and Mano J.-P.,** 2009, Engineering self-modeling systems, Engineering Societies In The Agents World ix, Berlin, 248–263p.
- Bolat, B., Erol, O., İmrak, C.,** 2004, Mühendislik Uygulamalarında Genetik Algoritmalar ve Operatörlerin Görevleri, Mühendislik ve Fen Bilimleri Dergisi, 4, 264-271p.

## KAYNAKLAR DİZİNİ (Devam)

- Bolme, D.S., Beveridge, J.R. Draper, B.A., Phillips, P.J. and Lui, Y.M,** 2011, Automatically searching for optimal parameter settings using a genetic algorithm, Computer Vision Systems - 8th International Conference, {ICVS}, Editör: James L. Crowley and Bruce A. Draper and Monique Thonnat, Springer, Sophia Antipolis, 6962, France, 213-222p.
- Bonjean, N., Bernon, C., and Glize, P.,** 2009, Engineering development of agents using the cooperative behaviour of their components, in G. Fortino, M. Cossentino, M.-P. Gleizes, and J. Pavon (eds.), MAS&S @ MALLOW'09, Turin, Vol. 494, CEUR Workshop Proceedings, (on line) p.
- Brax, N., Andonoff, E. , Gleizes, M. and Glize, P.,** 2013, "Self-adapted aided decision-making: Application to maritime surveillance, Proceedings of the 5th International Conference on Agents and Artificial Intelligence (ICAART), Editör: Joaquim Filipe and Ana L. N. Fred, SciTePress, Barcelona, Spain, 419-422p.
- Bullheimer, B., Hartl, R.F., Strauss, C.,** 1997, A new rank based version of the ant system: a computational study, Central European Journal for Operations Research and Economics, 7, 25-38p.
- Calvez, B., Hutzler, G.,** 2005, Automatic tuning of agent-based models using genetic algorithms, Proceedings of the 6th International Workshop on Multi-Agent Based Simulation (MABS'05), Editör: Jaime Simao Sichman and Luis Antunes, Springer, Utrecht, The Netherland, Cilt 3891, 41-57, 2005
- Calvez, B. and Hutzler, G.,** 2007a, Ant colony systems and the calibration of multi-agent simulations: a new approach, Multi-Agents for Modelling Complex Systems (MA4CS'07) Satellite Workshop of the European Conference on Complex Systems 2007 (ECCS'07), 2007, Germany. 16p.
- Calvez, B. and Hutzler, G.,** 2007b, Adaptive Dichotomic Optimization: a New Method for the Calibration Of Agent Based Models, 21st Annual European Simulation and Modelling Conference (ESM 2007), Malta, 415-419p.
- Campbell, A.P., Hummerl, J.R.,** 2007, The dynamic informayion architecture system, An Advanced Simulation Framework For Military And Civilian Applications, [http:// www. Dis.anl.gov/DIAS/paper/SCS/SCS.html](http://www.Dis.anl.gov/DIAS/paper/SCS/SCS.html), (erişim tarihi 15 Şubat 2007).

## KAYNAKLAR DİZİNİ (Devam)

- Carson, J. S.**, 2004, Introduction to modeling and simulation, Proceedings of the 2004 Winter In winter simulation conference, 7–13p.
- Cioppa, T. M., Lucas, T. W., Sanchez, S. M.**, 2004, Military applications of agent-based simulations. In Simulation Conference, Proceedings of the 2004 Winter, 1, IEEE.
- Collier, N., Howe, T., North, M.**, 2003, Onward and upward: the transition to repast 2.0, First Annual North American Association for Computational Social and Organizational Science Conference, 6, Pittsburg, 5p.
- Coy, S.P., Golden, B.L., Runger, G.C. and Wasil, E.A.**, 2001, Using experimental design to find effective parameter settings for heuristics, Journal of Heuristics, 7, 77–97p.
- Cramer F.**, 1998, Kaos ve Düzen, Alan Yayıncılık, İstanbul, 79p.
- Czarn, A., MacNish, C., Vijayan, K., Turlach, B. and Gupta, R.**, 2004, Statistical exploratory analysis of genetic algorithms, IEEE Transactions on Evolutionary Computation, 8, 405–421p.
- Çakırlar, İ.**, 2015, Etmen Temelli Benzetimler İçin Test Güdümlü Bir Yaklaşım, Doktora Tezi, Ege Üniversitesi.
- Çavuşlu, M.A., Karakuzu, C. ve Şahin, Ş.**, 2010, "Parçacık Sürü Optimizasyonu Algoritması ile Yapay Sinir Ağı Eğitiminin FPGA Üzerinde Donanımsal Gerçeklenmesi", Politeknik Dergisi, 13 (2), 83-92p.
- Darty, K., Saunier, J., Sabouret, N.**, 2015, Calibration of multi-agent simulations through a participatory experiment, Proceedings of the 2015 International Conference on Autonomous Agents and Multi Agent Systems{AAMAS}, İstanbul, 1683-1684p.
- Deliktaş, B., Türker, H.T., Coşkun, H. Bıkçe, M. Özdemir E.**, 2005, Genetik algoritma parametrelerinin betonarme kiriş tasarımı üzerine etkisi, Bölgesel Jeoloji-Tektonik ve Sismotektonik Deprem Kaynak Mekanizmaları ve Dalga Yayınımı Sempozyumu, Kocaeli, 2005, 46-54p.
- Di Marzo Serugendo, Giovanna, Gleizes, Marie-Pierre, Karageorgos, Anthony**, 2011, Self-organising software from natural to artificial adaptation, first editör, Heidelberg, Berlin: Springer-Verlag, 7-32p.

## KAYNAKLAR DİZİNİ (Devam)

- Dobslaw, F.**, 2010, A parameter tuning framework for metaheuristics based on design of experiments and artificial neural networks. Proceeding of the International Conference on Computer Mathematics and Natural Computing, Rome, 213-216p.
- Dođru, F.**, 2015, Güncel optimizasyon yöntemleri kullanılarak rezidüel gravite anomalilerinden parametre kestirimi, Hacettepe Üniversitesi Yerbilimleri Uygulama ve Araştırma Merkezi Bülteni, Ankara, 36 (1), 31-43p.
- Dorigo, M., Gambardella, L.M.**, 1997, Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evolutionary Computation, 1 (1), 53-66p.
- Dorigo, M. and Stützle, T.**, 2004, Ant colony optimization, MIT press, Cambridge, MA, 33-46p.
- Durfee, E.H., Lesser, V.R. and Corkili, D.D.**, 1989, Trends in cooperative distributed problem solving, IEEE Transaction on Knowledge and Data Engineering, KDE-1 (1), 63-83p.
- Eberhart, R. C., Shi, Y.**, 2001, Particle swarm optimization: developments, applications and resources, Proceedings of the Congress on Evolutionary Computation, 81-86p.
- Erdur, R. C.**, 2001, Yazılım Etmeni Teknolojisinin İnternet Tabanlı Yazılım Yeniden Kullanımına Uygulanması. Doktora Tezi, Ege Üniversitesi.
- Ergün, S., Aydoğan, T.**, 2015, A study to determine the misconceptions in the field of artificial intelligence, International Journal of Scientific and Technological Research, Vol 1 (1), 39-49p.
- Epstein, J. M.**, 1999, Agent-based computational models and generative social science, In Generative Social Science Studies in Agent-Based Computational Modeling, 4 (5), 41-60p.
- Fehler, M., Klügl, F., and Puppe, F.**, 2006, Approaches for resolving the dilemma between model structure refinement and parameter calibration in agent-based simulations, in AAMAS'06: Proc. of the fifth international joint conf. On Autonomous agents and multiagent systems, New York, NY, USA, ACM, 120-122p.

## KAYNAKLAR DİZİNİ (Devam)

- Ferber, J.**, 1999, Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence, Pearson Education Limited, Addison-Wesley, England, 528p.
- Fıđlalı A. ve Engin O.**, 2002, Genetik Algoritmalarla Akıř Tipi Çizelgelemede Üreme Yöntemi Optimizasyonu, İTÜ Dergisi, 1-6s.
- François, O. and Lavergne, C.**, 2001, Design of evolutionary algorithms a statistical perspective, IEEE Transactions on Evolutionary Computation, 5, 129–148p.
- Franklin, S. ve Graesser, A.**, 1996, Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents, Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag, <http://www.msci.memphis.edu/~franklin/AgentProg.html#classification> (Eriřim Tarihi: 05.02.2005).
- Gambardella, L.M., Dorigo, M.**, 1995, Ant-Q: A reinforcement learning approach to the traveling salesman problem, In Proceedings of the Eleventh International Conference on Machine Learning, California, 252-260p.
- Gandon, F.**, 2002, Distributed artificial intelligent and knowledge management: Ontologies and multi-agent systems for a corporate semantic web, Scientific Philosopher Doctorate Thesis in Informatics, INRIA and University of Nice-Sophia Antipolis – Doctoral School of Sciences and Technologies of Information and Communication (S.T.I.C), France..
- Gardelli, L., Viroli, M. and Omicini, A.**, 2009, Combining simulation and formal tools for developing self-organizing, Multi Agent Systems Simulation and Applications, Taylor and Francis Group, 133-161p.
- Geem ZW, Kim JH, Loganathan GV.**, 2001, A new heuristic optimization algorithm: harmony search, Simulation, 76(2), 60-68p.
- Gencer, C., Bali, Ö.**, 2010. Parçacık sürü optimizasyonu algoritması ile U-tipi hat dengeleme. Kara Harp Okulu Bilim Dergisi, 20 (1), 195-202p.
- Genesereth, M. R., and Ketchpel, S. P.**, 1994, Software agent, Communication of the ACM, 37 (7), 48–53p.

