



BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

**DİNAMİK SİSTEMLERİN ANFİS İLE
MODELLENMESİNDE YAYILIMCI REKABETÇİ
OPTİMİZASYON (ICA) ALGORİTMASININ
KULLANILMASI**

Mehmet SARIKOÇ
Yüksek Lisans Tezi

Danışman
Doç. Dr. Cihan KARAKUZU

BİLECİK, 2014
Ref. No: 10043137



BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

**DİNAMİK SİSTEMLERİN ANFİS İLE
MODELLENMESİNDE YAYILIMCI REKABETÇİ
OPTİMİZASYON (ICA) ALGORİTMASININ
KULLANILMASI**

Mehmet SARIKOÇ
Yüksek Lisans Tezi

Danışman
Doç. Dr. Cihan KARAKUZU

BİLECİK, 2014




BILECIK SEYH EDEBALI UNIVERSITY
Graduate School Of Sciences
Department of Computer Engineering

**USAGE OF IMPERIALIST COMPETITIVE
OPTIMIZATION ALGORITHM (ICA) FOR DYNAMIC
SYSTEMS MODELLING WITH ANFIS**

Mehmet SARIKOÇ
Master's Thesis

Thesis Advisor
Doç. Dr. Cihan KARAKUZU

BİLECİK, 2014

 <p>BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ FEN BİLİMLER ENSTİTÜSÜ</p>	<p>YÜKSEK LİSANS</p> <p><u>JÜRİ ONAY FORMU</u></p>
---	--

Bilecik Şeyh Edebali Üniversitesi Fen Bilimler Enstitüsü Yönetim Kurulu'nun ...05/06/2014... tarih ve ...25/4... sayılı kararıyla oluşturulan jüri tarafından ..19/06/2014... tarihinde tez savunma sınavı yapılan.....Mehmet SARIKOÇ.....'ın "Dinamik Sistemlerin ANFIS ile Modellenmesinde Yayılcı Rekabetçi Optimizasyon (ICA) Algoritmasının Kullanılması" başlıklı tez çalışması Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak oy birliği/oy çokluğu ile kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI) : Doç. Dr. Cihan KARAKUZU



ÜYE : Prof. Dr. Nejat YUMUŞAK



ÜYE : Yrd. Doç. Dr. Mehmet YAKUT



ONAY

Bilecik Şeyh Edebali Üniversitesi Fen Bilimler Enstitüsü Yönetim Kurulu'nun
...../...../..... tarih ve/..... sayılı kararı.

İMZA/MÜHÜR

ÖZET

Yayılcı Rekabetçi Algoritma (Imperialist Competitive Algorithm, ICA), belli bir popülasyondaki ülkelerin emperyal güç olabilme adına birbirileri arasındaki rekabeti konu alan sosyal tabanlı sezgisel bir optimizasyon algoritmasıdır. Bu çalışmada, doğrusal olmayan dinamik sistemlerin ANFIS (Adaptive Neuro-Fuzzy Inference System) bulanık çıkarım modeli kullanılarak modellenmesi problemi üzerine, Yayılcı Rekabetçi Algoritma (ICA) algoritmasının optimizasyon başarımı incelenmiştir. İnceleme için literatürden seçilen örnek dinamik sistemler kullanılmıştır. Her bir örnek dinamik sistemin problem parametreleri, ICA ile ayarlanarak, esnek ve matematiksel yapısı nedeniyle tercih edilen ANFIS ile modellenmiştir.

Her bir dinamik sistem için ICA (Standart ICA, Std-ICA) algoritması ile elde edilen sonuçlar bu çalışma esnasında yine ICA algoritmasından geliştirilen iki yeni algoritmanın (G1-ICA,G2-ICA) sonuçları ile kıyaslanmıştır. Çalışma çerçevesinde elde edilen istatistikî verilere göre G1-ICA algoritmasının başarımının diğer iki ICA algoritmasından daha iyi olduğu görülmüştür. Son olarak, bu çalışmada kullanılan sistemlerin diğer sezgisel algoritmalarla optimize edilmiş ANFIS modellerinin başarımı, bu çalışmada elde edilen ANFIS modellerinin başarımı ile istatistikî olarak incelenmiştir. Bu incelemelere göre ICA algoritmasının istenilene yakın değerler ortaya koyduğu gözlenmiştir.

Anahtar Kelimeler

Yayılcı Rekabetçi Algoritma (ICA), ANFIS, Bulanık Modelleme, Dinamik Sistem

ABSTRACT

Imperialist Competitive Algorithm is a social based heuristic optimization algorithm that deals with the challenge of some countries with certain populations among themselves so as to be the imperial power. In this study, ICA's (Imperialist Competitive Algorithm) optimization performance over the problem of non-linear dynamic system modelling has been studied using ANFIS (Adaptive Neuro Fuzzy Inference System) fuzzy model. For the study, sample dynamic systems selected from the literature have been used. Parameters of each sample dynamic system have been modelled with ANFIS preferred due to its flexible and mathematical structure using ICA optimization algorithm.

For each system, the outcomes obtained with ICA (Standard ICA, Std-ICA) have been compared to the outcomes of the two new algorithms developed from ICA during this study. According to the obtained statistical data, it has been observed that performance of G1-ICA algorithm was better than that of the others to two ICA algorithms. Finally, performance of ANFIS models obtained in this study have been statistical compared with the ANFIS models optimized using the other heuristic algorithms. According to this review, it has been observed that ICA revealed values close to the desired.

Key Words

Imperialist Competitive Algorithm, ANFIS, Fuzzy Modelling, Dynamic System

TEŐEKKÖR

Tez alıőmamda her konuda bilgi ve deneyimleri ile bana yol gsteren danıőman hocam Do.Dr. Cihan KARAKUZU' ya ve her zaman yanımda olan, ilgi ve desteęini hi esirgememiő olan aileme ok teőekkr ederim.

Mehmet SARIKO

Haziran-2014

İÇİNDEKİLER

	Sayfa No
ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
ÇİZELGELER DİZİNİ	vi
ŞEKİLLER DİZİNİ	vii
1.GİRİŞ	1
1.1.Literatür Taraması.....	3
1.2. Tez Çalışmasının Kapsamı.....	10
2. ÇALIŞMADA KULLANILAN SEZGİSEL ARAMA ALGORİTMALARI	11
2.1. ICA Algoritması.....	11
2.1.1. ICA algoritmasının çalışması.....	11
2.1.2. Başlangıç imparatorluklarını üretme.....	13
2.1.3. Bir imparatorluğun kolonilerinin emperyaliste doğru hareketleri.....	14
2.1.4. Emperyalist ve koloninin konum değış tokuşu	15
2.1.5. İmparatorluğun toplam gücü.....	16
2.1.6. Emperyalistlik yarış	16
2.1.7. Yakınsama.....	18
2.2. ICA ile Fonksiyon Optimizasyonu	19
2.2.1."Peaks" fonksiyonu optimizasyonu	20
2.2.2. Örnek 1: F1	25
2.2.3. Örnek 2: F2	29
2.2.4. Örnek 3: F3	33
2.2.5. Geliştirilen ICA algoritmalarının başarımlarını kıyaslaması.....	37
3. KULLANILAN DİNAMİK SİSTEMLER, BULANIK SİSTEM VE SEZGİSEL ÖĞRENME YÖNTEMLERİ	38
3.1. Örnek Dinamik Sistemler	38
3.2. ANFIS Bulanık Çıkarım Sistemi	42
3.3. Kullanılan Algoritma	44
3.4. Bireylerin Yapısı.....	45
3.5. ICA ile ANFIS Parametre Optimizasyonu.....	46
4. SEZGİSEL ARAMA ALGORİTMA TABANLI BULANIK SİSTEM	

OPTİMİZASYONU	47
4.1. ÖDS 1 ‘in Tanınması/Modellenmesi	48
4.1.1.ÖDS 1 için eğitim aşaması ve sonuçları	48
4.1.2.ÖDS 1’in test aşaması ve sonuçları.....	52
4.2. ÖDS 2’ nin Tanınması/Modellenmesi	54
4.2.2.ÖDS 2 için eğitim aşaması ve sonuçları	54
4.2.2.ÖDS 2’ in test aşaması ve sonuçları.....	58
4.3. ÖDS 3 ‘in Tanınması/Modellenmesi	60
4.3.1.ÖDS 3 için eğitim aşaması ve sonuçları	60
4.3.2.ÖDS 3’in test aşaması ve sonuçları.....	64
4.4. ÖDS 4 ‘in Tanınması/Modellenmesi	66
4.4.1.ÖDS 4 için eğitim aşaması ve sonuçları	66
4.4.2.ÖDS 4’in test aşaması ve sonuçları.....	70
4.5. ÖDS 5 ‘in Tanınması/Modellenmesi	72
4.5.1.ÖDS 5 için eğitim aşaması ve sonuçları	72
4.5.2.ÖDS 5’in test aşaması ve sonuçları.....	76
4.6. Std-ICA, G1-ICA ve G2-ICA Algoritmalarının Performansının Kıyaslanması	78
4.7. Literatüre Dayalı Başarım Kıyaslaması	80
5. SONUÇLAR	83
6. KAYNAKLAR	88
EK-1: Standart ICA ile fonksiyon (peaks) optimizasyonunda kullanılan kodlar...	96
EK-2: Literatür kıyaslaması için kullanılan uygunluk/kalite hesaplama fonksiyonu kodları	99
<u>ÖZGEÇMİŞ</u>	100

ÇİZELGELER DİZİNİ

		Sayfa No
Çizelge 2.1.	Geliştirilen ICA (G1-ICA, G2-ICA) algoritmalarının Standart ICA (Std-ICA) algoritması ile başarımların karşılaştırılması.....	37
Çizelge 3.1.	ABC, PSO ve DE öğrenmeli bulanık mantık tabanlı dinamik sistem tanıma/modelleme için literatürden seçilen örnek dinamik sistemler (ÖDS).....	39
Çizelge 3.2.	Her bir ÖDS için kullanılan ANFIS yapısı	40
Çizelge 3.3.	ICA parametreleri.....	45
Çizelge 4.1.	Std-ICA, G1-ICA ve G2-ICA yaklaşımlarının ÖDS modellemede <i>EĞİTİM</i> fazı başarımların karşılaştırılması.....	79
Çizelge 4.2.	Std-ICA, G1-ICA ve G2-ICA yaklaşımlarının ÖDS modellemede <i>TEST</i> fazı başarımların karşılaştırılması	80
Çizelge 4.3.	Eğitim seti için literatüre dayalı başarımların karşılaştırılması.....	82
Çizelge 4.4.	Test seti için literatüre dayalı başarımların karşılaştırılması.....	83
Çizelge 5.1.	Algoritmaların yerel minimuma takılma değerlendirme çizelgesi.....	84

ŞEKİLLER DİZİNİ

	Sayfa No
Şekil 2.1.	Yayılmacı rekabetçi algoritmanın akış diyagramı 12
Şekil 2.2.	İmparatorlukların başlangıç popülasyonları..... 14
Şekil 2.3.	(a) Koloninin emperyaliste hareketi (b) Koloninin yeni pozisyonu..... 15
Şekil 2.4.	(a) Koloni ve emperyalistin yer değişimi (b) Yer değişikliğinden sonraki durum 16
Şekil 2.5.	Emperyalistlik yarışma..... 17
Şekil 2.6.	ICA ile fonksiyon optimizasyonu blok yapısı..... 19
Şekil 2.7.	Peaks fonksiyonu yüzeyi(a) ve eş yükselti eğrileri (b) 20
Şekil 2.8.	Std-ICA algoritması ile “peaks” fonksiyon optimizasyonu için muhtelif adımlardaki birey pozisyonları..... 21
Şekil 2.9.	G1-ICA Algoritması ile Peaks fonksiyon optimizasyonu için muhtelif adımlardaki birey pozisyonları..... 22
Şekil 2.10.	G2-ICA Algoritması ile Peaks fonksiyon optimizasyonu için muhtelif adımlardaki birey pozisyonları..... 24
Şekil 2.11.	F1 fonksiyon yüzeyi(a) ve eş yükselti eğrileri (b) 25
Şekil 2.12.	Std-ICA Algoritması ile F1 fonksiyon optimizasyonu için muhtelif adımlardaki birey pozisyonları..... 26
Şekil 2.13.	G1-ICA Algoritması ile F1 fonksiyon optimizasyonu için muhtelif adımlardaki birey pozisyonları..... 27
Şekil 2.14.	G2-ICA Algoritması ile F1 fonksiyon optimizasyonu için muhtelif adımlardaki birey pozisyonları..... 28
Şekil 2.15.	F2 fonksiyon yüzeyi(a) ve eş yükselti eğrileri (b) 29
Şekil 2.16.	Std-ICA Algoritması ile F2 fonksiyon optimizasyonu için muhtelif adımlardaki birey pozisyonları..... 30
Şekil 2.17.	G1-ICA Algoritması ile F2 fonksiyon optimizasyonu için muhtelif adımlardaki birey pozisyonları..... 31
Şekil 2.18.	G2-ICA Algoritması ile F2 fonksiyon optimizasyonu için muhtelif adımlardaki birey pozisyonları..... 32
Şekil 2.19.	F3 fonksiyon yüzeyi (a) ve eş yükselti eğrileri (b) 33

Şekil 2.20.	Std-ICA Algoritması ile F3 fonksiyon optimizasyonu için muhtelif adımlardaki birey pozisyonları.....	34
Şekil 2.21.	G1-ICA Algoritması ile F3 fonksiyon optimizasyonu için muhtelif adımlardaki birey pozisyonları.....	35
Şekil 2.22.	G2-ICA Algoritması ile F3 fonksiyon optimizasyonu için muhtelif adımlardaki birey pozisyonları.....	36
Şekil 3.1.	ÖDS sistemleri için sistem tanımının <i>eğitim fazında</i> kullanılan giriş (u(k)) dizileri: (a) ÖDS 1 ve 2 için, (b) ÖDS 3 için [-1 1] aralığında rasgele genlikli 10 örnekleme periyotlu darbe (c) ÖDS 4 için [-5 5] aralığında rasgele genlikli ve rasgele örnekleme periyotlu darbe (d) ÖDS 5 için [-2 2] aralığında rasgele genlikli ve rasgele örnekleme periyotlu darbe	41
Şekil 3.2.	ÖDS sistemleri için sistem tanımının <i>test fazında</i> kullanılan giriş (u(k)) dizileri: (a) ÖDS 1 ve 2 için, (b) ÖDS 3 için [-1 1] aralığında rasgele genlikli 10 örnekleme periyotlu darbe (c) ÖDS 4 için [-5 5] aralığında rasgele genlikli ve rasgele örnekleme periyotlu darbe (d) ÖDS 5 için [-2 2] aralığında rasgele genlikli ve rasgele örnekleme periyotlu darbe	42
Şekil 3.3.	İki girişli- tek çıkışlı birinci dereceden kural polinomlu ANFIS mimarisi	44
Şekil 3.4.	ICA ile ANFIS optimizasyonu blok şeması (Yıldırım, 2012)	46
Şekil 4.1.	ICA öğrenmeli ANFIS ile ÖDS 1 için eğitim fazı sistem tanıma sonuçları.	49
Şekil 4.2.	ICA öğrenmeli ANFIS ile ÖDS 1'in bulanık modelleme eğitim fazında UF'lerin başlangıç ve son durumları.....	50
Şekil 4.3.	Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 1 için elde edilen modellerin karşılaştırılması	51
Şekil 4.4.	Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 1 modelleme eğitim seyri.....	52
Şekil 4.5.	ICA öğrenmeli ANFIS ile ÖDS 1 için test fazı sistem tanıma sonuçları.....	53

Şekil 4.6.	ICA öğrenmeli ANFIS ile ÖDS 2 için eğitim fazı sistem tanıma sonuçları.....	55
Şekil 4.7.	ICA öğrenmeli ANFIS ile ÖDS 2'in bulanık modelleme eğitim fazında UF'lerin başlangıç ve son durumları.....	56
Şekil 4.8.	Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 2 için elde edilen modellerin karşılaştırılması.....	57
Şekil 4.9.	Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 2 modelleme eğitim seyri.....	58
Şekil 4.10.	ICA öğrenmeli ANFIS ile ÖDS 2 için test fazı sistem tanıma sonuçları.....	59
Şekil 4.11.	ICA öğrenmeli ANFIS ile ÖDS 3 için eğitim fazı sistem tanıma sonuçları.....	61
Şekil 4.12.	ICA öğrenmeli ANFIS ile ÖDS 3'in bulanık modelleme eğitim fazında UF'lerin başlangıç ve son durumları.....	62
Şekil 4.13.	Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 3 için elde edilen modellerin karşılaştırılması.....	63
Şekil 4.14.	Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 3 modelleme eğitim seyri.....	64
Şekil 4.15.	ICA öğrenmeli ANFIS ile ÖDS 3 için test fazı sistem tanıma sonuçları.....	65
Şekil 4.16.	ICA öğrenmeli ANFIS ile ÖDS 4 için eğitim fazı sistem tanıma sonuçları.....	67
Şekil 4.17.	ICA öğrenmeli ANFIS ile ÖDS 4'in bulanık modelleme eğitim fazında UF'lerin başlangıç ve son durumları.....	68
Şekil 4.18.	Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 4 için elde edilen modellerin karşılaştırılması.....	69
Şekil 4.19.	Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 4 modelleme eğitim seyri.....	70
Şekil 4.20.	ICA öğrenmeli ANFIS ile ÖDS 4 için test fazı sistem tanıma sonuçları.....	71
Şekil 4.21.	ICA öğrenmeli ANFIS ile ÖDS 5 için eğitim fazı sistem tanıma sonuçları.....	73

Şekil 4.22.	ICA öğrenmeli ANFIS ile ÖDS 5'in bulanık modelleme eğitim fazında UF'lerin başlangıç ve son durumları.....	74
Şekil 4.23.	Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 5 için elde edilen modellerin karşılaştırılması	75
Şekil 4.24.	Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 5 modelleme eğitim seyri	76
Şekil 4.25.	ICA öğrenmeli ANFIS ile ÖDS 5 için test fazı sistem tanıma sonuçları.....	77
Şekil 5.1.	Dinamik sistemlerin ANFIS ile modellenmesinde ICA algoritmalarının birbirleriyle grafiksel olarak ortalama ölçüt (a), en iyi ölçüt (b) ve bir nesil süresi (c) değerleri karşılaştırması.....	86
Şekil 5.2.	Dinamik sistemlerin ANFIS ile modellenmesinde popüler algoritmaların ICA algoritması ile grafiksel olarak ortalama ölçüt (a), en iyi ölçüt (b) ve bir nesil süresi (c) değerleri karşılaştırması.....	87

1.GİRİŞ

Gelişen teknoloji ve son yıllar da bilgisayar bilimlerindeki gelişim hayatımızdaki birçok soruna kısa zamanda çözümler getirerek hayat standardımızı yükseltmiştir. Bunu yaparken karşısına gelen her sorun bir öncekinden daha karmaşık ve anlaşılması zor bir hal almıştır. Bu durum insanoğlunu karşılaşılan sorunlara, farklı yaklaşımlarla çözüm yolları bulmaya itmiştir. Farklı konular üzerinde çalışan birçok bilim insanı karşılaştıkları zorlukları karşısında çözüm yollarını, doğa ve canlıların üstesinden geldiği zorluklardan esinlenerek çözüme yoluna gitmiştir. Bu çözüm yollarını gelişen teknolojiler üzerine modelleyerek, karşılaşılan sorunlara yeni çözüm yolları geliştirmişlerdir. Özellikle doğrusal olmayan problemlerin çözümünde yeni yaklaşımlara ihtiyaç duyulmuştur. Bilgisayar destekli çözümlenelerde farklı optimizasyon algoritmaları geliştirilmiş ve bunlar, bazı mühendislik çalışmalarında yol gösterici olmuştur. (Bu yol gösterici çalışmalardan birisi de sezgisel optimizasyon algoritmalarıdır.) Bu çok çeşitli yol gösterici çalışmalardan ikisi; bulanık mantık ve sezgisel optimizasyon algoritmalarıdır.

Bulanık mantık kavramı ilk olarak 1965 yılında Zadeh tarafından ileri sürülmüştür. Araştırmacının kontrol üzerine çalışmalarında çok fazla, doğrusal olmayan denklemlerle uğraşması ve yöntem karmaşası gibi sebeplerden dolayı ortaya çıkmıştır. Buna göre bulanık mantık, insanların günlük konuşma dilini kullanarak, alanında uzman kişilerin deneyimlerini problem çözümüne dâhil ettiği esnek hesaplama yöntemlerinden biridir. Bu hesaplama yönteminin avantajlarından birisi de bir elemanın bir kümeye ait olması, dolayısıyla klasik kümelerdeki gibi 0 ya da 1 şeklinde olmayarak, 0 ile 1 arasında değişen üyelik değerleri ile ifade edilebilmesidir. Kısaca bulanık mantık, günlük yaşamda bazı sorulara basitçe “evet”-“hayır” ya da “var”-“yok” cevabı verilemeyen durumları ifade etmede kullanılır.

Bulanık mantığın bu karmaşık yöntemlerden uzak ve kolay ifade edilebilir hali, yaşam içerisinde gerçekleşen ve çoğu doğrusal olmayan fiziksel sistem davranışlarının modellenmesinde, bilim adamları tarafından kullanılmasını sağlamış, bunun sonucunda gerçeğe yakın sonuçlar elde edilerek uygulamalar kolaylaşmış, doğrusal olmayan sistemlerin başarımları artmıştır.

Sezgisel algoritmalar ise, herhangi bir amacı gerçekleştirmek veya hedefe ulaşmak için doğal olaylardan esinlenen ve çeşitli alternatif hareketlerden etkili olanlara karar vermek amacıyla, tanımlanan kriterler veya bilgisayar metotlarıdır. Bu algoritmaların, çözüm uzayında optimum çözüme yakınsaması ispat edilemez. Yani sezgisel algoritmalar yakınsama özelliğine sahip olmakla birlikte kesin çözümü garanti edemezler ancak bu kesin çözümün yakınlarında bir çözüm garanti edebilirler. Optimizasyon problemlerinin kesin çözümü bulma işleminin tanımlanamayacak yapıda olabilmesi, anlaşılabilirlik yönünden sezgisel algoritmaların karar verici açısından çok daha basit olabilmesi, sezgisel algoritmaların öğrenme amaçlı ve kesin çözümü bulma işleminin bir parçası olarak kullanılabilir olması gibi nedenlerle bu algoritmaya ihtiyaç duyulmaktadır (Karaboğa, 2004).

Genel amaçlı sezgisel yöntemler; biyoloji tabanlı, fizik tabanlı, sürü tabanlı, sosyal tabanlı, müzik tabanlı ve kimya tabanlı olmak üzere altı farklı grupta değerlendirilmiştir. Ayrıca bunların birleşimi olan melez yöntemler de bulunmaktadır. Genetik algoritma (GA), diferansiyel gelişim algoritması, karınca koloni algoritmaları, yapay sinir ağları, arı koloni algoritmaları ve yapay bağışıklık sistemleri biyolojik tabanlı algoritmalar; emperyalist yarışmacı algoritma, parlamenter optimizasyon algoritması ve tabu arama sosyal tabanlı algoritmalarıdır. Yapay kimyasal reaksiyon algoritması kimya tabanlı; armoni arama algoritması müzik tabanlı; ısıl işlem, büyük patlama büyük sıçrama, yer çekimsel arama algoritması, merkez kuvvet optimizasyonu, zeki su damlacıkları algoritması ve elektromanyetizma algoritması fizik tabanlı algoritmalar iken Parçacık Sürü Optimizasyonu (PSO), KSO ise sürü tabanlı algoritma ve modellerdir. Kültürel algoritmalar hem biyoloji hem de sosyal tabanlı algoritma olarak sınıflandırılabilir. (Alataş, 2007; Akyol ve Alataş, 2012).

Genel itibari ile günümüzde karşılaşılan 4 farklı sosyal tabanlı sezgisel optimizasyon algoritması bulunmaktadır. Bunlardan, üzerinde en çok çalışma yapılanı tabu arama algoritmasıdır (Glover, 1989). Diğerleri ise son yıllarda geliştirilmiş algoritmalar ki bunlar: Yayılımcı rekabetçi algoritma (Atashpaz-Gargari ve Lucas, 2007), parlamenter optimizasyon algoritması (Borji, 2007) ve öğretme-öğrenme tabanlı algoritma (Rao vd., 2012).

Yayılcı rekabetçi algoritma (Imperialist competitive algorithm-ICA), 2007 yılında Atashpaz-Gargari ve Lucas tarafından ortaya atılan sosyal tabanlı sezgisel bir optimizasyon algoritmasıdır. Bu algoritma, dünya üzerindeki ülkeleri belirli sayıdaki imparatorluklara dağıtan ve bu imparatorluklar arasında tek dünya düzenini amaçlayan bir emperyalistlik yarışın modellenmesidir.

2007 yılında ortaya çıkan yeni bir algoritma olmasına rağmen ilerleyen yıllar içerisinde üzerinde birçok geliştirmeler ve farklı konularda çalışmalar yapılmıştır. Buna rağmen, bir sonraki bölümde verilecek literatür özetinden de anlaşılacağı gibi, algoritma henüz bulanık bir sistemin (özellikle ANFIS) optimizasyonunda kullanılmamıştır.

Bu çalışma ile, ilk olarak ICA algoritmasının işleyişinin kavranması daha sonrasında bulanık işleyişi gerçekleyen ANFIS (Adaptive Neuro-Fuzzy Inference System) bulanık mantık modeli kullanılarak; ICA algoritmasının doğrusal olmayan dinamik sistemlerin bulanık modellenmesinde optimizasyon başarımının incelenmesi amaçlanmıştır.

1.1.Literatür Taraması

Atashpaz-Gargari ve Lucas (Atashpaz-Gargari ve Lucas, 2007) tarafından geliştirilen ICA algoritması ile ilgili olarak 2007 yılından günümüze değin çeşitli çalışmalar yapılmıştır. Aşağıda verilen literatür taramasına göre bu güne kadar ICA algoritması tabanlı bulanık mantık ile ilgili bir çalışma yapıldığına rastlanılmamıştır. ICA algoritması ile yapılan çalışmalardan bazılarının kısa özeti aşağıda verilmiştir.

Kaveh ve Talatahari (Kadeh ve Talathari, 2010) çalışmalarında ICA' yı iskelet yapılarını optimize etmek için kullanmışlardır. Elde edilen sonuçları, bilinen bazı meta-sezgisel algoritmalarla karşılaştırarak ICA algoritmalarının etkinliğini göstermişlerdir.

Duan vd. (Duan vd., 2010) çalışmalarında ICA algoritmasından geliştirilen Kaotik ICA ile şablon eşleştirme işlemi için kullanmışlardır. Kaotik ICA ile küresel yakınsama artırabildiği ve yerel en iyi çözüme düşme olayları önlenebildiği görülmüştür.

Nazari-Shirkouhi vd. (Nazari-Shirkouhi vd., 2010) kurumsal imalatta önemli bir sorun olan entegre ürün karması-dış kaynak optimizasyonu için meta-sezgisel bir

yöntem olan ICA algoritmasını kullanılmışlar ve sonuçları kısıtlar teorisi ve standart hesaplama ile karşılaştırmışlardır.

Niknam vd. (Niknam vd., 2011) çalışmalarında, K kümeleri içinde optimum kümelenen N nesnelere için K-MICA olarak adlandırılan, değiştirilmiş yayılımcı rekabetçi algoritma ve k-means birleşimine dayalı hibrit evrimsel optimizasyon üzerine çalışmışlardır. Bu çalışma ile elde edilen sonuçlar, ACO, PSO, SA, GA, TS, HBMO gibi algoritmalar ile karşılaştırılmıştır.

Mohammadi vd. (Mohammadi vd., 2011) hub yerleşim problemlerinin optimizasyonunda, en uygun yakın çözümlerini bulmak için gelişmiş bir meta sezgisel temele dayalı yayılımcı rekabetçi algoritma ve genetik algoritma kullanmışlardır. Elde edilen sonuçlar her iki algoritma arasında karşılaştırılmıştır.

Talatahari vd. (S. Talatahari vd., 2012) çalışmada yeni bir kaotik gelişmiş emperyalist rekabet algoritmasını (CICA) global optimizasyon için sunmuşlar ve farklı kaotik haritalar, algoritma hareket adımını geliştirmek için kullanılmışlardır.

Naimi vd. (Naimi vd., 2012) üretici-perakendeci tedarik zinciri koordinasyonu için iki seviyeli programlama yaklaşımı üzerine çalışmışlardır. Bu yaklaşımla ideal denge fiyatları, reklam harcamaları ve üretim politikalarını bulmak için ICA, değiştirilmiş ICA algoritması ve evrim strateji dahil olmak üzere birçok çözüm prosedürleri üzerinde çalışılmışlardır.

Mozafari vd. (Mozafari vd., 2012a) çalışmalarında yayılımcı rekabetçi algoritma kullanarak burkulma yükü fonksiyonel dereceli plakaların optimizasyonu üzerine çalışmışlardır. Sonuçta, ICA sonuçları, genetik algoritma (GA) elde edilenler ile karşılaştırılmıştır.

Mozafari vd. (Mozafari vd., 2012b) bir başka çalışmalarında ince dirençli interfaz optimize etmek için ICA uygulaması konusunu seçmişlerdir. ICA sonuçları FEM (Finite Element Method) ve GA (Genetic Algorithm) ile kıyaslanmış, buna göre ICA sonuçları ince bağlı katmanların ideal dizaynının da daha iyi performans göstermiştir.

Mozafari vd.'nin (Mozafari vd., 2012c) bu çalışmaları, iki malzeme tabakası arasında farklı şartlar arası bağlanan epoksi yapışkan tabakanın optimizasyonunun ICA

ile uygulamasını göstermektedir. ICA sonuçları, sonlu elemanlar metodu (FEM) ve genetik algoritma (GA) ile karşılaştırılmıştır.

Karami vd. (Karami vd., 2012) çalışmalarında yayılımcı rekabetçi algoritma (ICA) temelli yeni bir optimizasyon algoritmasını hava ısı transferini klasik bükülmüş bant uçları ile donatılmış soğutmalı ısı değiştiriciyi optimize etmek için kullanmış, basitlik, doğruluk ve zaman tasarrufu gibi bazı avantajlar sağladığı gözlemlemiştir.

Ebrahimzadeh vd. (Ebrahimzadeh vd., 2012) K-MICA kümeleme ve nöral ağları kullanarak kontrol grafiği örüntü tanıma üzerine çalışmışlardır. Bu teknik, değiştirilmiş yayılımcı rekabetçi algoritmanın (MICA) ve K-Means algoritmasının uygun bir kombinasyonudur.

Yousefi vd. (Yousefi vd., 2012) plaka fin ısı değiştiricinin optimizasyonu için ICA uygulaması üzerine çalışmışlardır. Nümerik sonuçlara göre, GA ile kıyaslandığında ICA en ideal düzenlemeleri bulmuş, yüksek geçerlilikte hesaplama zamanları elde etmiştir.

Ravadanagh ve Roshanagh (Ravadanagh ve Roshanagh, 2013) bu çalışmalarında orta gerilim trafonun optimum boyutlandırma, konumlama ve zamanlama çözümü, geniş dağıtım ağının optimum genişleme planlaması için ICA uygulaması önermişlerdir. Küresel çözüme ulaşmak amacıyla, ICA parametresi için etkili bir kodlama geliştirilmiştir.

Azami vd. (Azami vd., 2013) çalışmalarında durağan olmayan işaretlerin segmentasyonu için bir melez evrimsel bir yaklaşım önermişlerdir. Buna göre evrimsel algoritmalar ve FD' ye (Fractal Dimension) dayalı gelişmiş bir bölümlenme yöntemi, durağan olmayan sinyaller için önerilmiş, FD' nin (Fractal Dimension) iki kabul edilebilir parametresini seçmek için iki yetkili evrimsel algoritma yani, genetik algoritma (GA) ve yayılımcı rekabetçi algoritmayı (ICA) kullanılmışlardır.

Goldansaz vd. (Goldansaz vd., 2013) çoklu işlemci açık atölye çıkışını minimize etmek için melez emperyalist rekabet algoritması önermişlerdir. Genetik algoritma (GA) ile melez yayılımcı rekabetçi algoritmasını (ICA) bu sorunun çözümü için sunmuşlardır.

Razmjoooya vd. (Razmjoooya vd., 2013) Bu çalışmasında ten rengini bölümlendirme için melez bir hibrid sinir ağı yayılımcı rekabetçi algoritması

önermişlerdir. Önerilen algoritmada, çok katmanlı algılayıcı ağ (MLP) problemin kısıtlarını yönetirken, ICA algoritması, yüksek kaliteli çözümler ve minimum maliyet için arama yapmada kullanılmıştır.

Taher ve Bagherpour (Taher ve Bagherpour, 2013) çalışmalarında veri yollarının gerilimini ve toplam harmonik bozulmasını kabul edilebilir bir aralıkta koruyarak güç sistemi kayıplarını ve dengesizliklerini en aza indirmeyi ve ardından gelen ağ tasarrufunu en üst düzeye çıkarmayı amaçlayan, bir hibrit bal arısı koloni optimizasyon algoritmasını kullanarak optimum kapasitör yerleşimi amaçlamışlardır. Elde edilen sonuçlar, diğer yapay zeka teknikleri olan GA, PSO ve ICA algoritmaları ile kıyaslanmıştır.

Zarandi vd. (Zarandi vd., 2013) bu çalışmada, yayılımcı rekabetçi algoritma ve benzetilmiş tavlamanın hibritlenmesiyle ayarlanmış yeni bir bulanık fonksiyonlar modelinin hisse senedi fiyat tahmininde uygulanmasını gerçekleştirmişlerdir. Bu model mahalnobise mesafe ölçütü ve gürültü ret yöntemi ile olasılıklı c-mean kümelemenin genişletilmiş versiyonunun hibridizasyona dayanmaktadır. Yayılımcı rekabetçi algoritma (ICA) kümeleme parametrelerini başlatmak için kullanılmıştır.

Mousavi vd. (Mousavi vd., 2013) çalışmada, yeni ürün geliştirme projelerinde zaman tahmini için destek vektör regresyonu ve yayılımcı rekabetçi algoritmanın yeni bir modelini sunmuşlardır. Burada yayılımcı rekabetçi algoritma, destek vektör regresyonunun parametrelerini ayarlamak için kullanılmıştır.

Shabani vd. (Shabani vd., 2013) çalışmalarında güç sistemlerinde yük-frekans kontrolü için yayılımcı rekabetçi algoritmaya dayalı bir PID denetleyici tanıtmışlardır.

Enayatifar vd. (Enayatifar vd., 2013) bu çalışmada, görüntü şifreleme için kaotik harita ile ağırlıklı bir ayırık yayılımcı rekabetçi algoritma (Weighted Discrete Imperialist Competitive Algorithm-WDICA) birlikte kullanılmıştır. Önerilen yöntemde, öncelikle kaotik harita belirtilen sayıda şifre görüntüleri oluşturmak için kullanılmış daha sonra sonucu iyileştirmek için, WDICA, şifre görüntülere uygulanmıştır.

Moradia ve Zandieh (Moradia ve Zandieh, 2013) bu çalışmada üretim hızının varyasyonları minimize edecek tam zamanlı (JIT) sıralama sorununa yeni bir yayılımcı rekabetçi algoritma (ICA) sunmuşlardır.

Nourmohammadia vd. (Nourmohammadia vd., 2013) çok amaçlı U-tipi montaj hattı dengeleme problemi (UALBP) için, sosyo-politik evrim sürecinden esinlenerek oluşturulmuş olan yayılımcı rekabetçi algoritmayı (ICA) önermişlerdir. Elde edilen sonuçlar, GA ile kıyaslanmış ve başarımının daha yüksek olduğu görülmüştür.

Ahmadia vd. (Ahmadia vd., 2013) bu çalışmada, kuyuların petrol oranı tahmini için bulanık mantık, yapay sinir ağları (YSA) ve yayılımcı rekabetçi algoritmaya dayanan yeni bir yöntem sunmaktadırlar.

Abdollahi vd. (Abdollahi vd., 2013) bu çalışmada, doğrusal olmayan denklem sistemlerinin çözümü için yayılımcı rekabetçi algoritmayı (ICA) kullanmışlardır.

Afonso vd. (Afonso vd., 2013) bu çalışmada, arama sırasında daha iyi çözümler için çekim ve itme kavramını uygulayarak ICA' da bir geliştirme yapmışlar ve AR-ICA yaklaşımı önermişlerdir.

Mohammadi vd. (Mohammadi vd., 2013) bu çalışmada, belirsizlik altında merkezi kapsama yer sorunu için yeni bir stokastik çok amaçlı çok modlu taşımacılık modeli geliştirmişlerdir. Yeni bir, çok amaçlı yayılımcı rekabetçi algoritma (MOICA), problemin Pareto uygun çözümlerini elde etmek için önerilmiştir.

Amiri-Arefa vd. (Amiri-Arefa vd., 2013) bu çalışmalarında, olasılıksal bir çizgi bariyer ile merkez yere bağımlı tehcir sorunu üzerine çalışmışlardır. Çalışmada kullanılan optimizasyon yazılımı (MINLP), optimum çözümü bulamayan, zaman sınırlamalı büyük boyutlu test problemleri için genetik algoritma (GA) ve yayılımcı rekabetçi algoritma (ICA) gibi iki meta-sezgisel temelli algoritmayı uygulamaktadır.

Rahimi vd. (Rahimi vd., 2013) bu çalışmada, proje tamamlanma zamanını en aza indirmek için kaynak ve kimlik modu kısıtlamaları altında proje programlama sorununu çözmek için yayılımcı rekabetçi algoritma, tavlama benzetimi ve ayırıcı evrim olmak üzere üç meta-sezgisel algoritma önermişlerdir. Kullanılan algoritmaların kalitesini artırmak amacıyla bir yerel arama ve öğrenme modülü, meta-sezgisel algoritmalar ile birleştirilmiştir.

Mirhoseini vd. (Mirhoseini vd., 2014) bu çalışmalarında, gerçek güç kayıplarını azaltmak ve voltaj profilini artırmak için yeni bir geliştirilmiş uyarlanabilen yayılımcı rekabetçi algoritmaya dayalı tekrar yapılandırma metodu ileri sürmektedirler. ICA' yı

yeniden yapılandırma problemine adapte etmek için eşleştirme sistemi kullanılmakta ve ICA' nın araştırma kapasitesini ve yaklaşma hareketini artırmak için keşifsel bir yasaklanmış bölge metodu ileri sürülmektedir.

Mahari ve Zare (Mahari ve Zare, 2014) bu çalışmada büyük ölçekli güç sistemlerindeki nesil zamanlaması (GS) problemine olası en uygun çözüme karar vermek için ICA' ya dayanan yeni bir çözüm sunmaktadırlar. Oldukça sıkıntılı olan bu GS problemini çözmede, ICA' nın asimile operatörleri ve yeni ülkeleri geliştiren, geliştirilmiş bir ICA versiyonu (MICA) kullanılmıştır.

