



BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

**SEZGİSEL ARAMA ALGORİTMA TABANLI BULANIK
SİSTEM OPTİMİZASYONU**

Özlem YILDIRIM
Yüksek Lisans Tezi

Yrd. Doç. Dr. Cihan KARAKUZU

BİLECİK, 2012

BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

**SEZGİSEL ARAMA ALGORİTMA TABANLI BULANIK
SİSTEM OPTİMİZASYONU**

Özlem YILDIRIM
Yüksek Lisans Tezi

Yrd. Doç. Dr. Cihan KARAKUZU

BİLECİK, 2012



**BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS
JÜRİ ONAY FORMU**

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
29.11.2012 tarih ve 28 sayılı kararıyla oluşturulan jüri
tarafından 10.12.2012 tarihinde tez savunma sınavı yapılan
Özlem YILDIRIM'ın "Sezgisel Arama Algoritma
Tabanlı Bulanık Sistem Optimizasyonu" başlıklı tez çalışması
Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK
LİSANS tezi olarak oy birliği/oy çokluğu ile kabul edilmiştir.

JÜRİ

ÜYE Yrd. Doç. Dr. Cihan KARAKURU
(TEZ DANIŞMANI):

ÜYE: Prof. Dr. Yaşar BECERİKLİ

ÜYE: Yrd. Doç. Dr. Fuat KARAKAYA

ONAY

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

İmza/Mühür

ÖZET

ABC algoritması bal arıların doğadaki yiyecek kaynaklarını bulma ve nektar toplama davranış ilkelerine dayalı güncel bir sezgisel optimizasyon algoritmasıdır. Bu çalışmada, doğrusal olmayan dinamik sistemlerin ANFIS (Adaptive Neuro-Fuzzy Inference System) bulanık çıkarım modeli kullanılarak modellenmesi problemi üzerinde Yapay Arı Kolonisi (Artificial Bee Colony, ABC) algoritmasının optimizasyon başarımı incelenmiştir. İnceleme için literatürden seçilen beş denektaşı kullanılmıştır. Her bir denektaşı problem parametreleri ABC ile ayarlanan ANFIS ile modellenmiştir.

Her bir denektaşı için ABC ile elde edilen sonuçlar popüler ve sıklıkla kullanılan PSO ve DE'nin sonuçlarıyla kıyaslamalı olarak irdelenmiştir. Çalışma çerçevesinde elde edilen istatistiki verilere göre, ABC algoritmasının DE'ye çok yakın başarımla gösterdiği belirlenmiştir. Ancak bir nesil başına koşma zamanı kategorisinde ABC'nin başarımla en yakın rakibi olan DE'ye göre iki kat daha hızlı olduğu sonucuna varılmıştır. Bu sonucu dayalı olarak ABC algoritmasının gerçek zaman gömülü sistem uygulamalarında işletiminin diğerlerine nazaran daha verimli olacağı değerlendirilmiştir.

Anahtar Kelimeler: Yapay Arı Kolonisi Algoritması, ANFIS, Bulanık Modelleme, Dinamik Sistem

ABSTRACT

ABC algorithm is an up-to-date heuristic optimization method based on principles of honey bees behaviors in finding/gathering nectar from food sources in the nature. In this study, performance of the ABC algorithm is investigated on the problem of nonlinear dynamic systems modeling/ identification using ANFIS (Adaptive Neuro-Fuzzy Inference Systems) fuzzy inference model. For this investigation, five benchmark problems selected from the literature have been used.

Each benchmark problem is identified /modeled with ANFIS which of parameters tuned by ABC. Obtained results for each benchmark have been comparatively examined with the results of PSO and DE popular and commonly used algorithms. According to the obtained results in this study, it is determined that performance of ABC algorithm is very close to performance of DE algorithm. However, it is concluded that performance of ABC in terms of run time per generation was two times better according to that of DE. Based on this conclusion, it is evaluated that operating of ABC algorithm in the application of real time embedded systems would be more efficient.

Key words: Artificial Bee Colony, ANFIS, Fuzzy Modeling, Dynamic System.

TEŞEKKÜR

Tez çalışmamda her konuda bilgi ve deneyimleri ile bana yol gösteren danışman hocam Yrd.Doç.Dr. Cihan KARAKUZU'ya ve tez ile ilgili yardımlarından dolayı Yrd. Doç.Dr. Uğur YÜZGEÇ'e çok teşekkür ederim.

Destek ve anlayışından dolayı sevgili eşim Mehmet Süleyman YILDIRIM'a ve aileme çok teşekkür ederim.

Bilecik Üniversitesi Bilimsel Araştırma Projeleri Birimine 2010-02-BİL.01-002 kodlu proje desteğinden dolayı teşekkür ederim.

İÇİNDEKİLER

ÖZET.....	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER	iv
ÇİZELGELER DİZİNİ	vii
ŞEKİLLER DİZİNİ	viii
1. GİRİŞ.....	1
1.1. Literatür Taraması	3
1.2. Tez Çalışması Kapsamı	6
2. ÇALIŞMADA KULLANILAN SEZGİSEL ARAMA ALGORİTMALARI	7
2.1. Yapay Arı Kolonisi Algoritması	7
2.1.1. Doğada arılar	7
2.2.2. Arıların yiyecek bulma davranışı ve salınım dansı	7
2.2. Yapay Arı Kolonisi (ABC) ve Temel Adımları	10
2.2.1. Başlangıç yiyecek kaynağı bölgelerinin üretilmesi.....	11
2.2.2. İşçi arıların yiyecek kaynağı bölgelerine gönderilmesi.....	11
2.2.3. Gözcü arıların seleksiyonda kullanacakları olasılık değerlerinin hesaplanması	12
2.2.4. Gözcü arıların yiyecek kaynağı bölgesi seçmeleri	13
2.2.5. Kaynağı bırakma kriteri: Limit ve kâşif arı üretimi	13
2.2.6. ABC algoritması ile fonksiyon optimizasyon örneği	13
2.2.7. ABC algoritması ile PID denetleyici parametre optimizasyonu örneği	16
2.4. Parçacık Sürü Optimizasyonu (PSO, Partical Swarm Optimization)	19

2.5. Farksal Gelişim (DE, Differentiol Evalution) :	20
3. KULLANILAN DİNAMİK SİSTEMLER, BULANIK SİSTEM VE SEZGİSEL ÖĞRENME YÖNTEMLERİ.....	22
3.1. Örnek Dinamik Sistemler	22
3.2. ANFIS Bulanık Çıkarım Sistemi.....	25
3.3. Kullanılan Algoritmalar	27
3.3.1. ABC algoritması.....	27
3.3.2. PSO.....	29
3.3.3. DE.....	29
3.4. Bireylerin Yapısı	30
3.5.ABC ile ANFIS parametre optimizasyonu.....	30
4. SEZGİSEL ARAMA ALGORİTMA TABANLI BULANIK SİSTEM OPTİMİZASYONU	31
4.1. ÖDS 1 ‘in Tanınması/Modellenmesi.....	32
4.1.1.ÖDS 1 için eğitim aşaması ve sonuçları.....	32
4.1.2.ÖDS 1’in test aşaması ve sonuçları.....	38
4.2. ÖDS 2’nin Tanınması/Modellenmesi.....	42
4.2.1.ÖDS 2 için eğitim aşaması ve sonuçları.....	42
4.2.2. ÖDS 2’in test aşaması ve sonuçları.....	47
4.3. ÖDS 3’ün Tanınması/Modellenmesi.....	50
4.3.1. ÖDS 3 için eğitim aşaması ve sonuçları.....	50
4.3.2.ÖDS 3’ün test aşaması ve sonuçları	58
4.4. ÖDS 4’ün Tanınması/Modellenmesi.....	61
4.4.1. ÖDS 4 için eğitim aşaması ve sonuçları.....	61
4.4.2. ÖDS 4’ün test aşaması ve sonuçları	65
4.5. ÖDS 5’in Tanınması/Modellenmesi.....	68

4.5.1. ÖDS 5'in eğitim aşaması ve sonuçları	68
4.5.2. ÖDS 5'in test aşaması ve sonuçları	72
4.6. PSO ve DE Algoritmaları ile ABC'nin Performansının Kıyaslanması....	75
4.7. Literatüre Dayalı Başarım Kıyaslaması	79
5. SONUÇLAR	89
6. KAYNAKLAR	91
EK-1: Standart ABC ile fonksiyon (peaks) optimizasyonunda kullanılan kodlar..	96
EK-2: Literatür kıyaslaması için kullanılan uygunluk/kalite hesaplama fonksiyonu kodları	97
ÖZGEÇMİŞ.....	98

ÇİZELGELER DİZİNİ

Çizelge 3. 1.	ABC, PSO ve DE öğrenmeli bulanık mantık tabanlı dinamik sistem tanıma/modelleme için literatürden seçilen örnek dinamik sistemler (ÖDS)	23
Çizelge 3. 2.	Her bir ÖDS için kullanılan ANFIS yapısı.....	23
Çizelge 3. 3.	ABC parametreleri.....	28
Çizelge 3. 4.	PSO parametreleri.....	29
Çizelge 3. 5.	DE Parametreleri	29
Çizelge 4. 1.	ÖDS 1'in işaretlenen alan için elde edilen hata değeri.....	38
Çizelge 4. 2.	ÖDS 2'nin işaretlenen alan için elde edilen hata değeri.....	46
Çizelge 4. 3.	ÖDS 3'ün Standart ABC kullanılarak bir kez koşturulması sonucu elde edilen ANFIS parametrelerinin değişimi	55
Çizelge 4. 4.	ABC, PSO ve DE algoritmalarının ÖDS modellemede <i>EĞİTİM</i> fazı başarımların kıyaslaması	77
Çizelge 4. 5.	ABC, PSO ve DE algoritmalarının ÖDS modellemede <i>TEST</i> fazı başarımların kıyaslaması	78
Çizelge 4. 6.	ABC Parametreleri.....	80
Çizelge 4. 7.	PSO Parametreleri	80
Çizelge 4. 8.	DE Parametreleri	80

ŞEKİLLER DİZİNİ

Sayfa No

Şekil 2. 1.	Arıların yiyecek arama davranışı(Akay,2009)	10
Şekil 2. 2.	ABC ile fonksiyon optimizasyonu blok yapısı	14
Şekil 2. 3.	“Peaks” fonksiyonun tanımladığı yüzey.....	15
Şekil 2. 4.	Algoritmanın çeşitli nesil sayılarında arıların arama uzayı konumları.....	16
Şekil 2. 5.	ABC ile PID denetleç parametre optimizasyonu blok yapısı	17
Şekil 2. 6.	DA motor modeli öbek şeması (Babuska,1999).....	17
Şekil 2. 7.	ABC algoritmasının 100 kez koşturulması sonucu bulunan sonuçlar	19
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.....	24
Şekil 3.2.	ÖDS sistemleri için sistem tanımanın 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.....	25
Şekil 3. 3.	İki girişli-tek çıkışlı birinci dereceden kural polinomlu ANFIS mimarisi	26
Şekil 3. 4.	ABC ile ANFIS optimizasyonu blok yapısı	30
Şekil 4. 1.	ABC öğrenmeli ANFIS ile ÖDS 1 için eğitim fazı sistem tanıma sonuçları.	33
Şekil 4. 2.	PSO ve DE öğrenmeli ANFIS ile ÖDS 1 için eğitim fazı sistem tanıma sonuçları.....	34
Şekil 4. 3.	ABC öğrenmeli ANFIS ile ÖDS 1’in bulanık modelleme eğitim fazında UF’lerin başlangıç ve son durumları.	35
Şekil 4. 4.	PSO ve DE öğrenmeli ANFIS ile ÖDS 1’in bulanık modelleme eğitim fazında UF’lerin başlangıç ve son durumları.....	36
Şekil 4. 5.	ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 1 için elde edilen modellerin karşılaştırılması.....	37

Şekil 4. 6.	ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 1 modelleme eğitim seyri .38
Şekil 4. 7.	ABC öğrenmeli ANFIS ile ÖDS 1 için test fazı sistem tanıma sonuçları .40
Şekil 4. 8.	PSO ve DE öğrenmeli ANFIS ile ÖDS 1 için test fazı sistem tanıma sonuçları.....41
Şekil 4. 9.	ABC öğrenmeli ANFIS ile ÖDS 2 için eğitim fazı sistem tanıma sonuçları44
Şekil 4. 10.	PSO ve DE öğrenmeli ANFIS ile ÖDS 2 için eğitim fazı sistem tanıma sonuçları.....44
Şekil 4. 11.	ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 2 elde edilen modellerin karşılaştırılması45
Şekil 4. 12.	ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 2 modelleme eğitim seyri .46
Şekil 4. 13.	ABC öğrenmeli ANFIS ile ÖDS 2 için test fazı sistem tanıma sonuçları .48
Şekil 4. 14.	PSO ve DE öğrenmeli ANFIS ile ÖDS 2 için test fazı sistem tanıma sonuçları.....49
Şekil 4. 15.	ABC öğrenmeli ANFIS ile ÖDS 3 için eğitim fazı sistem tanıma sonuçları51
Şekil 4. 16.	PSO ve DE öğrenmeli ANFIS ile ÖDS 3 için eğitim fazı sistem tanıma sonuçları.....52
Şekil 4. 17.	ÖDS 3 için ABC öğrenmeli ANFIS ile eğitim fazı için kullanılan UF'lerin başlangıç ve son durumları53
Şekil 4. 18.	ÖDS 3 için PSO ve DE öğrenmeli ANFIS ile eğitim fazı için kullanılan UF'lerin başlangıç ve son durumları54
Şekil 4. 19.	ANFIS'in bir parametresinin eğitim süresince değişim seyri.....56
Şekil 4. 20.	ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 3 için elde edilen modellerin karşılaştırılması.....57
Şekil 4. 21.	ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 3 için modelleme eğitim seyri.....57
Şekil 4. 22.	ABC öğrenmeli ANFIS ile ÖDS 3 için test fazı sistem tanıma sonuçları .59
Şekil 4. 23.	PSO ve DE öğrenmeli ANFIS ile ÖDS 3 için test fazı sistem tanıma sonuçları.....60
Şekil 4. 24.	ABC öğrenmeli ANFIS ile ÖDS 4 için eğitim fazı sistem tanıma sonuçları62

Şekil 4. 25.	PSO ve DE öğrenmeli ANFIS ile ÖDS 4 için eğitim fazı sistem tanıma sonuçları.....	63
Şekil 4. 26.	ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 4 için elde edilen modellerin karşılaştırılması.	64
Şekil 4. 27.	ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 4 modelleme eğitim seyri .	64
Şekil 4. 28.	ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 4 için test fazı sistem tanıma sonuçları.....	66
Şekil 4. 29.	ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 4 için test fazı sistem tanıma sonuçları.....	67
Şekil 4. 30.	ABC öğrenmeli ANFIS ile ÖDS 5 için eğitim fazı sistem tanıma sonuçları	69
Şekil 4. 31.	PSO ve DE öğrenmeli ANFIS ile ÖDS 5 için eğitim fazı sistem tanıma sonuçları.....	70
Şekil 4. 32.	ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 5 için elde edilen modellerin karşılaştırılması.	71
Şekil 4. 33.	ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 5 modelleme eğitim seyri	71
Şekil 4. 34.	ABC öğrenmeli ANFIS ile ÖDS 5 için test fazı sistem tanıma sonuçları	73
Şekil 4. 35.	ABC öğrenmeli ANFIS ile ÖDS 5 için test fazı sistem tanıma sonuçları	74
Şekil 4. 36.	Üç üyelik fonksiyonu ile modelleme başarımı: sol kolon Jang (Jang, 1993)'den alınan sonuçlar, sağ kolon ABC ile elde edilen sonuçlar	82
Şekil 4. 37.	Beş üyelik fonksiyonu ile modelleme başarımı: sol kolon Jang (Jang, 1993)'den alınan sonuçlar, sağ kolon ABC ile elde edilen sonuçlar.....	83
Şekil 4. 38.	Yedi ÜF kullanan ANFIS ile elde edilen (sağ kolon: d, e, f) (Jang,1993) ile (sol kolon:a, b, c) kıyaslamalı sonuçlar: (a) ve(d) u(k) harici giriş, (b) ve (e) ANFIS çıkışı f(u(k)), (c) ve (f) sistem çıkışı y(k+1)	84
Şekil 4. 39.	Beş üyelik fonksiyonu kullanılarak ABC, PSO ve DE ile elde edilen eğitim (sol kolon) ve test (sağ kolon) sonuçları.....	85
Şekil 4. 40.	ABC, PSO ve DE ile beş üyelik fonksiyonlu ANFIS optimizasyonunda üyelik fonksiyonlarının başlangıç ve son durumları.....	86
Şekil 4. 41.	Yedi üyelik fonksiyonu kullanılarak ABC, PSO ve DE ile elde edilen eğitim (sol kolon) ve test (sağ kolon) sonuçları.....	87

Şekil 4. 42.	ABC, PSO ve DE ile yedi üyelik fonksiyonlu ANFIS optimizasyonunda üyelik fonksiyonlarının başlangıç ve son durumları.....	88
---------------------	---	----

1. GİRİŞ

Günümüzde gelişen teknolojiye bağımlı olarak çoğu konu zorlaşmış ve insanlar tarafından anlaşılması zor hale gelmiştir. Bu zorlukları aşmak, öğrenilmesi gereken konuları inceleyip uygulama yapabilmek için bilişim teknolojilerinden faydalanılmaya başlanmıştır. Bu nedenle insanlar geleneksel eğitime oranla, bilgisayar destekli eğitime yönelmiş; bu da hem zamandan tasarruf sağlamış hem de daha yüksek başarımlar elde edilmesine olanak sağlamıştır.

Geleneksel yöntemlerle laboratuvar ortamında yapılmaya çalışılan deneylerde elektrik ile ilgili çeşitli kazalar, kullanılan malzemelerin genellikle pahalı olması ve temininin zor olması gibi sayılabilecek birçok olumsuz etken bulunmaktadır. Fakat tüm bunlar bir bilgisayar benzetimi kullanılarak çok rahat bir şekilde yapılmaktadır. Bilgisayar benzetiminde bir sistemin optimizasyonu ya da bir devrenin incelenmesi gibi olaylarda gerekli elemanlar kullanılarak, sistemin başarımları laboratuvar ortamına göre daha kolay incelenebilmekte, sistemde değişiklikler yapılmasına imkân vererek verimin artması sağlanmaktadır. Bu nedenle bilgisayar destekli tasarım, modelleme ve benzetimde mükemmele ulaşma isteği yapay zekâ, yapay sinir ağları, bulanık mantığın ortaya çıkmasına neden olmuştur.

Günlük yaşantımızda bir olayı anlatırken, bir nesneyi tanımlarken kullandığımız ifadeler belirsizlik diğer bir tabir ile bulanıklık içerir. Bu belirsizlikler Bulanık Mantık'ın konusu olarak ele alınmıştır. Yani bulanık mantık düşüncelerimizdeki kesin doğru ya da yanlışlar arasındaki ihtimalleri değerlendirmeye yararmaktadır.

Bulanık mantık ilk kez Zadeh tarafından tanımlanmıştır. Klasik mantığa göre bir şey ya siyahtır ya da beyazdır. Bulanık mantığa göre ise kısmen siyah ya da kısmen beyaz olabilir. Zadeh bilgisayardaki 1 ve 0 kesinliği yerine insan düşüncesindeki gibi ara değerleri de almış ve gerçek hayattakine benzetmiştir.

Bilindiği üzere çoğu fiziksel sistem davranışı doğrusal değildir. Bulanık mantığın ortaya çıkışından bu zamana kadar birçok bilim adamı doğrusal olmayan sistemlerin modellenmesi üzerine çalışmışlardır. Bulanık mantığın kullanılmasıyla doğrusal olmayan sistemlerde sistem başarımları artmış, uygulama kolaylaşmış ve en önemlisi gerçeğe yakın sonuçlar alınmıştır. Kısaca bulanık mantık özellikle doğrusal

olmayan sistemlerin modellenmesi, kestirimi, sezimi, denetimi, vb. problemlerde üstün başarı sağlamıştır.

Gerçek hayatta problemlerin çoğu doğrusal olmayan bir yapıya sahiptir. Özellikle bazı problemlerdeki değişken sayılarının, zorluk derecelerinin yüksek olması klasik çözüm yöntemlerini başarısız kılmaktadır. Doğrusal olmayan problemlerde bulanık mantığın etkin kullanımının yanında doğadaki sistemlerden esinlenerek ortaya çıkarılmış birçok yöntem bu tip problemlerin çözümünde de kullanılmaktadır. Özellikle sürü tabanlı sezgisel arama algoritmaları problem çözümünde hızlıdır ve doğru sonuçlar verebilmektedir.

Genellikle bu sezgisel arama algoritmaların ortak noktaları bireylerin çevresiyle ve diğer bireylerle olan etkileşimini ele almalarıdır. Buna sürü zekâsı adı verilmektedir. Sürülerde hayatta kalma açısından yiyecek bulmak büyük önem taşımaktadır. Yiyecek bulma çalışmalarında doğada yaşayan canlılar sürekli bir iş bölümü içerisinde. Yapay Arı Kolonisi Algoritması (Artificial Bee Colony, ABC) bu tip algoritmaların en güncel olanlarından biridir. Bu algoritma bal arılarının zengin çiçek nektarları bulmak için öncelikle kâşif arıyı çevreye gönderip çiçek nektarı olan alanları tespit ederek kovana geri dönmeleri ve kovandaki işçi arılara farklı dans çeşitleriyle nektar kaynağının uzaklık ve verimlilik bilgisini vermelerinden esinlenerek 2005 yılında Kayseri Erciyes Üniversitesi Bilgisayar Mühendisliği Bölümü öğretim üyesi Prof. Dr. Derviş Karaboğa tarafından 2005 yılında yayınlanan çalışma (Karaboğa,2005) ile bilimsel camiaya tanıtılmıştır.

ABC algoritması dışında doğadaki sürülerin yiyecek bulma davranışlarından yola çıkarak modellenen birçok başarılı algoritmalar bulunmaktadır. Bu algoritmaların başlıcaları şunlardır: 70'li yıllarda Holland tarafından ve 80'li yıllarda Golberg tarafından geliştirilmiş Genetik Algoritma (GA), Dorigo tarafından özellikle mühendislik alanında sürekli optimizasyon problemleri için önerilmiş Karınca Kolonisi Algoritması (Ant Colony Optimization), 1995 yılında J.Kennedy ve R.C.Eberhart tarafından geliştirilmiş Parçacık Sürü Optimizasyonu (PSO), Price ve Storn (Price, 1999), tarafından 1995 yılında geliştirilmiş Farksal Gelişim (Differential Evolutionary, DE) algoritması.

ABC algoritmasının amacı en iyiye en az enerjiyle en kısa zamanda bulmaktır. ABC'nin bu özelliği kısa zamanda kendisine farklı kullanım alanları oluşturmuş ve oluşturmaya devam etmektedir. Bir sonraki bölümde verilen literatür özetinden de anlaşılacağı üzere, algoritma henüz bulanık bir sistemin (özellikle ANFIS) optimizasyonunda kullanılmamıştır.

Bu tezde bulanık işleyişi gerçekleyen ANFIS (Adaptive Neuro-Fuzzy Inference System) bulanık mantık modeli kullanılarak; ABC'nin doğrusal olmayan dinamik sistemlerin bulanık modellenmesinde optimizasyon başarımı, popüler ve sıklıkla kullanılan PSO ve DE'nin başarımlarıyla kıyaslamalı olarak incelenmiştir.

1.1. Literatür Taraması

Prof.Dr. Derviş KARABOĞA (Karaboğa, 2005) tarafından geliştirilen ABC algoritması ile ilgili olarak 2005 yılından 2012 yılına kadar çeşitli çalışmalar yapılmıştır. Aşağıda verilen literatür taramasına göre bu güne kadar ABC algoritması tabanlı bulanık mantık ile ilgili sadece Chaiyatham ve Ngamroo (Chaiyatham ve Ngamroo ,2012) tarafından bir çalışma yapıldığı görülmüştür. Yapılan çalışmalardan bazılarının kısa özeti aşağıda verilmiştir.

Chaiyatham ve Ngamroo (Chaiyatham ve Ngamroo ,2012) bulanık bir PID denetleyicinin parametre optimizasyonunu BCO (Bee Colony Optimization) dedikleri aslında ABC olan algoritmayla önermişlerdir. Çalışmada ABC'nin üstünlükleri üzerinde çalışılan (EZ aktif ve reaktif güç kontrol modeli) problem için vurgulanmıştır.

Su vd. (Su vd., 2012)'de veriden başarımlı geliştirilmiş T-S bulanık model çıkarımını otomatik olarak yapmak maksadıyla yeni bir yöntem önerilmiştir. Bu yöntemde, Yapay Arı Koloni (ABC) algoritmasına dayalı değişken genotip (genotype) düşüncesi yeni bir algoritmaya (VABC) dönüştürülmüştür.

Cuevas vd. (Cuevas, Sencion-Echauri, Zaldivar, 2012) tarafından yapılan çalışmada ABC algoritmasından türetilmiş DisABC adı verilen ikili optimizasyon algoritması geliştirilmiştir. PSO ve DE tabanlı muadilleriyle karşılaştırıldığında DisABC'nin ikili optimizasyon problemleri için iyi sonuçlar verdiği gözlenmiştir.

Karaboga, Cetinkaya (Karaboga, Cetinkaya, 2011) tarafından yapılan çalışmada ABC algoritması gürültüyü azaltmak için uyarlamalı FIR ve IIR tasarımında

kullanılmıştır. ABC, PSO ve DE ile karşılaştırılmış ve ABC yaklaşımının tasarımlarda ve gürültünün iptal edilmesinde başarılı olduğu görülmüştür.

Sonmez (Sonmez, 2011) tarafından yapılan çalışmada ABC algoritması ile bağ yapılarının (truss structure) ağırlığını minimize etme önerilmiştir. 200 den fazla bileşen ve sabit geometrili beş bağ yapısı örneği ile ABC algoritmasının literatürdeki diğer optimizasyon algoritmalarıyla karşılaştırıldığında bağ yapıları için güçlü bir arama ve optimizasyon tekniği olduğu gösterilmiştir.

Karaboga, Akay (Karaboga, Akay, 2011) tarafından yapılan çalışmada kısıtlı optimizasyon problemleri için değiştirilmiş ABC algoritması önerilmiş ve performansı HM, SR, ISR, OPA, ASCHEA, SMES, GA, DE, PSO algoritmaları ile karşılaştırılmıştır. ABC algoritmasının ele alınan kısıtlı optimizasyon problemlerini çözmek için kullanılabilir olduğu sonucuna varılmıştır.

Ziarati vd. (Ziarati, Akbari, Zeighami, 2011)'de arama kısıtlı proje çizelgeleme problemi üzerinde üç farklı arı koloni algoritması çalışılmış incelenen diğer algoritmalar ile karşılaştırıldığında arı algoritmalarının ele alınan problem için etkili olduğu bildirilmiştir.

Samrat vd. (Samrat, SibaK, Ajith, 2011) tarafından yapılan çalışmada ABC algoritması kullanılarak MOSFET küçük sinyali eşdeğer devre modelinin parametreleri belirlenmiş ve hesaplama zamanı ve çözüm kalitesi açısından ABC ve PSO'nun başarımı karşılaştırılmıştır. ABC algoritmasının PSO'ya göre daha hızlı yakınsadığı bildirilmiştir.

Wang vd. (Wang, Chen, Tellambura, 2010) tarafından yapılan çalışmada OFDM'de faz faktörlerinin en iyi kombinasyonunu aramak için değiştirilmiş ABC algoritması tabanlı yeni bir yöntem önerilmiştir. ABC-PTS adı verilen yöntem ile hesaplama yükünün azaltıldığı bildirilmiştir.

Xu, Duan (Xu, Duan, 2010)'de alçak irtifada uçağın görsel hedef tanınması için yeni bir şekil eşleştirme yaklaşımı tanımlanmıştır. ABC algoritmasının kullanıldığı yaklaşımın GA'ya göre daha iyi sonuç verdiği gösterilmiştir.

S. and Sishaj (S., Sishaj, 2010) makalede, ABC algoritması ekonomik yük dağıtım sorununu çözmek için kullanılmış ve ABC'nin karmaşık olmayan güç sistemi

dağıtım problemlerinin çözümü için umut verici bir yöntem olduğu sonucuna varılmıştır.

Zhang vd. (Zhang ,Ouyang, Ning, 2010)'da popüler olan veri analizi ve veri madenciliği tekniği için bir yaklaşım sunulmuştur. Bu çalışmada ABC algoritması N nesnesinin K kümeye optimal kümelenmesi amacıyla kullanılmıştır. Bu algoritma ile çok iyi bilinen gerçek veri kümeleri üzerinde popüler olan algoritmalar (GA, SA, TS, ACO) ve son zamanlarda önerilen K-NM-PSO ile karşılaştırılma yapılmıştır. Buna göre ABC algoritması çözüm kalitesi ve gerekli işlem zamanı açısından diğer algoritmalara göre daha iyi olduğu gösterilmiştir.

Karaboga, Ozturk (Karaboga, Ozturk, 2009) tarafından yapılan çalışmada; veri kümelerini sınıflama amacıyla yapay sinir ağı (YSA) eğitiminde ABC kullanılmıştır. Veri kümelerini sınıflamak için kullanılan diğer algoritmalarla karşılaştırılarak ABC'nin YSA eğitiminde etkin bir şekilde kullanılabileceği bildirilmiştir.

S.L, Yang (S.L, Yang, 2009) tarafından yapılan çalışmada sabit ters elektromanyetik problemler için küresel çözümlere ulaşma aracı olarak ABC algoritması önerilmiş ve test edilmiştir. Elde edilen sayısal sonuçlara göre ABC algoritmasının MPM (matrix pencil method) yöntemine rakip olduğu gösterilmiştir.

Alok (Alok, 2009) tarafından yayınlanan çalışmada yaprak kısıtlı minimum mesafeli ağaç (leaf-constrained minimum spanning tree) problemi olan LCMST problemi uygulama ile ABC algoritması arasında bir karşılaştırılma yapılmıştır. Yeni ABC yaklaşımı ile elde edilen sonuçlarda kısa sürede daha kaliteli sonuçlar elde edilmiştir.

Karaboğa (Karaboğa, 2009) tarafından yayınlanan makalede; IIR sayısal süzgeç tasarımı için ABC algoritması kullanılarak yeni bir yöntem tanımlanmış ve geleneksel olarak kullanılan (LSQ-nonlin) ve PSO algoritmaları ile başarımları karşılaştırılmıştır. Rapor edilen sonuçlara göre yapay arı kolonisi IIR filtresi tasarımı için alternatif bir yaklaşım olduğu belirlenmiştir.

Kang vd. (Kang, Li, Xu, 2009) tarafından yapay arı koloni algoritması (ABC) ile yalın Nelder-Mead yöntemi birleştirilmiş ve ters analiz problemleri için uygulanmıştır. Bu uygulama beton baraj sistemlerinin parametre tanımlamasında kullanılmıştır. Önerilen bu karma yaklaşımın başarımlarını görmek için iki örnek üzerinde

standart ABC ile gerçek kodlu GA karşılaştırılmıştır. Ele alınan problem için gerçek kodlu GA'nın standart ABC'den daha iyi başarımlar gösterdiği rapor edilmiştir.

Karaboğa ve Baştürk (Karaboğa ve Baştürk, 2008) tarafından yapılan çalışmada çok boyutlu matematiksel fonksiyon optimizasyonunda ABC algoritmasının performansını, Farksal Gelişim (DE), Parçacık Sürü Optimizasyonu (PSO) ve Evrimsel algoritma (EA) ile karşılaştırılmıştır. Karşılaştırılma sonucunda ABC algoritmasının performansının daha iyi olduğu sonucuna varılmıştır.

1.2. Tez Çalışması Kapsamı

Yukarıda kısaca ABC ile yapılan literatür çalışmaları özetlenmiştir. Yapılan özetten de anlaşılacağı üzere ABC algoritması dinamik sistemlerin bulanık modellenmesinde henüz kullanılmamıştır. Bu tez çalışmasında ABC algoritmasının dinamik sistem modelleme problemi için bulanık model (ANFIS) elde etmek amacıyla kullanımı üzerinde çalışılmıştır. Çalışmada ABC'nin başarımlarını bilinen diğer sezgisel arama algoritmaları ile kıyaslanmıştır. Bu çerçevede yapılan çalışmaların anlatıldığı bu tez kitabının organizasyonu şu şekildedir: İkinci bölümde bu çalışmada kullanılan rastgele arama tabanlı evrimsel algoritmalar (ABC, PSO, DE) tanımlanmış olup çalışmanın ana konusu olan ABC'nin işleyişi örnek matematiksel bir fonksiyonun optimizasyonu üzerinde detaylıca anlatılmıştır. 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 bal arılarının doğada yiyecek kaynağı bulurken nasıl bir yol izledikleri detaylı bir şekilde anlatılmış ve bu yapıyı modelleyen Yapay Arı Kolonisi (ABC) algoritması anlatılmıştır. Bunun yanında çalışmada karşılaştırmalar için kullandığımız iki önemli sezgisel arama algoritmaları hakkında da bilgi verilmiştir.

2.1. Yapay Arı Kolonisi Algoritması

2.1.1. Doğada arılar

Doğada bilinen en seçici türlerden biri olan bal arıları, en iyiyi bulmak için profesyonel bir arama süreci gerçekleştirmektedirler. Bu arama sürecinde kovadaki her arının yiyecek kaynaklarını araması, diğer arılarla iletişim, nektarların kovana getirilmesi gibi kendine ait görevleri vardır. Bu görevlere göre kovanda üç çeşit arı bulunmaktadır. Bunlar kraliçe arı, erkek ve işçi arılardır (Akay,2009).

2.2.2. Arıların yiyecek bulma davranışı ve salınım dansı

Arıların yaşama devam edebilmeleri için en önemli etken yiyecek kaynaklarıdır. Yiyecek kaynaklarına kovan etrafındaki nektar kaynakları da denilebilir. Bal arıları bu yiyecek kaynaklarına ulaşmada ve bu kaynakları kovana taşımada yuvaya yakınlık, zenginlik, enerji gibi. etkenler çok etkilidir. Bal arıları en az enerjiyle en kısa zamanda yiyecek kaynaklarını kovana taşımayı hedeflemektedirler. Çünkü bir nektar kaynağının niteliğini, nektar kaynağından getirilen nektarın harcanan enerjiye oranı belirler. Ayrıca kaynaklardan yiyecek getiren arıların diğer arıları bu kaynaklara yönlendirmek için kaynağın konum bilgisini diğer arılara iletmesi gerekir. Konumla ilgili bilgiyi alan arı bu hedefe ulaşmak için güneş ışığından faydalanır. Arılar yörüngeleri ile güneş arasındaki açıyı hesaplayabilmektedirler. Enerji tüketimine göre uzaklık belirleyen arılar, yüklerine göre farklı yükseklikte uçarak enerjilerini ayarlamaktadırlar (Akay,2009).

Tereshko, arı kolonisinin yiyecek arama davranışını reaksiyon difüzyon denklemlerine dayalı bir modelle tanımlamış ve kovanda yiyecek arama davranışında rol alan haberleşmenin nasıl gerçekleştiğini incelemiştir (Tereshko,2005). Tereshko'nun reaktif difüzyon denklemlerine dayalı olarak önerdiği sürü zekâsının ortaya çıkmasını

sağlayan minimal yiyecek arama modelinde üç temel bileşen vardır. Bunlar; Yiyecek kaynakları, görevi belirli işçi arılar ve görevi belirsiz işçi arılar (Kâşif ve gözcü arılar)dır (Tereshko ve Loengarov, 2005). Bu model iki modda çalışmaktadır. Bir yiyecek kaynağına yönelme ve kaynağı bırakma (Tereshko ve Loengarov, 2000). Akay çalışmasında bu öğeleri aşağıdaki gibi açıklamıştır (Akay, 2009).

Yiyecek kaynakları: Arıların yiyecek aramak için gittiği kaynaklardır. Bir yiyecek kaynağının değeri, kaynağın çeşidi, yuvaya olan uzaklığı, nektar miktarı veya nektarın çıkarılmasının kolaylığı gibi birçok faktöre bağlı olmasına rağmen basitlik açısından sadece kaynağın zenginliği tek bir kriter olarak alınabilir.

Görevi belirli işçi arılar: Bu işçi arılar daha önceden belirlenmiş kaynaklardan toplanan yiyeceğin kovana getirilmesi işi ile görevlendirilmişlerdir. Bir diğer görevi ise gittikleri kaynağın konum ve kalite bilgilerini kovandaki diğer arılarla dans ederek paylaşmaktır.

Görevi belirli olmayan işçi arılar: Bu arılar nektar toplanabilecek kaynakları arama eğilimindedirler. Görevi belli olmayan iki çeşit işçi arı vardır. Bunlar rastgele kaynak arayan kâşif arılar, kovanda bekleyen ve görevli arıları izleyerek bu arılar tarafından gelen bilgiye göre yeni kaynaklara yönelen gözcü arılardır.

Arılar arasında bilgi paylaşımı kolektif bilginin oluşumunda en önemli husustur. Bir kovan göz önüne alındığında, tüm kovanlarda ortak olan bazı bölümlere ayırmak mümkündür ve bu bölümlerden en önemlisi ise dans alanıdır. Yiyecek kaynağının kalitesi ve yeri ilgili bilgi paylaşımı dans alanında olmaktadır. Bir arı dans ederken diğer arılarda ona antenleri ile dokunur ve bulduğu kaynağın tadı ve kokusu ile ilgili bilgi alırlar. Ziyaret ettikleri kaynağa daha fazla arı yönlendirebilmek için kovandaki çeşitli alanlarda bu dansı gerçekleştirir ve kaynağa geri döner (Akay,2009).