## KAYNAKLAR DİZİNİ (Devam)

- Georgé, J.-P., Gleizes, M.-P., Glize, P. and Régis, C.,** 2003, Real-time simulation for flood forecast: an adaptive multi-agent system STAFF, Sempodium on Adaptive Agents and Multi-Agent Systems (AISB), Toulouse Cedex, 109-114p.
- Glover, F. and Laguna, M.,** 1997, Tabu search, Kluwer Academic Publishers, Norwell, Massachusetts.
- Goldsman, D., Nelson, B.L. and Schmeiser B.,** 1991, Methods for selecting the best system, in: WSC'91: Proceedings of the 23rd Conference on Winter Simulation, IEEE Computer Society, Washington, DC, USA, 177–186p.
- Goldberg D.E.,** 1989, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, USA, 372p.
- Greffenstette, J.J.,** 1986, Optimisation of control parameters for genetic algorithms, In IEEE Transactions on Systems, Man and Cybernetics, 16, 122–128p.
- Grimm, Revilla, E., Berger, U., Jeltsch, W. M., and Railsback, S.,** 2005, Pattern-oriented modeling of agent-based complex systems, Lessons from ecology. Science, 987–991p.
- Guivarch, V., Camps, V. and P'eninou, A.,** 2012, AMADEUS: an adaptive multi-agent system to learn a user's recurring actions in ambient systems, Advances in Distributed Computing and Artificial Intelligence Journal, 3 (1), 1-10p
- Gundogmus, B.,** Arařtırmacılıkta Yeni Paradigma, 2014, <http://bulentgundogmus.com/arastirmacilikta-yeni-paradigma/>, (Eriřim Tarihi, 5 Mart 2018).
- Gürcan, Ö.,** 2013, Biyolojik Sinir Yolaklarının Kendini-Örgütleyen Etmenler Kullanılarak Benzetilmesi, Doktora Tezi Ege Üniversitesi Bilgisayar Mühendisliđi, İzmir.
- Gürcan, Ö., Türker K.S., Mano JP., Bernon C., Dikenelli O., and Glize P.,** 2014, Mimicking human neuronal pathways in silico: an emergent model on the effective connectivity, Journal of Computational Neuroscience, 36 (2), 235-257p.
- Hanbay D.,** 2007, Yapay Sinir Ađı Tabanlı Akıllı Yöntemlerle Karmařık Sistemlerin Modellenmesi, Doktora Tezi, Fırat Üniversitesi, Elazığ.

## KAYNAKLAR DİZİNİ (Devam)

- Hayes-Roth, B.**, 1995, An architecture for adaptive intelligent systems, *Artificial Intelligence*, 72 (1-2), 329–365p.
- Heath, B., Hill, R., and Ciarallo, F.**, 2009, A survey of agent-based modeling practices (january 1998 to july 2008), *Journal of Artificial Societies and Social Simulation* 12 (4).
- Hesser, J. and Manner, R.**, 1991, Towards an optimal mutation probability in genetic algorithms, *Proceedings of the 1st Conference on Parallel Problem Solving from Nature*, number 496 in *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, New York, 23–32p.
- Hiroyasu, T., Miki, M. and Negami, M.**, Distributed genetic algorithms with randomized migration rate, *IEEE Proc. of Systems, Man and Cybernetics Conference (SMC'99)*, 1, 689-694p.
- Holland, J. H.**, 1975, *Adaptation in natural and artificial System*, Ann Arbor: The University of Michigan Press, 20.
- Hosseini, H., Shahbazian, M., Takassi, M. A.**, 2014, The design of robust soft sensor using anfis network, *Journal of Instrumentation Technology*, 2 (1), 9-16p.
- Imbault, F., Lebart, K.**, 2004, A stochastic optimization approach for parameter tuning of support vector machines, *17th International Conference on Pattern Recognition (ICPR)*, Cambridge, 597-600p.
- Jain, A. and Fogel, D. B.**, 2000, Case studies in applying fitness distributions in evolutionary algorithms, II. comparing the improvements from crossover and gaussian mutation on simple neural Networks, *Proceeding of the 2000 IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks*, 91-97p.
- Jang, J. S. R.**, 1997, *Neuro-Fuzzy and soft computing, A Computational Approach To Learning and Machine Intelligence*, Chapter 7, *Derivative-Free Optimization*, Prentice-Hall, USA, 173-196p.
- Jennings, N.R. ve Wooldridge, M.**, 1995, *Intelligent Agents: Theory and Practice*, <http://www.csc.liv.ac.uk/~mjw/pubs/ker95.pdf> (Erişim Tarihi: 23.07.2005).
- Juuso, E. K.**, 2006, *Modeling and Simulation with Intelligent Methods*, Sim-Ser-production simulation service, <https://www.sim-serv.com/pdf/whitepapers>.

## KAYNAKLAR DİZİNİ (Devam)

- Kaddoum, E. and Georg'e, J.-P.**, Collective Self-Tuning for Complex Product Design (short paper), in IEEE International Conference on Self-Adaptive and Self-Organizing Systems(SASO), Lyon, CPS, 2012, pp. (electronic medium).
- KAELBING, L.P.**, 1991, A situated-automataapproachto the design of embedded agents, SIGART Bulletin, 2 (4), 85-88p.
- Kang, F., Li, J. and Xu, Q.**, 2009, Structural inverse analysis by hybrid simplex artificial bee colony algorithms, Computer Structures, 87, 861-870p.
- Karaboga, D.**, 2005, An idea based on honey bee swarm for numerical optimization, Technical Report TR06, Erciyes University, Engineering Faculty Computer Engineering Department.
- Karaboga, D. and Basturk, B.**, 2007, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optimization, 39 (3), 459-471p.
- Karaboga, D. ve Basturk, B.**, 2008, On the performance of artificial bee colony (ABC) algorithm, Applied Soft Computing., 8, 687–697p.
- Karaboga, D. ve Akay, B.**, 2009, A comperative study of artificial bee colony algorithm, Applied Mathematics and Computation, 214, 108–132p.
- Karaboga, N.**, 2009, A new design method based on artificial bee colony algorithm for digital IIR filters, Journal of thr Franklin Institute, 346 (4), 328–348p.
- Kashan, M.H., Navahandi, N. ve Kashan, A.H.**, 2012, Design and economic optimization of shell and tube heat exchangers using artificial bee colony (abc) algorithm, Applied Soft Computing, 12, 352–352p.
- Kennedy, J. and Eberhart, R. C.**, 1995, Particle swarm optimization, Proc. of the IEEE Int. Conference on Neural Networks, 4, 1942-1948p.
- Kennedy, J., Eberhart, R. C., and Shi, Y.**, 2001, Swarm intelligence (first edition), San Francisco: Morgan Kaufmann Publishers, 261-392p.
- Khodier, M., Al-Aqeel, M.**, 2009, Linear and circular array optimization: a study using particle swarm intelligence, Progress In Electromagnetics Research B, 15, 347– 373p.
- Kiliç, S.**, 2008, Bulanık karar ortamında karınca kolonisi optimizasyonu yöntemiyle araç rotalama, Doktora Tezi, İstanbul Teknik Üniversitesi.

## KAYNAKLAR DİZİNİ (Devam)

- Korkmaz Tan, R., ve Bora, Ş.,** 2017a, Parameter Tuning Algorithms in Modeling And Simulation, International Journal Of Engineering Science and Application, 1 (2), 58-66p.
- Korkmaz Tan, R. ve Bora, Ş.,** 2017b, Modelleme ve Benzetim Ortamında Parametre Optimizasyonu ve Kullanılan Teknikler, Mühendislik Bilimleri ve Tasarım Dergisi, 5 (3), 685 – 697p.
- Korkmaz Tan, R. ve Bora, Ş.,** 2017c, Parameter Tuning of Complex Systems Modeled in Agent Based Modeling and Simulation, World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering, 11 (12), 1301-1310p.
- Korkmaz Tan, R. ve Bora, Ş.,** 2018, Parameter Tuning in Modeling and Simulations by Using Swarm Intelligence Optimization Algorithms, IEEE-2017 9th International Conference on Computational Intelligence and Communication Networks, 148-152p.
- Kota, R.,** 2002 Survey report on JACK: A framework tool for building multi-agent software systems for CMIF Projects, Buffalo New jersey.
- Köse, U., Güraksın, E., Deperlioğlu, Ö.,** 2015, Girdap optimizasyon algoritması tabanlı destek vektör makineleri ile diyabet tespiti, Tıp Teknolojileri Ulusal Kongresi (TIPTEKNO), Muğla, 471-474p.
- Krishnanand, K.N., and Ghose, D.,** 2005, Detection of multiple source locations using a glowworm metaphor with applications to collective robotics, IEEE Swarm Intelligence Symposium, 84-91p.
- Küçükdeniz, T.,** 2009, Sürü zekası optimizasyon tekniği ve tedarik zinciri yönetiminde bir uygulama, Doktora Tezi, İstanbul Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 26-48s.
- Lee, S., Kang, S., Han, D.,** 2006, Agent-Based flexible video conference system with automatic qos parameter tuning, Trends in Artificial Intelligence, 9th Pacific Rim International Conference on Artificial Intelligence (PRICAI), Guilin, 51-60p.
- Lasarczyk, C.W.G.,** 2007, Genetische programmierung einer algorithmischen chemie, Ph.D. Thesis, Technische Universiteit Dortmund.