Moradi vd. (Moradi vd., 2014) bu çalışmada, dağıtılan nesil (DG) kaynaklarını ve kapasitör banklarının eş zamanlı yerleştirimini, ölçüm problemleriyle büyük ölçüde öngörebilen, ICA ve GA ya dayanan etkili bir karma metot sunmaktadırlar. Buna göre elde edilen sonuçlar GA ve PSO algoritmaları ile karşılaştırılmıştır.

Nemati vd. (Nemati vd., 2014) bu çalışmada, başlama ve sınır değer sorunlarını çözmek için yayılımcı rekabet algoritma (ICA) ileri sürmektedirler. Böylece kısıtlanmış bir problem, ICA ile optimize edilmiş doğru bir uygunluk fonksiyonu tanımlamak için ceza yöntemi kullanımıyla, kısıtlanmamış bir sorun haline dönüştürülmektedir.

Navaei vd. (Navaei vd., 2014) bu çalışmada, elde tutma ve gecikme maliyetleri toplamını en aza indirmek için ikinci aşamadaki özdeş olmayan montaj makineleri ile iki aşamalı seri üretim planlaması problemine işaret etmektedirler. Bu sorunun çözümünde işaret edilen problemi çözmek için matematiksel model genişletildikten sonra, içerisinde ICA' nın da olduğu dört hibrit meta sezgisel yaklaşım geliştirilmiştir.

Huanga vd. (Huanga vd., 2014) bu çalışmada parçacık sürü optimizasyonu (PSO) ve karınca kolonisi optimizasyonu (ACO) gibi biyo-ilham akıllı algoritmaları, görüntü eşleştirme sorunlarını çözmek için uygulamışlardır. Bununla beraber karmaşık görüntü eşleştirme sorunlarını çözmek için yanal engelleme mekanizması ile yayılımcı rekabetçi algoritmanın (ICA) birleştirilmesi ile oluşan melez bir biyo-ilham optimizasyon yaklaşımını önermişlerdir.

Duan ve Huang (Duan ve Huang, 2014) bu çalışmada yayılımcı rekabetçi algoritma (ICA) tarafından eğitilmiş yapay bir sinir ağına (ANN) dayanan, UCAV' ın (Unmanned combat aerial vehicle) küresel optimum yol planlaması için yeni bir melez yöntem önermişlerdir. Suni arı kolonisi (ABC) algoritması ile karşılaştırmalı deney

sonuçları, önerilen yaklaşımın, sadece olasılık modelinden kaynaklanan evrimsel hesaplama belirsizliğini azaltmadığı, aynı zamanda çok daha yüksek hızı ile yerel noktaya düşmeyi önleyebildiğini göstermiştir.

Ravadanegh ve Gholizadeh (Ravadanegh ve Gholizadeh, 2014) bu çalışmada, elektrik dağıtım ağının ideal genişleme planlaması için geliştirilen yeni bir yayılımcı rekabetçi algoritma (ICA) uygulamasını sunmaktadırlar. Sistem parametreleri, varlık yönetimi ve coğrafi kısıtlamaların dinamik davranışını düşünmek için çok aşamalı bir genişleme planı önermişler, küresel çözüme ulaşmak amacıyla ICA parametreleri ile etkin bir kodlama geliştirmişlerdir.

Ghasemi vd. (Ghasemi vd., 2014) bu çalışmada, düzgün olmayan maliyet fonksiyonlarının optimum güç akış (OPF) problemi için öğretim öğrenme algoritmasını (TLA) ve yeni bir karma yayılımcı rekabetçi algoritmayı (MICA) önermektedirler. Buna göre OPF kontrol değişkenlerinin ideal ayarları için MICA ve TLA' nın birleştirilmesi elde edilen (MICA-TLA) hibrit bir algoritma kullanılmıştır.

Devika vd. (Devika vd., 2014) bu çalışmada, üçlü alt çizgi yaklaşımına dayalı sürdürülebilir bir kapalı döngü tedarik zinciri ağ tasarımı için meta-sezgisel melezleme tekniklerini karşılaştırmışlardır. Bunu sorunun çözümünde değişken komşuluk arama ve adaptif yayılımcı rekabetçi algoritmaya dayalı üç yeni bir hibrit meta-sezgisel yöntem geliştirmişler ve kullanmışlardır.

Taher vd. (Taher vd., 2014) bu çalışmada elektrik enerjisi sistemlerinde LFC için kesir dereceli PID kontrolör tasarımında yayılımcı rekabetçi algoritmayı kullanmayı önermişlerdir. Çalışmadaki kontrolör, ayarlı beş parametreye sahiptir bu nedenle, geleneksel PID ile karşılaştırıldığında fazladan iki serbestlik derecesi sağlamaktadır. İşte bu kontrolör parametrelerinin uygun ayarlanması amacıyla yayılımcı rekabetçi algoritma (ICA) kullanılmıştır.

Morshed ve Asgharpour (Morshed ve Asgharpour, 2014) bu çalışmada güç sistemi ekonomik yük dağıtım (ELD) sorunu çözmek için yayılımcı rekabetçi algoritma (ICA) ve ardışık kuadratik programlama (SQP) tekniğinden oluşan melez algoritmaya dayalı yeni bir yöntem (HIC-SQP) sunmuşlardır. ICA' nın güçlü yanlarına rağmen emperyalist sayılarının artmasıyla özellikle yerel optimumda tuzaklar oluşmasına neden

olabilmektedir. Bu sakıncayı hafifletmek içinde SQP tekniđi, güveni artırmada ICA sonuçlarına ince ayarlamalar için kullanılmıştır.

1.2. Tez Çalışmasının Kapsamı

Yukarıda kısaca ICA ile yapılan literatür çalışmaları özetlenmiştir. Yapılan özetten de anlaşılacağı üzere ICA algoritması dinamik sistemlerin bulanık modellenmesinde henüz kullanılmamıştır. Bu tez çalışmasındaki amaç, ICA algoritmasını öğrenmek, işleyişini kavramak varsa eksikliklerini giderici yönde geliştirme yapmak ve ICA algoritmasını dinamik sistem modelleme problemi üzerinde, bulanık model (ANFIS) elde etmek amacıyla sınınamaktır. Çalışmada ICA' nın başarımı bilinen diğer sezgisel arama algoritmaları ile kıyaslanmıştır. Bu çerçevede yapılan çalışmaların anlatıldığı bu tez kitabının organizasyonu şu şekildedir: İkinci bölümde bu çalışmanın ilk konusu olan, ICA' nın işleyişi örnek matematiksel bir fonksiyonun (peaks) optimizasyonu üzerinde detaylıca anlatılmış ve standart algoritmanın geliştirilmesinden elde edilen iki yeni yaklaşımla birlikte 3 farklı matematiksel fonksiyon optimizasyonu üzerinde çalışmaları örneklendirilmiştir. Üçüncü bölümde kullanılan temel bulanık model olması sebebiyle ANFIS ve üzerinde çalışılan örnek dinamik sistemler tanıtılmıştır. Dördüncü bölümde bulanık mantık tabanlı dinamik sistem modelleme ve bu modelin sezgisel optimizasyonu üç farklı algoritma için kıyaslamalı olarak irdelenmiştir. Son bölümde elde edilen sonuçların yorumu ve ileriye dönük yapılabilecek çalışmalar üzerinde durulmuştur.

2. ÇALIŞMADA KULLANILAN SEZGİSEL ARAMA ALGORİTMALARI

Bu bölümde, ICA algoritmasının çalışma mantığını anlatılmış ve *tepeler (peaks)* fonksiyonu üzerinde çalışması incelenmiştir. Algoritmanın fonksiyon optimizasyonundaki başarımını artırmak amacıyla iki farklı yaklaşımla iki yeni geliştirilmiş ICA algoritması tanımlanmış ve bu algoritmalar *tepeler (peaks)* fonksiyonu haricinde üç farklı fonksiyon üzerinde de işletilerek yerel minimuma takılıp takılmama açısından başarım değerlendirilmesi yapılmıştır.

2.1. ICA Algoritması

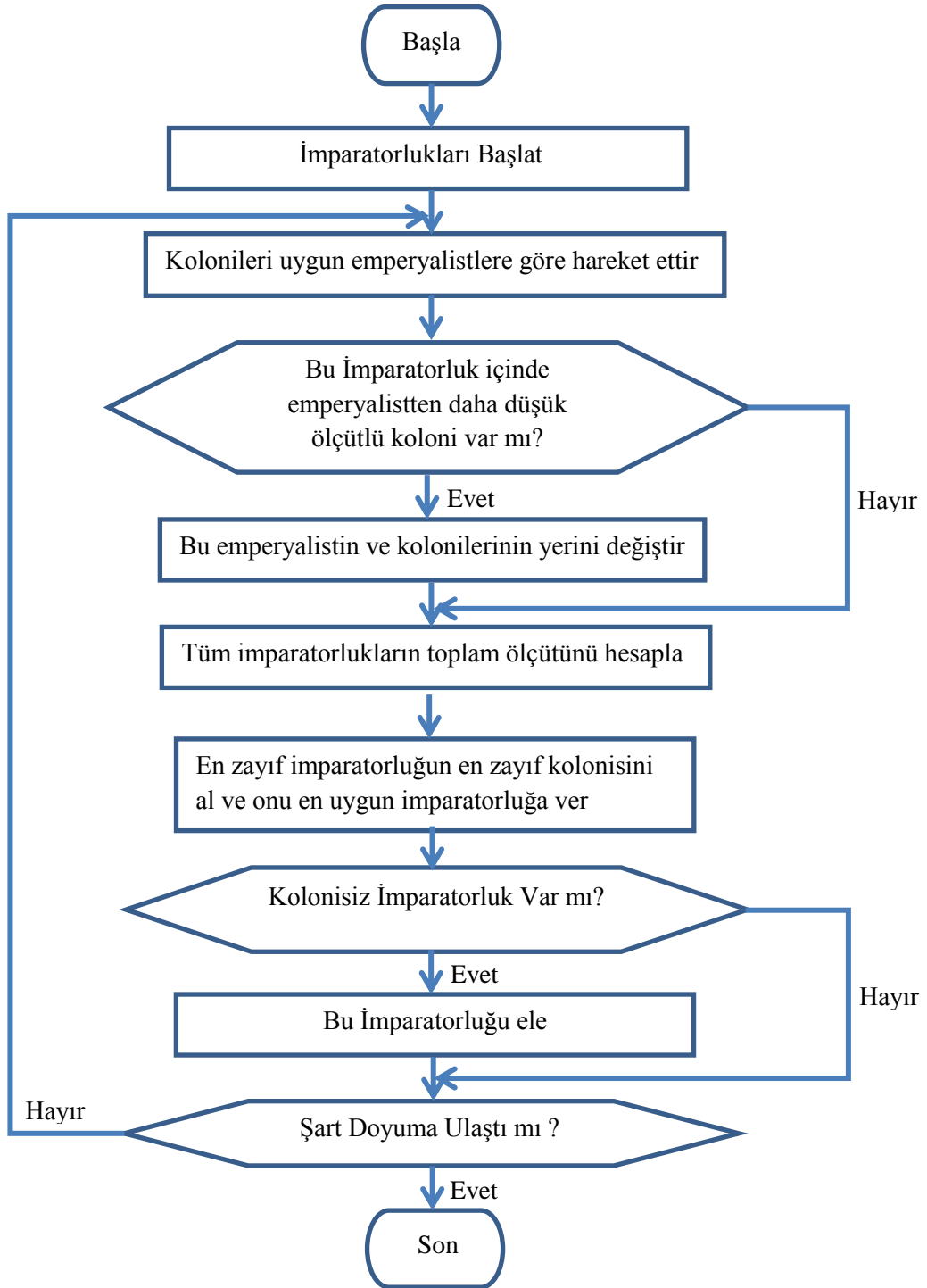
2.1.1. ICA algoritmasının çalışması

Diğer evrimsel algoritmalarından biri gibi yayılımcı rekabetçi algoritma da (ICA) bir başlangıç popülasyonu ile başlar. Bu popülasyondaki her birey dünya üzerindeki bir ülkeyi temsil etmektedir. Popülasyondaki en iyi ülkelerden bazıları emperyalist olmak için seçilir ve arta kalanlar da bu emperyalistlerin kolonileri olur. Başlangıç popülasyonundaki tüm koloniler adı geçen emperyalistler arasında güçleri doğrultusunda paylaşılır. Bir imparatorluğun gücü GA daki uygunluk değerinin benzeri, onun ölçütünün nispeten tersidir.

Tüm koloniler emperyalistler arasında paylaşıldıktan sonra, bu koloniler ilgili emperyalistlere doğru hareket etmeye başlar. Bir imparatorluğun toplam gücü o emperyalist ve ona bağlı koloniler olmak üzere her ikisinin gücüne bağlıdır. Burada bir imparatorluğun toplam gücü, emperyalist ülkenin gücüne, sahip olduğu kolonilerinin ortalama güçlerinin belirli bir yüzdesini ekleyerek tanımlanan bir gerçeği modellemektedir.

Daha sonra tüm imparatorluklar arasında emperyalistlik yarış başlar. Herhangi bir imparatorluk bu yarışta gücünü arttıramaz (veya gücünün en aza indirilmesine engel olamaz) ve başarılı olamazsa bu yarıştan elenir. Emperyalistlik yarış, sonuçta derece derece daha zayıf imparatorluklardan birinin gücünü azaltacak ve güçlü imparatorlukların gücünü arttıracaktır. Zayıf imparatorluklar güçlerini kaybedecek ve en sonunda güçlüler onları ele geçirecektir. İmparatorluklar arasındaki yarış ile kolonilerin uygun emperyaliste doğru bu hareketleri ve ele geçirme mekanizması umutla tüm ülkelerin aynı duruma dönüşmesine böylece dünya üzerinde tek bir imparatorluk kalmasına diğer tüm ülkelerin de o imparatorluğun kolonisi olmasını neden olacaktır.

Bu yeni dünya idealinde koloniler, emperyalist gibi güce ve aynı konuma sahip olacaktır. Şekil 2.1' de algoritmanın akış şeması görülmektedir (Atashpaz-Gargari ve Lucas, 2007:4662).



Şekil 2.1. Yayılımcı rekabetçi algoritmanın akış diyagramı

2.1.2. Başlangıç imparatorluklarını üretme

Optimizasyonun amacı problem değişkenlerinden en optimal çözümü elde etmektir. Optimize edilecek değişken değerleri için bir dizi oluşturulur. Genetik algorithmada bu “kromozom” olarak isimlendirilirken, yayılımcı rekabetçi algoritma da buna “ülke” adı verilir. Bir N_{var} boyutlu optimizasyon probleminde bir ülke $1 \times N_{var}$ dizisi ile ifade edilir. Bu dizi E.2.1’ deki gibi tanımlanır.

$$country = [p_1, p_2, \dots, p_{N_{var}}] \quad (E.2.1)$$

Ülkedeki değişken değerleri noktalı kayan sayılarla temsil edilir. Bir ülkenin ölçütü (Cost) $u_1, u_2, \dots, u_{N_{var}}$ değişkenlerindeki değer fonksiyonu ile ölçülür ve E.2.2 deki gibi tanımlanır.

$$Cost = f(country) = f(p_1, p_2, \dots, p_{N_{var}}) \quad (E.2.1)$$

Optimizasyon algoritmasına başlamak için N_{pop} boyutundaki başlangıç popülasyonu oluşturulmalıdır. İmparatorlukları oluşturmak için N_{imp} sayıdaki en güçlü ülkeler seçilmelidir. Geri kalan N_{col} ise bir imparatorluğa ait olan kolonileri gösterir. Bu şekilde emperyalist ve koloni olmak üzere 2 tip ülkeye sahip olunmaktadır.

Başlangıç imparatorluklarını oluşturmak için koloniler, emperyalistler arasında emperyalistlerin güçleri doğrultusunda dağıtılır. Böylece emperyalistlerin başlangıç koloni sayısı gücü ile doğru orantılı olmaktadır. Kolonileri emperyalistler arasında doğru orantılı dağıtmak için, bir emperyalistin normalize edilmiş ölçütü hesaplanır ve E.2.3’ te gösterildiği gibi tanımlanır:

$$C_n = c_n - maks_i\{c_i\} \quad (E.2.3)$$

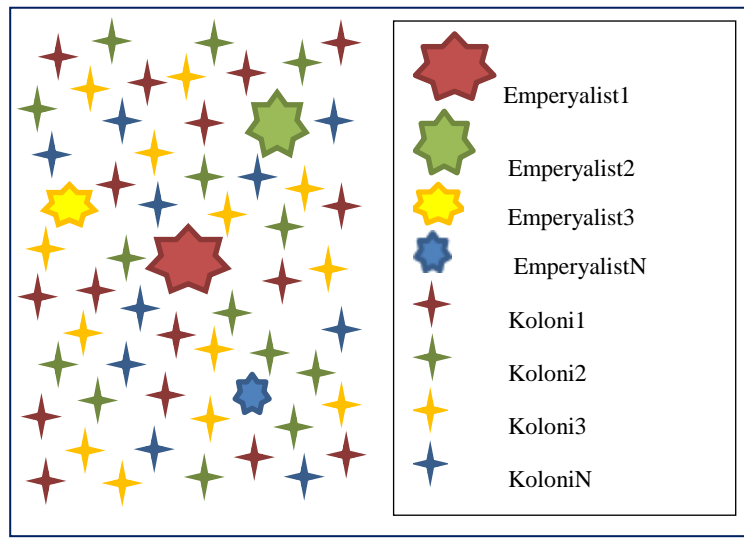
C_n , n. emperyalistin maliyeti ve c_n ise normalize edilmiş maliyetini ifade eder. Tüm emperyalistlerin normalize edilmiş ölçütlerine göre, her emperyalistin normalize edilmiş gücü ise E.2.4’ te gösterildiği gibi hesaplanır:

$$p_n = \left| \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \right| \quad (E.2.4)$$

Bir başka görüşe göre, bir emperyalistin sahip olduğu koloniler emperyalistin normalize edilmiş gücüdür. Böylece bir imparatorluğun kolonilerinin başlangıç sayısı E.2.5’ teki gibidir.

$$N.C._n = \text{round} \{p_n \cdot N_{col}\} \quad (\text{E.2.5})$$

$N.C._n$ n. İmparatorluğun başlangıç koloni sayısını, N_{col} tüm kolonilerin sayısını vermektedir. Kolonileri emperyalistlere dağıtmak için gelişigüzel $N.C._n$ koloni seçilmekte ve emperyaliste verilmektedir. Bu koloniler emperyalist ile birlikte n. imparatorluğu meydana getirir. Şekil 2.2 her imparatorluğun ilk popülasyonunu göstermekte ve daha güçlü imparatorlukların daha çok koloniye sahip olduğu görülebilmektedir. (Abdechiri vd., 2010:941; Kızılloluk ve Alataş, 2012).



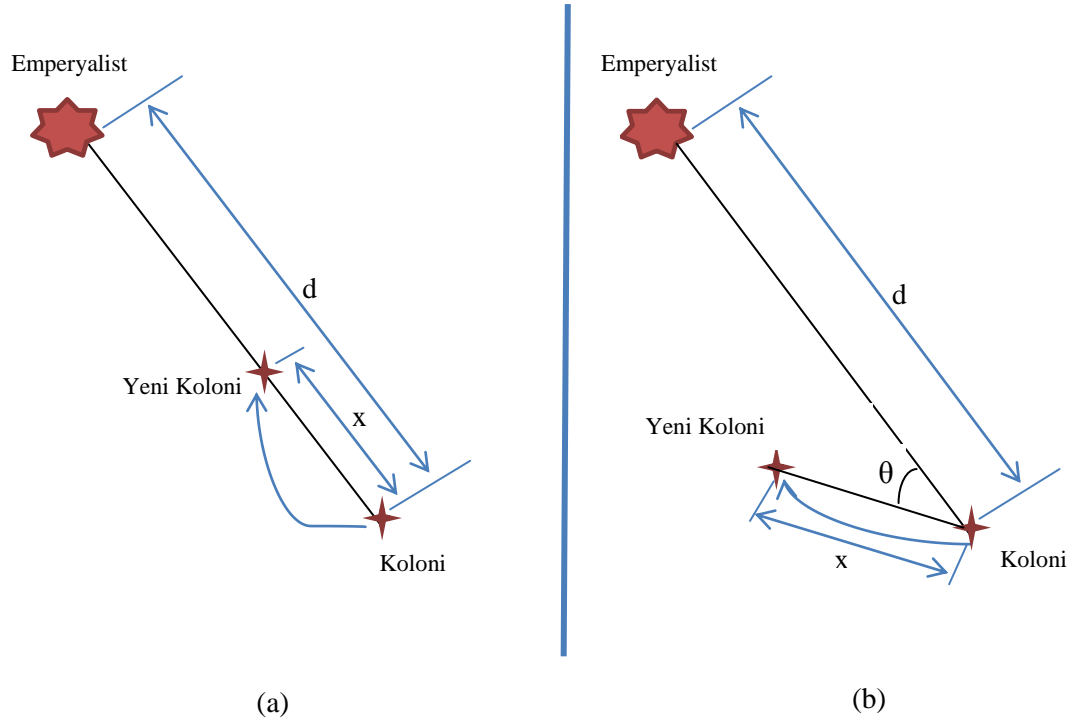
Şekil 2.2. İmparatorlukların başlangıç popülasyonları

2.1.3. Bir imparatorluğun kolonilerinin emperyaliste doğru hareketleri

Zamanla emperyalist ülkeler kolonilerini güçleri doğrultusunda arttırmaya başlar. Bu durum kolonilerin emperyaliste doğru hareketi şeklinde meydana gelir. Şekil 2.3.(a)' da koloninin x birimlik emperyaliste doğru olan hareketi gösterilir. Hareketin yönü koloniden emperyaliste doğru bir vektör şeklindedir. Şekildeki x rastgele bir değerdir.

$$x \sim U(0, \beta \times d) \quad (\text{E.2.6})$$

β 1 den büyük bir sayı ve d ise aradaki uzaklıktır. Koloninin emperyaliste yaklaşması için $\beta > 1$ olmalıdır.



Şekil 2.3. (a) Koloninin emperyaliste hareketi (b) Koloninin yeni pozisyonu

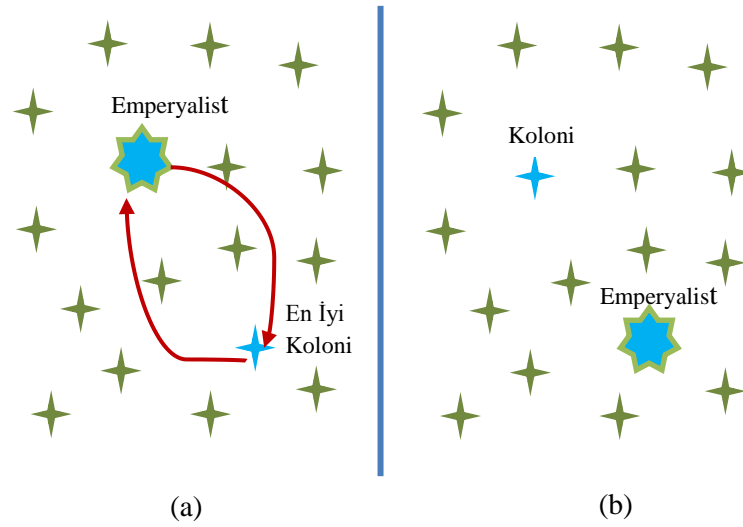
Emperyalistin çevresindeki farklı noktalara ulaşmak için hareketin yönüne rastgele bir sapma değeri eklenir. Şekil 2.3.(b)' de θ gelişigüzel bir değeri ifade eder.

$$\theta \sim U(-\gamma, \gamma) \quad (\text{E.2.7})$$

γ ise orijinal yönden ne kadarlık bir sapma olacağını ayarlayan parametredir. Bununla beraber β ve γ değerleri keyfi olarak alınmıştır. (Atashpaz-Gargari ve Lucas, 2007:4663; Kızılloluk ve Alataş, 2012).

2.1.4. Emperyalist ve koloninin konum değış tokuşu

Koloniler emperyalistlere doğru ilerlerken, emperyalistin maliyetinden daha düşük ölçütlü bir konuma sahip olabilirler. Bu durumda emperyalist koloni ile konum değış tokuşuna gider. Algoritma sonlanana dek emperyalistin yeni pozisyonu ve kolonilerin bu yeni pozisyonlara hareketleri ile süregider. Şekil 2.4.(a) koloni ile emperyalistin konum değıştirmesini gösterir. İmparatorlukta emperyalistten daha iyi ölçüt değeriine sahip bu koloni açık mavi renkle gösterilmektedir. Şekil 2.4.(b) de imparatorluğun, emperyalist ile koloninin konum değışikliğinden sonraki durumu gösterilmiştir (Atashpaz-Gargari ve Lucas, 2007:4663; Kızılloluk ve Alataş, 2012).



Şekil 2.4. (a) Koloni ve emperyalistin yer değişimi (b) Yer değişikliğinden sonraki durum

2.1.5. İmparatorluğun toplam gücü

Bir imparatorluğun toplam gücünü büyük oranda emperyalist ülkenin gücü etkiler. İmparatorlukta kolonilerin gücü pek ön plana çıkmaz da bu imparatorluğun toplam gücü üstünde yine de etkindir. Bu durum E.2.8’ de bir imparatorluğun toplam ölçütü tanımlanarak gösterilmiştir.

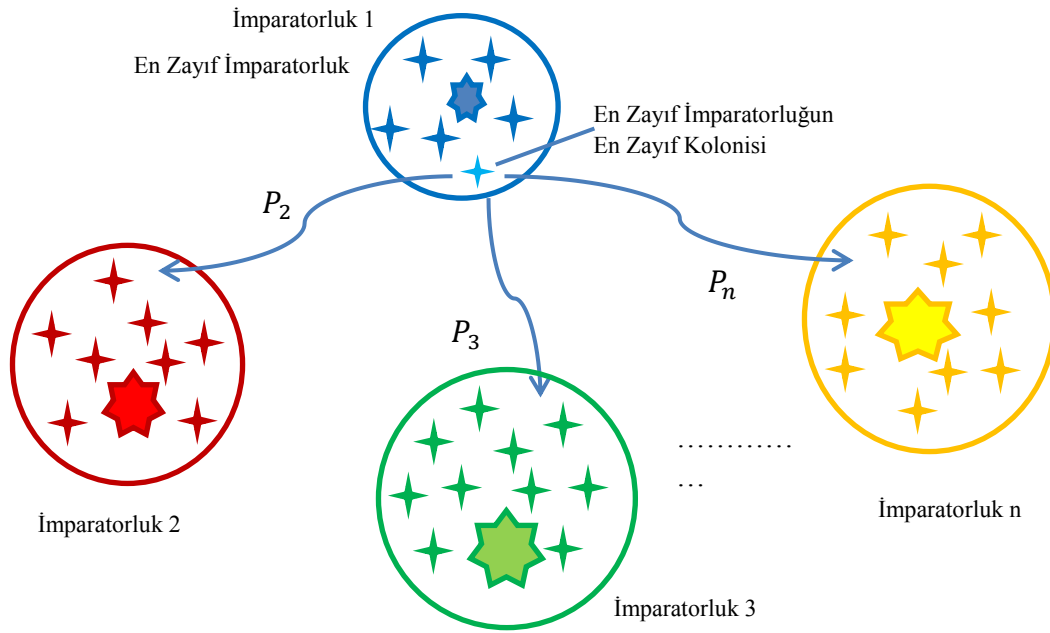
$$T.C._n = Cost(imperialist_n) + \xi \text{ mean}\{Cost(colonies \text{ of } empire_n)\} \quad (E.2.8)$$

n . imparatorluğun toplam ölçütü $T.C._n$ ’ dir ve ξ , 1’ den küçük olan pozitif bir sayıdır. ξ için küçük değerler tanımlanmak, belirlenen imparatorluğun gücünde emperyalistin rolünü artırırken, bu değeri artırmak imparatorluğun toplam gücüne karar vermede kolonilerin etkenliğini artırır. ξ için 0,1 değeri birçok uygulamada kullanılmıştır (Atashpaz-Gargari ve Lucas, 2007:4664; Kızılloluk ve Alataş, 2012).

2.1.6. Emperyalistlik yarış

Bütün imparatorluklar birbirlerinin kolonilerini ele geçirmeye çalışır ve kontrol ederler. Emperyalistlik yarış, süreç içerisinde zayıf imparatorlukların gücünde azalmalara ve daha güçlü olanların gücünde artışa neden olur. Bu yarış, zayıf imparatorlukların zayıf kolonilerinden bazılarını kaybetmelerine ve diğer

imparatorlukların bu kolonileri ele geçirmeye çalışmaları üzerine modellenmiştir. Şekil 2.5 modellenmiş emperyalistik yarışın büyük bir resmini gösterilmektedir. Bu yarışta imparatorlukların her biri diğer imparatorluğun zayıf kolonisini ele geçirme ihtimaline sahiptir. Fakat güçlü imparatorlukların bu koloni veya kolonileri ele geçirme olasılıkları daha fazla olmaktadır (Atashpaz-Gargari ve Lucas, 2007:4664; Kızılloluk ve Alataş, 2012).



Şekil 2.5. Emperyalistik yarışma

Yarışa başlarken her imparatorluğun toplam gücüne göre ele geçirme olasılığı bulunmalıdır.

$$N.T.C._n = T.C._n - maks_i\{T.C._i\} \quad (E.2.9)$$

E.2.9' daki $T.C._n$ n. imparatorluğun toplam ölçütü ve $N.T.C._n$ normalize edilmiş toplam ölçütüdür.

$$p_{p_n} = \left| \frac{N.T.C._n}{\sum_{i=1}^{N_{imp}} N.T.C._i} \right| \quad (E.2.10)$$

İlgili kolonileri imparatorlukların sahip olduğu ele geçirme olasılıkları arasında dağıtmak için, E.2.11 de gösterilen P vektörü oluşturulur.

$$P = [p_{p_1}, p_{p_2}, \dots, p_{p_{N_{imp}}}] \quad (\text{E.2.11})$$

Sonra P ile aynı boyutta rastgele sayılardan oluşan R vektörü oluşturulur.

$$R = [r_1, r_2, \dots, r_{N_{imp}}] \quad (\text{E.2.12})$$

$$r_1, r_2, \dots, r_{N_{imp}} \sim U(0,1)$$

Daha sonra P ' den R çıkarılarak D vektörü bulunur, işlem E.2.13' de gösterilmiştir:

$$D = P - R = [D_1, D_2, \dots, D_{N_{imp}}] = [p_{p_1} - r_1, p_{p_2} - r_2, \dots, p_{p_{N_{imp}}} - r_{N_{imp}}] \quad (\text{E.2.13})$$

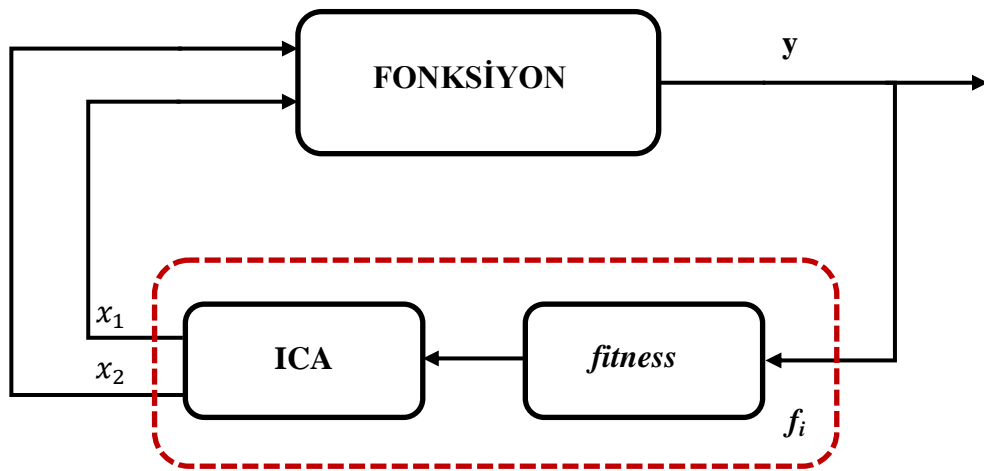
Vektör D 'ye göre D 'nin maksimum indeksi ile ilgili olan imparatorluğun kolonileri elde edilir (Atashpaz-Gargari ve Lucas, 2007:4664; Kızıloluk ve Alataş, 2012).

2.1.7. Yakınsama

Emperyalisttik yarışta güçsüz imparatorluklar geride kalır ve çöker. Ona ait koloniler ise diğer imparatorluklar arasında pay edilir. En güçlü imparatorluk haricindeki diğer imparatorluklar çöktükten sonra tüm koloniler, en güçlü olan ve tek kalan imparatorluğun kontrolüne girer. Artık bu yeni dünya idealinde aynı pozisyon ve ölçüte sahip olan koloniler, onlarla aynı pozisyon ve ölçüte sahip olan bir emperyalist tarafından kontrol edilecektir. Böyle bir durumda emperyalisttik yarışa son verilir ve algoritma durdurulur (Atashpaz-Gargari ve Lucas, 2007:4664; Kızıloluk ve Alataş, 2012).

2.2. ICA ile Fonksiyon Optimizasyonu

Standart ICA (Std-ICA) algoritmasının işleyişi ve çalışma mantığının anlaşılması amacıyla ilk etapta peaks fonksiyonu üzerinde çalışması incelenmiştir. Bu çalışmalar sırasında, Std-ICA algoritmasının başarımında etki etmesi düşünülen bazı değişkenlerde ve çalışma mantığı üzerinde değişiklikler öngörülmüştür. Bu değişikliklerden biri; imparatorlukların toplam gücüne karar vermede kolonilerin mi yoksa emperyalistlerin mi daha etkin rolü alacağını belirleyen “zeta” değerinde algoritma işleyişi içinde oynamalar yapmakken, diğeri; koloniler arasında en iyi ölçüte sahip olan emperyalisti, çevresindeki uzayı rastgele tarayarak daha iyi bir ölçüte sahip olacak konuma hareketlendirmeye çalışmaktır. Bu öngörüler sonucunda başarımın artırıldığı gözlenerek iki yeni ICA algoritması yaklaşımı geliştirilmiştir. Bu yaklaşımların kıyaslamalı başarımları için literatürden alınan üç farklı fonksiyon üzerinde işleyişleri incelenmiştir. Aşağıda Şekil 2.6’ de ICA ile fonksiyon optimizasyonu blok yapısı gösterilmektedir.



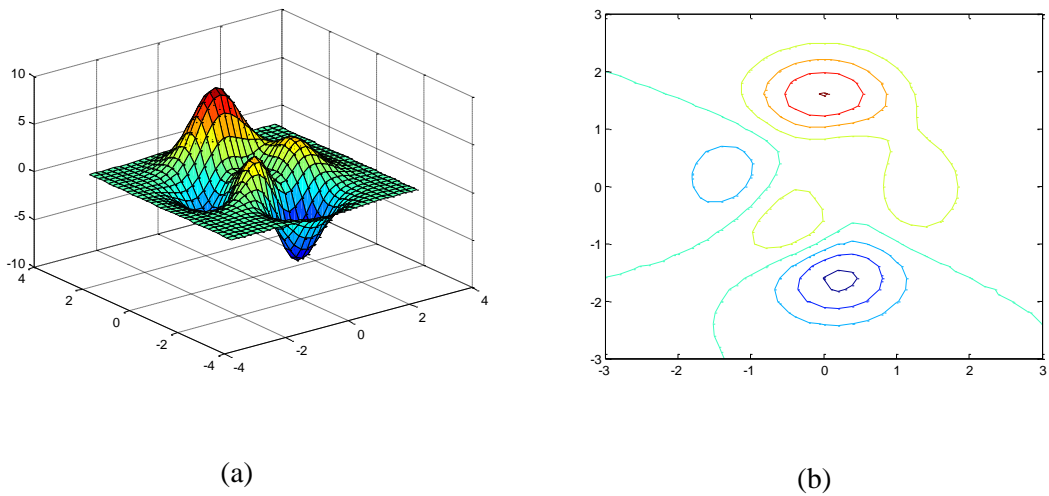
Şekil 2.6. ICA ile fonksiyon optimizasyonu blok yapısı

2.2.1."Peaks" fonksiyonu optimizasyonu

Peaks fonksiyonu aşağıdaki E.2.14 gibi tanımlanır.

$$z = 3(1 - x)2 \times e^{(-x^2 - (y+1)^2)} - 10 \times (x \div 5 - x^3 - y^5) \times e^{(x^2 - y^2 - 1 \div 3)} \times e^{-(x+1)^2 - y^2} \quad (\text{E.2.14})$$

Fonksiyonun eş yükselti eğrileri ve yüzeyi Şekil 2.7' de verilmiştir. Görüleceği gibi, fonksiyon birden çok tepe ve çukurlar içermektedir. Std-ICA algoritması ile yapılmak istenen bu fonksiyonun küresel minimum noktasını bulmaya çalışmasıdır.