Sato ve Hagiwara, arıların davranışlarına dayanan **bee system** olarak adlandırdıkları GA geliştirmişlerdir (Sato,Hagiwara, 1997). Bu modele göre bir arı yiyecek kaynağı ile ilgili bilgiyi bir dans aracıyla çalıştığı arılara iletmektedir. Grup zekâsının en güzel örneklerinden birini sergileyen bal arıları, en iyiye an az enerjiyi sarf ederek ulaşmayı amaçlamaktadırlar. Hedeflerine ulaşabilmek için tüm arılar ortak karar

vermektedirler. Bu karar verme sürecinde en önemli etken arı dansı denilen salınım dansıdır. Bu dansa kaynağın kovana uzaklığı, doğrultusu, zenginliği gibi gerekli her türlü bilgi gizlidir. Bu dans bilginin niteliğine göre değişmektedir. Örneğin “daire dansı” kaynağın kovana 15 metreden daha yakın mesafede olduğunu gösteriyor. Arılar 25-100 metre arasındaki besin kaynakları için de “sallanma dansı”nı, kovana 100 metreden daha uzak kaynaklar içinse kaynağın uzaklığını, koordinatlarını ve niteliğini bildiren “kuyruk dansı” ile birbiriyle iletişim kurmaktadır (Grüter ve Farina, 2009).

Tüm zengin kaynaklarla ilgili bilgiler dans alanında gözcü arılara iletiildiğinden, gözcü arılar birkaç dansı izledikten sonra hangisini tercih edeceğine karar verir. Zengin kaynaklarla ilgili daha fazla bilgi aktarımı olduğundan bu kaynakların seçilme olasılığı daha fazladır (Akay,2009).

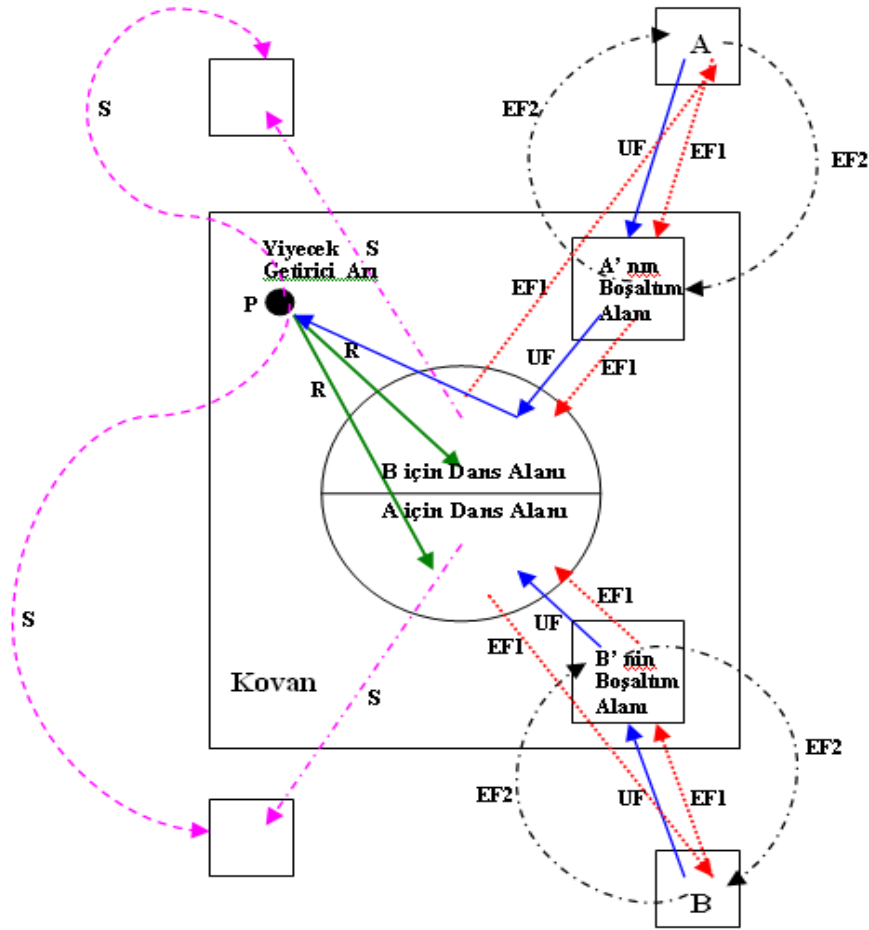
Yiyecek arayıcıların davranışları Şekil 2.1’de özetlenmiştir. Bu davranışların şekil üzerinde açıklamalarını Akay şöyle yapmıştır: A ve B bulunmuş kaynaklar olarak varsayılmıştır. Başlangıçta görevi belli olmayan ve kaynak bilgisinden yoksun arı kaynak aramaya başlayacaktır. Bu arı için iki durum mevcuttur: İlki; S ile gösterilen kâşif arı olabilir ve yiyecek aramaya başlayabilir İkincisi; Dans eden arıları izleyerek tarif edilen kaynaklara giden R ile gösterilen bir gözcü arı olabilir.

Kaynaklara giden arılar buldukları kaynaklardan nektar getirmeye başlarlar. Böylece bu arılar görevli bir arı haline gelmişlerdir. Nektarı kovana getiren arı için bundan sonra üç seçenek vardır (Tokmak, 2011):

i) Bilgi paylaşımında bulunmadan kaynaktan nektar getirmeye devam edebilir. Bu arı EF2 ile gösterilmiştir.

ii) Kaynağa dönmeden önce dans ederek nektar kaynağının yeri ve miktarı hakkında diğer arılara bilgi paylaşımında bulunarak diğer arıları bu kaynağa yönlendirebilir. Bu arı EF1 ile gösterilmiştir.

iii) Kaynağı terk ederek dans alanında gözcü arı olabilir. Bu arı UF ile gösterilmiştir.



Şekil 2. 1. Arıların yiyecek arama davranışı(Akay,2009)

2.2. Yapay Arı Kolonisi (ABC) ve Temel Adımları

Doğadan esinlenen algoritmaların sürü zekâsı yaklaşımları, böceklerin içgüdüsel problem çözme becerilerini kullanan etkili metasezgisel yöntemlerdir. Bu yöntemlerin içinde bal arılarının yiyecek arama davranışını modelleyen Yapay Arı Kolonisi algoritması güncel bir metasezgisel algoritmadır. Karaboğa, arıların yiyecek arama davranışını modelleyerek Yapay Arı Kolonisi (Artificial Bee Colony, ABC) algoritmasını geliştirmiştir (Karaboğa,2005).

ABC optimizasyon algoritması en fazla nektara sahip kaynağın yerini bulmaya çalışarak uzaydaki çözümlerden problemin çözümünü bulmaya çalışmaktadır (Akay,2009). ABC algoritmasının temel adımları aşağıdaki gibi özetlenebilir:

- 1:Başlangıç yiyecek kaynağı bölgelerinin üretilmesi
- 2: Tekrarla çevrim sayısı=maksimum nesil sayısı oluncaya değin.
- 3:İşçi arıların yiyecek kaynağı bölgelerine gönderilmesi
- 4:Olasılıksal seleksiyonda kullanılacak olasılık değerlerinin görevli arılardan gelen bilgiye göre hesaplanması
- 5: Gözcü arıların olasılık değerlerine göre yiyecek kaynağı bölgesi seçmeleri
- 6:Kaynağı Bırakma kriteri: Limit ve Kâşif Arı üretimi
- 7: Git 2

2.2.1.Başlangıç yiyecek kaynağı bölgelerinin üretilmesi

Algoritma E.2.1 ile rasgele yiyecek kaynakları üretmekle başlar.

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

E.2.1’de $i=1.....SN$, SN : Yiyecek kaynağı sayısı, $J=1.....D$, D : optimize edilecek parametre sayısını göstermektedir. x_j^{min} : j. parametrenin alt sınır, x_j^{max} : j. parametrenin üst sınır değerini göstermektedir.

Algoritma başlangıcında her kaynağa ait geliştirilememe sayacı $failure(i)$ sıfırlanmaktadır. Maksimum çevrim sayısı (MCN) veya tanımlanmış bir durdurma kriteri, algoritmanın durdurma kriteri olarak kullanılabilir (Karaboga, 2008; Akay, 2009).

2.2.2.İşçi arıların yiyecek kaynağı bölgelerine gönderilmesi

Her bir kaynağın bir görevli arısı vardır. Dolayısıyla yiyecek kaynaklarının sayısı görevli arıların sayısına eşittir. İşçi arı çalıştığı yiyecek kaynağı komşuluğunda yeni bir yiyecek kaynağı belirler ve bunun kalitesini değerlendirir. Bulunan yeni kaynak eskisinden daha iyi ise bunu hafızaya alır. Bu davranış E.2.2’deki denklemde verilmiştir (Akay, 2009).

$$V_{ij} = x_{ij} + \theta_{ij} * (x_{ij} - x_{kj}) \quad (E.2.2)$$

Eşitlik ile x_i gösterilen her bir yiyecek kaynağı için $[1,D]$ sınırları dikkate alınarak rasgele seçilen j parametresinin değiştirilmesi ile yeni V_i kaynağı bulunur. Bu işlem yine rasgele seçilen x_k komşu çözümünün ($k \in \{1,2 \dots SN\}$), j parametresinin farkı alınarak $[-1,1]$ aralığında rasgele seçilen θ sayısı ile ağırlandırıldıktan sonra mevcut kaynağın j parametresine eklenmekle gerçekleşir (Akay,2009). E.2.2'deki x_{ij} ile x_{kj} arasında farkla yeni aday çözüm üretilir. V_{ij} belirlenen parametre sınırları içine E.2.3 ile çekilir.

$$v_{ij} = \begin{cases} x_j^{min}, v_{ij} < x_j^{min} \\ v_{ij}, x_j^{min} \leq v_{ij} \leq x_j^{max} \\ x_j^{max}, v_{ij} > x_j^{max} \end{cases} \quad (E.2.3)$$

Sınırlar dâhilinde üretilen yeni v_i aday çözümün uygunluğu hesaplanarak bir uygunluk değeri atanmaktadır. Bunun içinde E.2.4 kullanılır. Burada f_i , v_i kaynağının yani çözümün maliyet değeridir. Bulunan uygunluk değerine göre aç gözlü (greedy) seçme işlemi uygulanarak yeni aday çözümün sürüye katılıp katılmayacağı belirlenir.

$$fitness_i = \begin{cases} \frac{1}{1+f_i} & f_i \geq 0 \\ 1 + abs(f_i) & f_i \leq 0 \end{cases} \quad (E.2.4)$$

Bulunan v_i çözümünün uygunluk değeri mevcut x_i çözümünün uygunluk değerinden daha nitelikli ise eski kaynak bilgilerinden vazgeçerek hafızasına yeni v_i çözümü sürüye katılır ve onun için geliştirememeye sayacı ($failure(i)$) sıfırlanır. Eğer bulunan çözüm eskisinden daha nitelikli değilse eski kaynağına dönülür ve geliştirememeye sayacını bir artırılır (Akay,2009).

2.2.3.Gözcü arıların seleksiyonda kullanacakları olasılık değerlerinin hesaplanması

Tüm görevli arılar buldukları kaynakla ilgili bilgiyi gözcü arılarla paylaşırlar. Gözcü arı ise aldığı bilgiler doğrultusunda nektar kaynağının miktarı ile ilişkili olarak bir kaynak seçer. Algoritmanın bu noktasında uygunluk değerlerine bağlı olan seçim işlemi rulet çarkı kullanılarak yapılır. Bir kaynağın uygunluk değerinin, diğer tüm

kaynakların uygunluk değeri toplamına (E.2.5) i. kaynağının oranı olasılık değeri olarak belirlenmiştir (Akay, 2009).

$$p_i = fitness_i / \sum_{i=1}^{sn} fitness_i \quad (E.2.5)$$

2.2.4. Gözcü arıların yiyecek kaynağı bölgesi seçmeleri

Her bir kaynak için $[0,1]$ aralığında rastgele bir sayı üretilerek bu sayı ile p_i karşılaştırılır. Bu karşılaştırılmada p_i değeri üretilen sayıdan büyük ise yeni çözüm üretilir ve kalitesi hesaplanır. Açgözlü seçimde seçilebilen yeni çözüm için çözüm geliştirememeye sayacı *failure* sıfırlanır. Aksi durumda gözcü arının *failure* sayacı bir artırılır.

2.2.5. Kaynağı bırakma kriteri: Limit ve kâşif arı üretimi

Bir çevrim sonucunda tüm işçi ve gözcü arılar arama süreçlerini tamamladıktan sonra kaynağın bitip bitmediğini “*failure*” adı verilen çözüm geliştirememeye sayacı belirler. Bir kaynak için çözüm geliştirememeye sayacı belli bir eşik değerinin üstündeyse, bu kaynak tükenmiş anlamına gelir ve işçi arının kendine yeni bir kaynak araması gerekmektedir. Kaynağın tükendiğini belirleyen eşik değeri ABC algoritması için en önemli parametresidir ve “limit” olarak adlandırılmaktadır (Akay,2009).

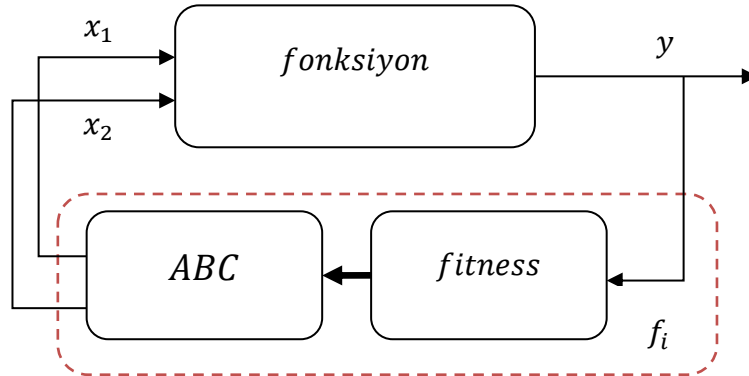
ABC algoritmasının kontrol parametreleri ile ilgili literatürde tavsiye edilen değerler şu şekildedir: Koloni büyüklüğü için genellikle 20-50 arası bir değer, limit için ise parametre sayısı (D) ve koloni büyüklüğü’nün (SN) çarpımı (SN*D) civarında bir değer önerilmektedir (Karaboğa, 2011).

2.2.6. ABC algoritması ile fonksiyon optimizasyon örneği

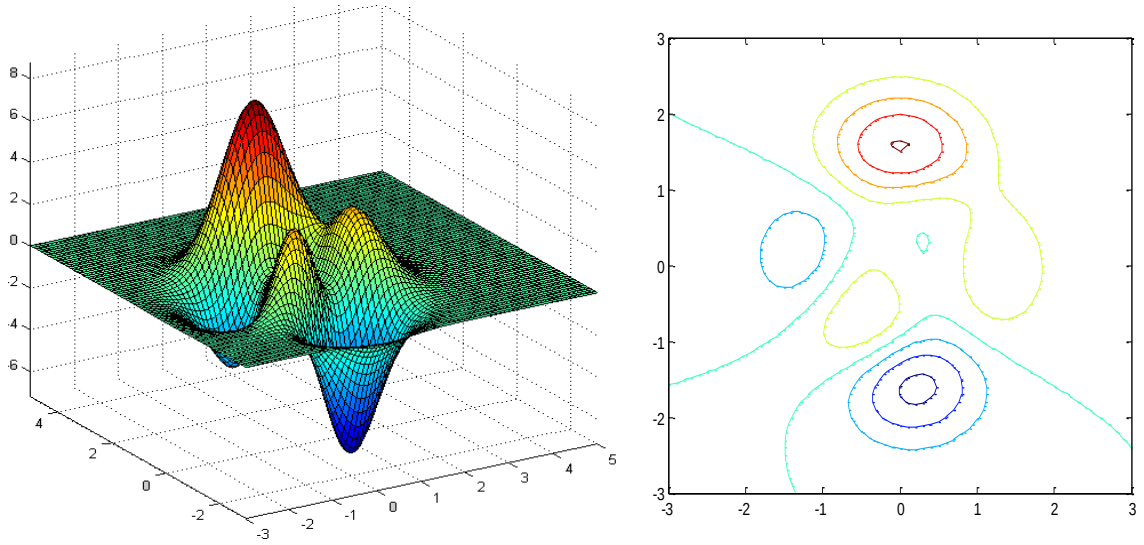
ABC algoritmasının çalışmasını anlamak ve başarısını test edebilmek için matematiksel ifadesi E.2.6’da verilen Şekil 1’den de görüleceği üzere birkaç yerel maksimumu ve minimumu bulunan “*peaks*” fonksiyonu üzerinde çalışılmış, algoritmaya küresel minimum buldurulmuştur.

$$\gamma = 3(1-x_1)^2 e^{-(x_1^2-(x_2+1)^2)} - 10\left(\frac{x_1}{5} - x_1^3 - x_2^5\right) e^{-(x_1^2+x_2^2)} - \frac{1}{3} e^{-((x_1+1)^2+x_2^2)} \quad (\text{E.2.6})$$

ABC ile yapılan fonksiyon optimizasyonu blok yapısı Şekil 2.2’de verilmiştir. Blok yapıdan da görüleceği üzere ABC fonksiyon için uygun girişleri belirlemektedir. Bu çalışmada fonksiyonun küresel minimumunu bulmak hedeflenmiştir. Bu bakımdan ABC’nin bireyleri fonksiyonun aday girişleridir. Algoritma işletilirken ABC her nesilde tüm bireylerini fonksiyona gönderip fonksiyon çıkışı hesaplanarak bireylerin uygunluk değerleri belirlenir. ABC’nin işleyişi sırasında her birey uygunluk değerleri baz alınarak her nesilde geliştirilmiş olur. Sonuçta bireyler fonksiyonun global minimumunu verecek giriş değerlerine doğru sürüklenirler. Ek-1’de bu uygulamada kullanılan yazılım kodları verilmiştir.



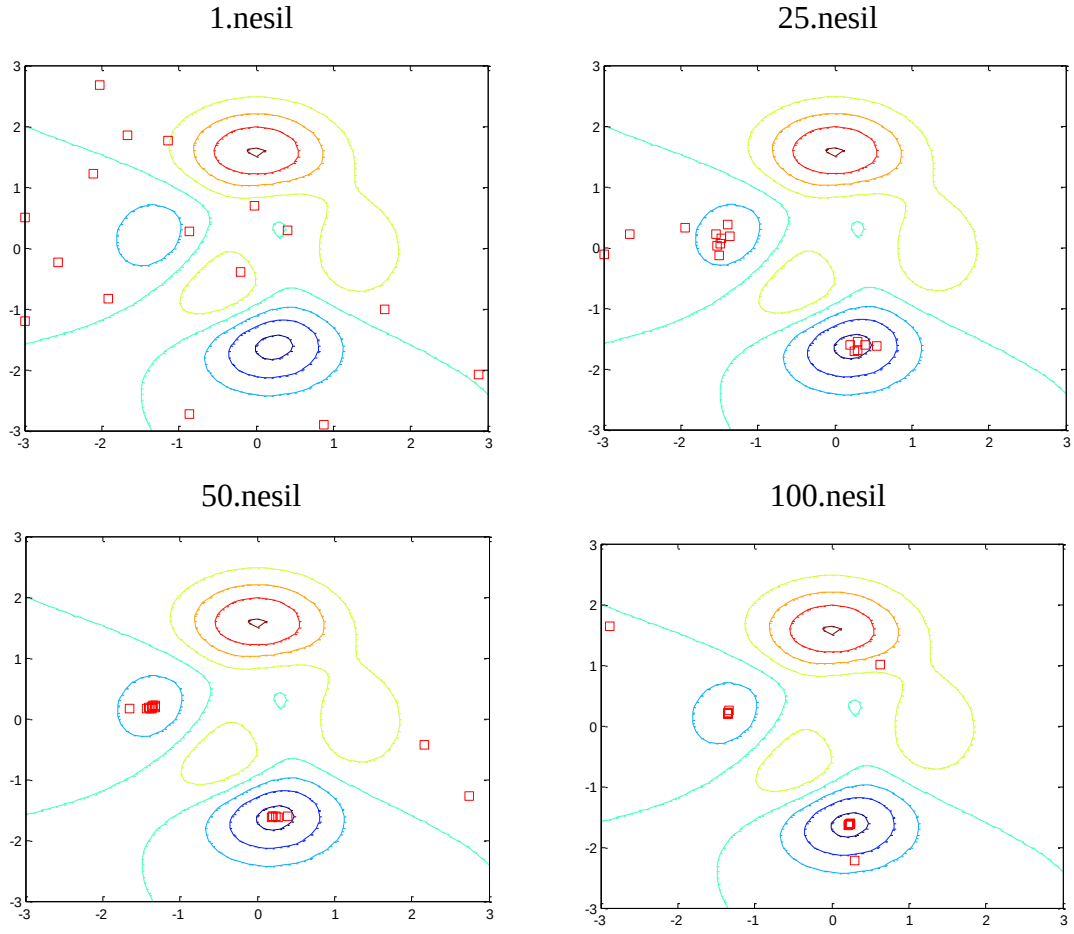
Şekil 2. 2. ABC ile fonksiyon optimizasyonu blok yapısı



Şekil 2. 3. “Peaks” fonksiyonun tanımladığı yüzey

ABC algoritması kullanılarak yapılan fonksiyon optimizasyonunda; koloni büyüklüğü 50, maksimum çevrim sayısı 100, limit 30 alınmıştır. Matematiksel ifadenin ABC algoritmasında kullanılan $ari_i = [x_{1i} \ x_{2i}]$ birey yapısını göstermektedir. Buna göre matematiksel fonksiyon için elde edilen her uygunluk değeri $f_i = \gamma(x_{1i}, x_{2i})$ şeklinde ifade edilmektedir. Algoritma 100 kez ayrı ayrı çalıştırıldığında bulduğu çözümlerin ortalamasının $x_1=0.2284$, $x_2=-1.6253$ olduğu gözlenmiş ve bu giriş değerlerine karşılık fonksiyonun $y=-6,5511$ sonucunu ürettiği belirlenmiştir. Bu değer Şekil 2. 3’den de anlaşılacağı üzere fonksiyon yüzeyindeki küresel minimumdur.

Şekil 2.3’de algoritmanın çalışmasının muhtelif adımlarında arıların ilgili uzaydaki pozisyonları görülmektedir. Şekilden arıların başlangıçta uzayda rasgele dağılmış olmalarına rağmen en son algortima adımında küresel minimuma çok yakın civarda toplandıkları görülmektedir.



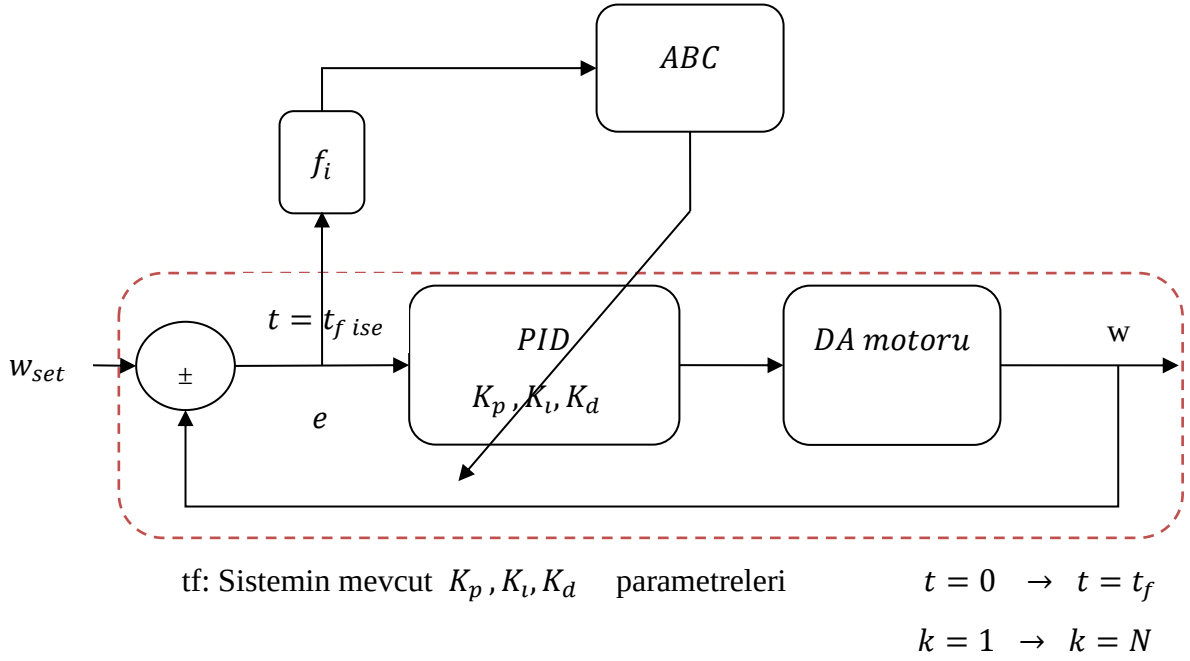
Şekil 2. 4. Algoritmanın çeşitli nesil sayılarında arıların arama uzayı konumları

2.2.7.ABC algoritması ile PID denetleyici parametre optimizasyonu örneği

Bu kısımda ABC algoritması kullanılarak doğru akım (DA) motoru hız kontrolü için PID parametrelerinin (K_p , K_d ve K_i) optimizasyonu gerçekleştirilmiştir. Şekil 2.5 'te optimizasyon için kullanılan blok model yapısı görülmektedir. Modelde kullanılan t_f : Sistemin mevcut K_p, K_i, K_d parametre için verilmiştir. K_p, K_i, K_d en uygun değeri bulabilmek için

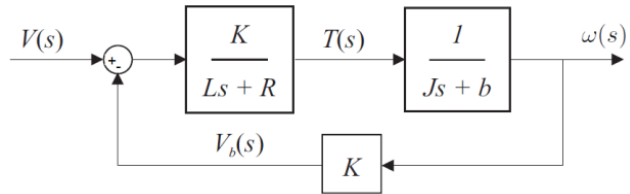
$$f_i = \frac{1}{N} \sum_{i=1}^N e_i^2 \quad (E.2.7)$$

eşitliği kullanılır.



Şekil 2. 5. ABC ile PID denetleç parametre optimizasyonu blok yapısı

Uygulamada kullanılan DA motorun modeli Şekil 2.4’de, motora ait parametreler ise Çizelge 2.1’de verilmiştir.



Şekil 2. 6. DA motor modeli öbek şeması (Babuska,1999)

Çizelge 2. 1. DA motoru parametreleri

p	93.3 kW	Hız	1150 devir/dk
V_a	210 V	R_a	0.0125 Ω
L_a	1.2 mH	B	1.89 Nm/rad/s
J	2.83 kg.m ²	K_b, K_d	1.78

ABC algoritması ile yapılan optimizasyonda; koloni büyüklüğü 20, maksimum çevrim sayısı 100, limit değeri 15 olarak alınmıştır. (limit=cicek_s*parametre_sayısı)

PID parametrelerinin optimizasyonu için sezgisel algoritmalarının bireyleri uygunluk değeri (E.2.7)'de verilen ortalama karesel hata (mse) ölçütü kullanılmıştır. Eşitlikte N bireylerin belirlediği değerlerle işleyen PID denetleç denetimi altında sayısal sistem çözümünde kullanılan toplam adım sayısını, ω_{set} motor açısal hız önayar (set) değeri, ω_a ise sistem çözümünde elde edilen açısal hız değerini ve k örnekleme indeksini göstermektedir.

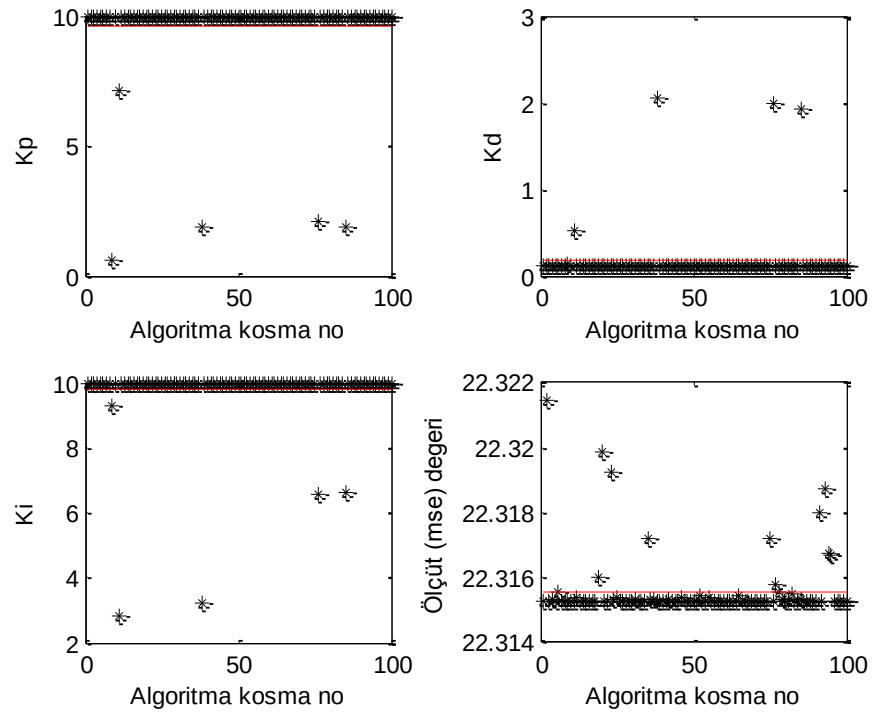
$$f_i = \frac{1}{N} \sum_{k=1}^N (\omega_{set}(k) - \omega_a(k))^2 \quad (E.2.8)$$

PID parametrelerinin sınır değerleri (E.2.8)'de verildiği biçimde kullanılmıştır.

$$0 \leq K_p \leq 10, \quad 0 \leq K_d \leq 3, \quad 0 \leq K_i \leq 10 \quad (E.2.9)$$

Algoritma MATLAB programlama dilinde kodlanarak Intel(R) Core(TM) i7, Q 740 @ 1.73 GHz CPU, 6 GB RAM donanımlı, 64 bit işletim sistemine sahip bir diz üstü bilgisayarda çalıştırılmıştır.

ABC algoritmasının başarımını ölçebilmek için algoritma 100 defa çalıştırılmış, her çalıştırma sonucu bulunan PID parametrelerinin ortalama değerlerinin sırasıyla $K_p=9.6366$, $K_d= 0.1194$, $K_i =9.7851$ olduğu bulunmuştur. Şekil 2.5'de 100 ayrı çalışma sonunda bulunan parametreler ve ölçüt değerleri verilmiştir. Görüleceği üzere algoritma %95 oranında hep aynı parametre değerlerini çözüm olarak işaret etmiştir. (Karakuzu vd., 2012)'de aynı problem için ABC'nin başarımı GA, TACO, PSO ve DE ile karşılaştırılmalı olarak incelenmiş olup, ABC'nin en anlamlı rakiplerinin PSO ve DE olduğu bildirilmiştir. Bu çalışmanın sonraki kısımlarında ABC'nin kıyaslamalı başarımı PSO ve DE algoritmaları kullanılmıştır.



Şekil 2. 7. ABC algoritmasının 100 kez koşturulması sonucu bulunan sonuçlar

2.4. Parçacık Sürü Optimizasyonu (PSO, Partical Swarm Optimization)

PSO 1995 yılında J.Kennedy ve R.C.Eberhart tarafından kuş sürülerinin davranışlarından esinlenilerek geliştirilmiş populasyon tabanlı sezgisel optimizasyon tekniğidir (Kennedy and Eberhart, 1995). Algoritma, rasgele atanmış bireylerinin (parçacık) ilgili arama uzayında kendilerinin ve sürüdeki tüm parçacıkların deneyimlerine dayalı olarak konumlarını her nesilde güncelleyerek optimum sonucu bulmaya çalışmaktadır. Algoritma temel olarak aşağıdaki işlem basamaklarından oluşur;

1. Rasgele üretilen başlangıç parçacıkları ile başlangıç sürüsü oluşturulur.
2. Sürü içerisindeki tüm parçacıkların uygunluk değerleri hesaplanır.
3. Her bir parçacık için mevcut nesilde yerel en iyi parçacık (pbest) bulunur.
4. Mevcut nesile kadar bulunan yerel eniyiler içerisinde küresel en iyi parçacık (gbest) belirlenir.
5. Parçacık hızları ve parçacıklar güncellenir.
6. Durdurma kriteri sağlanıncaya kadar 2,3,4,5 adımları tekrar edilir.

Bu çalışmada parçacık hızlarının güncellenmesi için algoritmanın yalın hallerinden biri olan E.2.10'da verilen yöntem (Clerc and Kennedy, 2002) ve geliştirilmiş şekli olan E.2.11'de (Çavuşlu vd.,2012) verilen yöntem kullanılmıştır.

$$v_i(n) = \xi [v_i(n-1) + c_1 r_1 (p_{i,lbest} - p_i(n)) + c_2 r_2 (p_{gbest} - p_i(n))] \quad (E.2.10)$$

$$v_i(n) = \xi [v_i(n-1) + c_1 r_1 (p_{i,lbest} - p_i(n)) + c_2 r_2 (p_{gbest} - p_i(n))] + \alpha \lambda(n) \quad (E.2.11)$$

2.5. Farksal Gelişim (DE, Differential Evolution) :

Farksal Gelişim algoritması Price ve Storn tarafından 1995 yılında geliştirilmiş, özellikle sürekli verilerin söz konusu olduğu problemlerde etkin sonuçlar verebilen, işleyiş ve operatörleri itibarıyla genetik algoritmaya dayanan popülasyon temelli sezgisel bir optimizasyon tekniğidir (Price, 1996). Temel olarak, DE algoritması popülasyon içerisinde rastgele seçilen iki bireyin ağırlık farkının üçüncü bir bireye eklenmesi mantığına dayanmaktadır.

DE algoritması 3 önemli parametreye sahiptir: skala faktörü (SF), çaprazlama olasılık sabiti (CR) ve popülasyon boyu (PS). Bir popülasyon PS kadar bireyi içerir. Skala faktörü (SF) çaprazlama olasılık sabitine (CR) göre daha hassastır (Price,1997). DE algoritmasının başında PS optimizasyon parametrelerine göre belirlenir ve optimizasyon sürecinde değiştirilmez. Normal şartlarda popülasyon boyu (PS) optimizasyon parametre sayısının 5-10 katı olarak seçilir. Başlangıç popülasyonu rastgele seçilir. DE algoritması 3 temel operatöre sahiptir: mutasyon, çaprazlama ve seçim (Price,1999). Mutasyon ve çaprazlama operatörleri yeni bireyler üretir ve seçim operatörü ile uygun olanlar belirlenir, bu şekilde popülasyonda sürekli en iyi bireylerin bulunması sağlanır.

Bu çalışma kapsamında pratikte en sık kullanılan DE stratejisi olan rastgele, en iyi bireyler / 1 fark vektörü / binom çaprazlaması olarak kısaca adlandırılan strateji kullanılmıştır. De'nin işleyişi E.2.12 ve E.2.13 ile özetlenebilir.

$$v_{i,g+1} = x_{i,g} + SF(x_{b,g} - x_{i,g}) + SF(x_{r_1,g} - x_{r_2,g}) \quad (E.2.12)$$

$$u_{i,g+1} = \begin{cases} v_{i,g+1}, & \text{if } r \leq CR \\ x_{i,g}, & \text{if } r > CR \end{cases} \quad (E.2.13)$$

Bu denklemlerde $r_1, r_2, r_3 \in [1, PS]$ rastgele üretilen katsayılarıdır. $x_{b,g}$ popülasyon içerisindeki en iyi uygunluk değerine sahip bireyi, $v_{i,g}$ mutasyona uğramış bireyi, $u_{i,g}$ deneme bireylerini göstermektedir. Rastgele katsayılar birbirinden farklı olmak zorundadır. Skala faktörü (SF) sabit bir değerdir ve $0 \leq SF \leq 2$ olarak değişir. Skala faktörü rastgele seçilen bireylerin fark vektörlerinin ağırlıklandırılmasında kullanılır.

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

Bu bölümde tez çalışmasında ABC algoritmasının bulanık sistem modellemede başarımını incelemek için kullanılan örnek dinamik sistemler, bulanık çıkarım sistemi ve sezgisel algoritmalar kısaca tanıtılmıştır.

3.1. Örnek Dinamik Sistemler

Bu çalışmada ABC algoritmasının bulanık sistem modellemede başarımını incelemek amaçlı kullanılmak üzere Çizelge 3.1’de verilen ve literatürde sıkça kullanılan dinamik sistemler kullanılmıştır.

Çizelge 3. 1’de uygulamada kullanılacak olan örnek dinamik sistemler ve bu sistemlerin girişlerinde kullanılan $u(k)$ dizileri verilmiştir. Uygulamada örnek dinamik sistemler modellenirken Şekil 3.1’de verilen ANFIS yapısı kullanılmıştır. Her bir örnek dinamik sistem için ANFIS yapısı oluşturulurken Çizelge 3. 2’de tanımlanan ANFIS dikkate alınmıştır. Her bir sistemin girişleri belirlendikten sonra, her girişe ait 2’şer adet “Gauss” üyelik fonksiyonu (ÜF) ve kural sayıları tanımlanmıştır.

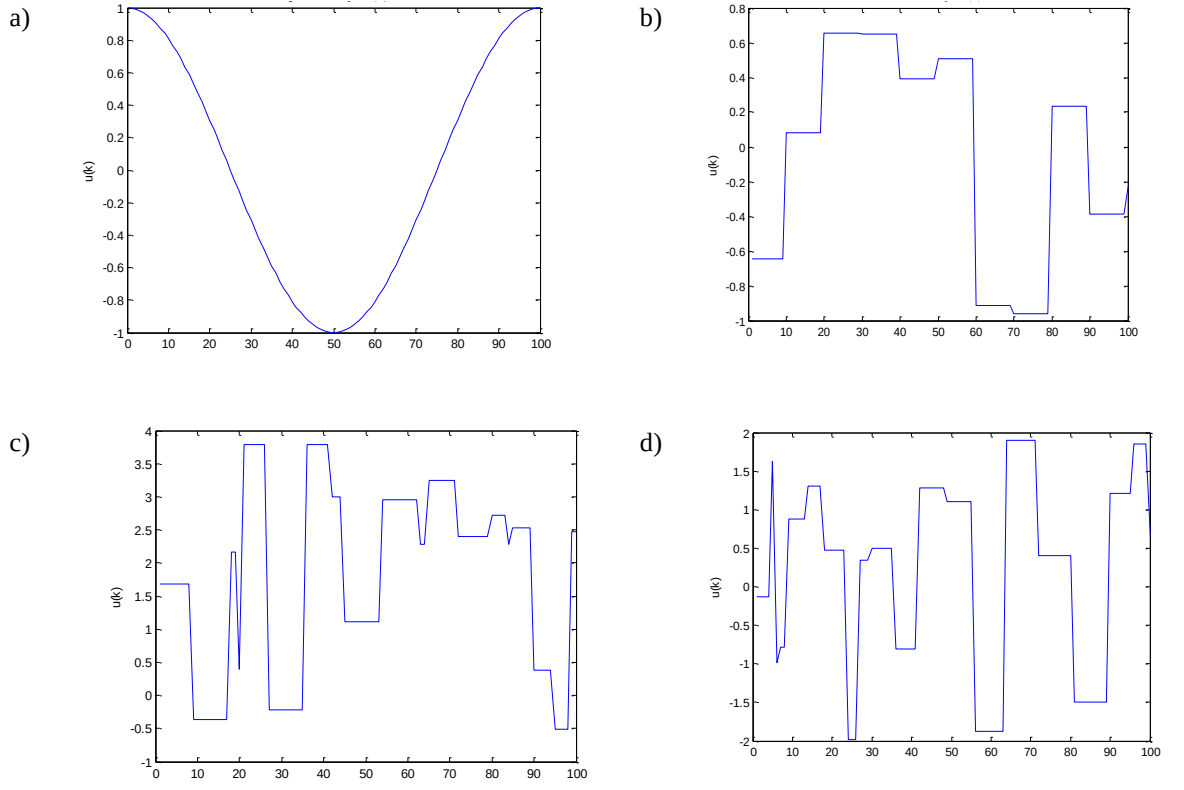
Örnek dinamik sistemlerin modellenmesi aşamasında öncelikle eğitim veri seti hazırlanmış ve kullanılmıştır. Eğitim seti kullanılarak sezgisel algoritmaların koşturulması elde edilen sonuçlar kaydedilmiştir. Tüm sistemler için eğitim tamamlandıktan sonra eğitimlerin başarısını daha iyi test edebilmek için, her bir eğitilen sistem eğitim setindekinden farklı olarak rasgele oluşturulan $u(k)$ dizileri kullanılarak ayrı ayrı test seti hazırlanmıştır. Test setleri yardımıyla eğitilen ANFIS sistem modelleri test edilmiştir. ÖDS sistemleri için Şekil 3.1’de eğitim fazında kullanılan, Şekil 3. 2’de test fazında kullanılan giriş dizileri ($u(k)$) verilmiştir.

Çizelge 3. 1. ABC, PSO ve DE öğrenmeli bulanık mantık tabanlı dinamik sistem tanıma/modelleme için literatürden seçilen örnek dinamik sistemler (ÖDS)

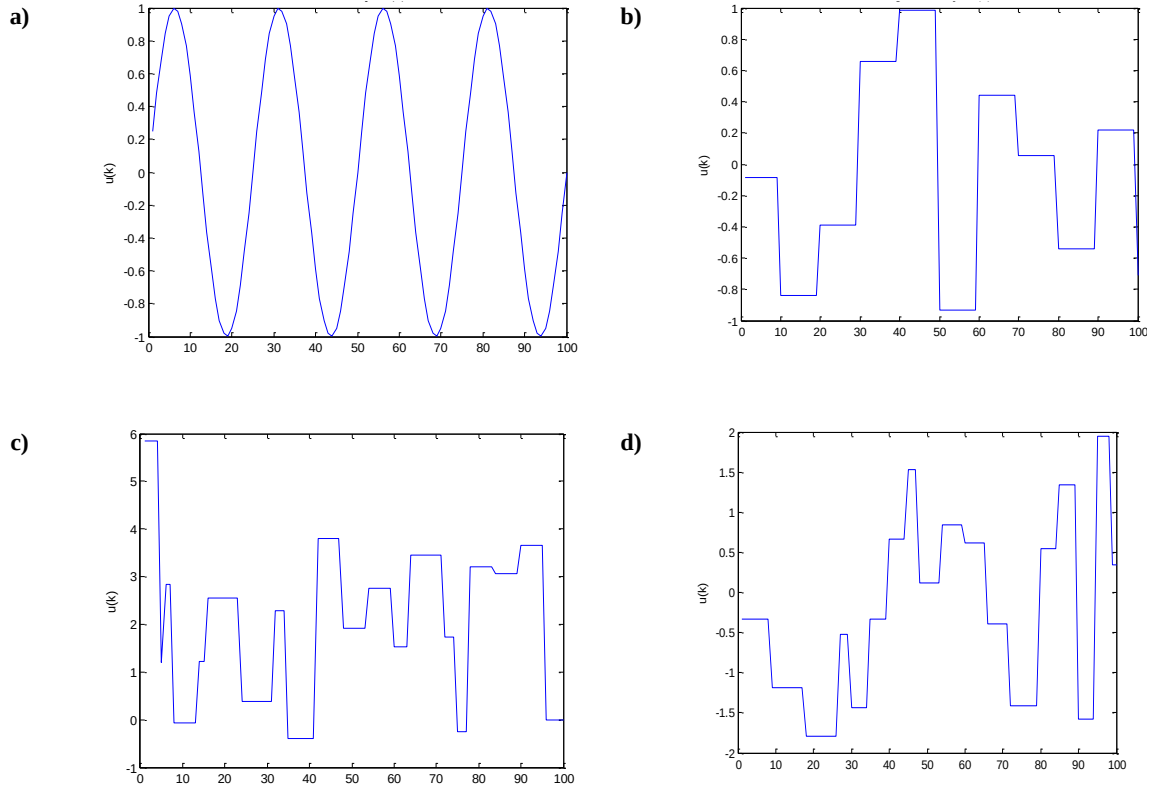
Örnek Dinamik Sistemler		Eğitim Seti	Test Seti
1	$y(k) = \frac{y(k-1) \cdot y(k-2) \cdot (y(k-1) + 2.5)}{1 + y^2(k-1) + y^2(k-2)} + u(k),$ (Narendra,1990)	$u(k) = \cos \frac{2\pi k}{100}$	$u(k) = \sin \frac{2\pi k}{25}$
2	$y(k+1) = \frac{y(k)}{1 + y^2(k)} + u^3(k), \quad (\text{Narendra,1990})$	$u(k)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

Çizelge 3. 2. Her bir ÖDS için kullanılan ANFIS yapısı

ÖDS No	Girişler	Giriş ÜF sayıları	Kural sayısı	Ayarlanacak 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



Şekil 3. 1.ÖDS sistemleri için sistem tanımının 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

Bulanık mantık; Küme teorisine dayanan matematiksel bir disiplindir ve klasik küme gösteriminin genişletilmesidir. Bulanık mantığın ve bu mantık kurallarını kullanan bulanık küme teorisinin Lotfi A. Zadeh tarafından geliştirilip 1965 tarihli orijinal makalesinde (Zadeh,1965) yayınlanmasından sonra belirsizlik içeren sistemlerin incelenmesi yeni bir boyut kazanmıştır.

Bulanık sistemler, bulanık küme teorisi, bulanık if-then kuralları ve bulanık mantığa dayalı popüler bir hesaplama yapısına sahiptir. Bulanık mantık ile sistem modellemesinde; bulanıklaştırma, kural çıkarma ve durulama olmak üzere üç temel

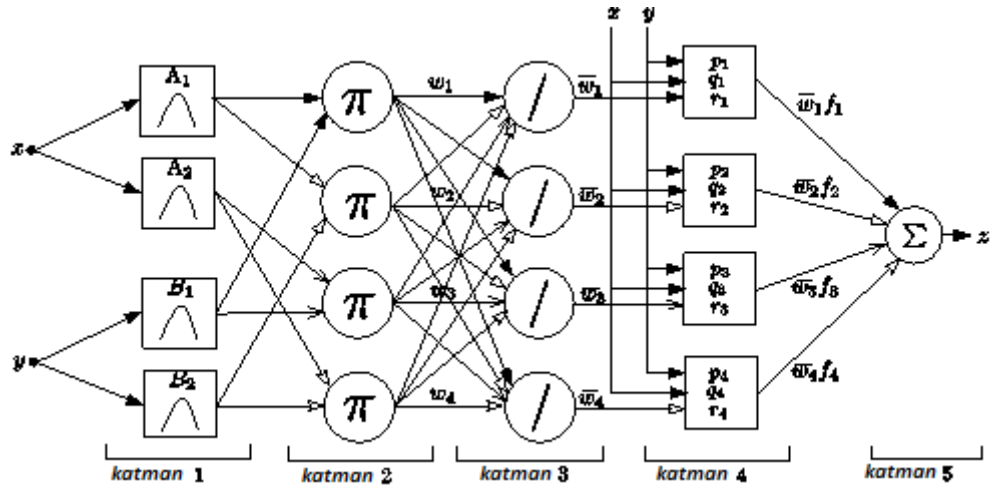
adım vardır. Bulanıklaştırma aşamasında, girişler bulanıklaştırılır. Yani sistem girişlerine uygun giriş üyelik fonksiyonları belirlenir ve “if-then” kuralları kullanılarak bulanık kümelerinin diğerleriyle ilişkilendirilmesi sağlanır. Durulama ile bulanık çıkış değerlerinin kesin bir çıkış değerine dönüştürülmesi sağlanır (Şenol, Yıldırım, 2008).

Sistem modellemek için bulanık girişimli çeşitli bulanık modeller vardır. Bunlardan en çok kullanılanları; Mamdani, Sugeno ve Tsukamoto bulanık modelleridir. Mamdani bulanık sistem modeli en basit model olup diğer modellerin temelini oluşturmaktadır. Mamdani bulanık mantık modelinin bir uyarlaması olan Sugeno bulanık mantık modeli ilk kez 1985 yılında kullanılmaya başlanmıştır.

Bulanık sistemlerin etkinliğinin artırılmasında, uyarlama tekniğinin katkısını sağlamaya yönelik çeşitli yöntemler geliştirilmiştir. Bunlardan biri de 1993’de Jang tarafından geliştirilmiştir (Jang,1993).

ANFIS Sugeno tipi bulanık sistemlerin yapısına sahip, her biri belli bir işlevi gerçekleştirmek üzere tasarlanmış 5 katmanlı bir yapıya sahiptir.

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



Şekil 3. 3. İki girişli-tek çıkışlı birinci dereceden kural polinomlu ANFIS mimarisi

Bu sistemin parametreleri 1. katmandaki üyelik fonksiyonlarının ve 4. katmanda imlenen kuralların parametreleridir. Bu parametreler uygun değerlere getirildiğinde sistem istenen $(x,y) \rightarrow z$ eşleşmesini gerçekleştirir. ANFIS ile ilgili detaylı bilgi (J.,Jang,1993)’den edinilebilir.

ABC algoritmasının bulanık sistemin eğitim performansını ölçebilmek için yaygın olarak kullanılan PSO ve DE algoritmalarından faydalanılmıştır. Bu iki algoritma ile aynı uygulamalar ve sonucunda karşılaştırmalar yapılmış ve erişilen bulgulara sonraki bölümlerde yer verilmiştir.

3.3. Kullanılan Algoritmalar

Yapılan çalışmada dinamik sistemleri ANFIS bulanık sistemi ile tanıma/modelleme için eğitim amaçlı ABC algoritması kullanılmıştır. ABC algoritmasının bulanık sistem modellemedeki başarısını ölçebilmek için PSO ve DE algoritmaları karşılaştırma için kullanılmıştır. Uygulamada kullanılan ABC, PSO ve DE algoritmaların özellikleri, yapısı ve parametre değerleri ile ilgili bilgiler aşağıda verilmiştir.

3.3.1. ABC algoritması

Çalışmada örnek dinamik sistemlerin tanıma/modellemesi yapılırken sadece standart ABC algoritması kullanılmamıştır. Uygulama için standart ABC algoritması modifiye edilerek iki adet geliştirilmiş ABC algoritması (G-ABC-I, G-ABC-II) tanımlanmış ve belirlenen dinamik sistem tanıma problemleri üzerinde üç farklı ABC algoritması ile çalışılmıştır. Bu algoritmaların kısaca tanımı aşağıda verilmiştir:

- **Standart ABC algoritması:** ABC algoritması üzerinde herhangi bir değişiklik yapılmamış halidir. Sistem tanıma ve modelleme için kullanılan standart ABC için yazılım kodları EK-1'de verilmiştir.
- **Geliştirilmiş ABC-I (G-ABC-I) :** Standart ABC algoritmasında bulunan θ (teta) değişkenin lineer bir şekilde azaltılması ile algorithmada bir değişiklik yapılmıştır. θ : $[-1 \ 1]$ aralığında düzgün dağılımlı rasgele bir sayıdır. Geliştirilmiş ABC-I için gözcü arı atama işlemleri E.3.1 ve E.3.2'da verildiği gibi tanımlanmıştır.

$$V_{ij} = x_{ij} + \theta * w_{ij} (x_{ij} - x_{kj}) \quad (E.3.1)$$

$$w_i = \frac{-1.4}{G_{max}} * n + 1.5 \quad (E.3.2)$$

Eşitliklerde; G_{max} : Maksimum nesil sayısı , n: Geçerli nesil sayısı

V_{ij} : i . gözcü arının j . boyuttaki değeri

X_{ij} : i . işçi arının j . boyuttaki değeri

X_{kj} : k . işçi arının j . boyuttaki değeri

k : rasgele belirlenen komşu arı indisi

- **Geliştirilmiş ABC-II (G-ABC-II)** : Standart ABC algoritmasında bulunan θ 'ya küçük bir normal dağılımlı rasgele sabitin eklenmesi ile algorithmada bir değişiklik yapılmıştır. Geliştirilmiş ABC-II 'de yapılan değişiklikler E.3.3 ve E.3.4'de verilmiştir.

$$katsayi = \lambda * rand[size(v)] \quad (E.3.3)$$

$$V_j = V_j + katsayi_j \quad (E.3.4)$$

Yukarıdaki E.3.3'de " λ " olarak görülen eklenen küçük sabit, giriş eğitim verilerinin öz değerlerine göre aşağıdaki gibi (E.3.5) belirlenmiştir:

$$X: \text{giriş veri kümesi olmak üzere; } \lambda \leq \frac{1}{\max(eig(XX^T))} \quad (E.3.5)$$

Sistem tanıma/modelleme için belirlenen üç farklı ABC algoritması için Çizelge 3.3'de verilen parametre değerleri kullanılmıştır. Her bir algoritma sürü büyüklüğü değeri 40 alınmış ve 50 kez koşturulmuştur. Algoritma ile optimize edilecek parametre sayısı ise Çizelge 3. 2' nin en sağdaki sütununda verilen sayı kadardır.

Çizelge 3. 3. ABC parametreleri

<i>Parametre Adı</i>	<i>Parametre Değerleri</i>
Koloni büyüklüğü (N)	40
Max nesil sayısı	1000
Limit	(N/2)*D*0.5
Algoritma koşma sayısı	50

3.3.2. PSO

Dinamik sistem tanıma problemleri üzerinde iki farklı PSO algoritması ile çalışılmıştır. Çalışmada standart PSO algoritmasının yanında, modifiye edilerek geliştirilmiş PSO algoritması (PSO-I) kullanılmıştır. Bu algoritmaların kısaca tanımı aşağıda verilmiştir.

- **Standart PSO algoritması:** PSO algoritmasının üzerinde bir değişiklik yapılmamış halidir.
- **Geliştirilmiş PSO-I (PSO-I) :** Standart PSO algoritmasında bulunan parçacık hızı değerine, daha önceden belirlenmiş “Lamda” katsayısının eklenmesiyle bir değişikliğe gidilmiştir (Çavuşlu vd., 2012).

Sistem tanıma/modelleme için belirlenen iki farklı PSO algoritması için Çizelge 3. 4’de verilen parametre değerleri kullanılmıştır.

Çizelge 3. 4. PSO parametreleri

<i>Parametre Adı</i>	<i>Parametre Değerleri</i>
Sürü Büyüklüğü	40
Max nesil sayısı	1000
c1, c2	2.05
Algoritma koşma sayısı	50

3.3.3. DE

Çalışmada standart DE algoritması kullanılmıştır. DE üzerinde hiçbir değişiklik yapılmamış basit halidir. Sistem tanıma/modelleme için belirlenen DE algoritması için Çizelge 3.5’de verilen parametre değerleri kullanılmıştır.

Çizelge 3. 5. DE Parametreleri

<i>Parametre Adı</i>	<i>Parametre Değeri</i>
Popülasyon büyüklüğü (NP)	40
Max nesil sayısı	1000
Çaprazlama sabiti	0.4
Skala faktörü	0.3
Algoritma koşma sayısı	50

3.4. Bireylerin Yapısı

Çalışmada dinamik sistem modelleme aracı olarak Şekil 3.1’de verilen ANFIS yapısı kullanılmıştır. 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 modellemesi yapılırken sistemler durumuna göre belirlenen girişler için ikişer adet üyelik fonksiyonu tanımlanmış ve bunlara ait kural sayıları Çizelge 3. 2’de verilmiştir. 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.

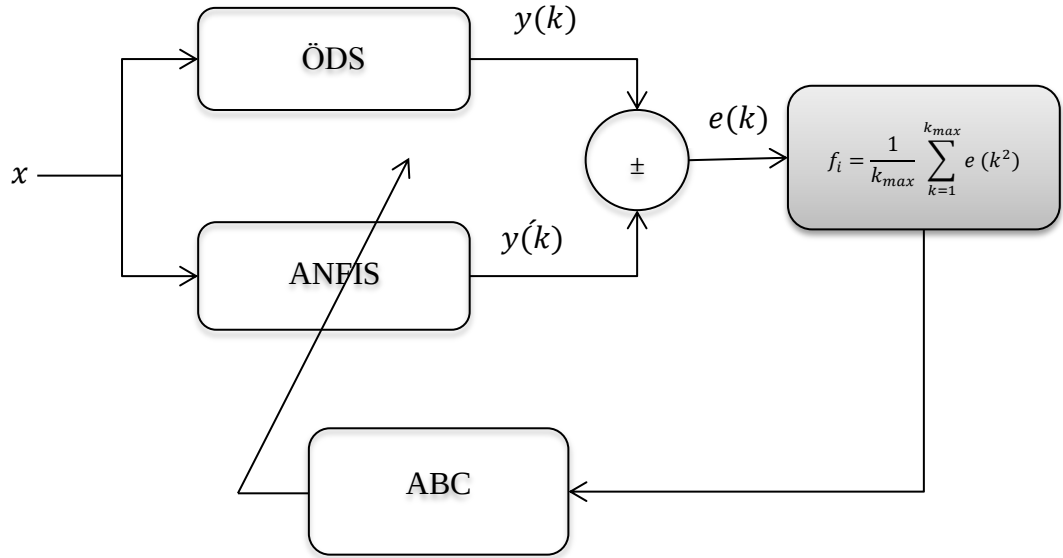
$$b_i = [c_{11} \ c_{12} \ c_{13} \ \dots \ \sigma_{11} \ \sigma_{12} \ \sigma_{13} \ \dots \ p_1 \ q_1 \ r_1 \ \dots \ p_R \ q_R \ r_R]$$

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 göstermektedir.

3.5.ABC ile ANFIS parametre optimizasyonu



Şekil 3. 4. ABC ile ANFIS optimizasyonu blok yapısı

Şekil 3.4’de optimizasyon için kullanılan blok model yapısı görülmektedir.

4. SEZGİSEL ARAMA ALGORİTMA TABANLI BULANIK SİSTEM OPTİMİZASYONU

Bu bölümde, ABC algoritması literatürde çok kullanılan PSO ve DE ile kıyaslanarak ÖDS başarımı incelenmiştir. Kıyaslama Bölüm 3 Çizelge 3.1’de verilen beş adet dinamik sistemin bulanık modellenmesi üzerine yapılmıştır. Tüm deneylerde kullanılan algoritmaların popülasyon büyüklüğü, maksimum nesil sayısı ve optimize edilecek parametre sayısı aynı kullanılmıştır. ABC, PSO ve DE algoritmalarında kullanılan popülasyon büyüklüğü 40, max nesil sayısı 1000 olarak alınmıştır. Algoritmalarla ilgili diğer parametre değerleri Bölüm 3.3’de verilmiştir.

Çizelge 3.1’de verilen dinamik sistemler üzerinde ABC, PSO ve DE algoritmaların başarımlarını kıyaslamaları yapılmıştır. ABC algoritması için standart ABC, G-ABC-I ve G-ABC-II olmak üzere üç, PSO algoritması için standart PSO ve G-PSO-I olmak üzere iki, DE için ise sadece standart DE olmak üzere toplam 6 değişik sezgisel algoritma kıyaslama amaçlı kullanılmıştır.

Her algoritma, her bir dinamik sistem için, rasgele oluşturulmuş başlangıç popülasyonları ile 50’şer kez ayrı ayrı koşturulmuştur. Her bir algoritmanın 50’şer kez koşması sonucu elde edilen sonuçlar istatistiki olarak irdelenmiştir.

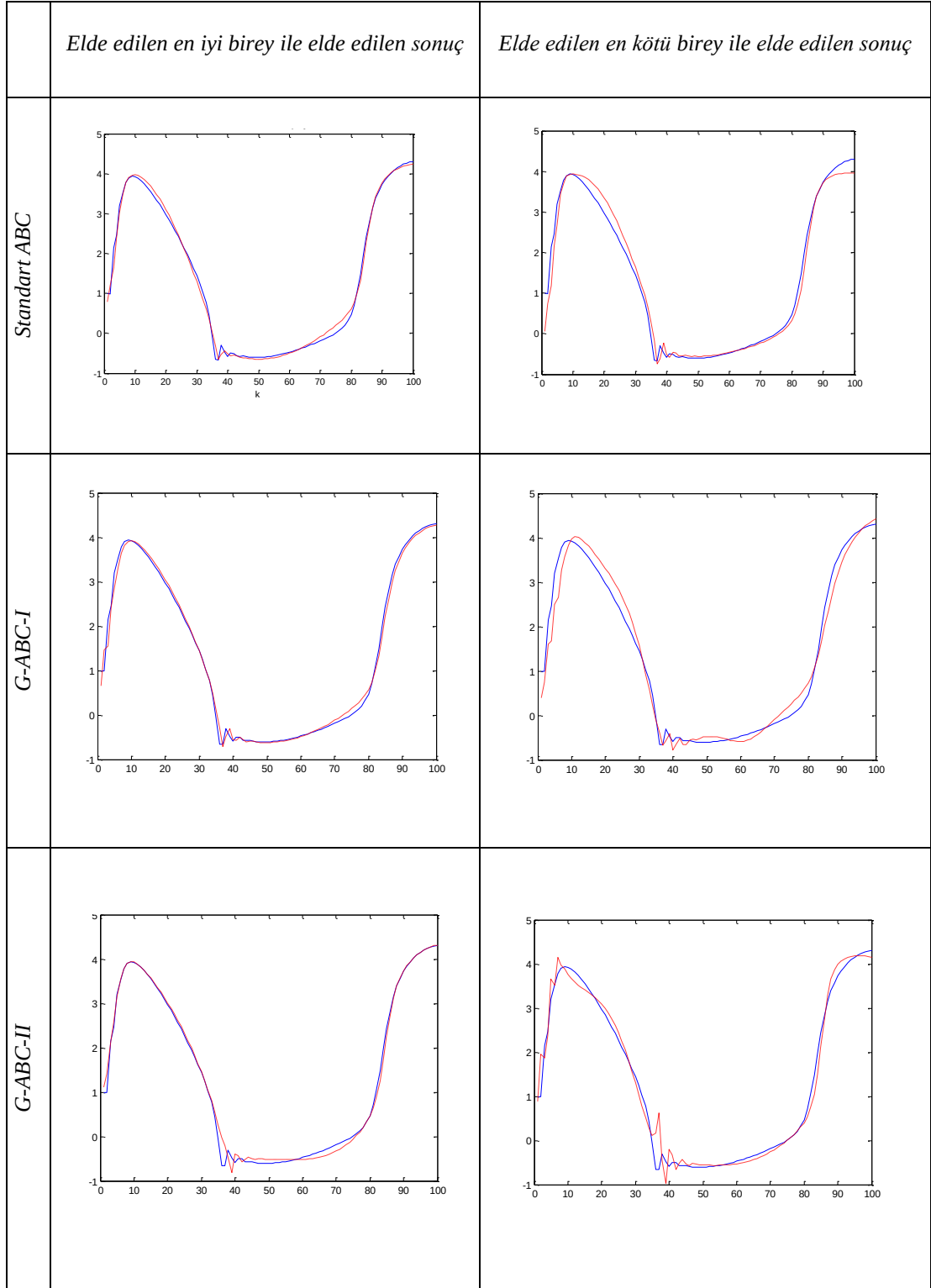
Çizelge 3.1’de verilen örnek dinamik sistemler sırasıyla ABC, PSO ve DE algoritmaları kullanılarak ANFIS ile modellenmiş ve her bir sisteme için elde edilen model sonuçları en iyi, en kötü, başlangıç durum ve son durum şeklinde kategorize edilerek aşağıdaki alt bölümlerde grafiksel olarak verilmiştir. Verilen şekillerde mavi düz çizgi ile verilenler istenen sonucu, kırmızı kesikli çizgiyle verilen ise elde edilen sonucu göstermektedir.

4.1. ÖDS 1 'in Tanınması/Modellenmesi

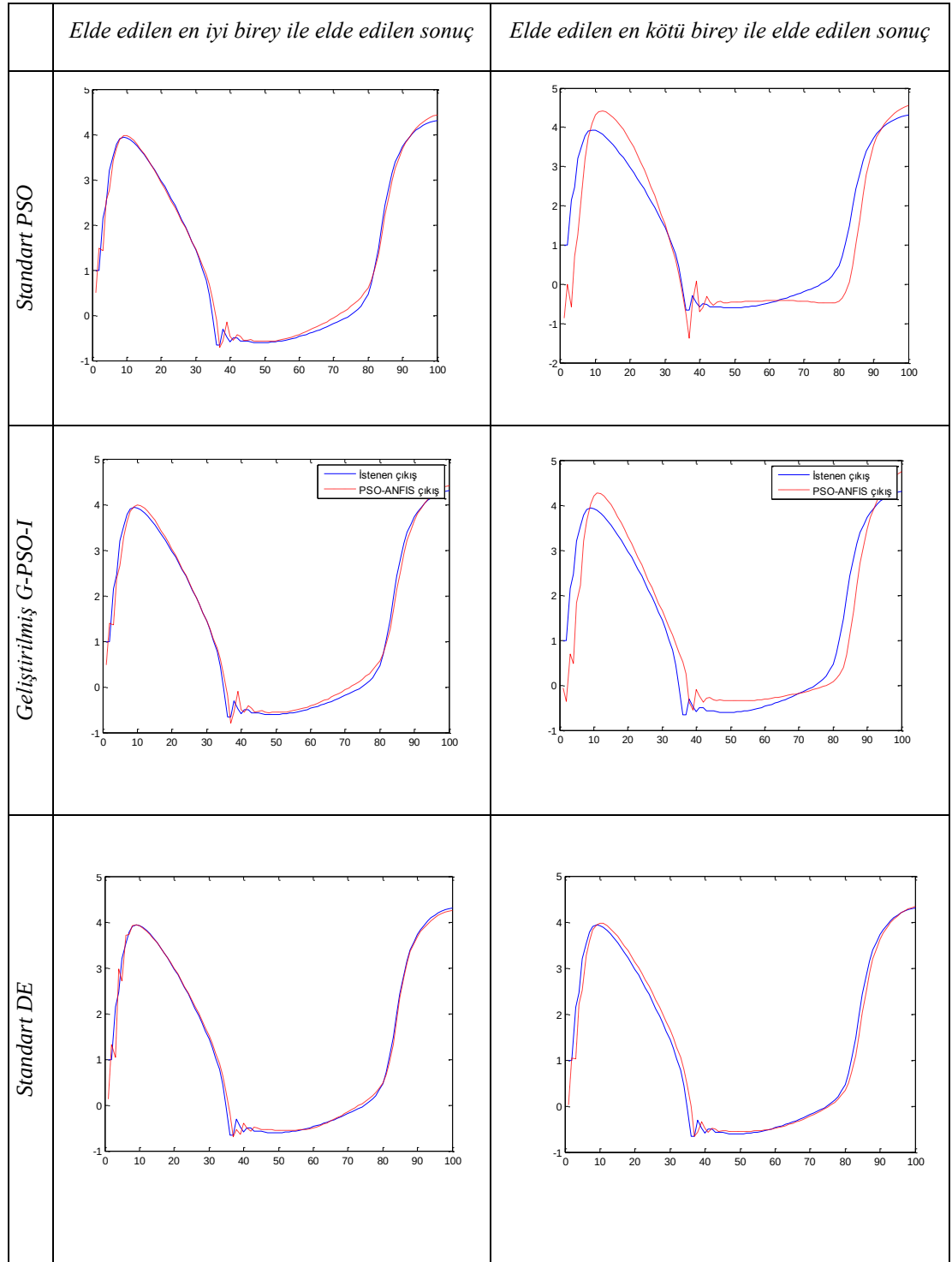
4.1.1.ÖDS 1 için eğitim aşaması ve sonuçları

ÖDS 1 için Şekil 3.2 (a)'da verilen $u(k)$ kullanılarak oluşturulan eğitim seti dizileri kullanılarak koşturulan ABC, PSO ve DE algoritmalarının başarımları sonuçları sırasıyla aşağıda verilmiştir. Bu uygulama sonucunda Şekil 4.1 ve Şekil 4.2'de ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 1 eğitim fazı için en iyi birey ile elde edilen en iyi ve en kötü birey ile elde edilen en kötü sonuç grafikleri verilmiştir.

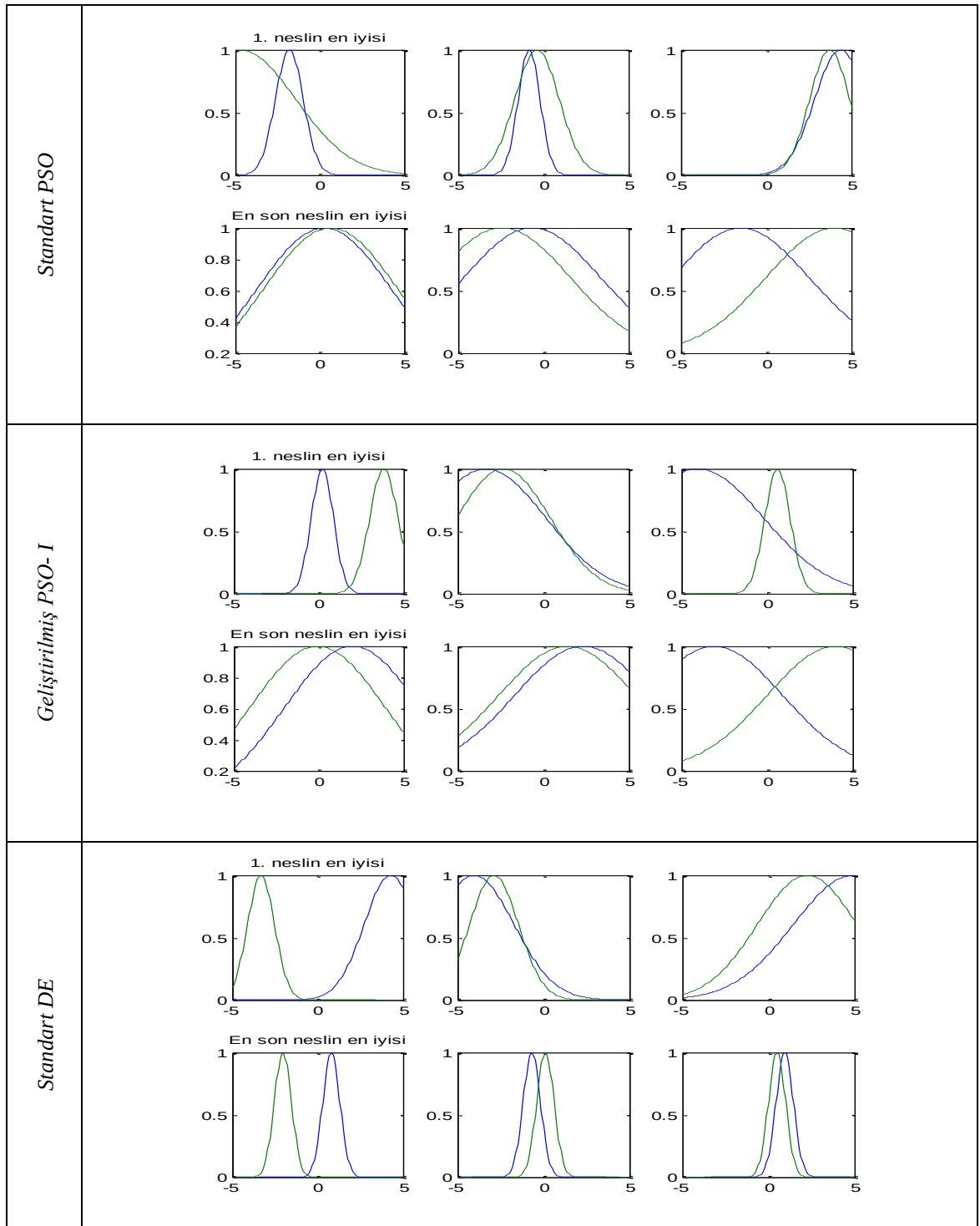
Bu grafikler detaylı bir şekilde incelendiğinde tüm algoritmaların ÖDS1'in eğitim setine göre birbirlerine yakın sonuçlar elde ettikleri görülmüştür. Şekil 4.3 ve Şekil 4.4'de Standart ABC, G-ABC-I, G-ABC-II, Standart PSO, G-PSO-I ve Standart DE'nin ÖDS 1 için bulanık modelleme eğitim fazında ÜF'lerin başlangıç ve son durum grafikleri verilmiştir. Bu şekillerde verilen grafikler ile her bir algoritmanın üyelik fonksiyonlarını üzerinde değişimler görülebilir. Şekillerde birinci sütunda verilen grafikler 1. Girişe, 2.sütun da verilenler 2.girişe, 3.sütunda verilenler ise 3. girişe ait ÜF'lerin ilk ve son durumunu göstermektedir.



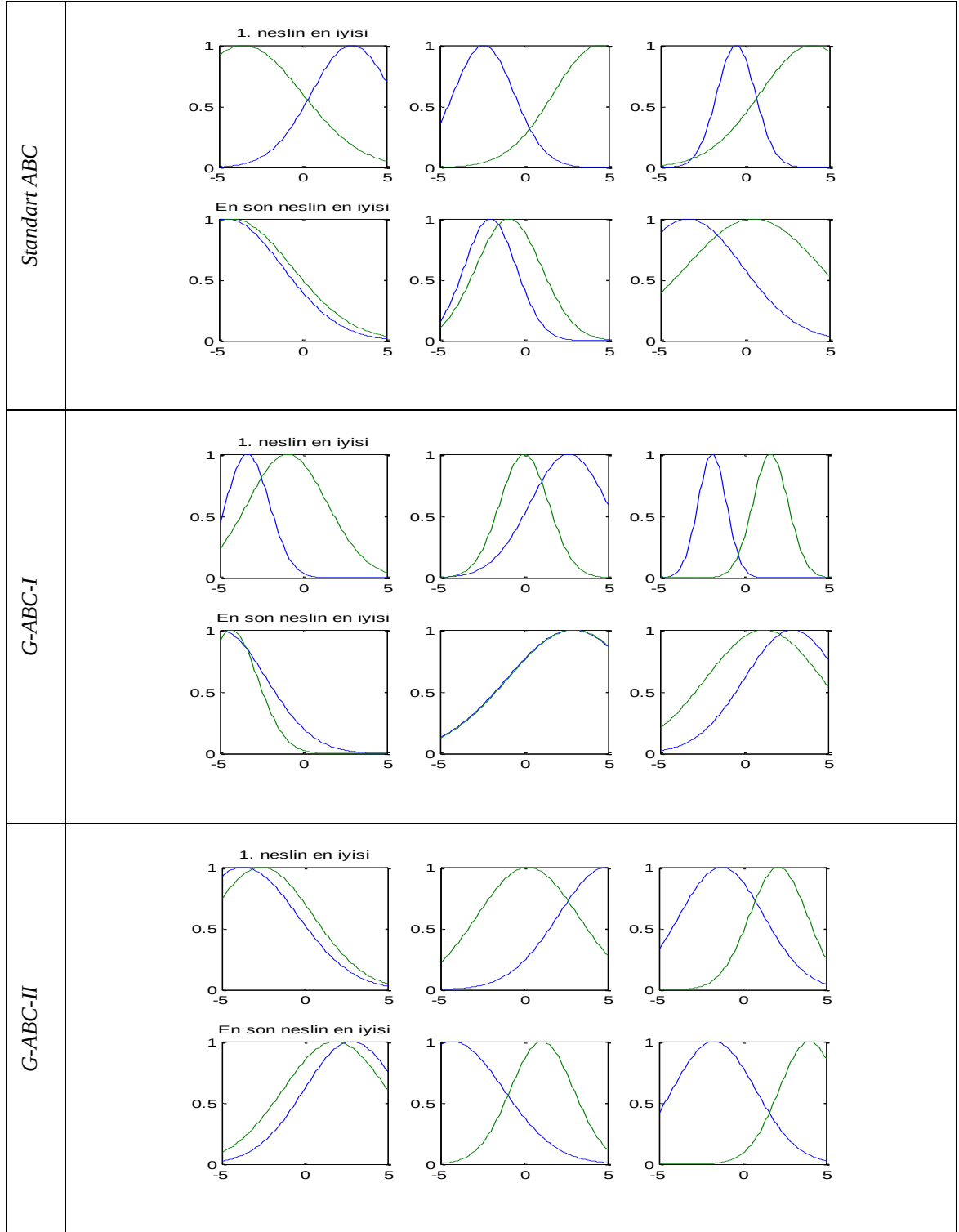
Şekil 4. 1. ABC öğrenmeli ANFIS ile ÖDS 1 için eğitim fazı sistem tanıma sonuçları.



Şekil 4. 2. PSO ve DE öğrenmeli ANFIS ile ÖDS 1 için eğitim fazı sistem tanıma sonuçları.



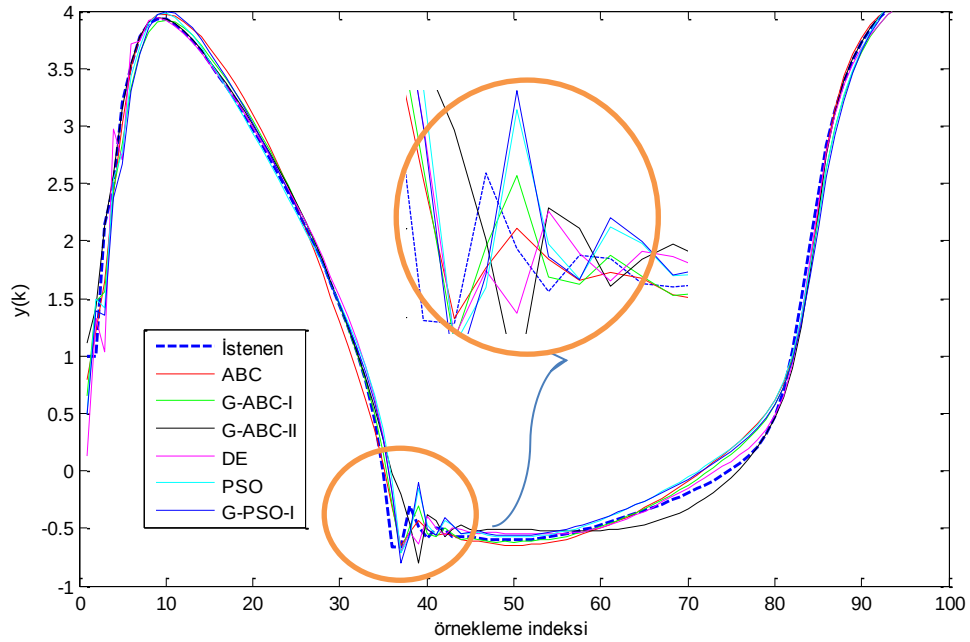
Şekil 4. 3. ABC öğrenmeli ANFIS ile ÖDS 1'in bulanık modelleme eğitim fazında UF'lerin başlangıç ve son durumları.



Şekil 4. 4. PSO ve DE öğrenmeli ANFIS ile ÖDS 1'in bulanık modelleme eğitim fazında UF'lerin başlangıç ve son durumları.

ÖDS 1'in ANFIS ile modellenmesi için Standart ABC, G-ABC-I, G-ABC-II, Standart PSO, G-PSO-I ve Standart DE ile ayrı ayrı koşturulması sonucunda elde edilen iyi ölçüt değerine sahip koşturmanın en iyi bireyi ile elde edilen modelleme sonuçları kıyaslanmıştır. Şekil 4.5'de Standart ABC, G-ABC-I, G-ABC-II, Standart PSO, G-PSO-I ve Standart DE için elde edilen en iyilerin karşılaştırılması verilmiştir.

Şekil 4.5'de algoritmaların ÖDS 1 için elde ettikleri performansı inceleyebilmek için grafik üzerinde kırılma noktası büyütülmüş ve daha detaylı bir inceleme yapılabilmektedir. Buna göre DE, ABC ve G-ABC-I istenene yakın sonuçlar elde ederken; PSO ve G-PSO-I' de ise başarımın kötü olduğu görülmektedir.

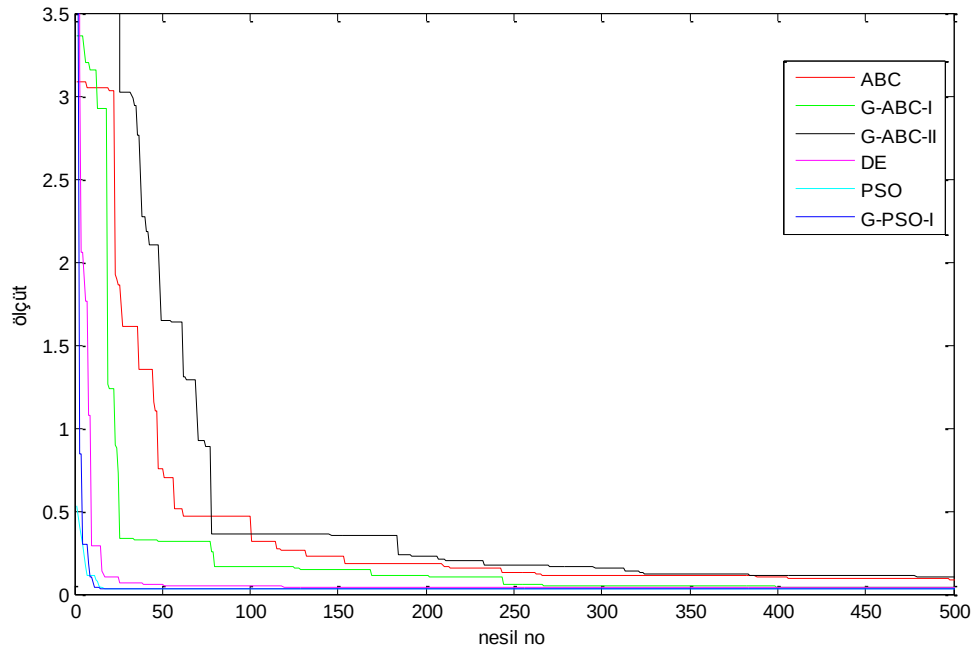


Şekil 4. 5. ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 1 için elde edilen modellerin karşılaştırılması

Şekil 4.5'de ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 1 için elde edilen modellerin karşılaştırılması yapılırken işaretlenen alan için aşağıda Çizelge 4.1'de hata değerleri verilmiştir.

Çizelge 4. 1. ÖDS 1'in işaretlenen alan için elde edilen hata değeri

	35	36	37	38	39	40	41	42	43
İstenen	-0,02252	-0,65816	-0,66551	-0,30067	-0,4835	-0,58764	-0,50024	-0,50788	-0,56924
ABC	0,051635	-0,31709	-0,6539	-0,53067	-0,43521	-0,51036	-0,56029	-0,54162	-0,55612
G-ABC-I	0,151302	-0,28871	-0,70921	-0,47976	-0,30763	-0,55293	-0,56963	-0,50092	-0,55091
G-ABC-II	0,272631	-0,02129	-0,19656	-0,46216	-0,80753	-0,38607	-0,43439	-0,57542	-0,51048
PSO	0,360377	-0,09374	-0,71584	-0,57767	-0,14839	-0,47328	-0,55709	-0,43145	-0,47118
G-PSO-I	0,274718	-0,1886	-0,79729	-0,54614	-0,10219	-0,50347	-0,55958	-0,40894	-0,46725
DE	0,297082	-0,1837	-0,69998	-0,53626	-0,63985	-0,39256	-0,49544	-0,56296	-0,49088



Şekil 4. 6. ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 1 modelleme eğitim seyri

Şekilde verilen eğitim seyri her bir algoritma için ayrı ayrı 50'şer koşmanın(en iyi bireyler için) ortalaması alınarak elde edilmiştir.

ÖDS 1 için algoritmaların eğitim fazında yakınsama hızları Şekil 4.6'da verilmiştir. Şekillerden görüleceği üzere, PSO ve türevinde çabuk premature yakınsama görülmektedir. Çabuk yakınsamanın algoritmaların local minimumda takılıp kalmalarına neden olduğu Şekil 4.5'de verilen sonuçlardan anlaşılmaktadır.

4.1.2.ÖDS 1'in test aşaması ve sonuçları

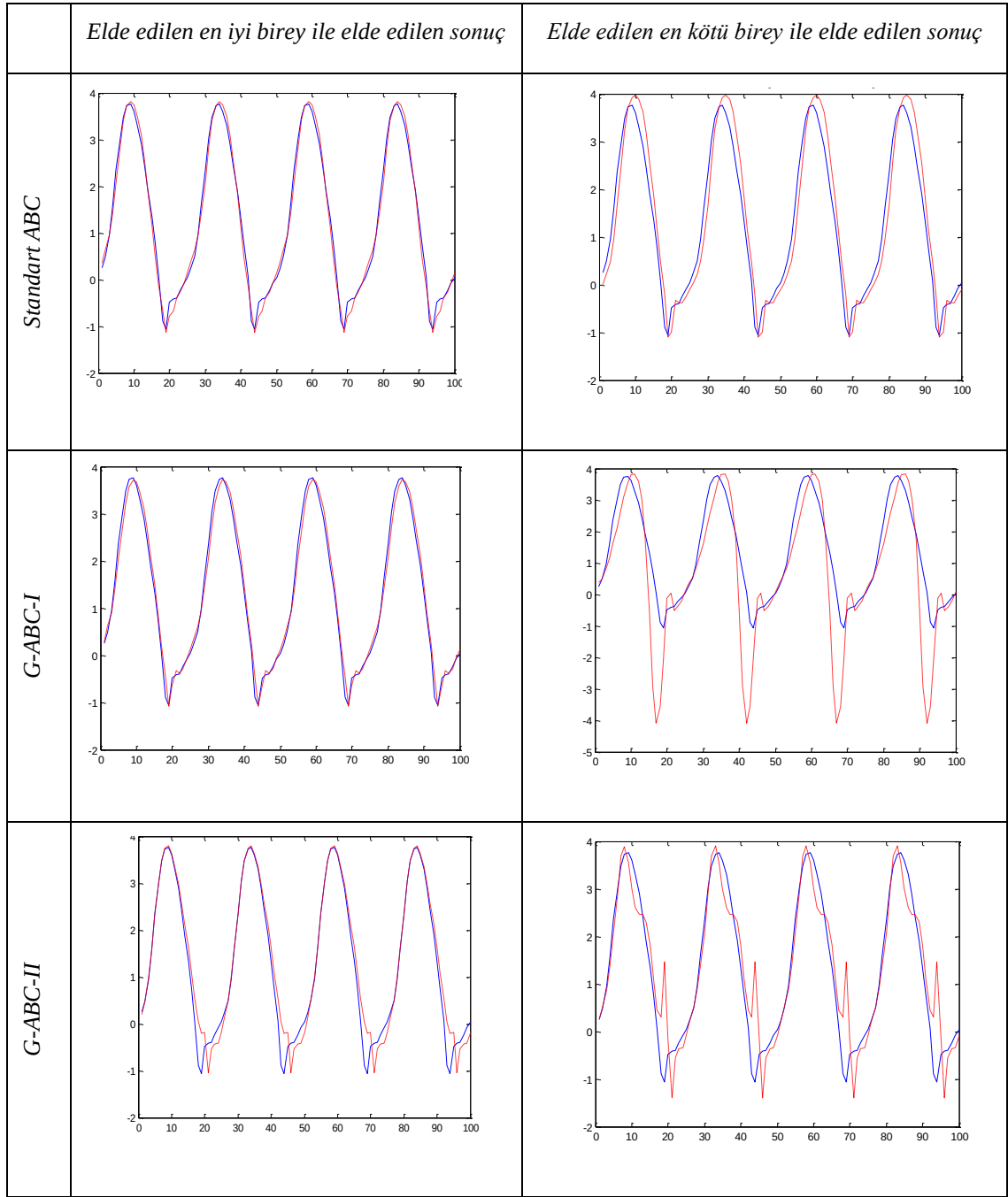
ÖDS 1 için Şekil 3.3 (a)'da verilen giriş ($u(k)$) dizisi kullanılarak oluşturulan veri seti (test seti) aracılığı ÖDS 1'in ABC, PSO ve DE öğrenmeli ANFIS modelindeki çıkış cevapları ve performansı incelenmiştir. ABC algoritması kullanılarak elde edilen

çıkış cevapları PSO ve DE algoritmaları ile elde edilen cevaplarla karşılaştırılmıştır. Çalışma ile ABC algoritmasının bulanık sistem optimizasyonundaki başarısı değerlendirilmiştir.

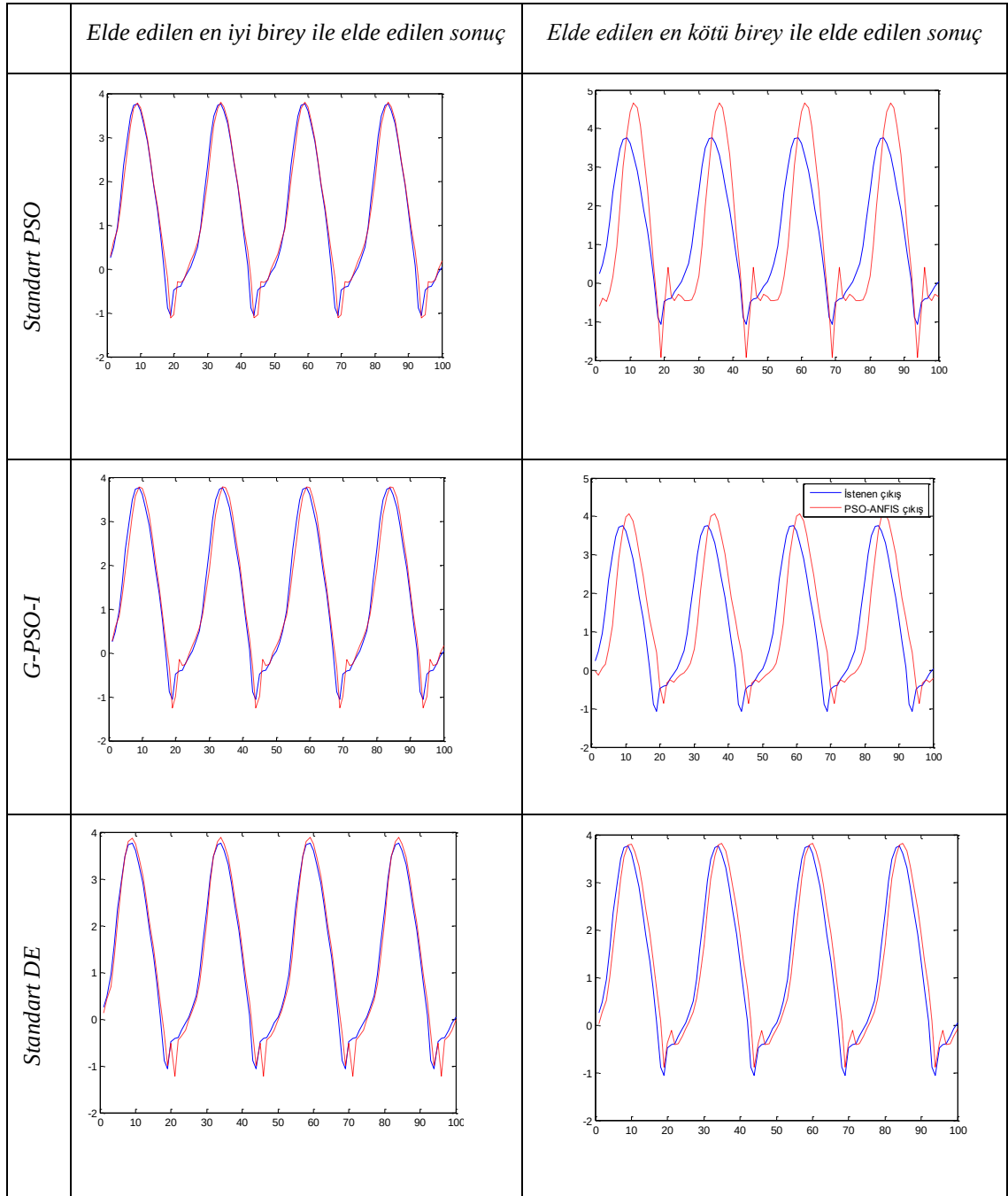
Bölüm 3 Şekil 3.3(a)'da verilen ÖDS 1 için girişlerden biri olan $u(k)$ dizisi için eğitim fazında; $u(k)=\cos \frac{2\pi k}{100}$ kullanılırken, test fazı için Şekil 3.3(b)'dan da görüleceği üzere $u(k)=\sin \frac{2\pi k}{25}$ olarak kullanılmıştır. ÖDS 1 için $u(k)$ giriş dizisi hazırlandıktan sonra, Çizelge 3.2'de belirlenen diğer girişlerle beraber test veri seti oluşturulmuştur.

Oluşturulan test seti ile ABC, DE ve PSO algoritmalarıyla elde edilen ÖDS 1 ANFIS modellerinin sonuçları Şekil 4.7 ve Şekil 4.8'de grafiksel olarak verilmiştir.

Şekil 4.7 ve Şekil 4.8' de verilen grafikler detaylı olarak incelendiğinde test fazında en iyi sonucun G-ABC-I algoritması ile alındığı görülmüştür.



Şekil 4. 7. ABC öğrenmeli ANFIS ile ÖDS 1 için test fazı sistem tanıma sonuçları



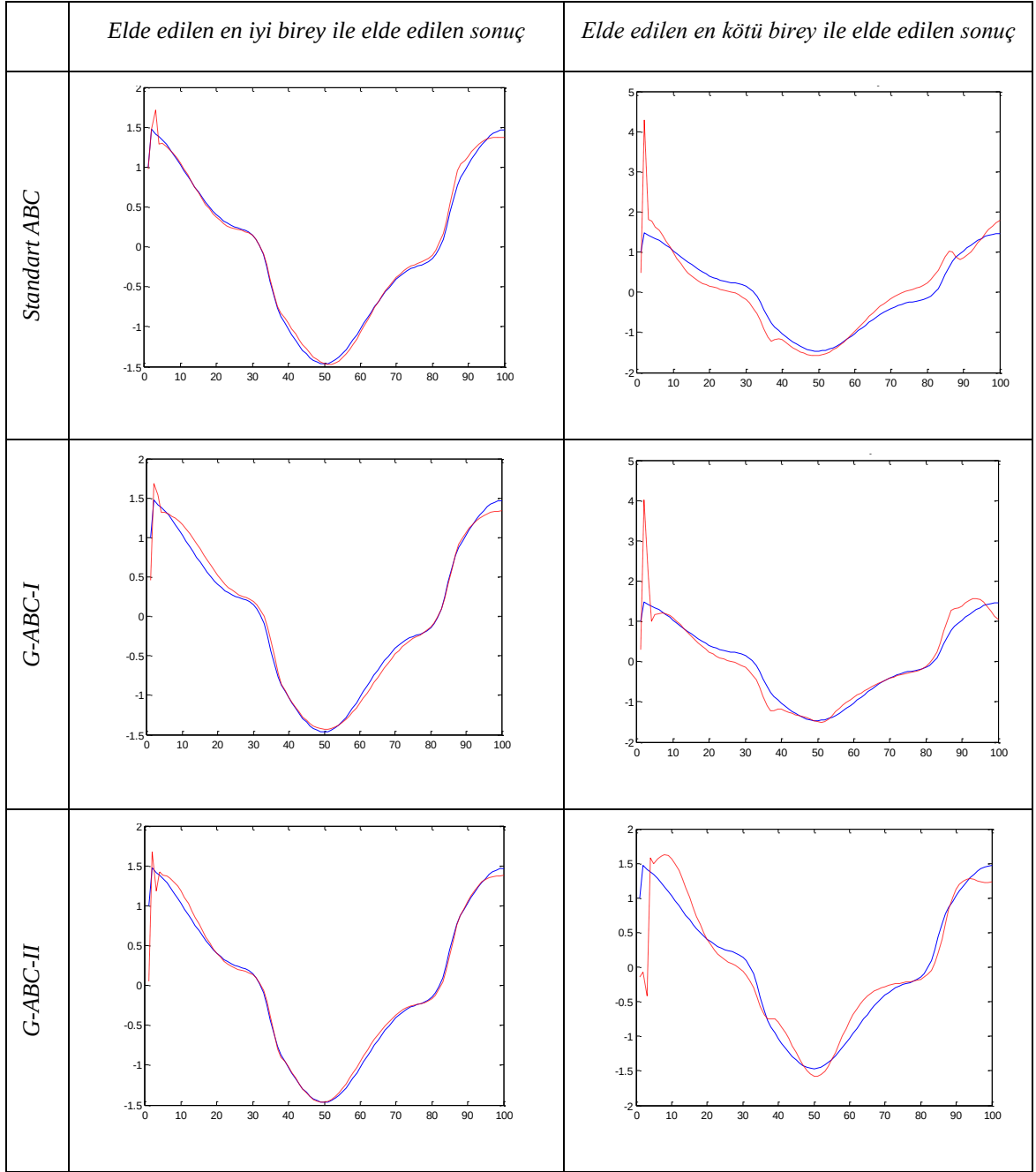
Şekil 4. 8. PSO ve DE öğrenmeli ANFIS ile ÖDS 1 için test fazı sistem tanıma sonuçları

4.2. ÖDS 2'nin Tanınması/Modellenmesi

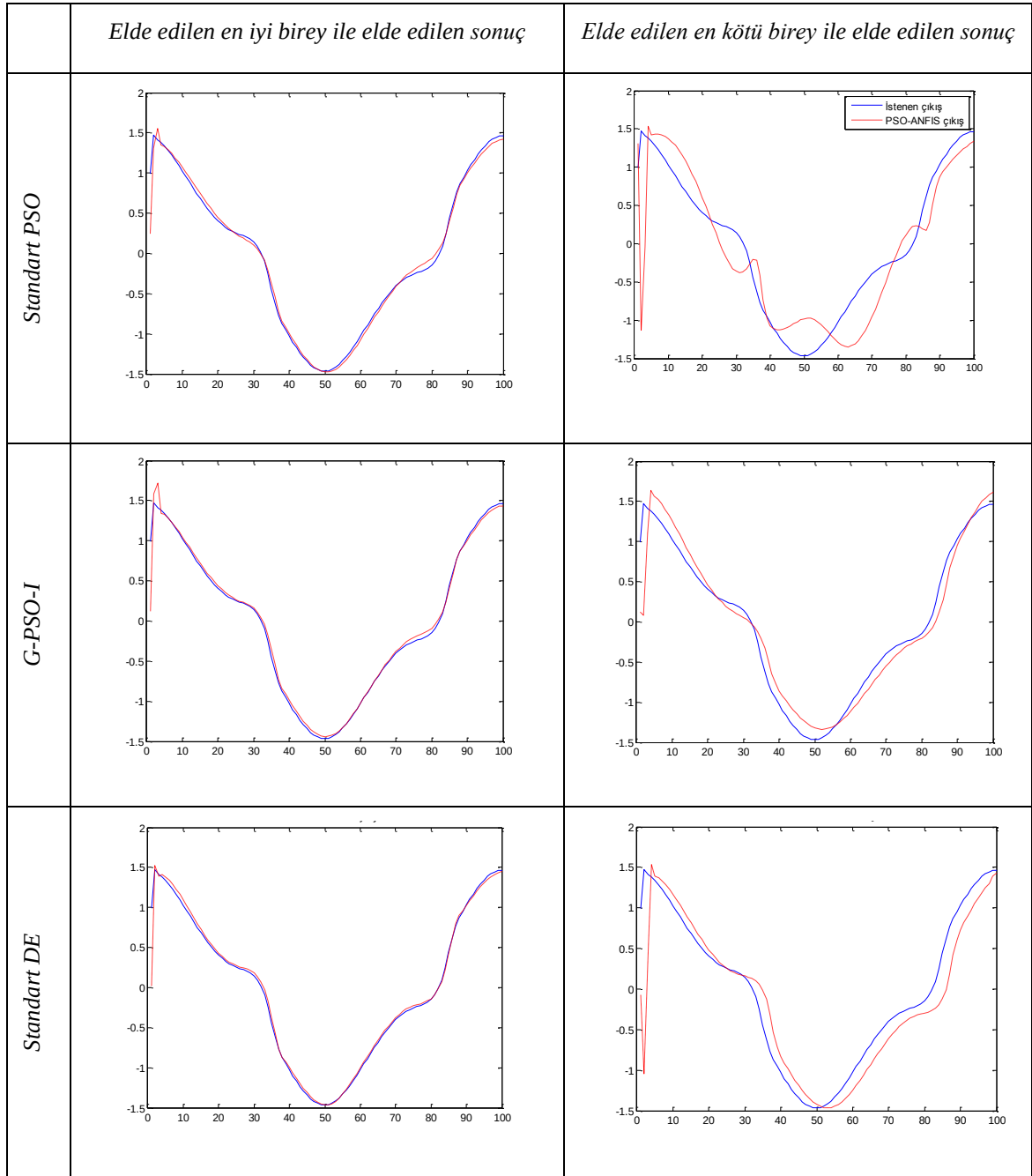
4.2.1.ÖDS 2 için eğitim aşaması ve sonuçları

ÖDS 2 için, Bölüm 3 Çizelge 3.2'de verilen giriş değişkenleri ve şekil 3.2(a)'da verilen $u(k)$ giriş dizisi eğitim veri seti hazırlanmıştır. ÖDS 2 için Şekil 3.2'de verilen eğitim veri seti kullanılarak sırasıyla ABC, PSO ve DE algoritmaları ANFIS optimizasyonu için 50'şer kez koşturulmuştur. Uygulama sonucunda ABC, PSO ve DE algoritmasının sistem tanıma sonuçları grafiksel olarak Şekil 4.9 ve Şekil 4.10'da listelenmiştir.

Şekil 4. 9 ve Şekil 4.10’da verilen şekillerde en iyi sonucu DE algoritmasının verdiği görülmüştür.



Şekil 4. 9. ABC öğrenmeli ANFIS ile ÖDS 2 için eğitim fazı sistem tanıma sonuçları



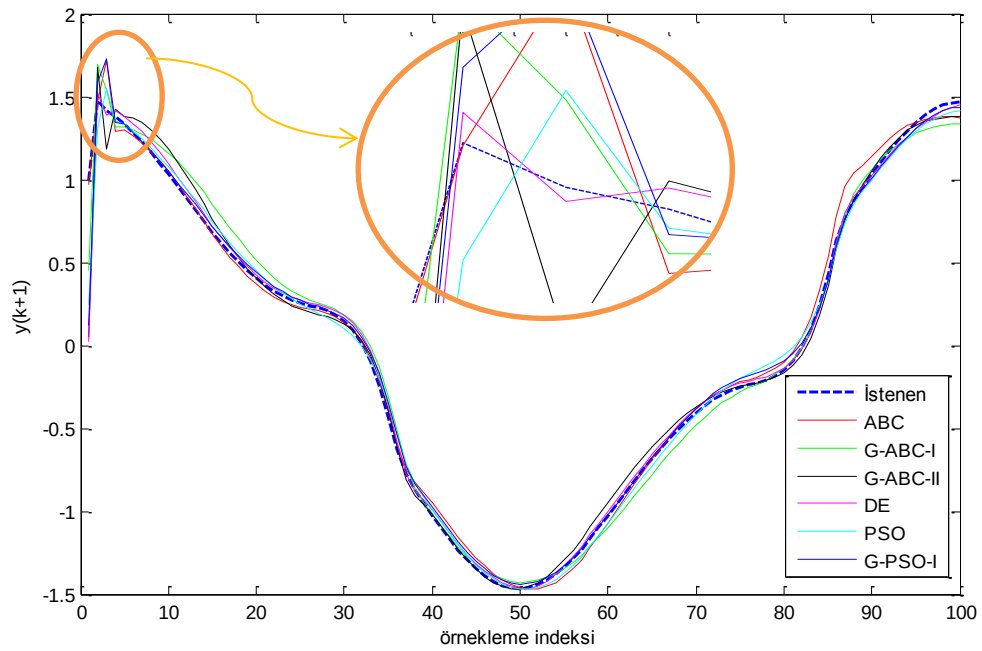
Şekil 4. 10. PSO ve DE öğrenmeli ANFIS ile ÖDS 2 için eğitim fazı sistem tanıma sonuçları

ÖDS 2'nin ANFIS ile modellenmesi için Standart ABC, G-ABC-I, G-ABC-II, Standart PSO, G-PSO-I ve Standart DE ile ayrı ayrı koşturulması sonucunda elde edilen iyi ölçüt değerine sahip koşmanın en iyi bireyi ile elde edilen modelleme sonuçları kıyaslanmıştır. Şekil 4.11'de Standart ABC, G-ABC-I, G-ABC-II, Standart PSO, G-PSO-I ve Standart DE için elde edilen en iyilerinin karşılaştırılması verilmiştir.

Şekil 4.12’de algoritmaların ÖDS 2 için elde ettikleri performansı inceleyebilmek için grafik üzerinde kırılma noktası büyütülmüş ve daha detaylı bir inceleme yapılabilmiştir. Buna göre DE ile istenene yakın sonuçlar elde edilirken; diğer algoritmalarda ise başarımın kötü olduğu görülmektedir.

Şekilde verilen eğitim seyri her bir algoritma için ayrı ayrı 50’şer koşmanın (en iyi bireyler için) ortalaması alınarak elde edilmiştir.

ÖDS 2 için algoritmaların eğitim fazında yakınsama hızları Şekil 4.12’de verilmiştir. Şekillerden görüleceği üzere, PSO ve türevinde çabuk premature yakınsama görülmektedir. Çabuk yakınsamanın algoritmaların local minimumda takılıp kalmalarına neden olduğu Şekil 4.12’de verilen sonuçlardan anlaşılmaktadır.

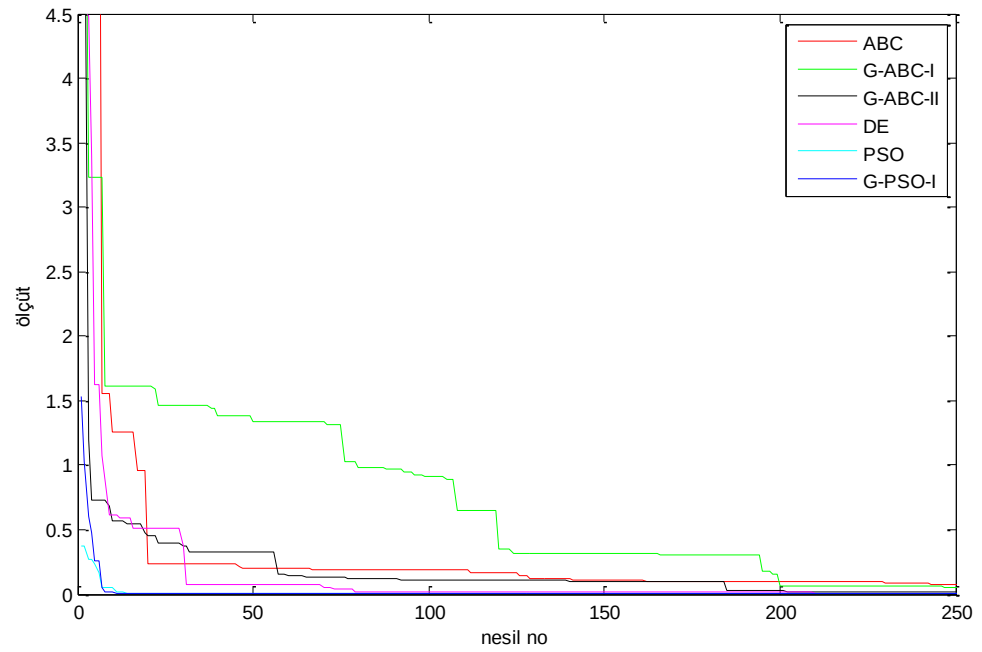


Şekil 4. 11. ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 2 elde edilen modellerin karşılaştırılması

Şekil 4.11’de ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 2 elde edilen modellerin karşılaştırılması yapılırken işaretlenen alan için aşağıda Çizelge 4.2’de hata değerleri verilmiştir.

Çizelge 4. 2. ÖDS 2'nin işaretlenen alan için elde edilen hata değeri

	1	2	3	4	5	6	7	8
İstenen	0,994092	1,476521	1,412095	1,380319	1,335346	1,283574	1,225606	1,162757
ABC	0,972806	1,472788	1,714482	1,28707	1,298444	1,260548	1,219414	1,169147
G-ABC-I	0,455304	1,688291	1,539641	1,315965	1,31505	1,296268	1,272734	1,24229
G-ABC-II	0,050487	1,674328	1,185494	1,421368	1,382528	1,370035	1,341893	1,300979
PSO	0,242163	1,306635	1,552474	1,353157	1,333626	1,290737	1,243388	1,190459
G-PSO-I	0,122145	1,586314	1,723399	1,343461	1,332973	1,283417	1,231258	1,17364
DE	0,020588	1,520857	1,392006	1,411112	1,379713	1,33756	1,286994	1,228324



Şekil 4. 12. ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 2 modelleme eğitim seyri

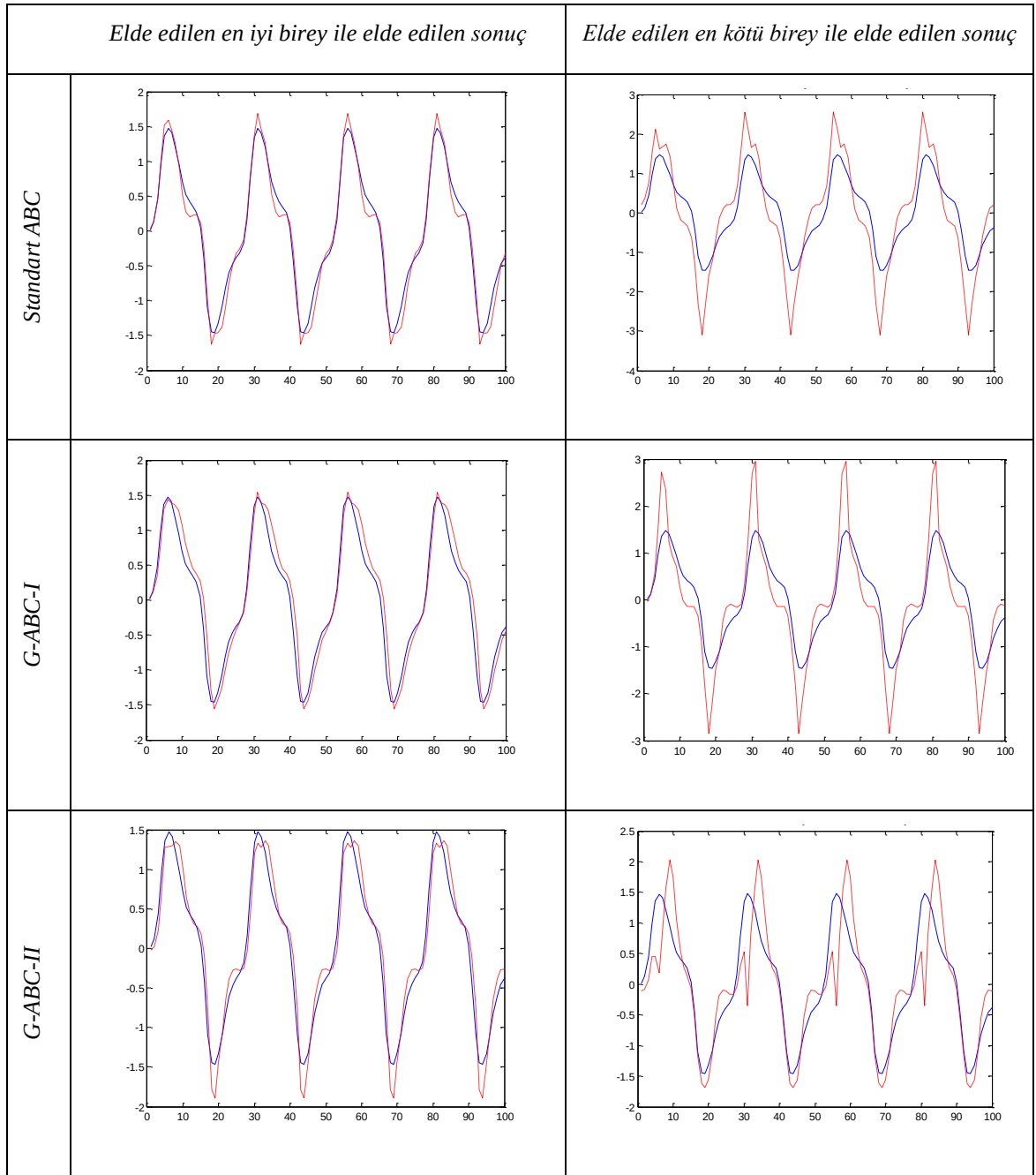
4.2.2. ÖDS 2'in test aşaması ve sonuçları

ÖDS 2 için Şekil 3.3 (b)'da verilen giriş ($u(k)$) dizisi kullanılarak oluşturulan veri seti (test seti) aracılığı ÖDS 2'nin ABC, PSO ve DE öğrenmeli ANFIS modelindeki çıkış cevapları ve performansı incelenmiştir. ABC algoritması kullanılarak elde edilen çıkış cevapları PSO ve DE algoritmaları ile elde edilen cevaplarla karşılaştırılmıştır. Çalışma ile ABC algoritmasının bulanık sistem optimizasyonundaki başarısı değerlendirilmiştir.

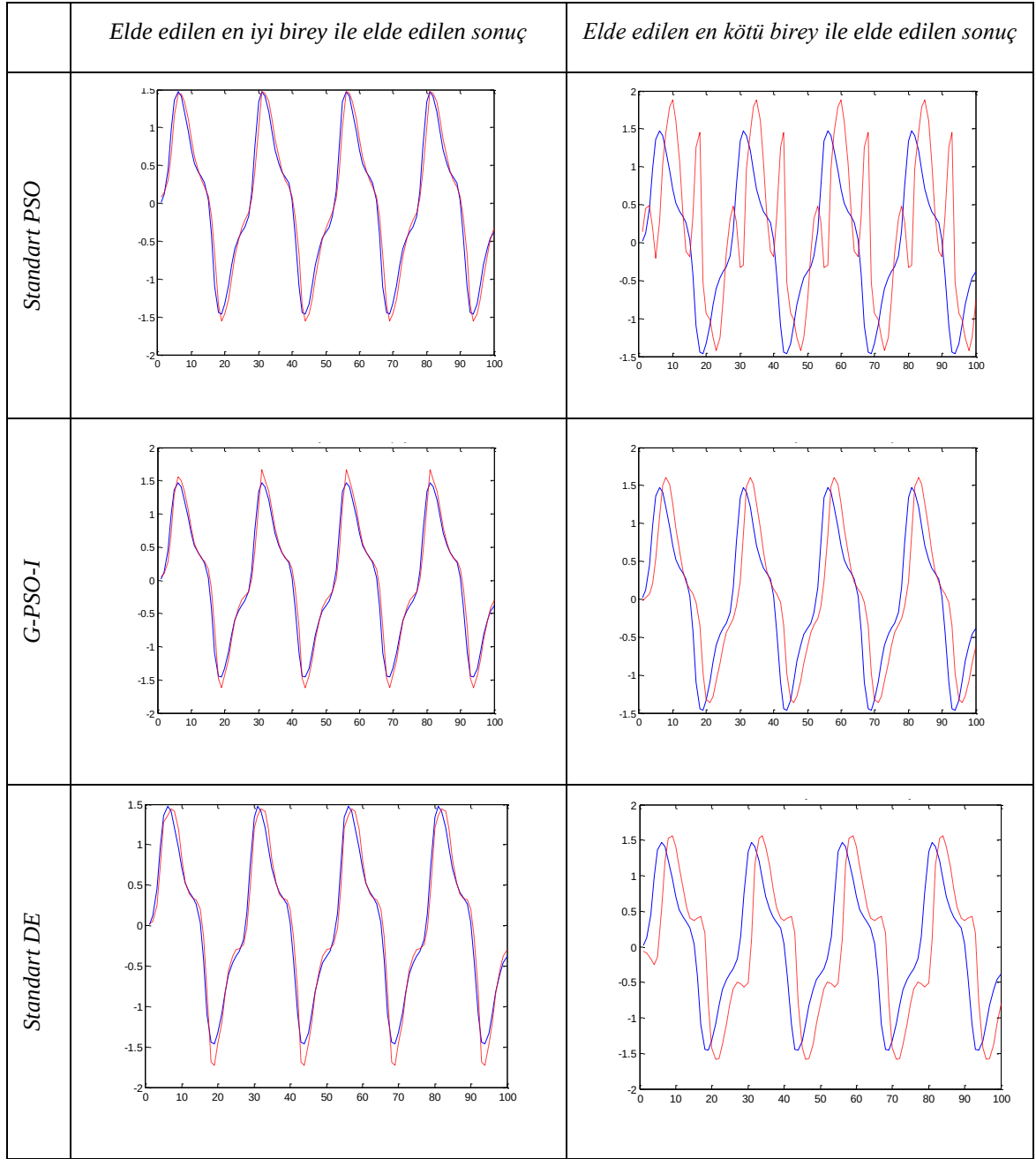
Bölüm 3 Şekil 3.3(b)'da verilen ÖDS 2 için girişlerden biri olan $u(k)$ dizisi için eğitim fazında; $u(k)=\cos \frac{2\pi k}{100}$ kullanılırken, test fazı için Şekil 3.3(b)'dan da görüleceği üzere $u(k)=\sin \frac{2\pi k}{25}$ olarak kullanılmıştır. ÖDS 2 için $u(k)$ giriş dizisi hazırlandıktan sonra, Çizelge 3.2'de belirlenen diğer girişlerle beraber test veri seti oluşturulmuştur.

Oluşturulan test seti ile ABC, DE, ve PSO algoritmalarıyla elde edilen ÖDS 2 ANFIS modellerinin sonuçları Şekil 4.13 ve Şekil 4.14'de grafiksel olarak verilmiştir.

Şekil 4.13 ve Şekil 4.14' de verilen grafikler detaylı olarak incelendiğinde test fazında en iyi sonucun Standart PSO algoritması ile alındığı görülmüştür.



Şekil 4. 13. ABC öğrenmeli ANFIS ile ÖDS 2 için test fazı sistem tanıma sonuçları



Şekil 4. 14. PSO ve DE öğrenmeli ANFIS ile ÖDS 2 için test fazı sistem tanıma sonuçları

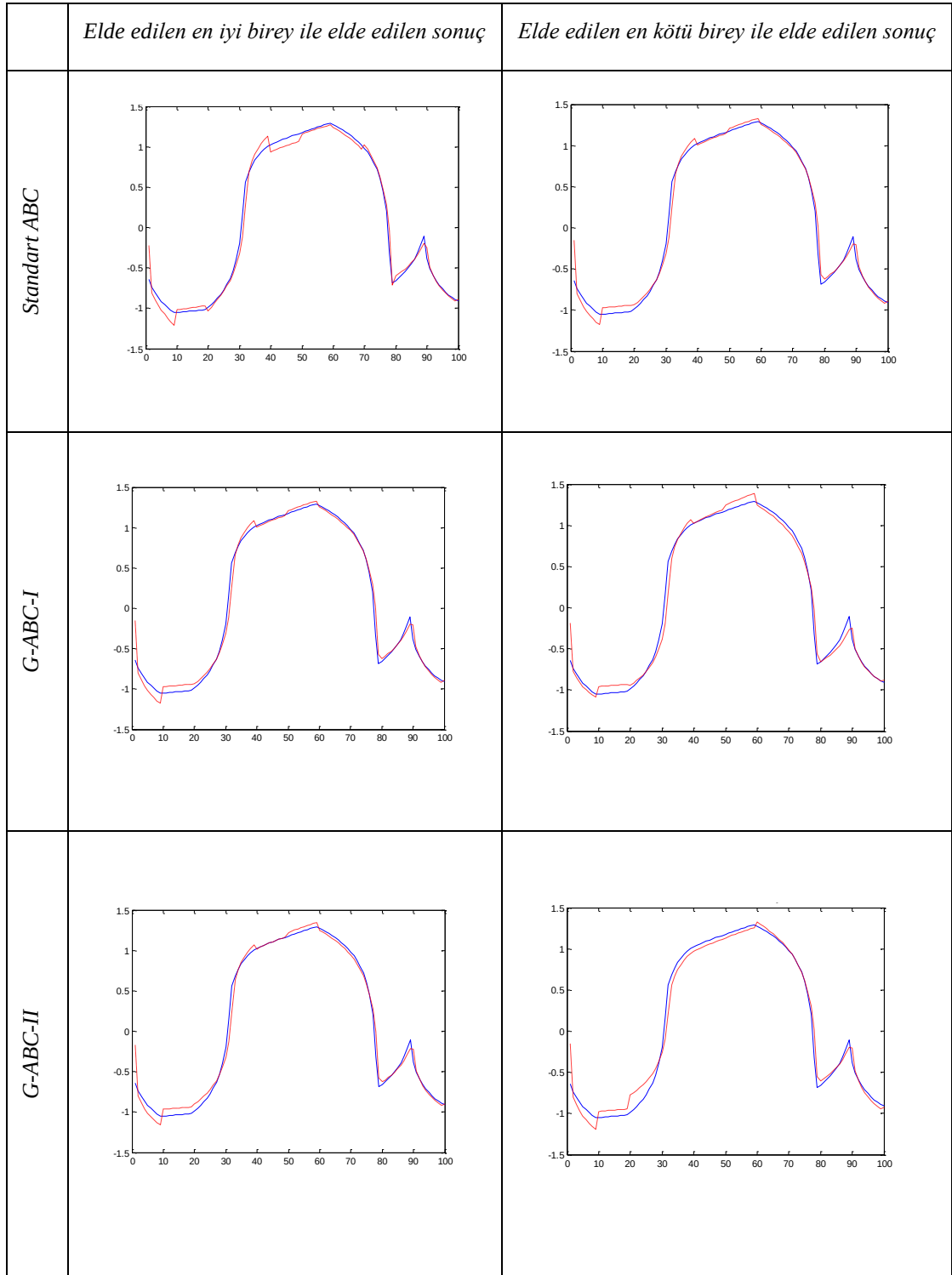
4.3. ÖDS 3'ün Tanınması/Modellenmesi

4.3.1. ÖDS 3 için eğitim aşaması ve sonuçları

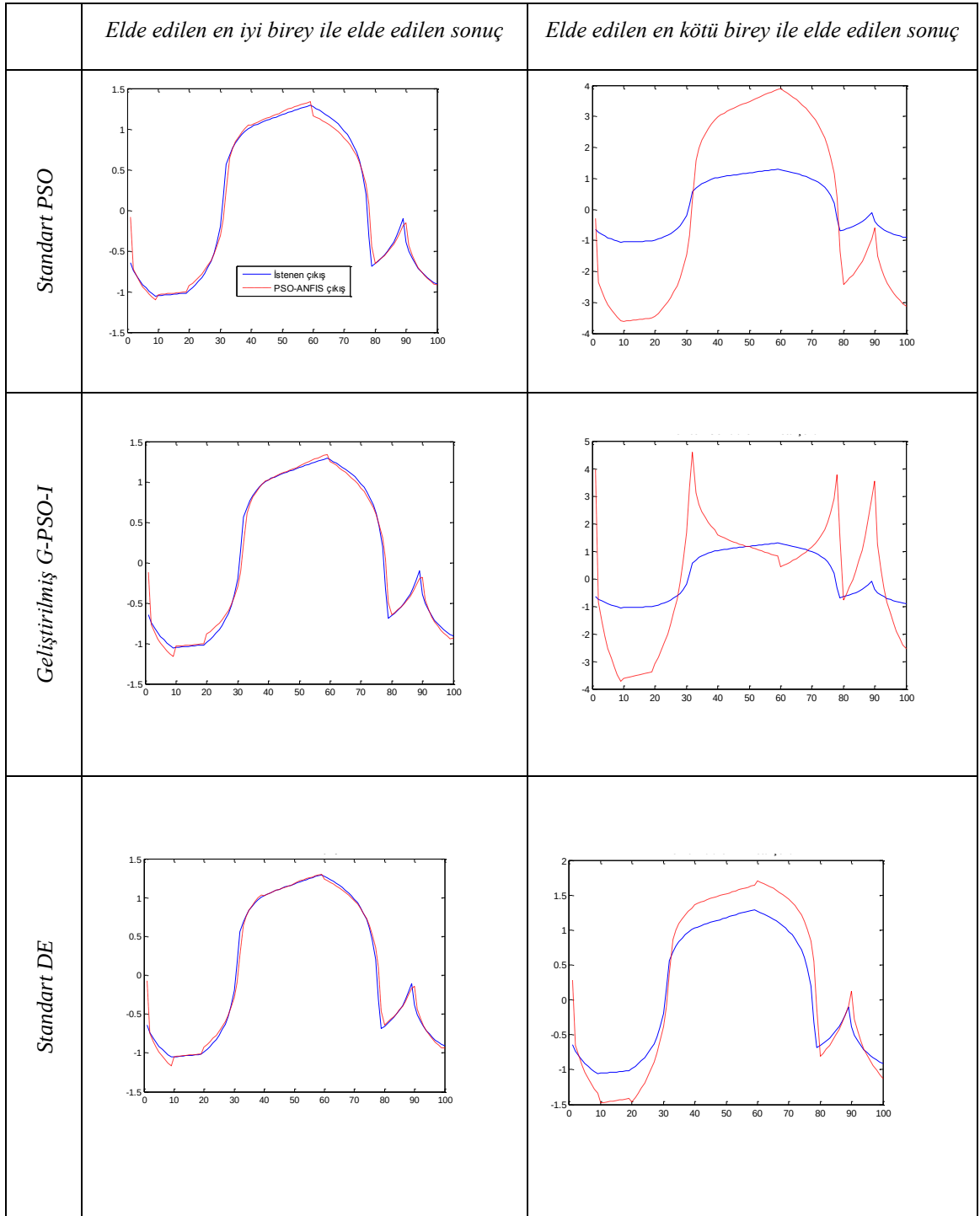
ÖDS 3 için Çizelge 3.1'de verilen fonksiyona uygun olarak ANFIS yapısında kullanılmak üzere Çizelge 3.2'de girişler belirlenmiştir. Belirlenen bu girişlerden veri setini etkileyen $u(k)$ giriş dizisi $[-1 \ 1]$ aralığında rasgele değerler ile oluşturulmuştur. ÖDS 3 için hazırlanan $u(k)$ dizisi grafiği Şekil 3.2 (b)'de eğitim için, Şekil 3.4(b)'de ise test seti için verilmiştir.

ÖDS 3'ün eğitim veri seti oluşturulduktan sonra koşturulan uygulama da ABC, PSO ve DE öğrenmeli ANFIS için Şekil 4.15 ve Şekil 4.16'da elde edilen en iyi birey için en iyi ve en kötü birey için en kötü sonuç grafikleri verilmiştir.

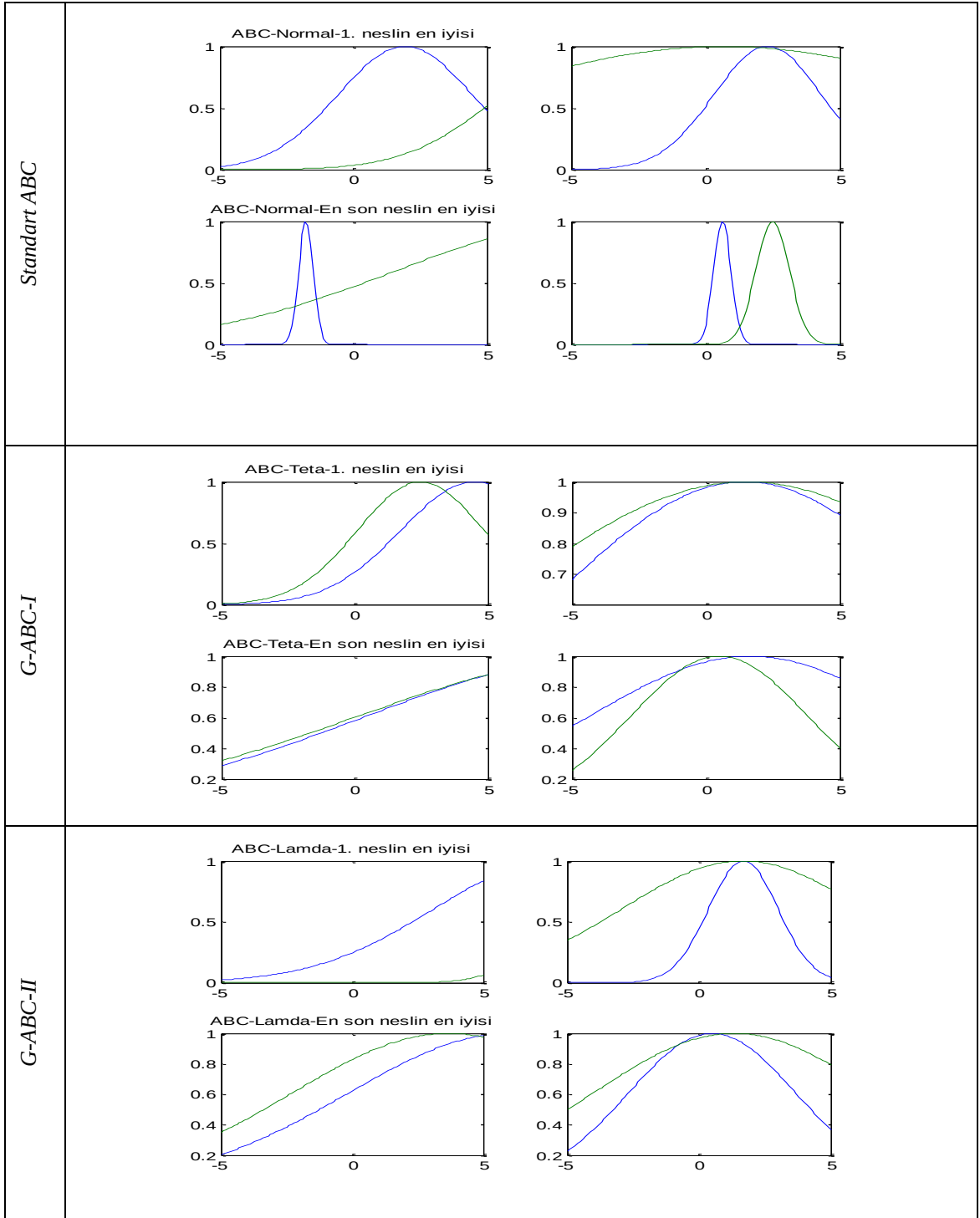
Şekil 4.17'de verilen grafikler ile her bir algoritmanın üyelik fonksiyonları üzerinde uyguladığı değişimler görülebilir. Şekillerde birinci sütunda verilen grafikler 1. girişe, 2.sütunda verilenler ise 2. girişe ait ÜF'lerin ilk ve son durumunu göstermektedir.



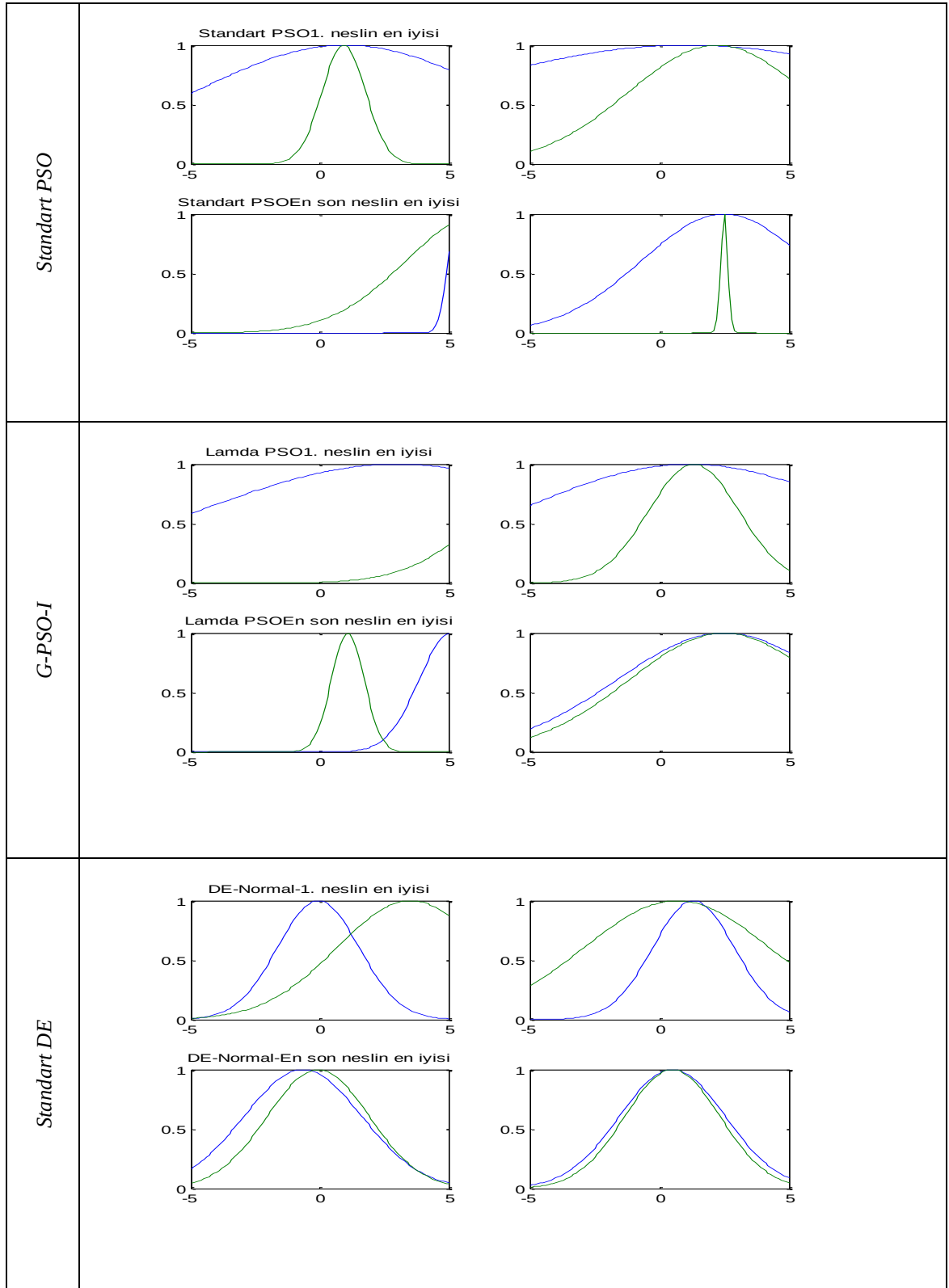
Şekil 4. 15. ABC öğrenmeli ANFIS ile ÖDS 3 için eğitim fazı sistem tanıma sonuçları



Şekil 4. 16. PSO ve DE öğrenmeli ANFIS ile ÖDS 3 için eğitim fazı sistem tanıma sonuçları



Şekil 4. 17. ÖDS 3 için ABC öğrenmeli ANFIS ile eğitim fazı için kullanılan UF'lerin başlangıç ve son durumları



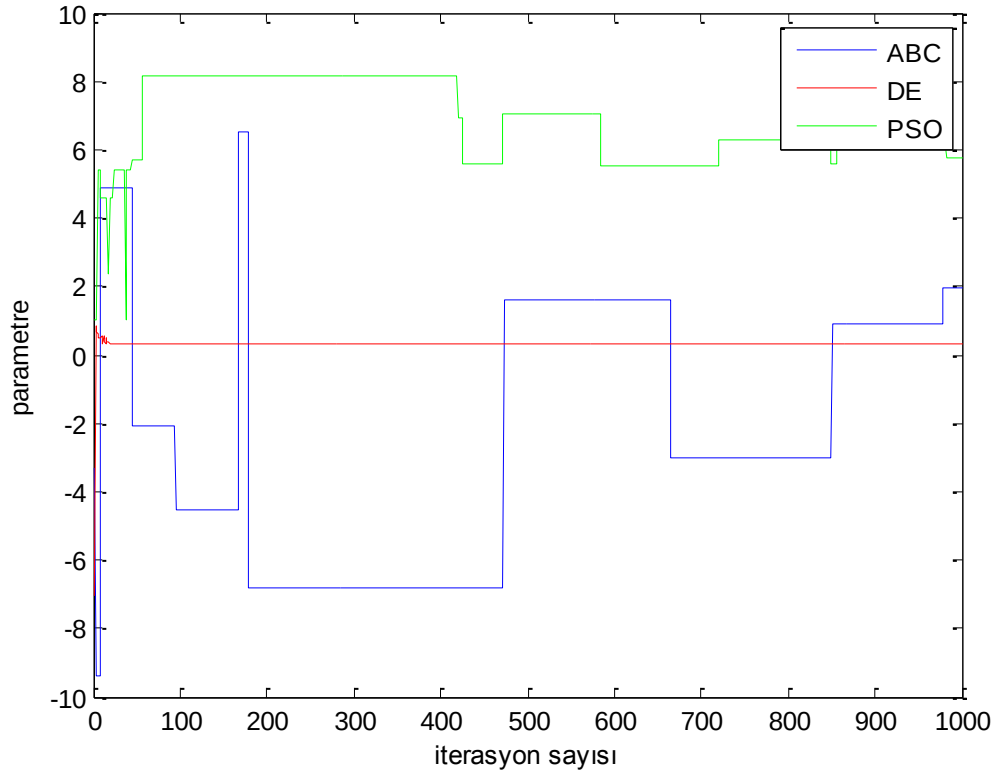
Şekil 4. 18. ÖDS 3 için PSO ve DE öğrenmeli ANFIS ile eğitim fazı için kullanılan UF'lerin başlangıç ve son durumları

ÖDS 3 için standart ABC baz alınarak bir nesil süresince üyelik fonksiyonlarının değişimi Çizelge 4.3’de verilmiştir. Algoritma ANFIS yapısı kullanılarak koşuturulmadan önce belirlenen sınır matrisine göre nasıl bir değişim gösterdiği incelenmiştir.

Çizelge 4. 3. ÖDS 3’ün Standart ABC kullanılarak bir kez koşuturulması sonucu elde edilen ANFIS parametrelerinin değişimi

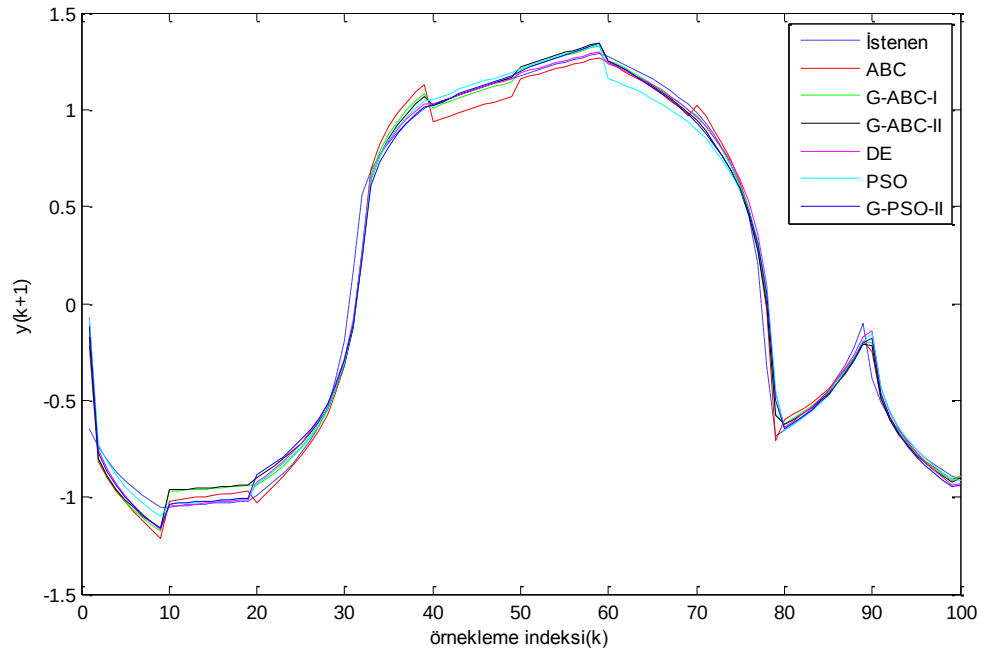
	Parametre	Başlangıç $n=1$	$n=100$	$n=450$	$n=700$	Son değer $n=1000$
1	c_{11}	1,6252036	0,4268606	3,579844	4,3307309	3,9237479
2	σ_{11}	-0,3426217	8,8553989	7,5026601	3,3477184	3,3342794
3	c_{12}	1,3133891	0,5	0,956577	2,4811606	1,1458794
4	σ_{12}	2,2518872	0,5	1,9989136	0,58966	0,5120684
5	c_{21}	-1,2440589	-10	5,718242	-8,2763055	-4,0863074
6	σ_{21}	-1,3074976	10	-8,1885003	-3,6672156	3,7186192
7	c_{22}	1,1594156	-4,4151187	5,5754907	1,6711882	-4,5320006
8	σ_{22}	5,6100491	-6,7474112	-5,7529214	-4,6851029	-4,2707841
9	p_1	-8,7864205	1,7794449	1,3950953	-4,1118179	-3,8335696
10	q_1	8,9131357	-3,0783714	-3,8635145	6,8137336	7,6619083
11	r_1	9,9454546	-10	-3,0361178	-2,7233718	-6,8464156
12	p_2	-4,46167	-4,0769138	4,2671801	5,2338731	-3,4525737
13	q_2	3,6720173	-0,1102185	2,9626164	-9,9542835	-5,8038276
14	r_2	-2,4114026	3,8492758	-7,1599891	-2,8820805	-3,1545889
15	p_3	-3,3029163	-4,5511468	-6,8356302	-3,0171915	1,9820433
16	q_3	3,0190976	1,4395955	0,679937	-4,1181314	2,5587251
17	r_3	-5,9029941	10	9,4790501	8,5183038	7,037454
18	p_4	7,8149457	5,7803697	9,6441508	-3,6725254	-4,6529015
19	q_4	-7,3630038	1,5106851	0,3651443	0,1158864	-4,28265
20	r_4	-0,8069841	-3,7051246	0,7118615	0,3187111	2,9536664

ÖDS 3 için eğitim boyunca 15. elemanı yani bulanık kontrolörün 3. kuralının p parametresinin değişimi standart ABC, PSO ve DE ile eğitim seyri hakkında fikir sahibi olunması amacıyla Şekil 4.19’da verilmiştir.

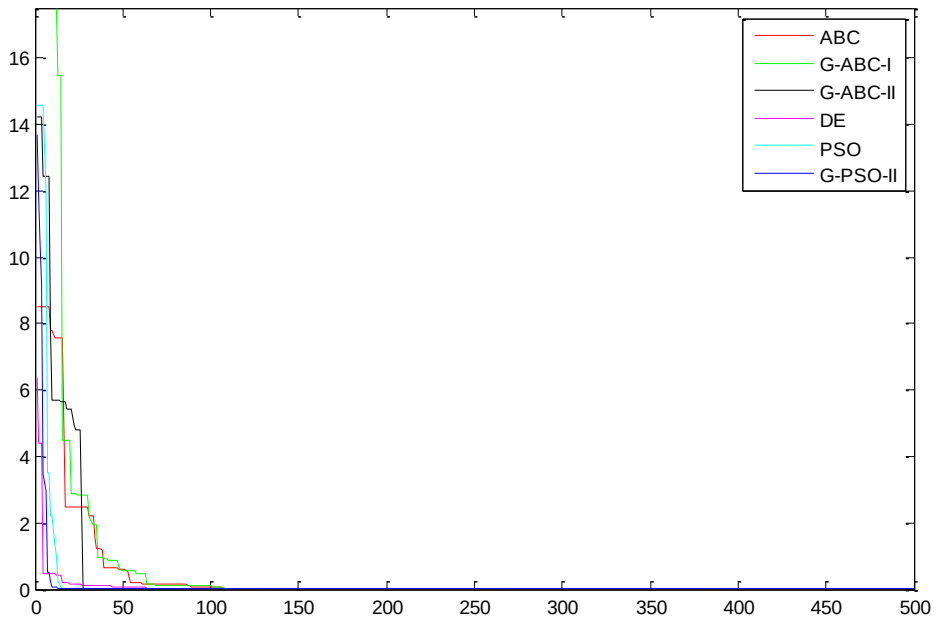


Şekil 4. 19. ANFIS'in bir parametresinin eğitim süresince değişim seyri

ÖDS 3'ün ANFIS ile modellenmesi için Standart ABC, G-ABC-I, G-ABC-II, Standart PSO, G-PSO-I ve Standart DE ile ayrı ayrı koşturulması sonucunda elde edilen iyi ölçüt değerine sahip koşmanın en iyi bireyi ile elde edilen modelleme sonuçları kıyaslanmıştır. Şekil 4.20'de Standart ABC, G-ABC-I, G-ABC-II, Standart PSO, G-PSO-I ve Standart DE için elde edilen en iyilerin karşılaştırılması verilmiştir. Şekil 4.20'de algoritmaların ÖDS 3 için elde ettikleri performansı incelendiğinde tüm algoritmaların istenene yakın cevaplar verdiği görülmektedir. ÖDS 3 için algoritmaların eğitim fazında yakınsama hızları verilmiştir. Şekilde verilen eğitim seyri her bir algoritma için ayrı ayrı 50'şer koşmanın (en iyi bireyler için) ortalaması alınarak elde edilmiştir. Şekilden görüleceği üzere, DE, PSO ve G-PSO-I 'nun çabuk yakınsama göstermiştir.



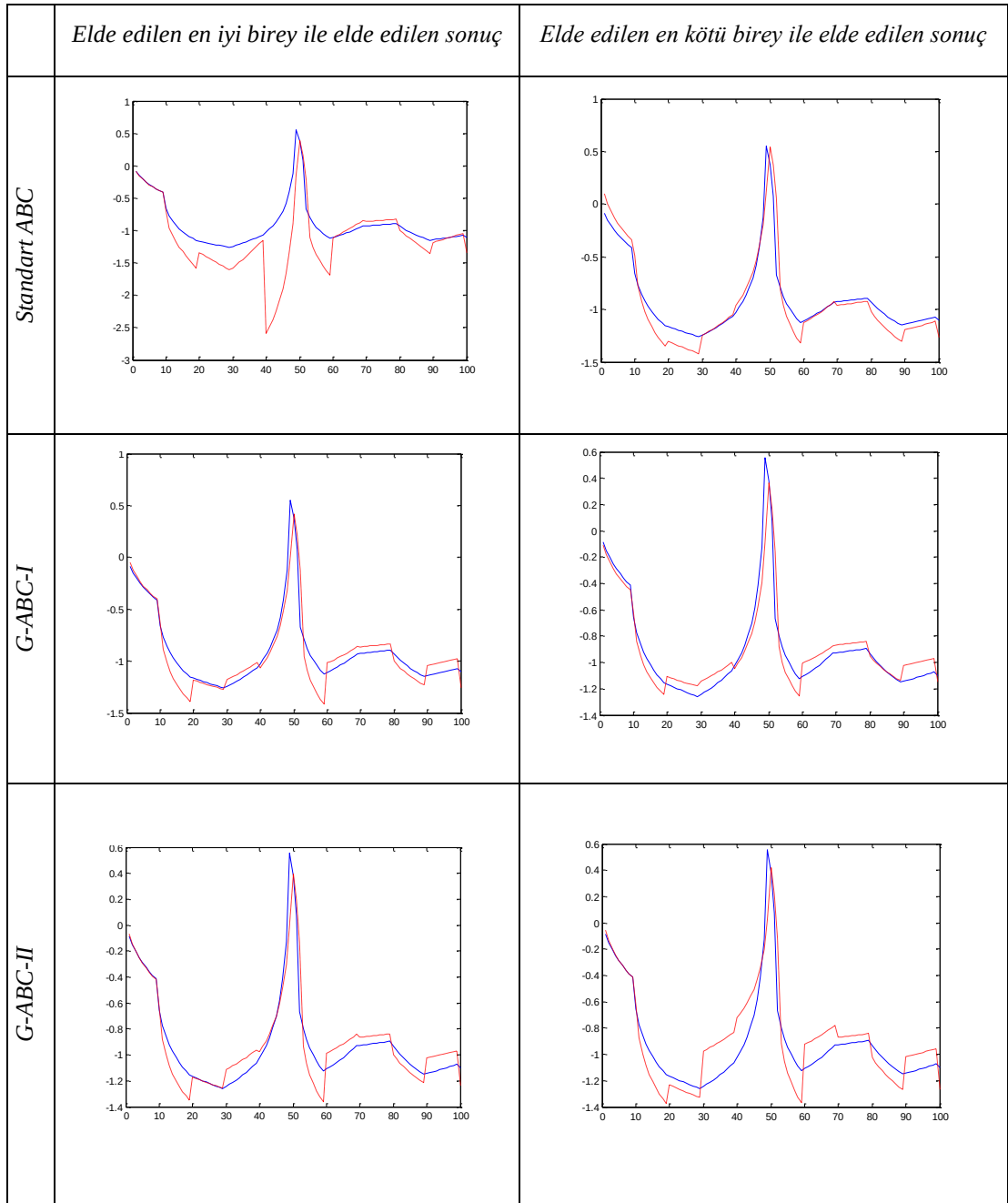
Şekil 4. 20. ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 3 için elde edilen modellerin karşılaştırılması



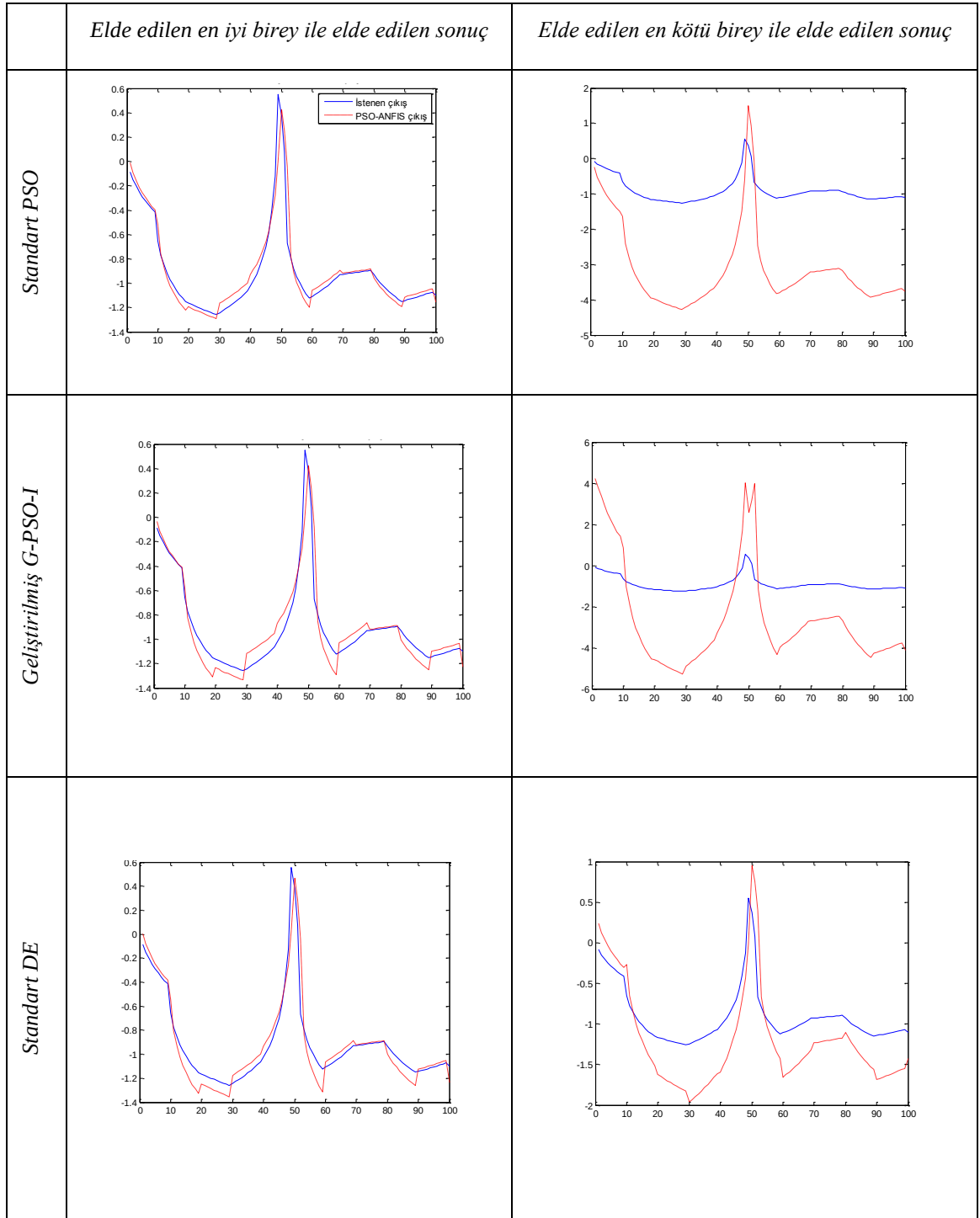
Şekil 4. 21. ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 3 için modelleme eğitim seyri

4.3.2.ÖDS 3'ün test aşaması ve sonuçları

ÖDS 3 için Çizelge 3.1'deki ÖDS 3 fonksiyonunda kullanılan $u(k)$ dizisinin rasgele değer atanması sonucunda eğitim sisteminde kullanılan örnekten daha farklı bir örnek elde edilmiş ve elde edilen yeni veri seti test için kullanılmıştır. Şekil 4.22 ve Şekil 4.23'de uygulamada kullanılan algoritmaların koşturulması sonucunda test fazında elde edilen en iyi bireyin sonuçları verilmiştir.



Şekil 4. 22. ABC öğrenmeli ANFIS ile ÖDS 3 için test fazı sistem tanıma sonuçları



Şekil 4. 23. PSO ve DE öğrenmeli ANFIS ile ÖDS 3 için test fazı sistem tanıma sonuçları

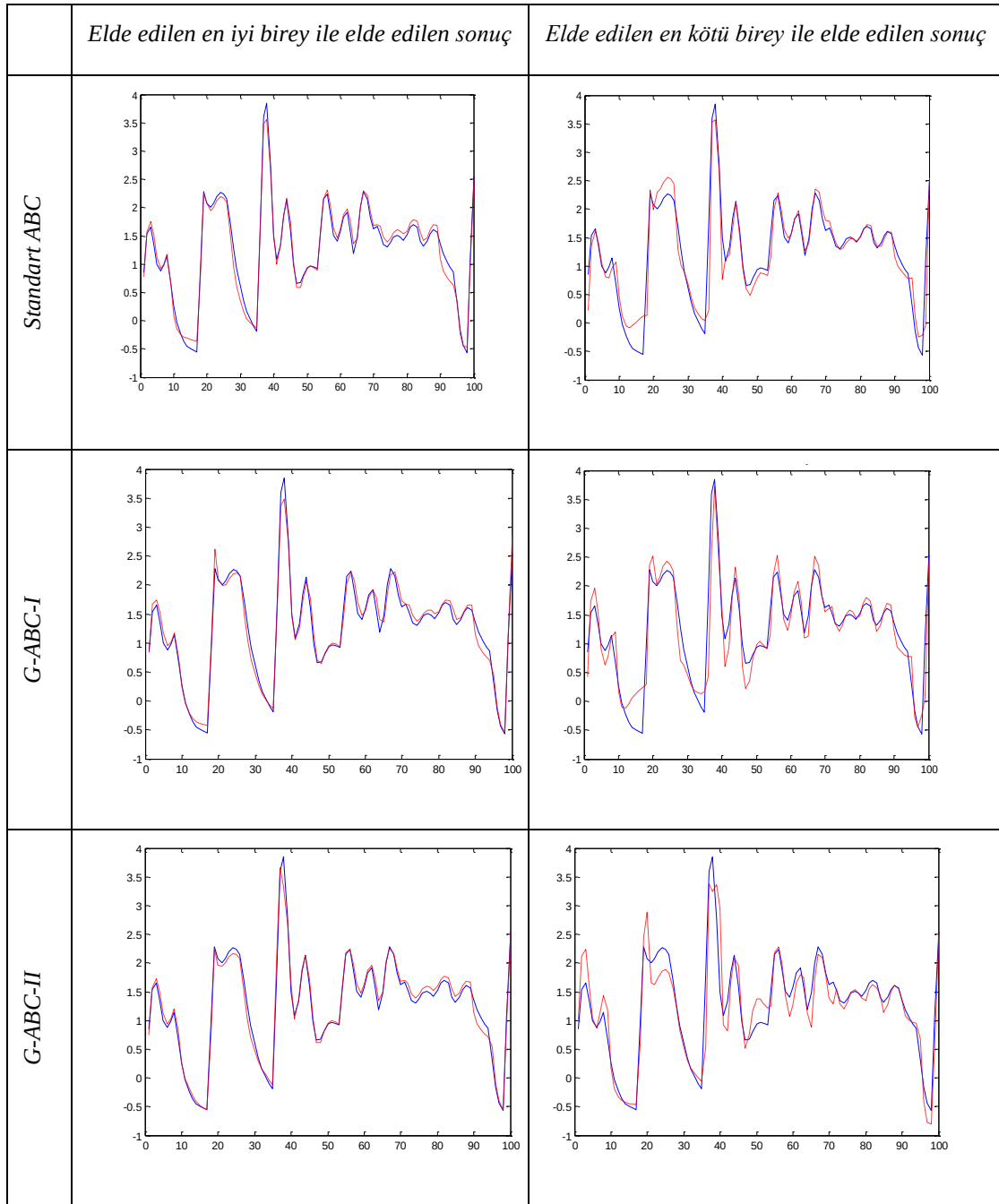
4.4. ÖDS 4'ün Tanınması/Modellenmesi

4.4.1. ÖDS 4 için eğitim aşaması ve sonuçları

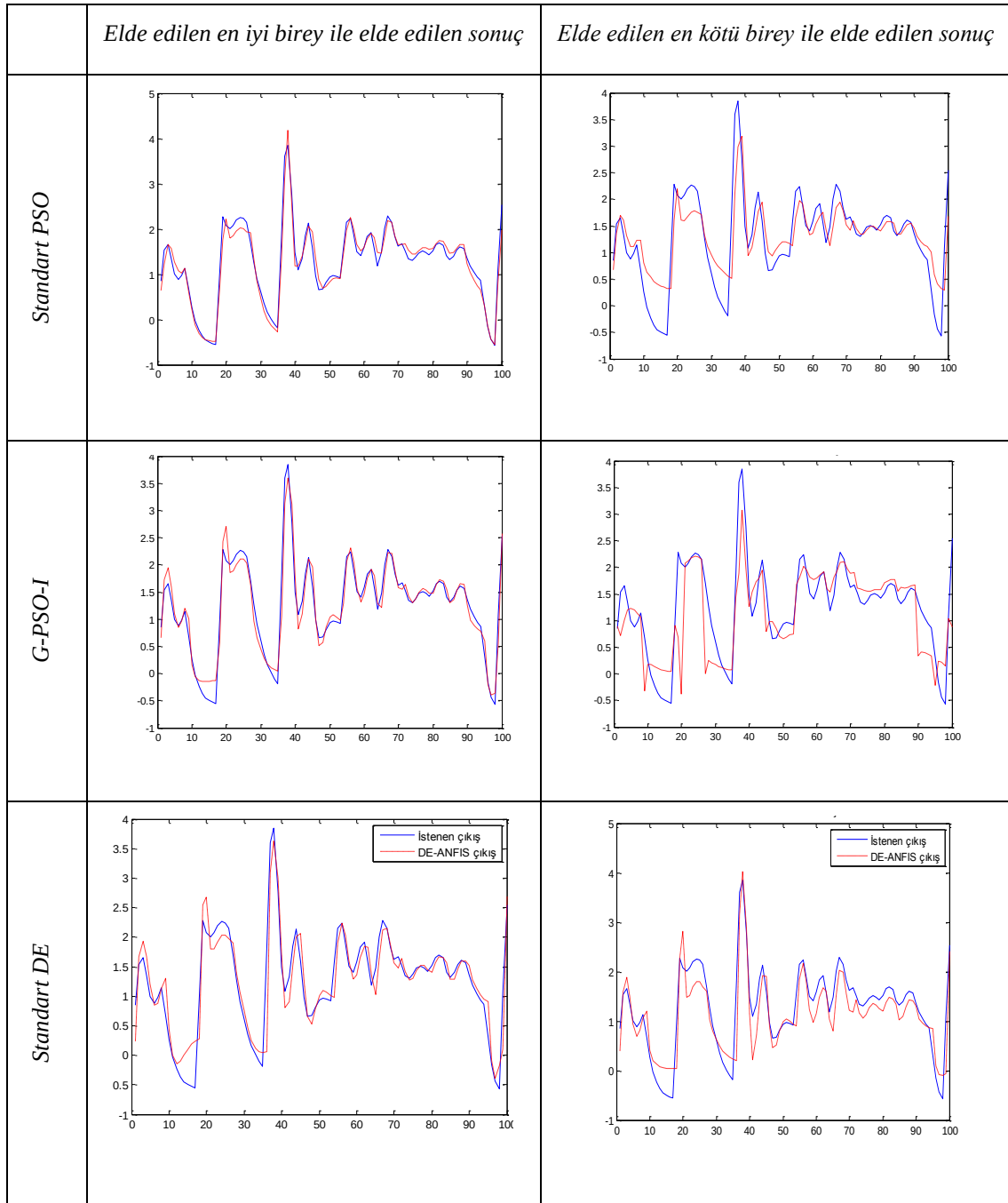
ÖDS 4 için Bölüm 3 Çizelge 3.1'de verilen sistem kullanılmıştır. Veri seti oluşturulmadan önce girişlerden önemli bir yere sahip olan $u(k)$ dizisi oluşturulmuştur. $u(k)$ dizisi oluşturulurken rasgele $[-5 \ 5]$ aralığında ve $[1 \ 20]$ darbe genliği dikkate alınarak, Şekil 3.2(c)'deki grafik elde edilmiştir.

ÖDS 4 için eğitim veri seti sırasıyla ABC, PSO ve DE kullanılarak 50'şer kez koşturulmuştur. Uygulama sonucunda sistem tanıma sonuçlarına göre Şekil 4.24 ve Şekil 4.25'de elde edilen en iyi birey için en iyi ve en kötü birey için en kötü sonuç grafikleri verilmiştir.

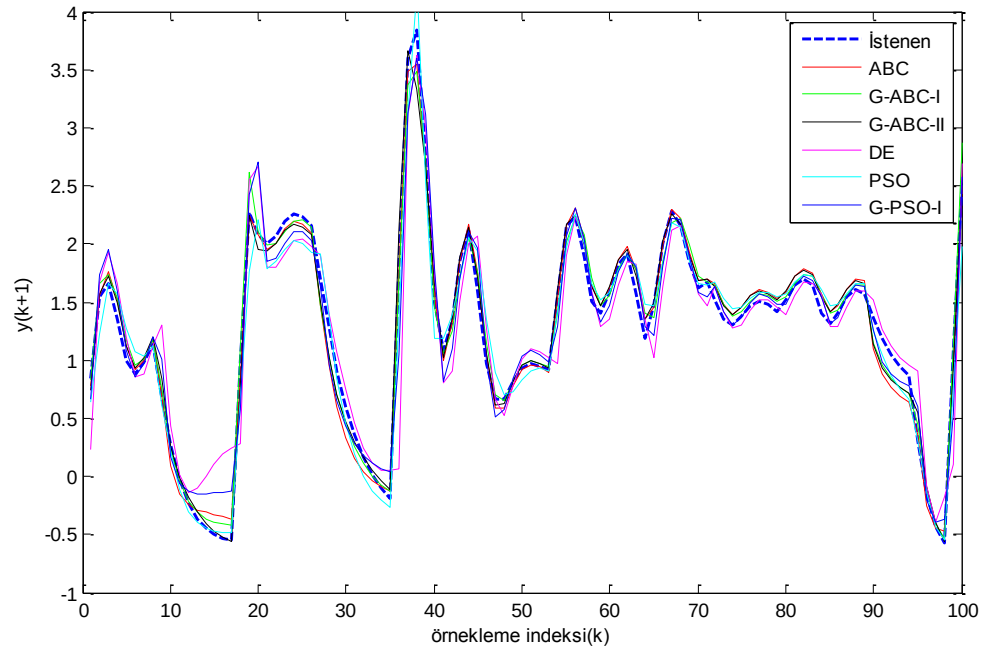
ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 4'ün eğitim fazının sistem tanıma sonuçlarına göre elde edilen modellerin karşılaştırılması Şekil 4.26'da verilmiştir. ÖDS 4 için tüm algoritmaların koşturulması sonucunda ANFIS öğrenmeli modelleme eğitim seyri Şekil 4.27'de verilmiştir. Bu şekle göre PSO ve türevlerinin çabuk yakınsadığı görülmektedir.



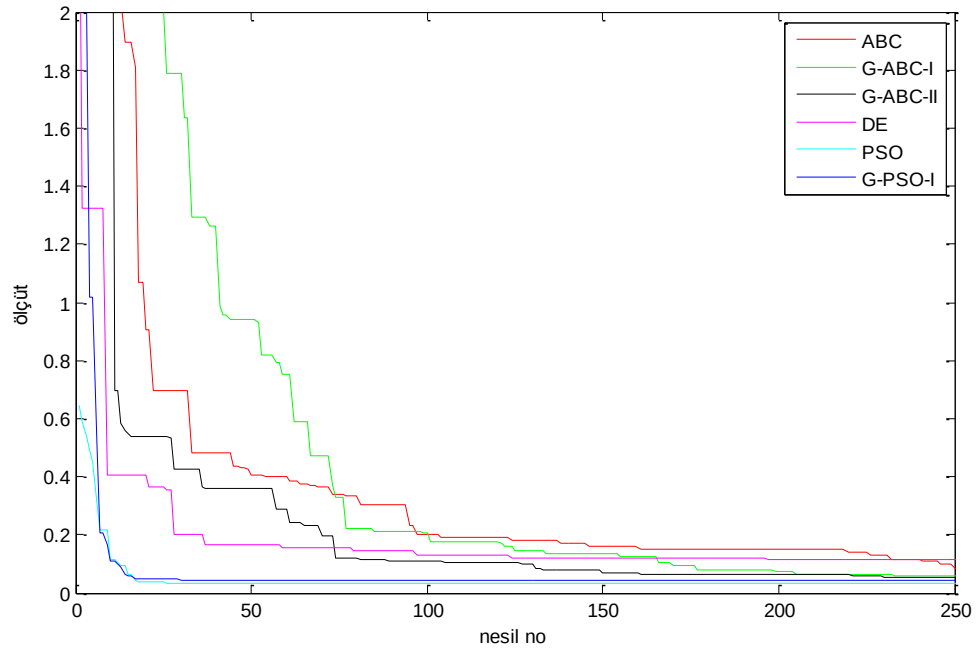
Şekil 4. 24. ABC öğrenmeli ANFIS ile ÖDS 4 için eğitim fazı sistem tanıma sonuçları



Şekil 4. 25. PSO ve DE öğrenmeli ANFIS ile ÖDS 4 için eğitim fazı sistem tanıma sonuçları



Şekil 4. 26. ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 4 için elde edilen modellerin karşılaştırılması.

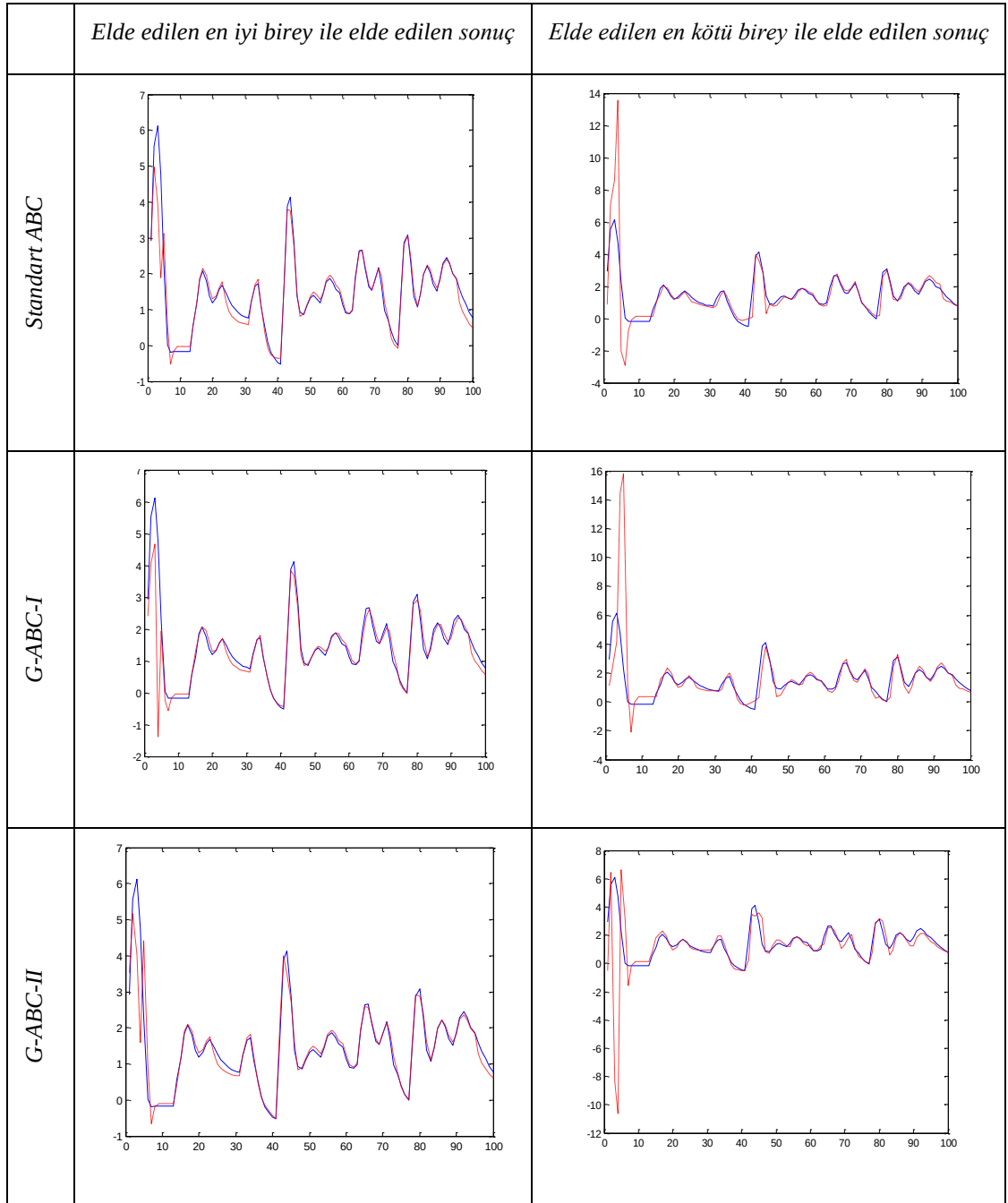


Şekil 4. 27. ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 4 modelleme eğitim seyri

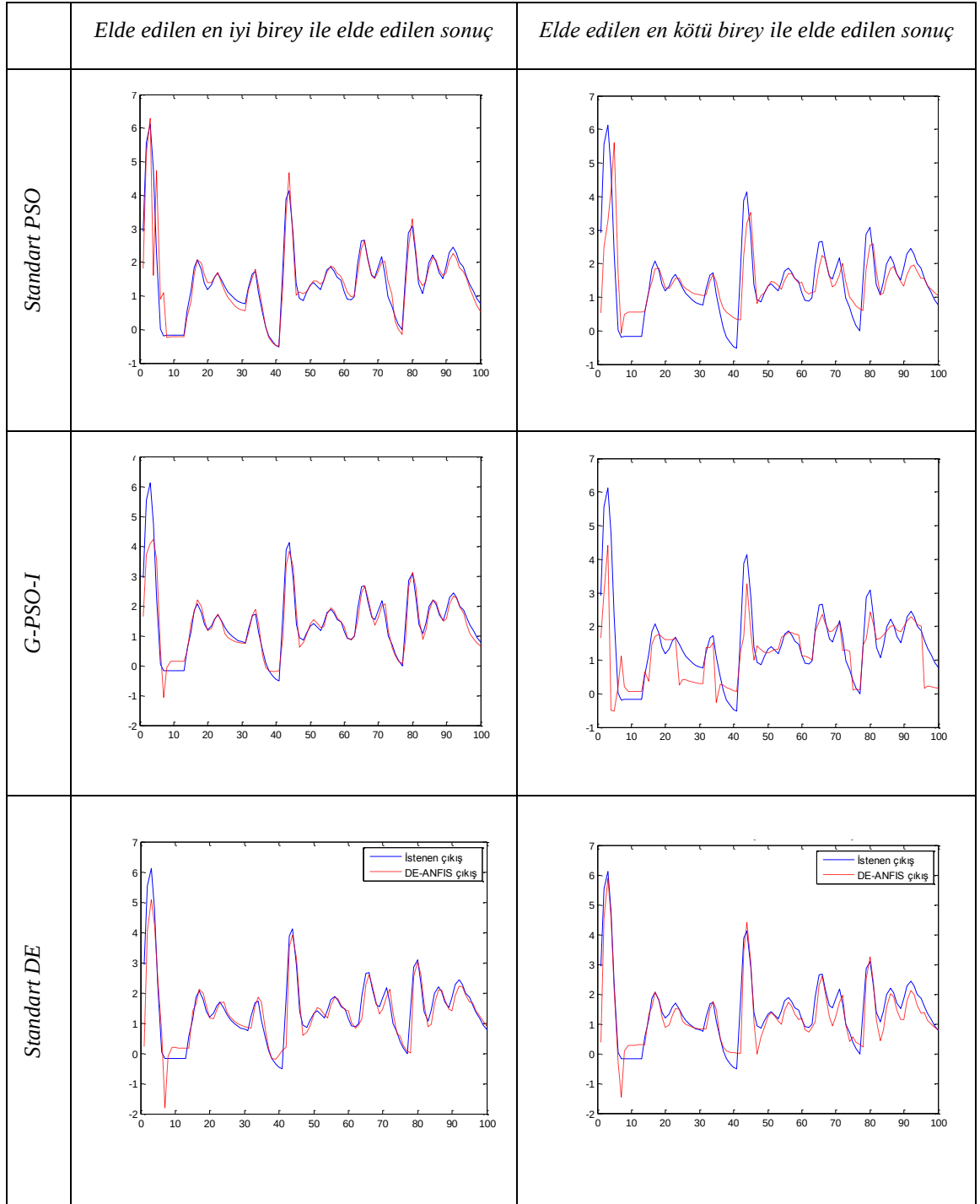
4.4.2. ÖDS 4'ün test aşaması ve sonuçları

ÖDS 4 için test veri seti hazırlanabilmesi için $u(k)$ dizisinin rasgele değer atanması sonucu eğitim sisteminde kullanılan örnekten daha farklı bir örnek elde edilmiştir. Test veri seti kullanılarak standart ABC, G-ABC-I, G-ABC-II, Standart PSO, G-PSO-I ve Standart DE için ÖDS 4 koşturulduğunda elde edilen en iyi birey için en iyi ve en kötü birey için en kötü test sonucu Şekil 4.28 ve Şekil 4.29'da verilmiştir.

Şekil 4.28 ve Şekil 4.29 detaylı olarak incelendiğinde; ABC, G-ABC-I, G-ABC-II, Standart PSO, G-PSO-I ve Standart DE'nin ANFIS öğrenmeli test fazı sonuçlarında hiçbir algoritmanın gerçeğe yakın sonuç vermediği görülmektedir.



Şekil 4. 28. ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 4 için test fazı sistem tanıma sonuçları



Şekil 4. 29. ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 4 için test fazı sistem tanıma sonuçları

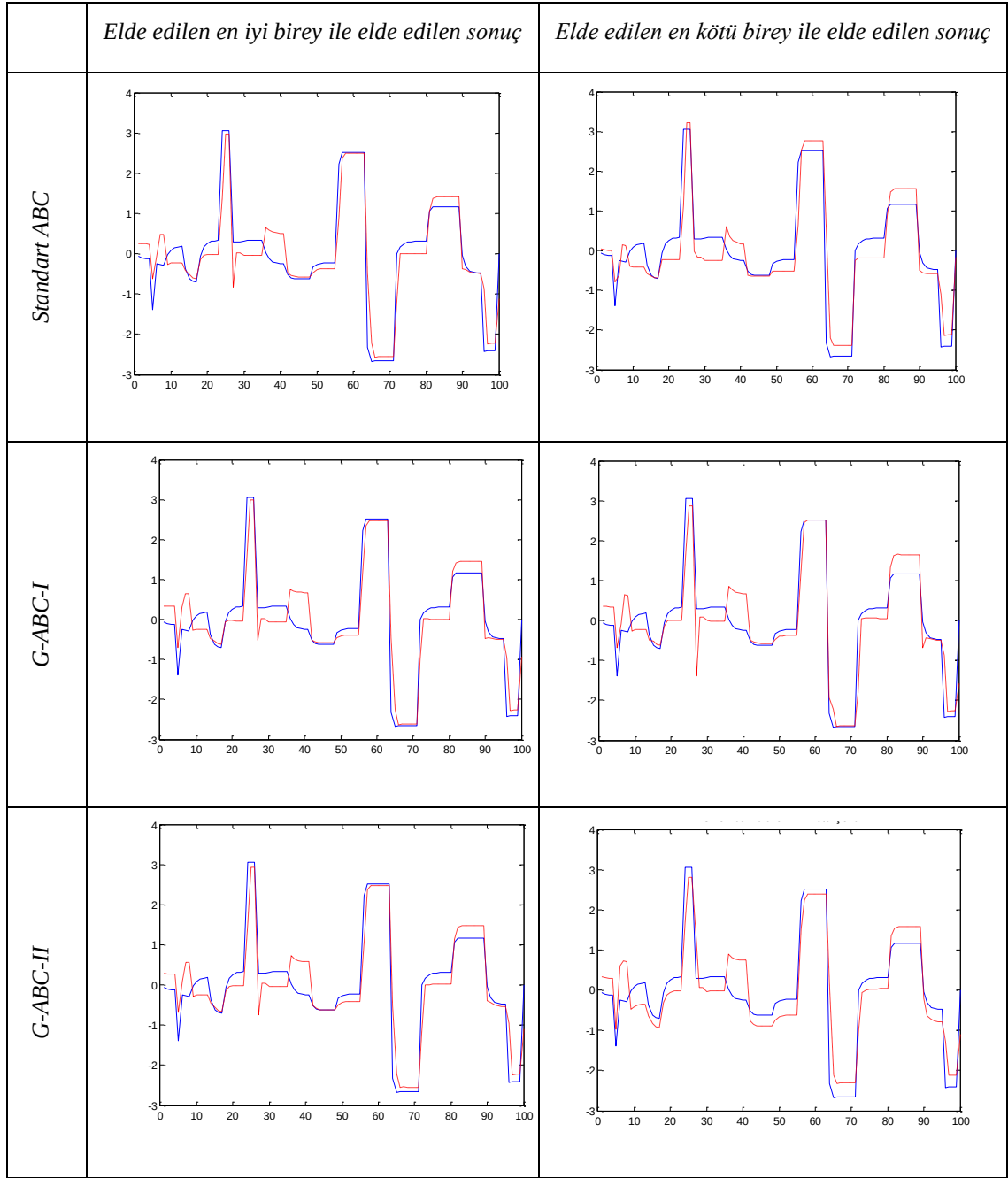
4.5. ÖDS 5'in Tanınması/Modellenmesi

4.5.1. ÖDS 5'in eğitim aşaması ve sonuçları

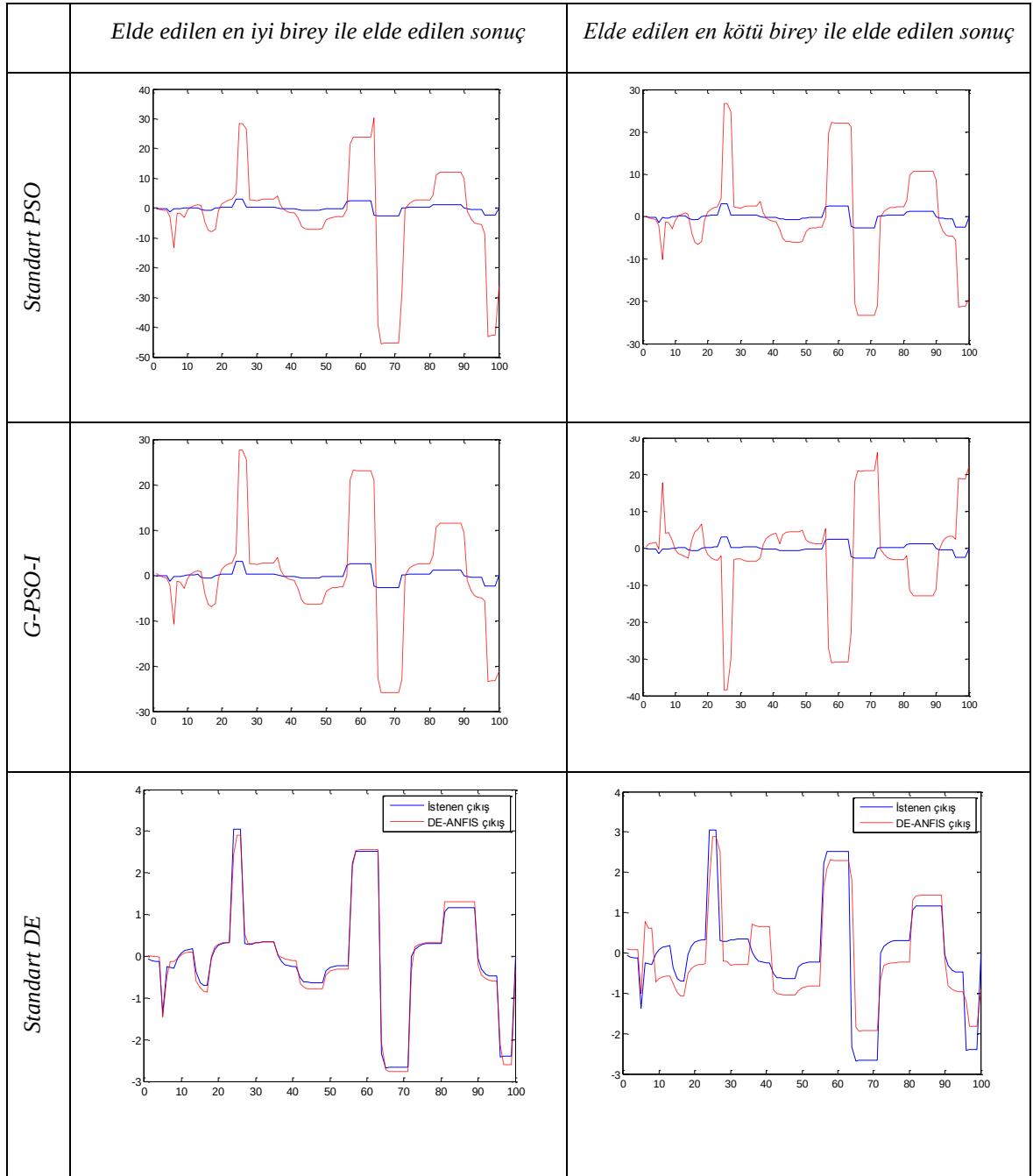
ÖDS 5 için, Bölüm 2 Çizelge 3.2'de verilen giriş değişkenleri ve Şekil 3.2(d)'de verilen $u(k)$ dizisi kullanılarak eğitim veri seti hazırlanmıştır. Buna göre $u(k)$ dizisi oluşturulurken rasgele $[-5 \ 5]$ aralığında ve $[1 \ 20]$ darbe genliği dikkate alınarak Şekil 3.3(d)'deki $u(k)$ giriş dizisi elde edilmiştir. ÖDS 5 için eğitim veri seti kullanılarak sırasıyla ABC, PSO ve DE algoritmaları ANFIS optimizasyonu için kullanılarak 50'er kez koşturulmuştur.

Uygulama da ABC, PSO ve DE algoritmasının sistem tanıma sonuçları Şekil 4.30 ve Şekil 4.31'de, elde edilen en iyi birey ve en kötü birey için sonuç grafikleri verilmiştir. Sistem tanıma sonucuna ait grafikler sırasıyla incelendiğinde istenen çıkışa göre en yakın cevabı standart DE'nin verdiği diğer algoritmaların ise kötü olduğu görülmektedir.

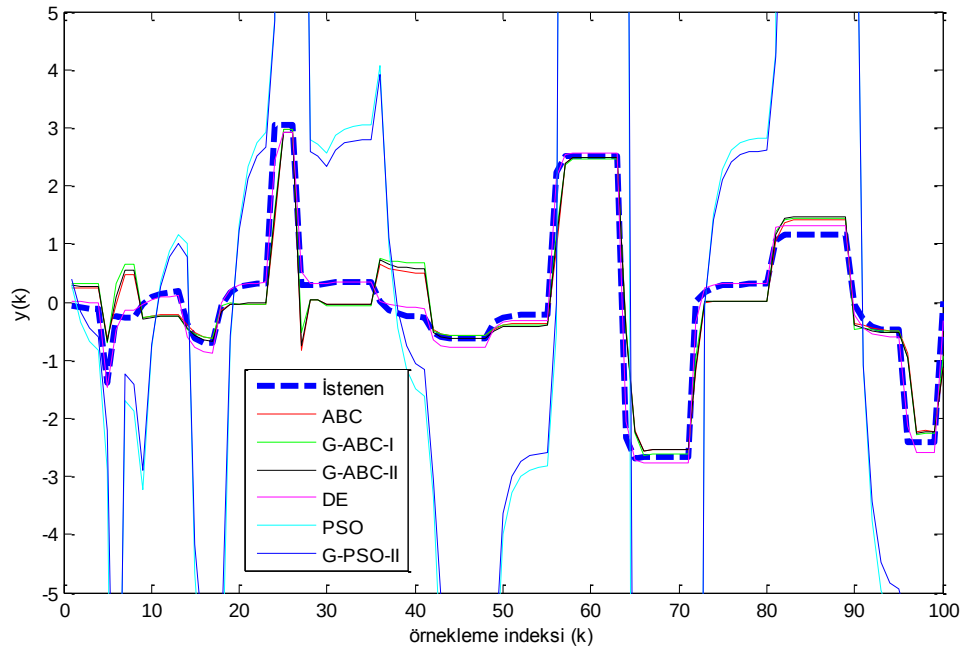
ÖDS 5 'in ANFIS modellenmesi için koşturulan standart ABC, G-ABC-I, G-ABC-II, Standart PSO, G-PSO-I ve Standart DE ile ayrı ayrı koşturulması sonucunda elde edilen en iyi ölçüt değerine sahip koşmanın en iyi bireyi ile elde edilen modelleme sonuçları kıyaslanmıştır. İstenen çıkışa bakıldığında ABC ve DE algoritmaların daha iyi sonuç verdiği gözlenmiş, PSO algoritmaları istenene göre fazlaca uzakta olduğu görülmüştür. Ayrıca her bir algoritmanın ANFIS yapısı kullanılarak koşturulması sonucunda elde edilen modelleme eğitim seyri Şekil 4.33'de ayrıntılı bir şekilde verilmiştir.



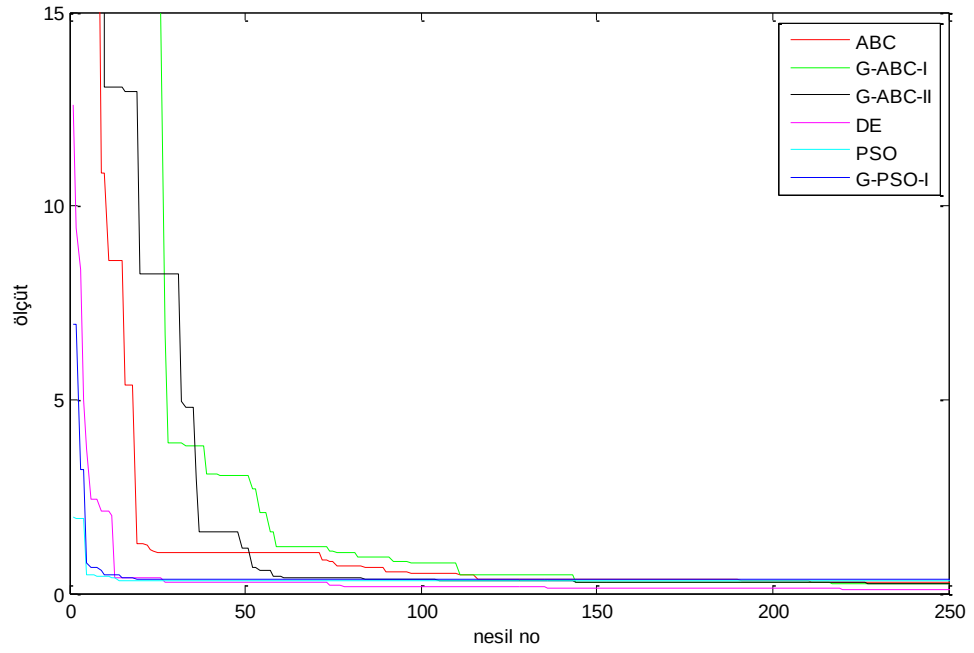
Şekil 4. 30. ABC öğrenmeli ANFIS ile ÖDS 5 için eğitim fazı sistem tanıma sonuçları



Şekil 4. 31. PSO ve DE öğrenmeli ANFIS ile ÖDS 5 için eğitim fazı sistem tanıma sonuçları



Şekil 4. 32. ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 5 için elde edilen modellerin karşılaştırılması.



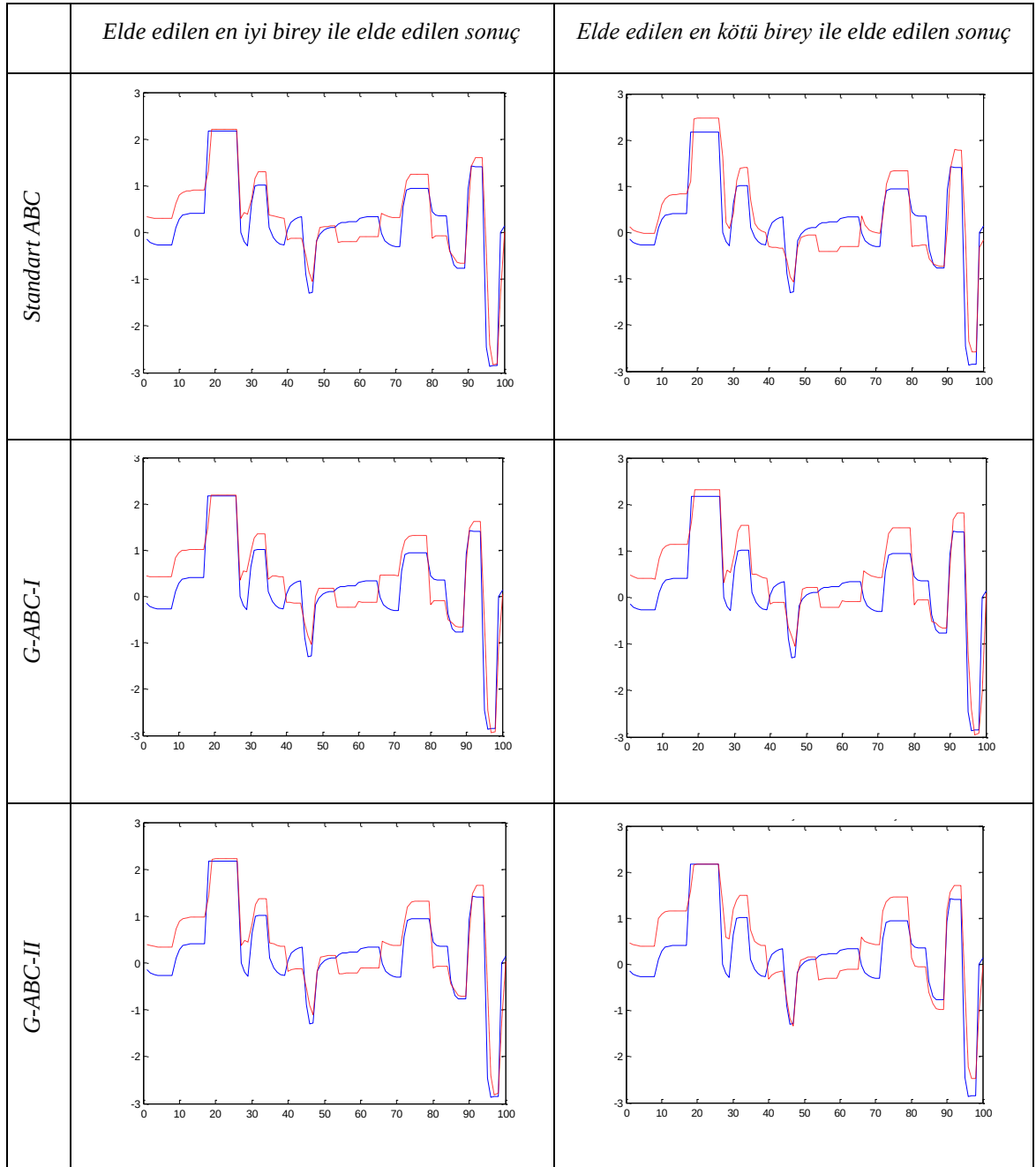
Şekil 4. 33. ABC, PSO ve DE öğrenmeli ANFIS ile ÖDS 5 modelleme eğitim seyri

4.5.2. ÖDS 5'in test aşaması ve sonuçları

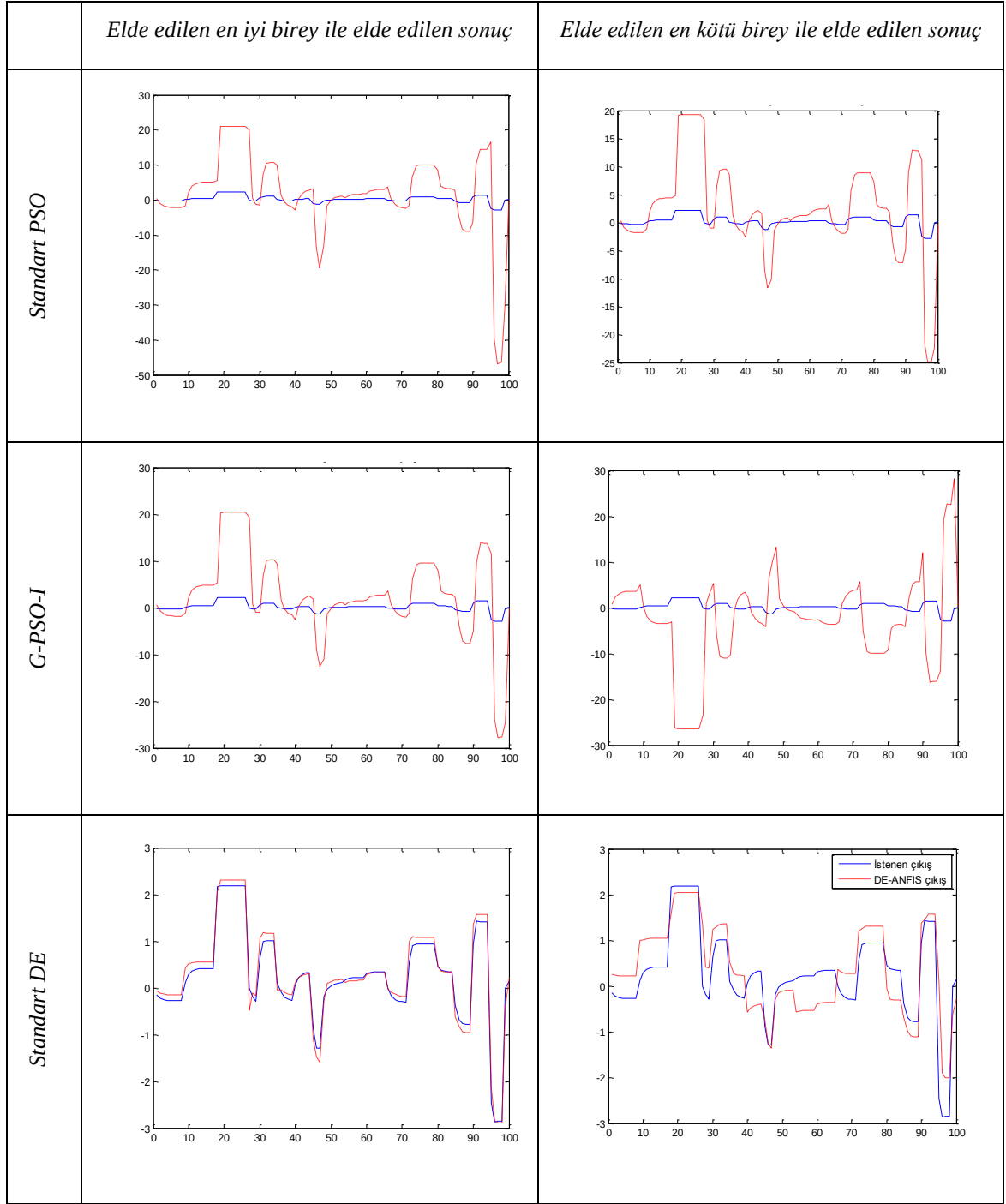
ÖDS 5 için $[-2 \ 2]$ Aralığında rasgele genlikli ve rasgele örnekleme periyotlu $u(k)$ dizisi Şekil 3.3(d)'de verilmiştir. Oluşturulan veri (test seti) seti aracılığı ile ÖDS 5'in ABC, PSO ve DE öğrenmeli ANFIS modelindeki çıkış cevapları ve performansı incelenmiştir. ABC algoritması kullanılarak elde edilen çıkış cevapları PSO ve DE algoritmaları ile elde edilen cevaplarla karşılaştırılmıştır.

Oluşturulan test seti ile ABC, PSO ve DE algoritmalarıyla elde edilen ÖDS 5 ANFIS modellerinin sonuçları Şekil 4.34 ve Şekil 4.35'de sırayla tüm algoritmaları kapsayacak şekilde verilmiştir.

Verilen şekillere bakıldığında hiçbir algoritmanın ÖDS 5 test fazı için istenen sonuca yakın bir cevap vermediği gözlenmiştir.



Şekil 4. 34. ABC öğrenmeli ANFIS ile ÖDS 5 için test fazı sistem tanıma sonuçları



Şekil 4. 35. ABC öğrenmeli ANFIS ile ÖDS 5 için test fazı sistem tanıma sonuçları

4.6. PSO ve DE Algoritmaları ile ABC'nin Performansının Kıyaslanması

ABC algoritmasının bulanık mantık kullanılarak dinamik modellemede başarımını değerlendirmek için Bölüm 3'de Çizelge 3.1'de verilen beş farklı sistem kullanılmıştır. Çizelge 3.1'de verilen her bir sistem için standart ABC algoritması ile 50'şer kez koşturulmuştur. Bunun yanında standart ABC algoritmasında değişiklikler yapılarak elde edilen G-ABC-I ve G-ABC-II algoritmaları da kullanılmıştır. Modellemede ABC algoritmasının başarımını değerlendirmesi için hem eğitim hem de test fazı için üç farklı ABC algoritması koşturulmuştur. ABC algoritmalarıyla elde edilen sonuçlar, literatürde çok kullanılan Parçacık Sürü Optimizasyon (PSO) ve Farksal Gelişim (DE)'e ait sonuçlarla kıyaslanmıştır. Bu algoritmalara ait bilgilere Bölüm 3'de verilmiştir.

Başarım kıyaslamasında her bir sistem için eğitim fazında kullanılan algoritmaların popülasyon büyüklükleri ve maksimum nesil sayıları aynı alınmıştır. Koşturulan algoritmaların eğitim seti için başarım kıyaslaması Çizelge 4. 4'de, test seti için başarım kıyaslaması Çizelge 4. 5'de verilmiştir.

Çizelge 4. 4 ile yapılan kıyaslamada her bir sistem için sırasıyla ortalama, standart sapma, en iyi, en kötü, 1 nesil süresi, başarım sayısı olmak üzere sonuçlar 6 ayrı kategoride istatistiki olarak irdelenmiştir. Bu kategorileri sırasıyla aşağıdaki gibi tanımlayabiliriz:

- Ortalama: 50 ayrı koşma sonunda bulunan eğitim ölçüt değerlerinin ortalaması.
- Standart sapma: 50 ayrı koşma sonunda bulunan eğitim ölçüt değerlerinin standart sapması.
- En iyi: 50 ayrı koşma sonucu elde edilen ölçütler içinde en iyi ölçüt değeri.
- En kötü: 50 ayrı koşma sonucu elde edilen ölçütler içinde en kötü ölçüt değeri.
- 1 nesil süresi: 50 ayrı koşma sonunda bulunan 1 nesil süresi.

Çizelge 4. 4'de sonuçlara göre ABC algoritması PSO ve DE algoritmasıyla kıyaslandığında; ABC ve DE algoritmasının en iyi başarımı gösterdiği, PSO algoritmasının ise diğer iki algoritmaya göre daha kötü başarım sergilediği görülmektedir. 1 nesil süresi kategorisinde DE, PSO ve ABC karşılaştırıldığında en iyi koşma zamanı ABC algoritmasına aittir. Bu sonuca dayalı olarak, gerçek zamanlı

uygulamalarda ABC, DE ve PSO kullanımına göre daha elverişli olduğu anlaşılmaktadır. ABC algoritmasının diğer algoritmalara göre daha basit bir yapısı ve kontrol parametre sayısının az olması da algoritmaya bir üstünlük kazandırmaktadır.

Çalışmada ABC algoritmasının bulanık ağ yapısı üzerinde başarımını daha iyi inceleyebilmek için farklı veri setleri (test seti) kullanılarak her bir sistem için elde edilen ANFIS modeli test edilmiştir. Bu veri setleri kullanılarak ABC, PSO ve DE algoritmaları için elde edilen sonuçlar Çizelge 4. 5’de verilmiştir. Elde edilen sonuçlara göre genel olarak bulanık sistem modellemede DE'nin daha başarılı olduğu ancak ABC ve türevlerinin başarımlarının DE'nin başarımına çok yakın olduğu sonucuna varılmıştır. İncelemede ele alınan PSO algoritmasının başarımının ise ABC ve DE'nin başarımlarına göre daha kötü olduğu görülmüştür.

Çizelge 4. 4. ABC, PSO ve DE algoritmalarının ÖDS modellemeye *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)
ABC	Ortalama	0.036994	0.045392	0.012165	0.050564	0.255944
	Std sapma	0.012073	0.032781	0.013409	0.019603	0.014218
	En iyi	0.013664	0.003924	0.004422	0.01568	0.236442
	En kötü	0.058886	0.145957	0.056209	0.092783	0.289824
	1 nesil süresi (s)	0.026451	0.02701	0.015753	0.024448	0.016379
	Başarım sayısı	1	1	0	3	0
	Ortalama Başarım Sayısı	5/25=0.20				
G-ABC-I	Ortalama	0.036842	0.040892	0.016855	0.050983	0.246771
	Std sapma	0.011271	0.022851	0.013653	0.018154	0.00849
	En iyi	0.016055	0.009197	0.005407	0.015181	0.235473
	En kötü	0.071235	0.117495	0.092333	0.106934	0.271231
	1 nesil süresi (s)	0.024406	0.025045	0.015585	0.024542	0.015048
	Başarım sayısı	2	0	0	0	4
	Ortalama Başarım Sayısı	6/25=0.24				
G-ABC-II	Ortalama	0.038039	0.046537	0.018718	0.052317	0.253359
	Std sapma	0.012515	0.025984	0.012315	0.016816	0.014552
	En iyi	0.015961	0.012833	0.004421	0.0135	0.235528
	En kötü	0.060354	0.111644	0.066238	0.105594	0.30777
	1 nesil süresi (s)	0.025276	0.0247	0.015538	0.025028	0.015218
	Başarım sayısı	0	1	1	1	0
	Ortalama Başarım Sayısı	3/25=0.12				
PSO	Ortalama	0.079258	0.030886	0.212148	0.120247	0.413184
	Std sapma	0.063467	0.027827	0.568219	0.043797	0.044641
	En iyi	0.027028	0.008052	0.005427	0.03366	0.319359
	En kötü	0.437995	0.197535	3.759978	0.220542	0.512254
	1 nesil süresi (s)	0.047024	0.045804	0.02863	0.047661	0.031011
	Başarım sayısı	0	0	0	0	0
	Ortalama Başarım Sayısı	0				
G-PSO-I	Ortalama	0.084378	0.023013	0.172979	0.114145	0.449172
	Std sapma	0.056409	0.011735	0.460139	0.048272	0.090489
	En iyi	0.028781	0.009869	0.003724	0.043578	0.350809
	En kötü	0.296594	0.050653	1.96447	0.311882	0.820956
	1 nesil süresi (s)	0.048889	0.045896	0.030915	0.048223	0.033067
	Başarım sayısı	0	2	0	0	0
	Ortalama Başarım Sayısı	2/25=0.08				
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	0.33775	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
	Başarım sayısı	2	1	4	1	1
	Ortalama Başarım Sayısı	9/25=0.36				

Çizelge 4. 5. ABC, PSO ve DE algoritmalarının ÖDS modellemeye *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)
ABC	En iyi	0.027788	0.020664	0.025316	0.094835	0.203715
	En kötü	1.97251	0.801725	547.1167	5.72806	1322.705
	Başarım Sayısı	0	1	0	0	0
	Ortalama Başarım Sayısı	1/10=0.1				
G-ABC-I	En iyi	0.032866	0.035922	0.017745	0.120933	0.201923
	En kötü	1.869034	1.317354	10.55646	21.5831	344.9939
	Başarım Sayısı	0	0	0	0	0
	Ortalama Başarım Sayısı	0/5=0				
G-ABC-II	En iyi	0.023195	0.028244	0.024321	0.08736	0.201685
	En kötü	2.494889	0.924081	5.162191	6.997366	154.9019
	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	0.324227
	En kötü	1.278393	1.094454	51.89824	0.732723	0.931145
	Başarım Sayısı	0	0	0	2	0
	Ortalama Başarım Sayısı	2/10=0.2				
G-PSO-I	En iyi	0.037084	0.022213	0.014848	0.119945	0.393983
	En kötü	1.066521	0.272192	18.25806	0.742309	5.139894
	Başarım Sayısı	0	1	1	0	0
	Ortalama Başarım Sayısı	2/10=0.2				
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	1	0	2
	Ortalama Başarım Sayısı	4/10=0.4				

4.7. Literatüre Dayalı Başarım Kıyaslaması

Yukarıda detaylı bir şekilde sonuçları verilen ÖDS'ler için literatürde ANFIS kullanılarak yapılmış bir modelleme belirlenemediğinden Jang (Jang,1993)'da verilen 3. örnek için ABC, DE ve PSO algoritmalarının başarımı irdelenmiştir. Aşağıda E.4.1 ve E.4.2'de bu örneğin matematiksel modeli verilmiştir.

$$y(k+1) = 0.3 y(k) + 0.6y(k-1) + f(u(k)) \quad \text{E.4.1}$$

$$f(u(k)) = 0.6 \sin(\pi u) + 0.3 \sin(3\pi u) + 0.1\sin(5\pi u) \quad \text{E.4.2}$$

Jang (Jang,1993)'ın bu sistemin ANFIS ile modellemesinde kullandığı aynı giriş, mimari, ÜF ve kural yapısı kullanılmıştır. Farklı olarak burada ANFIS parametre optimizasyonu sezgisel algoritmalarla yapılmıştır. Sistem modellemesi eğitim fazı için eğitim seti hazırlanırken E.4.3'de verilen harici giriş kullanılmıştır. Test seti hazırlanırken ise E.4.4'de verilen harici giriş kullanılmıştır. Üyelik fonksiyonu olarak da E.4.5'deki genelleştirilmiş çan fonksiyonu kullanılmıştır.

$$u(k) = \sin \frac{2\pi k}{250} \quad \text{E.4.3}$$

$$u(k) = 0.5 \sin \left(\frac{2\pi k}{250} \right) + 0.5\sin(2\pi k/25) \quad \text{E.4.4}$$

$$\mu_{A_i}(x) = \frac{1}{1 + \left[\left(\frac{x - c_i}{a_i} \right)^{2bi} \right]} \quad \text{E.4.5}$$

E.4.1'de verilen ayrık dinamik sistemin ANFIS ile modellemesi Jang (Jang, 1993)'da verildiği gibi sırasıyla 3,5 ve 7'şer adet üyelik fonksiyonu ve kural ile yapılırken parametrelerin (ÜF ve kural) optimizasyonu için ABC, PSO ve DE algoritmaları koşturulmuştur. Her bir modellemede üyelik fonksiyonların başlangıç ve son durumları, ANFIS çıkışları elde edilmiştir. Jang (Jang,1993)'da elde edilen sonuçlara ait sayısal başarım ölçütü verilmediğinden sadece grafiksel olarak kıyaslama yapılabilmektedir.

Optimizasyon için ABC, PSO ve DE algoritmalarında kullanılan parametre değerleri sırasıyla Çizelge 4.6 - Çizelge 4.8'de verilmiştir. Bu kıyaslama örneğinde ANFIS ile sistem tanıma ve modellemede kullanılan standart ABC için arı uygunluk/kalite değerleri hesaplanırken kullanılan yazılım kodları EK-2'de verilmiştir.

Çizelge 4. 6. ABC Parametreleri

<i>Parametre Adı</i>	<i>Parametre Değerleri</i>
Koloni büyüklüğü (N)	70
Max nesil sayısı	250
Limit	$(N/2)*D*0.5$
Algoritma koşma sayısı	1

Çizelge 4. 7. PSO Parametreleri

<i>Parametre Adı</i>	<i>Parametre Değerleri</i>
Sürü Büyüklüğü	70
Max nesil sayısı	250
c1, c2	2.05
Algoritma koşma sayısı	1

Çizelge 4. 8. DE Parametreleri

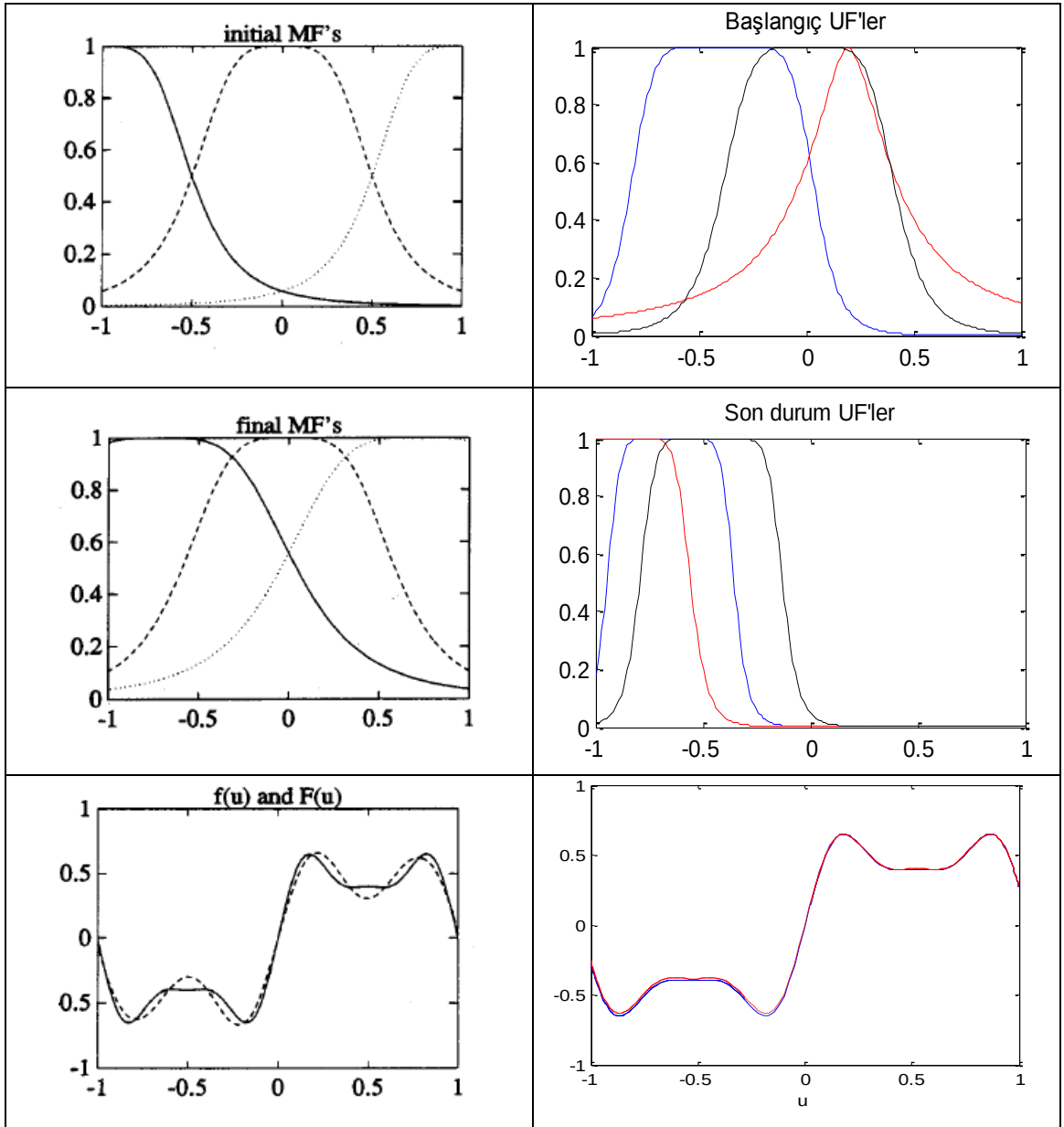
<i>Parametre Adı</i>	<i>Parametre Değeri</i>
Popülasyon büyüklüğü (NP)	70
Max nesil sayısı	250
Çaprazlama sabiti	0.4
Skala faktörü	0.3
Algoritma koşma sayısı	1

Şekil 4.36'da 3 üyelik fonksiyonu ile elde edilen başarımla (Jang,1993)'de verilen başarımla kıyaslamalı olarak verilmiştir. Şekilde sol taraftaki sütun (Jang,1993)'de verilen sonuçları, sağ taraftaki sütunda ise ABC algoritmasının koşturulması sonucu elde edilen sonuçları göstermektedir. Şekilden de görüleceği üzere, ABC'nin başarımı (Jang,1993)'de verilen başarıma kıyasla oldukça iyidir. Şekil 4.37'de ise 5 üyelik fonksiyonu ile ABC algoritmasının koşturulması sonucu elde edilen sonuçlar kıyaslamalı olarak verilmiştir. Şekil 4.38'de de 7 üyelik fonksiyonu ile elde edilen sonuçlar kıyaslamalı olarak verilmiştir. Şekil 4.36 ve Şekil 4.37 incelendiğinde ANFIS bulanık modelinin 5 ve 7 üyelik fonksiyonu kullanılarak ABC algoritması ile optimizasyonunda başarımın gayet iyi olduğu görülebilir.

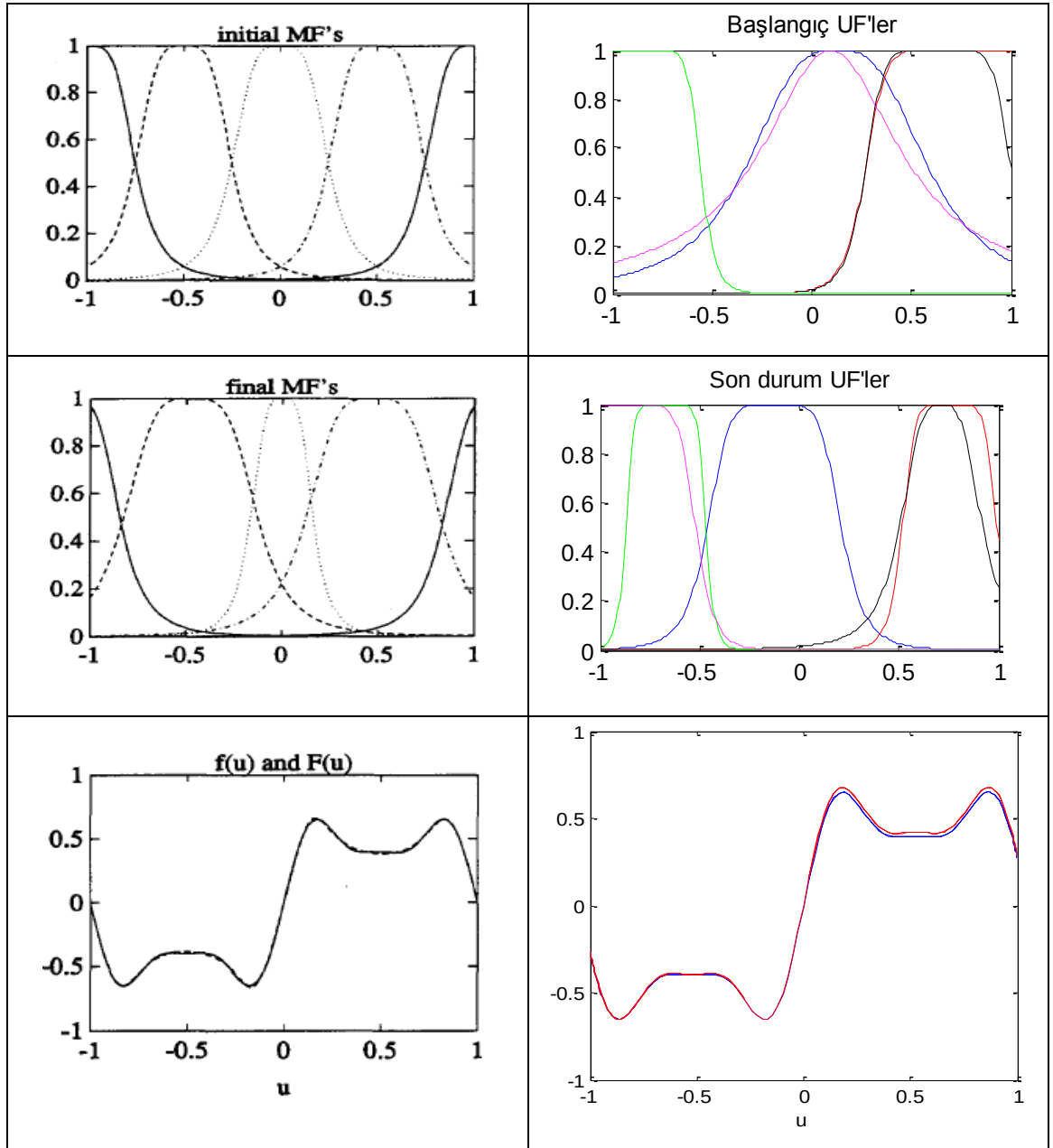
Bu tez çalışmasında kullanılan üç sezgisel algoritmanın başarımı hakkında da genel bir değerlendirme elde edebilmek amacıyla, 5 ÜFlı ve 5 kurallı ANFIS modeli standart ABC, standart PSO ve standart DE ile optimize edilmiş olup elde edilen eğitim ve test sonuçları Şekil 4.39'da verilmiştir. Şekil 4.40'da ise E.4.1'de verilen sistemin ANFIS modeli optimize edilirken başlangıç ve son durum üyelik fonksiyonlarının durumu verilmiştir. Benzer şekilde, Şekil 4.41 ve Şekil 4.42'de de yedi üyelik fonksiyonu kullanılarak elde edilen sonuçlar verilmiştir. ANFIS çıkışı ve sistem çıkışı ile ilgili tüm kıyaslamalı şekillerde, düz çizgi ile verilen istenen durumu, kesikli çizgi ile verilen ise elde edilen durumu göstermektedir.

3,5 ve 7 üyelik fonksiyonu kullanılarak elde edilen sonuçlar; Jang (Jang,1993)'da sayısal değerler verilmediği için grafiksel olarak incelenmiştir. ABC, PSO ve DE ile elde edilen grafiksel sonuçlara bakıldığında, algoritmaların hem eğitim fazında ($k \leq 250$) hem de test fazında ($k > 250$) başarılı oldukları görülmüştür. Burada dikkat edilmesi gerekli en önemli husus şudur: Jang (Jang,1993)'da ANFIS karmaşık türevsel hesaplamaların yapıldığı karma bir öğrenme algoritmasıyla optimize edilmişken; bu çalışmada ABC, PSO ve DE ile yalın olarak sadece algoritmaların koşturulması ile optimize edilmiştir.

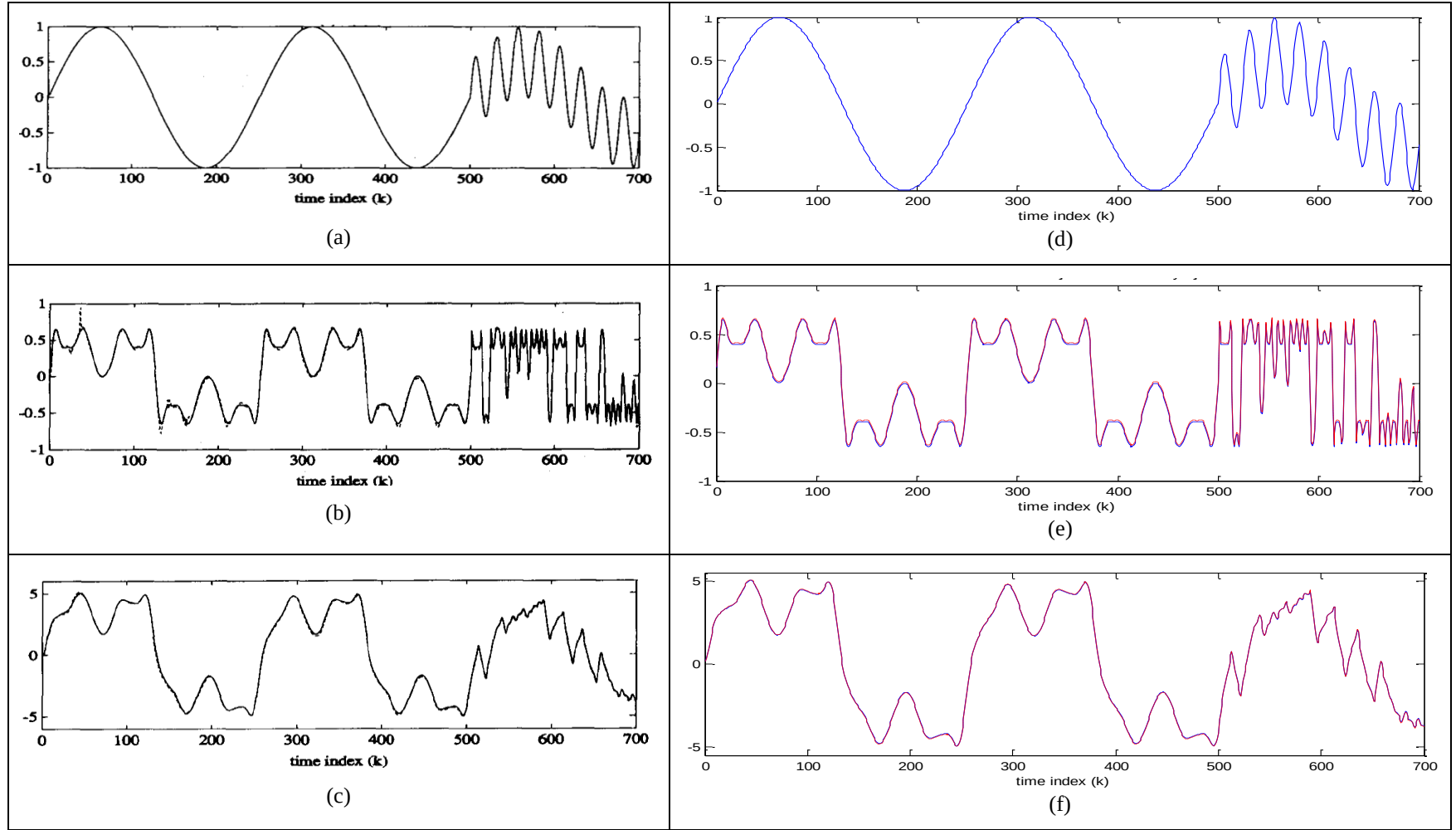
Sonuç olarak, Şekil 4.36-4.42'de verilen kıyaslamalı sonuçlar topluca değerlendirildiğinde sezgisel algoritma tabanlı ANFIS optimizasyonunun üstün başarımlar sağladığı görülmüştür. Bu üstünlük Şekil 4.36'nun son satırındaki şekillerde daha net olarak görülebilir.



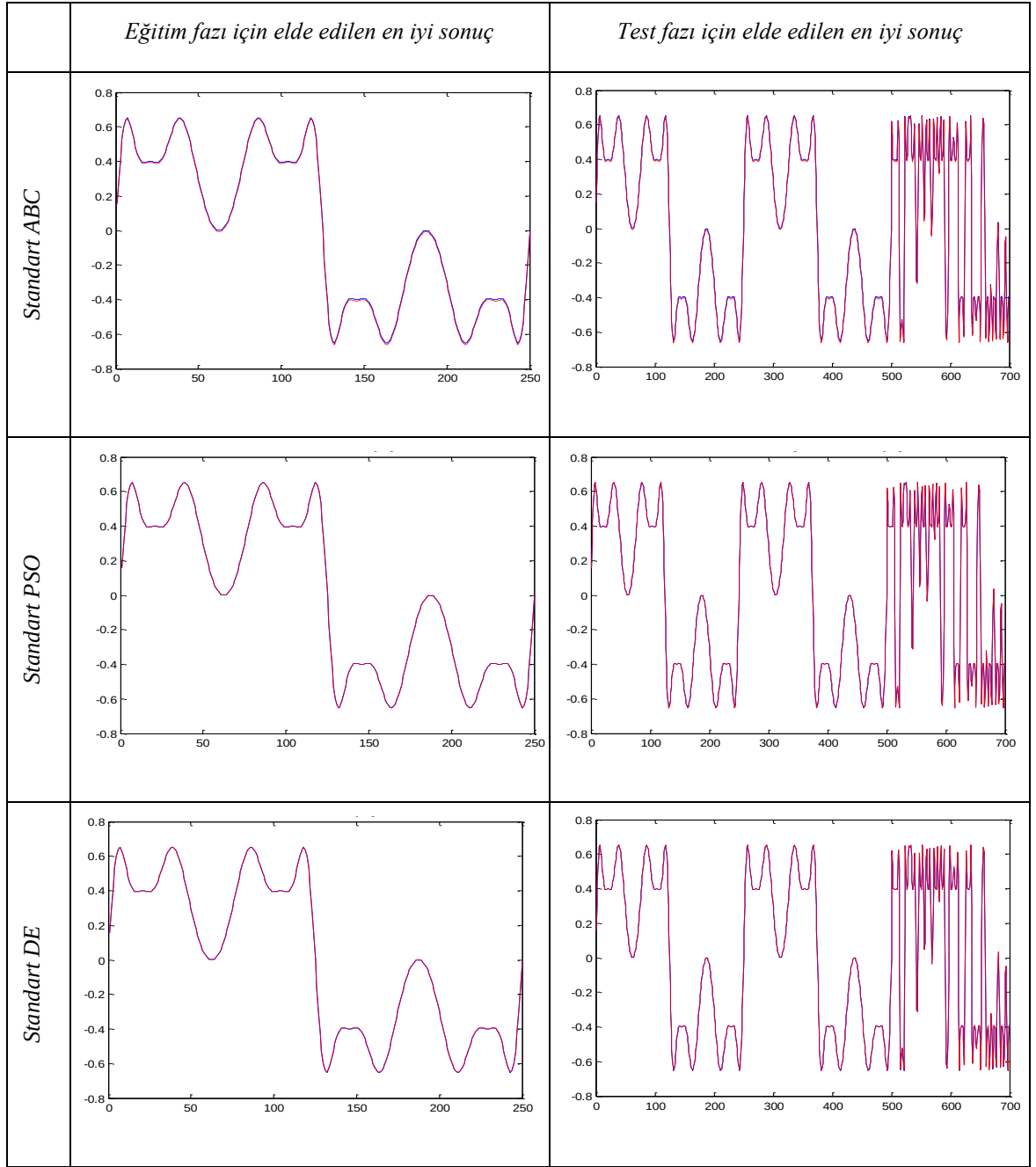
Şekil 4. 36. Üç üyelik fonksiyonu ile modelleme başarımı: sol kolon Jang (Jang,1993)'den alınan sonuçlar, sağ kolon ABC ile elde edilen sonuçlar



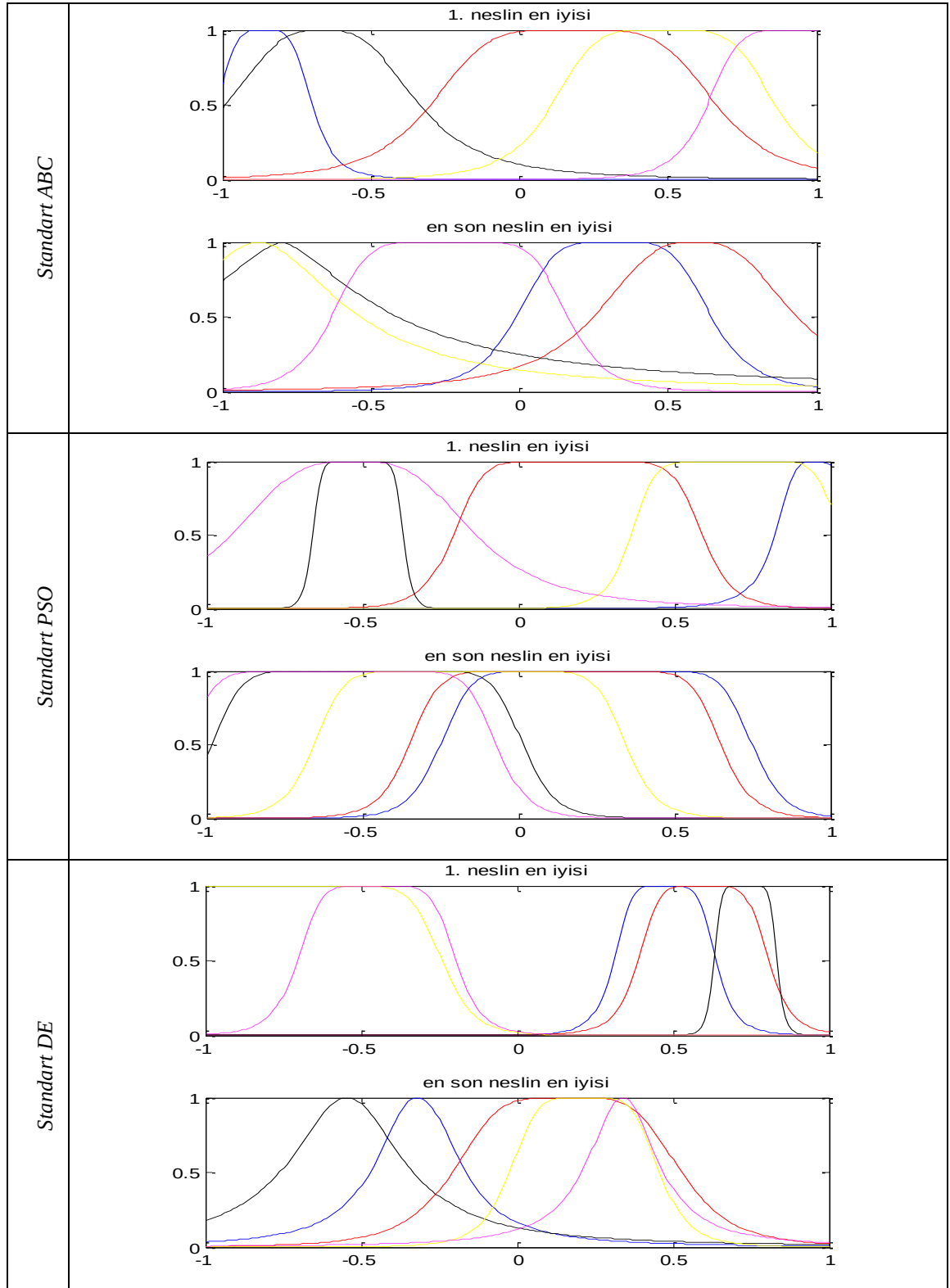
Şekil 4. 37. Beş üyelik fonksiyonu ile modelleme başarımı: sol kolon Jang (Jang, 1993)'dan alınan sonuçlar, sağ kolon ABC ile elde edilen sonuçlar



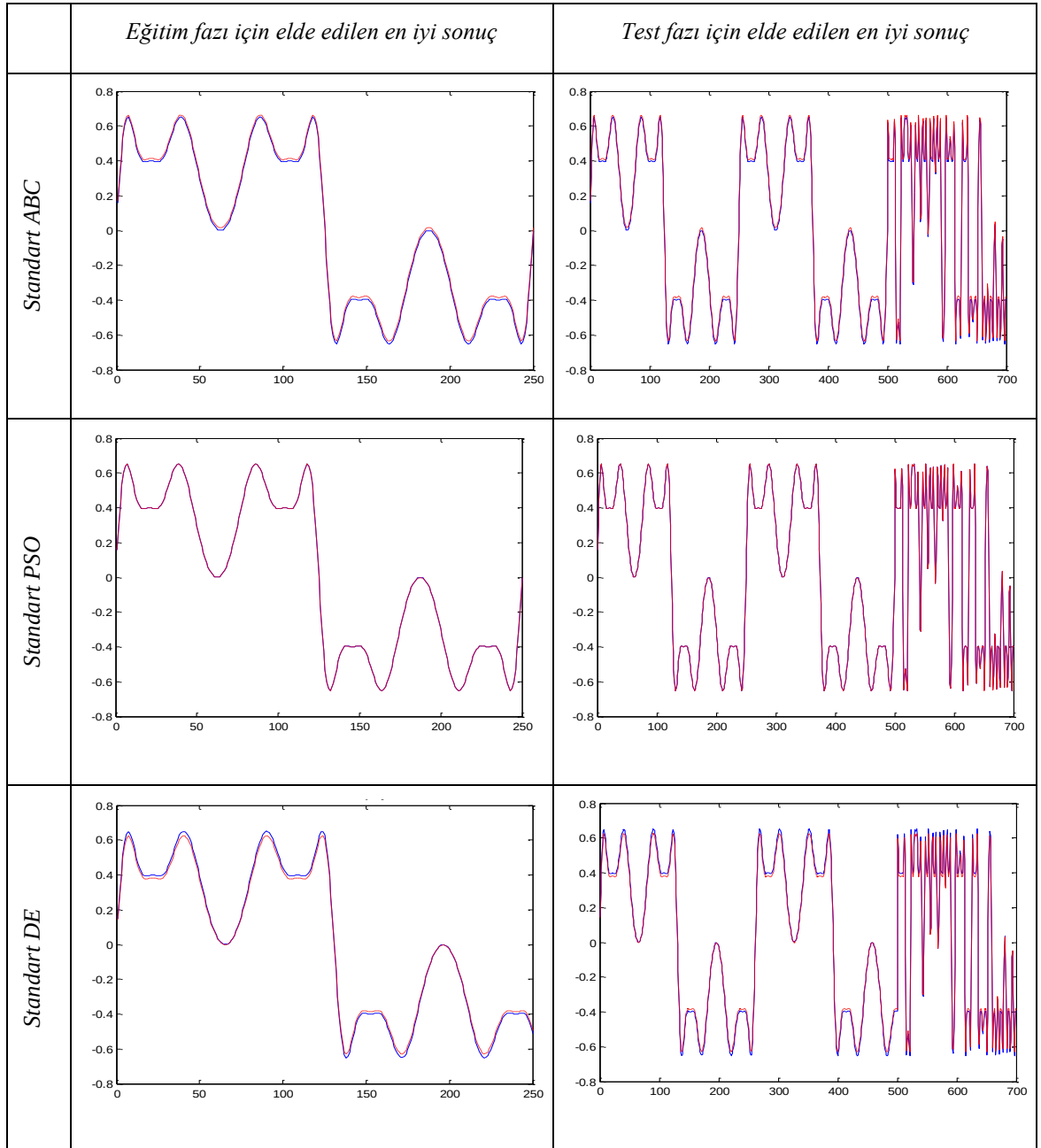
Şekil 4. 38. Yedi ÜF kullanan ANFIS ile elde edilen (sağ kolon: d, e, f) (Jang,1993) ile (sol kolon:a, b, c) kıyaslamalı sonuçlar: (a) ve(d) $u(k)$ harici giriş, (b) ve (e) ANFIS çıkışı $f(u(k))$, (c) ve (f) sistem çıkışı $y(k+1)$



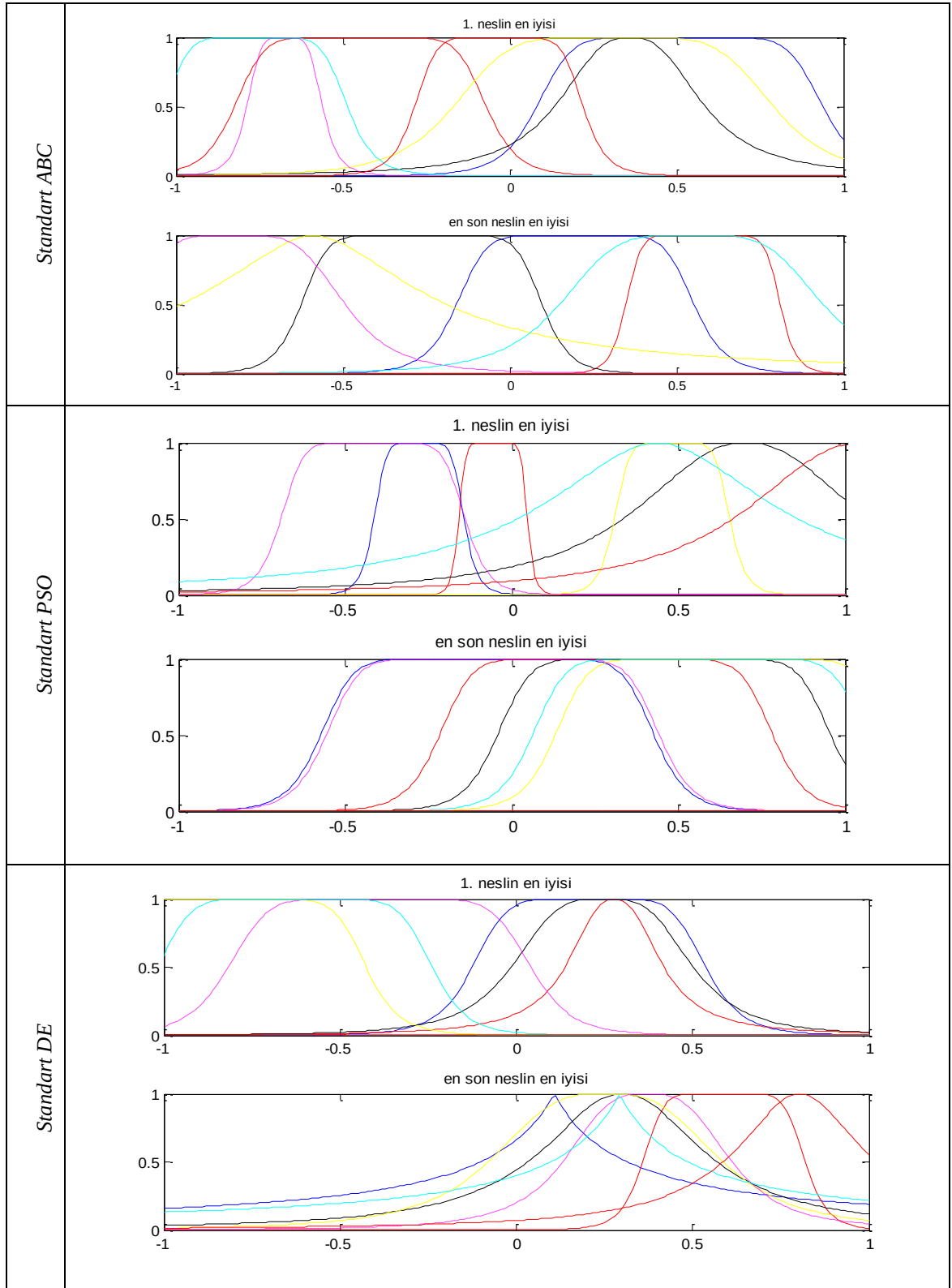
Şekil 4. 39. Beş üyelik fonksiyonu kullanılarak ABC, PSO ve DE ile elde edilen eğitim (sol kolon) ve test (sağ kolon) sonuçları



Şekil 4. 40. ABC, PSO ve DE ile beş üyelik fonksiyonlu ANFIS optimizasyonunda üyelik fonksiyonlarının başlangıç ve son durumları



Şekil 4. 41. Yedi üyelik fonksiyonu kullanılarak ABC, PSO ve DE ile elde edilen eğitim (sol kolon) ve test (sağ kolon) sonuçları



Şekil 4. 42. ABC, PSO ve DE ile yedi üyelik fonksiyonlu ANFIS optimizasyonunda üyelik fonksiyonlarının başlangıç ve son durumları

5. SONUÇLAR

Bu tez çalışmasında, bulanık işleyişi gerçekleyen ANFIS (Adaptive Neuro-Fuzzy Inference System) bulanık mantık çıkarım modeli kullanılarak; ABC'nin doğrusal olmayan dinamik sistemlerin bulanık modellenmesinde optimizasyon başarımı incelenmiş ve literatürde sıklıkla kullanılan PSO ve DE'nin başarımlarıyla kıyaslanmıştır. Bölüm 1'de verilen literatür özetinden de anlaşılacağı üzere, ABC'nin bu tez çalışması kapsamında yapılan çalışmaya benzer bir kullanımı henüz mevcut değildir. Bu bakımdan bu çalışmanın ilgili alanda katkı sunduğu değerlendirilmektedir.

Bölüm 3'de verilen ANFIS ağ yapısına göre dinamik sistemler için gerekli parametre optimizasyon araçları oluşturulmuş, sistemlerin hem eğitilmesinde hem de test edilmesinde üç algoritma için popülasyon büyüklüğü, parametre, maksimum nesil sayısı ve koşma sayısı eşit alınarak uygulamalar yapılmıştır. Her bir algoritmanın başarımı ayrı ayrı elde edilmiş ve detaylı bir şekilde incelenerek kıyaslama yapılmıştır.

ABC algoritmasının doğrusal olmayan dinamik sistemlerin bulanık modellenmesindeki başarımı popülasyon büyüklüğünün önemli olduğu PSO ve DE algoritmaları ile kıyaslandığında, bu tez kapsamında yapılan çalışmalardan elde edilen sonuçlara göre, genel olarak bulanık sistem modellemede DE'nin daha başarılı olduğu ancak ABC ve bu çalışmada geliştirilen türevlerinin başarımlarının DE'nin başarımına çok yakın olduğu sonucuna varılmıştır. İncelemede ele alınan PSO algoritmasının başarımının ise ABC ve DE'nin başarımlarına göre daha kötü olduğu görülmüştür.

Algoritmaların bir nesil için koşturma süresine bakıldığında en iyi bir nesil için ortalama koşma zamanı ABC algoritmasına aittir. Öyle ki, optimizasyon başarımında DE ABC'den daha iyi sonuç vermesine rağmen, bir nesil için ABC'nin ortalama koşma süresinin DE'ninkine göre iki kat daha kısa olduğu gözlenmiştir. Bu sonuca dayalı olarak, gerçek zamanlı (özellikle gömülü sistem tabanlı) uygulamalarda ABC işletiminin, DE ve PSO işletimine göre daha elverişli olduğu anlaşılmaktadır. ABC algoritmasının diğer algoritmalara göre daha basit bir yapı ve daha az sayıda kontrol parametresinin olması da algoritmaya ayrı bir üstünlük kazandırdığı değerlendirilmiştir.

Bu tez kapsamında yapılan çalışmalar sonucunda çok yeni ve güncel bir algoritma olan ABC'nin bulanık sistem parametrelerinin optimizasyonunda etkin bir şekilde kullanılabileceği gösterilmiştir. Bu çalışmada ABC algoritmasının doğrusal olmayan dinamik sistemlerin bulanık mantık tabanlı optimizasyonu başarımında bazı ÖDS'ler için istenilene yakın sonuçlarında elde edilemediği görülmüştür. Bu sonucun sebebi olarak; ANFIS yapısında önceden belirlenen ve sabit tutulan üyelik fonksiyon sayısı, kural sayısı, giriş sayısı, parametreler için sabit alt ve üst sınır değerlerinin kullanılması gösterilebilir. Bahsedilen bu kısıtlamaların optimizasyon sürecinde ABC algoritmasının performansını olumsuz etkilediği düşünülmektedir. Bu nedenle ileride yapılacak çalışmalarda ANFIS için özellikle kural ve ÜF sayısının uyarlamalı olarak belirlendiği problemlerde ABC algoritmasının başarımının değerlendirilmesi gerektiği önerilmektedir. Ancak, belirtilen bu yapısal ANFIS öğrenme yönteminin daha karmaşık olacağı unutulmamalıdır. Bu bağlamda literatürde henüz bir şablon şeklinde yapısal öğrenme yöntemi geliştirilmiş değildir.

6. KAYNAKLAR

- Akay B., “Nümerik Optimizasyon Problemlerinde Yapay Arı Kolonisi (Artificial Bee Colony) Algoritmasının Performans Analizi”, **Erciyes Üniversitesi Fen Bilimleri Enstitüsü**, Doktora Tezi, 301s, Kayseri, 2009.
- Alataş Bilal, “Chaotic bee colony algorithms for global numerical optimization”, **Expert Systems with Applications**, 37 (2010) 5682–5687.
- Alok Singh ,“An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem”, **Applied Soft Computing**, 9 (2009) 625–631.
- Alok Singh, Shyam Sundar, “An artificial bee colony algorithm for the minimum routing cost spanning tree problem”, **Soft Computing**, Volume 15, Issue 12, pp 2489-2499, December 2011.
- Babuska R., "Fuzzy System, Modeling and Identification", <http://www.dsc.tudelft.nl/~babuska/transp/fuzzmod.pdf> (son erişim 19 Şubat 2012).
- Banharnsakun Anan, Achalakul Tiranee, Sirinaovakul Booncharoen, “The best-so-far selection in Artificial Bee Colony algorithm “, **Applied Soft Computing** 11, 2888–2901, 2011.
- Chang, Ping-Teng, Lee Jung-Hua, “A fuzzy DEA and knapsack formulation integrated model for project selection”, **Computers & Operations research**, Volume 39 Issue 1, Pages 112-125, January 2012.
- Chaiyatham Theerawut, Ngamroo Issarachai, “A bee colony optimization based-fuzzy logic-pid control design of electrolyzer for microgrid stabilization”, **International Journal of Innovative Computing**, Information and Control Volume 8, Number 9, September 2012.
- Clerc M. and Kennedy J., “The particle swarm-explosion, stability, and convergence in a multidimensional complex space”, **IEEE Trans. Evol. Comput.**, vol. 6, no. 1, pp. 58–73, 2002.
- Coelho Leandro dos Santos, Alotto Piergiorgio, “Gaussian Artificial Bee Colony Algorithm Approach Applied to Loney's Solenoid Benchmark Problem” , **IEEE Transactions on Magnetics**, vol. 47, no. 5, May 2011.
- Cuevas Erik, Sencion-Echauri Felipe, Zaldivar Daniel, “Multi-circle detection on images using artificial bee colony (ABC) optimization”, **Applied Soft Computing** 12, 342–352, 2012.
- Çavuşlu Mehmet Ali, Karakuzu Cihan, Karakaya Fuat, "Neural identification of dynamic systems on FPGA with improved PSO learning", **Applied Soft Computing** 12, 2707–2718, 2012.
- Dornhaus A., Klügl F., Puppe F., Tautz J., “Task Selection in Honeybees-Experiments Using Multi-Agent Simulation.”, in **Proc of GWAL'98**.

- Gao Wei-feng , Liu San-yang, “A modified artificial bee colony algorithm”, **Computers & Operations Research** 39,687–697, 2012.
- Grüter C., Farina M.,” The Honeybee Waggle Dance: Can We Follow The Steps?” **Trends in Ecology & Evolution** , 24(5), 242-247, 2009.
- Hiroyasu, T., Miki, M., Ono, Y., & Minami, Y. ,”Ant colony for continuous functions”. **The Science and Engineering**, Doshisha University, 2000.
- Holland J. H., “Adaption in Natural and Artificial Systems”, University of Michigan Pres, Ann Arbor, MI, 1975.
- Ho S. L., Yang Shiyong ,“An artificial bee colony algorithm for inverse problems”, **International Journal of Applied Electromagnetics and Mechanics** 31, 181–192 181, 2009.
- Horng Ming-Huwi, “Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation”, **Expert Systems with Applications** 38, 13785–13791, 2011.
- Huang Yueh-Min, Lin Jin-Chen, “A new bee colony optimization algorithm with idle-time-based filtering scheme for open shop-scheduling problems” , **Expert Systems with Applications** 38,5438–5447, 2011.
- Jang S. - R. J.,“ANFIS: adaptive-network-based fuzzy inference system”, **IEEE Trans. on Systems, Man, and Cybernetics**, vol. 23, No. 3, pp. 665-684, 1993.
- Kang Fei, Li Junjie, Xu Qing , “Structural inverse analysis by hybrid simplex artificial bee colony algorithms”, **Computers and Structures** 87,861–870, 2009.
- Karaboğa D., Yapay Zeka Optimizasyon Algoritmaları, **Atlas Yayın Dağıtım**, Şubat 2011.
- Karaboga, D., “An Idea Based On Honey Bee Swarm For Numerical Optimization. Technical Report-TR06”, **Erciyes University Engineering Faculty Computer Engineering Department**, Kayseri, 2005.
- Karaboga, D., Akay, B.,” On The Performance Of Artificial Bee Colony (ABC) Algorithm.”, **Applied Soft Computing**, 8(1), 687-697, 2008.
- Karaboga Nurhan, “Digital IIR Filter Design Using Differential Evolution Algorithm”, **EURASIP Applied Signal Processing**, No.8, p:1269-1276, 2005.
- Karaboga Dervis and Okdem Selcuk, “A Simple and Global Optimization Algorithm for Engineering Problems, Differential Evolution Algorithm”, **Turk J Elec Engin**, Vol.12, No.1, 2004.
- Karaboga Dervis, Akay Bahriye, “A modified Artificial Bee Colony (ABC) algorithm for constrained optimization problems”, **Applied Soft Computing** 11, 3021–3031, 2011.

- Karaboğa D, Başturk B., "On the performance of artificial bee colony (ABC) algorithm", **Applied Soft Computing** 8,687–697, 2008.
- Karaboga, Dervis; Ozturk, Celal , " Neural networks training by artificial bee colony algorithm on pattern classification", **Neural Network World**, Mayıs 2009.
- Karaboga Nurhan , "A new design method based on artificial bee colony algorithm for digital IIR filters", **Journal of the Franklin Institute** 346,328–348, 2009.
- Karaboga Nurhan, Cetinkaya Mehmet Bahadır, "A novel and efficient algorithm for adaptive filtering: Artificial bee colony algorithm", **Turk J Elec Eng & Comp Sci**, Vol.19, No.1, TUBITAK, 2011.
- Karakuzu C., Kesler M., Yüzgeç U., "Güncel Sezgisel Arama Algoritmalarının Denetleyici Parametrelerinin Optimizasyonunda Başarım Kıyaslaması", **6 th IATS'2011**.
- Karakuzu C., Yıldırım Ö., Yüzgeç U., Kesler M., " ABC Algoritmasının PID Denetleç Parametrelerinin Optimizasyonunda Kıyaslamalı Başarımı", **ASYU 2012 Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu Bildiriler Kitabı**, Sf. 101-105, Trabzon, Türkiye, 3-4 Temmuz 2012.
- Kennedy, J. and Eberhart, R. C. "Particle swarm optimization" Proc. IEEE int'l conf. on neural networks Vol. IV, pp. 1942-1948. IEEE service center, Piscataway, NJ, 1995.
- Kosko, B., "Neural networks and fuzzy systems, A Dynamical Systems Approach", Englewood Cliffs.,NJ, Prentice Hall, 1991.
- Koorush Ziarati, Reza Akbari, Vahid Zeighami, "On the performance of bee algorithms for resource-constrained project scheduling problem", **Applied Soft Computing** 11, 3720–3733, 2011.
- L.A. Zadeh, "Fuzzy sets", **Information and Control** 8, 338-353, 1965.
- Li Guoqiang, Niu Peifeng, Xiao Xingjun, "Development and investigation of efficient artificial bee colony algorithm for numerical function optimization", **Applied Soft Computing** 12, 320–332, 2012.
- Narendra K. S., Parthasarathy K., "Identification and control of dynamical systems using neural networks", **IEEE Transactions on Neural Networks**, vol. 1, no 1, pp. 4-27, 1990.
- Ok Yeşim, "Adaptif Sinirsel Bulanık Çıkarım Sistemi (Anfis) İle Türkiye’de Orta Dönemli Elektrik Enerjisi Talep Tahmini", Gazi Üniversitesi, **Fen Bilimleri Ens.**, Yüksek Lisans Tezi, Ankara, Şubat 2010.
- Oussar Y., Rivals I., Dreyfus L., "Training wavelet networks for nonlinear dynamic input output modeling", **Neurocomputing**, cilt 20, syf 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), 2003.

- Passino, K. M., Yurkovich, S., "Fuzzy Control", **Addison Wesley**, California, 1997.
- Pei-Wei TSai , Jeng-Shyang Pan, Bin-Yih Liao ve Shu-Chuan Chu , "Enhanced artificial bee colony optimization", **International Journal of Innovative Computing**, Volume 5, Number 12, December 2009.
- Price K., "An Introduction to Differential Evolution", **D. Corne, M. Dorigo, and F. Glover, Eds. London, U.K., McGraw-Hill**, pp. 79–108, ISBN: 007-709506-5, New Ideas in Optimization, 1999.
- Price K. V., "Differential Evolution a Fast and Simple Numerical Optimizer", In: Smith, M., Lee, M., Keller, J., Yen., J. (eds.): Biennial Conference of the North American Fuzzy Information Processing Society, NAFIPS. IEEE Press, New York, p:524-527, 1996.
- S. Hemamalini and Sishaj P. Simon , "Artificial Bee Colony Algorithm for Economic Load Dispatch Problem with Non-smooth Cost Functions", **Electric Power Components and Systems**, 38:786–803, 2010.
- Samrat L.Sabat, , SibaK.Udgata, Ajith Abraham , "Artificial bee colony algorithm for small signal model parameter extraction of MESFET", **Engineering Applications of Artificial Intelligence** , 2010
- Sastry P. S., Santharam G., Unnikrishnan K. P., "Memory neuron networks for identification and control of dynamical systems", **IEEE Transaction on Neural Networks**, cilt 5, no 2, 1994.
- Sato T., Hagiwara M., "Bee System: Finding solution by a concentrated search system, Man and Cybernetics", **IEEE International Conference on Computational Cybernetics and Simulation**, volume 4. 3954-3959. Orlando, FL.USA,1997.
- Sonmez Mustafa, "Artificial Bee Colony algorithm for optimization of truss structures ,Applied Soft Computing 11, 2406–2418, 2011.
- Su Zhi-gang, Wang Pei-hong, Shen Jiong, "Convenient T-S fuzzy model with enhanced performance using a novel swarm intelligent fuzzy clustering technique", **Journal of Process Control**, Volume 22, Issue 1, January 2012, Pages 108-124
- Şen, Z., "Mühendislikte bulanık (fuzzy) mantık ile modelleme prensipleri" **Su Vakfı Yayınları, Bilge Yayıncılık**, İstanbul (2004).
- Şenol C., Yildirim T., "Bulanık-Sinir Ağı Yapısı için Yeni Bir Karma Yaklaşım", **ELECO'2008 Elektrik-Elektronik ve Bilgisayar Mühendisliği Sempozyumu**, Bursa, 26-30 Kasım 2008.
- Tereshko, V., Loengarov, A., "Collective decision making in honey-bee foraging dynamics.", **Computing and Informaton Systems**, 9(3), 1-7, 2005.
- Tokmak M., "Yapay Arı Kolonisi Algoritması ile Ders Çizelgeleme Probleminin Çözümü", Yüksek Lisans Tezi, **Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü**, Isparta, 2011.

- Tsoukalas, L., Uhrig, R., “Neuro fuzzy approach for anticipatory control of complex systems”, **IEEE International Conference in Fuzzy Systems**, (1):587-593,1996.
- Yeh Wei-Chang, Hsieh Tsung-Jung ,“Artificial bee colony algorithm-neural networks for S-system models of biochemical networks approximation”, **Neural Comput & Applic** 21:365–375, 2012 .
- Xu Chunfan, Duan Haibin, “Artificial bee colony (ABC) optimized edge potential function (EPF) approach to target recognition for low-altitude aircraft”, **Pattern Recognition Letters** 31:1759–1772, 2010.
- Wang Yajun, Chen Wen, Tellambura Chintha, “A PAPR Reduction Method Based on Artificial Bee Colony Algorithm for OFDM Signals”, **IEEE Transactions on Wireless Communications**, vol. 9, no. 10, October 2010.
- Zhang Changsheng, Ouyang Dantong, Ning Jiaxu , “An artificial bee colony approach for clustering”, **Expert Systems with Applications** 37 :4761–4767, 2010.
- Zhang Y., Wu L., Wang S. , “Magnetic resonance brain image classifica- tion by an improved artificial bee colony algorithm”, **Progress In Electromagnetics Research**, Vol. 116, 65,79, 2011.

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

```

ariS=50;
cicekS=ariS/2;
D=2;
snr=[-3 -3 ;5 5];
limit=cicekS*D;
iteS=100;
eniyicozum=zeros(100,1);

for i=1:cicekS
    for j=1:D
        x(i,j)=snr(1,j)+rand*(snr(2,j)-snr(1,j));
    end

    fx(i)=fxhesapla(x(i,:));
    fitness(i)=1/(1+fx(i));
    hataS(i)=0;
end

for ite=1:iteS
    for i=1:cicekS
        satir=x(i,:);
        sutun=fix(rand*D)+1;
        k=fix(rand*cicekS)+1;
        teta=(rand-0.5)*2;
        j=sutun;
        v=x(i,j)+teta*(x(i,j)-x(k,j));

        if (v<snr(1,sutun))
            v=snr(1,sutun);
        end

        if (v>snr(2,sutun))
            v=snr(2,sutun);
        end
        satir(sutun)=v;
        gkalite=fxhesapla(satir);
        kalite=fx(i);
        if (gkalite<kalite)
            x(i,j)=v;
            fx(i)=gkalite;
            hataS(i)=0;
            fitness(i)=1/(1+gkalite);
        else
            hataS(i)=hataS(i)+1;
        end
    end

    end

    if (max(hataS)>=limit)
        i=find(hataS==max(hataS));
        for j=1:D
            x(i,j)=snr(1,j)+rand*(snr(2,j)-snr(1,j));
            hataS(i)=0;
        end
    end

    sonuc=fx';
    min(fx);
    eniyicozum(ite)=min(fx)
    [enkucukdeger enkucukindis]=min(fx);
    eniyi=x(enkucukindis,:);
    a(ite)=min(fx);
    b(ite)=ite;

    for i=1:cicekS
        t=(ite-1)*cicekS+i;
        X(t)=x(i,1);
        Y(t)=x(i,2);
        Z(t)=fx(i);
    end

end

toplamlFitness=sum(fitness);
for i=1:cicekS
    p(i)=fitness(i)/toplamlFitness;
end

```

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

```

function E=fxhesapla(ari,veri)

x3=veri(:,3);
G(1,:)=gbellmf(x3,[ari(1) ari(6) ari(11)]);
G(2,:)=gbellmf(x3,[ari(2) ari(7) ari(12)]);
G(3,:)=gbellmf(x3,[ari(3) ari(8) ari(13)]);
G(4,:)=gbellmf(x3,[ari(4) ari(9) ari(14)]);
G(5,:)=gbellmf(x3,[ari(5) ari(10) ari(15)]);

E=0;
for ite=1:length(x3)

for i=1:5
    Tnorm(i)=G(i,ite);
end

T=sum(Tnorm);

    wT=0;

    for o=1:5
        w(o)=Tnorm(o)/T;
    end

    wfT=0;

    for o=1:5
        wf(o)=(ari((o+15))*(x3(ite))+ari((o+20)));
        wfT=wfT+wf(o);
    end

    cikis=wfT;
    e=cikis-veri(ite,3); %
    E=e*e+E;

end

E=sqrt((1/length(x3))*E);
end

```

ÖZGEÇMİŞ

Kişisel Bilgiler

Adı Soyadı: Özlem YILDIRIM

Doğum Yeri ve Tarihi: Antalya – 25.08.1984

Eğitim Durumu

Lisans Öğrenimi: Süleyman Demirel Üniversitesi, Teknik Eğitim Fakültesi, Bilgisayar Sistemleri Öğretmenliği

Bilimsel Faaliyetleri: Karakuzu C., Yıldırım Ö., Yüzgeç U., Kesler M., " ABC Algoritmasının PID Denetleç Parametrelerinin Optimizasyonunda Kıyaslamalı Başarımı", ASYU 2012 Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu Bildiriler Kitabı, Sf. 101-105, Trabzon, Türkiye, 3-4 Temmuz 2012.

İş Deneyimi

Çalıştığı Kurum: 700. Yıl ATL, METEM – Bilgisayar Öğretmeni