## KAYNAKLAR DİZİNİ (Devam)

- Lemouzy, S., Camps, V. and Glize, P.**, 2011, Principles and properties of a mas learning algorithm: A comparison with standard learning algorithms applied to implicit feedback assessment", in Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT '11), 2, IEEE Computer Society, Washington, DC, USA, 228–235p.
- Li, G., Niu, P. and Xiao, X.**, 2012, Development and investigation of efficient artificial bee colony algorithm for numerical function optimization, Applied Soft Computing., 12, 320–332p.
- Luck M. and d’Inverno M.**, 2004, Understanding agent systems, Springer-Verlag Berlin Heidelberg, Germany
- Luke S., Cioffi-Revilla, C., Panait L. Sullivan K., Mason**, 2004, A new multi-agent simulation toolkit , Proceedings of the 2004 SwarmFest Workshop, 2004.
- Ma, M., Liang, J., Guo, M., Fan, Y. ve Yin, Y.**, SAR image segmentation based on artificial bee colony algorithm, Applied Soft Computing, 11, 5205–5214p.
- Macal, C. and North, M.**, 2006, Tutorial on agent-based modeling and simulation part 2: how to model with agents, in WSC '06: Proceedings of the 38th conference on Winter simulation, Orlando, Florida, 73–83p.
- Macal, C., and North M. J.**, 2009, Agent-based modeling and simulation, In Proceedings of the 2009 Winter Simulation Conference, ed. M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, Institute of Electrical and Electronic Engineers, Piscataway, New Jersey, , 86-98 pp.
- Maes P.**, 1991, The Agent network Architecture, SIGART Bulletin, 2 (49), 115-120p..
- Maes, P.**, 1995, Artificial life meets entertainment: life like autonomous agents. Commun. ACM, 38 (11), 108–114p.
- Mala, M.**, 2009, Askeri Harekatın planlanmasında çoklu Etmen Tabanlı Modelleme ve Benzetim Mimarisi, Doktora Tezi, Sakarya Üniversitesi.
- Man, K.F., Tang, K.S. ve Kwong, S.**, 2005. Genetic Algorithms, International Journal of Adaptive Control and Signal Processing, 19 (1), 59 - 69p.

## KAYNAKLAR DİZİNİ (Devam)

- Maniezzo, V., Colorni, A., and Dorigo, M.** 1994. The ant system applied to the quadratic assignment problem, *IEEE Transactions on Knowledge and Data Engineering*, IRIDIA, Universite Libre de Bruxelles, , 11 (5), Belgium, 769-778p.
- Manuel, F., Franziska, K. and Frank, P.**, 2004, Techniques for analysis and calibration of multi-agent simulations, *Engineering Societies in the Agent World (ESAW)*, , Editör: Marie Pierre Gleizes and Andrea Omicini and Franco Zambonelli, Springer, 3451, Würzburg, 305-321p.
- Manuel, F., Franziska, K. and Frank, P.**, 2006, Approaches for Resolving the Dilemma between Model Structure Refinement and Parameter Calibration in AgentBased Simulations, In 5th International Joint Conference on Autonomous Agents and Multiagent Systems., Hakodate, 120-122p.
- Maron, O. and Moore, A.**, 1997, The racing algorithm: model selection for lazy learners, in: *Artificial Intelligence Review*, Kluwer Academic Publishers, Norwell, MA, 11, USA, 193–225p.
- Martinjak, I.**, 2007, Comparison of Heuristic Algorithms for the N-Queen Problem, 2007, Proceedings of the ITI 2007 29th Int. Conf. on Information Technology Interfaces, Cavtat, Croatia, 759-764p.
- Masehian, E., Akbaripour, H. and Mohabbati-Kalejahi, N.**, 2014, “Solving the n-Queens Problem Using a Tuned Hybrid Imperialist Competitive Algorithm”, *The International Arab Journal of Information Technology*, Vol. 11(6), 550-559p.
- Masehian, E., Akbaripour, H. and Mohabbati-Kalejahi, N.**, 2013 Landscape analysis and efficient metaheuristics for solving the n-queens problem, *Computational Optimization and Applications*, 1-25p.
- Mellouli, S., Mineau, G. W., and Pascot, D.**, 2002, Modelling a multi-agent system environment. In *ESAW*, 227–240 pp.
- McClelland, J. L., & Rumelhart, D. E.**, 1986, Parallel distributed processing, explorations in the microstructure of cognition, Psychological and biological models, MIT Press, Cambridge, MA, 2.
- Mcculloch, W. S. and Pitts, W.**, 1990, A Logical Callculus of The Ideas Immanent in Nervous Activityy, *Bulletin of Mothemnticnl Biology*, Cilt 52 (1/2), 99-115p.

## KAYNAKLAR DİZİNİ (Devam)

- Miller J. H., and Page S. E.,** 2007, *Complex Adaptive Systems: An Introduction to Computational Models of Social Life*, Princeton University Press, 10-31p.
- Minar, N.R., Burkhart, C., Langton C., Askenaz M.,** 1996, *The Swarm Simulation System: A Toolkit For Building Multi-Agent Simulation*, Santa Fe,  
<https://pdfs.semanticscholar.org/3df5/d4c65795f85259c7c175c08cac58c53f203b.pdf>, (Erişim Tarihi: 17.06.2017)
- Mitchell, M.,** 1999, *An Introduction to genetic algorithms*, Cambiridge: MIT Pres, London England, 0-262-13316-4 (HB), 2-24p..
- Minsky, M. and Riecken, D.,** 1994, A conversation with Marvin Minsky about agents. *Commun. ACM*, 37 (7), 22-29p.
- Mitchell M.,** 2009, *Complexity, a guided tour*, Oxford University Press, USA.
- Montagna, S. and Roli, A.,** 2009, Parameter tuning of a stochastic biological simulator by metaheuristics, in XI Conf. of the Italian Assoc. for Artif. Intell. 2009 (AI\*IA 2009), *Lecture Notes in Computer Science*, 5883, Berlin: Springer, 466-475p.
- Myers, R., and Hancock, E.R.,** 2001, Empirical modelling of genetic algorithms, *Evolutionary Computation* , 9, 461-493p.
- Nannen, V., Eiben, A.E.,** 2006, A method for parameter calibration and relevance estimation in evolutionary algorithms, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'06)*, Ed: M. Keijzer, Morgan Kaufmann, San Francisco, 183-190p.
- Nannen, V. and Eiben, A.E.,** 2007a, Efficient Relevance Estimation and Value Calibration of evolutionary algorithm parameters, in: *IEEE Congress on Evolutionary Computation*, 103-110p.
- Nannen, V. and Eiben, A.E.,** 2007b, Relevance Estimation and Value Calibration of evolutionary algorithm parameters, *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, Ed: M. M. Veloso, Hyderabad, India 1034-1039p.
- Naylor, T., Balintfy J., Burdick, D., and Chu, K.,** 1966, *Computer Simulation Techniques*. John Wiley, New York.

### KAYNAKLAR DİZİNİ (Devam)

- North M., Collier N.T., Vos J.R.**, 2006, Experiences creating three implementations of the repast agent modeling toolkit, *ACM Transactions on Modeling and Computer Simulation* 16 (1), 1-25p.
- Nwana, H. S.**, 1995, Software agents: An overview, *Knowledge Engineering Review*, 11 (2), 205–244p.
- Osman, I.H. and Laporte, G.**, 1996, Metaheuristics, A bibliography, *Ann. Oper. Res.*, 63, 513–623p.
- Öksüm, E., Dolmaz, M.N.**, 2006, Prizma Modelleri Kullanılarak Sentetik Mağnetik Anomalilerin Gauss-Newton Yöntemi ile Ters Çözülmesi, *Süleyman Demirel Üniversitesi-Fen Bilimleri Enstitüsü Dergisi*, 10 (3), 437-446p.
- Öztürk, A.**, 2007, Güç Sistemlerindeki Gerilim Kararlılığının Genetik Algoritma ile İncelenmesi, Doktora Tezi, Sakarya Üniversitesi.
- Pegden, C. D., Shannon, R. E., and Sadowski, R. P.**, 1995, Introduction to simulation using SIMAN (2nd Ed.), Singapore: McGraw-Hill.
- Pereira, A., Duarte, P., Reis, L.P.**, 2008, Agent-based ecological model calibration - on the edge of a new approach, *Computing Research Repository*, abs/0809.1686, 107-113p.
- Polack, F. A. C., Andrews, P. S., Ghetiu, T., Read, M., Stepney, S., Timmis, J., and Sampson, A. T.**, 2010, Reflections on the simulation of complex systems for science. In *Proc. of International Conference on Computational Science (ICCS'2010)*, 276–285p.
- Ramos, I., Goldberg, M., Goldberg, E. and Neto, A.**, 2005, Logistic regression for parameter tuning on an evolutionary algorithm, in: *Proceedings of the 2005 IEEE Congress on Evolutionary Computation IEEE Congress on Evolutionary Computation*, , IEEE Press, 2, Edinburgh, UK, 1061–1068p.
- Rebaudengo, M. and Reorda, M.S.**, 1992, An experimental analysis of effects of migration in parallel genetic algorithms, *EWDP93:IEEE/Euromicro Workshop on Parallel and Distributed Processing, Gran Canaria (E)*, Gennaio, 232-238p.
- Reeves, C.R.**, 1995, *Modern heuristic techniques for combinatorial problems*, McGrawHill Book Company, New York, 192p.