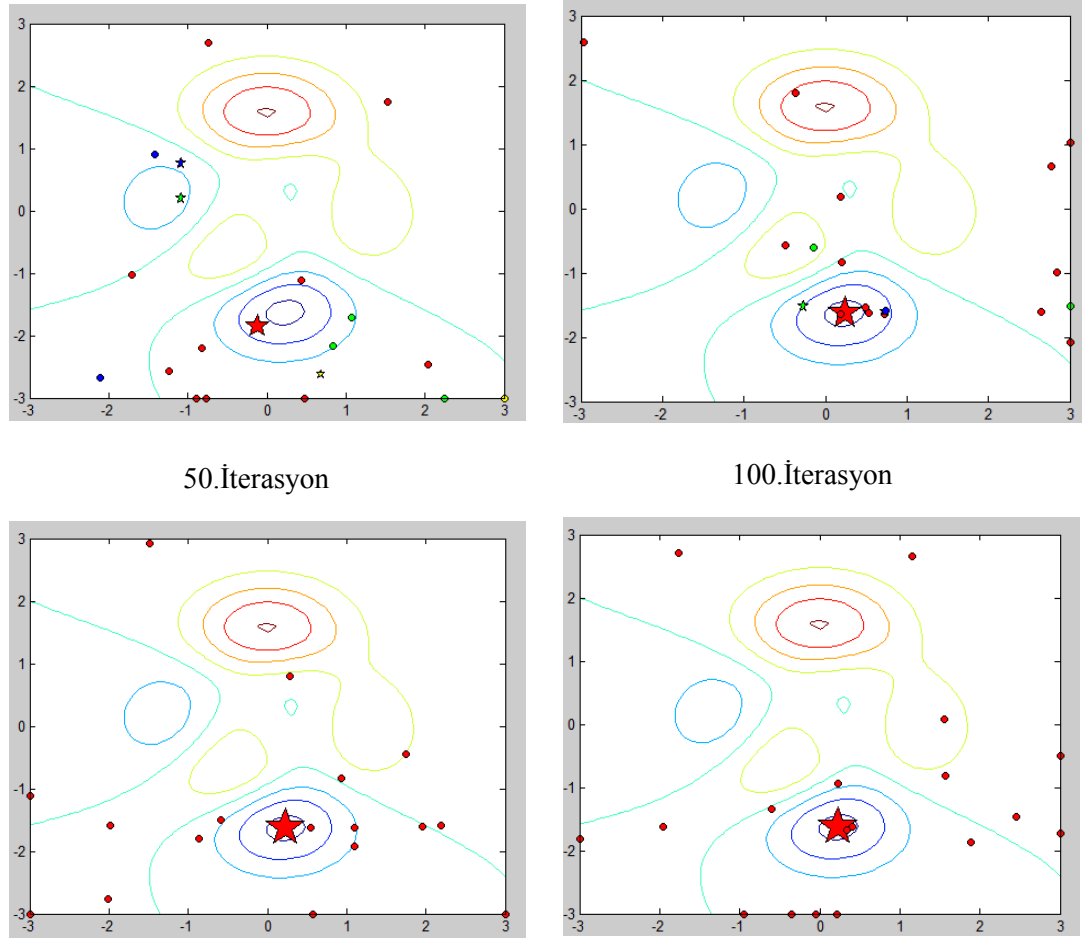


Şekil 2.7. Peaks fonksiyonu yüzeyi(a) ve eş yükselti eğrileri (b)

Yüzey grafiğinden de görüleceği gibi peaks fonksiyonu üç farklı tepe ve üç farklı çukura sahiptir.

Bu fonksiyonun barındırdığı 3 çukurun en derini olan; küresel minimum noktası: -6,5511' dir.

Sdt-ICA algoritması ile peaks fonksiyon optimizasyonu yapılırken, 20 koloni 4 emperyalist için 100 adımda algoritma koşturularak bireylerin gözlemlenmesi amaçlanmıştır. Buna göre muhtelif adımlardaki algoritmanın işleyişi ve birey görüntüleri Şekil 2.8' deki gibidir.



Şekil 2.8. Std-ICA algoritması ile “peaks” fonksiyon optimizasyonu için muhtelif adımlardaki birey pozisyonları

Çalışmanın başlangıcında ICA Algoritması ile Peaks fonksiyon optimizasyonu, algoritmanın işleyişini ve çalışma mantığını anlamak adına incelenmiştir. Bu incelemeler sırasında iki yeni yaklaşım geliştirilmiştir. Bu yaklaşımların ilkinde ICA (Standart ICA / Std-ICA) algoritmasında kritik öneme sahip “zeta” değişkenine dikkat edilmiştir. Bu değişken, imparatorluğun toplam gücünü belirlemede rol oynamaktadır. 1’ den küçük pozitif bir sayı olan “zeta” değeri, sıfıra ne kadar yakın olursa, imparatorluğun toplam gücü belirlenirken emperyalistler de kolonilere oranla o denli etken olmaktadır. Bu değişken, Atashpaz-Gargari ve Lucas’ ın 2007 yılındaki (Atashpaz-Gargari ve Lucas, 2007) çalışmasında 0,1 olarak alınırken algoritma kaynak kodlarında 0,02 olarak alınmıştır. Bu yaklaşımda “zeta” değerini makaledeki

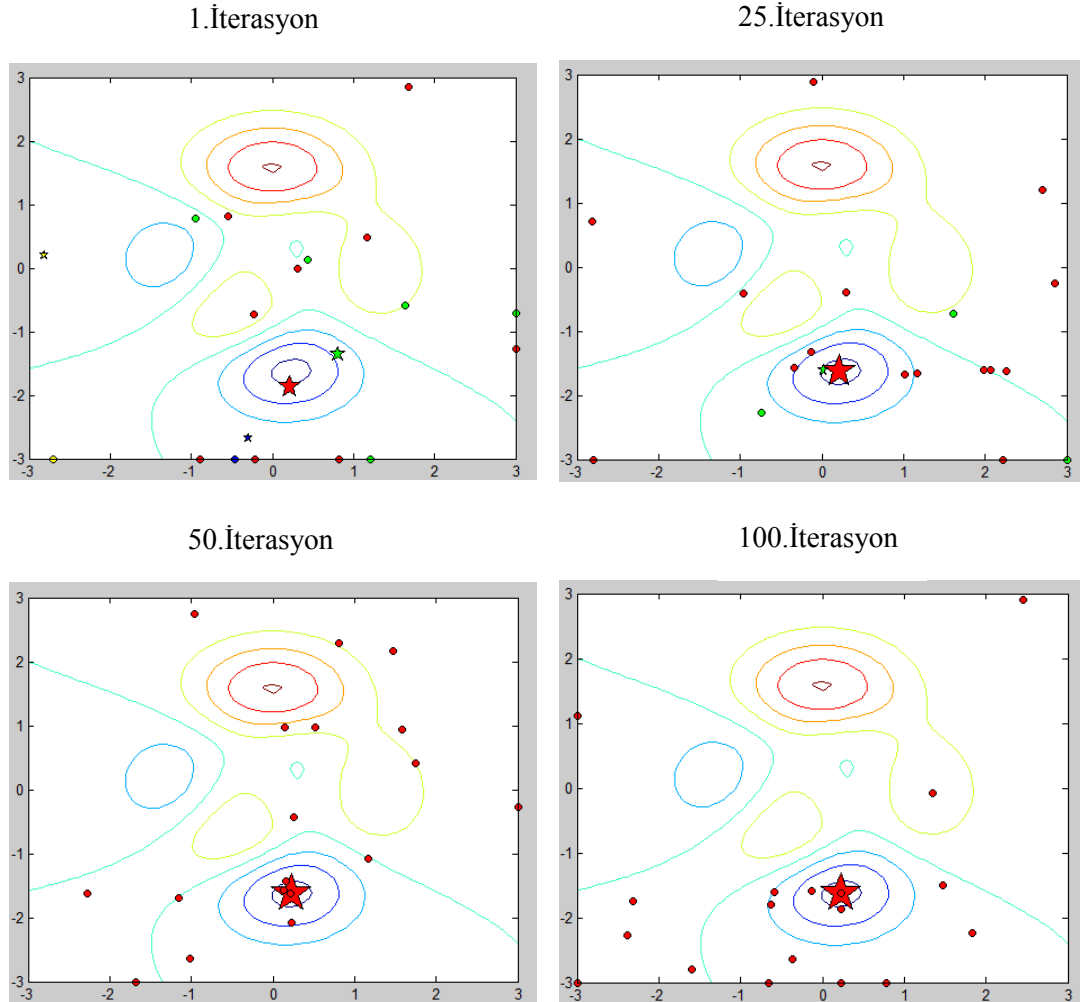
değerlerden kaynak koddaki değerlere çekilerek imparatorluk toplam gücü hesaplanırken ilerleyen her adımda emperyalistlerin daha etken olması amaçlanmıştır. Bundan yola çıkılarak “zeta” değerini algoritmanın koşturulması esnasında her iterasyonla üstel olarak azaltıldığı üstel azaltım yaklaşımı (G1-ICA)’dır.

G1-ICA yaklaşımı için “zeta” değeri E.2.15’deki eşitlik ile hesaplanır.

$$\xi = \xi \times e^{(-0.0005 \times iteS)} \quad (E.2.15)$$

Matemtiksel ifadelerde yer alan terimlerin anlamı aşağıdaki gibidir:

ξ : Zeta , iteS: iterasyon



Şekil 2.9. G1-ICA Algoritması ile Peaks fonksiyonu optimizasyonu için muhtelif adımlardaki birey pozisyonları

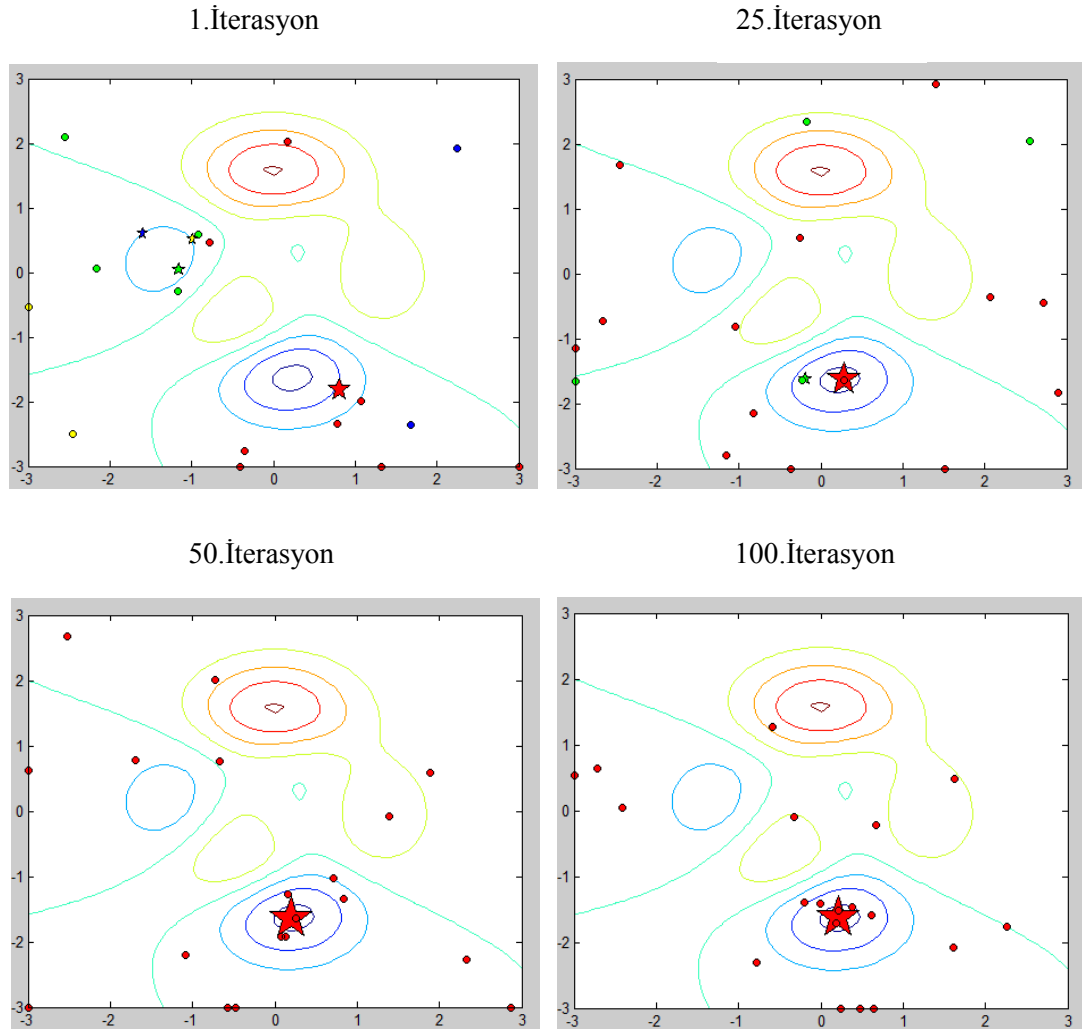
Diğer yaklaşımda ise imparatorluk içerisinde kendi kolonileri arasında en iyi ölçüte sahip olan emperyalistleri, çevresindeki uzayı rastgele tarayarak daha iyi bir ölçüte sahip olacak konuma hareketlendirmeye çalışılmaktadır. Bunun için her imparatorluğun emperyalistini, ele geçirilen her kolonide, çevresinde rastgele on adet konum belirlenmektedir. Emperyalistler bu konumlarda olsaydı oluşacak ölçüt değerleri hesaplanarak emperyalistlerin sahip olduğu o anki ölçüt değeri ile kıyaslanıp daha küçük ölçüt değerine sahip olacağı konuma hareketlendirilerek, emperyalistleri o konuma yerleştirilmesine çalışılmaktadır. Bu sayede fonksiyon yüzeyinde daha hızlı bir şekilde minimum noktaların bulunmasını amaçlayan yaklaşım (G2-ICA)'dır.

G2-ICA yaklaşımı için yeni konum değerleri E.2.16' daki eşitlik ile hesaplanır.

$$hYenPoz = EmpPoz + 0.005 \times randn \times size(EmpPoz) \quad (E.2.16)$$

Matematiksel ifadelerde yer alan terimlerin anlamı aşağıdaki gibidir:

hYenPoz: Hesaplanan yeni pozisyon, *EmpPoz*: İlgili emperyalistin pozisyonu



Şekil 2.10. G2-ICA Algoritması ile Peaks fonksiyonu optimizasyonu için muhtelif adımlardaki birey pozisyonları

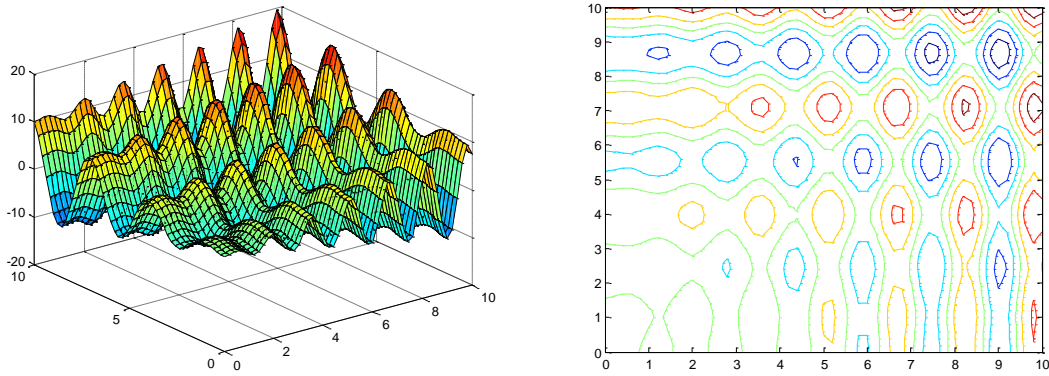
Üstteki kısımda anlatılan iki yeni (G1-ICA, G2-ICA) yaklaşımı ve Std-ICA algoritması peaks fonksiyonu optimizasyonu ile 100 adımda 20 koloni 4 emperyalist olacak şekilde 100 kez koşturulmuştur. Peaks fonksiyonunun optimizasyonundan sonra bu iki yeni (G1-ICA, G2-ICA) yaklaşımı ve Std-ICA algoritması 3 farklı fonksiyon optimizasyonu aynı yöntemle örneklendirilecektir. Örneklendirmeler sonunda birbirleri ile başarımlarının karşılaştırılabilmesi için her bir algoritma ile elde edilen sonuçlar Çizelge 2.1' de gösterilecektir.

2.2.2. Örnek 1: F1

F1 fonksiyonu aşağıdaki E.2.17 gibi tanımlanır.(Ramin, 2011)

$$z = x\sin(4x) + 1.1y\sin(2y) \quad (\text{E.2.17})$$

Fonksiyonun eş yükselti eğrileri ve yüzeyi Şekil 2.11' de verilmiştir. Görüleceği gibi, fonksiyon çok fazla sayıda tepe ve çukurlar içermektedir. Bu fonksiyonun küresel minimum noktası; -18.5547 olarak elde edilmiştir.

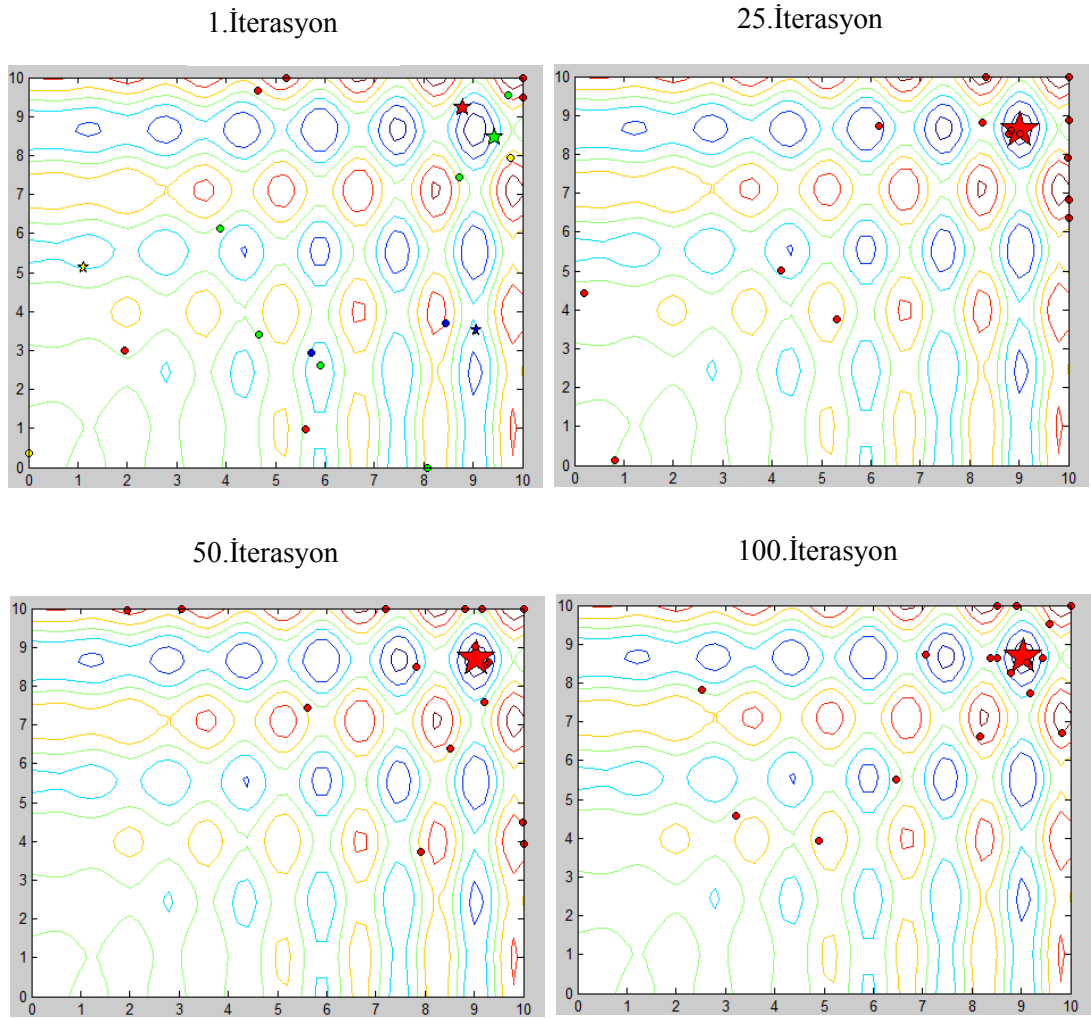


(a)

(b)

Şekil 2.11. F1 fonksiyonu yüzeyi(a) ve eş yükselti eğrileri (b)

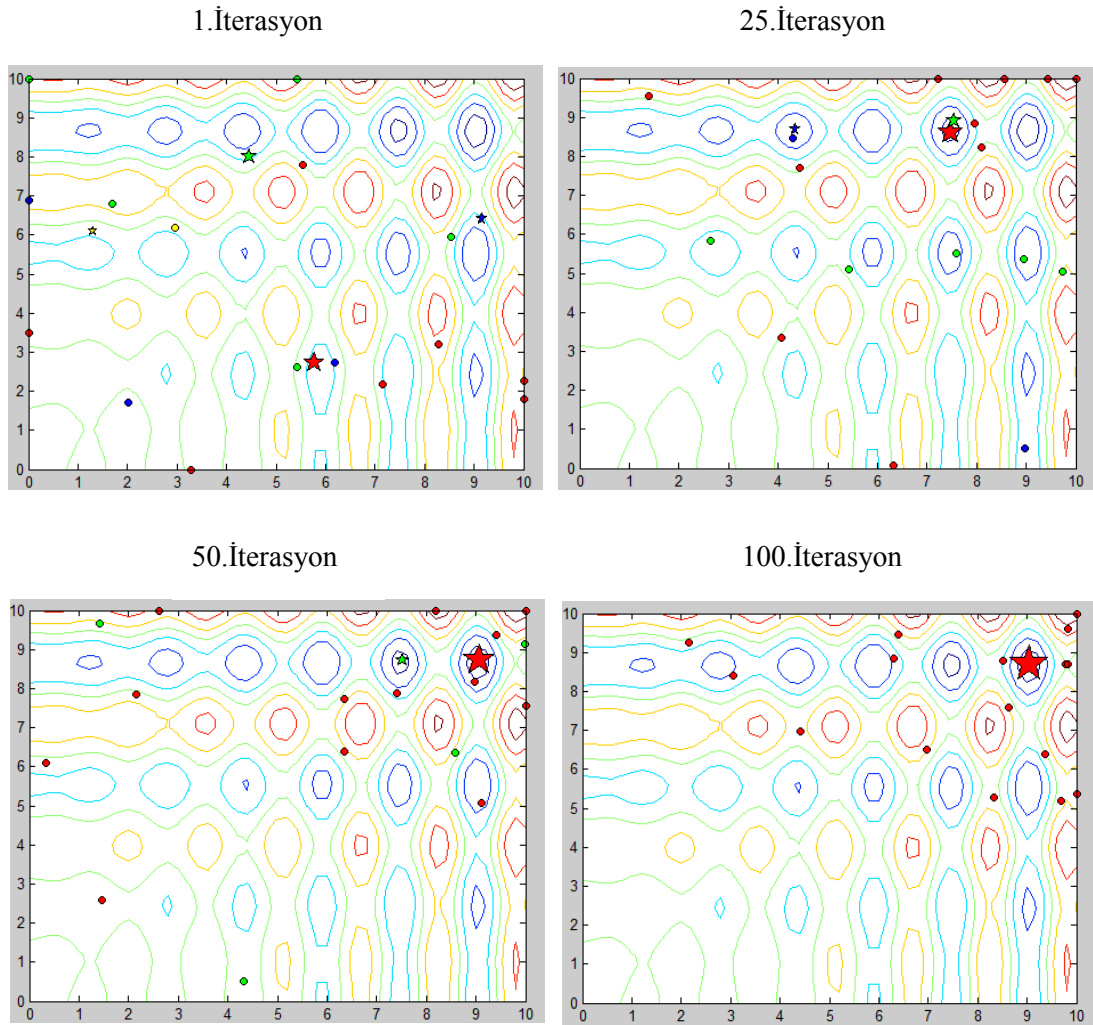
Std-ICA, G1-ICA ve G2-ICA algoritmalarının başarımlarını kıyaslamasını görmek amacıyla ilk örneklendireceğimiz fonksiyonumuzun E.2.17' de matematiksel ifadesi görülmektedir. Buna göre Std-ICA yaklaşımı ile elde edilen sonuçlar aşağıda Şekil 2.12' de gösterilmektedir.



Şekil 2.12. Std-ICA Algoritması ile F1 fonksiyon optimizasyonu için muhtelif adımlardaki birey pozisyonları

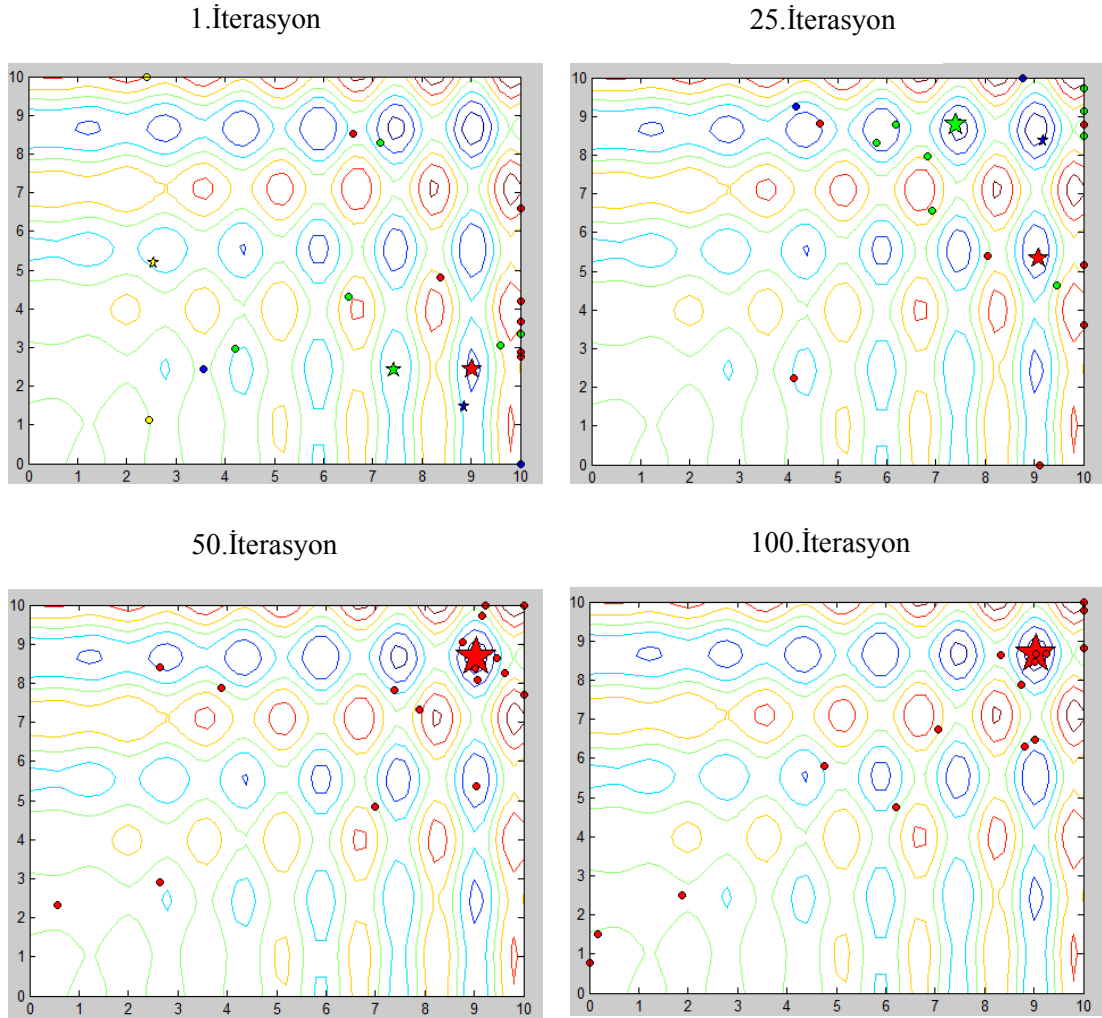
Örnek 1 fonksiyonumuz için sonuçlara bakıldığında Std-ICA algoritması fonksiyonun çok fazla sayıda yerel minimumlara sahip olmasına rağmen başarılı olduğu ve 25. İterasyonda küresel minimuma yerleştiği görülmektedir.

Örnek 1 fonksiyonumuzu G1-ICA ve G2-ICA yaklaşımları için koşturduğumuzda da elde edilen sonuçlar Std-ICA gibi başarılı olduğunu göstermektedir. Aşağıda Şekil 2.13’ de algoritma değişkenlerinden zeta değerini her İterasyonda üstel azaltım yöntemi ile değiştirdiğimiz G1-ICA yaklaşımına ait sonuçlar gösterilmektedir.



Şekil 2.13. G1-ICA Algoritması ile F1 fonksiyon optimizasyonu için muhtelif adımlardaki birey pozisyonları

Aşağıda Şekil 2.14’ de emperyalistleri daha iyi ölçüt değerine sahip edecek, hemen yakınındaki konumlara hareketlendiren G2-ICA yaklaşımına ait sonuçlar gösterilmektedir.



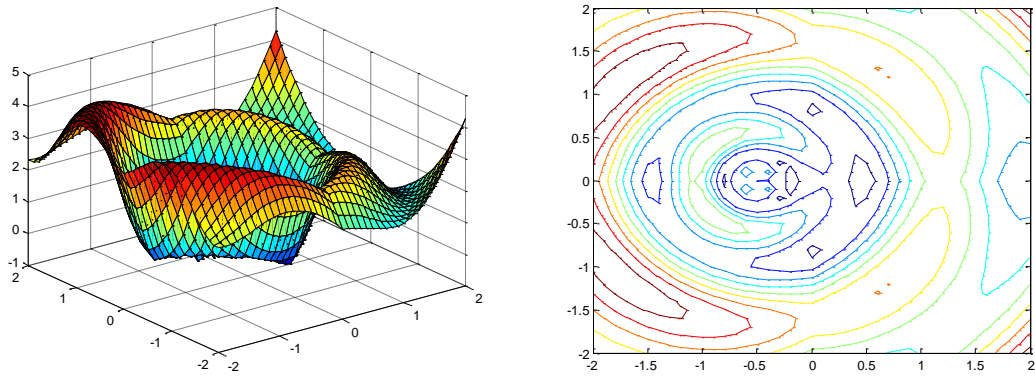
Şekil 2.14. G2-ICA Algoritması ile F1 fonksiyonu optimizasyonu için muhtelif adımlardaki birey pozisyonları

2.2.3. Örnek 2: F2

F2 fonksiyonu aşağıdaki E.2.18 gibi tanımlanır. (Ramin, 2011)

$$z = (x^2 + y^2)^{0.25} \times \sin\{30[(x + 0.5)^2 + y^2]^{0.1}\} + |x| + |y| \quad (\text{E.2.18})$$

Fonksiyonun eş yükselti eğrileri ve yüzeyi Şekil 2.15' te verilmiştir. Görüleceği gibi, fonksiyon irili ufaklı çok sayıda tepe ve çukurlar içeren karmaşık bir yapıya sahiptir. Bu fonksiyonun küresel minimum noktası; -0.2471 olarak elde edilmiştir.

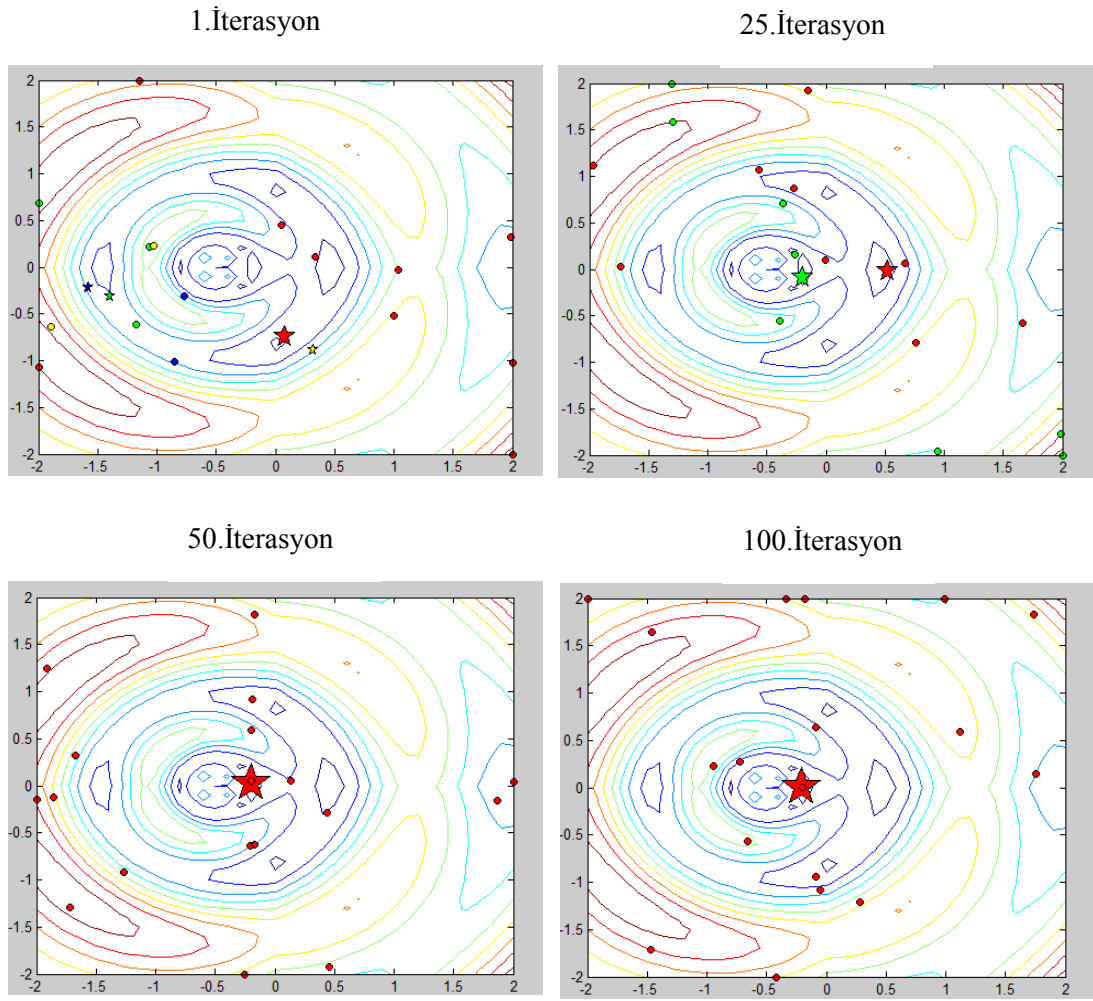


(a)

(b)

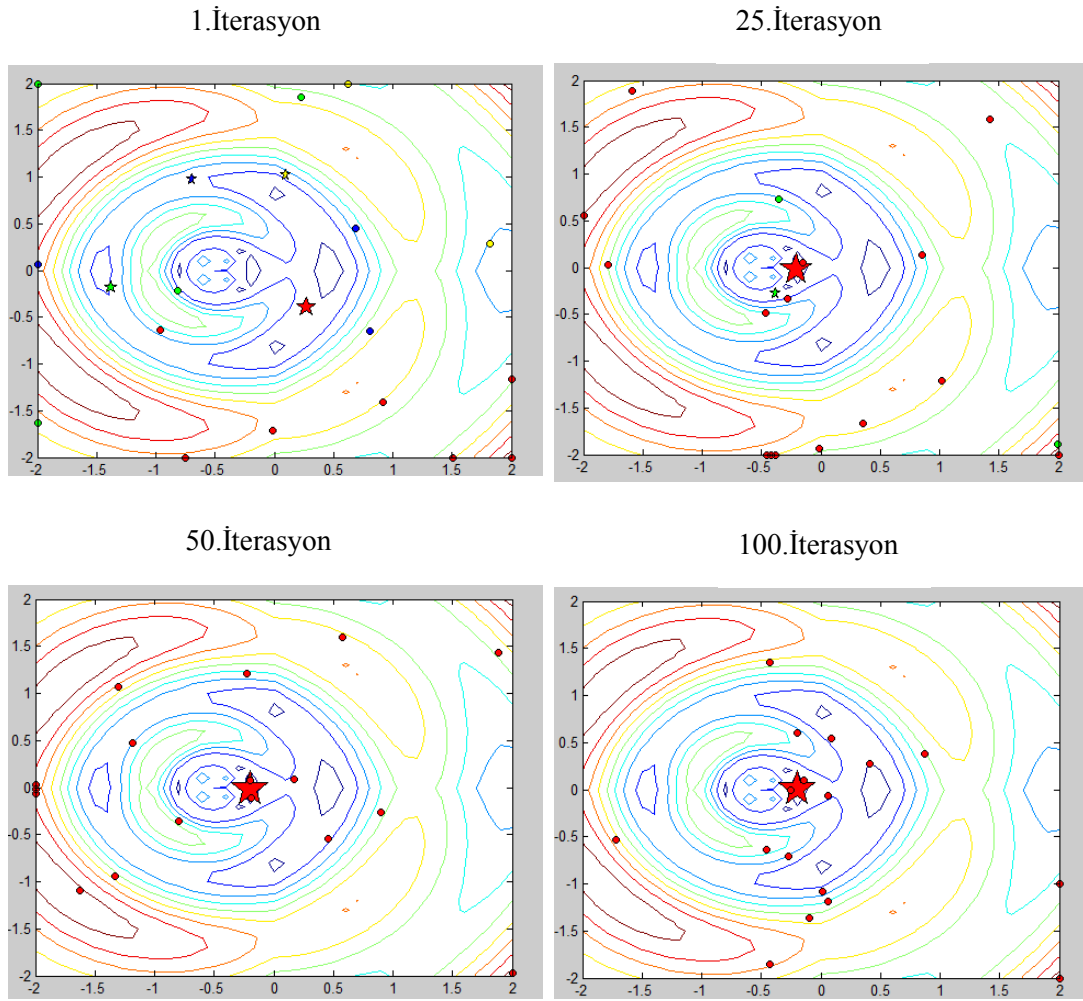
Şekil 2.15. F2 fonksiyonu yüzeyi(a) ve eş yükselti eğrileri (b)

Std-ICA, G1-ICA ve G2-ICA algoritmalarının başarımlarını kıyaslamasını görmek amacıyla örneklendireceğimiz fonksiyonumuzun E.2.18' de matematiksel ifadesi görülmektedir. Buna göre Std-ICA yaklaşımı ile elde edilen sonuçlar aşağıda Şekil 2.16' da gösterilmektedir.