### KAYNAKLAR DİZİNİ (Devam)

- Railsback, S., Lytinen, S., and Jackson, S.,** 2006, Agent-based simulation platforms: Review and development recommendations, *SIMULATION* 82 (9), 609p.
- Ren, Z. ve Anumba, C.J.,** 2003, Multi- Agent Systems in Construction- State of the Art and Prospects, *Automation in Construction*, CICE, Loughborough University, 13 (3), 421-434p.
- Riccardo, P., Kennedy, J., Blackwell, T.,** 2007, Particle swarm optimization, *Swarm Intelligence*, 1 (1), 33-57p.
- Kara, R.,** 2006, Nonlinear Dinamik Sistemlerde Kaos, Dallanma Ve Fraktaller, İstanbul Teknik Üniversitesi - Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi.
- Russell, S. J., Norvig, P., Candy, J. F., Malik, J. M., and Edwards, D. D.,** 1996, *Artificial intelligence: a modern approach*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 5-59p.
- Sallans, B., Pfister, A., Karatzoglou, A. and Dorffner, G.,** 2011, Simulation and validation of an integrated markets model", *J. Artificial Societies and Social Simulation*, Cilt 6, Sayı 4, 2003, 693-706p.
- Salwala, C. Kotrajaras, V. and Horkaew, P.,** 2010, Improving Performance for Emergent Environments Parameter Tuning and Simulation in Games Using GPU, *Computer Science and Information Technology (ICCSIT)*, 2010 3rd IEEE International Conference on, 2, 37-41p.
- Shi, Y, Eberhart, R. C.,** 1998a, Parameter selection in particle swarm optimization, *Evolutionary programming VII içinde*, Springer Berlin Heidelberg, 591-600p.
- Schwefel, H.P.,** 1995, *Evolution and Optimum Seeking* (John Wiley and Sons, Inc, New York, Chicester, Brisbane, Toronto, Singapore)
- Samrat, L., Udgata, S. and Abraham, A.,** 2010, Artificial bee colony algorithm for small signal model parameter extraction of MESFET, *Eng. App. Artificial Intelligence* 11, 1573-2916p.
- Saraçoğlu, H. ve Demirören, A.,** 2008, Parametreleri Genetik Algoritma ile Ayarlanan Bulanık Kontrolör Yardımıyla Otomatik Gerilim Kontrolü", *Elektrik Elektronik-Bilgisayar Mühendisliği 12. Ulusal Kongresi*, Eskişehir.

## KAYNAKLAR DİZİNİ (Devam)

- Saraçoğlu, B., Güvenç, U., Dursun, M., Poyraz, G., Duman, S.,** 2013, Biyocoğrafya Tabanlı Optimizasyon Metodu Kullanarak Asenkron Motor Parametre Tahmini. İleri Teknoloji Bilimleri Dergisi, 2 (1), 46-54p.
- Serugendo, G. D. M., Irit, M. P., and Karageorgos, A.,** 2006, Self-organisation and emergence in MAS: An overview. Informatica, Simon, H.A. 1964. The architecture of complexity, General Systems Yearbook, 30 (1), 6376p.
- Shakeri, C.,** 1998, Discovery of Design methodologies for the integration of multidisciplinary design problems, Chapter 5- Multi-gent Systems, p83103, <http://www.wpi.edu/Pubs/ETD/Available/etd-101399155624/unrestricted/cirrus.pdf> (Erişim Tarihi: 21.04.2006).
- Shannon, R. E.,** 1998, Introduction to the art and science of simulation, In Winter Simulation Conference, 7–14p.
- Shen, J. ve Lesser, V.,** 2006, Communication management using abstraction in distributed bayesian networks, <http://mas.cs.umass.edu/~jyshen/papers/aamas06-bn.pdf> (Erişim Tarihi: 13.03.2006).
- Singh, A.,** 2009, An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem, Applied Soft Computing, 9, 625631p.
- Smith, G.J., Gero, J.S., E.D. Brazier, F.M.T.,** 2002, interaction and experience: situated agent and sketching, agents in design, Key Centre of Design Computing and Cognition, University of Sidney, 115-132p.
- Smit, S.K. and Eiben A.E.,** 2009, Comparing parameter tuning methods for evolutionary algorithms, in: IEEE Congress on Evolutionary Computation, IEEE Press, Ed: Trondheim, 399–406p.
- Smit, S.K. and Eiben, A.E.,** 2010a, Beating the ‘world champion’ evolutionary algorithm via REVAC tuning, in: IEEE Congress on Evolutionary Computation, IEEE Computational Intelligence Society, IEEE Press, Barcelona, Spain, 1–8p.
- Smit, S.K. and Eiben, A.E.,** 2010b, Parameter tuning of evolutionary algorithms, Applications of Evolutionary Computation, in: Lecture Notes in Computer Science, Ed: generalist vs. specialist, in: C. Di Chio, et al., 6024, 542–551p.

## KAYNAKLAR DİZİNİ (Devam)

- Storn, R., Price, K.,** 1995, differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces, Technical Report TR-95-012, International Computer Science Institute, Berkeley.
- Stützle, T. and Hoos, H.,** 1997, The MAX-MIN ant system and local search for the traveling salesman problem. Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC'97), USA, 313-329p.
- Stützle T., Hoos H.,** 1998, "Improvements on the Ant System, Introducing the MAX-MIN Ant System, Artificial Neural Networks and Genetic Algorithms", Springer Verlag, Wien New York, 245-249p.
- Stützle, T.,** 1999, Local Search Algorithms for Combinatorial Problems: Analysis, Improvements and New Applications, Infix, Sankt Augustin, Germany, 220, 203.
- Sun, X., Zhang, L., Tan, H., Bao, J., Strouthos, C., and Zhou, X.,** 2012, Multi-scale agent-based brain cancer modeling and prediction of tki treatment response: Incorporating egfr signaling pathway and angiogenesis, BMC Bioinformatics 13 (1), 218p.
- Sycara, K.,** 1998, Multiagent Systems, AI Magazine, vol. 10, no. 2, <http://www.findarticles.com/p/articles> (Erişim Tarihi: 24.03.2005).
- Şahin, Y., Eroğlu, A.,** 2014, kapasite kısıtlı araç rotalama problemi için metasezgisel yöntemler: bilimsel yazın taraması, Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 19 (4), 337-355p.
- Şeker, E.M.,** 2006, Etmen tabanlı sistemlerle kullanıcı hareketinin modellenmesi: müze örneği, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi
- Taguchi, G., Yokoyama, T.,** 1993, Taguchi Methods: Design of Experiments, ASI Press, 1993.
- Tamer, S. , Karakuzu C.** 2006, Parçacık Sürüsü Optimizasyon Algoritması ve Benzetim Örnekleri, ELECO 2006, Elektronik Bildirileri Kitabı, [http://www.emo.org.tr/ekler/e5d75028d92047a\\_ek.pdf](http://www.emo.org.tr/ekler/e5d75028d92047a_ek.pdf), (Erişim Tarihi: 05.11.2017)
- Terano, T.,** 2007, Exploring the vast parameter space of multi-agent based simulation, Multi-Agent-Based Simulation VII, 4442, 1–14p.

## **KAYNAKLAR DİZİNİ (Devam)**

- Tesfatsion, L., and Judd, K. L.**, 2006, Handbook of computational economics: agent-based computational economics, 2, 829-1660p
- Tisue, S., & Wilensky, U.** 2004, NetLogo: a simple environment for modeling complexity, International Conference on Complex Systems, Boston, 16-21p.
- Troitzsch, K. G.**, 1996, Multilevel simulation, Social Science Microsimulation, 33-62p
- Turkay, M.**, 2006, Optimization models and solution algorithms, New Frontiers in Total Quality and Strategic Management, Editör:S. Kingir, Gazi Publishing, Ankara, 309-328p.
- Valenzano, R.A., Sturtevant, N.R., Schaeffer, J., Buro, K., Kishimoto, A.**, 2010. Simultaneously Searching with Multiple Settings: An Alternative to Parameter Tuning for Suboptimal Single-Agent Search Algorithms. Proceedings of the 20th International Conference on Automated Planning and Scheduling-ICAPS, Toronto, 177-184p.
- Volterra, V.**, 1978, Variazioni et fluttuazioni del numero d'individui in specie animali conviventi, 6 (2), 31-113p. Scudo ve Ziegler (Trans.), R. Comitato Talassografico Memoria (original printing 1927)
- Voss S., Martello, S., Osman, I. H. and Roucairol, C.**, 1999, Metaheuristics: advances and trends in local search paradigms for optimization, Kluwer Academic Publishers, Boston.
- Weiss, G.**, 1999, Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. MIT Press.
- White, T., Pagurek, B., Oppacher, F.**, 1998, ASGA: Improving the ant system by integration with genetic algorithms. Proceedings of the Third Genetic Programming Conference, Wiskonsin, 610-617p.
- Wooldridge, M.**, 2002, An Introduction to Multiagent Systems, John Wiley & Sons, Chichester, England.
- Yang, X. S.**, 2008, Firefly algorithm, Nature-Inspired Metaheuristic Algorithms, Luniver Press, Frome, UK, 79-90p.
- Yang, X.S.**, 2009, Firefly algorithms formultimodal optimization, Proceedings of the Stochastic Algorithms: Foundations and Applications, Lecture Notes in Computing Sciences, Springer, Sapporo, 5792, Japan, 178-178p.

**KAYNAKLAR DİZİNİ (Devam)**

- Yanıköđlu, H., Özkara, E., Yüceer, M.**, 2010, Kinetik model parametrelerinin belirlenmesinde kullanılan optimizasyon tekniklerinin kıyaslanması, 9. Ulusal Kimya Mühendisliđi Kongresi (UKMK-9), Ankara.
- Yeniay, Ö.**, 2001, An Overview of Genetic Algorithms, Anadolu Üniversitesi Bilim ve Teknoloji Dergisi, 2 (1), 37-49p.
- Yeo M. F. ve Agyel E. O.**, 1996, Optimising Engineering Problems Using Genetic Algorithms, Engineering Computations, 15 (2), 268-280p.
- Yüceer, M., Atasoy, İ., Berber, R.**, 2004. Kinetik Modellerde Optimum Parametre Belirleme İçin Bir Yazılım: PARES, 6. Ulusal Kimya Mühendisliđi Kongresi- UKMK-6, İzmir, 7-10s.
- Yüceer, M., Atasoy, I., Berber, R.**, 2005, An integration based optimization approach for parameter estimation in dynamic models, Computer Aided Chemical Engineering, 20 (1), 631-636p.
- Yüceer, M., Atasoy, I., Berber, R.**, 2008, A software for parameter estimation in dynamic models, Brazilian J. of Chem. Engineering, 25 (4), 813 – 821p.
- Zhu, G. ve Kwong, S.** 2010, Gbest-guided artificial bee colony algorithm for numerical function optimization", Appl. Math. Comput., 217, 3166–3173p. 2010.



## **EKLER**

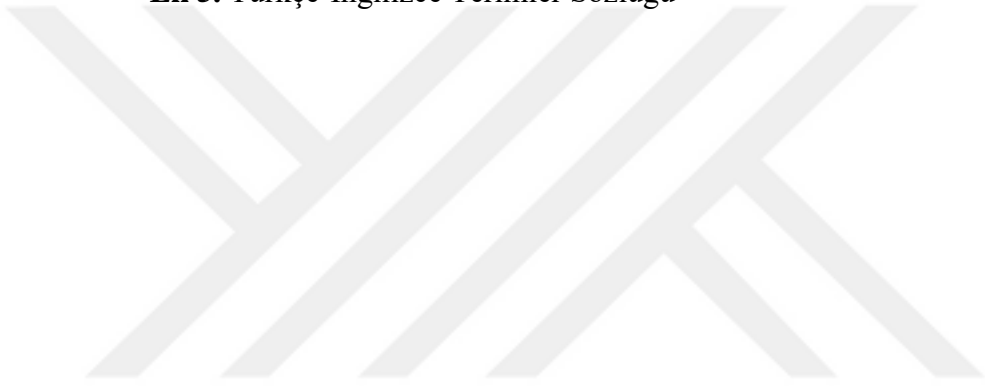
**Ek 1.** Geliştirilen ÇUPA, AGA, APSO, AABC ve AFA Algoritmalarına Ait Akış Diyagramları.

**Ek 2.** Geliştirilen Parametre Ayarlama Aracının Akış Diyagramı

**Ek 3.** Geliştirilen Parametre Ayarlam Aracının Sınıf Diyagramı-1

**Ek 4.** Geliştirilen Parametre Ayarlam Aracının Sınıf Diyagramı-2

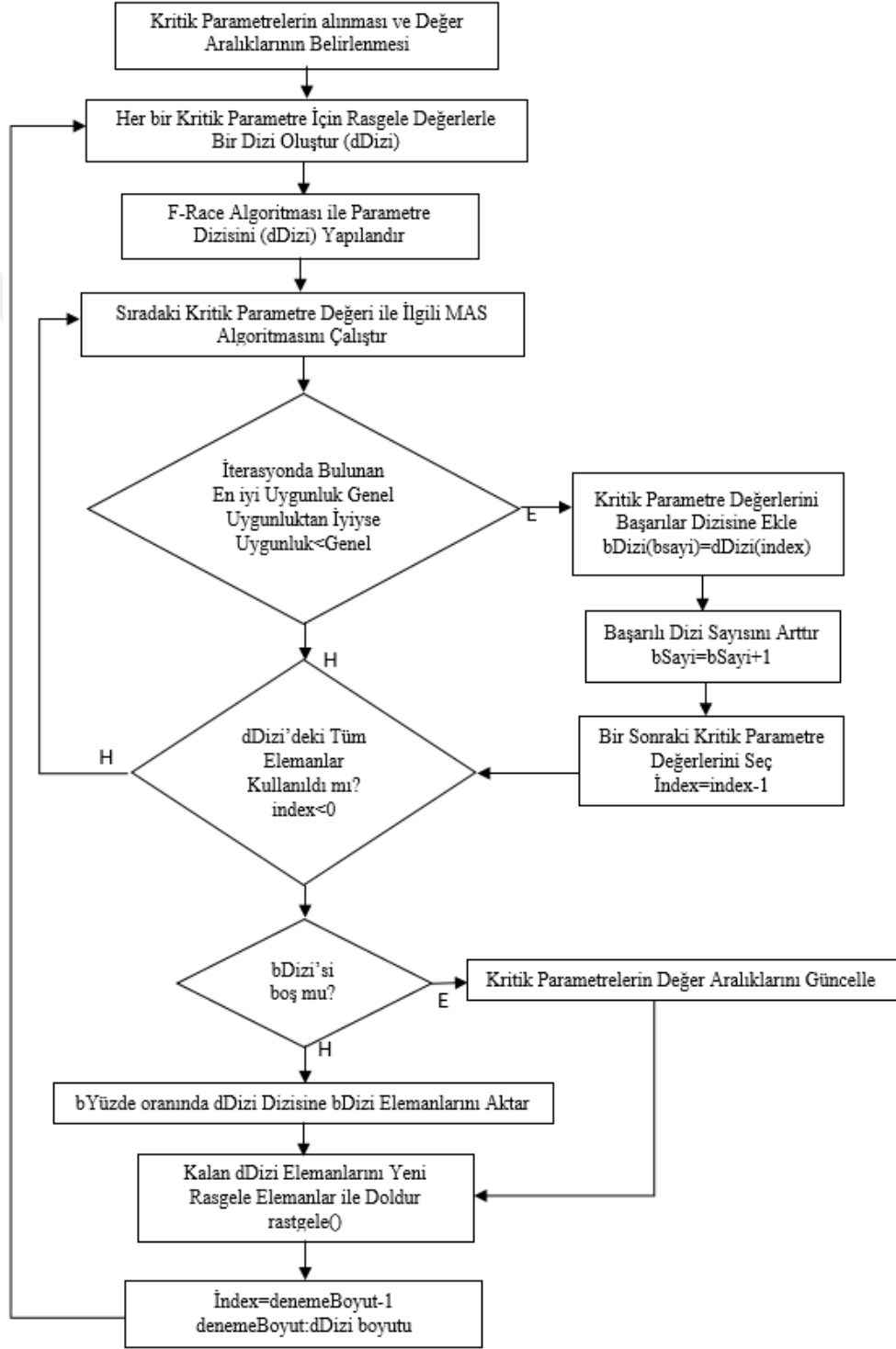
**Ek 5.** Türkçe-İngilizce Terimler Sözlüğü



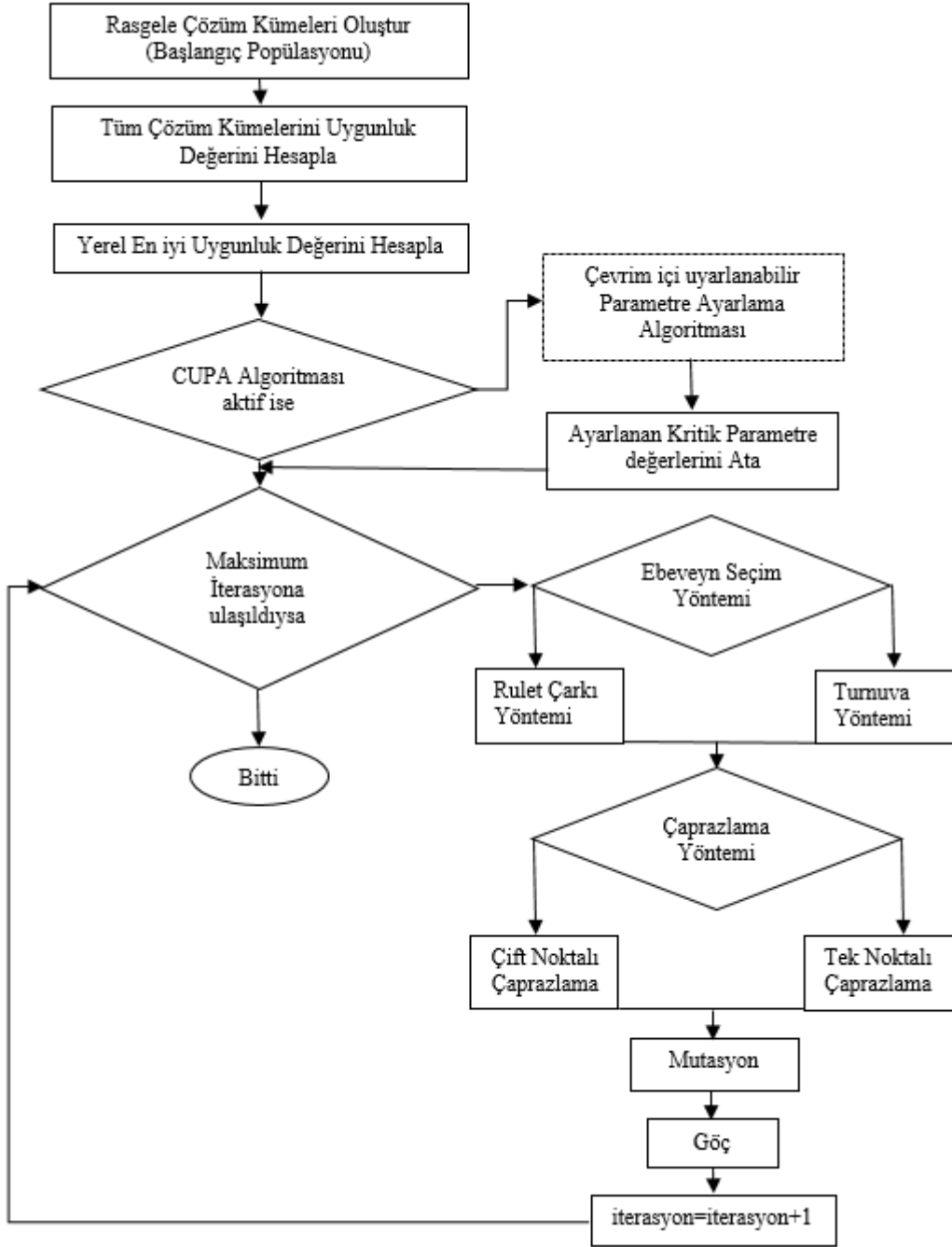


# EK 1. GELİŞTİRİLEN ÇUPA, AGA, APSO, AABC, AFA ALGORİTMALARINA AİT AKIŞ DİYAGRAMLARI

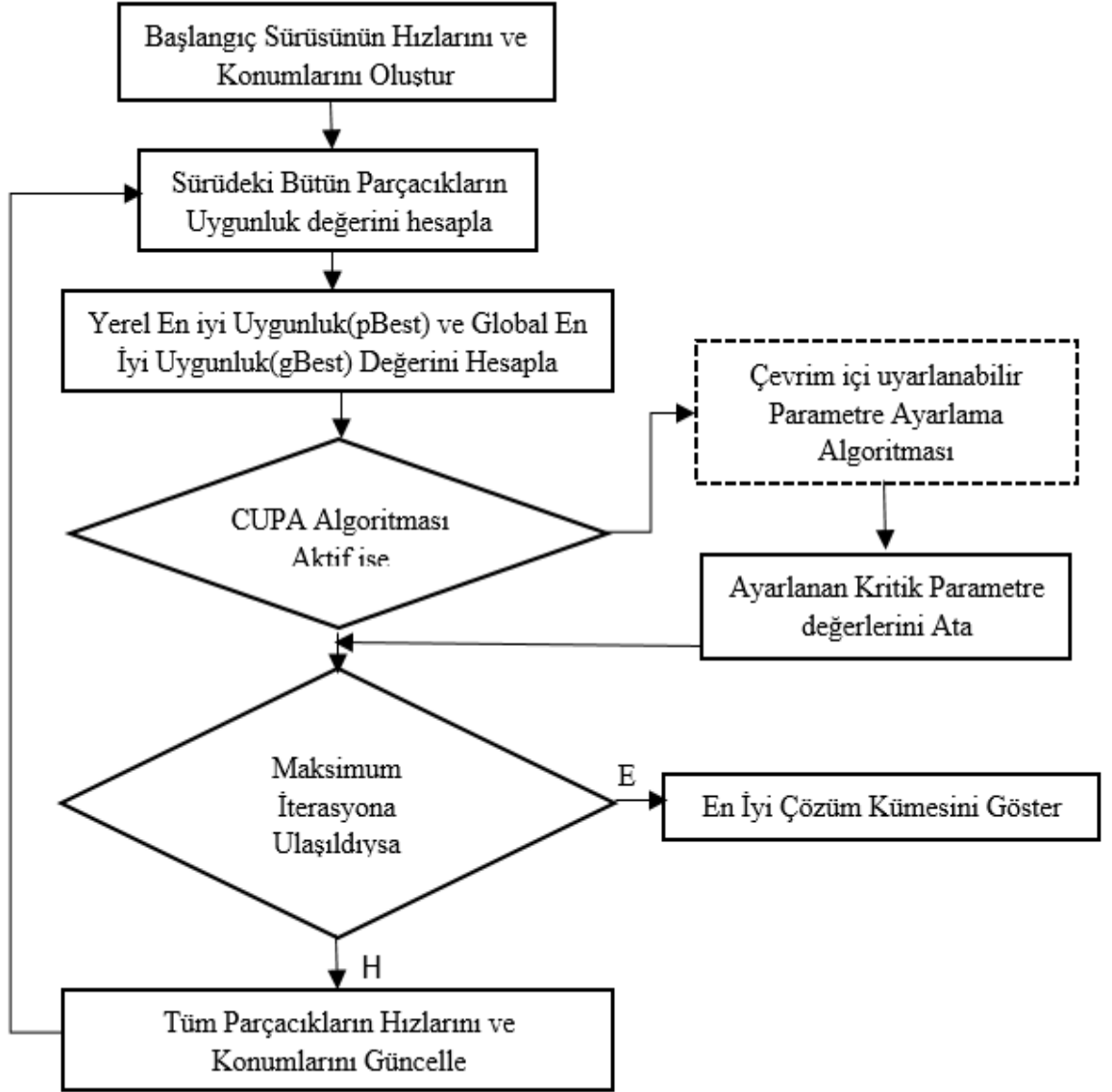
## Çevrim İçi Uyarlanabilir Parametre Ayarlama (ÇUPA) Algoritması Akış Diyagramı



## Uyarlanabilir Genetik Algoritması (AGA) Akış Diyagramı

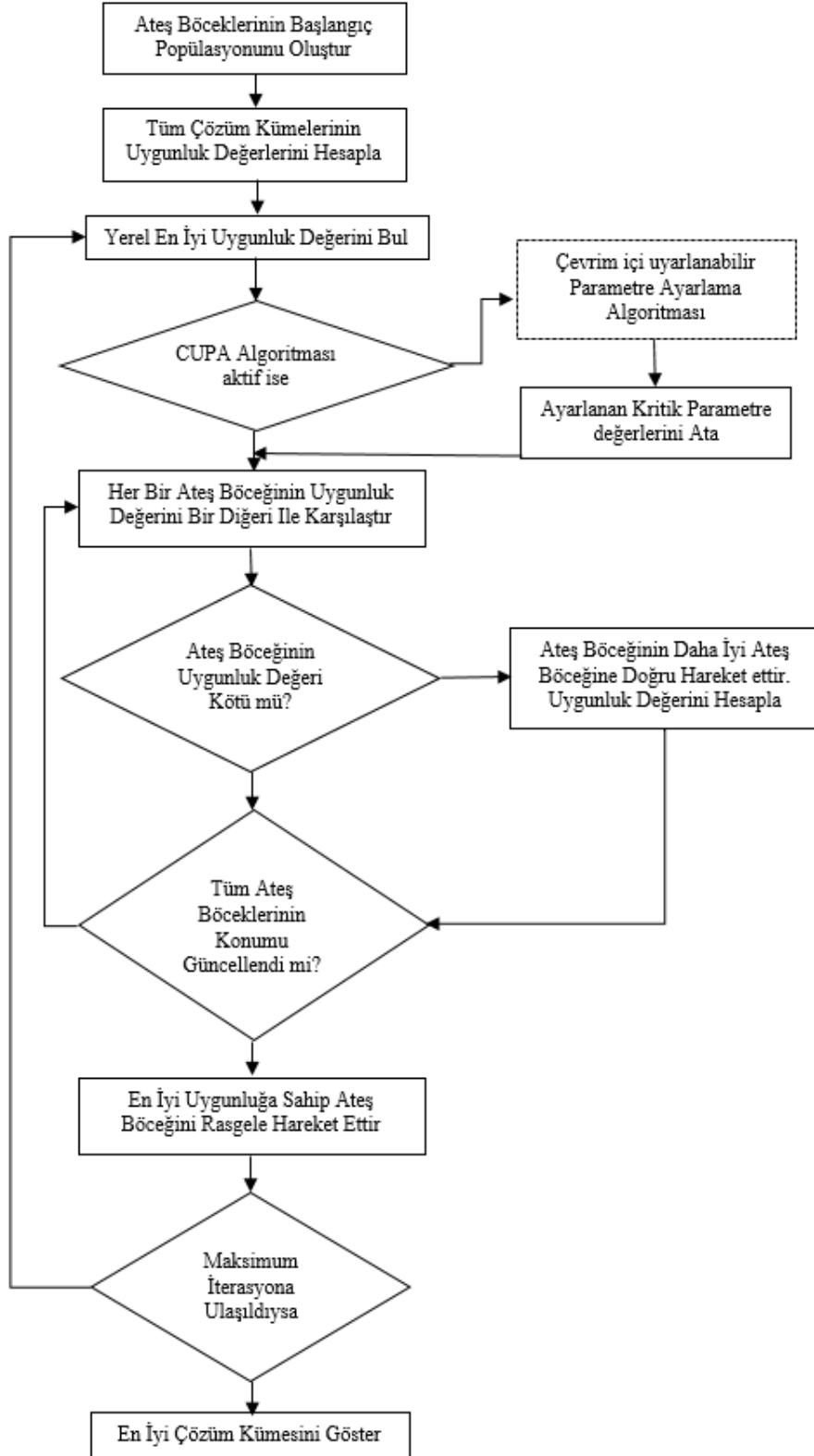


## Uyarlanabilir Parçacık Sürü Optimizasyonu (APSO) Akış Diyagramı

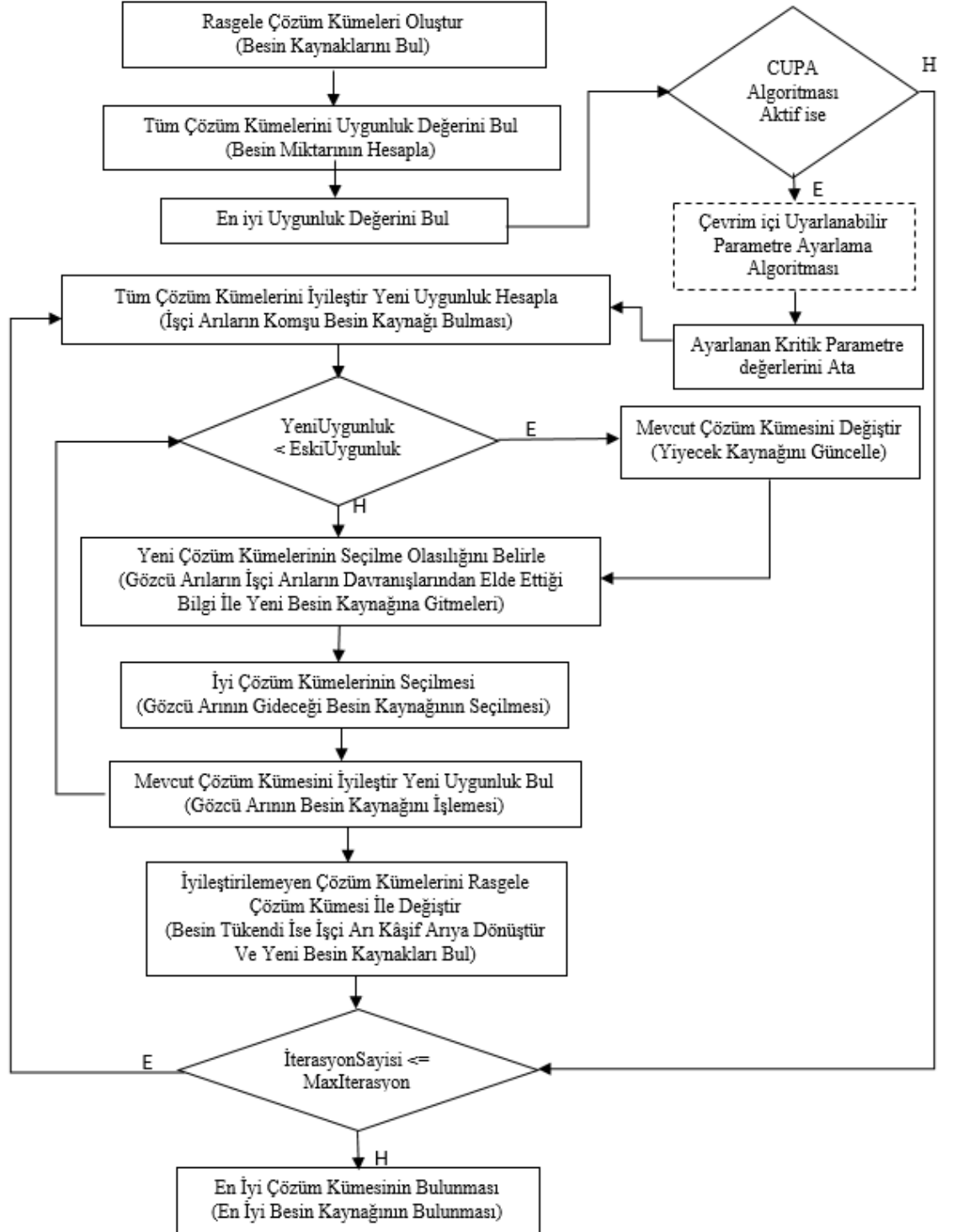


## Uyarlanabilir Ateş Böceği Algoritması (Adaptive Firefly Algorithms-AFA)

### Akış Diyagramı

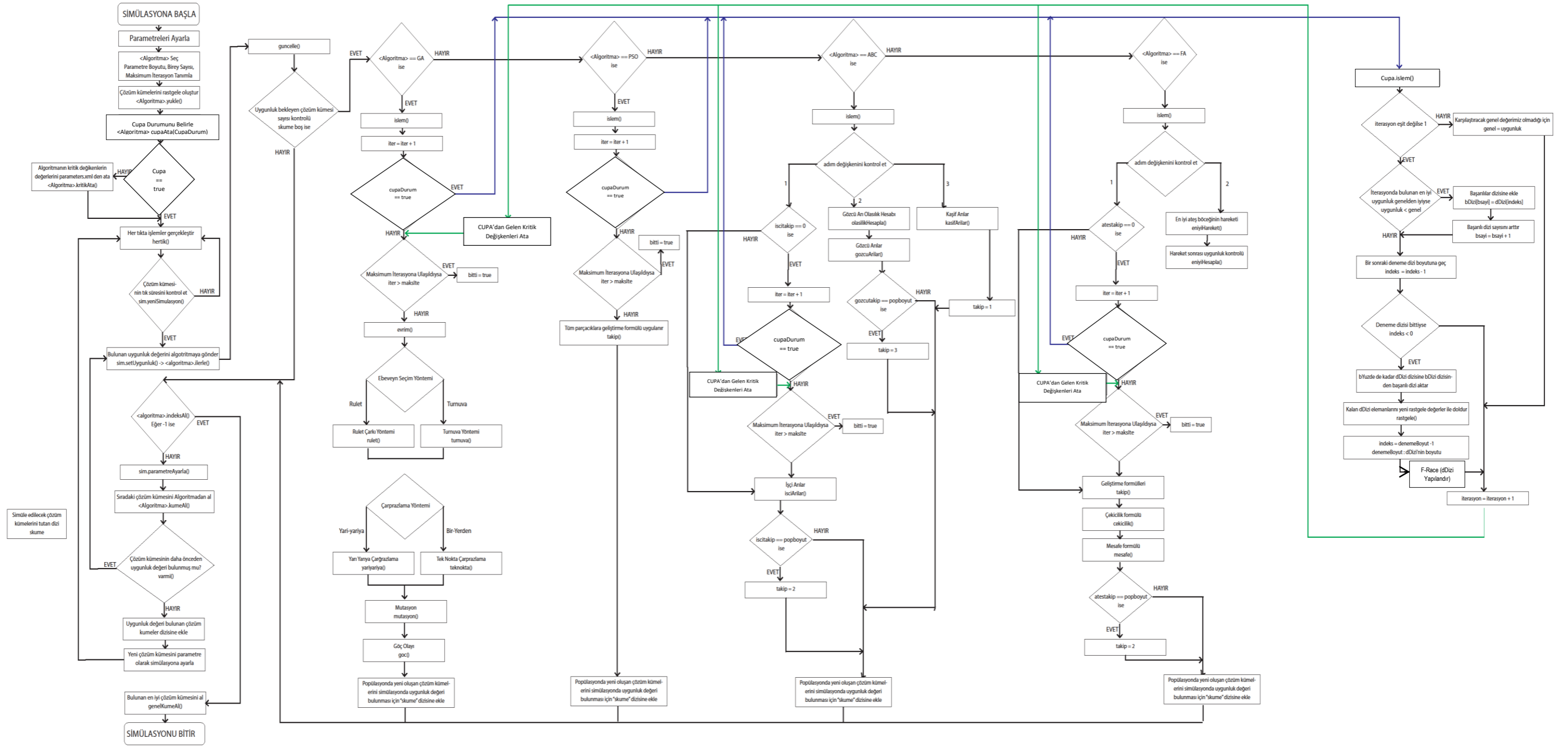


## Uyarlanabilir Yapay Ari Kolonisi (Adaptive Artificial Bee Colony - AABC) Algoritması Akış Diyagramı



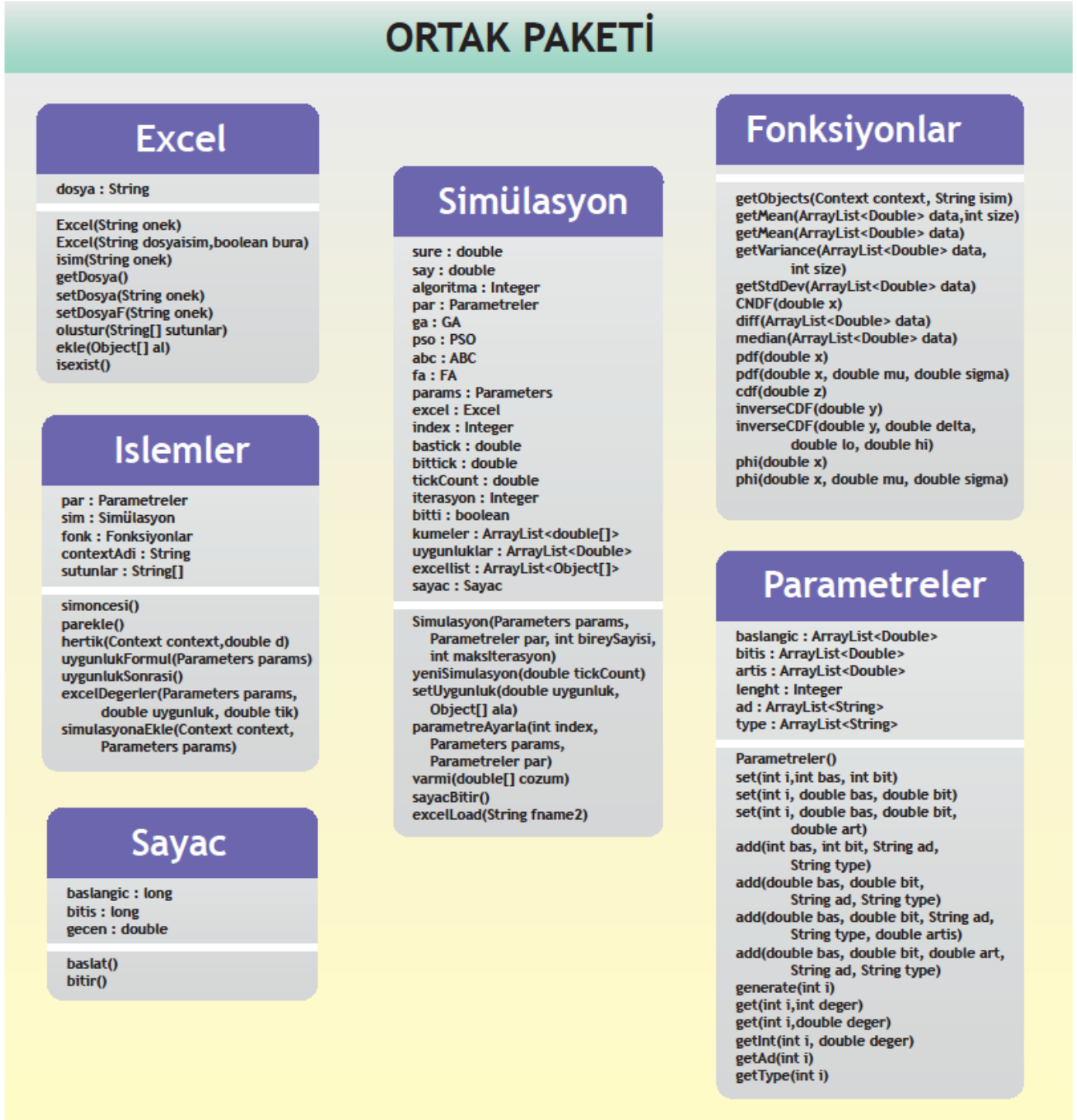


## Ek 2. Geliştirilen Parametre Ayarlama Aracının Akış Diyagramı



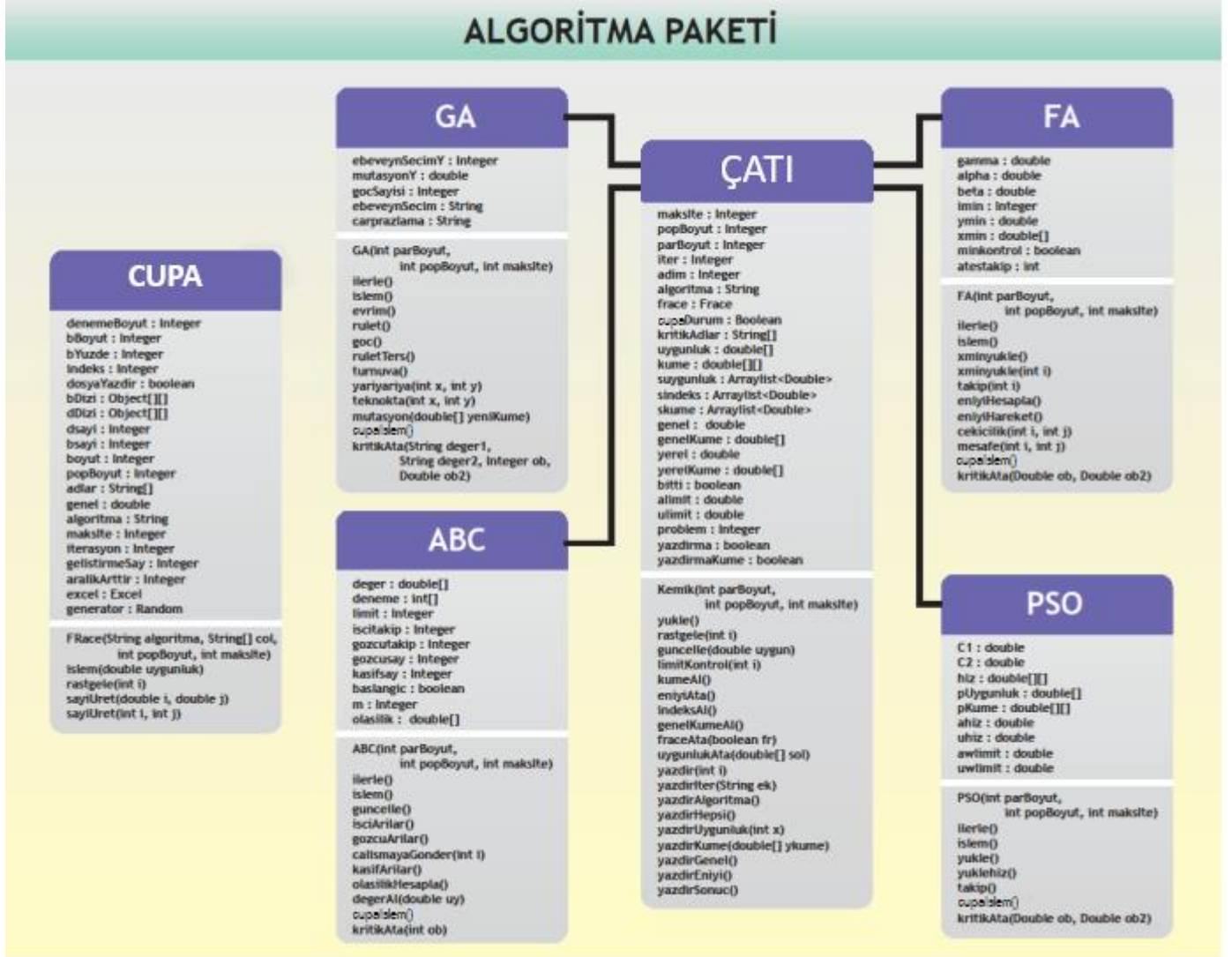


### Ek3. Geliştirilen Parametre Ayarlama Aracı Sınıf Diyagramı-1





## Ek 4. Geliştirilen Parametre Ayarlama Aracı Sınıf Diyagramı-2





## EK 5. TÜRKÇE-İNGİLİZCE TERİMLER SÖZLÜĞÜ

Uyarlanabilir	Adaptive
Eylem	Action
Etmen	Agent
Fedakar	Altruistic
Karınca kolonisi eniyilemesi	Ant colony optimization
Yaklaşım	Approximation
Yapay arı kolonisi	Artificial bee colony
Cazibe	Attractiveness
Özerk	Autonomous
Bakteri	Bacteria
İnanç	Belief
Çapraz değişim	Crossover
Guguk kuşu arama algoritması	Cuckoo search algorithm
Çözümeye yaklaşma hızı	Convergence speed
Varsayılan	Default
Deterministik	Deterministic
Diferansiyel	Differential
İşçi	Employed
Beliren	Emergent
Belirme	Emergence
Eşdeğer	Equivalent
Evrimsel stratejileri	Evolution strategies
Evrimsel hesaplama	Evolutionary computation
Faydalanma	Exploitation
Ateş böceği algoritması	Firefly algorithm

Uygunluk Fonksiyonu	Fitness Function
Uygunluk değeri	Fitness value
Genetik algoritmalar	Genetic algorithms
Sezgisel	Heuristic
Birey	Individual
Yoğunlaştırma	Intensification
Atalet Değeri	Inertia weight
Işık soğurma katsayısı	Light absorption coefficient
Işık yoğunluğu	Light intensity
Yerel	Local
Yerel en iyi	Local optimum
Elle	Manual
En üst seviyeye çekmek	Maximize
Ortalama	Mean
Deniz gözetleme	Maritime surveillance
Meta-sezgisel	Meta-heuristic
Meta-sezgisel algoritmalar	Meta-heuristic algorithms
En alt seviyeye çekmek	Minimize
Model tabanlı	Model Based
Çok değişkenli	Multivariable
Doğrusal olmayan	Non-linear
Çevrimdışı ayarlama	Offline tuning
Çevrimiçi ayarlama	Online tuning
Gözlemci	Onlooker
En iyi	Optimal
Eniyileme	Optimization
Parametre ayarlama	Parameter tuning

Parçacık	Particle
Parçacık sürü eniyilemesi	Particle swarm optimization
Fenomen	Phenomenon
Konum	Position
Gösterim	Representation
Üreme	Reproduction
Özellik	Sample
Ölçekleme faktörü	Scaling factor
Kâşif	Scout
Seçim	Selection
Tavlama benzetimi	Simulated annealing
Benzetim	Simulation
Önem	significance
Çözüm kümesi	Solution set
Rassal	Stochastic
Sürü	Swarm
Tabu arama	Tabu search
Zamanlayıcı	Timer
Ayarlanmış	Tuned
Ayarlama	Tuning
Faydalı	Utility
Hız	Velocity