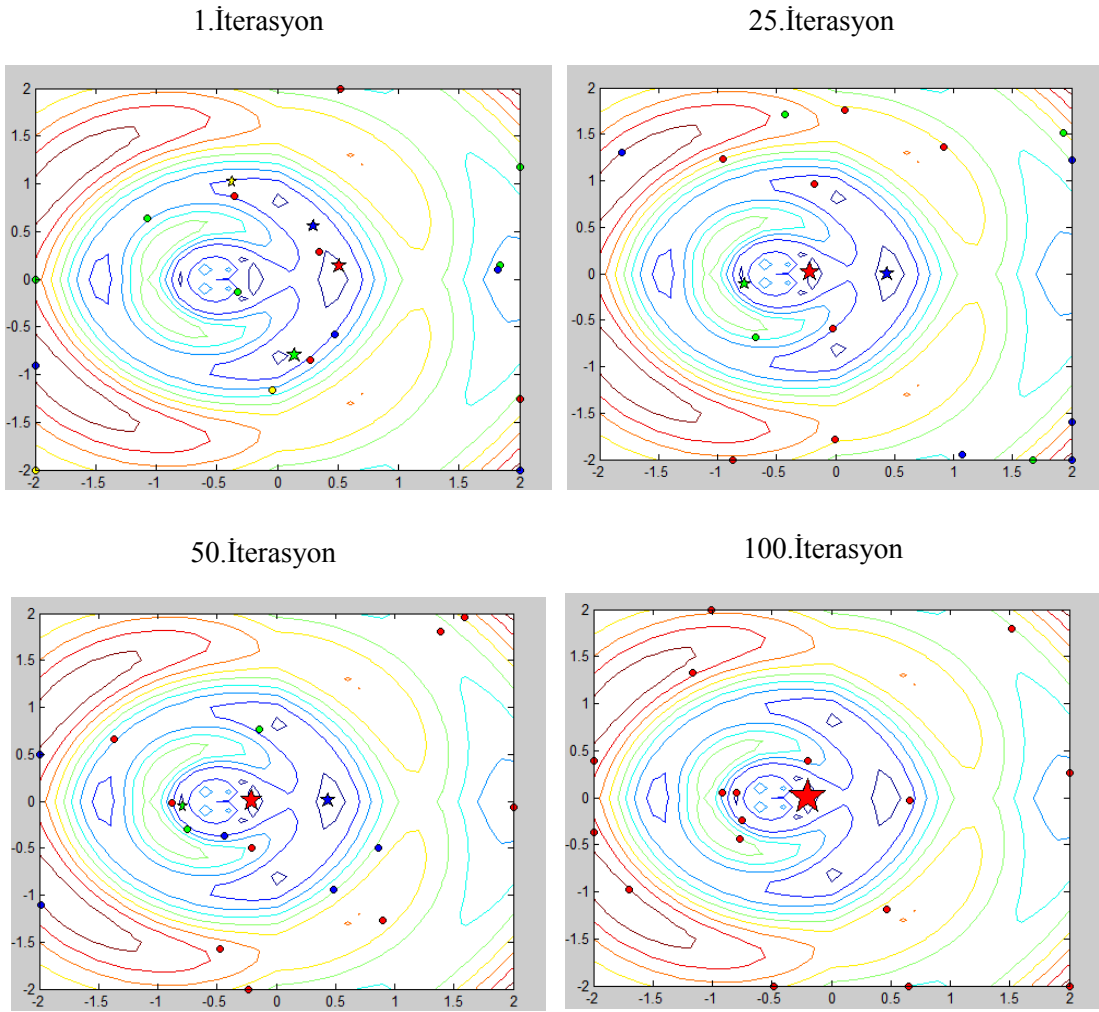
Şekil 2.16. Std-ICA Algoritması ile F2 fonksiyon optimizasyonu için muhtelif adımlardaki birey pozisyonları

Aşağıda Şekil 2.17' de algoritma değişkenlerinden zeta değerini her İterasyonda üstel azaltım yöntemi ile değiştirdiğimiz G1-ICA yaklaşımına ait sonuçlar gösterilmektedir.



Şekil 2.17. G1-ICA Algoritması ile F2 fonksiyon optimizasyonu için muhtelif adımlardaki birey pozisyonları

Aşağıda Şekil 2.18' de emperyalistleri daha iyi ölçüt değerine sahip edecek, hemen yakınındaki konumlara hareketlendiren G2-ICA yaklaşımına ait sonuçlar gösterilmektedir.



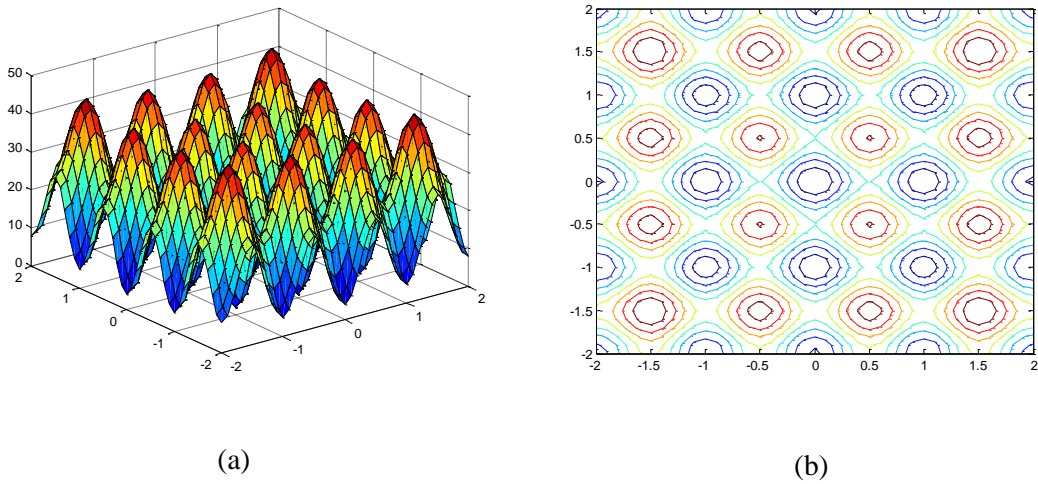
Şekil 2.18. G2-ICA Algoritması ile F2 fonksiyon optimizasyonu için muhtelif adımlardaki birey pozisyonları

2.2.4. Örnek 3: F3

F3 fonksiyonu aşağıdaki E.2.19 gibi tanımlanır. (Ramin, 2011)

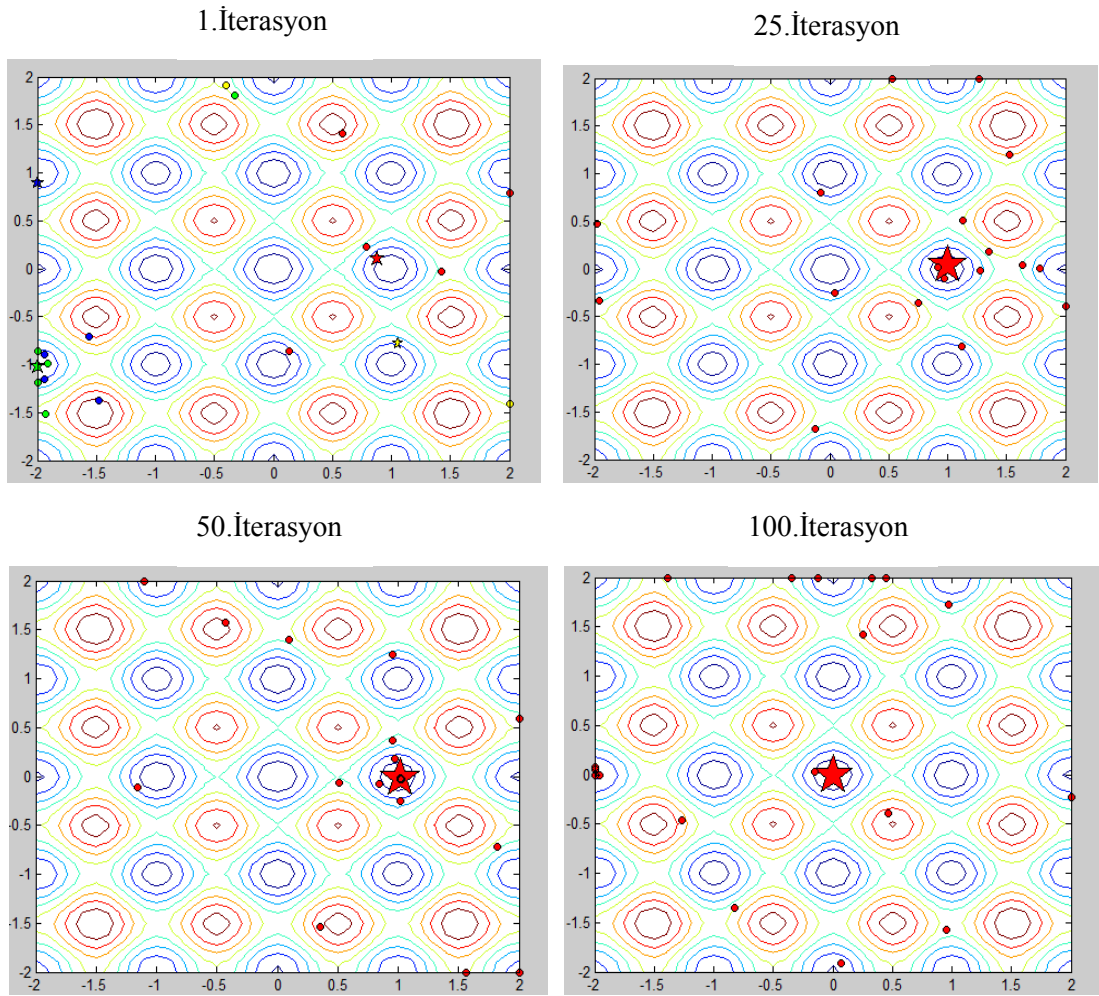
$$z = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)) \quad (\text{E.2.19})$$

Fonksiyonun eş yükselti eğrileri ve yüzeyi Şekil 2.19' da verilmiştir. Görüleceği gibi, fonksiyon benzer yapıda çok sayıda tepe ve çukur içeren yapıya sahiptir. Bu fonksiyonun küresel minimum noktası; 0 olarak elde edilmiştir.



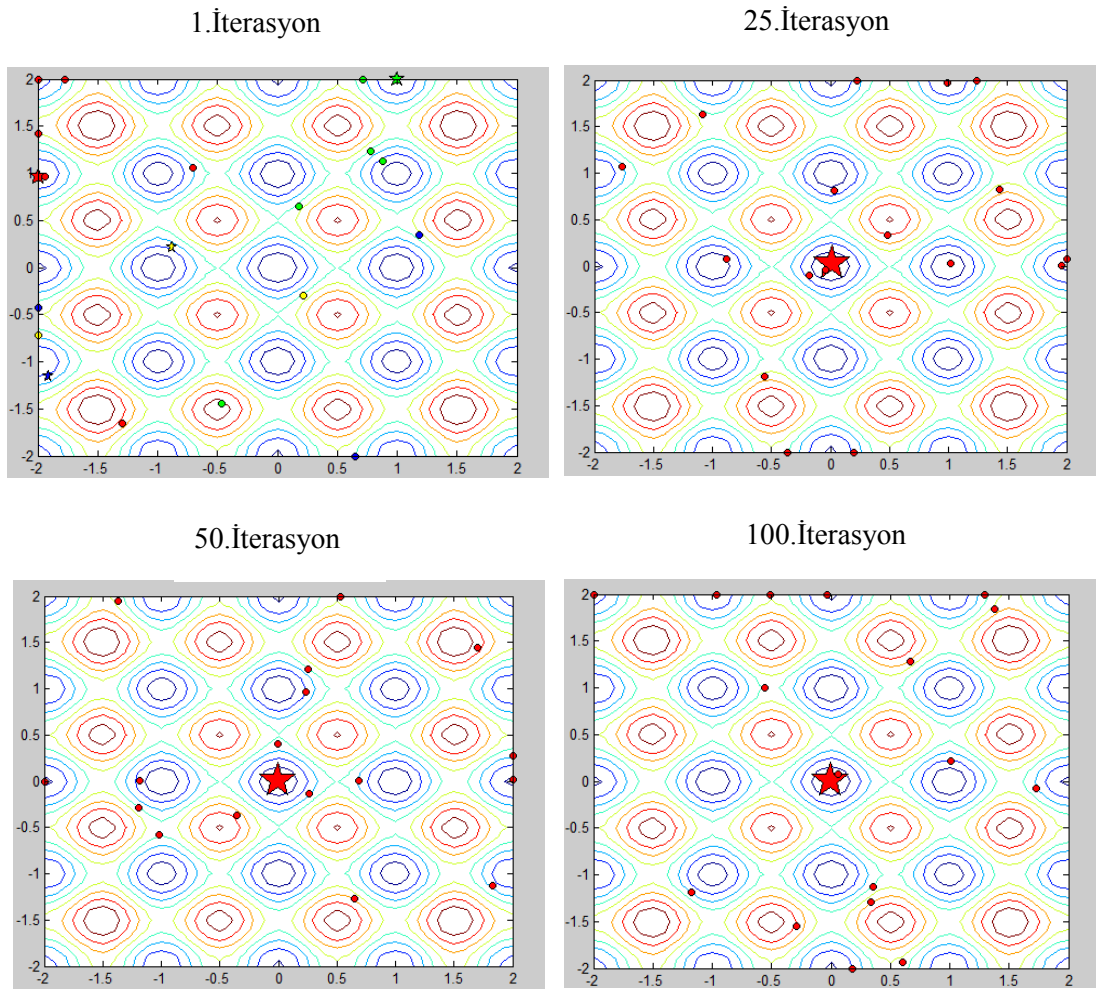
Şekil 2.19. F3 fonksiyonu yüzeyi (a) ve eş yükselti eğrileri (b)

Std-ICA, G1-ICA ve G2-ICA algoritmalarının başarımlarını kıyaslamasını görmek amacıyla örneklendireceğimiz fonksiyonumuzun E.2.19' da matematiksel ifadesi görülmektedir. Buna göre Std-ICA yaklaşımı ile elde edilen sonuçlar aşağıda Şekil 2.20' de gösterilmektedir.



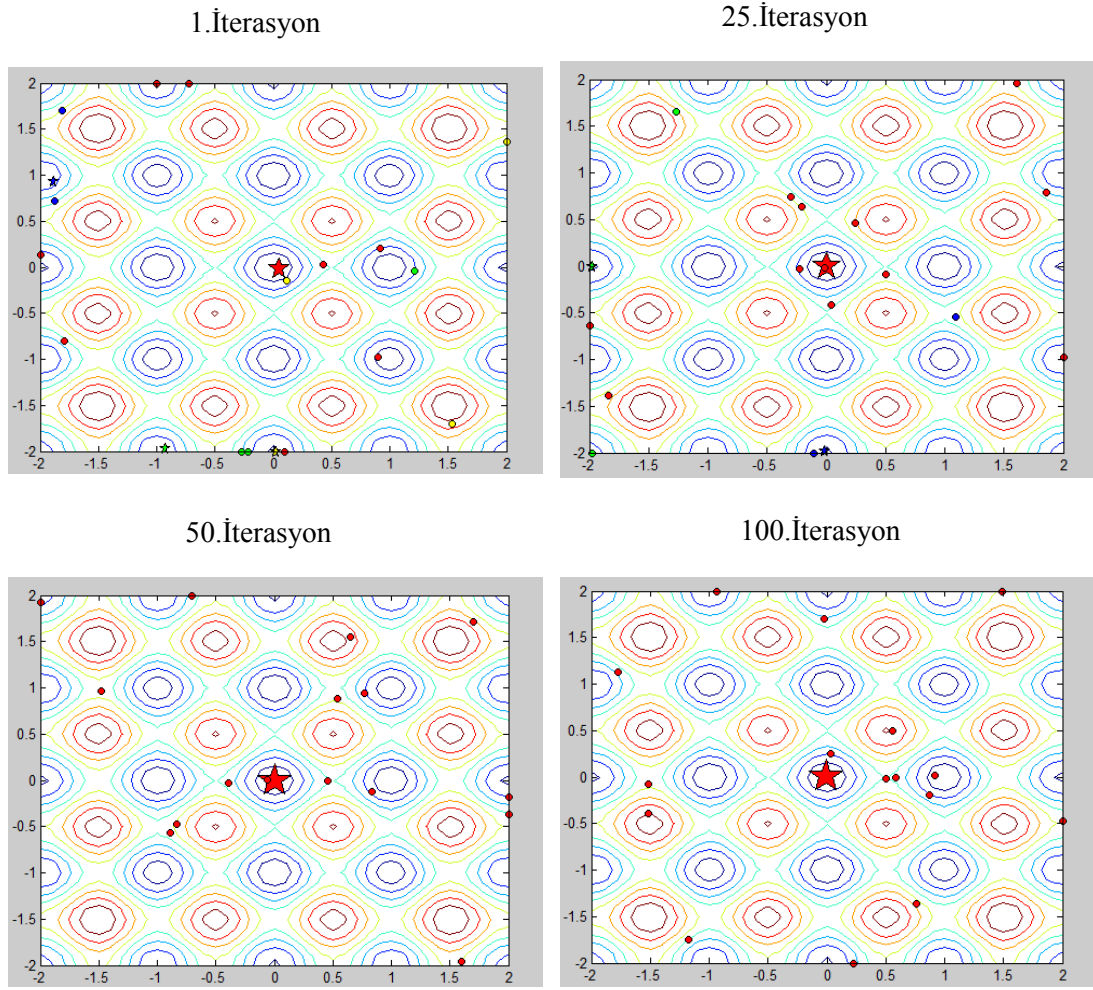
Şekil 2.20. Std-ICA Algoritması ile F3 fonksiyon optimizasyonu için muhtelif adımlardaki birey pozisyonları

Aşağıda Şekil 2.21' de zeta değişkeni için üstel azaltım yöntemi kullanan G1-ICA yaklaşımına ait sonuçlar gösterilmektedir.



Şekil 2.21. G1-ICA Algoritması ile F3 fonksiyon optimizasyonu için muhtelif adımlardaki birey pozisyonları

Aşağıda Şekil 2.22' de ilgili emperyalistleri hareketlendiren G2-ICA yaklaşımına ait sonuçlar gösterilmektedir.



Şekil 2.22. G2-ICA Algoritması ile F3 fonksiyon optimizasyonu için muhtelif adımlardaki birey pozisyonları

2.2.5. Geliştirilen ICA algoritmalarının başarımlarını kıyaslaması

Bu çalışmalar sırasında Standart ICA (Std-ICA) algoritmasının yanında 2 yeni ICA yaklaşımı geliştirilmiştir. Bunlar G1-ICA ve G2-ICA' dır. G1-ICA yaklaşımı; Std-ICA algoritmasında kullanılan “zeta” değişkenini üstel azaltım yöntemini kullanarak her ilerleyen iterasyon ile, ilgili emperyalistlerin imparatorluklarının toplam gücü hesaplanırken daha etkin olmasını sağlayan bir mantığa sahiptir. G2-ICA yaklaşımı ise; emperyalistleri daha iyi ölçüt değerlerine sahip edecek, hemen yakınlarındaki rastgele konumlara hareketlendirerek daha kısa zamanda minimum noktalara ulaşmayı sağlayan bir mantığa sahiptir. Geliştirilen ICA (G1-ICA, G2-ICA) algoritmalarının Standart ICA (Std-ICA) algoritması ile fonksiyon optimizasyonu üzerine başarımlarını kıyaslamaları Çizelge 2.1' de gösterilmiştir.

Çizelge 2.1. Geliştirilen ICA (G1-ICA, G2-ICA) algoritmalarının Standart ICA (Std-ICA) algoritması ile başarımlarını kıyaslaması

Fonksiyonlar	Yaklaşımlar	Küresel Minimum Bulma Sayısı	Yerel Minimum Bulma Sayısı	Doğruluk Katsayıları
Peaks	Std-ICA	100	0	-6,223
	G1-ICA	100	0	
	G2-ICA	100	0	
ÖRNEK 1:F1	Std-ICA	100	0	-18,5547
	G1-ICA	100	0	
	G2-ICA	100	0	
ÖRNEK 2:F2	Std-ICA	82	18	-0,2471
	G1-ICA	82	18	
	G2-ICA	85	15	
ÖRNEK 3:F3	Std-ICA	92	8	0
	G1-ICA	97	3	
	G2-ICA	98	2	

3. KULLANILAN DİNAMİK SİSTEMLER, BULANIK SİSTEM VE SEZGİSEL ÖĞRENME YÖNTEMLERİ

Bu bölümde ICA algoritmasının bulanık sistem modellemede başarımını görebilmek için kullanılan örnek dinamik sistemler ve bulanık çıkarım sistemi tanıtılmıştır.

3.1. Örnek Dinamik Sistemler

Bu çalışmada ICA algoritmasının bulanık sistem modellemede başarımını görebilmek amacıyla literatürde sıkça rastlanan ve Çizelge 3.1’ de verilen dinamik sistemler kullanılmıştır.

Çizelge 3.1. ABC, PSO ve DE öğrenmeli bulanık mantık tabanlı dinamik sistem tanıma/modelleme için literatürden seçilen örnek dinamik sistemler (ÖDS)

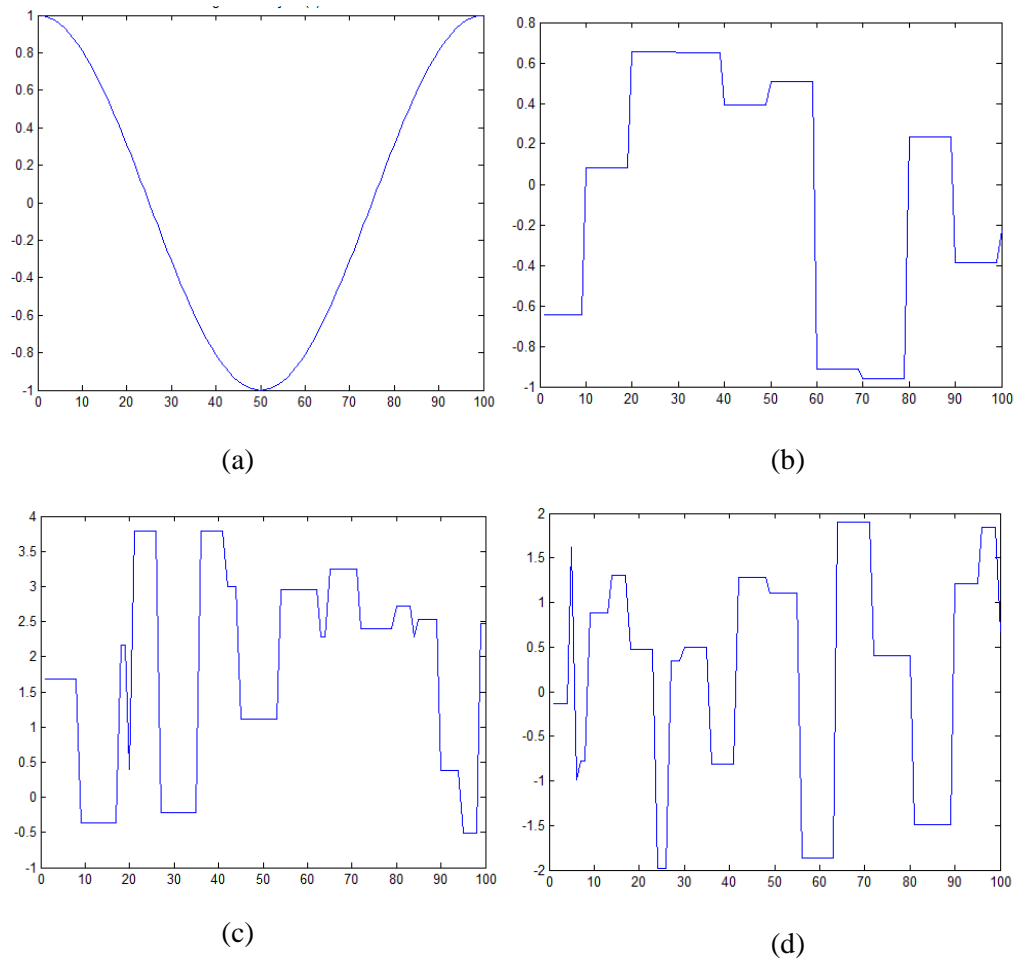
	Örnek Dinamik Sistemler (ÖDS)	Eğitim Seti	Test Seti
1	$y(k) = \frac{y(k-1).y(k-2).(y(k-1)+2.5)}{1+y^2(k-1)+y^2(k-2)} + u(k)$ <p>(Narendra, 1990)</p>	$u(k) = \cos \frac{2\pi k}{100}$	$u(k) = \sin \frac{2\pi k}{25}$
2	$y(k+1) = \frac{y(k)}{1+y^2(k)} + u^3(k)$ <p>(Narendra, 1990)</p>	$u(k) = \cos \frac{2\pi k}{100}$	$u(k) = \sin \frac{2\pi k}{25}$
3	$y(k+1) = y(k) + u(k)e^{-3 y(k) }$ <p>(Babuska, 2012)</p>	<p>[-1 1] Aralığında rasgele</p>	<p>[-1 1] Aralığında rasgele</p>
4	$y(k+1) = \frac{24+y(k)}{30}y(k) - 0.8 \frac{u(k)^2}{1+u(k)^2}y(k-1) + 0.5$ <p>(Oussar, 1998)</p>	<p>[-5 5] Aralığında rasgele genlikli</p>	<p>[-5 5] Aralığında rasgele genlikli</p>
5	$y(k+1) = 0.5 * \left(\frac{y(k)}{1+y^2(k)} + (1+u(k))u(k)(1-u(k)) \right)$ <p>(Sastry, 1994)</p>	<p>[-2 2] Aralığında rasgele genlikli</p>	<p>[-2 2] Aralığında rasgele genlikli</p>

Bu çizelge de başarımlı kıyaslaması yapılırken kullanılacak olan dinamik sistemler ve bunların girişlerinde kullanılacak giriş dizileri olan $u(k)$ ' lar verilmiştir. Dinamik sistemler modellenirken ANFIS yapısı kullanılmış olup her bir örnek dinamik sistem için Çizelge 3.2' deki tanımlamalara göre ANFIS yapısı kurulmuştur. Sistem girişleri belirlendikten sonra her giriş için iki adet "Gauss" üyelik fonksiyonu U(F) ve kural sayıları tanımlanmıştır.

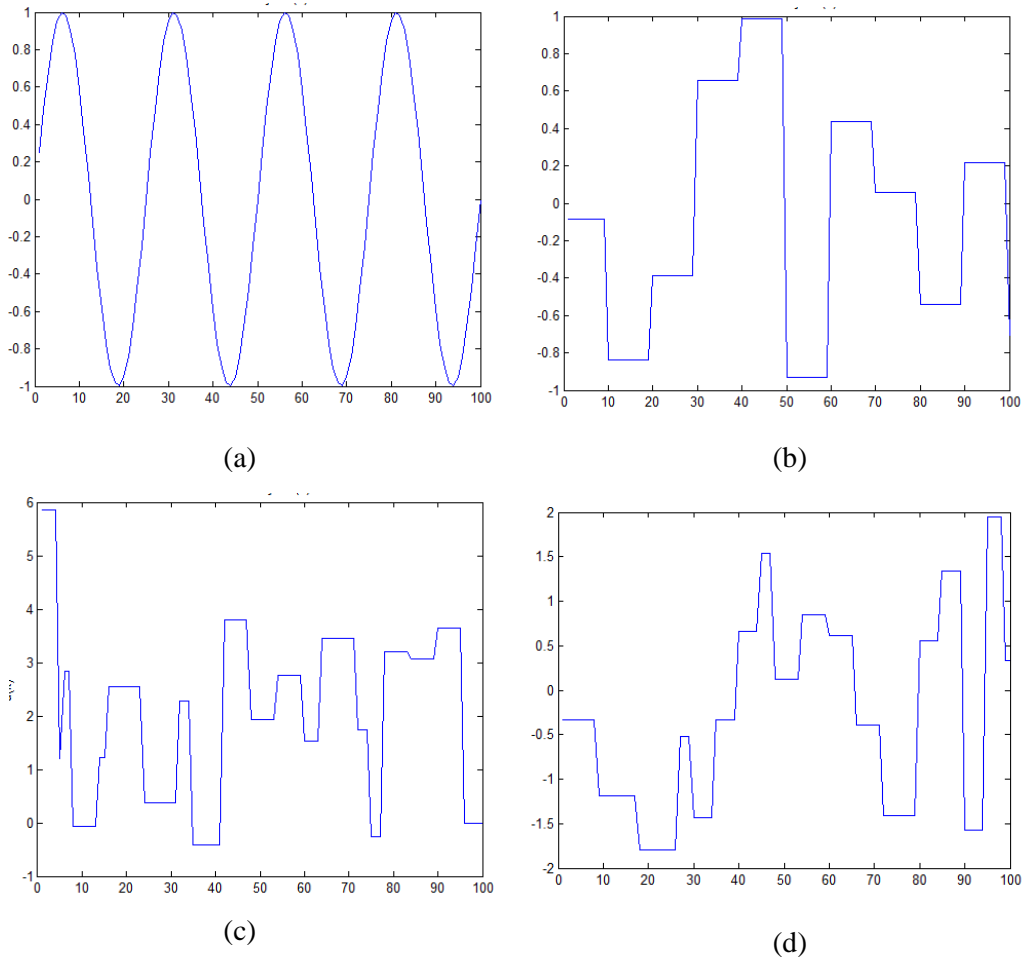
Çizelge 3.2. Her bir ÖDS için kullanılan ANFIS yapısı

ÖDS No	Girişler	Giriş ÜF Sayıları	Kural Sayıları	Parametre Sayısı (D)
1	$u(k), y(k-2), y(k-1)$	2, 2, 2	8	36
2	$u(k), y(k), y(k-1)$	2, 2, 2	8	36
3	$u(k), y(k)$	2, 2	4	20
4	$u(k), y(k), y(k-1)$	2, 2, 2	8	36
5	$u(k), y(k)$	2, 2	4	20

Örnek dinamik sistemler modellenirken ilk olarak eğitim veri seti hazırlanmış, bu eğitim seti ile Std-ICA, G1-ICA ve G2-ICA algoritmaları koşturulmuş ve elde edilen sonuçlar kaydedilmiştir. Örnek sistemlerin eğitimleri tamamlandıktan sonra eğitimlerinin başarısı daha iyi anlamak amacıyla eğitimini tamamlayan bütün sistemlerin eğitim setinden farklı olarak rastgele seçilen $u(k)$ dizileri kullanılarak ayrı ayrı test seti hazırlanmıştır. Hazırlanan bu test seti ile ANFIS sistem modelleri test edilmiştir. Şekil 3.1 ve Şekil 3.2’ de sırasıyla örnek dinamik sistemler için eğitim fazında ve test fazında kullanılan giriş dizileri verilmiştir.



Şekil 3.1. ÖDS sistemleri için sistem tanımanın *eğitim fazında* kullanılan giriş ($u(k)$) dizileri: (a) ÖDS 1 ve 2 için, (b) ÖDS 3 için $[-1 \ 1]$ aralığında rasgele genlikli 10 örnekleme periyotlu darbe (c) ÖDS 4 için $[-5 \ 5]$ aralığında rasgele genlikli ve rasgele örnekleme periyotlu darbe (d) ÖDS 5 için $[-2 \ 2]$ aralığında rasgele genlikli ve rasgele örnekleme periyotlu darbe



Şekil 3.2. ÖDS sistemleri için sistem tanımının *test fazında* kullanılan giriş ($u(k)$) dizileri: (a) ÖDS 1 ve 2 için, (b) ÖDS 3 için $[-1 \ 1]$ aralığında rasgele genlikli 10 örnekleme periyotlu darbe (c) ÖDS 4 için $[-5 \ 5]$ aralığında rasgele genlikli ve rasgele örnekleme periyotlu darbe (d) ÖDS 5 için $[-2 \ 2]$ aralığında rasgele genlikli ve rasgele örnekleme periyotlu darbe

3.2. ANFIS Bulanık Çıkarım Sistemi

Bilimsel çalışmalarda üzerinde çalışılan olaylar ve sistemler matematiksel modellerle tanımlanmaktadır. Oluşturulan matematiksel modeller, o olayın veya sistemin gelecekte alacağı durum ve göstereceği davranış biçimini de ortaya konmasını sağlamaktadır. Fakat günlük yaşantımızda karşı karşıya kalınan problemlerin birçoğu çeşitli sebeplerden dolayı modellenememekte ya da kesin bir durumu ifade etmekte yetersiz kalmaktadır. İşte karşı karşıya kaldığımız bu tip problemlerin incelenmesinde

ve çözümlenmesinde “Bulanık Mantık Yaklaşımı” kullanılabilir. (Türkbey, 2003; Şentürk, 2010)

İlk kez 1965 yılında L. A. Zadeh tarafından tanımlanan bulanık mantığın temeli, sözel ifadeler ve bunlar arasındaki mantıksal ilişkiler üzerine oluşturulmuştur. Sözel ifadeler aslında matematiksel bir temele dayandırılmıştır. Bu matematiksel temel, bulanık küme teorisi ve bulanık mantık olarak ifade edilmektedir. Bulanık mantık, bilinen klasik mantığın aksine 0 veya 1 olmak üzere iki seviyeli değil, $[0,1]$ aralığında çok seviyeli işlemlerin ifadesinde kullanılmıştır. (Elmas, 2003; Şentürk, 2010)

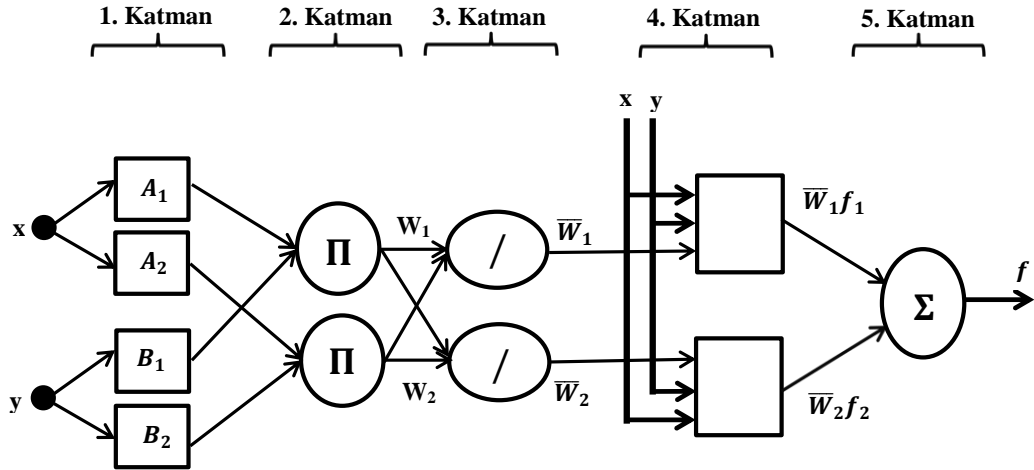
Bulanık mantık ile modelleme üç aşamadan oluşmaktadır. Bunlar: bulanıklaştırma, bulanık kural çıkarma ve durulama olarak isimlendirilmektedir. Bulanıklaştırma aşamasında, sistem girişlerinin üyelik fonksiyonları oluşturularak, uygun bulanık kümeler üyelik dereceleri belirlenmektedir. Bulanık kural çıkarma aşaması, üyelik fonksiyonlarını ve “eğer-öyleyse” kurallarını kullanarak, bulanık kümenin diğeriyle ilişkilendirilmesinden oluşmaktadır. Durulama aşaması ise bulanık çıkış değerinin, bir kesin çıkış değerine dönüştürülmesini ifade etmektedir. (Şenol ve Yıldırım, 2008)

Bulanık sistemler, genel anlamda, giriş değişkenlerinden çıkış değişkenlerini elde etmek amacıyla bulanık kümeleri kullanmaktadır. Bulanık çıkarım sistemlerinde kural tabanında kullanılan bulanık kuralların sonuç kısımları yardımıyla Bulanık Sistemleri; Mamdani ve Takagi-Sugeno-Kang olmak üzere ikiye ayırabiliriz. Sugeno tipi modelleme 1985 yılında kullanılmıştır. Mamdani tipi modellemenin uyarlaması şeklindedir ve giriş verilerine uygulanacak işlemler aynıdır. Tek farklılıkları çıktı verilerinden oluşmaktadır. Sugeno tipi modellemede çıktı değişkenleri girdilerin bir fonksiyonu şeklinde üyelik fonksiyonlarına sahip olmaktadır ve çıktı üyelik fonksiyonları doğrusal veya sabit olmalıdır (Çakıroğlu vd., 2011).

1993 yılında Jang tarafından geliştirilen ve parametreleri optimize edilerek kullanılan Sugeno tipi bulanık çıkarım sistemlerine, Adaptif Sinir Ağına Dayalı Bulanık Çıkarım Sistemi (Adaptive Neuro-Fuzzy Inference System-ANFIS) denilmektedir (Çakıroğlu vd., 2011). Aslında ANFIS, Sugeno tipi bulanık sistemlerin, sinirsel öğrenme yeteneğine sahip bir ağ yapısı temsiline oluşmaktadır. Bu ağ, her biri belli

bir fonksiyonu gerçekleştirmek üzere, katmanlar halinde yerleştirilmiş düğümlerin birleşiminden meydana gelmiştir. (Tsoukalas ve Uhrig, 1996; Özçalık ve Uygur, 2003).

Aşağıdaki Şekil 3.3’de iki giriş, iki kural, tek çıkışlı Sugeno Bulanık çıkarımına eşdeğer ANFIS mimarisi verilmiştir.



Şekil 3.3. İki girişli- tek çıkışlı birinci dereceden kural polinomlu ANFIS mimarisi

Bu sistemin parametreleri 1. Katmandaki üyelik fonksiyonlarının ve 4. Katmanda belirtilen kuralların parametreleridir. Bu parametreler uygun değere gelmesi durumunda sistem istenen $(x,y) \rightarrow z$ eşleşmesini gerçekleştirecektir. ANFIS ile ilgili detaylı bilgi (Jang, 1993)’den edinilebilir. (Yıldırım, 2012).

3.3. Kullanılan Algoritma

Bu çalışmada ICA algoritması ve bu algorithmadan yola çıkılarak geliştirilen iki yeni yaklaşım olan G1-ICA ve G2-ICA algoritmaları eğitim amaçlı olarak kullanarak örnek dinamik sistemleri ANFIS bulanık sistemi ile modellemeye çalışılmıştır. Bu üç algoritma ile ilgili temel bilgiler Bölüm 2’ de “Peaks Fonksiyon Optimizasyonu” başlığı altında verilmiştir.

ICA algoritmasının ve Std-ICA algoritmasından geliştirilen iki yaklaşımın (G1-ICA, G2-ICA) bulanık sistem modellemedeki başarımını kıyaslayabilmek için daha

önce benzer konuda yapılan çalışmadaki (Yıldırım, 2012) algoritma özellikleri, yapısı ve parametre değerleri benzer şekilde oluşturulmuştur. Algoritmalarındaki toplam koloni sayısı 40 alınmış emperyalist sayısı 4 olarak belirlenmiştir. Algoritma 1000 kez 50 tekrarlı olarak çalıştırılmıştır. Buna göre üç ICA algoritması için Çizelge 3.3’ de verilen parametre değerleri kullanılmıştır.

Çizelge 3.3. ICA parametreleri

Parametre Adı	Parametre Değeri
Ülke Sayısı	40
Emperyalist Sayısı	4
İterasyon Sayısı	1000
Algoritma Tekrar Sayısı	50

3.4. Bireylerin Yapısı

Çalışmada dinamik sistem modelleme aracı olarak Şekil 3.3’ de verilen ANFIS yapısı kullanılmıştır. ANFIS bulanık mantık modellemesi yapılırken örnek sistemlerin durumuna göre belirlenen girişler için ikişer adet üyelik fonksiyonu tanımlanmış ve bunlara ait kural sayıları Çizelge 3.2’de verilmiştir. Buna göre Çizelge 3.1’de verilen her bir ÖDS için Çizelge 3.2’de görüldüğü gibi girişler, giriş üyelik fonksiyon sayıları, belirlenen üyelik fonksiyonuna göre kural ve parametre sayısı belirlenmiştir. ANFIS bulanık mantık modelinde sistemleri modelleyebilmek için Sugeno yöntemi ve üyelik fonksiyonu olarak yaygın olarak kullanılan gauss fonksiyonu seçilmiştir. Buna göre ANFIS optimizasyonu için kullanılan bireylerin yapısı aşağıdaki gibidir.

$$b_i = [c_{i1} \ c_{i2} \ c_{i3} \ \dots \ \sigma_{i1} \ \sigma_{i2} \ \sigma_{i3} \ \dots \ p_1 \ q_1 \ r_1 \ \dots \ p_R \ q_R \ r_R]$$

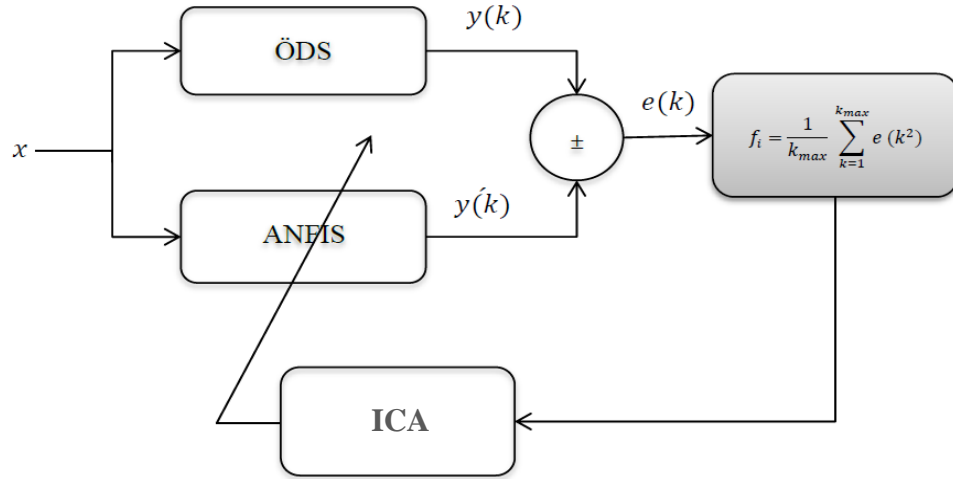
c_{ij} = i. Girişin j. ÜF’sinin merkezini;

σ_{ij} = i. Girişin j.ÜF’sinin standart sapmasını;

$p_k \ q_k \ r_k$ ise k. kuralın sonuç (kural) parametrelerini gösterir (Yıldırım, 2012).

3.5. ICA ile ANFIS Parametre Optimizasyonu

Şekil 3.4’de optimizasyon için kullanılan blok şeması görülmektedir



Şekil 3.4. ICA ile ANFIS optimizasyonu blok şeması(Yıldırım, 2012).

4. SEZGİSEL ARAMA ALGORİTMA TABANLI BULANIK SİSTEM OPTİMİZASYONU

Bu bölümde ICA algoritmasının ÖDS üzerindeki başarımı incelenmiştir. İnceleme Bölüm 3 de verilen 5 farklı dinamik sistemin bulanık modellenmesi üzerine kurulmuştur. Daha önce benzer konuda yapılan çalışma (Yıldırım, 2012) temel alınmış olup kullanılan popülasyon büyüklüğü, maksimum nesil sayısı ve algoritmalarda kullanılan diğer parametreler Bölüm 3’de de verildiği gibi aynen o çalışmaya uygun olarak hazırlanmıştır. Buna göre ICA algoritmasında da popülasyon büyüklüğü (ülke sayısı) 40, maksimum nesil sayısı (iterasyon sayısı) 1000 olarak alınmış olup, her bir dinamik sistem için Std-ICA, G1-ICA ve G2-ICA algoritmaları ayrı ayrı 50’ şer kez koşturularak elde edilen sonuçlar istatistiki olarak incelenmiştir.

Bu incelemeler esnasında Bölüm 3 te Çizelge 3.1’ te verilen örnek dinamik sistemler ANFIS ile bulanık modellenerek elde sonuçlar en iyi, en kötü, ilk durum ve son durum olarak sınıflandırılarak aşağıda grafiksel olarak detaylandırılmıştır. Buna göre mavi düz çizgiler beklenen sonuç iken, kırmızı kesik çizgiler elde edilen sonucu işaret etmektedir.

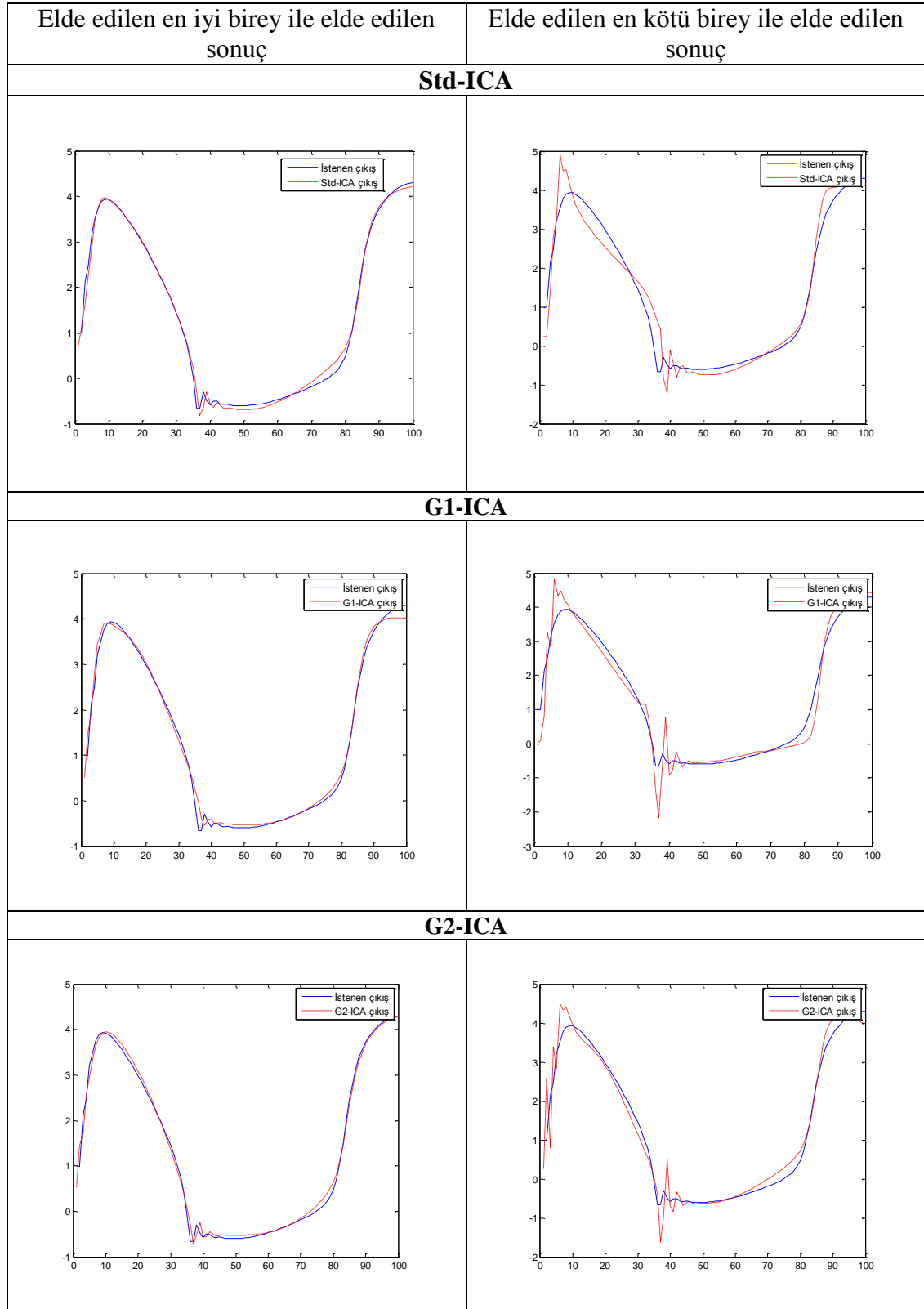
ICA algoritması ve geliştirilen iki yeni yaklaşımda elde edilen sonuçlar grafiksel olarak gösterildikten sonra yine sonuç değerleri de bir çizelge üzerinde her bir ÖDS için gruplandırılarak eğitim ve test fazları için başarımların kıyaslaması yapılmıştır. Daha sonrasında ise ICA algoritmalarının, (Yıldırım, 2012) çalışmasındaki diğer sezgisel arama algoritmalarıyla olan başarımların kıyaslaması Bölüm 4 Çizelge 4.3 ve Çizelge 4.4 te yapılmıştır.

4.1. ÖDS 1 'in Tanınması/Modellenmesi

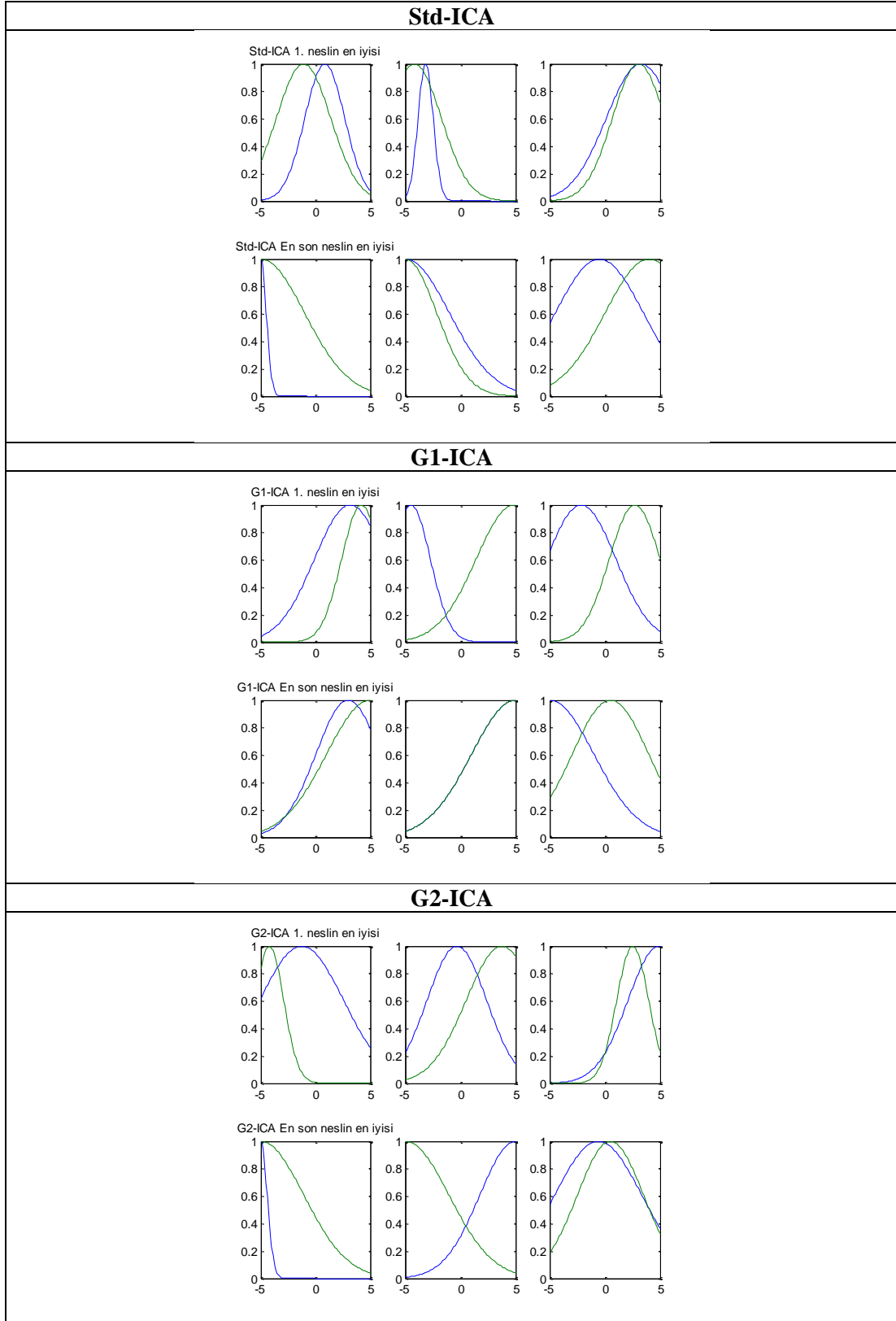
4.1.1.ÖDS 1 için eğitim aşaması ve sonuçları

Bu bölümde Bölüm 3 Şekil 3.1 (a) da verilen giriş dizilerinden oluşturulan eğitim seti dizileri ÖDS 1 için çalıştırılarak başarımları sonuçları incelenmiştir. Buna göre aşağıda Şekil 4.1' de ICA öğrenmeli ANFIS ile ÖDS 1 eğitim fazı için elde edilen sonuçlar grafiksel olarak verilmiştir. Grafikler, elde edilen en iyi çözüm ve en kötü çözüm olmak üzere ikiye ayrılarak düzenlenmiştir. Grafikler incelendiğinde Std-ICA, G1-ICA ve G2-ICA algoritmalarının sonuçları arasında çok bariz farklılıklar olmadığı, birbirine yakın sonuçlar ürettikleri fakat en iyi sonucu Std-ICA algoritması verdiği görülmüştür.

Şekil 4.2' de ise Std-ICA, G1-ICA ve G2-ICA' nın ÖDS 1 için bulanık modelleme eğitim fazındaki üyelik fonksiyonlarının (UF) ilk durumları ve son durumları grafiksel olarak hazırlanmıştır. Bu sayede her algoritmanın üyelik fonksiyonlarında değişim gözlenmek istenmiştir. Buna göre ilk grafikler; 1. Giriş' e ait, ortadaki grafikler; 2.Giriş' e ve son grafikler de 3. Giriş' e ait üyelik fonksiyonlarının ilk ve son durumlarını göstermektedir. Mavi renkli çizgi birinci üyelik fonksiyonunu, yeşil renkli çizgi ikinci üyelik fonksiyonunu ifade etmektedir.

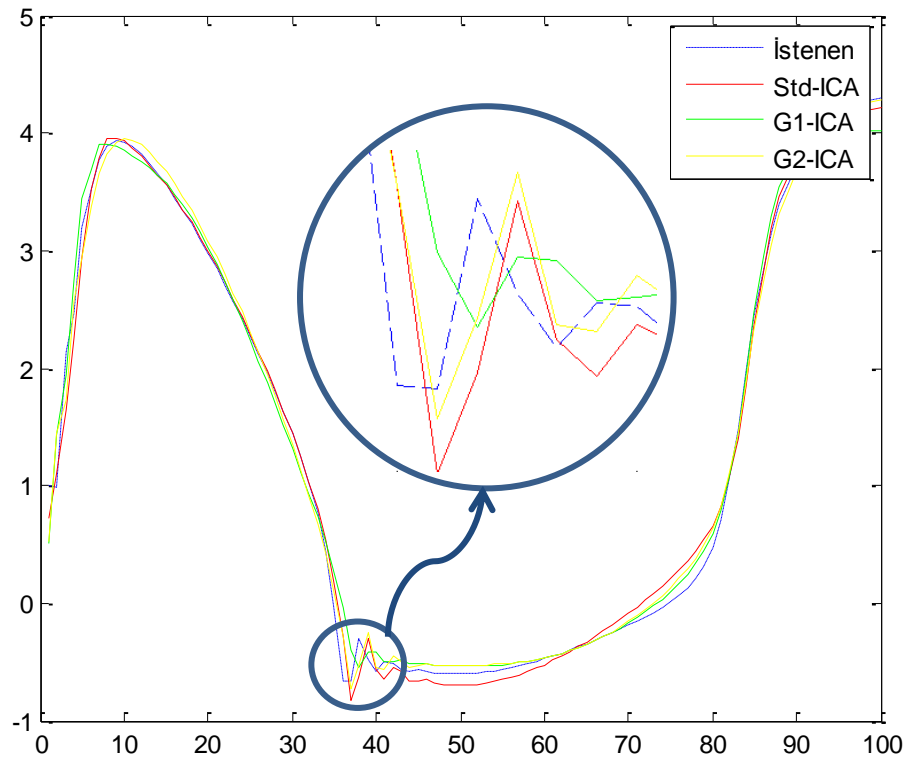


Şekil 4.1. ICA öğrenmeli ANFIS ile ÖDS 1 için eğitim fazı sistem tanıma sonuçları.



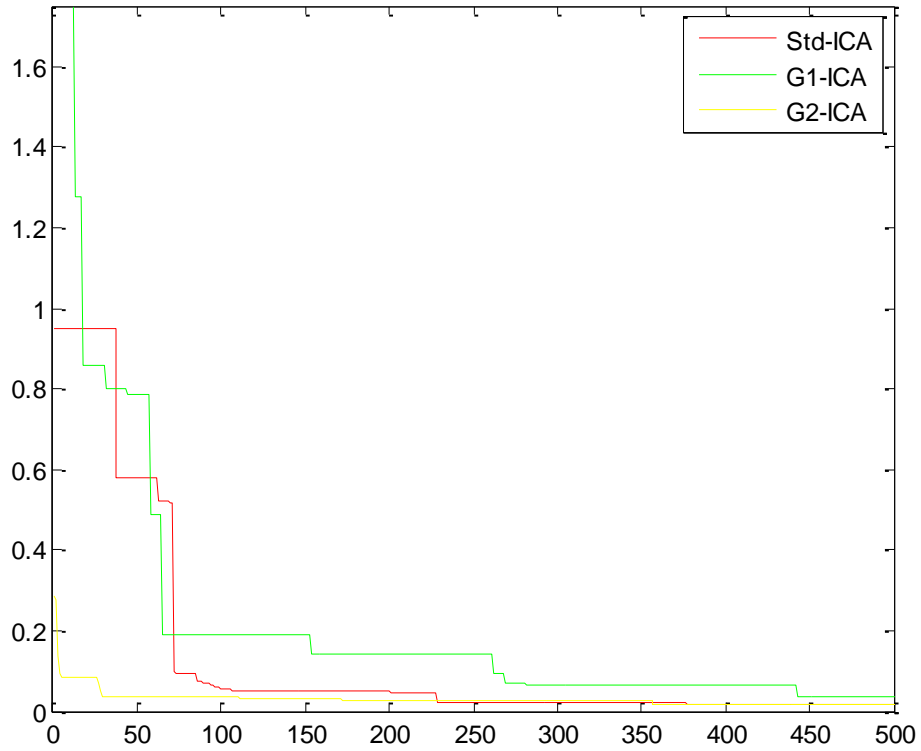
Şekil 4.2. ICA öğrenmeli ANFIS ile ÖDS 1'in bulanık modelleme eğitim fazında UF'lerin başlangıç ve son durumları.

Yukarıdaki grafiklerle gösterilmeye çalışılan, Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ÖDS 1 modellemesinin başarımlarını kıyaslamayı kolaylaştırmak amacıyla, bu algoritmaların ayrı ayrı koşturulması sonucunda elde edilen en iyi ölçüt değerlerine sahip modelleme sonuçlarının karşılaştırılmasının yapıldığı grafik Şekil 4.3’te verilmiştir. Grafik üzerinde kırılganlığın olduğu kısım büyütülerek daha net bir gözlem yapılması sağlanmıştır. Buna göre üç algoritmanın da istenene yakın sonuçlar ürettikleri görülmektedir.



Şekil 4.3. Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 1 için elde edilen modellerin karşılaştırılması

Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 1 modelleme eğitim seyrinde yakınsama karşılaştırılmasının yapıldığı grafik Şekil 4.4’te verilmiştir. Buna göre elde edilen en iyi ölçüt değerlerinin her bir adımdaki değişimi gözlenmiştir. Bu konuda G2-ICA’nın başarısı rahatlıkla görülmektedir.



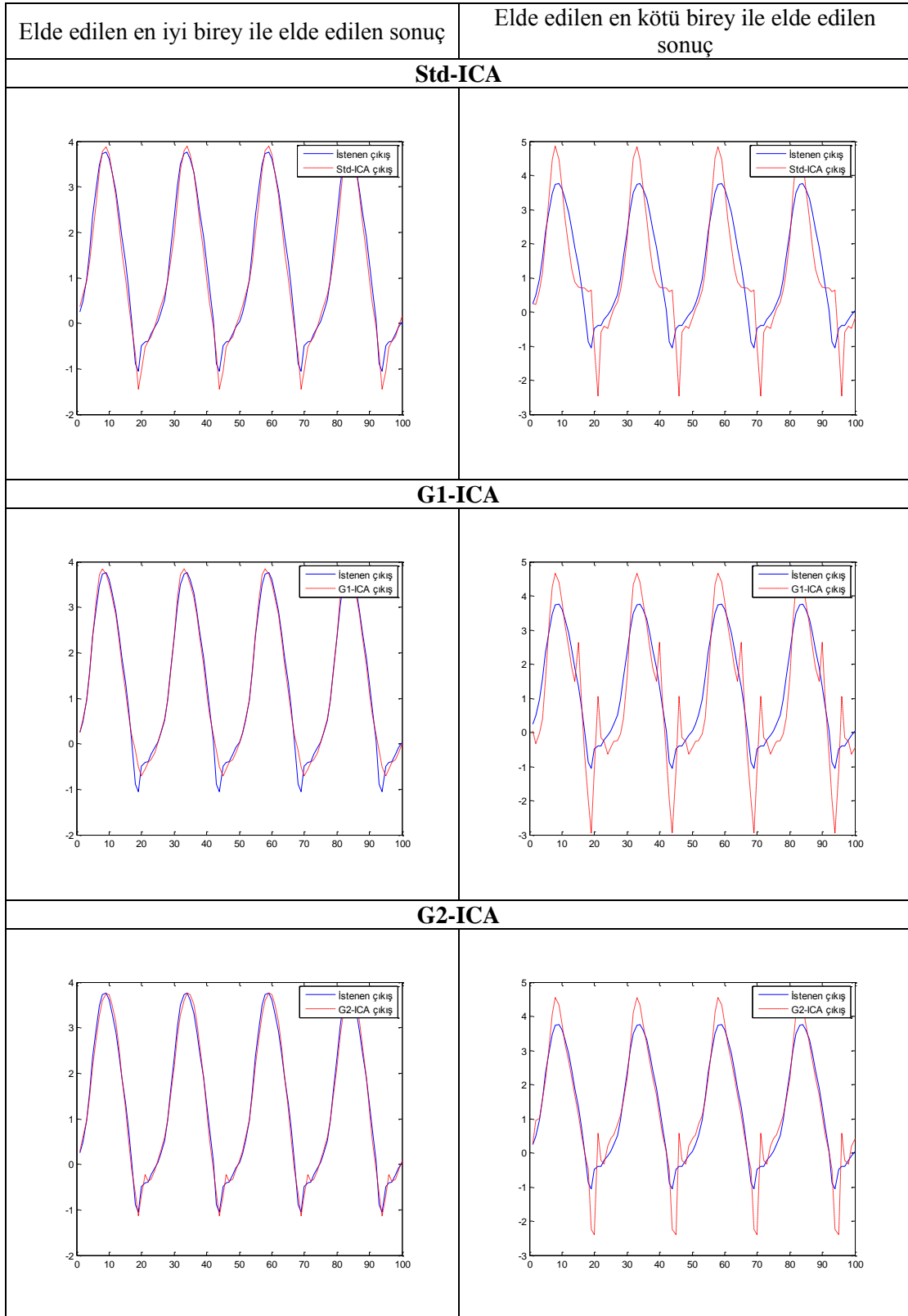
Şekil 4.4. Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 1 modelleme eğitim seyri

4.1.2.ÖDS 1'in test aşaması ve sonuçları

Bölüm 3 Şekil 3.2 (a) da verilen giriş dizisi ($u(k)$) kullanılarak oluşturulan veri seti ile ÖDS 1' in ICA öğrenmeli ANFIS modelindeki çıkışları ve performansı incelenmiştir. ICA algoritmalarının verdiği çıkış cevapları kendi aralarında karşılaştırılarak en başarılı sonucu hangi ICA algoritmasının verdiği incelenmiştir

ÖDS 1 için Bölüm 3 Şekil 3.1 (a) da verilen $u(k)$ ile eğitim seti hazırlanmıştır. Test seti için ise 3 Şekil 3.2 (a) da verilen $u(k)$ ile yine Bölüm 3 de yer alan Çizelge 3.1 de belirlenen girişler göz önüne alınarak test veri seti hazırlanmıştır.

Meydana getirilen test seti, Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 1 modelleme sonuçları Şekil 4.5'de grafiksel olarak verilmiştir. Grafikler incelendiğinde ICA öğrenmeli ANFIS ile ÖDS 1 için test sistem tanıma çalışmasında en başarılı sonucu G2-ICA algoritmasının verdiği görülmüştür.



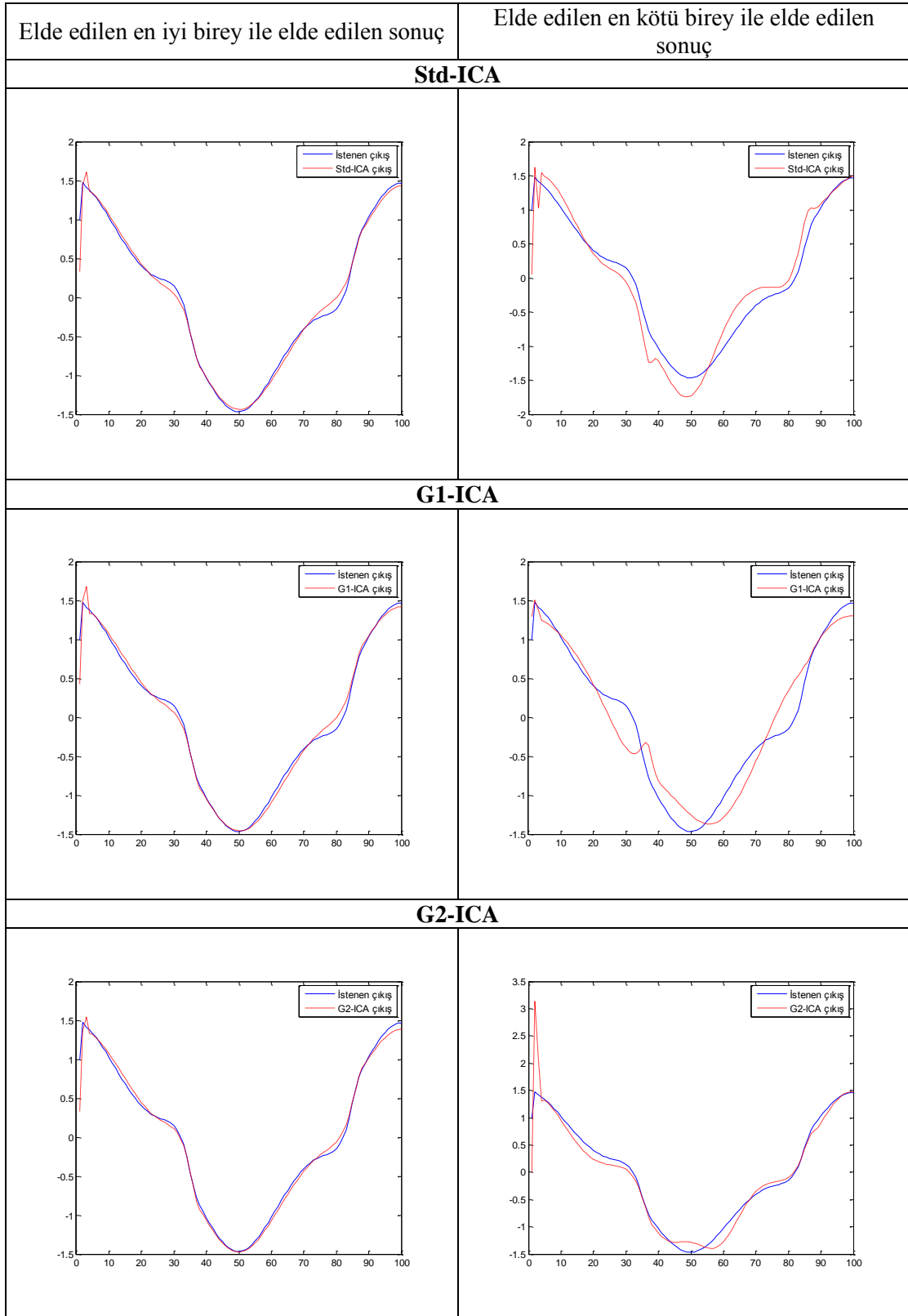
Şekil 4.5. ICA öğrenmeli ANFIS ile ÖDS 1 için test fazı sistem tanıma sonuçları

4.2. ÖDS 2' nin Tanınması/Modellenmesi

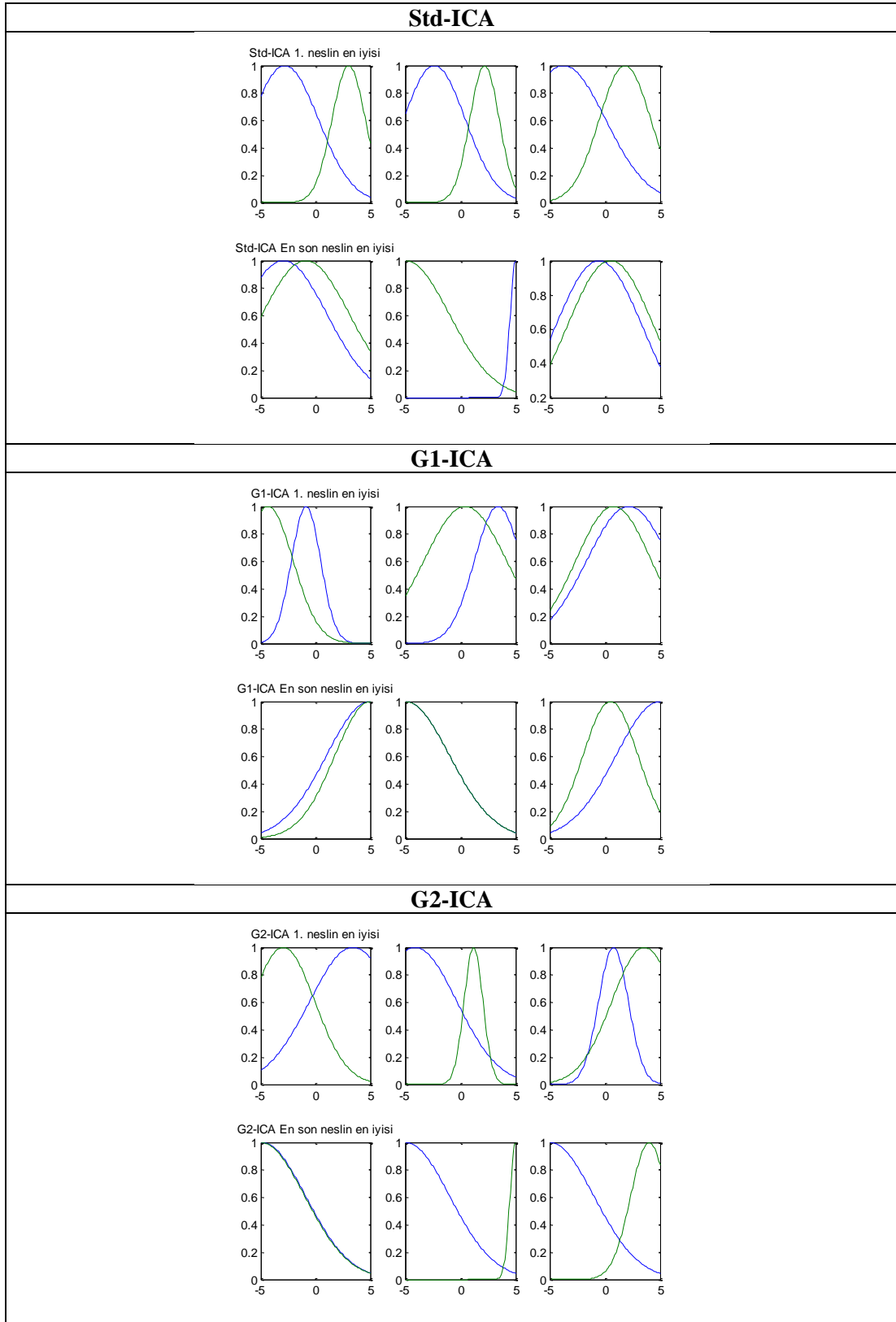
4.2.2.ÖDS 2 için eğitim aşaması ve sonuçları

Bu bölümde Bölüm 3 Şekil 3.1 (a) da verilen giriş dizilerinden oluşturulan eğitim seti dizileri ÖDS 2 için çalıştırılarak başarımları incelenmiştir. Buna göre aşağıda Şekil 4.6' da ICA öğrenmeli ANFIS ile ÖDS 2 eğitim fazı için elde edilen sonuçlar grafiksel olarak verilmiştir. Grafikler, elde edilen en iyi çözüm ve en kötü çözüm olmak üzere ikiye ayrılarak düzenlenmiştir. Grafikler incelendiğinde Std-ICA, G1-ICA ve G2-ICA algoritmalarının sonuçları arasında çok bariz farklılıklar olmadığı, birbirine yakın sonuçlar ürettikleri fakat en iyi sonucu G2-ICA algoritmasının verdiği görülmüştür.

Şekil 4.7 de ise Std-ICA, G1-ICA ve G2-ICA' nın ÖDS 2 için bulanık modelleme eğitim fazındaki üyelik fonksiyonlarının (UF) ilk durumları ve son durumları grafiksel olarak hazırlanmıştır. Bu sayede her algoritmanın üyelik fonksiyonlarında değişim gözlenmek istenmiştir. Buna göre ilk grafikler; 1. Giriş' e ait, ortadaki grafikler; 2.Giriş' e ve son grafikler de 3. Giriş' e ait üyelik fonksiyonlarının ilk ve son durumlarını göstermektedir. Mavi renkli çizgi birinci üyelik fonksiyonunu, yeşil renkli çizgi ikinci üyelik fonksiyonunu ifade etmektedir.

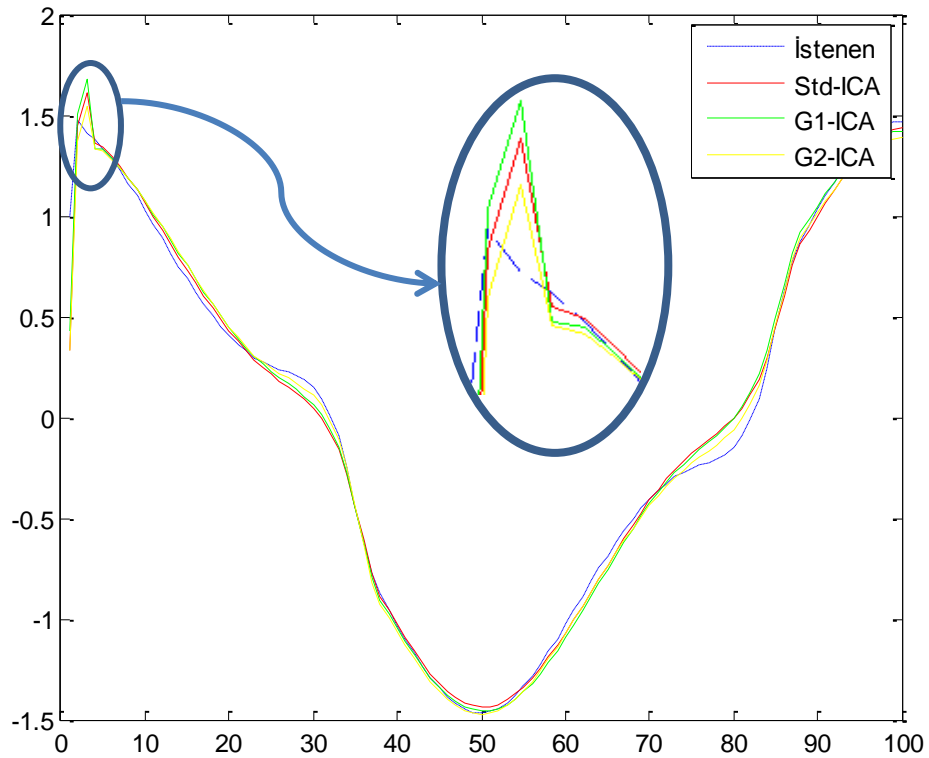


Şekil 4.6. ICA öğrenmeli ANFIS ile ÖDS 2 için eğitim fazı sistem tanıma sonuçları.



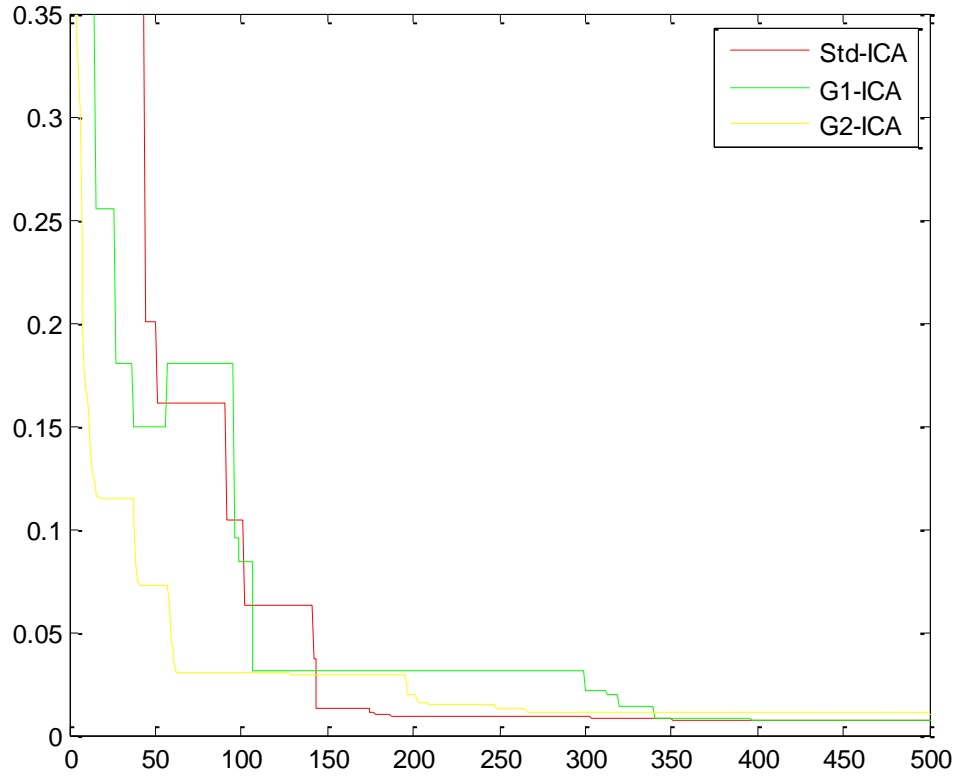
Şekil 4.7. ICA öğrenmeli ANFIS ile ÖDS 2'in bulanık modelleme eğitim fazında UF'lerin başlangıç ve son durumları.

Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ÖDS 2 modellemesinin başarımlarını kıyaslamasını kolaylaştırmak amacıyla, bu algoritmaların ayrı ayrı koşturulması sonucunda elde edilen en iyi ölçüt değerlerine sahip modelleme sonuçlarının karşılaştırılmasının yapıldığı grafik Şekil 4.8’ de verilmiştir. Grafik üzerinde kırılmanın olduğu kısım büyütülerek daha net bir gözlem yapılması sağlanmıştır. Buna göre üç algoritmanın da istenene yakın sonuçlar ürettikleri görülmektedir.



Şekil 4.8. Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 2 için elde edilen modellerin karşılaştırılması

Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 2 modelleme eğitim seyrinde yakınsama karşılaştırılmasının yapıldığı grafik Şekil 4.9’ da verilmiştir. Buna göre elde edilen en iyi ölçüt değerlerinin her bir adımdaki değişimi gözlenmiştir. Bu konuda Std-ICA ve G2-ICA’ nın G1-ICA’ a nazaran daha erken istenen değerlere yakınsadığı görülmüştür.

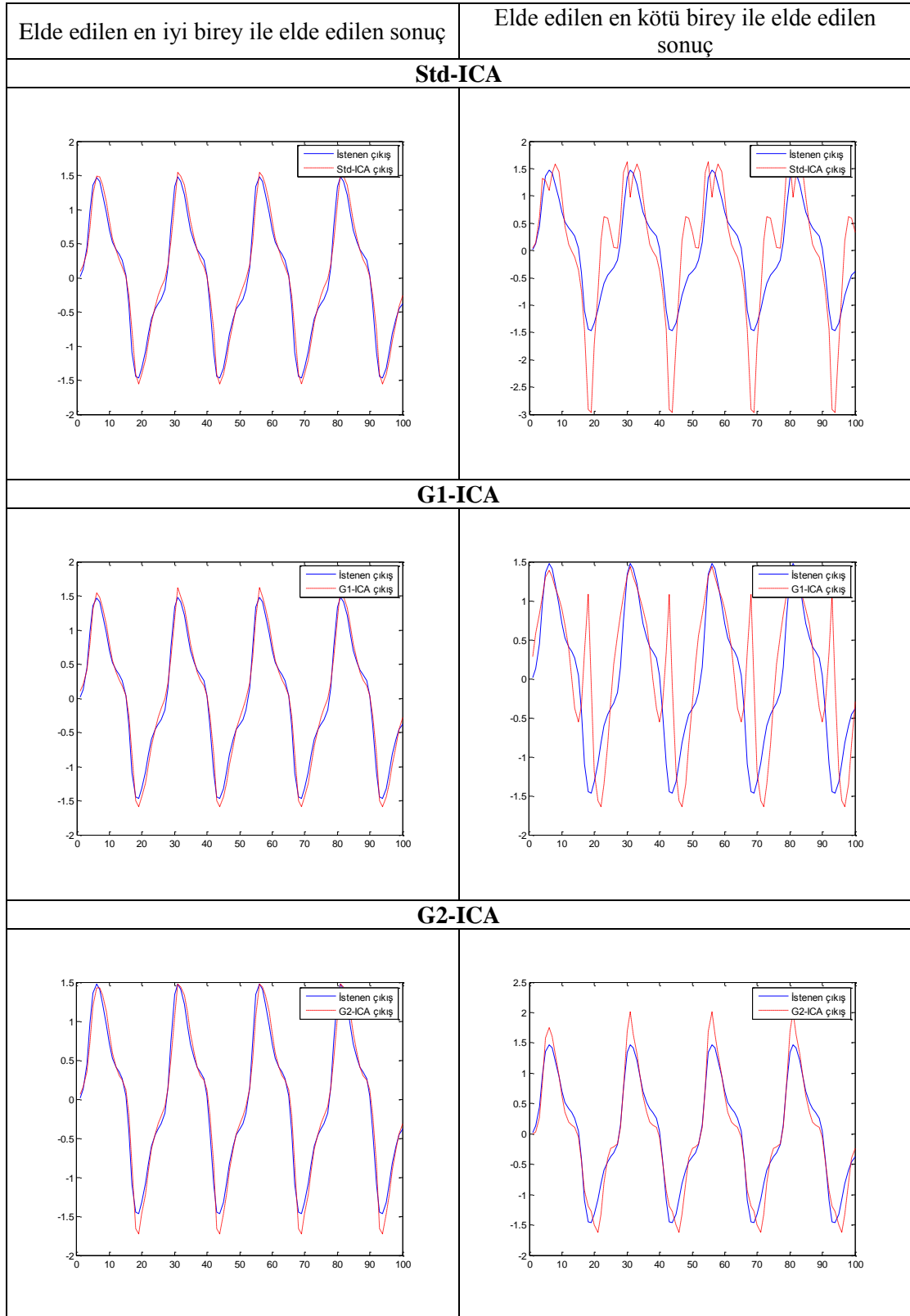


Şekil 4.9. Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 2 modelleme eğitim seyri

4.2.2.ÖDS 2' in test aşaması ve sonuçları

ÖDS 2 için Bölüm 3 Şekil 3.1 (a) da verilen $u(k)$ ile eğitim seti hazırlanmıştı. Test seti için ise Bölüm 3 Şekil 3.2 (a) da verilen $u(k)$ ile yine Bölüm 3 de yer alan Çizelge 3.1' de belirlenen girişler göz önüne alınarak test veri seti hazırlanmıştır.

Meydana getirilen test seti, Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 1 modelleme sonuçları Şekil 4.10' de grafiksel olarak verilmiştir. Grafikler incelendiğinde ICA öğrenmeli ANFIS ile ÖDS 2 için test sistem tanıma çalışmasında en başarılı sonucu Std-ICA algoritmasının verdiği görülmüştür.



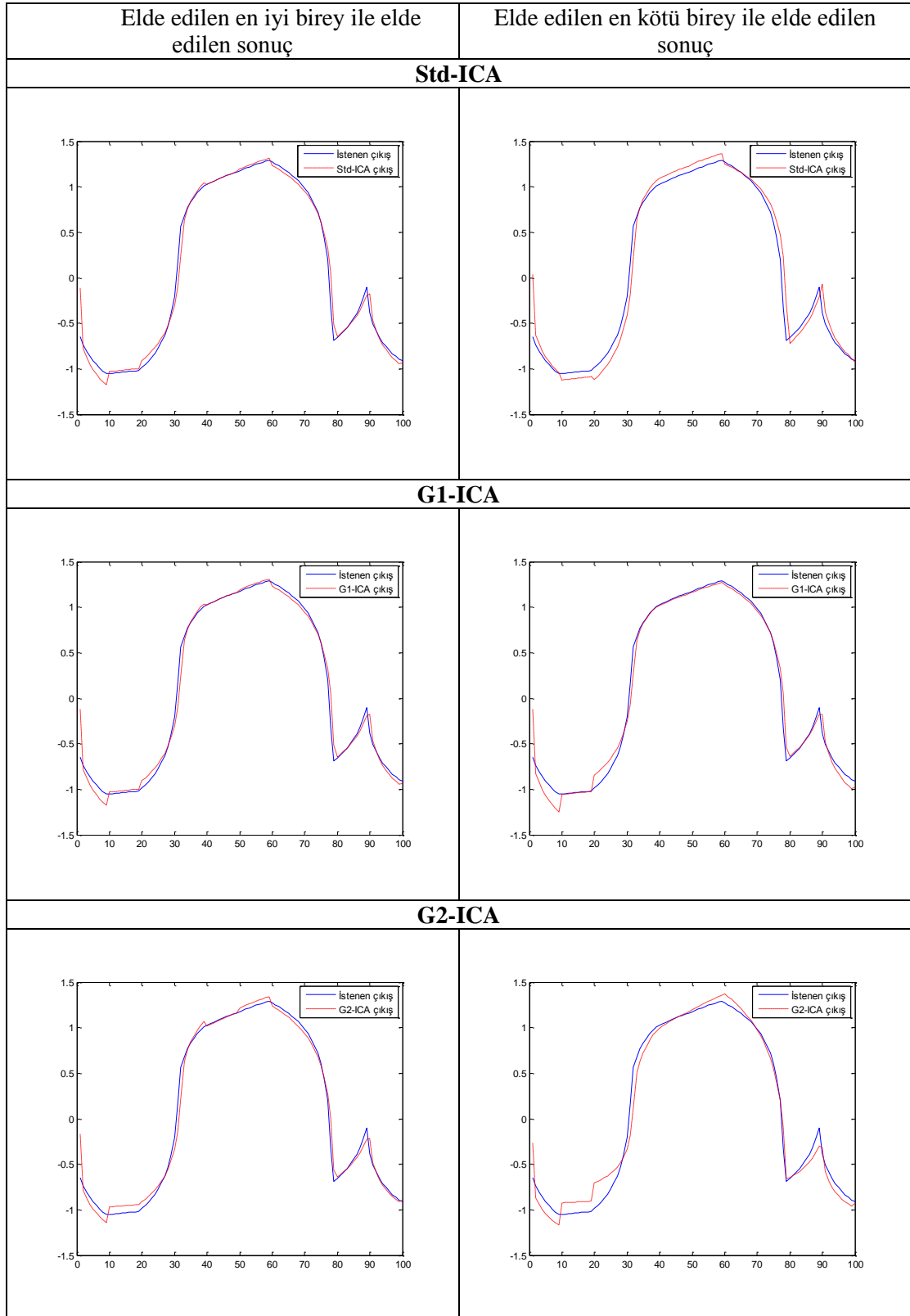
Şekil 4.10. ICA öğrenmeli ANFIS ile ÖDS 2 için test fazı sistem tanıma sonuçları

4.3. ÖDS 3 'in Tanınması/Modellenmesi

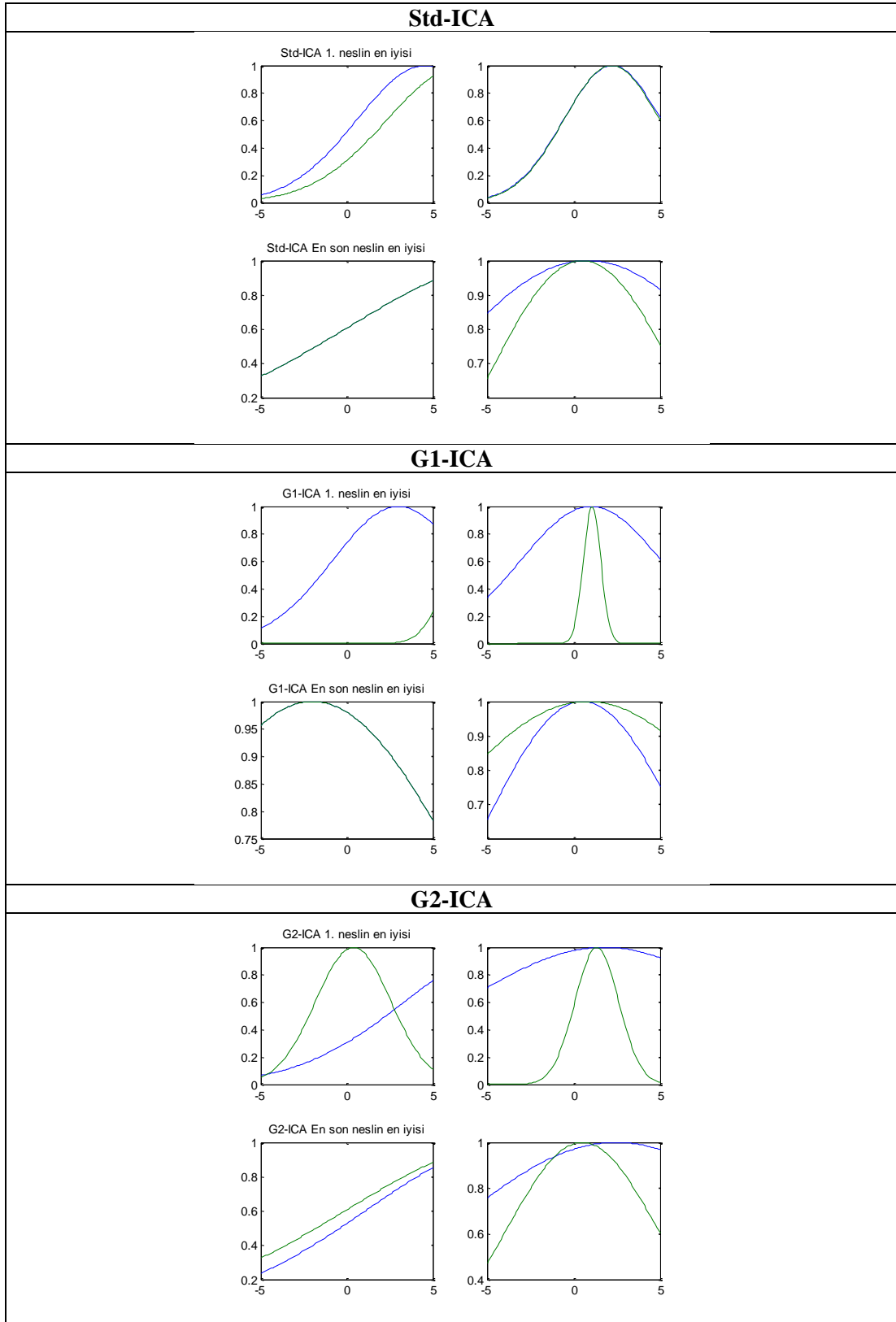
4.3.1.ÖDS 3 için eğitim aşaması ve sonuçları

Bu bölümde Bölüm 3 Şekil 3.1 (b) de verilen giriş dizilerinden oluşturulan eğitim seti dizileri ÖDS 3 için çalıştırılarak başarımları sonuçları incelenmiştir. Buna göre aşağıda Şekil 4.11' de ICA öğrenmeli ANFIS ile ÖDS 3 eğitim fazı için elde edilen sonuçlar grafiksel olarak verilmiştir. Grafikler, elde edilen en iyi çözüm ve en kötü çözüm olmak üzere ikiye ayrılarak düzenlenmiştir. Grafikler incelendiğinde Std-ICA, G1-ICA ve G2-ICA algoritmalarının sonuçları arasında çok bariz farklılıklar olmadığı, birbirine yakın sonuçlar ürettikleri fakat G1-ICA algoritmasının başarımlarının daha yüksek olduğu görülmüştür.

Şekil 4.12' de ise Std-ICA, G1-ICA ve G2-ICA' nın ÖDS 3 için bulanık modelleme eğitim fazındaki üyelik fonksiyonlarının (UF) ilk durumları ve son durumları grafiksel olarak hazırlanmıştır. Bu sayede her algoritmanın üyelik fonksiyonlarında değişim gözlenmek istenmiştir. Buna göre ilk grafikler; 1. Giriş' e ait, son grafikler ise; 2. Giriş' e ait üyelik fonksiyonlarının ilk ve son durumlarını göstermektedir. Mavi renkli çizgi birinci üyelik fonksiyonunu, yeşil renkli çizgi ikinci üyelik fonksiyonunu ifade etmektedir.

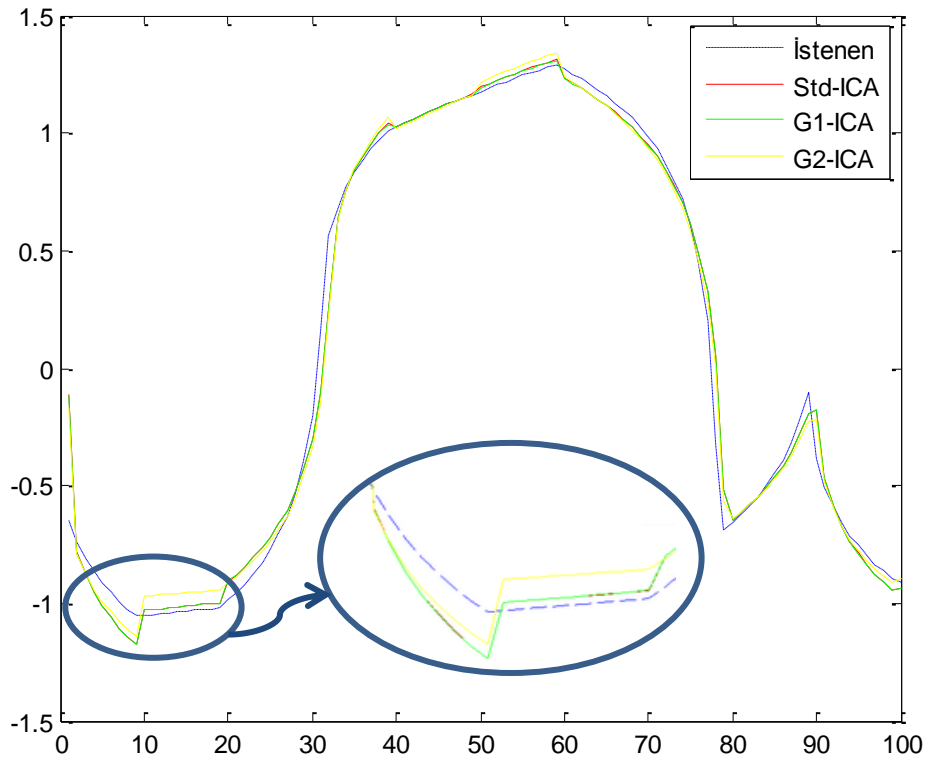


Şekil 4.11. ICA öğrenmeli ANFIS ile ÖDS 3 için eğitim fazı sistem tanıma sonuçları.



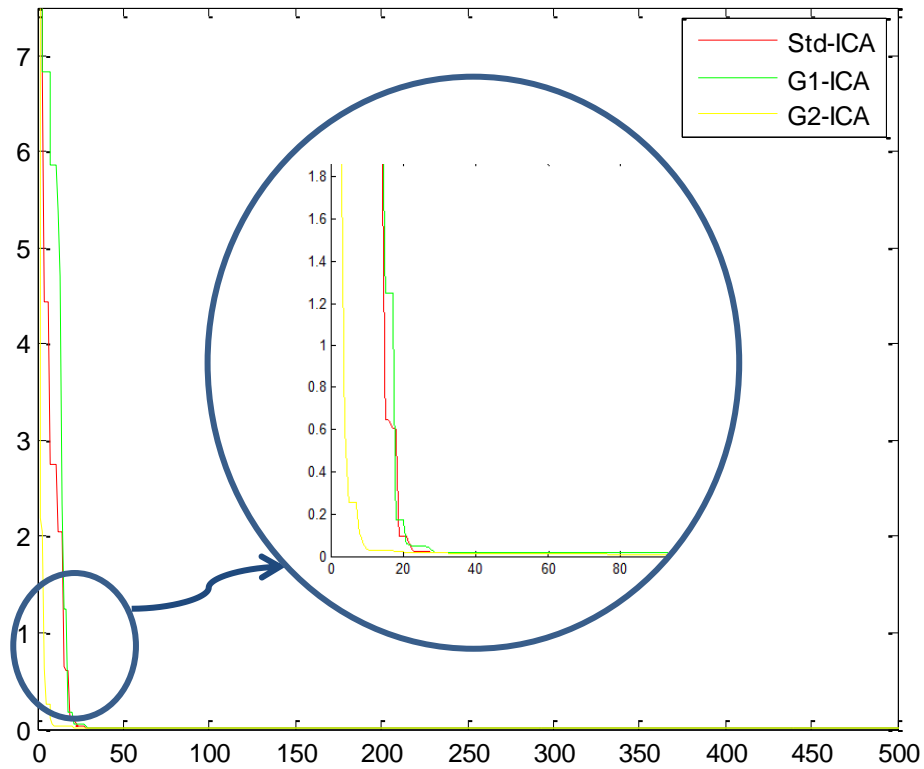
Şekil 4.12. ICA öğrenmeli ANFIS ile ÖDS 3'in bulanık modelleme eğitim fazında UF'lerin başlangıç ve son durumları.

Yukarıdaki grafiklerle gösterilmeye çalışılan, Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ÖDS 3 modellemesinin başarımlarını kıyaslamayı kolaylaştırmak amacıyla, bu algoritmaların ayrı ayrı koşturulması sonucunda elde edilen en iyi ölçüt değerlerine sahip modelleme sonuçlarının karşılaştırılmasının yapıldığı grafik Şekil 4.13’ te verilmiştir. Grafik üzerinde kırılmanın olduğu kısım büyütülerek daha net bir gözlem yapılması sağlanmıştır. Buna göre üç algoritmanın da istenene yakın sonuçlar ürettikleri görülmektedir.



Şekil 4.13. Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 3 için elde edilen modellerin karşılaştırılması

Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 3 modelleme eğitim seyirinde yakınsama karşılaştırılmasının yapıldığı grafik Şekil 4.14’ te verilmiştir. Buna göre elde edilen en iyi ölçüt değerlerinin her bir adımdaki değişimi gözlenmiştir. Bu konuda G2-ICA’ nın daha erken istenene değerlere yakınsadığı görülmüştür.

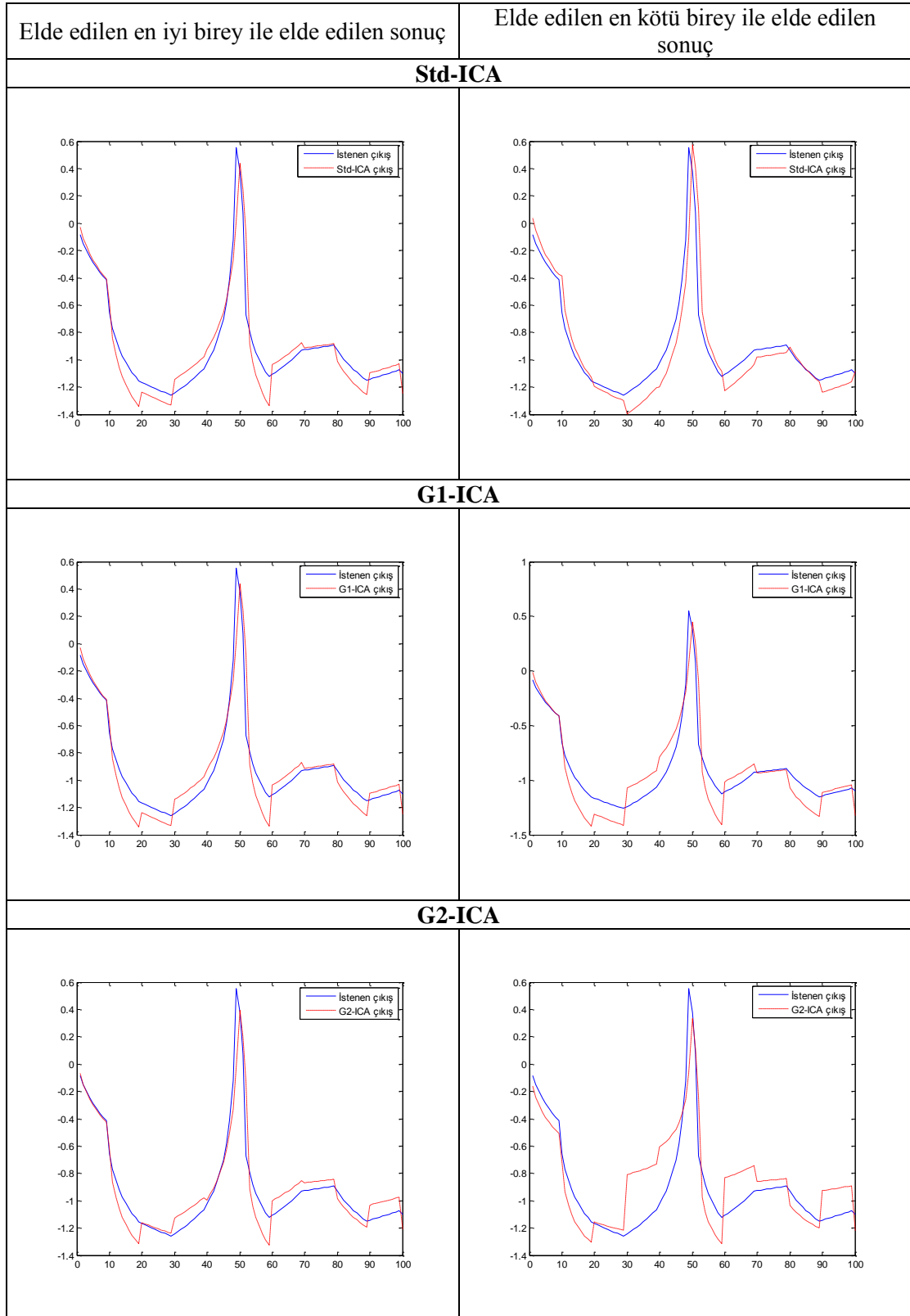


Şekil 4.14. Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 3 modelleme eğitim seyri

4.3.2.ÖDS 3'in test aşaması ve sonuçları

ÖDS 3 için Bölüm 3 Şekil 3.1 (b) de verilen $u(k)$ ile eğitim seti hazırlanmıştır. Test seti için ise Bölüm 3 Şekil 3.2 (b) de verilen $u(k)$ ile yine Bölüm 3 de yer alan Çizelge 3.1 de belirlenen girişler göz önüne alınarak test veri seti hazırlanmıştır.

Meydana getirilen test seti, Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 3 modelleme sonuçları Şekil 4.15' de grafiksel olarak verilmiştir. Grafikler incelendiğinde ICA öğrenmeli ANFIS ile ÖDS 3 için test sistem tanıma çalışmasında en başarılı sonucu G2-ICA algoritmasının verdiği görülmektedir.



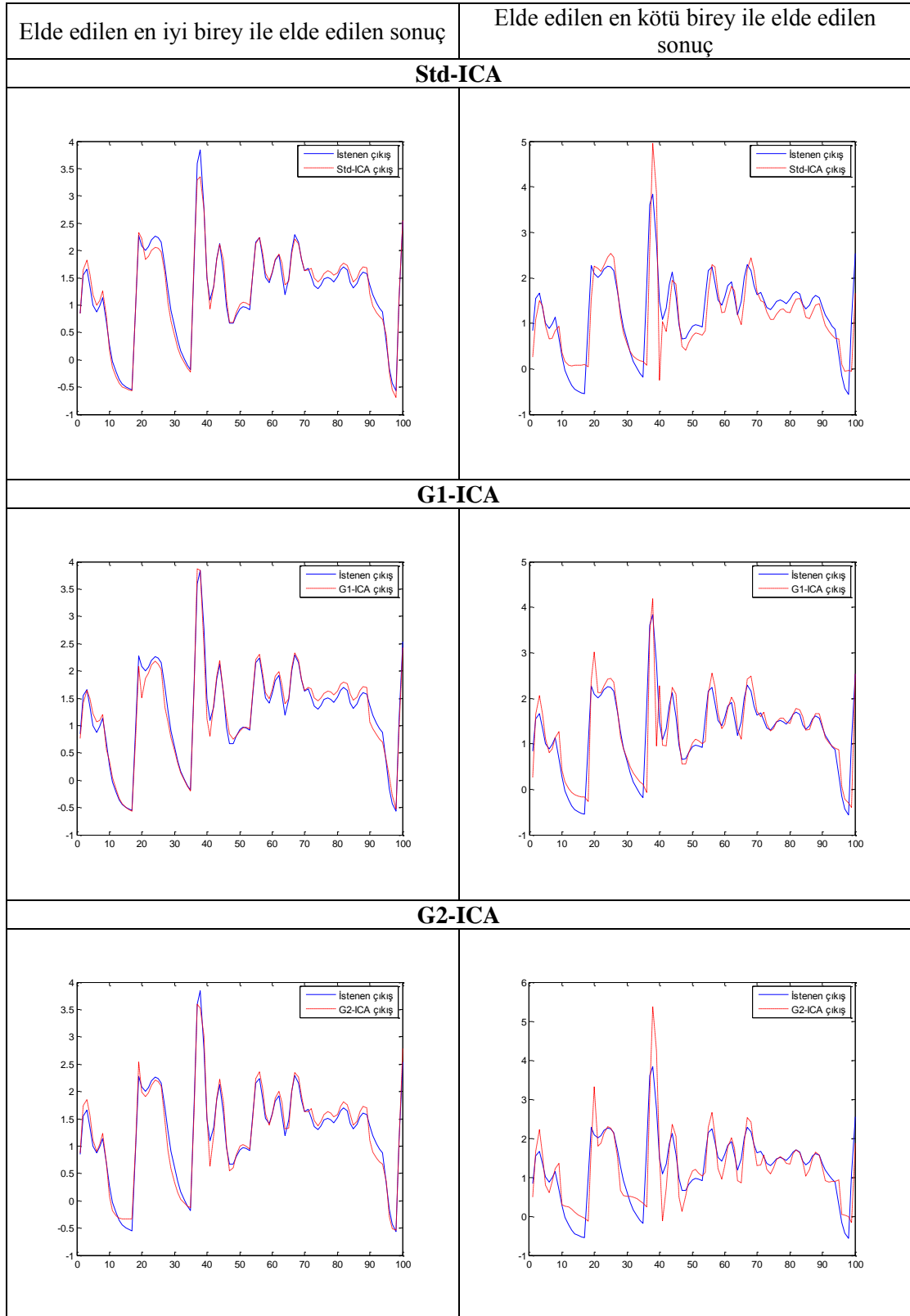
Şekil 4.15. ICA öğrenmeli ANFIS ile ÖDS 3 için test fazı sistem tanıma sonuçları

4.4. ÖDS 4 'in Tanınması/Modellenmesi

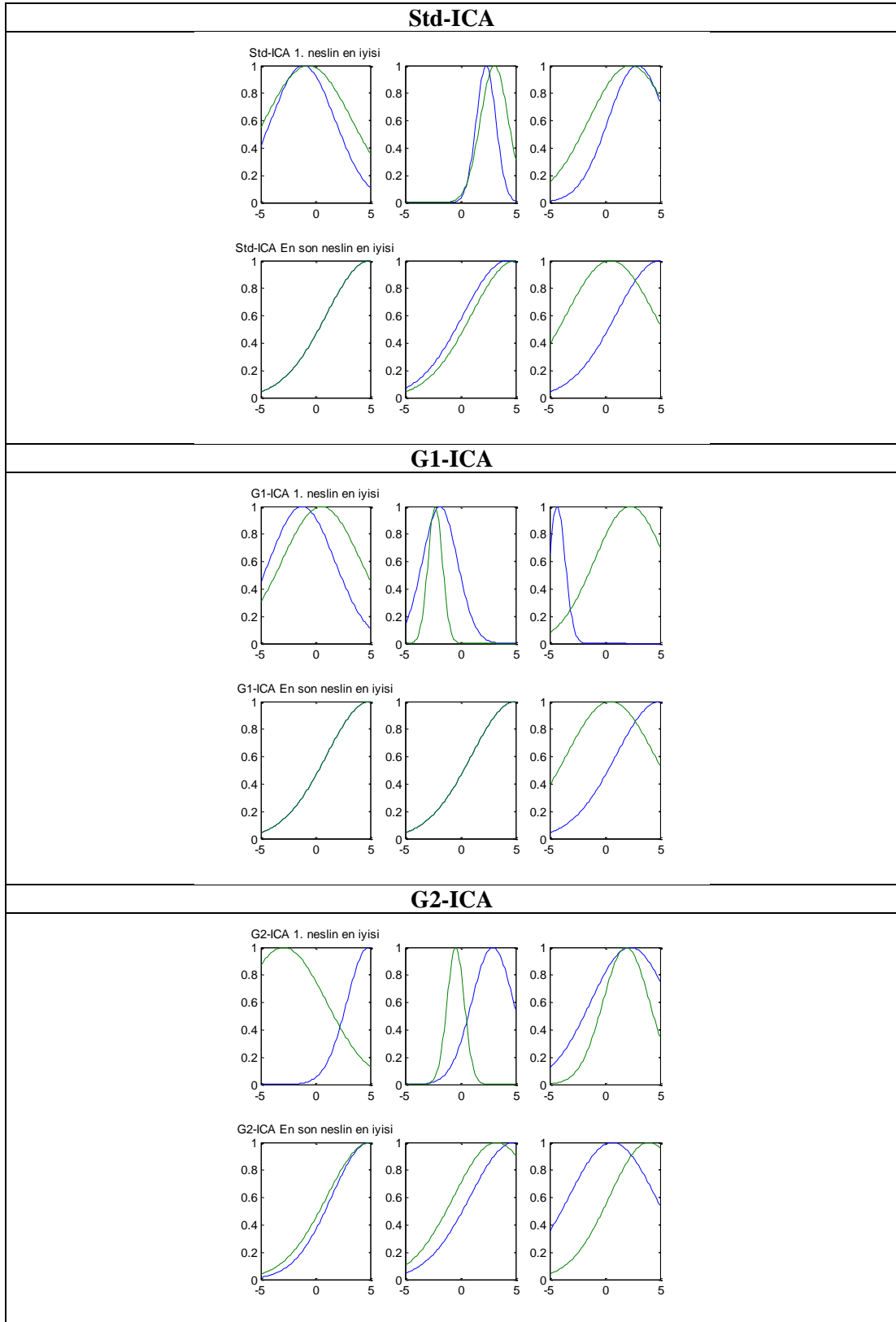
4.4.1.ÖDS 4 için eğitim aşaması ve sonuçları

Bu bölümde Bölüm 3 Şekil 3.1 (c) de verilen giriş dizilerinden oluşturulan eğitim seti dizileri ÖDS 4 için çalıştırılarak başarımları incelenmiştir. Buna göre aşağıda Şekil 4.16' da ICA öğrenmeli ANFIS ile ÖDS 4 eğitim fazı için elde edilen sonuçlar grafiksel olarak verilmiştir. Grafikler, elde edilen en iyi çözüm ve en kötü çözüm olmak üzere ikiye ayrılarak düzenlenmiştir. Grafikler incelendiğinde Std-ICA, G1-ICA ve G2-ICA algoritmalarından en başarılı sonucun Std-ICA algoritmasının verdiği görülmüştür.

Şekil 4.17' de ise Std-ICA, G1-ICA ve G2-ICA' nın ÖDS 4 için bulanık modelleme eğitim fazındaki üyelik fonksiyonlarının (UF) ilk durumları ve son durumları grafiksel olarak hazırlanmıştır. Bu sayede her algoritmanın üyelik fonksiyonlarında değişim gözlenmek istenmiştir. Buna göre ilk grafikler; 1. Giriş' e ait, ortadaki grafikler; 2.Giriş' e ve son grafikler de 3. Giriş' e ait üyelik fonksiyonlarının ilk ve son durumlarını göstermektedir. Mavi renkli çizgi birinci üyelik fonksiyonunu, yeşil renkli çizgi ikinci üyelik fonksiyonunu ifade etmektedir.

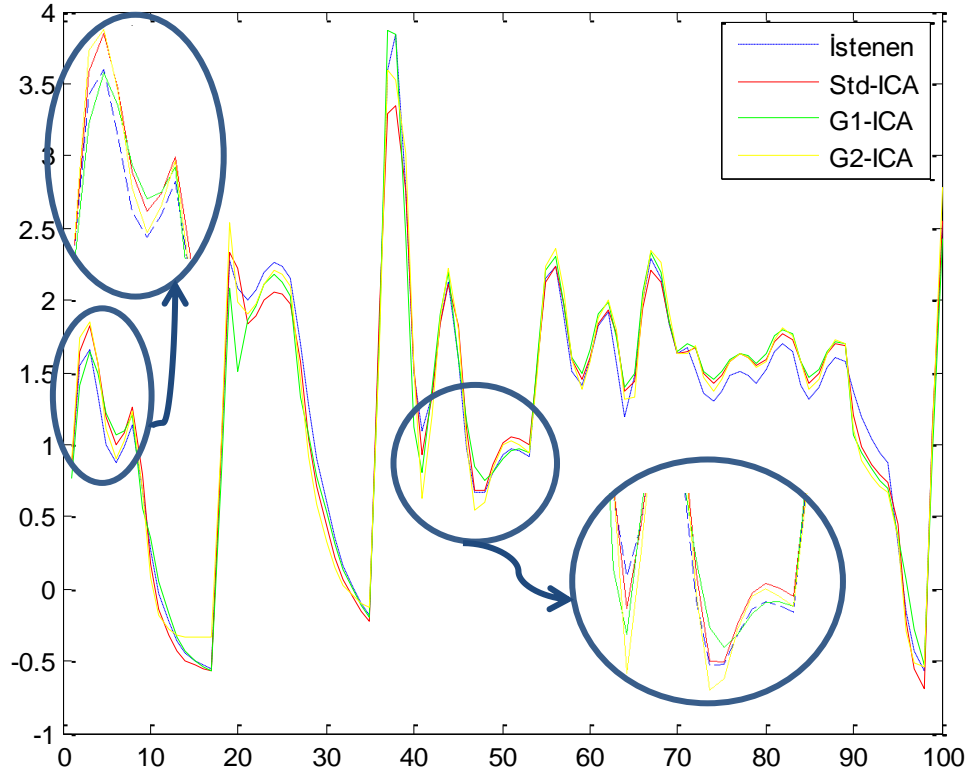


Şekil 4.16. ICA öğrenmeli ANFIS ile ÖDS 4 için eğitim fazı sistem tanıma sonuçları.



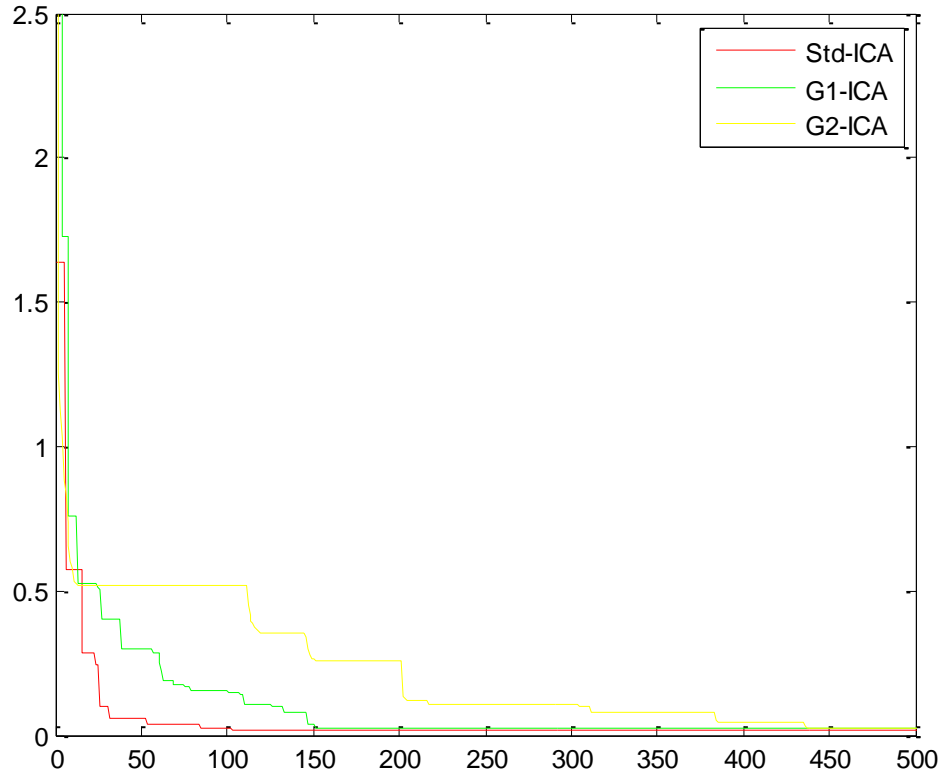
Şekil 4.17. ICA öğrenmeli ANFIS ile ÖDS 4'in bulanık modelleme eğitim fazında UF'lerin başlangıç ve son durumları.

Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ÖDS 4 modellemesinin başarımlarını kıyaslamasını kolaylaştırmak amacıyla, bu algoritmaların ayrı ayrı koşturulması sonucunda elde edilen en iyi ölçüt değerlerine sahip modelleme sonuçlarının karşılaştırılmasının yapıldığı grafik Şekil 4.18’ te verilmiştir. Grafik üzerinde kırılmanın olduğu kısım büyütülerek daha net bir gözlem yapılması sağlanmıştır.



Şekil 4.18. Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 4 için elde edilen modellerin karşılaştırılması

Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 4 modelleme eğitim seyrinde yakınsama karşılaştırılmasının yapıldığı grafik Şekil 4.19’ da verilmiştir. Buna göre elde edilen en iyi ölçüt değerlerinin her bir adımdaki değişimi gözlenmiştir. Bu konuda Std-ICA’ nın başarısı rahatlıkla görülmektedir.

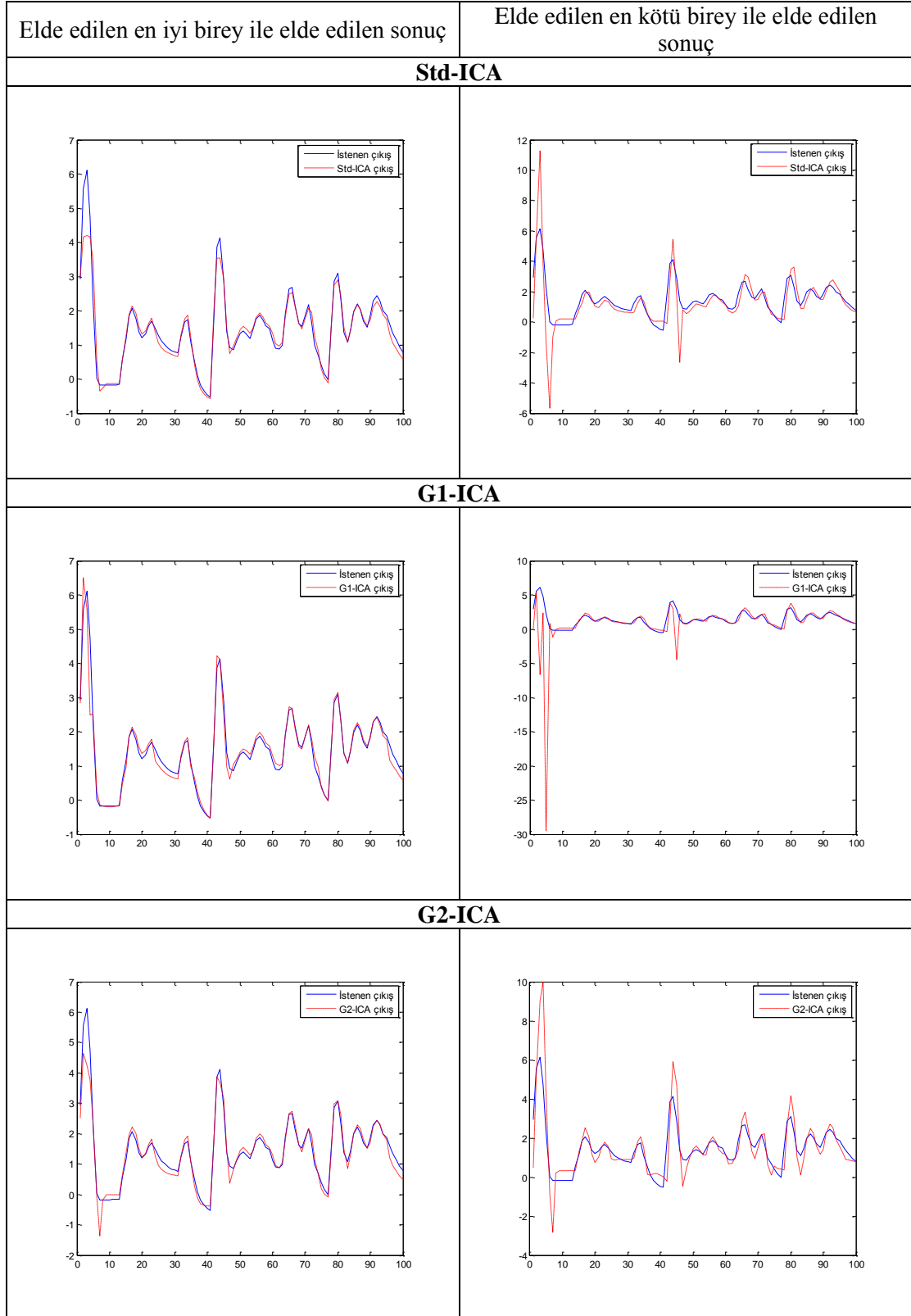


Şekil 4.19. Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 4 modelleme eğitim seyri

4.4.2.ÖDS 4'in test aşaması ve sonuçları

ÖDS 4 için Bölüm 3 Şekil 3.1 (c) de verilen $u(k)$ ile eğitim seti hazırlanmıştır. Test seti için ise Bölüm 3 Şekil 3.2 (c) de verilen $u(k)$ ile yine Bölüm 3 de yer alan Çizelge 3.1 de belirlenen girişler göz önüne alınarak test veri seti hazırlanmıştır.

Meydana getirilen test seti, Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 4 modelleme sonuçları Şekil 4.20' de grafiksel olarak verilmiştir. Grafikler incelendiğinde ICA öğrenmeli ANFIS ile ÖDS 4 için test sistem tanıma çalışmasında en başarılı sonucu Std-ICA algoritmasının verdiği görülmektedir.



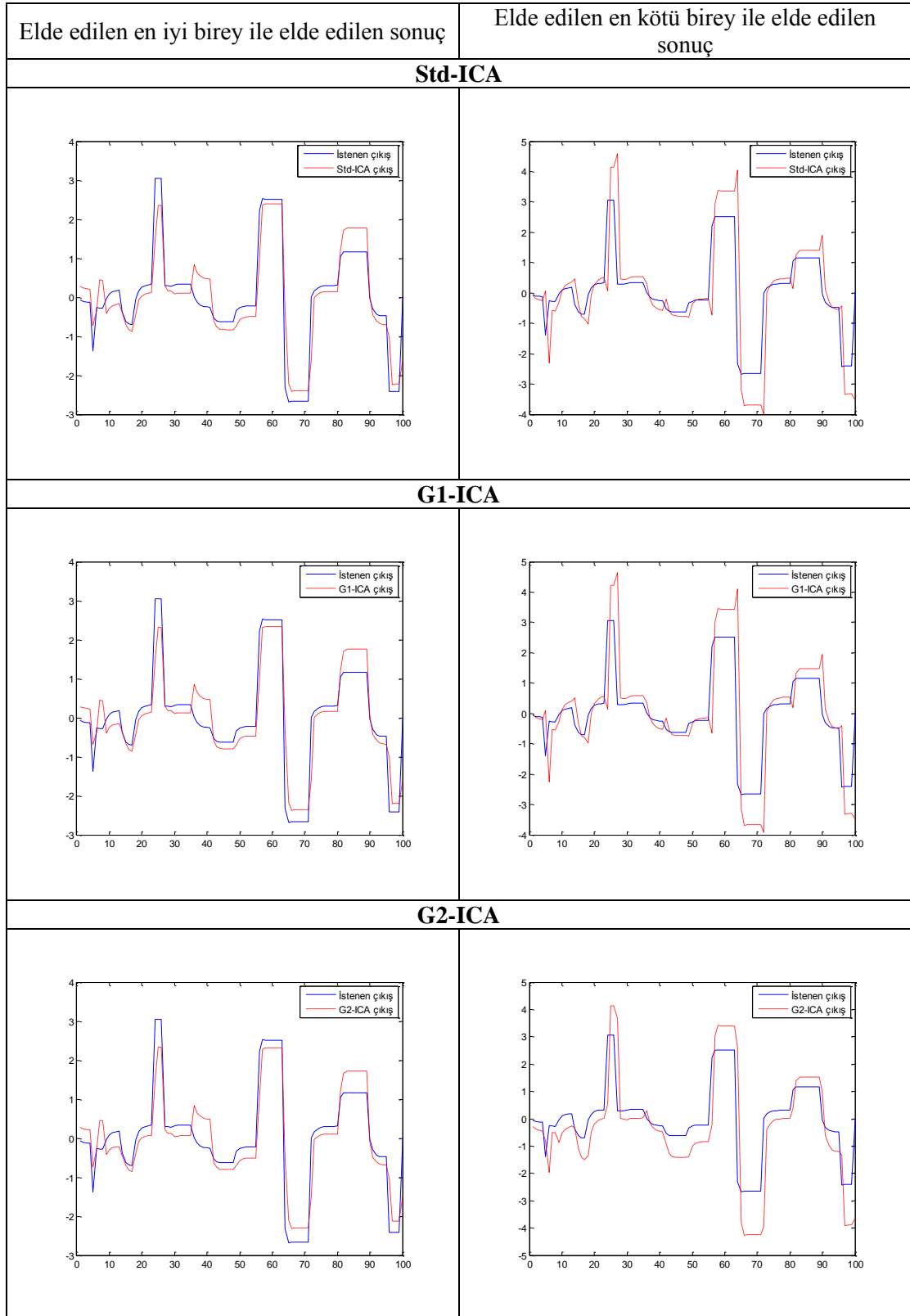
Şekil 4.20. ICA öğrenmeli ANFIS ile ÖDS 4 için test fazı sistem tanıma sonuçları

4.5. ÖDS 5 'in Tanınması/Modellenmesi

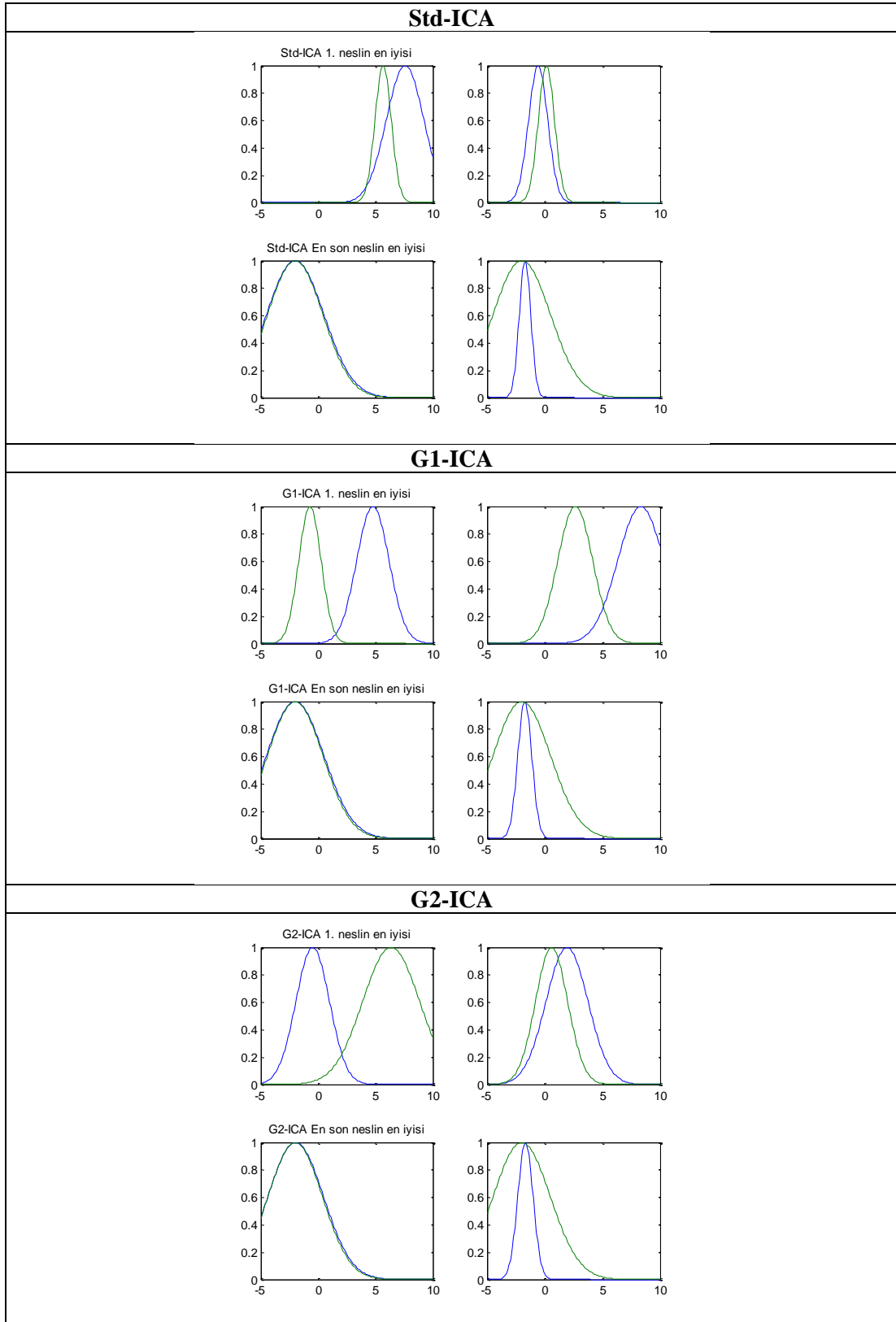
4.5.1.ÖDS 5 için eğitim aşaması ve sonuçları

Bu bölümde Bölüm 3 Şekil 3.1 (d) de verilen giriş dizilerinden oluşturulan eğitim seti dizileri ÖDS 5 için çalıştırılarak başarımları sonuçları incelenmiştir. Buna göre aşağıda Şekil 4.21' de ICA öğrenmeli ANFIS ile ÖDS 5 eğitim fazı için elde edilen sonuçlar grafiksel olarak verilmiştir. Grafikler, elde edilen en iyi çözüm ve en kötü çözüm olmak üzere ikiye ayrılarak düzenlenmiştir. Grafikler incelendiğinde Std-ICA, G1-ICA ve G2-ICA algoritmalarının sonuçları arasında çok bariz farklılıklar olmadığı, fakat en iyi sonucu G2-ICA algoritmasının verdiği görülmüştür.

Şekil 4.22' de ise Std-ICA, G1-ICA ve G2-ICA' nın ÖDS 5 için bulanık modelleme eğitim fazındaki üyelik fonksiyonlarının (UF) ilk durumları ve son durumları grafiksel olarak hazırlanmıştır. Bu sayede her algoritmanın üyelik fonksiyonlarında değişim gözlenmek istenmiştir. Buna göre ilk grafikler; 1. Giriş' e ait, son grafikler ise; 2. Giriş' e ait üyelik fonksiyonlarının ilk ve son durumlarını göstermektedir. Mavi renkli çizgi birinci üyelik fonksiyonunu, yeşil renkli çizgi ikinci üyelik fonksiyonunu ifade etmektedir.

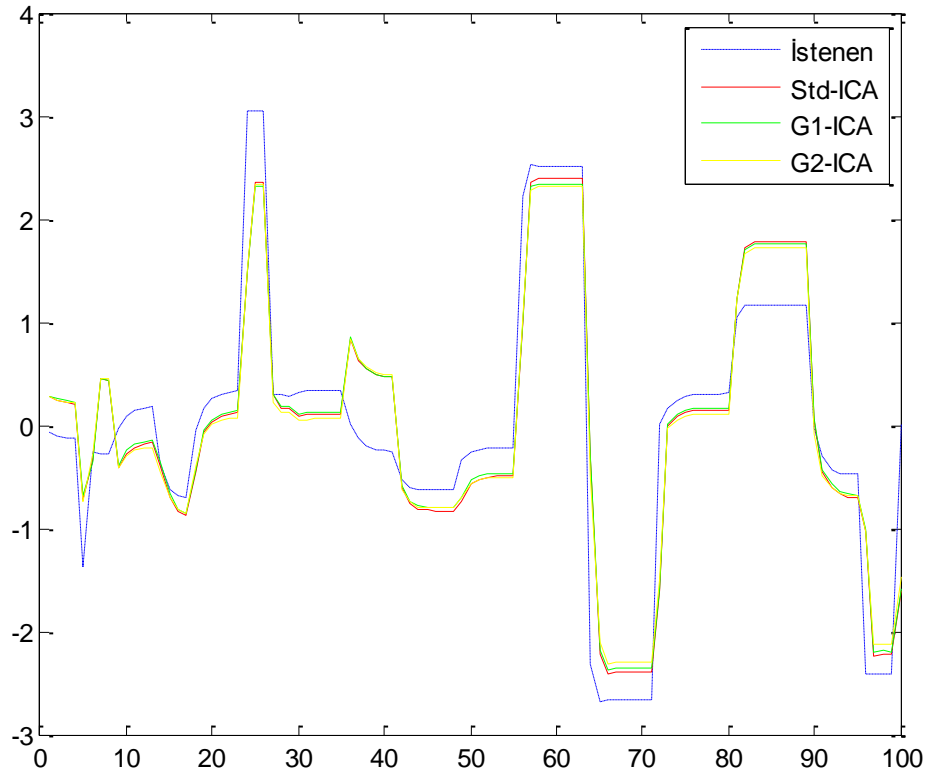


Şekil 4.21. ICA öğrenmeli ANFIS ile ÖDS 5 için eğitim fazı sistem tanıma sonuçları.



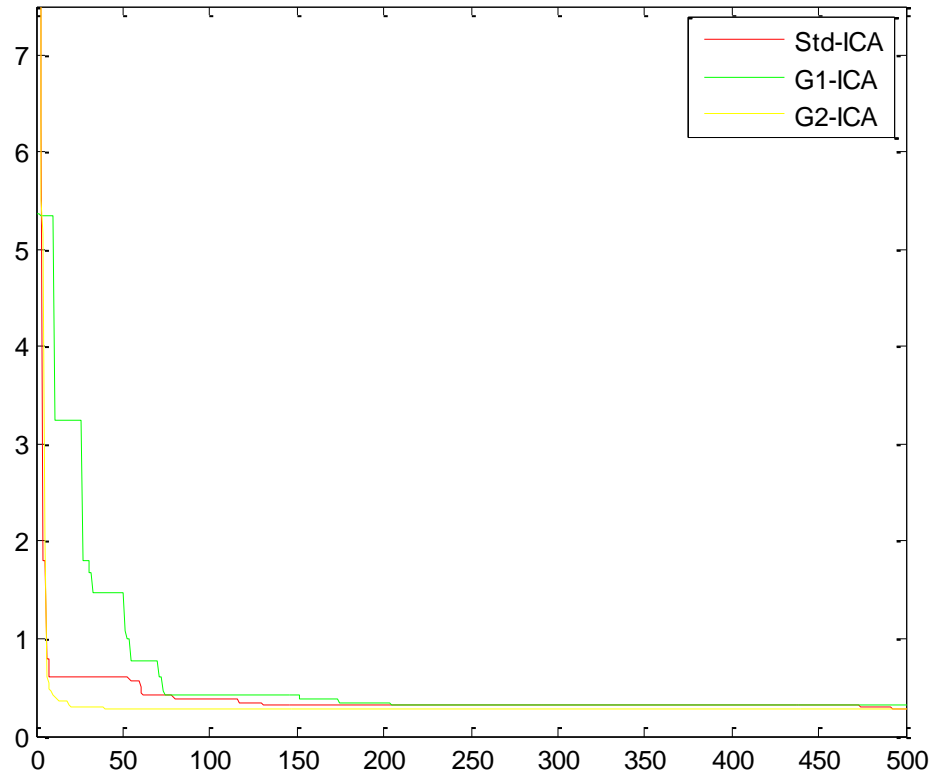
Şekil 4.22. ICA öğrenmeli ANFIS ile ÖDS 5'in bulanık modelleme eğitim fazında UF'lerin başlangıç ve son durumları.

Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ÖDS 5 modellemesinin başarımlarını kıyaslamasını kolaylaştırmak amacıyla, bu algoritmaların ayrı ayrı koşturulması sonucunda elde edilen en iyi ölçüt değerlerine sahip modelleme sonuçlarının karşılaştırılmasının yapıldığı grafik Şekil 4.23’ te verilmiştir. Grafik üzerinde kırılmanın olduğu kısım büyütülerek daha net bir gözlem yapılması sağlanmıştır.



Şekil 4.23. Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 5 için elde edilen modellerin karşılaştırılması

Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 5 modelleme eğitim seyrinde yakınsama karşılaştırılmasının yapıldığı grafik Şekil 4.24’ de verilmiştir. Buna göre elde edilen en iyi ölçüt değerlerinin her bir adımdaki değişimi gözlenmiştir. Bu konuda G2-ICA’ nın başarısı rahatlıkla görülmektedir.

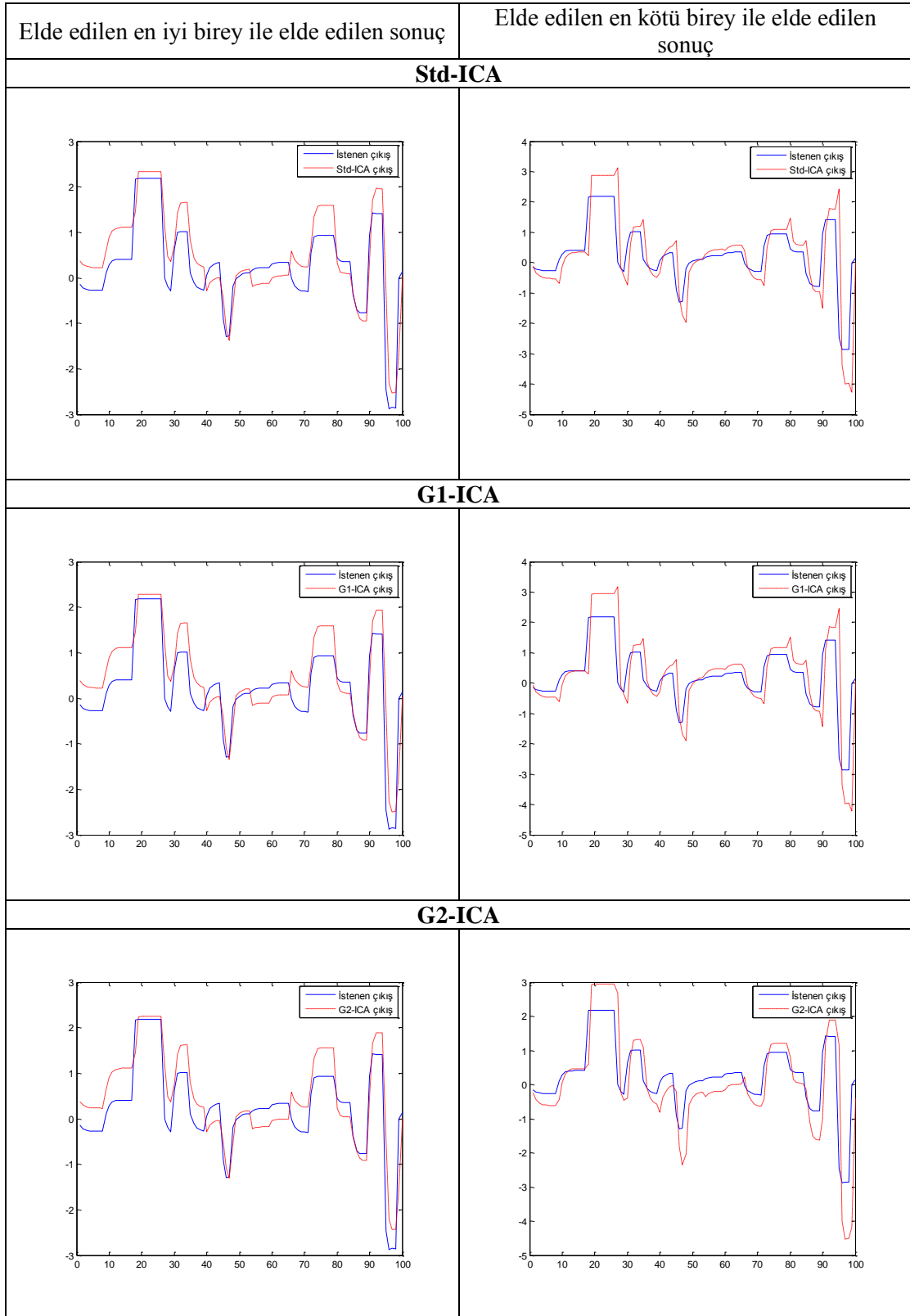


Şekil 4.24. Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 5 modelleme eğitim seyri

4.5.2.ÖDS 5'in test aşaması ve sonuçları

ÖDS 5 için Bölüm 3 Şekil 3.1 (d) de verilen $u(k)$ ile eğitim seti hazırlanmıştır. Test seti için ise Bölüm 3 Şekil 3.2 (d) de verilen $u(k)$ ile yine Bölüm 3 de yer alan Çizelge 3.1 de belirlenen girişler göz önüne alınarak test veri seti hazırlanmıştır.

Meydana getirilen test seti, Std-ICA, G1-ICA ve G2-ICA öğrenmeli ANFIS ile ÖDS 4 modelleme sonuçları Şekil 4.25' de grafiksel olarak verilmiştir. Grafikler incelendiğinde ICA öğrenmeli ANFIS ile ÖDS 5 için test sistem tanıma çalışmasında en başarılı sonucu Std-ICA algoritmasının verdiği görülmüştür.



Şekil 4.25. ICA öğrenmeli ANFIS ile ÖDS 5 için test fazı sistem tanıma sonuçları

4.6. Std-ICA, G1-ICA ve G2-ICA Algoritmalarının Performansının Kıyaslanması

ICA algoritmasının ANFIS bulanık mantık çıkarım modeli kullanılarak doğrusal olmayan dinamik sistemler üzerinde bulanık modelleme optimizasyon başarımını değerlendirmek amacıyla Bölüm 3' te Çizelge 3.1' de gösterilen 5 farklı sistem üzerinde çalışılmıştır. Gösterilen her bir sistem için Std-ICA ve ICA algoritması üzerinden geliştirilen G1-ICA ve G2-ICA algoritmaları, hem eğitim fazı hem de test fazı Bölüm 3' te Çizelge 3.3' de verilen parametrelere uygun olarak 50' şer kez koşturulmuştur. Her bir algoritmadan elde edilen sonuçlar kendi aralarında kıyaslanmış, eğitim seti için başarımların kıyaslaması Çizelge 4.1' de, test seti için başarımların kıyaslaması Çizelge 4.2' de verilmiştir.

Çizelge 4.1' de her bir sistem için kıyaslama 5 farklı kategoride istatistiki olarak incelenmiştir. Bu kategoriler sırasıyla ortalama, standart sapma, en iyi, en kötü, 1 nesil süresi gibi başlıklardan oluşmaktadır. Bu kategorileri aşağıdaki gibi tanımlayabiliriz:

- Ortalama: 50 ayrı koşma sonunda oluşan eğitim ölçüt değerlerinin ortalaması.
- Standart sapma: 50 ayrı koşma sonunda oluşan eğitim ölçüt değerlerinin standart sapması.
- En iyi: 50 ayrı koşma sonucu oluşan ölçütler içinde en iyi ölçüt değeri.
- En kötü: 50 ayrı koşma sonucu oluşan ölçütler içinde en kötü ölçüt değeri.
- 1 nesil süresi: 50 ayrı koşma sonunda oluşan 1 neslin saniye cinsinden süresi.

Çizelge 4.1. Std-ICA, G1-ICA ve G2-ICA yaklaşımlarının ÖDS modellemede *EĞİTİM* fazı başarımlarını kıyaslaması

Algoritma	Ölçüt	ÖDS 1 (Maksimum nesil=1000)	ÖDS 2 (Maksimum nesil=1000)	ÖDS 3 (Maksimum nesil=1000)	ÖDS 4 (Maksimum nesil=1000)	ÖDS 5 (Maksimum nesil=1000)
Std-ICA	Ortalama	0,048425	0,015059	0,009329	0,078148	0,395600
	Std. Sapma	0,031149	0,006579	0,001462	0,049579	0,252983
	En İyi	0,014787	0,007337	0,008617	0,016241	0,280369
	En Kötü	0,153291	0,051500	0,019170	0,206076	1,405025
	1 Nesil Süresi (s)	0,193708	0,078060	0,072226	0,133825	0,112316
	Başarım Sayısı	1	2	1	3	1
	Ortalama Başarım Sayısı	8/25=0,32				
G1-ICA	Ortalama	0,048007	0,016586	0,009117	0,083323	0,422034
	Std. Sapma	0,024518	0,008763	0,000376	0,040319	0,240954
	En İyi	0,019903	0,006910	0,008613	0,021339	0,279880
	En Kötü	0,162674	0,060344	0,010391	0,166496	1,407984
	1 Nesil Süresi (s)	0,173324	0,063935	0,108904	0,170359	0,108185
	Başarım Sayısı	2	1	3	2	2
	Ortalama Başarım Sayısı	10/25=0,4				
G2-ICA	Ortalama	0,049896	0,017999	0,009405	0,090580	0,397784
	Std. Sapma	0,021524	0,010181	0,001259	0,047868	0,270110
	En İyi	0,015125	0,006566	0,008410	0,021915	0,272056
	En Kötü	0,128072	0,052427	0,014584	0,224736	1,322431
	1 Nesil Süresi (s)	0,280625	0,165612	0,178266	0,266595	0,201637
	Başarım Sayısı	2	2	1	0	2
	Ortalama Başarım Sayısı	7/25=0,28				

Bu kıyaslamaya göre bu üç algoritma arasında elde edilen sonuçların birbirine çok yakın olması ile birlikte en yüksek başarımları G1-ICA algoritmasının verdiği görülmüştür.

Aynı zamanda ICA algoritmalarının ANFIS yapısı üzerinde başarımlarını daha iyi görebilmek amacıyla test seti için farklı veri setleri kullanılarak her bir sistem için test edilmiştir. Bu test seti ile üç ICA algoritması için elde edilen sonuçlar Çizelge 4.2' de

verilmiştir. Ulaşılan sonuçlara göre Std-ICA ve G2-ICA algoritmaları aynı başarıyı göstererek G1-ICA algoritmadan daha başarılı olarak ön plana çıkmışlardır.

Çizelge 4.2. Std-ICA, G1-ICA ve G2-ICA yaklaşımlarının ÖDS modellemede *TEST* fazı başarımlarını kıyaslaması

Algoritma	Ölçüt	ÖDS 1 (Maksimum nesil=1000)	ÖDS 2 (Maksimum nesil=1000)	ÖDS 3 (Maksimum nesil=1000)	ÖDS 4 (Maksimum nesil=1000)	ÖDS 5 (Maksimum nesil=1000)
Std-ICA	En İyi	0,035793	0,010392	0,009683	0,063372	0,199689
	En Kötü	0,707772	0,435965	0,022296	1,961994	0,857823
	Başarım Sayısı	0	1	1	2	1
	Ortalama Başarım Sayısı	5/10=0,5				
G1-ICA	En İyi	0,033026	0,015273	0,009733	0,069660	0,212914
	En Kötü	1,045441	0,563748	0,026497	12,55323	0,862122
	Başarım Sayısı	0	0	0	0	0
	Ortalama Başarım Sayısı	0/10=0				
G2-ICA	En İyi	0,022349	0,019990	0,009054	0,073042	0,232500
	En Kötü	0,423047	0,351634	0,045183	3,083744	0,849116
	Başarım Sayısı	2	1	1	0	1
	Ortalama Başarım Sayısı	5/10=0,5				

4.7. Literatüre Dayalı Başarım Kıyaslaması

Yukarıda sonuçları anlatılan ÖDS' ler için literatürde ANFIS kullanılarak ICA algoritmalarının başarımını araştıran bir çalışma bulunmamaktadır. Buna rağmen, (Yıldırım, 2012)' da yapılmış çalışmada yine aynı ÖDS'ler için ANFIS yapısı kullanılan bir modelleme ile ABC, DE ve PSO algoritmalarının başarımını incelemiştir.

Çalışmamızın bu kısmında, Bölüm 3'deki ANFIS ağ yapısı için dinamik sistemlere ait gerekli parametre optimizasyon araçları oluşturularak, sistemlerin eğitilmesinde ve test edilmesinde; ICA algoritması için popülasyon büyüklüğü, parametre, maksimum nesil sayısı ve koşma sayısı (Yıldırım, 2012) çalışmasına uygun alınarak elde edilen sonuçlar, o çalışmada ANFIS bulanık mantık çıkarım modeli kullanılarak ABC, DE ve PSO algoritmaları ile elde etmiş olduğu sonuçlara, eğitim seti

ve test seti için ayrı ayrı olmak üzere eklenerek, birleştirilmiş karşılaştırma çizelgeleri hazırlanmıştır. Çizelge 4.3' te eğitim seti için, Çizelge 4.4' de ise test seti için başarımların kıyaslamalarını görmek mümkündür.

Buna göre başarımların kıyaslamaları incelendiğinde, ICA algoritmaları istenilen değerlere yakın sonuçlar çıkarsa da eğitim seti için, ABC algoritmasının üstünlüğü görülmektedir. Test seti incelendiğinde ise başarımları en yüksek algoritmanın ICA algoritmasının olduğu görülmüştür.

Çizelge 4.3. Eğitim seti için literatüre dayalı başarımların kıyaslaması

Algoritma	Ölçü	ÖDS 1 (Maksimum nesil =1000)	ÖDS 2 (Maksimum nesil =1000)	ÖDS 3 (Maksimum nesil =1000)	ÖDS 4 (Maksimum nesil =1000)	ÖDS 5 (Maksimum nesil =1000)
ABC	Ortalama	0,036994	0,045392	0,012165	0,050564	0,195731
	Std sapma	0,012073	0,032781	0,013409	0,019603	0,038999
	En iyi	0,013664	0,003924	0,004422	0,01568	0,171836
	En kötü	0,058886	0,145957	0,056209	0,092783	0,453153
	1 nesil süresi (s)	0,026451	0,02701	0,015753	0,024448	0,064086
	Başarımlar sayısı	3	2	1	4	4
	Ortalama Başarımlar Sayısı	14/25=0.56				
PSO	Ortalama	0,079258	0,030886	0,212148	0,120247	9,361079
	Std. sapma	0,063467	0,027827	0,568219	0,043797	7,557416
	En iyi	0,027028	0,008052	0,005427	0,03366	0,734169
	En kötü	0,437995	0,197535	3,759978	0,220542	28,09871
	1 nesil süresi (s)	0,047024	0,045804	0,02863	0,047661	0,083383
	Başarımlar sayısı	0	0	0	0	0
	Ortalama Başarımlar Sayısı	0/25=0				
DE	Ortalama	0,039115	0,018553	0,00405	0,123121	0,281597
	Std sapma	0,004217	0,098136	0,001221	0,01032	0,139422
	En iyi	0,033775	0,010484	0,002383	0,112719	0,020042
	En kötü	0,054579	0,131649	0,00828	0,156764	0,551824
	1 nesil süresi (s)	0,040384	0,041445	0,034371	0,040374	0,32614
	Başarımlar sayısı	2	0	4	1	1
	Ortalama Başarımlar Sayısı	8/25=0.32				
Std-ICA	Ortalama	0,048425	0,015059	0,009329	0,078148	0,3956
	Std. Sapma	0,031149	0,006579	0,001462	0,049579	0,252983
	En İyi	0,014787	0,007337	0,008617	0,016241	0,280369
	En Kötü	0,153291	0,0515	0,01917	0,206076	1,405025
	1 Nesil Süresi(s)	0,193708	0,07806	0,072226	0,133825	0,112316
	Başarımlar Sayısı	0	3	0	0	0
	Ortalama Başarımlar Sayısı	3/25=0.12				

Çizelge 4.4. Test seti için literatüre dayalı başarımların kıyaslaması

Algoritma	Ölçüt	ÖDS 1 (Maksimum nesil =1000)	ÖDS 2 (Maksimum nesil =1000)	ÖDS 3 (Maksimum nesil =1000)	ÖDS 4 (Maksimum nesil =1000)	ÖDS 5 (Maksimum nesil =1000)
ABC	En iyi	0,027788	0,020664	0,025316	0,094835	0,124672
	En kötü	1,97251	0,801725	547,1167	5,72806	1315,189
	Başarım Sayısı	1	0	0	0	0
	Ortalama Başarım Sayısı	1/10=0.1				
PSO	En iyi	0,054068	0,025113	0,023815	0,083096	0,39773
	En kötü	1,278393	1,094454	51,89824	0,732723	28,57756
	Başarım Sayısı	0	0	0	1	0
	Ortalama Başarım Sayısı	1/10=0.1				
DE	En iyi	0,047424	0,024831	0,031532	0,204259	0,025181
	En kötü	0,195956	0,675152	2,356835	0,919691	0,640239
	Başarım Sayısı	1	0	0	0	2
	Ortalama Başarım Sayısı	3/10=0.3				
ICA	En İyi	0,035793	0,010392	0,009683	0,063372	0,199689
	En Kötü	0,707772	0,435965	0,022296	1,961994	0,857823
	Başarım Sayısı	0	2	2	1	0
	Ortalama Başarım Sayısı	5/10=0.5				

5. SONUÇLAR

Bu tez çalışmasında, ICA algoritmasının işleyişini kavrayarak fonksiyon optimizasyonu üzerine yapılan gözlemlere dayalı olarak algoritmanın geliştirilmesi; standart ICA ve geliştirilen algoritmaların bulanık sistem optimizasyonunda başarımlarının detaylı bir biçimde incelenmesi amaçlanmıştır.

Bu maksatla ilk aşamada, ICA algoritmasının (Std-ICA) çalışma mantığını anlamak için *tepeler (peaks)* fonksiyonu üzerinde çalışması incelenmiş, fonksiyon optimizasyonundaki başarımını artırmak amacıyla da iki farklı yaklaşımla iki yeni geliştirilmiş ICA algoritması (G1-ICA, G2-ICA) tanımlanmıştır. Bu üç algoritma *tepeler (peaks)* fonksiyonu haricinde üç farklı fonksiyon üzerinde de işletilerek yerel minimuma takılıp takılmama açısından başarımlar değerlendirilmiştir. Çizelge 5.1’ den görüldüğü üzere G2-ICA ismini verdiğimiz; imparatorluklardaki emperyalistlerin çevresinde daha iyi ölçüt değerine sahip olabileceği konuma hareket ettiren algoritmanın daha başarılı sonuçlar verdiği görülmüştür.

Çizelge 5.1. Algoritmaların yerel minimuma takılma değerlendirme çizelgesi

Kullanılan Algoritmalar	Koşturma Sayısı	F1 Fonksiyonu	F2 Fonksiyonu	F3 Fonksiyonu
		Yerel Minimuma Takılma Sayısı		
Std-ICA	100	0	18	8
G1-ICA	100	0	18	3
G2-ICA	100	0	15	2

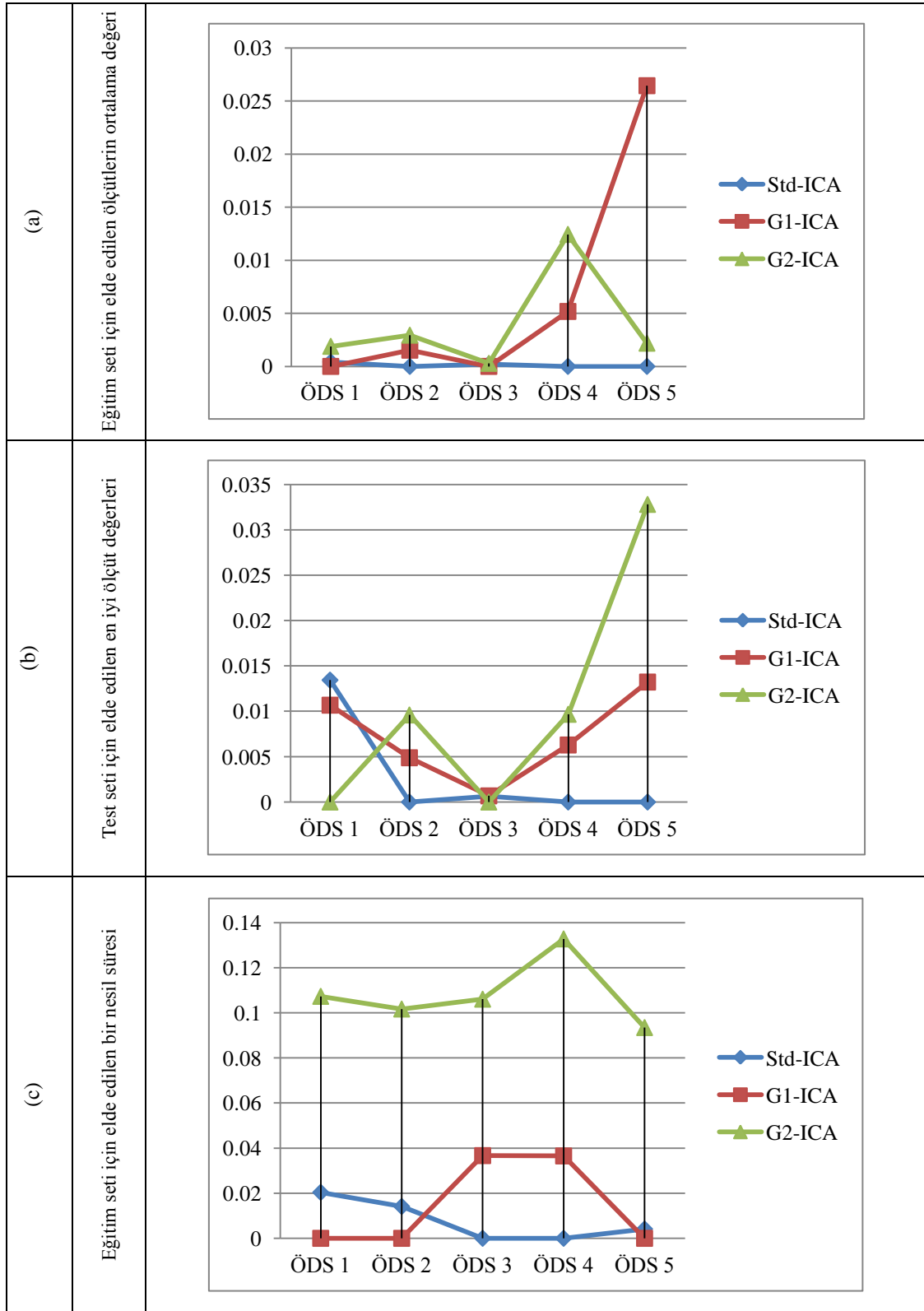
İkinci aşamada, bulanık işleyişi gerçekleyen ANFIS bulanık mantık çıkarım modeli kullanılarak; ICA’ nın doğrusal olmayan dinamik sistemlerin bulanık modellenmesinde optimizasyon başarımı incelenmiş, önce yukarıda anılan ICA algoritmalarının kendi aralarında, sonra da ANFIS modeli kullanılarak aynı doğrusal olmayan dinamik sistemlerin aynı bulanık sistemle modellendiği popüler algoritmalarla başarımları değerlendirilmiştir. Doğrusal olmayan dinamik sistemlerin bulanık mantık tabanlı modellenmesinde ANFIS optimizasyonu başarımların kıyaslaması sonucu; Std-ICA,

G1-ICA ve G2-ICA algoritmaları arasında sonuçların birbirine çok yakın olduğu, başarımı en yüksek olan algoritmanın aşağıda verilen grafikler ışığında G1-ICA olduğu sonucuna varılmıştır. Şekil 5.1' de her bir ÖDS için en iyi sonucu veren algoritmanın verdiği ölçüt değerine göre normalize edilmiş diğer algoritmaların verdiği ölçüt değerleri gösterilmektedir. Bu verilen grafiksel sonuçlara göre yatay eksene en yakın seyre sahip algoritma en iyi sonuçları vermiş olarak değerlendirilmelidir. Bu grafiksel değerlendirme kriterine göre Std-ICA diğer iki algoritmaya göre üstün başarımlar sergilemiştir.

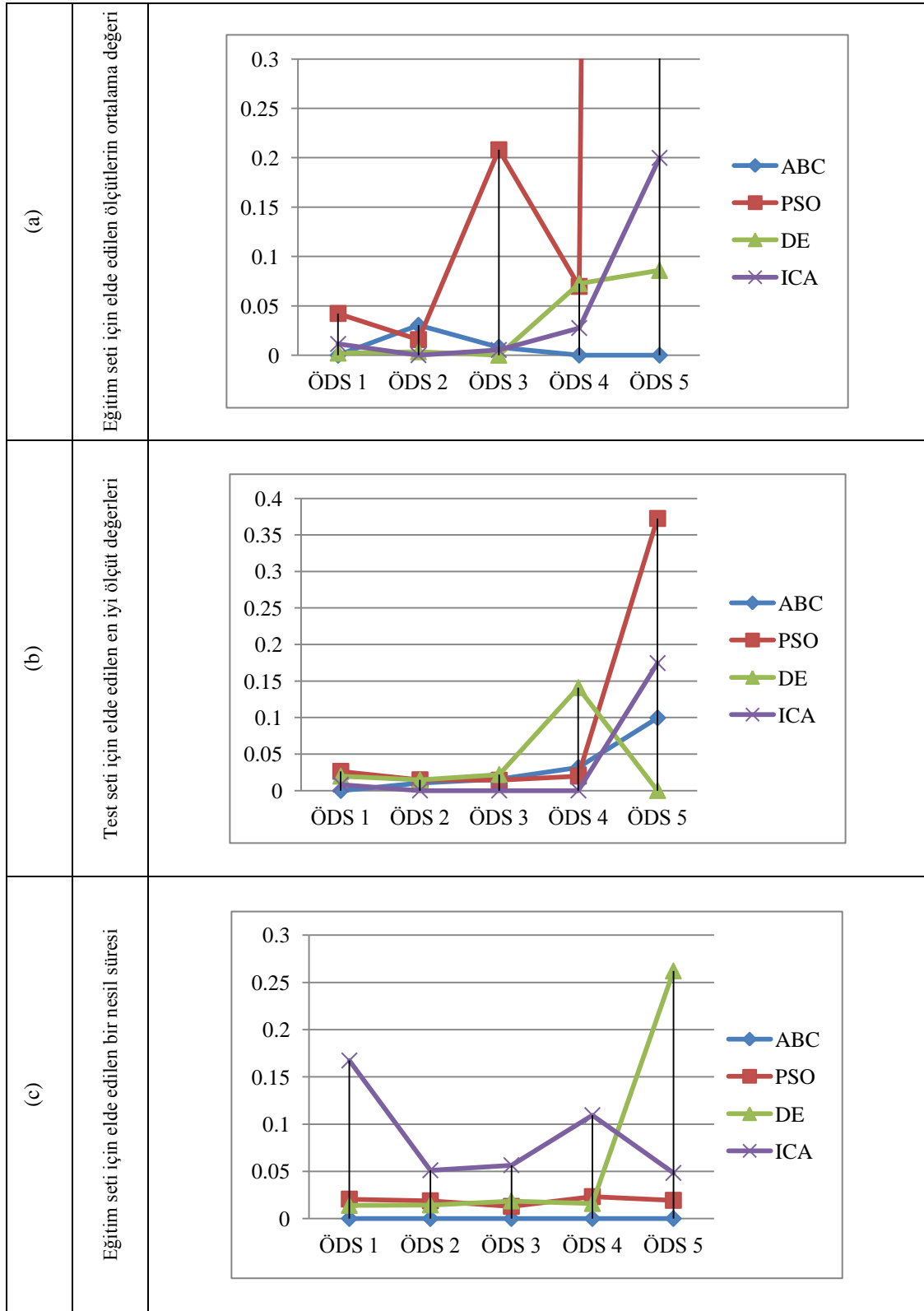
Benzer yaklaşımla grafiksel olarak ICA algoritmasının ABC, PSO ve DE gibi iyi bilinen ve yaygın olarak kullanılan algoritmalarla kıyaslanması hem eğitim seti hem de test seti için yapılmıştır. Şekil 5.2' deki sonuçlara dayalı olarak, eğitim seti için başarımın en yüksek olduğu algoritmanın ABC algoritması olduğu, test seti için ise ICA algoritmasının olduğu görülmüştür.

ICA algoritmalarının bu kıyaslamadaki en zayıf yönü; Şekil 5.2 (c) ile gösterilmeye çalışıldığı gibi 1 nesil süresinin diğer algoritmalara göre çok gerilerde kalıyor olmasıdır. İleriki çalışmalarda, ICA algoritmasının bu sakıncası giderilirse, doğrusal olmayan dinamik sistemlerin ANFIS ile bulanık modellenmesindeki başarımının daha da artırılmasının mümkün olabileceği değerlendirilmektedir.

Bölüm 1' de verilen literatür özetinden de anlaşılacağı üzere, dinamik sistemlerin ANFIS ile modellenmesinde ICA algoritmasının kullanılması ile ilgili benzer bir kullanımına rastlanmamıştır. Bu bakımdan çalışmanın ilgili alanda katkı sunduğu düşünülmektedir. Bu yapılan çalışma sonucunda, çok yeni ve güncel bir algoritma olan ICA'nın bulanık sistem parametrelerinin optimizasyonunda etkin bir şekilde kullanılabileceği gösterilmiştir.



Şekil 5.1. Dinamik sistemlerin ANFIS ile modellenmesinde ICA algoritmalarının birbirleriyle grafiksel olarak ortalama ölçüt (a), en iyi ölçüt (b) ve bir nesil süresi (c) değerleri karşılaştırması



Şekil 5.2. Dinamik sistemlerin ANFIS ile modellenmesinde popüler algoritmaların ICA algoritması ile grafiksel olarak ortalama ölçüt (a), en iyi ölçüt (b) ve bir nesil süresi (c) değerleri karşılaştırması

6. KAYNAKLAR

- Abdechiri, M., Faez, K., Bahrami, H., “Adaptive Imperialist Competitive Algorithm”, *9th IEEE International Conference on Cognitive Informatics (ICCI)*, 940-945 (2010).
- Abdollahi, M., Isazadeh, A., Abdollahi, D., “Imperialist competitive algorithm for solving systems of nonlinear equations”, *Computers and Mathematics with Applications*, 65: 1894–1908 (2013).
- Afonso, L.D., Mariani, V.C., Coelho, L.D.S., “Modified imperialist competitive algorithm based on attraction and repulsion concepts for reliability-redundancy optimization”, *Expert Systems with Applications*, 40: 3794–3802 (2013).
- Ahmadia, M.A., Ebadib, M., Shokrollahic, A., Majidic, S.M.J., “Evolving artificial neural network and imperialist competitive algorithm for prediction oil flow rate of the reservoir”, *Applied Soft Computing*, 13: 1085–1098 (2013).
- Akyol, S., Alataş, B., “Güncel sürü zekâsı optimizasyon algoritmaları”, *Nevşehir Üniversitesi Fen Bilimleri Enstitü Dergisi*, 1: 36-50 (2012).
- Alataş, B., “Kaotik Haritalı Parçacık Sürü Optimizasyon Algoritmaları Geliştirme”, Doktora Tezi, *Fırat Üniversitesi Fen Bilimleri Enstitüsü*, Elazığ, 2007.
- Amiri-Arefa, M., Javadiana, N., Tavakkoli-Moghaddamb, R., Babolic, A., Shiripoura, S., “The center location-dependent relocation problem with a probabilistic line barrier”, *Applied Soft Computing*, 13: 3380–3391 (2013).
- Atashpaz-Gargari, E., Lucas, C., “Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition”, *IEEE Congress on Evolutionary Computation*, 4661-4667 (2007).
- Atashpaz-Gargari, E., “Imperialist Competitive Algorithm (ICA)-MATLAB Kaynak Kodları”, <http://www.mathworks.com/matlabcentral/fileexchange/22046-imperialist-competitive-algorithm--ica->, (son erişim Mayıs 2014).
- Azami, H., Sanei, S., Mohammadi, K., Hassanpour, H., “A hybrid evolutionary approach to segmentation of non-stationary Signals”, *Digital Signal Processing*, 23: 1103–1114 (2013).

KAYNAKLAR (Devam Ediyor)

- Babuska, R., “Fuzzy System, Modeling and Identification”, <http://www.dcsc.tudelft.nl/~babuska/transp/fuzzmod.pdf> (son erişim Şubat 2012).
- Borji, A., “A New Global Optimization Algorithm Inspired by Parliamentart Political Competitions”, *Lecture Notes in Computer Science*, 4827/2007, 61-71 (2007).
- Çakıroğlu, M.A., Erenoğlu, E., Kasap, S., Ekiz, Y., “Çelik lif katkılı betonların tahribatsız deney yöntemleriyle elde edilen basınç dayanımının ANFIS metoduyla tahmini”, *SDU International Technological Science*, 3(3): 14-22 (2011).
- Devika, K., Jafarian, A., Nourbakhsh, V., “Designing a sustainable closed-loop supply chain network based on triple bottom line approach: A comparison of metaheuristics hybridization techniques”, *European Journal of Operational Research*, 235: 594–615 (2014).
- Duan, H., Xu, C., Liu, S., Shao, S., “Template matching using chaotic imperialist competitive algorithm”, *Pattern Recognition Letters*, 31: 1868–1875 (2010).
- Duan, H., Huang, L., “Imperialist competitive algorithm optimized artificial neural networks forUCAV global path planning”, *Neurocomputing*, 125: 166–171 (2014).
- Ebrahimzadeh, A., Addeh, J., Rahmani, Z., “Control chart pattern recognition using K-MICA clustering and neural networks”, *ISA Transactions*, 51: 111–119 (2012).
- Elmas, Ç., “Bulanık mantık denetleyiciler”, *Seçkin Yayıncılık*, Ankara, (2003).
- Enayatifar, R., Abdullah, A.H., Lee, M., “A weighted discrete imperialist competitive algorithm (WDICA) combined with chaotic map for image encryption” , *Optics and Lasers in Engineering*, 51: 1066–1077 (2013).
- Ghasemi, M., Ghavidel, S., Rahmani, S., Roosta, A., Falah, H., “A novel hybrid algorithm of imperialist competitive algorithm and teaching learning algorithm for optimal power flow problem with non- smooth cost functions”, *Engineering Applications of Artificial Intelligence*, 29: 54–69 (2014).

KAYNAKLAR (Devam Ediyor)

- Glover, F., “Tabu search - part I”, *ORSA Journal on Computing*, 1(3): 190-206 (1989).
- Goldansaz, S.M., Jolai, F., Anaraki, A.H.Z., “A hybrid imperialist competitive algorithm for minimizing makespan in a multi-processor open shop”, *Applied Mathematical Modelling*, 37: 9603–9616 (2013).
- Huanga, L., Duana, H., Wang, Y., “Hybrid bio-inspired lateral inhibition and Imperialist Competitive Algorithm for complicated image matching”, *Optik*, 125: 414– 418 (2014).
- Jang, J.R.S., “ANFIS: adaptive-network-based fuzzy inference system”, *IEEE Transactions on Systems, Man and Cybernetics*, 23(3): 665-684 (1993).
- Karaboğa, D., “Yapay Zeka Optimizasyon Algoritmaları”, *Atlas Yayın Dağıtım*, İstanbul, 2004.
- Karamanoğlu, M., Bilgehan, M., Sivrikaya, Osman., “Yangına Maruz Betonarme Döşemelerin Moment Kapasitesinin Bulanık Sinir Ağı Yöntemi ile Tahmini”, *Bilimde Modern Yöntemler Sempozyumu-BMYS 2010*, Diyarbakır (2010).
- Karami, A., Rezaei, E., Shahhosseni, M., Aghakhani, M., “Optimization of heat transfer in an air cooler equipped with classic twisted tape inserts using imperialist competitive algorithm”, *Experimental Thermal and Fluid Science*, 38: 195–200 (2012).
- Kaveh, A., Talatahari, S., “Optimum design of skeletal structures using imperialist competitive algorithm ”, *Computers and Structures*, 88: 1220–1229 (2010).
- Kızılloluk, S., Alataş, B., “Sosyal tabanlı güncel sezgisel optimizasyon algoritmaları”, *C.Ü. İktisadi ve İdari Bilimler Dergisi*, 13(2): 39-55 (2012)
- Mahari, A., Zare, K., “A solution to the generation scheduling problem in power systems with large-scale wind farms using MICA” *Electrical Power and Energy Systems*, 54: 1–9 (2014).

KAYNAKLAR (Devam Ediyor)

- Mirhoseini, S.H., Hosseini, S.M., Ghanbari, M., Ahmadi, M., “A new improved adaptive imperialist competitive algorithm to solve the reconfiguration problem of distribution systems for loss reduction and voltage profile improvement”, *Electrical Power and Energy Systems*, 55: 128–143 (2014).
- Mohammadi, M., Jolai, F., Rostami, H., “An M/M/c queue model for hub covering location problem”, *Mathematical and Computer Modelling*, 54: 2623–2638 (2011).
- Mohammadi, M., Jolai, F., Tavakkoli-Moghaddam, R., “Solving a new stochastic multi-mode p-hub covering location problem considering risk by a novel multi-objective algorithm”, *Applied Mathematical Modelling*, 37: 10053–10073 (2013).
- Moradi, H., Zandieh, M., “An imperialist competitive algorithm for a mixed-model assembly line sequencing problem”, *Journal of Manufacturing Systems*, 32: 46–54 (2013).
- Moradi, M.H., Zeinalzadeh, A., Mohammadi, Y., Abedini, M., “An efficient hybrid method for solving the optimal sitting and sizing problem of DG and shunt capacitor banks simultaneously based on imperialist competitive algorithm and genetic algorithm”, *Electrical Power and Energy Systems*, 54: 101–111 (2014).
- Morshed, M.J., Asgharpour, A., “Hybrid imperialist competitive-sequential quadratic programming (HIC-SQP) algorithm for solving economic load dispatch with incorporating stochastic wind power: A comparative study on heuristic optimization techniques”, *Energy Conversion and Management*, 84: 30–40 (2014)
- Mousavi, S.M., Tavakkoli-Moghaddam, R., Vahdani, B., Hashemi, H., Sanjari, M.J., “A new support vector model-based imperialist competitive algorithm for time estimation in new product development projects”, *Robotics and Computer-Integrated Manufacturing*, 29: 157–168 (2013).

KAYNAKLAR (Devam Ediyor)

- Mozafari, H., Ayob, A., Kamali, F., “Optimization of Functional Graded Plates for Buckling Load by using Imperialist Competitive Algorithm”, *Procedia Technology*, 1: 144 – 152 (2012a).
- Mozafari, H., Abdi, B., Ayob, A., “Application of imperialist competitive algorithm for optimizing a thin resistant interphase”, *Procedia Technology*, 1: 187 – 193 (2012b).
- Mozafari, H., Abdi, B., Ayob, A., “Optimization of Adhesive-Bonded Fiber Glass Strip using Imperialist Competitive Algorithm”, *Procedia Technology*, 1: 194 – 198 (2012c).
- Narendra, K.S., Parthasarathy K., “Identification and control of dynamical systems using neural networks”, *IEEE Transactions on Neural Networks*, 1(1): 4-27 (1990).
- Navaei, J., Ghomi, S.M.T.F., Jolai, F., Mozdgird, A., “Heuristics for an assembly flow-shop with non-identical assembly machines and sequence dependent setup times to minimize sum of holding and delay costs”, *Computers & Operations Research*, 44: 52–65 (2014).
- Nazari-Shirkouhi, S., Eivazy, H., Ghodsi, R., Rezaie, K., Atashpaz-Gargari, E., “Solving the integrated product mix-outsourcing problem using the Imperialist Competitive Algorithm”, *Expert Systems with Applications*, 37: 7615–7626 (2010).
- Nemati, K., Shamsuddin, S.M., Darus, M., “An optimization technique based on imperialist competition algorithm to measurement of error for solving initial and boundary value problems”, *Measurement*, 48: 96–108 (2014).
- Niknam, T., Fard, E.T., Pourjafarian, N., Rousta, A., “An efficient hybrid algorithm based on modified imperialist competitive algorithm and K-means for data clustering”, *Engineering Applications of Artificial Intelligence*, 24: 306–317 (2011).

KAYNAKLAR (Devam Ediyor)

- Nourmohammadia, A., Zandieh, M., Tavakkoli-Moghaddamca, R., “An imperialist competitive algorithm for multi-objective U-type assembly line design”, **Journal of Computational Science**, 4: 393–400 (2013).
- Oussar, Y., Rivals, I., Dreyfus, L., “Training wavelet networks for nonlinear dynamic input output modeling”, **Neurocomputing**, 20: 173-188 (1998).
- Özçalık, H.R., Uygur, A.F., “Dinamik Sistemlerin Uyumlu Sinirsel-Bulanık Ağ Yapısına Dayalı Etkin Modellenmesi”, **KSÜ Fen ve Mühendislik Dergisi**, 6(1): 36-46 (2003).
- Öztürk, M., Hançer, M., Ulutaş, M.S., Can, M., Tartik, M., Günel, A., Şahin U., Şahin, A.D., Akyüz, D.E., “Bulanık Mantık Hesaplamalarına Dayalı Binalarda Isı Kayıp-Kazanç Yaklaşımı”, **Bilimde Modern Yöntemler Sempozyumu-BMYS 2010**, Diyarbakır (2010).
- Rahimi, A., Karimi, H., Afshar-Nadjafi, B., “Using meta-heuristics for project scheduling under mode identity constraints”, **Applied Soft Computing**, 13: 2124–2135 (2013).
- Ramin, R., “Cuckoo Optimization Algorithm”, **Applied Soft Computing**, 11(8): 5508–5518 (2011).
- Rao, R.V., Savsani, V.J., Vakharia, D.P., “Teaching–Learning-Based Optimization: An optimization method for continuous non-linear large scale problems”, **Information Sciences**, 183(1): 1-15 (2012).
- Ravadanegh, S.N., Roshanagh, R.G., “A heuristic algorithm for optimal multistage sizing, siting and timing of MV distribution substations”, **Electric Power Systems Research**, 105: 134– 141 (2013).
- Ravadanegh, S.N., Roshanagh, R.G., “On optimal multistage electric power distribution networks expansion planning”, **Electrical Power and Energy Systems**, 54: 487–497 (2014).

KAYNAKLAR (Devam Ediyor)

- Razmjoooya, N., Mousavib, B.S., Soleymani, F., “A hybrid neural network Imperialist Competitive Algorithm for skin color segmentation”, *Mathematical and Computer Modelling*, 57: 848–856 (2013).
- Sadigh, A.N., Mozafari, M., Karimi, B., “Manufacturer–retailer supply chain coordination: A bi-level programming approach”, *Advances in Engineering Software*, 45: 144–152 (2012).
- Sastry, P.S., Santharam, G., Unnikrishnan, K.P., “Memory neuron networks for identification and control of dynamical systems”, *IEEE Transaction on Neural Networks*, 5(2) (1994).
- Shabani, H., Vahidi, B., Ebrahimpour, M., “A robust PID controller based on imperialist competitive algorithm for load-frequency control of power systems”, *ISA Transactions*, 52: 88–95 (2013).
- Şenol, C., Yildirim, T., “Bulanık-Sinir Ağı Yapısı için Yeni Bir Karma Yaklaşım”, *ELECO’2008 Elektrik-Elektronik ve Bilgisayar Mühendisliği Sempozyumu*, Bursa, (2008).
- Şentürk, S., “Faktöriyel tasarıma adaptif ağ tabanlı bulanık mantık çıkarım sistemi ile farklı bir yaklaşım”, *Dumlupınar Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 22: 57-74 (2010).
- Subaşı, S., Şahin, İ., Çomak, B., “Tahribatsız Test Sonuçları Kullanılarak Uçucu Kül İkameli Betonlarda Basınç Dayanımının ANFIS İle Tahmini”, *SDU International Journal of Technologic Sciences*, 2(3) (2010).
- Taher, S.A., Bagherpour, R., “A new approach for optimal capacitor placement and sizing in unbalanced distorted distribution systems using hybrid honey bee colony algorithm”, *Electrical Power and Energy Systems*, 49: 430–448 (2013).
- Taher, S.A., Fini, M.H., Aliabadi, S.F., “Fractional order PID controller design for LFC in electric power systems using imperialist competitive algorithm”, *Ain Shams Engineering Journal*, 5: 121–135 (2014)

KAYNAKLAR (Devam Ediyor)

- Talatahari, S., Azar, B.F., Sheikholeslami, R., Gandomi, A.H., “Imperialist competitive algorithm combined with chaos for global optimization”, *Commun Nonlinear Sci Numer Simulat*, 17: 1312–1319 (2012).
- Tartik, M., Şahin, M., “Saccharomyces Cerevisia’da Gliserol Üretimini ANFIS İle Modelenmesi”, *Bilimde Modern Yöntemler Sempozyumu-BMYS 2010*, Diyarbakır (2010).
- Tsoukalas, L.H., Uhrig, R.E., “Neuro fuzzy approach for anticipatory control of complex systems”, *IEEE International Conference in Fuzzy Systems*, (1):587-593 (1996).
- Türkbey, O., “Makina sıralama problemlerinde çok amaçlı bulanık küme yaklaşımı”, *Gazi Üniversitesi Müh. Mim. Fakülte Dergisi*, 18: 63-77 (2003).
- Yıldırım, Ö., “Sezgisel arama algoritma tabanlı bulanık sistem optimizasyonu”, Yüksek Lisans Tezi, *Bilecik Üniversitesi Fen Bilimleri Enstitüsü*, Bilecik, 2012.
- Yousefi, M., Darus, A.N., Mohammadi, H., “An imperialist competitive algorithm for optimal design of plate-fin heat exchangers”, *International Journal of Heat and Mass Transfer*, 55: 3178–3185 (2012).
- Zarandi, M.H.F., Zarinbal, M., Ghanbari, N., Turksen, I.B., “A new fuzzy functions model tuned by hybridizing imperialist competitive algorithm and simulated annealing Application: Stock price prediction”, *Information Sciences*, 222: 213–228 (2013).

EK-1: Standart ICA ile fonksiyon (peaks) optimizasyonunda kullanılan kodlar

%% peaks fonksiyonuna hazırlık

```
xxx=-3:0.2:3;
yyy=xxx;
[X,Y]=meshgrid(xxx,yyy);
Z=peaks(X,Y);
figure(3);
mesh(X,Y,Z);
```

%% Problem Bildirimi

```
ProblemParams.CostFuncName = 'BenchmarkFunction'; % ÖlçütFonksiyon Seçimini Burdan yapar
ProblemParams.CostFuncExtraParams = 10; % hangi fonk. üzerinde çalışılacağını belirler.
ProblemParams.NPar = 2; % Fonksiyonda kullanılacak parametre sayısı
ProblemParams.VarMin = -2; % Optimizasyon parametreleri alt limit
ProblemParams.VarMax = 2; % Optimizasyon parametreleri üst limit
```

%VarMin ve VarMax boyutunu değiştirme genel formu

```
if numel(ProblemParams.VarMin)==1
    ProblemParams.VarMin=repmat(ProblemParams.VarMin,1,ProblemParams.NPar);
    ProblemParams.VarMax=repmat(ProblemParams.VarMax,1,ProblemParams.NPar);
end
```

```
ProblemParams.SearchSpaceSize = ProblemParams.VarMax - ProblemParams.VarMin;
```

%% Algoritma Parametre Ayarı

```
AlgorithmParams.NumOfCountries = 20; % Başlangıç ülke sayısı
AlgorithmParams.NumOfInitialImperialists = 4; % Başlangıç imperyalist ülke sayısı
AlgorithmParams.NumOfAllColonies = AlgorithmParams.NumOfCountries -
    AlgorithmParams.NumOfInitialImperialists;

AlgorithmParams.NumOfDecades = 100; % İterasyon Sayısı
AlgorithmParams.AssimilationCoefficient = 2; % Orijinal makaledeki "beta" değeri
AlgorithmParams.AssimilationAngleCoefficient = .5; % Orijinal makaledeki "gama" değeri
AlgorithmParams.Zeta = 0.02; % Total Cost of Empire = Cost of Imperialist +
Zeta * mean(Cost of All Colonies);
```

```
AlgorithmParams.StopIfJustOneEmpire = false; % Tek bir imparatorluk kalana kadar çalışacak
AlgorithmParams.UnitingThreshold = 0.02; % Belli bir arama uzayı oranı iki imparatorluğu
    birleştirir
```

%% Görünüm Ayarı

```
DisplayParams.PlotEmpires = true; % "true" ise çizdir.
if DisplayParams.PlotEmpires
    DisplayParams.EmpiresFigureHandle = figure('Name','Plot of Empiresm','NumberTitle','on');
    DisplayParams.EmpiresAxisHandle = axes;
end
DisplayParams.PlotCost = true; % "true" ise çizdir.
if DisplayParams.PlotCost
    DisplayParams.CostFigureHandle = figure('Name','Plot of Minimum and Mean
        Costs','NumberTitle','on');
    DisplayParams.CostAxisHandle = axes;
end
ColorMatrix = [1 0 0 ; 0 1 0 ; 0 0 1 ; 1 1 0 ; 1 0 1 ; 0 1 1 ; 1 1 1 ;
```

```

0.5 0.5 0.5; 0 0.5 0.5 ; 0.5 0 0.5 ; 0.5 0.5 0 ; 0.5 0 0 ; 0 0.5 0 ; 0 0 0.5 ;
1 0.5 1 ; 0.1*[1 1 1]; 0.2*[1 1 1]; 0.3*[1 1 1]; 0.4*[1 1 1]; 0.5*[1 1 1]; 0.6*[1 1 1]];
DisplayParams.ColorMatrix = [ColorMatrix ; sqrt(ColorMatrix)];
DisplayParams.AxisMargin.Min = ProblemParams.VarMin;
DisplayParams.AxisMargin.Max = ProblemParams.VarMax;

%% Başlangıç İmparatorluklarını Oluştur
InitialCountries = GenerateNewCountry(AlgorithmParams.NumOfCountries , ProblemParams);
% Herbir ülkenin ölçütünü hesapla en düşük ölçütlü
% daha fazla güçlüdür.

if isempty(ProblemParams.CostFuncExtraParams)
    InitialCost = feval(ProblemParams.CostFuncName,InitialCountries);
else
    InitialCost = feval(ProblemParams.CostFuncName,
        InitialCountries,ProblemParams.CostFuncExtraParams);
end
[InitialCost,SortInd] = sort(InitialCost); % Artan düzende maliyetini sıralayım.En iyi
% ülkeler yüksek yerlerde olacaktır
InitialCountries = InitialCountries(SortInd,:); % Ölçütlerine göre popülasyonları sırala

Empires = CreateInitialEmpires(InitialCountries,InitialCost,AlgorithmParams,ProblemParams);

%% Ana Döngü
MinimumCost = repmat(nan,AlgorithmParams.NumOfDecades,1);
MeanCost = repmat(nan,AlgorithmParams.NumOfDecades,1);
if DisplayParams.PlotCost
    axes(DisplayParams.CostAxisHandle);
    if any(findall(0)==DisplayParams.CostFigureHandle)
        h_MinCostPlot=plot(MinimumCost,'r','LineWidth',1.5,'YDataSource','MinimumCost');
        hold on;
        h_MeanCostPlot=plot(MeanCost,'k','LineWidth',1.5,'YDataSource','MeanCost');
        hold off;
        pause(0.005);
    end
end
for Decade = 1:AlgorithmParams.NumOfDecades
    Remained = AlgorithmParams.NumOfDecades - Decade % Geriye Kalan İterasyon Sayısı
    for ii = 1:numel(Empires) % Herbir imparatorluk için tek tek bak

        %% Asimilasyon; Kolonilerin Emperyalistler Doğru Hareketi
        Empires(ii) = AssimilateColonies(Empires(ii),AlgorithmParams,ProblemParams);

        %% Devrim/Değişim, Sosyo-Politik Özelliklerin Ani Değişimi
        Empires(ii) = RevolveColonies(Empires(ii),AlgorithmParams,ProblemParams);

        %% Yeni Maliyet Değerlendirmesi
        if isempty(ProblemParams.CostFuncExtraParams)
            Empires(ii).ColoniesCost = feval(ProblemParams.CostFuncName,Empires(ii).ColoniesPosition);
        else
            Empires(ii).ColoniesCost = feval(ProblemParams.CostFuncName,Empires(ii).ColoniesPosition,
                ProblemParams.CostFuncExtraParams);
        end
    end
    %% Egemen imparatorluk- Güç Egemenliği
    Empires(ii) = PossesEmpire(Empires(ii));

```

%% İmparatorluğun Toplam Gücünü Hesaplama

```
Empires(ii).TotalCost = Empires(ii).ImperialistCost + AlgorithmParams.Zeta *
    mean(Empires(ii).ColoniesCost);
```

```
end
```

%% Benzer İmparatorlukları Birleştirme (Ele geçiriş)

```
Empires = UniteSimilarEmpires(Empires,AlgorithmParams,ProblemParams);
```

%% Emperyalistlik Yarış

```
Empires = ImperialisticCompetition(Empires);
if numel(Empires) == 1 && AlgorithmParams.StopIfJustOneEmpire
    break
end
```

%% Sonuçları Görüntüle

```
figure(1)
contour(X,Y,Z); % Ülkeleri 2Boyut olarak Çiz
hold on
DisplayEmpires(Empires,AlgorithmParams,ProblemParams,DisplayParams);
ImerialistCosts = [Empires.ImperialistCost];
MinimumCost(Decade) = min(ImerialistCosts);
MeanCost(Decade) = mean(ImerialistCosts);
if DisplayParams.PlotCost
    refreshdata(h_MinCostPlot);
    refreshdata(h_MeanCostPlot);
    drawnow;
    pause(0.001);
end
end % Algoritmanın sonu
MinimumCost(end)
```

EK-2: Literatür kıyaslaması için kullanılan uygunluk/kalite hesaplama

fonksiyonu kodları

```
function Cost=fxhesapla_egt(Colonies,veri)

a=size(Colonies);
x1=veri(:,1); x2=veri(:,2); x3=veri(:,3);

for p=1:a(1)
    ica=Colonies(p,:);

    %%UF tanımları
    G(1,:)=gaussmf(x1,[ica(7) ica(1)]);
    G(2,:)=gaussmf(x1,[ica(8) ica(2)]);
    %
    G(3,:)=gaussmf(x2,[ica(9) ica(3)]);
    G(4,:)=gaussmf(x2,[ica(10) ica(4)]);
    %
    G(5,:)=gaussmf(x3,[ica(11) ica(5)]);
    G(6,:)=gaussmf(x3,[ica(12) ica(6)]);

    E=0;

    for ite=1:length(x1)
        Tnorm(1)=G(1,ite)*G(3,ite)*G(5,ite);
        Tnorm(2)=G(1,ite)*G(3,ite)*G(6,ite);
        Tnorm(3)=G(1,ite)*G(4,ite)*G(5,ite);
        Tnorm(4)=G(1,ite)*G(4,ite)*G(6,ite);

        Tnorm(5)=G(2,ite)*G(3,ite)*G(5,ite);
        Tnorm(6)=G(2,ite)*G(3,ite)*G(6,ite);
        Tnorm(7)=G(2,ite)*G(4,ite)*G(5,ite);
        Tnorm(8)=G(2,ite)*G(4,ite)*G(6,ite);

        T=sum(Tnorm);

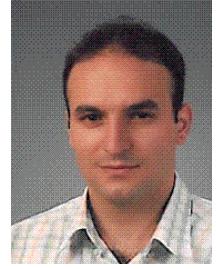
        wT=0;
        for o=1:8
            w(o)=Tnorm(o)/T; %%%
        end

        wfT=0;
        for o=1:8
            wf(o)=w(o)*(ica((o+4)*3-2)*x1(ite)+ica((o+4)*3-1)*x2(ite)+ica((o+4)*3));
            wfT=wfT+wf(o);
        end

        cikis=wfT;
        e=cikis-veri(ite,4); %
        E=e*e+E;
    end
    Cost(p,1)=(1/length(x1))*E;
end

end
```

ÖZGEÇMİŞ



Kişisel Bilgiler

Adı Soyadı : Mehmet SARIKOÇ
Doğum Yeri ve Tarihi : Kayseri - 10.08.1984

Eğitim Durumu

Lisans Öğrenimi : Sakarya Üniversitesi, Teknik Eğitim Fakültesi,
Bilgisayar Sistemleri Öğretmenliği
Bildiği Yabancı Diller : İngilizce
Bilimsel Faaliyetleri : -

İş Deneyimi

Stajlar : Türk Telekom Kayseri İl Müdürlüğü,
Bilgi İşlem-2001
TCDD Kayseri Sinyal Şefliği-2004
Projeler : -
Çalıştığı Kurumlar : Sivrihisar ATL, TL ve EML / Eskişehir-
Bilgisayar Öğretmenliği
Türk Telekom Teknik ve Endüstri Meslek
Lisesi / Eskişehir – Bilgisayar Öğretmenliği

İletişim

Adres : Bahçelievler Mah. Balkan Cad. No:76/15
Tepebaşı/ESKİŞEHİR
Tel : 0 506 264 42 20
E-Posta Adresi : m_sarikoc@hotmail.com

Tarih:

İmza: