



BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

KAOTİK TABANLI FARKSAL GELİŞİM ALGORİTMASI
(KFGA)

MEHMET ESER
Yüksek Lisans Tezi

Tez Danışmanı
Yrd. Doç. Dr. Uğur YÜZGEÇ

BİLECİK, 2014
Ref No:10045509



BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

KAOTİK TABANLI FARKSAL GELİŞİM ALGORİTMASI
(KFGA)

MEHMET ESER
Yüksek Lisans Tezi

Tez Danışmanı
Yrd. Doç. Dr. Uğur YÜZGEÇ

BİLECİK, 2014



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

**Chaotic based Differential Evolutionary Algorithm
(CDE)**

Mehmet ESER
Master's Thesis

Thesis Advisor
Asst. Prof. Dr. Uğur YÜZGEÇ

BİLECİK, 2014



**BİLECİK ŞEYH EDEBALI
ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS
JÜRİ ONAY FORMU**

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 05.06.2014 tarih ve ...25.-6.... sayılı kararıyla oluşturulan jüri tarafından 26.06.2014 tarihinde tez savunma sınavı yapılan Mehmet ESER'in "KAOTİK TABANLI FARKSAL GELİŞİM ALGORİTMASI (KFGA)" başlıklı tez çalışması Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak oy birliği/oy çokluğu ile kabul edilmiştir.

JÜRİ

ÜYE (TEZ DANIŞMANI): Yrd. Doç. Dr. Uğur YÜZGEÇ

ÜYE: Yrd. Doç. Dr. Sıtkı ÖZTÜRK

ÜYE: Yrd. Doç. Dr. Suhap ŞAHİN

ONAY

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

İMZA MÜHÜR

ÖZET

Bu çalışmada optimizasyon problemlerinin çözümünde en çok kullanılan evrimsel algoritmalarından farksal gelişim algoritmasının, (FGA) temelini oluşturan rastgele sayı üretim süreci yerine, kaotik sistem tabanlı sayı üretici geliştirilmiştir. Kaotik sistemlerden Lorenz ve Rössler çekici sistemleri kullanılmıştır. Önerilen kaotik tabanlı FGA yapısı literatürden alınan on optimizasyon problemi için koşturulmuştur. Ayrıca FGA stratejileri, popülasyon büyüklükleri, skala faktörü ve çaprazlama sabitinin KFGA yapısı üzerindeki etkileri incelenmiştir. Klasik FGA ile önerilen KFGA (Kaotik Tabanlı Farksal Gelişim Algoritması)'nın karşılaştırılması yapılmış, önerilen kaotik yapının avantaj ve dezavantajları vurgulanmıştır. Bu tezde önerilen KFGA yapılarının gerçek zamanlı optimizasyon problemlerinde uygulanması planlanmaktadır. Ayrıca diğer kaotik sistemler için de FGA yapısının geliştirilmesi düşünülmektedir.

Anahtar sözcükler: Kaotik Tabanlı Farksal Gelişim Algoritması, Farksal Gelişim Algoritması, Optimizasyon, Yapay Zeka.

ABSTRACT

In this study, the chaotic system based number generator was developed instead of the random number generators which are the basis of Differential Evolution Algorithm (DE) that is the most used evolutionary algorithms in solving optimization problems. Lorenz and Rössler chaotic attractive systems were used to do the number generators. The proposed chaotic based DE structure was run for ten optimization problem taken from literature. In addition, the effects of DE strategies, the population size, the scale factor and the crossover constant on CDE structure were examined. The comparison of classic DE and the proposed CDE (Chaotic based Differential Evolutionary Algorithm) structure was presented; the advantages and disadvantages of the proposed chaotic structure were emphasized. The application of proposed CDE structures in this thesis for the real-time optimization problems will be in the future works. Also in the development of DE algorithm will be used other chaotic system.

Keywords: Chaotic-Based Differential Evolution Algorithm, Differential Evolution Algorithm, Optimization , Artificial Intelligence.

TEŞEKKÜR

Vergileriyle okumamı sağlayan yüce Milletime, bu imkanları bize sunan Sayın Rektörümüz Prof. Dr. Azmi ÖZCAN'a, sabır, bilgi ve tecrübesiyle kendisinden çok şey öğrendiğim, tezimin hazırlanmasında büyük emekleri olan değerli hocam Yrd. Doç. Dr. Uğur YÜZGEÇ'e, manevi desteklerinden dolayı Yrd. Doç. Dr. Nazım İMAL'a, yardımlarından dolayı görev yaptığım MYO Müdürü Doç. Dr. Çağlayan AÇIKGÖZ'e ve Müdür Yardımcıları Öğr.Gör. İbrahim KILIÇ ve Öğr.Gör. Metehan ÖNAL'a, bana fikirler veren, destek olan, yardım eden isimlerini sayamadığım görev arkadaşlarıma, desteğinden, sabrından ve fedakarlığından dolayı sevgili eşime, varlığıyla moral kaynağım olan biricik kızıma, büyük fedakarlıklarla okumamı sağlayan, rahmetle andığım babama, her diplomamda büyük payı olan anneme teşekkür ederim.

İÇİNDEKİLER

TEZ ONAY SAYFASI	Sayfa No
ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER	iv
ÇİZELGELER DİZİNİ	vi
ŞEKİLLER DİZİNİ	ix
SİMGELER VE KISALTMALAR DİZİNİ	xiii
1. GİRİŞ	1
2. KAOS VE KAOTİK SİSTEMLER	7
2.1 Kaos ve Kaotiklik	7
2.1.1 Kaotik sistemlerde determinizm.....	10
2.1.2 Başlangıç koşullarına aşırı duyarlılık.....	11
2.1.3 Kaos analizi için gerek ve yeter koşullar.....	11
2.2 Kaos Ölçme Yöntemleri	12
2.2.1 Çeker oluşturma (Attractor Construction).....	13
2.2.2 Lyapunov üstelleri	13
2.2.3 Poincaré kesiti (Poincaré Section).....	14
2.2.4 Doğrusalsızlığın tespiti (Detection of Nonlinearity)	14
2.2.5 Fraktal boyut analizi	14
2.3 Örnek Kaotik Sistemler	15
2.3.1 Lorenz sistemi	15
2.3.2 Chua sistemi	19
2.3.3 Rössler sistemi.....	21
2.3.4 Mackey-Glass sistemi.....	22
3. KAOTİK TABANLI FARKSAL GELİŞİM ALGORİTMASI (KFGA)	24
3.1 Farksal Gelişim Algoritması (FGA)	24
3.1.1 Başlangıç toplumunun oluşturulması	27
3.1.2 Mutasyon operatörü.....	30
3.1.3 Çaprazlama operatörü.....	32

3.1.4	Seçim (Seleksiyon) işlemi	32
3.2	Kaotik Tabanlı Farksal Gelişim Algoritması (KFGA).....	33
4.	KFGA TABANLI OPTİMİZASYON UYGULAMASI	36
4.1	Kullanılan Test Fonksiyonları.....	36
4.2	Lorenz Tabanlı KFGA Optimizasyon Sonuçları.....	40
4.3	Rössler Tabanlı KFGA Optimizasyon Sonuçları	43
4.4	Farklı KFGA Stratejilerinin Karşılaştırılması (Kaotik Fonksiyon-Lorenz Çekicisi)	45
4.5	Farklı KFGA Stratejilerinin Karşılaştırılması (Kaotik Fonksiyon-Rössler Çekicisi)	48
4.6	Farklı Popülasyon Değerleri için Lorenz Tabanlı KFGA'nın Performansı ..	51
4.7	Farklı Popülasyon Değerleri için Rössler Tabanlı KFGA'nın Performansı..	54
4.8	KFGA Parametrelerinin İncelenmesi	57
4.8.1	Lorenz tabanlı KFGA yapısında skala faktörünün (F) etkisi.....	57
4.8.2	Lorenz tabanlı KFGA yapısında çaprazlama sabiti (CR) etkisi	60
4.8.3	Rössler tabanlı KFGA yapısında skala faktörünün (F) etkisi.....	63
4.8.4	Rössler tabanlı KFGA yapısında çaprazlama sabiti (CR) etkisi.....	66
4.9	KFGA Yapısında Çeşitlilik (Diversity) İncelenmesi.....	69
4.10	Klasik FGA ile KFGA Performanslarının Karşılaştırılması	72
5.	SONUÇ.....	75
	KAYNAKLAR.....	79
	ÖZGEÇMİŞ.....	83

ÇİZELGELER DİZİNİ

	Sayfa No
Çizelge 4.1. Kullanılan optimizasyon test fonksiyonları	36
Çizelge 4.2. Lorenz Tabanlı KFGA Optimizasyon Sonuçları (50 defa koşturma, NP=20, F=1.2 ve CR=0.7).....	41
Çizelge 4.3. Rössler Tabanlı KFGA Optimizasyon Sonuçları (50 defa koşma, NP=20, F=1.2 ve CR=0.7).....	43
Çizelge 4.4. Farklı FGA Stratejileri	45
Çizelge 4.5. Lorenz Tabanlı KFGA Optimizasyon Sonuçları (Ortalama Sonuç ve MSE) (20 defa koşma, NP=20, F=1.2 ve CR=0.7)	45
Çizelge 4.5. Lorenz Tabanlı KFGA Optimizasyon Sonuçları (Ortalama Sonuç ve MSE) (20 defa koşma, NP=20, F=1.2 ve CR=0.7) (devamı).....	46
Çizelge 4.6. Lorenz Tabanlı KFGA Optimizasyon Sonuçları (Ortalama İterasyon ve Ortalama Süre) (20 defa koşma, NP=20, F=1.2 ve CR=0.7).....	46
Çizelge 4.6. Lorenz Tabanlı KFGA Optimizasyon Sonuçları (Ortalama İterasyon ve Ortalama Süre) (20 defa koşma, NP=20, F=1.2 ve CR=0.7) (devamı).....	47
Çizelge 4.7. Rössler Tabanlı KFGA Optimizasyon Sonuçları (Ortalama Sonuç ve MSE) (20 defa koşma, NP=20, F=1.2 ve CR=0.7)	48
Çizelge 4.7. Rössler Tabanlı KFGA Optimizasyon Sonuçları (Ortalama Sonuç ve MSE) (20 defa koşma, NP=20, F=1.2 ve CR=0.7) (devamı).....	49
Çizelge 4.8. Rössler Tabanlı KFGA Optimizasyon Sonuçları (Ortalama İterasyon ve Ortalama Süre) (20 defa koşma, NP=20, F=1.2 ve CR=0.7).....	49
Çizelge 4.8. Rössler Tabanlı KFGA Optimizasyon Sonuçları (Ortalama İterasyon ve Ortalama Süre) (20 defa koşma, NP=20, F=1.2 ve CR=0.7).....	50

Çizelge 4.9. Lorenz Tabanlı KFGA Optimizasyon Sonuçları ¹ (NP=10)	51
Çizelge 4.10. Lorenz Tabanlı KFGA Optimizasyon Sonuçları ² (NP=20)	51
Çizelge 4.11. Lorenz Tabanlı KFGA Optimizasyon Sonuçları ³ (NP=30).....	52
Çizelge 4.12. Lorenz Tabanlı KFGA Optimizasyon Sonuçları ⁴ (NP=40)	52
Çizelge 4.13. Lorenz Tabanlı KFGA Optimizasyon Sonuçları ⁵ (NP=50)	52
Çizelge 4.14. Rössler Tabanlı KFGA Optimizasyon Sonuçları ¹ (NP=10).....	54
Çizelge 4.15. Rössler Tabanlı KFGA Optimizasyon Sonuçları ² (NP=20).....	54
Çizelge 4.16. Rössler Tabanlı KFGA Optimizasyon Sonuçları ³ (NP=30).....	55
Çizelge 4.17. Rössler Tabanlı KFGA Optimizasyon Sonuçları ⁴ (NP=40).....	55
Çizelge 4.18. Rössler Tabanlı KFGA Optimizasyon Sonuçları ⁵ (NP=50).....	55
Çizelge 4.19. Lorenz Tabanlı KFGA Optimizasyon Sonuçları ¹ (F=0.6)	58
Çizelge 4.20. Lorenz Tabanlı KFGA Optimizasyon Sonuçları ² (F=0.8)	58
Çizelge 4.21. Lorenz Tabanlı KFGA Optimizasyon Sonuçları ³ (F=1.0)	58
Çizelge 4.22. Lorenz Tabanlı KFGA Optimizasyon Sonuçları ⁴ (F=1.2)	59
Çizelge 4.23. Lorenz Tabanlı KFGA Optimizasyon Sonuçları ⁵ (F=1.4)	59
Çizelge 4.24. Lorenz Tabanlı KFGA Optimizasyon Sonuçları ¹ (CR=0.3)	61
Çizelge 4.25. Lorenz Tabanlı KFGA Optimizasyon Sonuçları ² (CR=0.4)	61
Çizelge 4.26. Lorenz Tabanlı KFGA Optimizasyon Sonuçları ³ (CR=0.5)	61
Çizelge 4.27. Lorenz Tabanlı KFGA Optimizasyon Sonuçları ⁴ (CR=0.6)	62
Çizelge 4.28. Lorenz Tabanlı KFGA Optimizasyon Sonuçları ⁵ (CR=0.7)	62
Çizelge 4.29. Rössler Tabanlı KFGA Optimizasyon Sonuçları ¹ (F=0.6)	64

Çizelge 4.30. Rössler Tabanlı KFGA Optimizasyon Sonuçları ² (F=0.8)	64
Çizelge 4.31. Rössler Tabanlı KFGA Optimizasyon Sonuçları ³ (F=1.0)	64
Çizelge 4.32. Rössler Tabanlı KFGA Optimizasyon Sonuçları ⁴ (F=1.2)	65
Çizelge 4.33. Rössler Tabanlı KFGA Optimizasyon Sonuçları ⁵ (F=1.4)	65
Çizelge 4.34. Rössler Tabanlı KFGA Optimizasyon Sonuçları ¹ (CR=0.3)	67
Çizelge 4.35. Rössler Tabanlı KFGA Optimizasyon Sonuçları ² (CR=0.4)	67
Çizelge 4.36. Rössler Tabanlı KFGA Optimizasyon Sonuçları ³ (CR=0.5)	67
Çizelge 4.37. Rössler Tabanlı KFGA Optimizasyon Sonuçları ⁴ (CR=0.6)	68
Çizelge 4.38. Rössler Tabanlı KFGA Optimizasyon Sonuçları ⁵ (CR=0.7)	68
Çizelge 4.38. Klasik FGA ile Lorenz tabanlı KFGA (KFGA1) ve Rössler tabanlı KFGA (KFGA2) performanslarının karşılaştırılması	73

ŞEKİLLER DİZİNİ

Sayfa No

Şekil 2.1. Euler metodu ile çözümlenmiş Lorenz Çekicisine ait çıkışların fazör diyagramı ve üç eksen görünümü ($x_0 = 0.1, y_0 = 0.1, z_0 = 0.1$).....	17
Şekil 2.2. Euler metodu ile çözümlenmiş Lorenz Çekicisinin x,y,z durum değişkenlerinin zamana göre değişimi ($x_0 = 0.1, y_0 = 0.1, z_0 = 0.1$)	17
Şekil 2.3. Runge-Kutta metodu ile çözümlenmiş Lorenz Çekicisine ait çıkışların fazör diyagramları ve üç eksen görünümü ($x_0 = 0.1, y_0 = 0.1, z_0 = 0.1$).....	18
Şekil 2.4. Runge kutta metodu ile çözümlenmiş Lorenz Çekicisinin x,y,z durum değişkenlerinin zamana göre değişimi ($x_0 = 0.1, y_0 = 0.1, z_0 = 0.1$)	18
Şekil 2.5. Euler metodu ile çözümlenmiş Chua devresinin durum değişkenlerinin zamana göre kaotik değişimleri ($\alpha=15.6, \beta=28, m_0=-1.143, m_1=-0.714, x_0 = 0.7, y_0 = 0, z_0 = 0$)	20
Şekil 2.6. Euler metodu ile çözümlenmiş Chua devresinin fazör diyagramları ve üç eksen görünümü ($\alpha=15.6, \beta=28, m_0=-1.143, m_1=-0.714, x_0 = 0.7, y_0 = 0, z_0 = 0$)	20
Şekil 2.7. Runge-Kutta metodu ile çözümlenmiş Rössler sisteminin x,y ve z durum değişkenlerinin zamana göre değişimi ($x_0 = 0.1, y_0 = 0.1, z_0 = 0.1$)	21
Şekil 2.8. Runge-Kutta metodu ile çözümlenmiş Rössler Çekicisine ait fazör diyagramları ve üç eksen görünümü ($x_0 = 0.1, y_0 = 0.1, z_0 = 0.1$)	22
Şekil 2.9. Euler metodu ile çözümlenmiş Mackey-Glass sisteminin durum değişkeninin zamana göre değişimi ($a = 0.2, b = 0.1, c = 10, x_0 = 0.1$)	23
Şekil 2.10. Euler metodu ile çözümlenmiş Mackey-Glass Çekicisine ait çıkışlar ($a = 0.2, b = 0.1, c = 10, x_0 = 0.1$)	23
Şekil 3.1. FGA Kaba Kodu	26

Şekil 3.2. FGA'nın akış diyagramı.....	29
Şekil 3.3. FGA'nın şematik diyagramı (Yiğit, 2007).....	30
Şekil 3.4. Kaotiklik için sayı üreticisine ait kaba kod.....	33
Şekil 3.5. KFGA'nın akış diyagramı.....	34
Şekil 3.6 Kaotik Tabanlı FGA Kaba Kodu (Yüzgeç ve Eser,2014).....	35
Şekil 4.1. Optimizasyon Test Problemleri.	38
Şekil 4.2. Ackley optimizasyon test problemi için Lorenz tabanlı KFGA'nın 10 tekrar sonucu (TS1-TS10) maliyet değişimleri ve ortalama maliyet değişimi (NP=20, F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1).....	42
Şekil 4.3. Himmelblau optimizasyon test problemi için Lorenz tabanlı KFGA'nın 10 tekrar sonucu (TS1-TS10) maliyet değişimleri ve ortalama maliyet değişimi (NP=20, F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1)	42
Şekil 4.4. Holdertable optimizasyon test problemi için Rössler tabanlı KFGA'nın 10 tekrar sonucu (TS1-TS10) maliyet değişimleri ve ortalama maliyet değişimi (NP=20, F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1)	44
Şekil 4.5. Rastrigin optimizasyon test problemi için Rössler tabanlı KFGA'nın 10 tekrar sonucu (TS1-TS10) maliyet değişimleri ve ortalama maliyet değişimi (NP=20, F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1).....	44
Şekil 4.6. Guinta optimizasyon test problemi için Lorenz tabanlı KFGA'nın farklı stratejilerinde maliyet değişimleri (NP=20, F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, tekrar sayısı=20).....	47
Şekil 4.7. Testtubeholder optimizasyon test problemi için Rössler tabanlı KFGA'nın farklı stratejilerinde maliyet değişimleri (NP=20, F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, tekrar sayısı=20).....	50
Şekil 4.8. Ackley optimizasyon test problemi için Lorenz tabanlı KFGA'nın farklı	

popülasyon sayılarında maliyet değişimleri (F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1, tekrar sayısı=20)	53
Şekil 4.9. Himmelblau optimizasyon test problemi için Rössler tabanlı KFGA'nın farklı popülasyon sayılarında maliyet değişimleri (F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1, tekrar sayısı=20)	57
Şekil 4.10. Rastrigin optimizasyon test problemi için Lorenz tabanlı KFGA'nın farklı skala değerlerinde (F) maliyet değişimleri (NP=20, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1, tekrar sayısı=20)	60
Şekil 4.11. Testtubeholder optimizasyon test problemi için Lorenz tabanlı KFGA'nın farklı Çaprazlama olasılık değerlerinde (CR) maliyet değişimleri (NP=20, F=1.2, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1, tekrar sayısı=20).....	63
Şekil 4.12. Holdertable optimizasyon test problemi için Rössler tabanlı KFGA'nın farklı Skala değerlerinde (F) maliyet değişimleri (NP=20, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1, tekrar sayısı=20)	66
Şekil 4.13. Guinta optimizasyon test problemi için Rössler tabanlı KFGA'nın farklı Çaprazlama olasılık değerlerinde (CR) maliyet değişimleri (NP=20, F=1.2, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1, tekrar sayısı=20).....	69
Şekil 4.14. Schweffel optimizasyon test problemi için Lorenz tabanlı KFGA'nın farklı iterasyonlardaki popülasyonlarının değişimi (NP=20, F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1)	70
Şekil 4.15. Rosenbrock optimizasyon test problemi için Lorenz tabanlı KFGA'nın farklı iterasyonlardaki popülasyonlarının değişimi (NP=20, F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1)	70
Şekil 4.16. Sixhumpcamel optimizasyon test problemi için Rössler tabanlı KFGA'nın farklı iterasyonlardaki popülasyonlarının değişimi (NP=20, F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1)	71
Şekil 4.17. Penholder optimizasyon test problemi için Rössler tabanlı KFGA'nın farklı	

iterasyonlardaki popülasyonlarının değişimi (NP=20, F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1) 71

Şekil 4.18. Sixhumpcamel optimizasyon test problemi için FGA ile Lorenz (KFGA1) ve Rössler tabanlı (KFGA2) yapılarının sonuçları (NP=20, F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1, tekrar sayısı=50) 72

Şekil 4.19. Rosenbrock optimizasyon test problemi için FGA ile Lorenz (KFGA1) ve Rössler tabanlı (KFGA2) yapılarının sonuçları (NP=20, F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1, tekrar sayısı=50) 73

SİMGELER VE KISALTMALAR DİZİNİ

- MSE : Mean squared error (Ortalama hata karesi)
- DE : Differential Evolution Algorithm (Farksal gelişim algoritması)
- FGA : Farksal gelişim algortiması
- KFGA : Kaotik tabanlı farksal gelişim algortiması
- CR : Çaprazlama sabiti
- F : Skala faktörü
- NP : Popülasyon değeri

1. GİRİŞ

Optimizasyon, bir probleme en uygun çözümler üretebilme çabası olarak tanımlanabilmektedir. Optimizasyonun tarihteki izlerine II. Dünya Savaşı'nda rastlanmaktadır. O dönemde İngiliz Ordusu düşman güçlerini saf dışı bırakmak için ihtiyaç duydukları kaynakların kıtlığı ve bu kaynakların paylaşımı gibi problemler ile karşı karşıya kalmışlardır. İngilizler bu gibi problemlere bilimsel çözümler üretmesi için matematikçilerin oluşturduğu bir ekibe başvurmuşlardır. Bu ekip tarafından geliştirilen yöntemler, İngiltere'nin bu mücadeledeki hava savaşını kazanmasına önemli ölçüde vesile olmuştur. Bu olayda görüldüğü gibi optimizasyon metotları, insanların mevcut kaynakları belirli şartlar altında en iyi şekilde değerlendirme arayışlarından var olmuştur (Yılmaz, 2014).

Burada bahsedilen şekliyle optimizasyon, benzetim sonuçlarına dayanır. Problem ayrıca, hem sürekli değişkenlerin varlığı, hem de ayrı ayrı bileşenlerin kataloglardan ve veritabanlarından seçimi olarak tanımlanır. Bunun sonucu olarak problem doğrusal değildir. Karmaşık metot ya da genetik algoritma gibi derece derece değişmeyen metotların uygulanmasının bir nedeni de budur. Diğer ana neden, bu metotlar çok modelli araştırma alanlarında yerleştirildiğinde genel uygunluğu daha sağlıklıdır. Bu metotlar algoritmalarda hiçbir değiştirme yapmaksızın geniş bir problem sahasına uygulanabilirler. Optimizasyonun bir tarafı tasarım taslaklarının değerlendirilmesidir. İkinci tarafı yeni ve daha umut vadeden tasarımların meydana getirilmesidir. Böylece, optimizasyon hem analizi (değerlendirme) hem de sentezi (yeni çözümlerin üretimi) içerir. Değerlendirme genellikle tasarım taslağının ne kadar iyi olduğunu belirleyen ve değer biçimini içeren amaç (maliyet) fonksiyonu vasıtasıyla yapılır. Her optimizasyon probleminde olduğu gibi amaç fonksiyonunun biçimlendirilmesi optimizasyonun sonuçları noktasında en önemli noktadır (Uzun, 2006).

Optimizasyon, bir gerçel fonksiyonu en küçükleme ya da en büyükleme amacı ile gerçel ya da tamsayı değerlerini, tanımlı bir aralıkta seçerek, bir fonksiyona yerleştirip, sistematik olarak bir problemi incelemek ya da çözmek işlemlerini ifade eder. Optimizasyon problemlerini çözmek için günümüzde, matematiksel ve sezgisel

teknikleri kullanan, birçok optimizasyon algoritması bulunmaktadır. Matematiksel (klasik) algoritmalar genellikle probleme özgü tasarlanan ya da problemin tüm çözüm uzayını tarayarak problemi çözmeye çalışan algoritmalarlardır. Büyük boyutlu lineer olmayan problemlerde, çözümü bulmaları çok maliyetli ve uzun süreler almaktadır. Sezgisel algoritmalar, çözüm uzayı geniş olan problemlerde, tüm çözüm uzayını taramadan, sezgisel olarak çok kısa sürelerde çözümü ya da çözüme en yakın sonucu elde edebilen algoritmalarlardır. Bunların başlıcaları Genetik Algoritma (Genetic Algorithm, GA), Benzetilmiş Tavlama (Simulated Annealing, SA), Farksal Gelişim Algoritması (Differential Evolution Algorithm, DE), Parçacık Sürü Optimizasyonu (Particle Swarm Optimization, PSO) ve Bal Arılarında Evlilik Optimizasyonu (Marriage in Honey Bee Optimization, MBO) olarak sayılabilir (Çelik, 2013).

Özellikle geçen 30-40 yılda evrim (evolution) prensibine dayalı algoritmalara ilgi artmış ve buna bağlı olarak evrimsel algoritmayla ilgili literatürdeki çalışmaların sayısı da hızla artmıştır. (Holland, 1975; Goldberg, 1989; Davis,1991; Michalewicz, 1992; Pham and Karaboğa, 2000). Optimizasyon problemleriyle karşılaşıldığında genellikle ilk yapılan, probleme özel sezgisel bir yaklaşım tekniği belirlemektir. Gelişime dayalı algoritmalar, genel metotlarla karşılaştırıldığında oldukça üstün performans göstermektedirler. Son zamanlarda bu prensibe dayanan tekniklerin hepsini temsilen ortak bir terim olarak Evrimsel Hesaplama terimi yaygın olarak kullanılmaya başlanmıştır (Karaboğa, 2011).

Sürekli parametrelerin söz konusu olduğu problemlerin çözümüne yönelik geliştirilmiş algoritmalarından olan Farksal Gelişim Algoritması (FGA), Price ve Storn tarafından 1995 yılında geliştirilmiştir (Storn ve Price, 1995; Price ve Storn, 1995). Popülasyon tabanlı sezgisel bir algoritma olan FGA, özellikle tamamen düzenlenmiş uzayda tanımlı ve gerçek değerli tasarım parametrelerini içeren fonksiyonları optimize etmek amacıyla kullanılan bir algoritmadır (Karaboğa, 2004). FGA'nın temel fikri popülasyondaki iki bireyin arasındaki farkın bir üçüncü bireye ilave edilmesidir. Genetik Algoritmalarından farklı olarak, FGA tüm optimizasyon süreci boyunca sabit olan çaprazlama oranı, ölçekleme faktörü ve popülasyon boyu gibi birkaç kontrol parametresine sahiptir ve kendi yapısı içerisinde kayan noktalı sayıların gerçek kodlamasını kullanır. Kontrol parametrelerinin değerleri arama algoritmasının verimi ve

elde edilen çözümün kalitesini artırmak için dikkatli bir şekilde seçilmelidir. FGA'nın verimliliği ve gürbüzlüğü tam olarak bu kontrol parametrelerinin ayarlanmasına bağlıdır (Yüzgeç, 2011).

FGA'da kullanılan basit mutasyon işlemi, algoritmanın performansını geliştirmekte ve onu daha gürbüz yapmaktadır. Bu özelliğinin yanı sıra diğer özellikleri içinde şunlar söylenebilir: hızlı, basit, kolay kullanılabilir ve değiştirilebilir, etkili küresel optimizasyon kabiliyetli, doğal olarak paralel kayan-nokta formatına bağlı hassasiyet sınırlamalı matris çarpımları ve sıralama işlemleri olmadığı için hesaplama maliyeti açısından avantajlı, daha önceden tanımlanmış herhangi bir olasılık dağılımlı mutasyon kullanmamakta, tam sayı, ayrık ve karışık parametre optimizasyonuna kolaylıkla uyarlanabilir, amaç fonksiyonunun veya sınırlama fonksiyonların türevine gerek duymaz, düz yüzeylerde çalışabilir, gürültülü ve zamana bağlı amaç fonksiyonları için kullanılabilir, tek bir koşmada çoklu çözümler üretebilir ve özellikleri doğrusal olmayan sınırlamalı optimizasyon problemlerinde etkilidir (Karaboğa, 2011).

FGA yapısının ilk ortaya çıkmasından bu yana konuyla ilgili birçok çalışmalar yapılmıştır. Bu çalışmalardan bir kısmı algoritmanın iyileştirilmesi üzerinedir. Bergey ve Ragsdale çalışmalarında, FGA'da seçim aşamasında, seçim baskısı adı verilen bir yöntem öne sürmüşler ve önerdikleri modifiye edilmiş FGA yapısını klasik FGA ile karşılaştırmasını yapmışlardır (Bergey ve Ragsdale, 2005). Hrstka ve Kucerova çalışmalarında, lokal noktalardan kaçınmak için farksal operatörlere dayanan, gerçek kod tabanlı GA (Gelişim Algoritması) önermişler ve önerdikleri algoritmayı literatüreden alınan çeşitli optimizasyon problemleri üzerinde koşturmuşlardır (Hrstka ve Kucerova, 2004). Becerra ve Coello çalışmalarında, daha az uygunluk fonksiyonu kullanarak, diğer FGA'larla aynı başarıyı gösteren Kültürel FGA yapısı sunmuşlardır (Becerra ve Coello, 2005). Sun ve arkadaşları çalışmalarında, FGA'yla elde edilen diferansiyel bilgiler ile Dağıtım Algoritması ile elde edilen global bilgileri kullanarak hem FGA, hem de Dağıtım Algoritmasından daha iyi performans elde eden hibrit bir algoritma geliştirmişlerdir (Sun vd, 2005). Farklı problemlere yönelik, farklı yapılar ve farklı parametrelerle çalışan birçok FGA geliştirilmiştir. Zaharie (2002)'nin kontrol parametrelerinin seçimi konusunda, Lin ve arkadaşlarının (2004) farklı değişken tipleriyle çalışabilen FGA'lara yönelik çalışmaları bulunmaktadır (Keskintürk, 2006).

Tizhoosh (2005) ve Rahnamayan vd. (2006) tarafından geliştirilen karşıtlık temelli FGA'da algoritmada, temel olarak eş zamanlı her bir jenerasyonda en iyi çözümü elde etmek için öngörü uzayı ile bu uzayın karşıtı arama uzayı da kullanması fikrine dayanmaktadır.

Brest vd. (2006) tarafından adaptif FGA yapısında skala faktörü (F), çaprazlama olasılık sabiti (CR) gibi kontrol parametreleri rastgele işlemlere bağlı olarak kendiliğinden adaptif bir yaklaşımla ayarlanır. Kısa zamanda çözüme varmayı kontrol parametrelerinin en iyi değerleri sağlamaktadır.

Yüzgeç (2011) tarafından geliştirilen adaptif karşıtlık temelli FGA yapısında FGA yapısının parametrelerinin adaptiflik yapısına, karşıtlık temelli FGA'da kullanılan atlama katsayısına adaptiflik eklenmesi önerilmiştir.

Terzi (2009) tarafından geliştirilen algoritmada ağırlıklı olarak sürekli eniyileme problemlerinin çözümünde kullanılan yeni bir çözüm yöntemi önerilmiştir. Sürekli eniyileme problemlerine yönelik olarak geliştirilmiş olduğundan, kesikli ya da permütasyonel eniyileme problemleri için etkin uygulamaları bulunmamaktadır. Bu algoritma, FGA permütasyonel eniyileme problemlerinde kullanılmak üzere yeniden düzenlenmiş ve geliştirilmiştir.

Bu çalışmada, FGA'da kullanılacak olan kaotiklik kavramının bilim tarihine girişi ise 18. Yüzyılda olmuştur. Kaos alanındaki en önemli keşiflerden biri, 1963 yılında havanın basitleştirilmiş bir modelini çalışmak üzere basit bir matematiksel bilgisayar programı yazan meteorolog Edward Lorenz tarafından yapılmıştır (Gündüz, 2002). Lorenz, hava tahmininde başlangıç koşullarındaki çok küçük değişikliklerin belirgin farklılıklara sebep olduğunu fark etmiştir. Lorenz'in atmosfer modelinde kullandığı matematik 1970'lerde geniş bir biçimde araştırılmış ve zamanla, kaotik bir sistemin temel özelliği olarak iki farklı başlangıç koşulları dizisindeki en küçük farklılığın, sistemin durumunda büyük farklılıklara neden olacağı, bilinen bir gerçek haline gelmiştir. Dinamik sistemler, deterministik ve stokastik olarak sınıflandırılırlar. Deterministik dinamik sistemler, durumları zamanla deterministik şekilde değişen sistemlerdir (Kennedy, 1999). Evrensel olarak analog ve sayısal haberleşme sistemlerinde taşıyıcı olarak kullanılan sinüzoidal sinyaller, sürekli deterministik dinamik sistemlerin periyodik çözümleridir. Deterministik dinamik sistemler aynı zamanda geniş güç spektrumlu sürekli "gürültüsel" yapı ile karakterize edilen, periyodik

olmayan sinyallerden oluşan kaotik sinyalleri de içine alır. Zaman bölgesindeki kaotik sinyaller rastgele bir şekilde ortaya çıkarlar. Kaotik sistemler için iki temel kural söylenebilir:

- a) Başlangıç şartlarına hassas bağımlılık: Kaotik sistemler, başlangıç şartlarına çok hassas bir biçimde bağımlıdırlar.
- b) Rastgele olmama durumu: Kaotik sistemler, tarafınızdan belirlenemese de içsel bir düzene uyarlar, rastgele değildirler. (Yardım, Afacan, 2009).

Nükleer fizik, katı hal fiziği, lazer optiği, kimya, biyoloji, tıp, ekoloji, astronomi, sosyoloji, ekonomi, uluslararası ilişkiler, tarih, hidrolik, atmosferik, elektrik, elektronik, makine gibi mühendisliğin ve diğer bilimlerin çok çeşitli dallarında kaos varlığının ortaya konması, konuyla ilgili yapılan yoğun çalışmalar ve beraberinde yaşanan gelişmeler kaos ve kaotik sistemlerle ilgili bir çok uygulama alanının doğmasına yol açmıştır. Kaos olayına ve kaotik sistem dinamiğine yönelik geçen 10-15 yıl içerisinde çok büyük bir ilgi olmuştur. Kaos ve kaotik sistemlerle ilgili oluşan uygulama alanlarına örnek olarak;

- meteorolojik tahmin,
- kaotik paralel dağılımlı işleme,
- deterministik doğrusal olmayan tahmin,
- kimliklendirme ve doğrusal olmayan sistemlerin modellenmesi,
- doğrusal olmayan filtreleme,
- biyomedikal ve tıbbi uygulamalar,
- dinamik bilgi sıkıştırma ve kodlama, kaotik güvenilir haberleşme,
- hassas desen tanıma,
- kaotik dinamiklerin müzik ve sanat amaçlı kullanımı,
- kaotik salınımların yapay olarak oluşturulması,
- kaotik sistemlerin elektronik,
- optik ve optoelektronik olarak gerçekleştirilmesi,
- kaotik titreşim ve salınımların belirlenmesi ve kontrol edilmesi,
- lazerlerin kontrolü,
- türbülans kontrolü,
- vinç ve gemi salınımlarının kontrolü,

- popülasyon tahmini olarak verilebilir (Pehlivan, 2007).

Kaos kavramının ve kaotik sistem özelliklerinin ortaya konmasıyla literatürde kaos olayıyla ilgili çalışmalar iki ana bölümde odaklanmıştır. Bunlardan ilki, kaosun ve kaotik davranışın olumsuz olarak algılandığı ve bu tür davranışların görülmemesi arzulan sistem yapılarında kaotik kontrol çalışmalarıdır (Murali vd., 1995; Pehlivan, 2007).

Kaos ve kaotik sistem dinamiği ile ilgili ikinci çalışma alanı ise; bu derece ilginç özelliklere sahip kaotik işaretler ve sistemlerden olumlu yönde yararlanma fikri doğrultusunda yapılan çalışmalar olmuştur (Pehlivan, 2007).

Yapılan tez çalışması beş bölümden oluşmaktadır. Birinci bölümde tez konusuna genel giriş ve literatür araştırması verilmiştir. İkinci bölümde kaotik sistemler hakkında genel bilgiler ve kaotik sistemlerin çözümlerine ait sonuçlar sunulmuştur. Üçüncü bölümde genel FGA yapısı ve kaotik tabanlı FGA yapısı anlatılmıştır. Dördüncü bölümde önerilen KFGA'nın performansını değerlendirmek için literatürden alınan on adet optimizasyon problemi kullanılarak, algoritma belirli tekrar sayısı kadar koşturulmuştur. Beşinci bölümde (tartışma ve sonuç bölümünde) KFGA ile elde edilen bulguların karşılaştırılmasından elde edilen sonuçlar tartışılarak, gelecekte yapılacak olan çalışma başlıklarından bahsedilmiştir.

2. KAOS VE KAOTİK SİSTEMLER

2.1 Kaos ve Kaotiklik

Klasik dönemdeki bilimsel yöntemler, sistemlerdeki düzensiz, kararsız ve birbirleriyle bağlantısız gibi gözükten davranışları göz ardı etmiş, gürültü veya dış etki olarak adlandırmış ve tasarım yoluyla bunlardan kaçınmaya çalışmıştır (Yiğit, 2009).

Ancak 17. yüzyılda Isaac Newton'un çalışmaları, fiziksel sistemlerin hareketlerinin diferansiyel denklemlerle ifade edilebileceğini göstermiş ve diferansiyel denklemleri parçacık ve gezegen hareketlerinin incelenmesinde kullanarak dünyanın güneş etrafındaki hareketini hesaplayan problemi çözmüştür. Daha sonra gelen bilim adamları Newton'un çalışmasını Güneş, Dünya ve Ay'ın hareketini hesaplayacak şekilde genişletmek istemişler ancak uzun yıllar başarısız olmuşlardır (Strogatz,1994; Alligood vd. 1997; Yiğit, 2009).

Dinamik sistemlerin doğuşu onyedinci yüzyılın ikinci yarısına, Newton'un diferansiyel hesap, hareket kanunları ve yerçekimi teorisini buluşuna kadar uzanır. Ondokuzuncu yüzyılın sonlarında, 1892 yılında Fransız matematikçi Henri Poincare yeni ufuklar açan bir araştırma (Holmes, 1990) ile basit dinamik kuralların çok karmaşık kararlı-hal davranışlarına yol açabileceğini, zamana göre değişimi Hamilton denklemleri ile yönlendirilen mekanik sistemlerin karmaşık davranışlar gösterebileceğini keşfetti. Günümüzde bu davranışlar, "kaotik davranışlar" olarak adlandırılmaktadır. Ayrıca Poincare, kaotik yörünge denilen çok karmaşık yörüngelerin mümkün olduğunu ve başlangıç şartlarına hassas bağlılık gibi kaotik dinamiklerin çok önemli özelliklerini gösterdi (Pehlivan, 2007).

Van der Pol, 1927 yılında Nature Magazine adlı dergide (Kennedy, 1995) yayımlanan "Frequency demultiplication" adlı makalesinde, sinüzoidal kaynakla sürülen, neon tüp osilatöründeki periyot çoğullama olayını telefon ahizesindeki kulaklığı kullanarak gözlemiştir. Van der Pol, kapasite değerinin değişimi ile frekanstaki değişimleri bir değerden sonra sık sık düzensiz bir gürültü şeklinde kulaklığıyla fark etmiş. Van der Pol, Feigenbaum'un 1975 yılında söylediği periyot çoğullama, kaosa götürdüğünü devrede gözlemiş, fakat o zamanki bilgilerle çıkan sonucu açıklayamadığı

için kaosu gürültü sanmıştır. 1986 yılında (Kennedy, 1986) M. Peter Kennedy, Van der Pol'un çalışmasını tekrar incelemiş ve gürültü olarak adlandırılan şeyin aslında kaos olduğunu göstermiştir (Pehlivan, 2007).

Bilimsel toplumun önemli bir kısmı dikkatini bu çeşit olaylara çevirmesi 1960 yıllarının sonlarını buldu. Böylece yeni bir bilim olan "Kaos Bilimi" gelişmeye başladı. Edward Lorenz bu ilerlemenin öncüsü olmuştur. MIT bilimcisi E. N. Lorenz hava durumunu önceden belirleyebilmek için atmosferdeki akışkan ısı-yayınımını benzetim yaparken, yeni tip düzensiz salınımlar gözlemlemiştir (Lorenz, 1963) ve bir model önermiştir. Lorenz'in keşfinin önemi, yayınlanmasından çok yıl sonralara kadar anlaşılammıştır. Lorenz sistemi geniş ölçüde çalışılmış ve dağıtık sistemlerin kaotik davranışlarını tanımlamak için ilk örnek olarak kabul edilmiştir. 1975 yılında M. J. Feigenbaum' un periyot çoğullamayı kaosun bir belirtisi olarak sunduğu çalışma (Moon, 1987) bunlardan birisidir. 1975 yılında, Li ve Yorke (Li, 1975) bu çeşit davranışı belirtmek için "kaos" terimini kullanmayı önermişlerdir (Pehlivan, 2007).

1970'li yılların ortalarında bilgisayarların gelişmesi, hızlanması ve yaygın kullanımı ile dinamik sistemler teorisi üzerindeki gelişmeler, matematik, mühendislik ve farklı bilimsel alanlardaki geniş sayıda araştırma gruplarını kaotik davranışları gözlemlemeye yöneltti. Kaosun Ekoloji, Kimya, Akışkanlar mekaniği, katı-hal aygıtları, biyoloji ve gökyüzü mekaniğine varacak kadar çeşitlilikte alanlardaki problemlerle ilişkili olduğu gösterildi. Aslında kaotik dinamikler uzun zamandan beri bilinmesine rağmen doğrusal modellerin bilimdeki başarısı, güçlü bilgisayarların eksikliği gibi sebeplerle geniş çeşitlilikte uygulamalar ancak son kırk yıl içinde görülmeye başlanmıştır. Yüksek hızlı bilgisayarlar ve bilgisayar grafikleri, doğrusal olmayan dinamikler ve kaos alanındaki ilerlemelerin en önemli unsurlarından olmuştur (Pehlivan, 2007).

Kaos, en kısa tarifiyle, doğrusal olmayan olayları açıklamaya yarayan bir bilim dalıdır. Karmaşık, ama kendi iç düzenine sahip bir süreçtir. Özellikle dikkat edilmesi gereken bir nokta, kaosun rastgelelik olmadığıdır. Kaos, karmaşık davranışlar gösteren kendine has bir "düzen" dir. Dinamik sistemlerde bilinen en karmaşık kararlı hal davranışı kaostur. Kaos ile ilgili çalışmalar, doğrusal olmayan dinamik sistemler teorisinin bir kısmıdır. Bu durum daha çok "deterministik kaos" olarak bilinir. Aynı

zamanda nedeni ve seyri bilinemeyen, hesaplanamaz olan “rastlantısal (stokastik) kaos” kavramı da mevcuttur. Fakat bilimin ilgilendiği daha ziyade deterministik kaostur (Pehlivan, 2007).

Kaos kavramı günlük dildeki kullanımından farklı olarak bilimsel anlamda “düzensizliğin içindeki düzen” manasında kullanılmaktadır. Kısaca günlük dildeki kullanımı ile bilimsel kullanımı arasında oldukça önemli fark vardır. Kavram ile ilgili en doğru tanımı veren teorik fizikçi Jensen, kaos’u “kompleks, doğrusal olmayan dinamik sistemlerin düzensiz ve öngörülemez davranışı” şeklinde ifade eder. Tanımda yer alan kompleks ifadesi karmaşıklığa, doğrusal olmayan (nonlinear) ifadesi özgün bir matematiksel yapıya, dinamik ifadesi ise sabit olmayan değişken bir yapıya işaret etmektedir (Erdemir, 2008; İskurt, 2012).

Kaos, deterministik bir sistemin düzensiz yani hiç beklenmedik bir şekilde davranabilmesidir. Kaotik sistemlerin en önemli özelliği başlangıç şartlarına hassas duyarlılıklarıdır. Deterministik bir sistemin başlangıç durumu ve denklemleri biliniyorsa, sistemin sonraki davranışı belirlenebilir. Kaotik sistemlerde, sistemin zaman içindeki gelişimini tam olarak belirleyebilmek için başlangıç değerlerini sonsuz hassasiyetle bilmek gerekmektedir. Çünkü kaotik sistemler doğrusal olmadıkları için hata zamanla üstel olarak artacaktır (Keunen, 1996; Williams, 1997; Baker, 1990; Yılmaz, Güler, 2006). Newton kanunlarından elde edilen dinamik denklemler düzgün akışları ifade edebilirken, akışkanın akış hızı belirli bir değeri aştıktan sonra akışta girdaplar oluşur ve Newton kanunları geçerliliğini yitirir. Yani artık akış kaotiktir. Sigara dumanı birtakım düzensiz helezonlar şeklinde dönerek yükselir. Bir sigara dumanının havada yaptığı şekiller tamamen düzensiz ve bağımsız rastlantıların ürünü olarak görülebilir. Ancak bir teorik fizikçi dumanın bu dinamiğinin aslında ortamdaki birçok parametre ve etken ile belirlendiği görüşündedir (Gleick, 2008; İskurt, 2012).

Eskiden beri bilimdeki temel bir inanış, deterministik sistemlerin önceden belli olmasıdır. Verilen deterministik model, başlangıç şartları ile çalışma altındaki bir sistemi tanımlar ise, sistem davranışı bütün zamanlar için önceden bilinebilir. Oysa gerçek hayatta gerçekleşen olayların çoğunda sistem, bileşenlerinin belli bölgelerdeki değişimleri için doğrusal bir davranış gösterir. Bu bölgelerin dışında sistem doğrusal olmayan davranış sergiler (Pehlivan, 2007).

Kaosun ve kaotik işaretlerin başlıca önemli özellikleri aşağıda sıralanmıştır:

- Zaman boyutunda düzensizlik,
- Başlangıç şartlarına aşırı hassasiyet,
- Sınırsız sayıda değişik periyodik salınımlar içermesi,
- Gürültü ve benzeri geniş güç spektrumuna sahip olması,
- Limit kümesinin parçalı (fraktal) boyutlu olması,
- Genliği ve frekansı tespit edilemeyen, ancak sınırlı bir alanda değişen işaretler içermesidir (Pehlivan, 2007).

Kaosun meydana gelmesi, belirli parametrelere bağlı olduğu gibi sistemin yapısına da bağlıdır. Kaos genellikle kararsız, karmaşık ve doğrusal olmayan sistemlerde ortaya çıkmaktadır. Karmaşık sistemler, çok sayıda elemanın birbiriyle etkileştiği, pek çok serbestlik derecesi olan, yani çeşitli davranış şekilleri gösterebilen, genellikle de dışarıyla madde ve enerji alışverişi yapan, incelenmesi zor sistemlerdir. Doğrusal olmayan bir sistem, değişim anında değişim kurallarının da değiştiği bir sistemdir ve sistem, dışarıdan gelebilecek etkilere karşı açıksa sistemden beklenmeyen davranış biçimleri görülebilir. Örneğin, hava direncinin hızın küpüyle değiştiği bir sarkaç deneyinde, dışarıdan periyodik bir kuvvetin etkisiyle sürtünme katsayısının belli bir değerinden sonra kaotik bir davranış görülmektedir (Strogatz,1994; Yılmaz, Güler, 2006).

Gerçek hayattaki zamana göre oluşan polen üretimi, nüfus artışı, ekonomik değişimler, dünya buz kütleleri gibi süreçler kaotik olabilir. Fizik, kimya, matematik, iletişim, biyoloji, fizyoloji, sosyoloji, ekonomi, tarih, ekoloji, astronomi, hidrolik, atmosferik, uluslararası ilişkiler, solar sistemler, mühendislik gibi alanlarda kaotik çalışmalar görülmektedir (Adem, 1997; Williams, 1997; Baker, 1990; Yılmaz, Güler, 2006).

2.1.1 Kaotik sistemlerde determinizm

Dinamik sistemler çeşitli dinamik kurallar ile tanımlanabilir ve bu dinamik sistemlerin çözümleri tektir. Bu nedenle bu tip sistemler determinist olarak adlandırılmaktadır. Dinamik sistemin durumu tüm zamanlar için tektir. Sistemin herhangi bir zamandaki durumuna karar verilebilir (Baker, 1990; Yiğit, 2009).

Determinist doğrusal olmayan dinamik sistemlerin çözümleri ise rastgele

(istatistiğe bağılı durumlar) olabilir. Bu tip davranış determinist kaos olarak adlandırılmaktadır. Burada önemli olan temel nokta, çözümlerinin gerçekten rastgele olup olmadığıdır. Rastgelelik temel olarak gürültü aracılığıyla üretildiğinden buradaki sonuçlar rastgele değildir. Sonuçların rastgele görünmesinin nedeni başlangıç koşullarına bağımlı olan determinist denklemlerde kullanılan değişkenlerin değerindeki çok küçük değişimlerdir.

Kaotik sistemler determinist olmasına karşın öngörülebilir değildir. Kaotik sistemlerin başlangıç koşullarına aşırı duyarlılığı, başlangıç değerinin hassas ölçülememesi veya başlangıç değerinin irrasyonel sayılarla ifade ediliyor olabilmesi öngörülebilirliğe engel olur. Kaotik sistemler doğrusal olmayan yapılarından ötürü öngörülebilir olmaması, ama aynı zamanda determinist bir yapıya da sahip olması, bu kavramların ayrı ayrı ele alınması gerektiğini göstermiştir.

2.1.2 Başlangıç koşullarına aşırı duyarlılık

Kaotik dinamiklerin önemli özelliği, sistem iki farklı başlangıç koşulunda başlatıldığında gözlemlenebilir. Bu küçük farklar kaotik olmayan sistemlerde ölçme hatası olarak ifade edilebilir ve hata zamanla doğrusal olarak artmaktadır. Fakat kaotik sistemlerde hata üstel olarak artmakta ve sistemin gelecekteki durumları kestirilemez olmaktadır. Bu olgu başlangıç koşullarına duyarlılık olarak bilinmektedir. Bu alandaki ilk çalışma meteorolojik olayları modellemek için hava tahminlerinden elde edilen sayısal veriler kullanılarak bir model geliştirme fikri sonucu ortaya çıkmıştır. Ancak hava durumunun sahip olduğu kaotik durumdan ötürü sayısal verilerin hassasiyeti her bir denemede artmasına rağmen geçerli bir model elde edilememiş, verilerdeki küçük bir ihmal farklı sonuçlara neden olmuştur. Kaos hali binlerce kararlı ve kararsız durumların birlikte var olduğu bir durumdur. Kararlı bir hal en küçük bir değişiklikle kararsızlığa, kararsızlık ise kararlılığa dönüşebilir. Bu nedenle “kararlı kararsızlık” durumu diye de tanımlanabilir. Kararlı kararsız bölgeler aşırı derecede küçük olduğu için kararlı bölgedeki en küçük bir değişiklik sistemi kaosa götürür.

2.1.3 Kaos analizi için gerek ve yeter koşullar

Bir sistemde kaos analizi yapılabilmesi için bazı gerekli şartlar vardır. Bunlardan ilki, sistemde doğrusal olmayan eleman veya elemanlar olmasıdır. Doğrusal sistemde

kaosun gözlenmeyeceği bilinmektedir. Şartlardan ikincisi ise, sürekli zamanlı sistemler için ve ayrık zamanlı sistemler için farklılık göstermektedir. Sistem sürekli zamanlı bir sistem ise, kaosun aranabilmesi için ikinci şart en az 3. dereceden bir sistem olmasıdır. Fakat ayrık zamanlı sistemde böyle bir şart yoktur. Ayrık zamanlı sistem birinci dereceden dahi olsa kaos analizi yapılabilir. Lojistik harita (logistic map) buna bir örnektir. Kaos analizi yapabilmek için bahsedilen iki şart gerekli şartlardır, yeterli şartlar değildir. Yani bu şartlara uyan sistemlerde kaos analizi yapılabilir fakat kesinlikle kaotik davranış gösterir şeklinde bir yargıya varılamaz. Bir sistemde kaos gözlenebilmesi için yeter şart ise sistem yörüngesinin başlangıç koşullarına duyarlı olmasıdır.

2.2 Kaos Ölçme Yöntemleri

Bir sistemin kaotik olup olmadığını anlamak için ilk olması gereken şey, sistemin davranışına dair olabildiği kadar uzun süreyle kaydedilmiş bir değişkenler kayıdır. Sistemin zamanla değişen parametrelerini gösteren ve sistemin zaman içinde nasıl bir davranış gösterdiğinin bir yansıması olan bu tip verilere “zaman serileri” adı verilir. Zaman serileri, herhangi bir değişkenin, zamanın bir fonksiyonu olarak değişimini gösteren verilerdir. İnsan kafatasının üzerinden kaydedilen elektroensefalogram verileri, yıllara göre fiyat endeksleri, yerkaşu hareketlerinin kayıtları, yahut çalışan bir makinanın yüksek frekanslı titreşimleri, zaman serilerine bir kaç örnek olarak sayılabilir. Bir zaman serisinin kaotik olup olmadığını anlamak için sık kullanılan bazı matematiksel yöntemler şunlardır (http://www.sinancanan.net.tr/2011/01/kaos-karmasklk-bilimi-ve-yeni-bilimsel_22.html) :

- Çeker oluşturma (attractor construction)
- Lyapunov Üstelleri
- Poincaré Kesiti (Poincaré Section)
- Doğrusalsızlığın Tespiti (Detection of Nonlinearity)
- Fraktal boyut analizi
- Yörüngenin izlenmesi (Zaman serileri)
- Faz uzayının incelenmesi
- Güç spektrumu
- Çatallaşma Diyagramı (Kavak, 2009)

2.2.1 Çeker oluşturma (Attractor Construction)

Zamanla değeri değişen bir değişkenin kaotik analizi için ilk basamaklardan birisi genellikle sistemin davranışının faz uzayındaki görünümünün elde edilmesidir. Bir dizi karmaşık hesap gerektiren bu süreç, bilgisayarlar yardımıyla bugün kolaylıkla gerçekleştirilebilmektedir. Bazı yazılımların içinde bu işlem için kullanılabilir hazır makrolar ve kod parçaları mevcuttur. Çeker oluşturmak için bilinmesi gereken en önemli parametre “gömme boyutu” (embedding dimension) denen parametredir. Gömme boyutu, sistemin davranışlarını etkileyen bağımsız dinamik kaynakların sayısını tahmin eden bir hesaplama ve böylece incelenen sistemin davranışının en iyi biçimde görsel hale getirilebilmesi için kaç boyutlu bir faz uzayına ihtiyaç olduğu bu şekilde hesaplanır. Görsel tutarlılık açısından üç boyuttan daha büyük gömme boyutları pek tercih edilmese de bazı karmaşık kaotik sistemlerde çok daha büyük boyutlu faz uzaylarına ihtiyaç duyulabilmektedir. Gereken bir diğer parametre de “zaman gecikmesi” (time delay) parametresidir. Bu hesaplama sonucunda, zaman serisinin hangi zaman aralıklarında geciktirilerek grafiğe dökülmesi gerektiği hesaplanır.

2.2.2 Lyapunov üstelleri

İlk defa Aleksandr Mikhailovich Lyapunov (1857-1918) tarafından tanımlanan bu yöntem, bir zaman serisinin kaotik bileşenler içerip içermediğinin anlaşılmasına yarayan matematiksel bir analiz yöntemidir. Lyapunov üsteli, bir sistemin olası durumlarını gösteren “çeker”ler üzerinde, başlangıçta yakın komşu olan iki rastgele noktanın birbirlerinden ayrılma derecesinin sayısal bir ifadesidir. Eğer bu komşu noktalar hızla birbirlerinden ayrılıyorsa, hesaplanan en büyük Lyapunov üsteli pozitif bir değerde olacaktır ve bu da incelenen sistemin davranışının kaotik olduğuna dair önemli bir işarettir. Başka bir deyişle Lyapunov üsteli, “başlangıç şartlarına hassas bağlılık” özelliğinin sayısal bir göstergesidir.

En büyük Lyapunov üstelinin pozitif olması kaotik durumun bir göstergesidir. Lyapunov üstellerinin sayısı sistemin kurgulandığı faz uzayının boyut sayısına göre değişir. Örneğin, üç boyutlu bir faz uzayında karşılaşılabilecek Lyapunov üstelleri (λ_1 , λ_2 , λ_3) şöyledir; (-,-,-): sabit nokta, (0,-,-): limit döngü, (0,0,-): simit, (+,0,-): garip çeker (kaos). Lyapunov üsteli hesaplamaları genellikle uzun süreli ve temiz kaydedilmiş

zaman serileri üzerinde en iyi sonucu verirken, daha kısa süreli ve kısmen gürültülü sinyaller üzerinde yapılacak hesaplamalar için ilave bazı algoritmalar kullanılması gerekir.

2.2.3 Poincaré kesiti (Poincaré Section)

Oluşturulan çekerler (attractor) genellikle çok karmaşık yapılara sahip olabilirler ve görsel olarak incelenmesi çoğu zaman oldukça zordur. Poincaré kesitleri olarak bilinen yöntem bu zorluğu aşmadaki en önemli yardımcılardan birisidir. Adından da anlaşılacağı üzere, bu yöntemle, karmaşık yapılı kaotik çekerlerin istenen herhangi bir noktasından geçen kesitler alınarak, bu kesitlerin görünümüne ve özelliklerine göre sistem hakkında bazı yargılara varılabilir.

Faz uzayına çizilen çekerlerden elde edilen kesitlerin görüntüleri sistemin dinamikliği hakkında da bir fikir verir. Sonuç olarak, Poincaré kesitindeki noktaların dağılımı tek ve küçük bir bölgede sonlu sayıda ise hareket periyodik, kapalı bir eğri ise hareket yarı periyodik, belirli alanlarda yoğunlaşmış kümeler şeklinde ise hareket kaotiktir.

2.2.4 Doğrusalsızlığın tespiti (Detection of Nonlinearity)

Bir zaman serisinde izlenen sinyallerin doğrusal olup olmadığını anlamının da bazı matematiksel yolları vardır. Bir dizi karmaşık matematiksel teknikle, bilgisayarların hızlı işlem gücünü de kullanarak bugün bu işlemler hızlı bir biçimde yapılabilmektedir. Bu amaçla en çok kullanılan yöntem “vekil veri analizi” (surrogate data analysis) denen yöntemdir. Bu analiz tipinde, eldeki sinyalin bir benzerini oluşturmak için doğrusal (linear) bir algoritma kullanılır ve üretilen yapay (vekil) sinyalle gerçek sinyal arasındaki ilişkiler incelenir. Eğer ilişki yoksa, sonuçta sinyalin doğrusal olmadığı gösterilmiş olur. Bu yöntemin yanında daha başka bir çok hesaplama tekniği de önerilmiştir, fakat hepsinin de sadece belli durumlarda geçerli olmasına neden olan bazı zayıflıkları vardır (Yılmaz ve Güler, 2006).

2.2.5 Fraktal boyut analizi

Daha önce bahsedildiği gibi “fraktal” terimi, değişik ölçeklerde artarak

karşımıza çıkan karmaşıklığın bir ifadesidir. Bir geometri alanı olmasının yanı sıra, özellikle zaman serilerinin karmaşıklık ve kaotiklik özelliklerini belirlemek için kullanılır. Fraktal sinyal analizi, fraktal doku analizi gibi farklı tekniklerin kullanıldığı bu tip analizlerde temel amaç zaman serisinin karmaşıklığının saptanmasıdır. Bir zaman serisinin fraktal boyutlarının artışı sürecin karmaşıklığının bir ölçüsü olarak kullanılmaktadır (http://www.sinancanan.net.tr/2011/01/kaos-karmasklk-bilimi-ve-yeni-bilimsel_22.html).

2.3 Örnek Kaotik Sistemler

2.3.1 Lorenz sistemi

Her ne kadar kaos kavram ve teorisinin babası olarak J.Henri Poincare kabul edilse de, teoriye en önemli katkıyı 1960 yılında MIT’de meteoroloji Profesörü olan Edward Lorenz yapmıştır. Edward Lorenz ile kaos teorisine doğru gidiş hızlanmıştır. Lorenz bir meteoroloji modeli kurmuştur. Bu modelde hava tahmini yapmayı amaçlamış ve modelde gözlemlendiği değişimleri 1963 yılında çıkardığı “Deterministic Non-Periodic Flow” adlı makalesinde yayınlamıştır. Burada Kaos’u, “rastlantısal davranan, düzgün geometrik yapıya sahip düzen” olarak tanımlamıştır (Lorenz, 1963; Ertokatlı, 2013).

Nonlinear ilişkilerde önce on iki, sonra sadece üç değişken kullanan Lorenz, bilgisayarında sürekli olarak değişen ama aynı koşulları tam anlamıyla iki kez asla tekrarlamayan koşullardan oluşan bir sürekli dizi üretmeyi başarmıştı. Nispeten basit matematiksel kuralları kullanarak “kaos” yaratmıştı. Lorenz’in bilgisayarı, onun seçtiği herhangi bir parametreyle başlayarak, aynı hesapları defalarca ama asla aynı sonucu vermeksizin mekanik bir biçimde yinelenmiştir. Bu “aperiyodiklik” (yani, düzenli döngülerin olmayışı) bütün kaotik sistemlerin özelliğidir. Aynı zamanda Lorenz, elde ettiği sonuçlar her defasında farklı olmasına rağmen, en azından sık sık ortaya çıkan “desen” izlerinin varolduğunu fark etti. Diğer bilimciler de, elektronik osilatörün (titreştirici) matematiksel modellenişinden gezegen yörüngelerinin incelenişine kadar pek çok farklı kaotik sistemde benzer “desenler” buldular. Gleick, bu ve diğer durumlarda “gelişigüzel görünen davranışın içinde yapı izlerinin” bulunduğunu kaydeder. Kaotik sistemlerin mutlaka kararsız olması gerekmediği ya da belirsiz bir

dönem boyunca sürebileceği düşüncesi giderek daha da belirginleşti. (Yazaroğlu ve Pehlivan, 2010). Lorenz'in önerdiği otonom doğrusal olmayan birinci dereceden adi diferansiyel denklem sistemi, E.2.1 denkleminde verilmiştir (Lorenze,1963).

$$\begin{aligned}\dot{x} &= s \cdot (y - x) \\ \dot{y} &= -x \cdot z + r \cdot x - y \\ \dot{z} &= x \cdot y - b \cdot z\end{aligned}\tag{E.2.1}$$

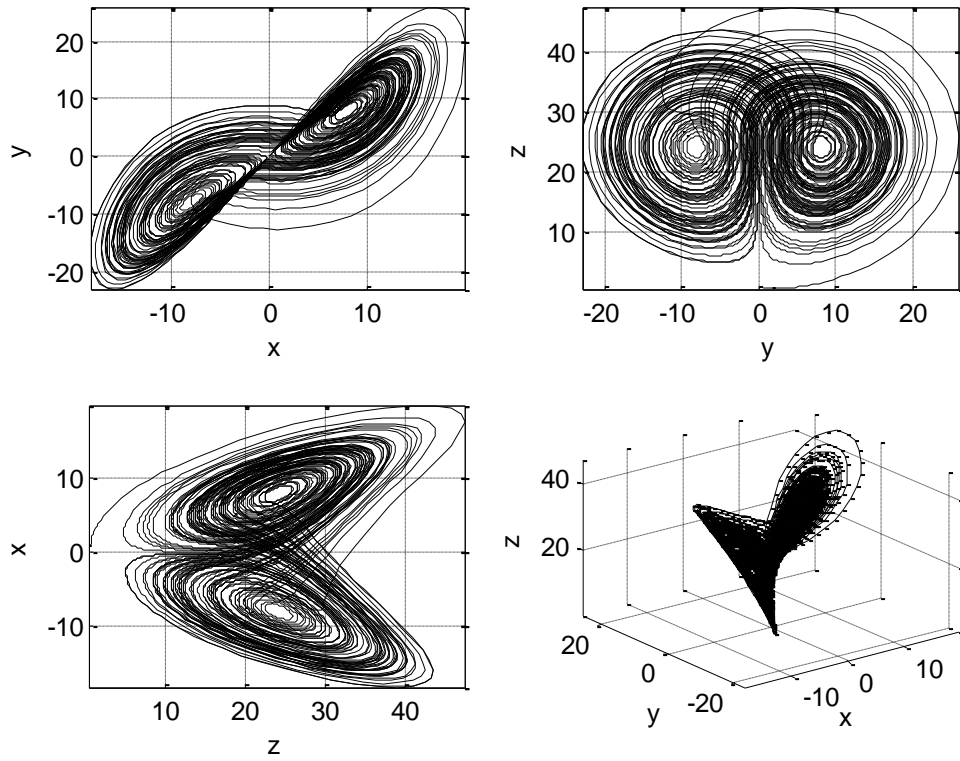
Bu denklem sisteminde s, r ve b sistem parametreleridir (İskurt, 2012). Sistemin karakteristik özelliği, spektrumu geniş bir frekans bölgesine yayılmış periyodik olmayan salınımlar üretmesidir. Bu salınımlar gürültüye benzediği ve tahmini zor bir şekilde başlangıç koşullarına bağlı oldukları için gizli haberleşmede kullanılabilmesi fark edilmiş ve Lorenz kaotik osilatörü kullanan kripto sistemleri geliştirilmiştir. E.2.2 denkleminde çözümde kullanılan 4. derece Runge-Kutta yöntemine ait denklemler verilmiştir.

$$\begin{aligned}\dot{y} &= f(x, y), \quad y(x_0) = y_0 \\ k_1 &= hf(x_n, y_n) \\ k_2 &= hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1\right) \\ k_3 &= hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2\right) \\ k_4 &= hf(x_n + h, y_n + k_3) \\ y_{n+1} &= y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)\end{aligned}\tag{E.2.2}$$

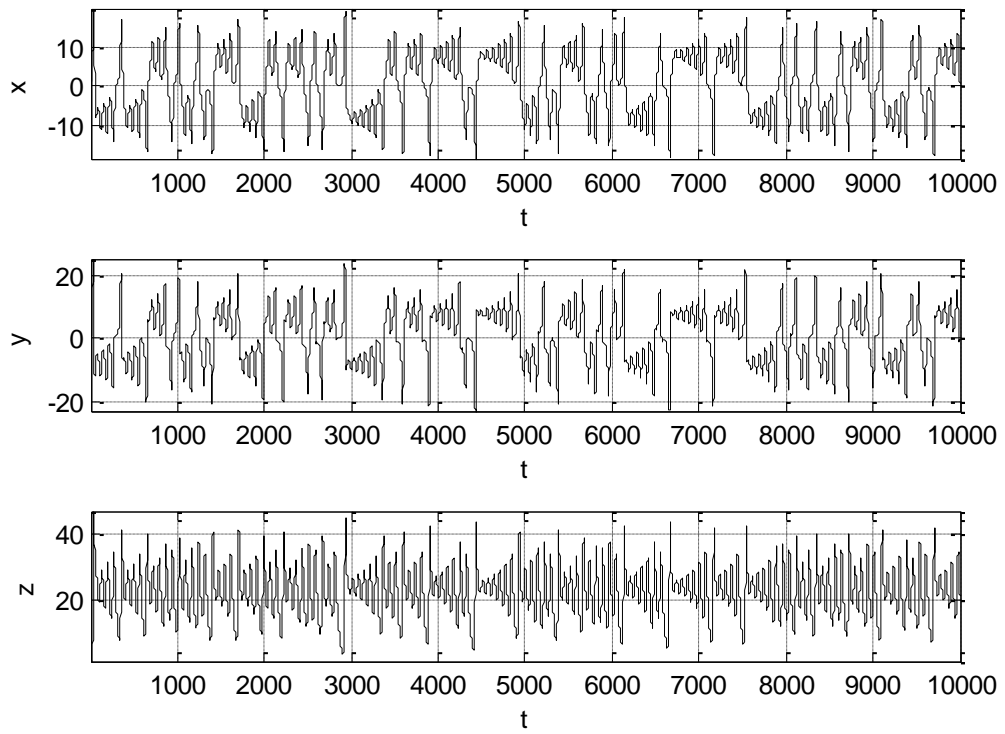
E.2.3'de çözümde kullanılan Euler metoduna ait denklem verilmiştir.

$$\begin{aligned}\dot{y} &= f(x, y), \quad y(x_0) = y_0 \\ y_{n+1} &= y_n + h \cdot f(x_n, y_n)\end{aligned}\tag{E.2.3}$$

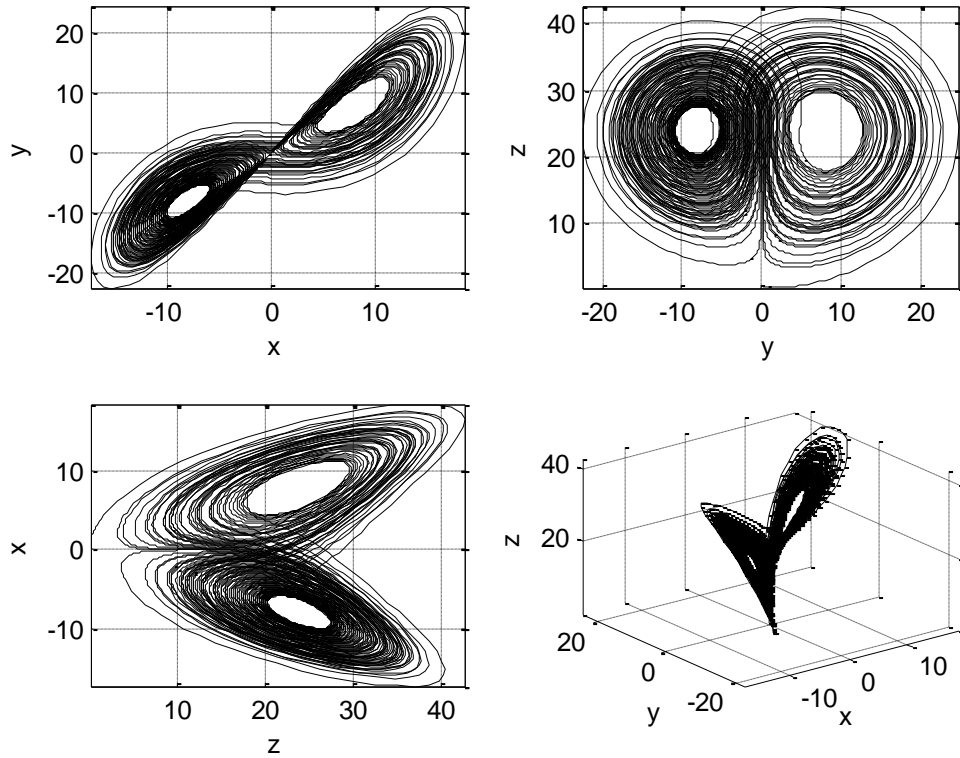
Lorenz kaotik sisteminin Euler yöntemi ile çözümlenerek bulunan çıkış değerleri Şekil 2.1'de fazör diyagramı ve üç eksen görünümü, Şekil 2.2'de durum değişkenlerinin zamana bağlı değişimleri gösterilmiştir. Şekil 2.3'de Lorenz sisteminin Runge-Kutta (4.derece) yöntemiyle bulunan çözümlerinin fazör diyagramı ve üç eksen görünümü, Şekil 2.4'de durum değişkenlerinin zamana bağlı değişimleri verilmiştir.



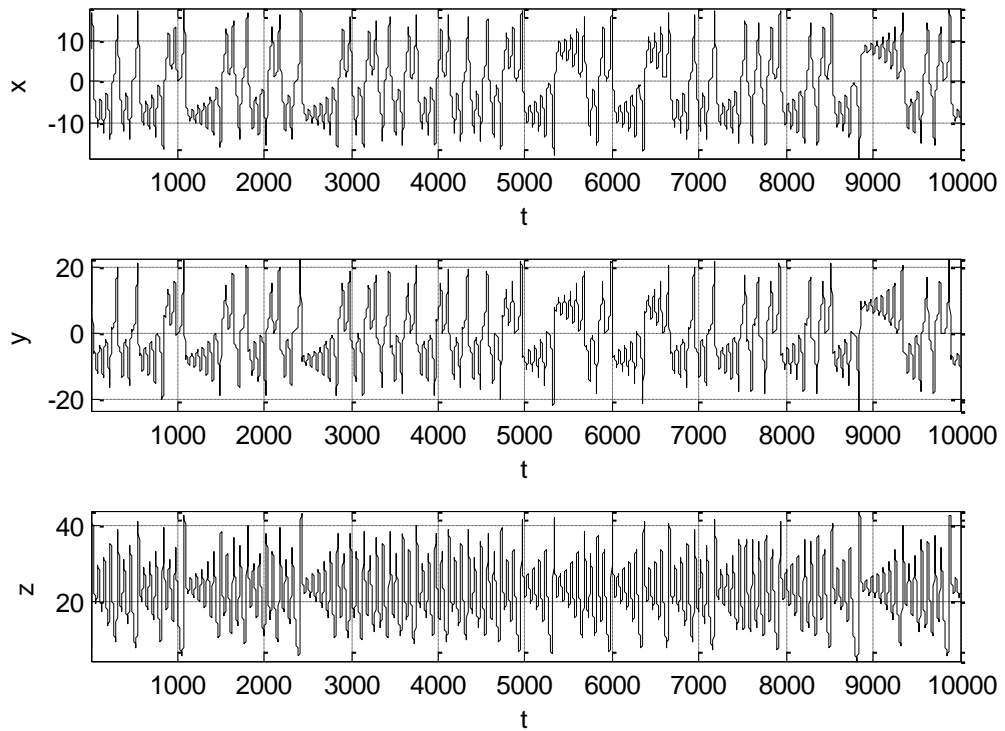
Şekil 2.1. Euler metodu ile çözümlenmiş Lorenz Çekicisine ait çıkışların fazör diyagramı ve üç eksen görünümü ($x_0 = 0.1$, $y_0 = 0.1$, $z_0 = 0.1$)



Şekil 2.2. Euler metodu ile çözümlenmiş Lorenz Çekicisinin x, y, z durum değişkenlerinin zamana göre değişimi ($x_0 = 0.1$, $y_0 = 0.1$, $z_0 = 0.1$)



Şekil 2.3. Runge-Kutta metodu ile çözümlenmiş Lorenz Çekicisine ait çıkışların fazör diyagramları ve üç eksen görünümü ($x_0 = 0.1$, $y_0 = 0.1$, $z_0 = 0.1$)



Şekil 2.4. Runge kutta metodu ile çözümlenmiş Lorenz Çekicisinin x,y,z durum değişkenlerinin zamana göre değişimi ($x_0 = 0.1$, $y_0 = 0.1$, $z_0 = 0.1$)

2.3.2 Chua sistemi

Elektronik devre alanında kaos durumu ilk kez Leon Chua tarafından incelenmiştir. Chua konuyu en basit şekliyle ele almak amacıyla Chua devresini tasarlamıştır. Chua devresi, kaos ve çataallaşmanın gösterilmesi için basit ve tipik üçüncü mertebeden otonom bir devredir. Son yıllarda, bir çok Chua devresi gerçekleştirilmiştir. Chua devresi ile özellikle senkronizasyonu kullanılarak güvenli haberleşme alanında ve pratik kaotik sistemleri açıklamak gibi çok önemli çalışmalar mevcuttur. Bu dinamik denklemlerle elde edilen sonuçlar güvenli ve gizli haberleşme için gerekli olan kaotik taşıyıcı özelliğini sağlarlar (İskurt, 2012).

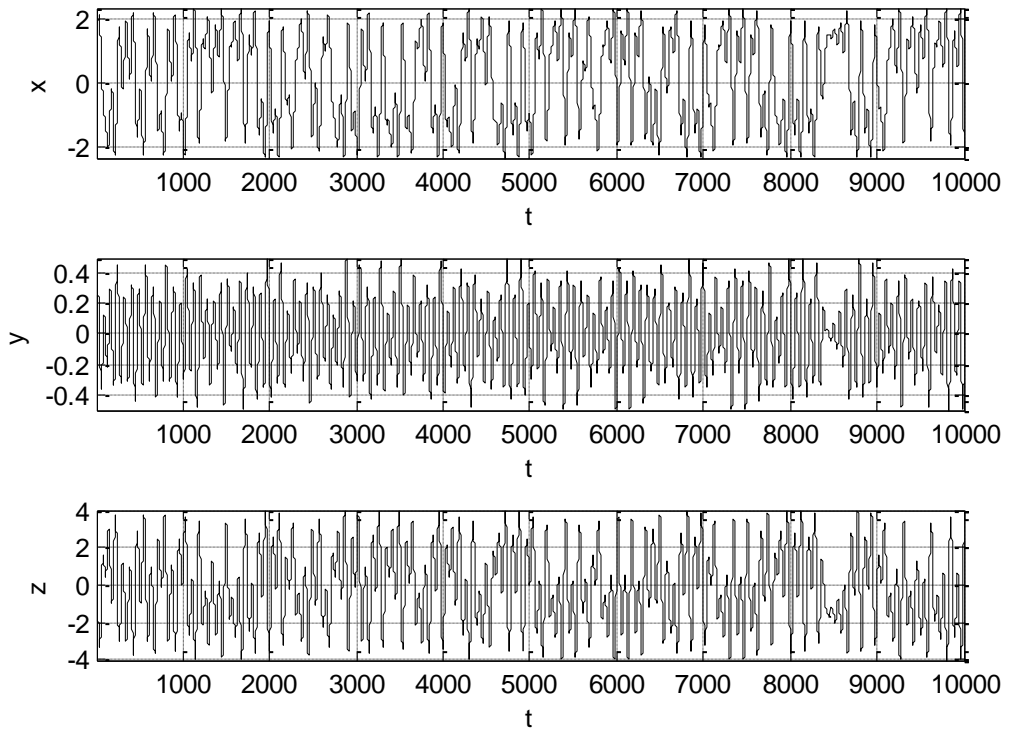
Chua devresi, en karmaşık kaosun varlığının deneysel olarak kurulabildiği, sayısal olarak doğrulanabildiği ve matematiksel olarak kanıtlanabildiği en basit devrelerden biridir. Chua elektronik devresinin sistem yapısı basittir. Dört doğrusal eleman ve bir doğrusal olmayan eleman olan Chua diyodundan oluşmaktadır. Chua diyonu farklı aktif devre yapılarıyla da oluşturulabilir (Pehlivan, 2007). Chua devre denklemleri, E.2.4 denklemdeki üç adi diferansiyel denklemle tanımlanır.

$$\begin{aligned}\frac{dx}{dt} &= \alpha[y - x - f(x)] \\ \frac{dy}{dt} &= x - y + z \\ \frac{dz}{dt} &= -\beta y\end{aligned}\tag{E.2.4}$$

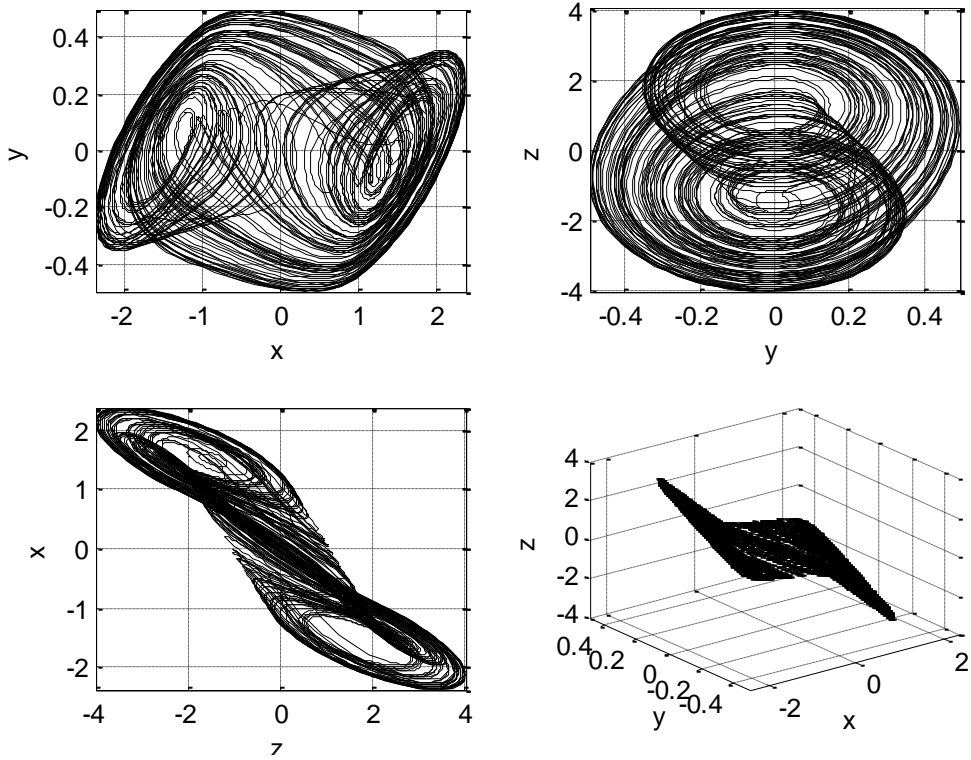
Burada, z değişkeni i_L akımını, x değişkeni V_{C_1} gerilimini, y değişkeni V_{C_2} gerilimini göstermektedir. Chua diodunun i_R akımı iki kırılma noktasına sahip kısmi doğrusal fonksiyon olan $f(x)$ ile ifade edilir. Analitik ifadesi ise aşağıdaki gibidir.

$$f(x) = m_0 \cdot V_{C_1} + \frac{1}{2} \cdot (m_1 - m_0)(|V_R + B_P| - |V_R - B_P|)\tag{E.2.5}$$

Chua kaotik sistemi, $\alpha = 15.6$, $\beta = 28$, $m_0 = -1.143$, $m_1 = -0.714$ değerleri alınarak, başlangıç değerleri $x_0 = 0.7$ ve $y_0 = z_0 = 0$ için Euler metodu ile çözülmüştür. Şekil 2.5’de Chua kaotik sisteminin Euler yöntemi ile çözülmüş durum değişkenlerinin zamana göre değişimleri gösterilmiştir. Chua kaotik sisteminin fazör diyagramları ve üç boyutlu görünümü ise Şekil 2.6’da verilmiştir.



Şekil 2.5. Euler metodu ile çözümlenmiş Chua devresinin durum değişkenlerinin zamana göre kaotik değişimleri ($\alpha=15.6$, $\beta=28$, $m_0=-1.143$, $m_1=-0.714$, $x_0 = 0.7$, $y_0 = 0$, $z_0 = 0$)



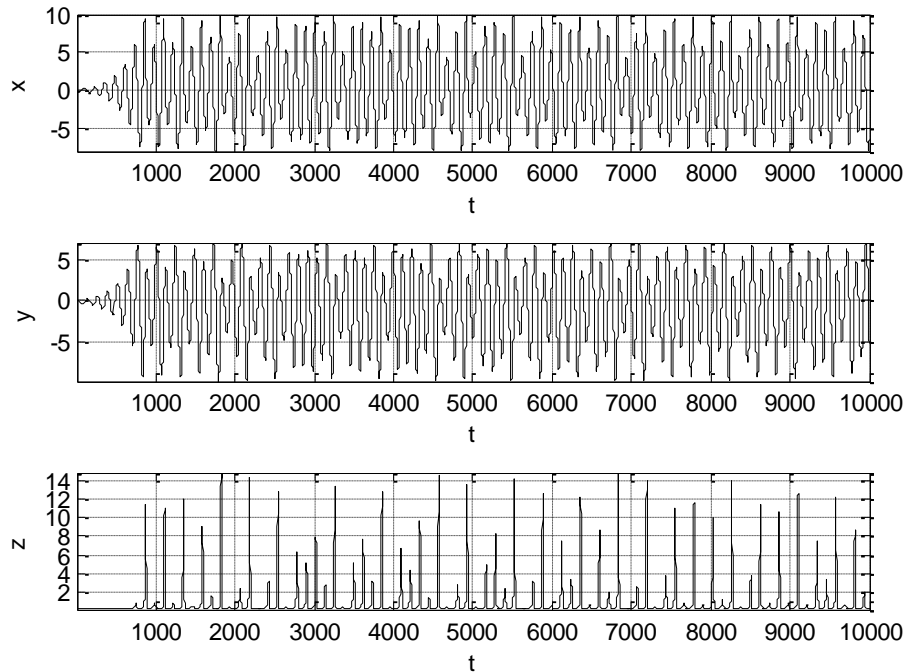
Şekil 2.6. Euler metodu ile çözümlenmiş Chua devresinin fazör diyagramları ve üç eksen görünümü ($\alpha=15.6$, $\beta=28$, $m_0=-1.143$, $m_1=-0.714$, $x_0 = 0.7$, $y_0 = 0$, $z_0 = 0$)

2.3.3 Rössler sistemi

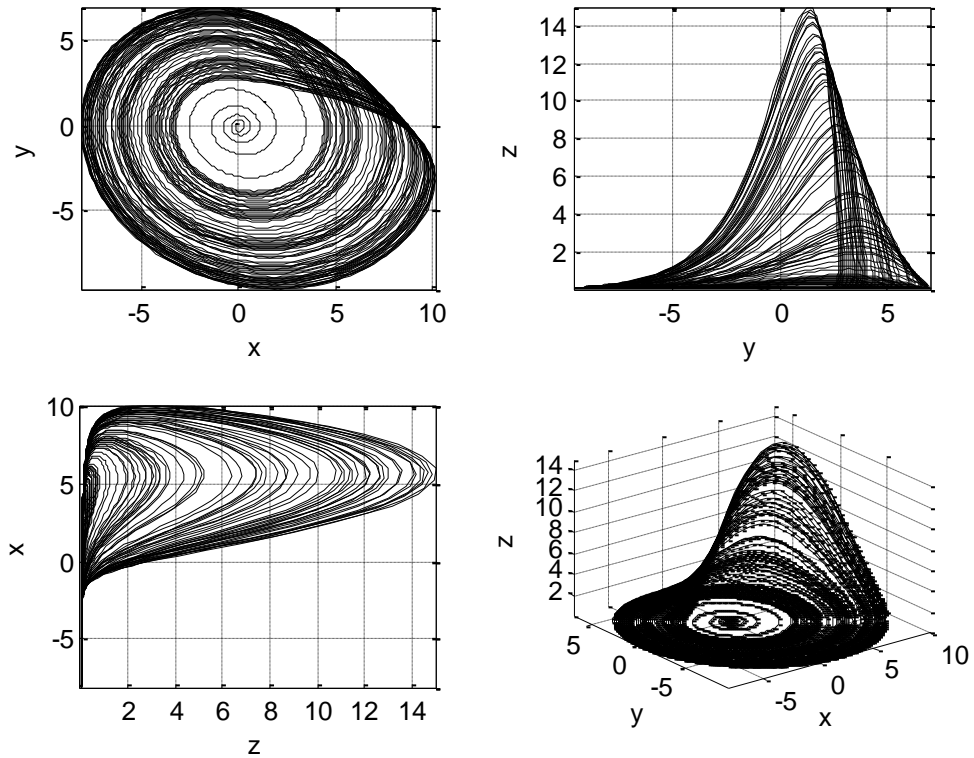
Rössler sistemi üç lineer olmayan diferansiyel denklemden oluşur. Bu diferansiyel denklemler, Rössler çekicisinin fraktal özellikleri ile ilişkilendirilmiş kaotik dinamikleri sergileyen sürekli zaman dinamik sistemlerini tanımlar. Rössler çekicisi, Lorenz çekicisine benzer şekilde davranır, ancak niteliksel olarak analiz etmek daha kolaydır. 1976 yılında, Rössler'in önerdiği (Rössler, 1976) kaotik denklem sistemi E.2.6'te verilmiştir

$$\begin{aligned}\dot{x} &= -y - z \\ \dot{y} &= x + a.y \\ \dot{z} &= b + z.(x - c)\end{aligned}\tag{E.2.6}$$

Sistem, bir adet ikinci dereceden doğrusal olmayan terim (xz) olmak üzere toplam yedi terim içermektedir. $a = 0.2$, $b = 0.2$ ve $c = 5$ parametreleri ve başlangıç şartları için kaotik çözümler elde edilmiştir. Rössler sisteminin $x_0 = 0.1$, $y_0 = 0.1$, $z_0 = 0.1$ başlangıç koşulları için Runge-Kutta yöntemi ile bulunan durum değişkenlerinin zamana bağlı değişimleri Şekil 2.7'de gösterilmiştir. Şekil 2.8'de Rössler sisteminin fazör diyagramları ve üç eksen görünümü verilmiştir.



Şekil 2.7. Runge-Kutta metodu ile çözümlenmiş Rössler sisteminin x, y ve z durum değişkenlerinin zamana göre değişimi ($x_0 = 0.1$, $y_0 = 0.1$, $z_0 = 0.1$)



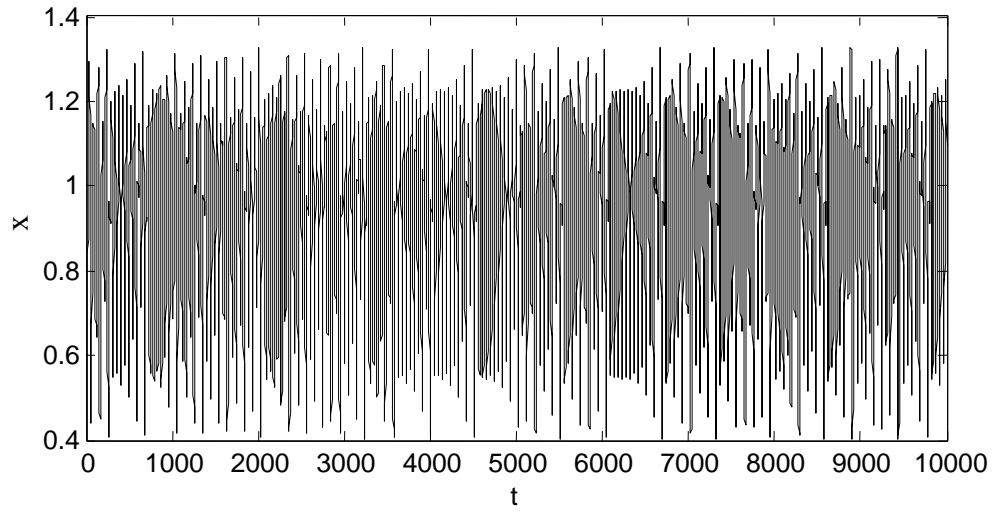
Şekil 2.8. Runge-Kutta metodu ile çözümlenmiş Rössler Çekicisine ait fazör diyagramları ve üç eksen görünümü ($x_0 = 0.1$, $y_0 = 0.1$, $z_0 = 0.1$)

2.3.4 Mackey-Glass sistemi

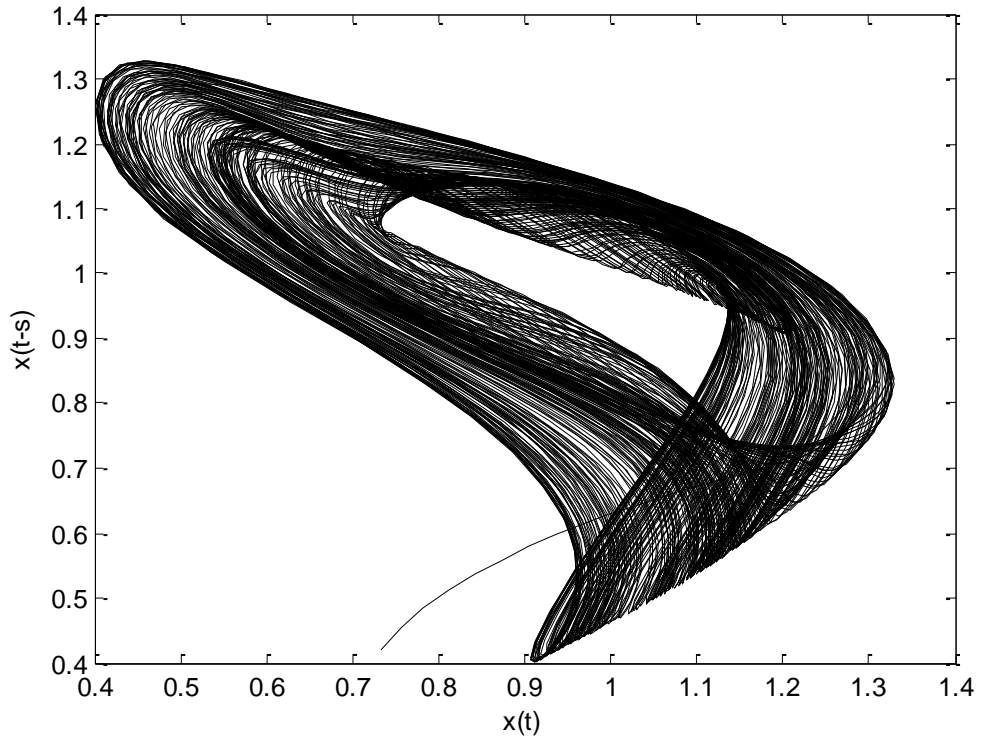
Psikolojik kontrol sistemlerinde kullanılmak üzere 1977 yılında Mackey ve Glass tarafından geliştirilen birinci dereceden lineer olmayan diferansiyel zaman gecikmeli eşitliktir (Mackey ve Glass, 1977). Bu eşitlikler, çeşitli dalga formlarında, sınırlı osilasyonlarda dinamik bir davranış göstermektedir. Mackey-Glass zaman serisine ait matematiksel ifade E.2.7'deki gibidir. Burada $\tau > 17$ seçildiğinde kaotik zaman serisi oluşmaktadır.

$$\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1+x^c(t-\tau)} - bx(t) \quad (\text{E.2.7})$$

$a = 0.2$, $b = 0.1$, $c = 10$ ve $\tau = 17$ parametreleri için kaotik çözümler elde edilmiştir. Euler yöntemi ile çözülen Mackey-Glass sistemine ait zaman serileri Şekil 2.9'da gösterilmiştir. Yine Mackey-Glass sistemine ait $x(t)$ ve $x(t-s)$ grafiği Şekil 2.10'da verilmiştir.



Şekil 2.9. Euler metodu ile çözümlenmiş Mackey-Glass sisteminin durum değişkeninin zamana göre değişimi ($a = 0.2, b = 0.1, c = 10, x_0 = 0.1$)



Şekil 2.10. Euler metodu ile çözümlenmiş Mackey-Glass Çekicisine ait çıkışlar ($a = 0.2, b = 0.1, c = 10, x_0 = 0.1$)

3. KAOTİK TABANLI FARKSAL GELİŞİM ALGORİTMASI (KFGA)

3.1 Farksal Gelişim Algoritması (FGA)

Evrimsel prensibine dayalı teknikler bilinen optimizasyon metotlarının noksanlıklarının üstesinden gelmek için kullanılmaktadır (Storn ve Price,1997). Son zamanlarda bu prensibe dayalı tekniklerin hepsini temsilen ortak bir terim olarak evrimsel hesaplama terimi yaygın olarak kullanılmaya başlanmıştır. Bu sınıfa giren algoritmalara örnek olarak genetik algoritmalar, evrimsel programlama (evolutionary programming) evrimsel stratejiler (evolution strategies) ve Farksal Gelişim Algoritması (FGA) vs sayılabilir (Storn ve Price, 1997). Bir problemi çözmeye kullanılacak herhangi bir evrimsel algoritma aşağıdaki beş elemana ihtiyaç duymaktadır.

- Problem için aday çözümlerin temsili,
- Çözümlerin başlangıç popülasyonunu oluşturacak bir yöntem,
- Çözümleri uygunluk açısından değerlendirmeye tabii tutacak değerlendirme fonksiyonu,
- Sonraki iterasyonlar için operatörler,
- Kontrol parametrelerinin değerleri.

Evrimsel algoritma tek bir bireyle değil bireylerin popülasyonu ile ilgilenir. Her birey mevcut problem için muhtemel bir çözümü temsil eder ve bir veri yapısı olarak tanımlanır. Her birey uygunluk ölçüt değerini belirlemek amacıyla değerlendirilir ve daha uygun bireyleri seçmek suretiyle her iterasyon sonunda yeni popülasyon oluşturulur (Koyuncu. 2006).

Evrimsel hesaplama teknikleri arasında oldukça benzerlikler olmasına rağmen birçok farklılıklar da mevcuttur. Örneğin çözümlerin temsili için farklı veri yapıları kullanılır ve dolayısıyla kullanılan evrimsel operatörler de farklı olabilir. Örneğin FGA'da bireylerin seçimi için kullanılacak çok sayıda seçme metodu bulunmaktadır (Karaboğa, 2004; Arslan, 2010).

FGA basit ama güçlü popülasyon tabanlı bir algoritmadır (Price,1996). Özellikle tamamen düzenlenmiş uzayda tanımlı ve gerçek değerli tasarım parametrelerini içeren fonksiyonları küresel olarak optimize etmek amacıyla kullanılan bir direkt arama

algoritmasıdır. Gerçek parametrelili optimizasyon problemleri bilimde, mühendislikte ve iş ortamlarında karşılaşılan pratik problemlerin önemli ve geniş bir sınıfını oluşturmaktadır. Zor optimizasyon problemleriyle karşılaşıldığında genellikle ilk yapılması gereken probleme özel sezgisel bir yaklaşım tekniği belirlemek olacaktır. Uzman bilgisinin tasarıma ilave edilmesine olanak sağlayan böyle teknikler gelişime dayalı algoritmalar gibi genel metotlarla karşılaştırıldığında oldukça üstün başarımlar göstermektedir. Örneğin amaç fonksiyonu lineer polinom formunda ise bu tür problemi çözmek amacıyla geliştirilmiş Simpleks metodu, Karmakar algoritması gibi başarılı yöntemler mevcuttur. Bu yöntemler bu tip problemlerde yüzlerce hatta binlerce tasarım değişkeninin bulunması durumunda bile küresel optimal çözümlerini bulabilme kabiliyetine sahiptir. Amaç fonksiyonu lineer bir polinom şeklinde değilse ve problem için özel metodlar yoksa o zaman genellikle Levenberg-Marquardt algoritması Branch and Bound metodu gibi yöntemlere başvurulmaktadır (Karaboğa, 2004; Arslan, 2010).

Maalesef, doğrusal olmama, gürültü, yüksek boyutluluk, çok modluluk, sınırlamalar, değişmeme (düz olma) ve türev alınamama gibi özellikler tek başlarına veya birlikte bu tür deterministik metotların başarımlarını olumsuz etkilemektedirler. Stokastik algoritmalar gibi gelişime dayalı algoritmalar, sınırlamaların çoğunun üstesinden gelebilmektedirler. Optimizasyon işleminin tamamen düzenlenmiş uzay domeninde sınırlandırılma bilgisi, kendi başına gelişime dayalı algoritmaların başarımlarını artırmak için kullanılabilecek probleme özel önemli bir bilgidir. Tipik bir gelişime dayalı algoritmanın temel basamakları aşağıdaki gibidir (Karaboğa, 2004; Arslan, 2010).

Adım 1. Başlangıç popülasyonunun oluşturulması

Adım 2. Maliyet Fonksiyon Hesabı

Adım 3. TEKRARLA

Adım 3.1. Evrimsel Operatörler

Adım 3.2. Maliyet Fonksiyon Hesabı

Adım 3.3. Yeni Popülasyon Oluşturma

DURDURMA KRİTERİ SAĞLANINCAYA KADAR

Gelişime dayalı algoritmaların genel amaçlı sayısal optimizasyon algoritmaları olarak incelenmesi ve değerlendirilmesi ile ilgili çalışmalara literatürde oldukça sık rastlanmaktadır. Ancak bu tartışmaların çoğu, orijinalinde bu amaç için tasarlanmamış tümleşik (combinatorial) tip algoritmalarla elde edilen sonuçlar üzerinedir. FGA ise bir ayrık optimizasyon algoritması değil özellikle sayısal optimizasyon için geliştirilmiş bir gelişim algoritmasıdır. Daha önce tanımlanmış, olasılık dağılım fonksiyonuna dayalı olarak çalışan genetik algoritma gibi gelişim tabanlı algoritmaların tersine FGA yapısı rastgele olarak seçilmiş amaç vektör çiftlerinin farklarına dayalı bir mutasyon işlemi kullanır (Karaboğa, 2004; Arslan, 2010).

Şekil 3.1'de FGA'ya ait kaba (pseudo) kod verilmektedir. FGA'ya ait bu kaba kod, temel olarak, popülasyon içerisinde rastgele seçilen iki bireyin ağırlık farkının üçüncü bir bireye eklenmesi mantığına dayanmaktadır.

Input: Popülasyon boyu, Problem boyutu, Sınır değerler,
Çaprazlama sabiti (CR), Skala faktörü (F)
Output: En iyi maliyet fonksiyonu, En iyi bireyler
Popülasyon ilk değer atama (Popülasyon boyu, Problem boyutu, Sınır değerler)
Maliyet hesapla(Popülasyon)
En iyi birey ← Popülasyon içindeki en iyi maliyete sahip birey
En iyi maliyet değeri ← Maliyet hesapla(En iyi birey)
While(Durdurma kriteri)
 Yeni popülasyon sıfırla
 Mutasyon bireyleri hesapla (Skala faktörü, FGA stratejisi)
 Çaprazlama işlemi (Çaprazlama sabiti)
 Maliyet hesapla(Yeni popülasyon)
 If (Maliyet(Yeni birey) ≤ Maliyet(Eski birey))
 Yeni Popülasyon ← Yeni birey
 Else
 Yeni Popülasyon ← Eski birey
 End
 Popülasyon ← Yeni Popülasyon
En iyi birey ← Popülasyon içindeki en iyi maliyete sahip birey
En iyi maliyet değeri ← Maliyet hesapla(En iyi birey)
End
Return(En iyi maliyet fonksiyonu, En iyi bireyler)

Şekil 3.1. FGA Kaba Kodu

FGA'nın farklı stratejileri mutasyon operatörü içerisinde kullanılmaktadır. Aşağıda bu stratejilerden en sık kullanılanlardan bazıları verilmiştir:

1. en iyi birey / 1 fark vektörü / üstel çaprazlama
2. rastgele bireyler / 1 fark vektörü / üstel çaprazlama
3. rastgele ve en iyi bireyler / 1 fark vektörü / üstel çaprazlama
4. en iyi birey / 2 fark vektörü / üstel çaprazlama
5. rastgele bireyler / 2 fark vektörü / üstel çaprazlama
6. en iyi birey / 1 fark vektörü / binom çaprazlama
7. rastgele bireyler / 1 fark vektörü / binom çaprazlama
8. rastgele ve en iyi bireyler / 1 fark vektörü / binom çaprazlama
9. en iyi birey / 2 fark vektörü / binom çaprazlama
10. rastgele bireyler / 2 fark vektörü / binom çaprazlama

Bu genel gösterim tarzında, ilk kısım popülasyon içinden seçilen bireyleri, ikinci kısım seçilen bireylerin oluşturacağı fark vektörünün sayısını, son kısım ise çaprazlama tipini belirtmektedir (Yüzgeç, 2011).

3.1.1 Başlangıç toplumunun oluşturulması

Çözüme başlamadan önce, tasarım değişkenlerinin alabilecekleri minimum ve maksimum değerleri tanımlanır ve başlangıç popülasyonu E.3.1 eşitliğine göre rastgele türetilir (Storn ve Price, 1997; Karaboğa, 2004; Gürarlan, 2011).

$$x_{i,j}^0 = x_{min,j} + rand().(x_{max,j} - x_{min,j}) \quad (E.3.1)$$

$$i = 1, \dots, NP \text{ ve } j = 1, \dots, D$$

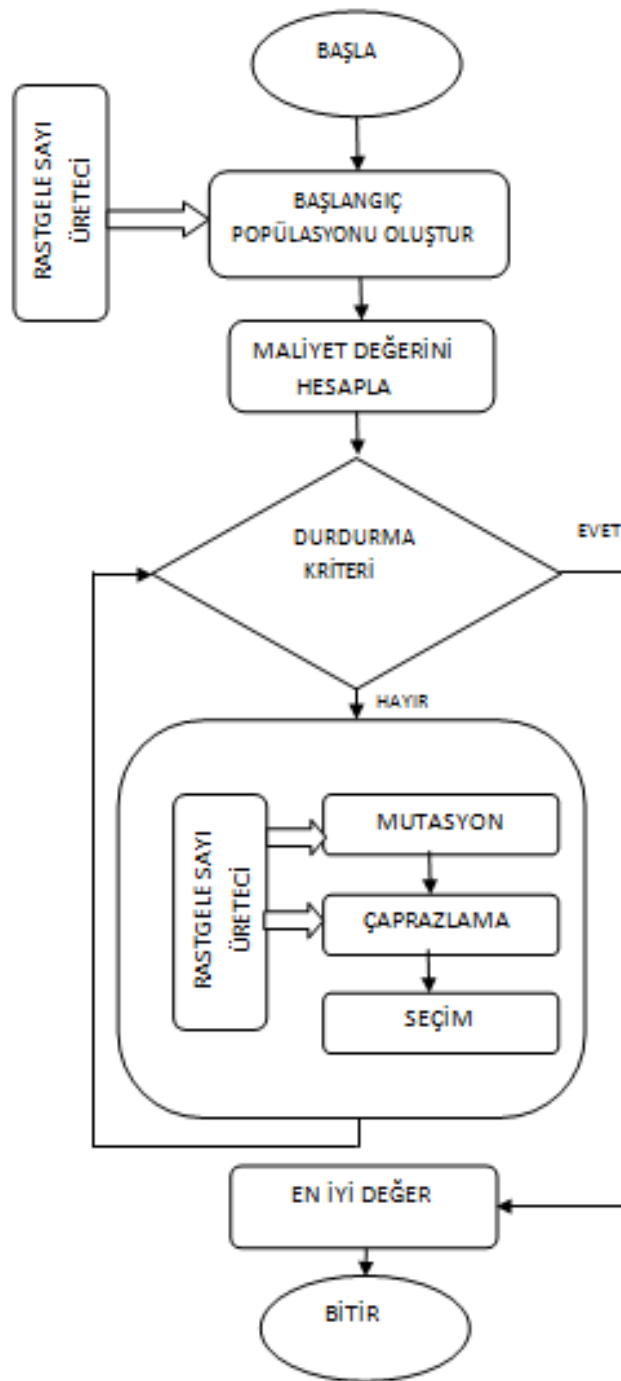
Burada NP popülasyondaki birey sayısını, D değişken sayısını, $x_{i,j}^0$ başlangıç bireylerini, $x_{min,j}$ ve $x_{max,j}$ tasarım değişkenlerinin alabileceği alt ve üst sınır değerlerini göstermektedir. Toplumdaki birey sayısının seçimi incelenen probleme göre değişmektedir. Literatürde NP =10 x D alınması tavsiye edilmektedir (Storn ve Price, 1997).

FGA'da popülasyon büyüklüğü (NP) algoritmanın performansı açısından önemlidir. Yeni bireylerin oluşturulmasını sağlayacak mevcut birey dışında üç adet bireye daha ihtiyaç vardır. Popülasyon büyüklük değerinin üçten büyük olması gerekir (Hiçdurmaz, 2013). Dolayısıyla popülasyon büyüklüğü $NP \geq 4$ olmalıdır. Popülasyon değerinin aşırı küçük seçilmesi, yetersiz örneklemeye sebep olacağından yakınsamayı sağlamakta zorlaşacak ve araştırma belirli bir alt en iyi noktaya doğru sürüklenecektir. Popülasyon değerinin aşırı büyük seçilmesi, bir iterasyon boyunca bireylerin gelişimi için oldukça uzun zaman gerektirecektir. Bu durum özellikle gerçek zamanlı problemlerin çözümünde istenmeyen bir durumdur. Bu yüzden uygun bir popülasyon büyüklüğü belirlenmelidir.

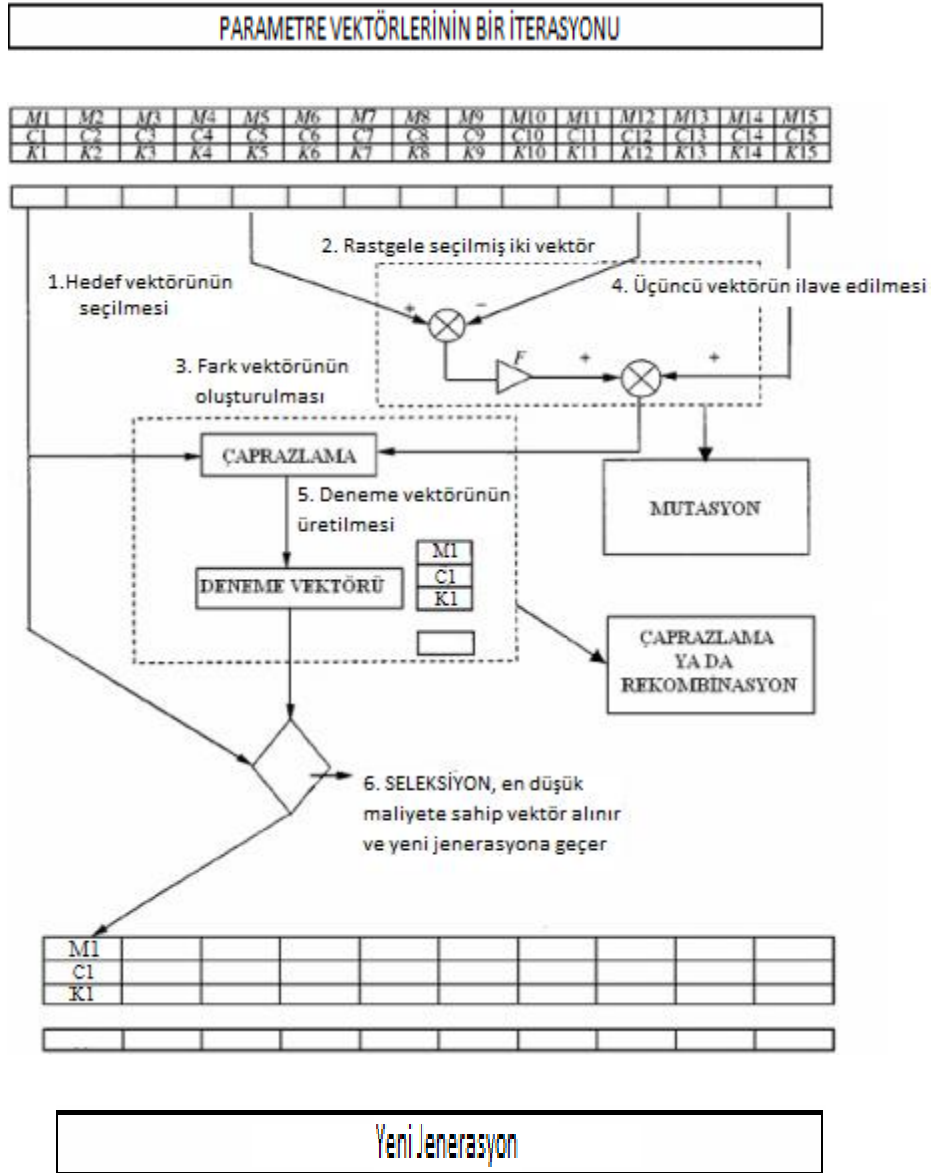
Gürarslan (2011) çalışmasına göre NP değerinin sadece değişken sayısına bağlı olmadığı, aynı zamanda problemin çözüm uzayındaki yerel optimum nokta sayısı ile de bağlantılı olduğu öne sürülmüştür.

Şekil 3.2'de FGA'nın akış diyagramı gösterilmiştir. Bu akış diyagramında durdurma kriteri sağlanana kadar sırasıyla mutasyon, çaprazlama ve seçim işlemleri yapılarak her iterasyonda çözüm için yeni bireyler elde edilir, durdurma kriteri sağlandığında ise bulunan en iyi çözüm değeri çıkışa aktarılır ve program sona erdirilir. Durdurma kriteri, genelde kullanılan maksimum iterasyon sayısından küçük olma şartıdır. Buna ilave kriterler de kullanılabilir.

Şekil 3.3'de ise FGA'nın şematik diyagramı gösterilmiştir. Şekilden görüldüğü üzere önce rastgele bir hedef vektör seçilir, rastgele seçilmiş iki vektörün farkı alınır ve üçüncü bir vektör ilave edilir (mutasyon), mutasyon sonucu elde edilen vektör ile hedef vektör çaprazlanarak deneme vektörü elde edilir, elde edilen deneme vektörü ile hedef vektör karşılaştırılarak en iyi maliyete sahip vektör yeni jenerasyon için seçilir.



Şekil 3.2. FGA'nın akış diyagramı



Şekil 3.3. FGA'nın şematik diyagramı (Yiğit, 2007).

3.1.2 Mutasyon operatörü

Parametre optimizasyonu açısından mutasyon işlemi, var olan bir amaç vektörünün bir veya daha fazla parametreleri üzerine, rastgele üretilmiş bir artışın ilave edilme işlemine karşılık gelmektedir. İyi bir mutasyon işleminin amacı, var olan amaç vektörlerini doğru zamanda doğru miktar ile doğru yönde hareket ettirecek artışları başka bir ifadeyle adımları üretmektir. Bu hedefi başarmak esas olarak adımları üretecek dağılımın karakteristiklerine bağlıdır.

FGA yapısı, bir amaç vektörüne mutasyon işlemini, rastgele seçilmiş amaç vektörler çiftinin ağırlıklaştırılmış farkının bu amaç vektörüne ilave edilmesiyle gerçekleştirir. Algoritmanın basit mutasyon işlemi sık sık aritmetik çaprazlama işlemiyle karıştırılır.

Aritmetik çaprazlama, iki vektörün genel doğrusal kombinasyonu iken mutasyon üç vektörün doğrusal kombinasyonunun özel bir halidir. Her iki durumda da mevcut amaç vektörlerinin doğrusal kombinasyonu olan yeni vektörler üretilmektedir. Kısacası FGA yapısı, gerçek parametre optimizasyonuna rastgele amaç vektörlerinin doğrusal kombinasyonlarını örnekleyen, mutasyon ve yeniden birleştirme gibi iki basit operasyon kullanan normal gelişime dayalı bir yaklaşımdır. Aritmetik çaprazlama ile mutasyon birbirlerine benzemelerine rağmen amaç vektörleri uzayını oldukça farklı şekilde araştırırlar.

Örnek olarak en sık kullanılan beş FGA stratejisi için mutasyona uğramış bireyler ($v_{i,g}$) aşağıdaki şekilde bulunur:

$$v_{i,g+1} = x_{i,g} + F(x_{b,g} - x_{i,g}) + F(x_{r_1,g} - x_{r_2,g}) \quad (E3.2)$$

$$v_{i,g+1} = x_{r_3,g} + F(x_{r_1,g} - x_{r_2,g}) \quad (E3.3)$$

$$v_{i,g+1} = x_{b,g} + F(x_{r_1,g} - x_{r_2,g}) \quad (E3.4)$$

$$v_{i,g+1} = x_{b,g} + F(x_{r_1,g} - x_{r_2,g}) + F(x_{r_3,g} - x_{r_4,g}) \quad (E3.5)$$

$$v_{i,g+1} = x_{r_1} + F(x_{r_2,g} - x_{r_3,g}) + F(x_{r_4,g} - x_{r_5,g}) \quad (E3.6)$$

Bu denklemlerde $r_1, r_2, r_3, r_4, r_5 \in [1, NP]$ rastgele katsayılardır. $x_{b,g}$ popülasyon içerisindeki en iyi maliyet fonksiyonuna sahip bireyi göstermektedir. Rastgele katsayılar birbirinden farklı olmak zorundadır. Skala faktörü (F) sabit bir değerdir ve $0 \leq F \leq 2$ olarak değişir. Skala faktörü rastgele seçilen bireylerin fark vektörlerinin yükseltilmesinin kontrolünde kullanılır (Yüzgeç, 2011).

3.1.3 Çaprazlama operatörü

Yeni bireyler çaprazlama operatörü ile aşağıdaki şekilde bulunabilir:

$$u_{i,g+1} = \begin{cases} v_{i,g+1}, & \text{if } r \leq CR \\ x_{i,g}, & \text{if } r > CR \end{cases} \quad (E3.7)$$

Bu denklemde $r \in [0, 1]$ rastgele sayı üreticini ve $u_{i,g}$ deneme bireylerini göstermektedir. Çaprazlama olasılık sabiti (CR) $\in [0, 1]$ genellikle kullanıcı tarafından belirlenir. Eğer $CR = 1$ ise, oluşacak her yeni birey mutant olarak belirlenir, yani rastgele seçilen fark vektörüne göre seçilir. Diğer taraftan $CR = 0$ durumunda, tüm bireyler önceki jenerasyondan gelir. Diğer bir deyişle, popülasyonda değişim olmaz, yeni bireyler oluşmaz. Rastgele sayı üretici, mutant bireylerden ($v_{i,g+1}$) en az birinden bir sonraki jenerasyona birey ($u_{i,g+1}$) üretilmesini garanti eder (Yüzgeç, 2011).

3.1.4 Seçim (Seleksiyon) işlemi

Bir sonraki iterasyondaki bireylerin $x_{i,g}$ tespitinde seçim işleminin işletilmesi sırasında, deneme bireyinin ($u_{i,g+1}$) maliyet fonksiyonu değeri hedef bireyin ($x_{i,g}$) maliyet fonksiyonu ile karşılaştırılır ve sonraki iterasyona, optimizasyon problemine göre yeni birey seçilir. Aşağıdaki eşitlikte maksimum problemi için seçim operatörünün denklemi verilmiştir:

$$x_{i,g+1} = \begin{cases} u_{i,g+1}, & \text{if } f(u_{i,g+1}) < f(x_{i,g}) \\ x_{i,g}, & \text{yoksa} \end{cases} \quad (E3.8)$$

Burada f maliyet fonksiyonunu göstermektedir. Eğer deneme bireyi ($u_{i,g+1}$) hedef bireyden ($x_{i,g}$) daha iyi bir maliyet fonksiyonuna sahipse, bir sonraki iterasyonda kullanılacak birey ($x_{i,g+1}$) deneme bireyi ($u_{i,g+1}$) olarak seçilir. Aksi durumda popülasyondaki hedef bireyde değişiklik olmaz (Yüzgeç, 2011).

Gelişim stratejisi, alternatif olarak birleştirilmiş ebeveyn-çocuk popülasyonunun en iyi başarımları gösteren üyelerinin gelişimine müsaade edildiği deterministik bir seçme işlemi kullanılmaktadır. Her iki seçme işleminin dezavantajı, her jenerasyon sonunda

popülasyonun kalite değerlerine göre sıraya dizilme maliyetidir. FGA'nın seçme işleminde, yeni üretilen vektör ebeveynine göre daha gelişmiş veya en azından aynı gelişme seviyesinde değilse ebeveyn vektör en az bir jenerasyon daha popülasyonda kalmaya devam etmekte ve başka bir vektörle yer değiştirmemektedir (Yiğit, 2007).

3.2 Kaotik Tabanlı Farksal Gelişim Algoritması (KFGA)

Klasik FGA'da bireylerin seçimi için rastgele seçim metodu kullanılmaktadır. Rastgele çalıştırıldığında üretilen değerlerin aynı değerler veya benzer değerler olma olasılığı vardır. Üretilen değerlerin aynı değerler veya benzer değerler olması çeşitliliği azaltabileceğinden dolayı istenmeyen bir durumdur. Kaotik tabanlı FGA'da ise rastgele seçim metodu yerine kaotik denklemlerin çözümleri kullanılarak, birbirinden farklı bireylerin seçimi için sayılar üretilmektedir. Kaotik tabanlı FGA'da her tekrarda farklı kaotiklik elde etmek ve böylece birbirinden farklı sayı dizileri oluşturmak için sayı üreticinde bir başlangıç tarihi belirlenip kodun çalıştırıldığı saat ile farkı saniye cinsinden alınarak bulunan değer, kaotik sistemin başlangıç değerleri (x_0, y_0, z_0) olarak kullanılır. Şekil 3.4' de kaotiklik için sayı üreticine ait kaba kod bulunmaktadır.

Başlangıç tarihi belirle

Şimdiki saati bul

Başlangıç tarihi ile şimdiki tarihin farkını saniye cinsinden bul

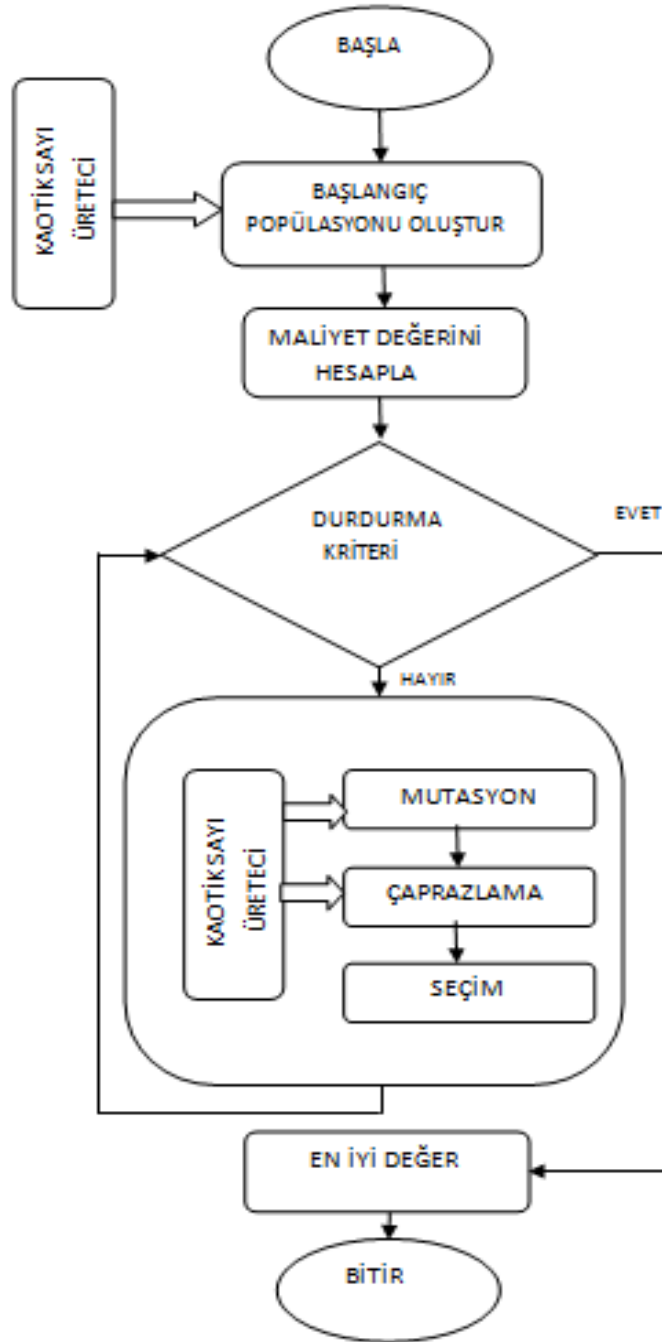
Bulunan değeri kaotik başlangıç değeri olarak al

Şekil 3.4. Kaotiklik için sayı üreticine ait kaba kod

Yapılan bu çalışmada en sık kullanılan kaotik denklemlerden Lorenz kaotik sistemi ve Rössler kaotik sistemi kullanılmıştır.

Lorenz kaotik sistemi ve Rössler kaotik sistemi denklem çözümünde Euler metodu yerine Runge-Kutta metodu (4. derece) kullanılmıştır.

Şekil 3.5'de KFGA'nın akış diyagramı verilmiştir.



Şekil 3.5. KFGA'nın akış diyagramı.

Şekil 3.6'da KFGA algoritmasına ait kaba kod verilmektedir. KFGA'ya ait bu kaba kod, temel olarak, popülasyon içerisinde kaotik denkleme göre seçilen iki bireyin ağırlık farkının üçüncü bir bireye eklenmesi mantığına dayanmaktadır.

Input: Popülasyon boyu, Problem boyutu, Sınır değerler,
Çaprazlama sabiti (*CR*), Skala faktörü (*SF*)

Output: En iyi maliyet fonksiyonu, En iyi bireyler

Popülasyon ilk değer atama_kaotik(Popülasyon boyu, Problem boyutu, Sınır değerler)

Maliyet hesapla(Popülasyon)

En iyi birey←Popülasyon içindeki en iyi maliyete sahip birey

En iyi maliyet değeri← Maliyet hesapla(En iyi birey)

While(Durdurma kriteri)

 Yeni popülasyon sıfırla

 Mutasyon bireyleri hesapla_kaotik (Skala faktörü, FG stratejisi)

 Çaprazlama işlemi_kaotik (Çaprazlama sabiti)

 Maliyet hesapla(Yeni popülasyon)

If (Maliyet(Yeni birey)≤ Maliyet(Eski birey))

 Yeni Popülasyon←Yeni birey

Else

 Yeni Popülasyon← Eski birey

End

 Popülasyon← Yeni Popülasyon

En iyi birey←Popülasyon içindeki en iyi maliyete sahip birey

En iyi maliyet değeri← Maliyet hesapla(En iyi birey)

End

Return(En iyi maliyet fonksiyonu, En iyi bireyler)

Şekil 3.6 Kaotik Tabanlı FGA Kaba Kodu (Yüzgeç ve Eser,2014)

4. KFGA TABANLI OPTİMİZASYON UYGULAMASI

Bu tez çalışmasında FGA'da temel olarak kullanılan rastgele sayı üretici yerine, kaotik sistemlere dayanan sayı üretici içeren kaotik tabanlı farksal gelişim algoritması (KFGA) geliştirilmiş ve önerilen KFGA yapısının performansının test edilebilmesi için literatürden alınan matematiksel ifadeleri Çizelge 4.1'de verilen optimizasyon test fonksiyonları kullanılmıştır. Bu bölümde farklı FGA stratejileri ve farklı FGA parametreleri için koşturmalar yapılarak başarımları karşılaştırılmıştır.

4.1 Kullanılan Test Fonksiyonları

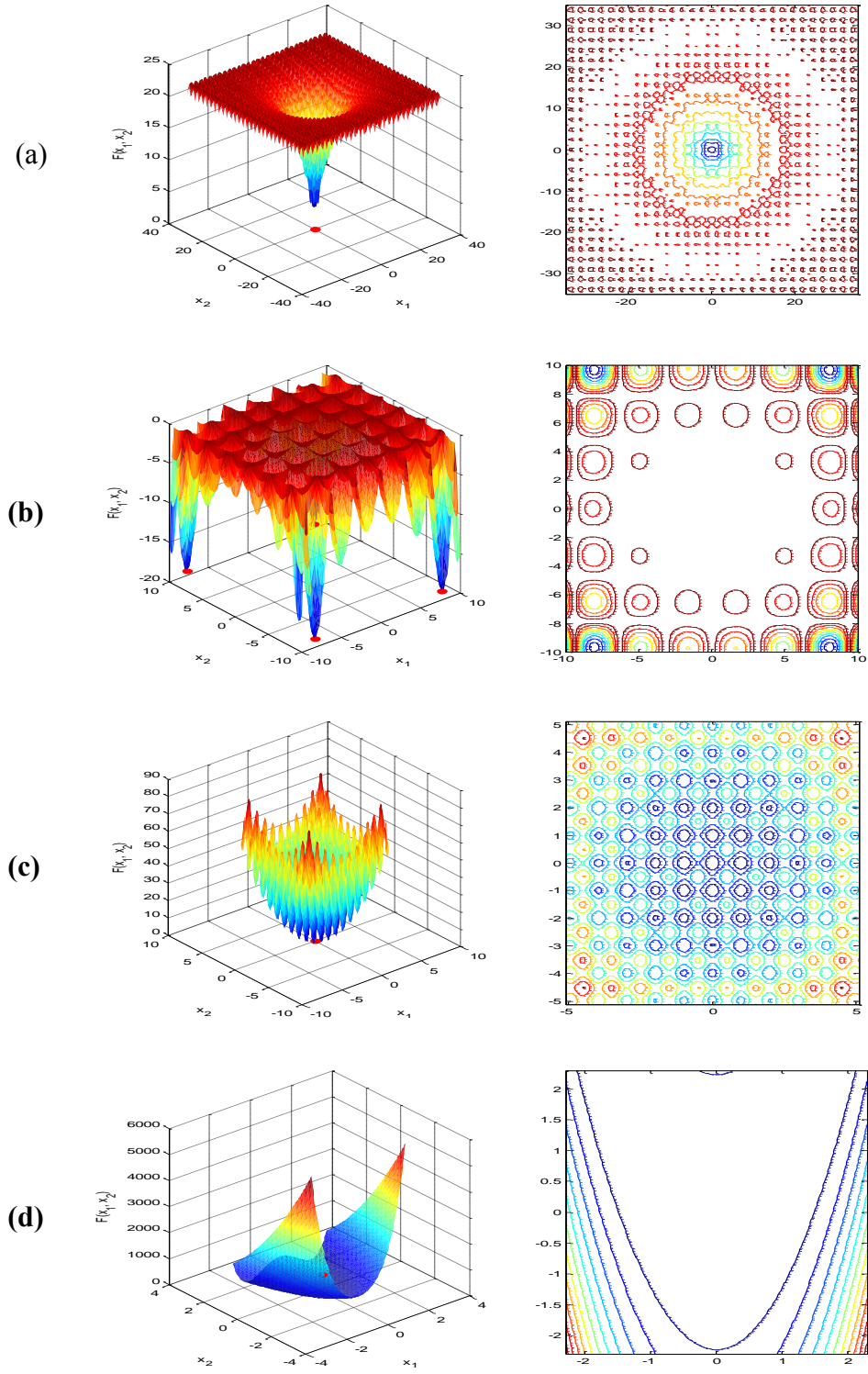
Çizelge 4.1'de literatürden alınan optimizasyon test fonksiyonlarının denklemleri verilmiştir. Seçilen optimizasyon test fonksiyonlarından, bazıları benzer özelliklere sahipken bazıları da farklı özelliklere sahiptir. Rosenbrock, Guinta, Six-hump Camel, Himmelblau optimizasyon test fonksiyonları genel olarak düz arama yüzeylerine sahip olup, sezgisel arama algoritmalarının bu yüzeylerdeki performansı için kullanılmıştır. Testtubeholder ve Penholder optimizasyon test fonksiyonları birden çok tepe ve çukurlardan oluşmakta olup sezgisel arama algoritmalarının bu lokal noktadaki performansı için kullanılmıştır. Ackley ve Holdertable optimizasyon test fonksiyonları bir/birden çok genliği büyük minimum noktasına sahip lokal minimumları da içeren fonksiyonlardır. Rastrigin ve Schwefel optimizasyon test fonksiyonları farklı genlik değerlerine sahip birden fazla çukur ve tepelerden oluşmaktadır. Şekil 4.1'de iki boyut için optimizasyon test problemlerine ait üç boyutlu görünüm ve izdüşüm grafikleri verilmiştir.

Çizelge 4.1.Kullanılan optimizasyon test fonksiyonları

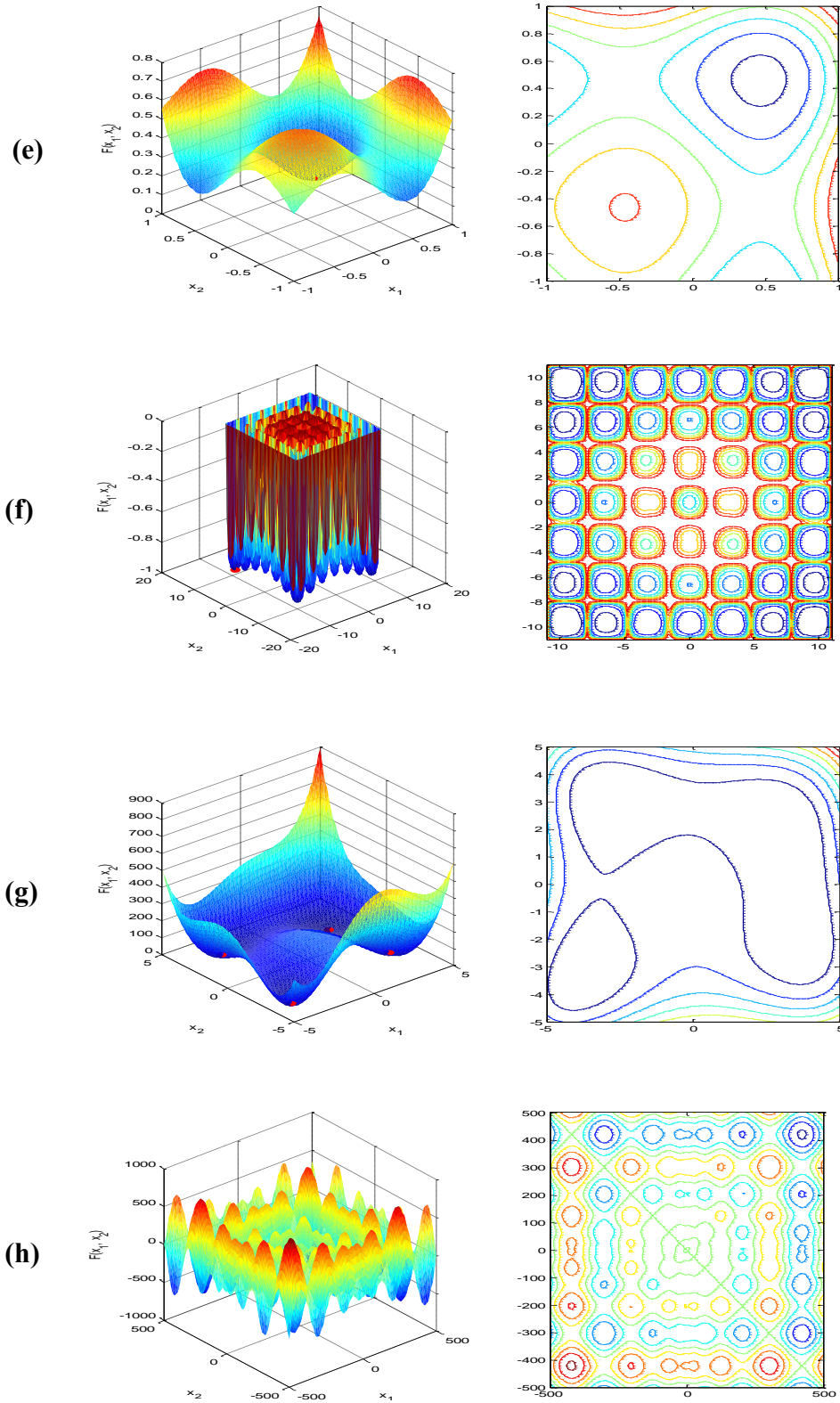
f1-Ackley	$f(x) = -a \exp \left(-b \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2} \right) - \exp \left(\frac{1}{d} \sum_{i=1}^d \cos(cx_i) \right) + a$ $+ \exp(1)$
-----------	--

Çizelge 4.1.Kullanılan optimizasyon test fonksiyonları (devam ediyor)

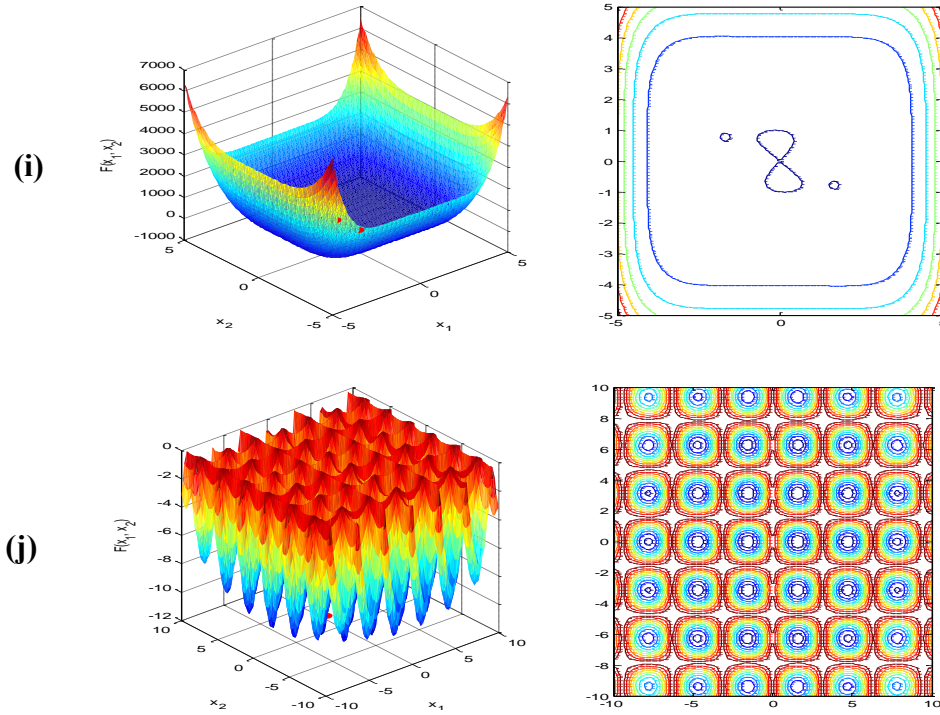
f2- Holdertable	$f(x) = - \left \sin(x_1) \cos(x_2) \exp \left(\left 1 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi} \right \right) \right $
f3-Rastrigin	$f(x) = 10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)]$
f4-Rosenbrock	$f(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
f5-Guinta	$f(x) = 0.6 + \sum_{i=1}^n \left[\sin^2 \left(1 - \frac{16}{15} x_i \right) - \frac{1}{50} \sin \left(4 - \frac{64}{15} x_i \right) - \sin \left(1 - \frac{16}{15} x_i \right) \right]$
f6-Penholder	$f(x) = -\exp \left \exp \left(\left -\frac{\sqrt{x_1^2 + x_2^2}}{\pi} + 1 \right \right) \cos(x_1) \cos(x_2) \right ^{-1}$
f7-Himmelblau	$f(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$
f8- Schwefel	$f(x) = 418.9829d - \sum_{i=1}^d x_i \sin \sqrt{ x_i }$
f9-Six-hump Camel	$f(x) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3} \right) x_1^2 + x_1 x_2 + (-4 + 4x_2^2) x_2^2$
f10-Testtubeholder	$f(x) = -4 \left \exp \left(\left \cos \left(\frac{1}{200} x_1^2 + \frac{1}{200} x_2^2 \right) \right \right) \sin(x_1) \cos(x_2) \right $



Şekil 4.1. Optimizasyon Test Problemleri.
 (a) Ackley (b) Holdertable (c) Rastrigin (d) Rosenbrock



Şekil 4.1. Optimizasyon Test Problemleri (devamı)
 (e) Guinta (f) Penholder (g) Himmelblau (h) Schwefel



Şekil 4.1. Optimizasyon Test Problemleri (devamı)
(i) Sixhumpcamel (j) Testtubeholder

Geliştirilen KFGA yapısı, iki farklı kaotik denklemlerle (Lorenz ve Rössler), 10 farklı optimizasyon test fonksiyonu kullanılarak, Intel(R) Core(TM)2 Duo , P7350@ 2.00 GHz CPU, 4 GB RAM donanımlı, 64 bit işletim sistemine sahip bir bilgisayarda kodlanarak çalıştırılmıştır. KFGA ile yapılan optimizasyon uygulamalarında farklı belirtilmediği sürece kullanılan parametreler NP (Popülasyon değeri) için 20, maksimum iterasyon sayısı için 200, F (skala faktörü) için 1.2 ve CR (çaprazlama sabiti) için 0.7 olarak alınmış olarak alınmıştır.

4.2 Lorenz Tabanlı KFGA Optimizasyon Sonuçları

Literatürden alınmış olan on adet optimizasyon test fonksiyonu (bkz. Çizelge 4.1) için DE/rand/1/bin FGA stratejisi kullanılarak, Lorenz tabanlı KFGA yapısı ile elde edilen sonuçlar Çizelge 4.2’de verilmiştir. Bu kısımda yapılan optimizasyon uygulamasında yukarıda belirtilen strateji için KFGA yapısı 50 defa koşturulmuştur.

Çizelge 4.2’deki MSE değerlerine bakıldığında f3 ve f8 test fonksiyonlarının dışında (f3_MSE= 0.310445, f8_MSE= 10863.01), diğer test fonksiyonları ile yapılan koşturmaların başarımının yüksek olduğu söylenebilir. f3 ve f8 test fonksiyonları

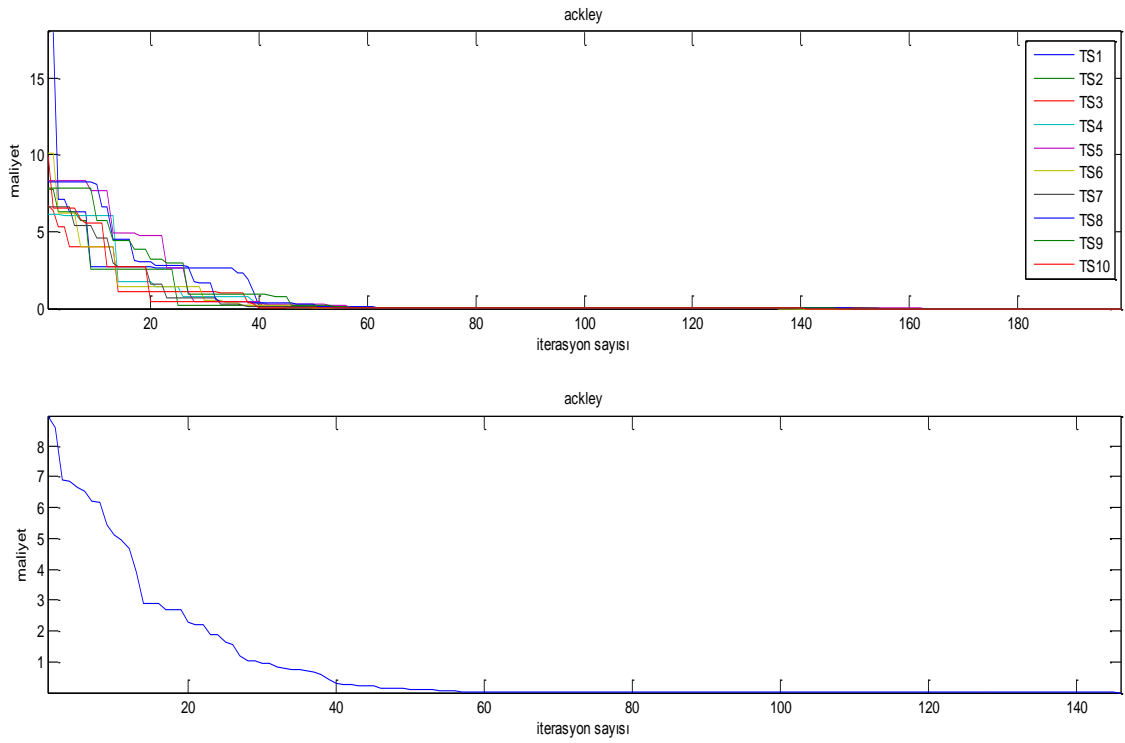
dışında yapılan koşturmalarla başarımların değerleri birbirlerine yakın olmasına rağmen en yüksek başarımların f9 test fonksiyonu ile yapılan koşturmada sağlanmıştır. f1, f2, f4, f7 test fonksiyonlarıyla yapılan koşturmalarla da global sonuca yakın değerler elde edilmiştir.

Lorenz tabanlı KFGA yapısının 50 koşturma sonunda elde edilen ortalama sürelerine bakıldığında, genel olarak sürelerin kısa olduğu ve gerçek zamanlı uygulamalar için önerilen algoritmanın kullanılabilir olduğu söylenebilir. Ortalama iterasyon sayısı, beklenildiği gibi ortalama süreler ile benzer şekilde değişmektedir. Ortalama iterasyon sayısının en fazla 199 olmasının nedeni maksimum iterasyon sayısının 200 olarak seçilmesinden kaynaklanmaktadır. Durdurma kriteri içerisinde maksimum iterasyona varma şartına ilave olarak en iyi maliyet değerinin belirlenen bir Δ değerinden küçük olma şartı konulmuştur. (Δ :Algoritma koşturma sırasında her bir döngüde Δ ' den en iyi değer büyük olduğu ve maksimum iterasyon değerinden küçük olduğu sürece algoritma koşturilmaya devam eder)

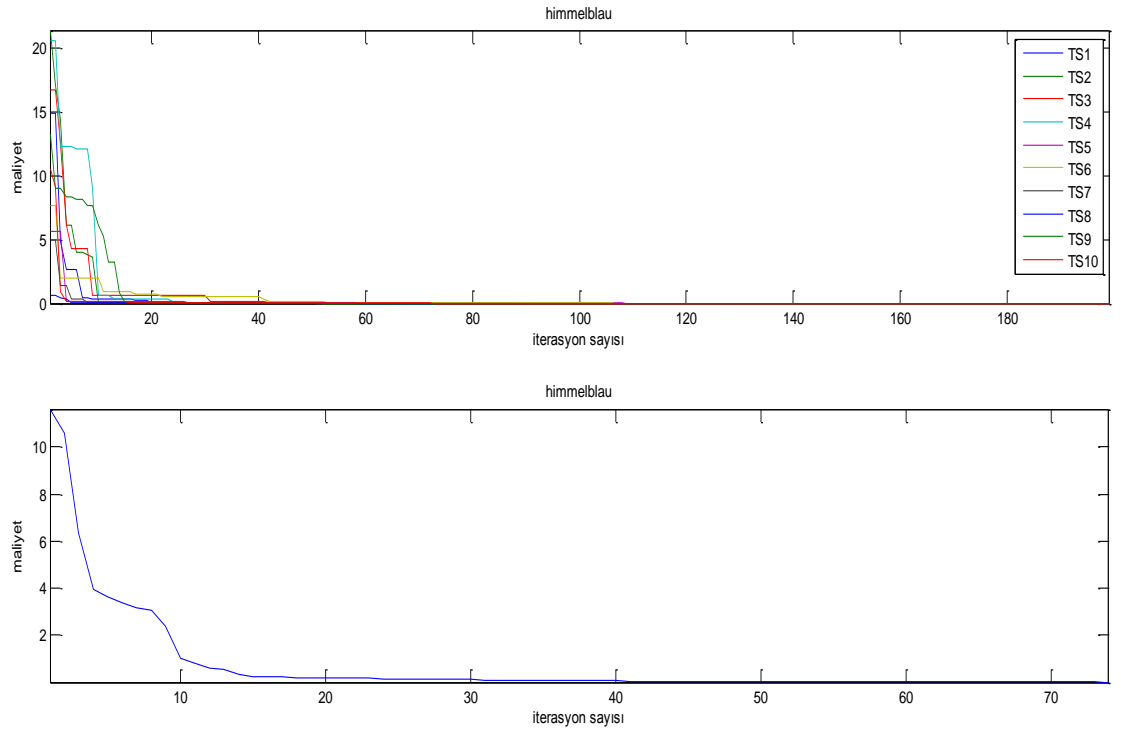
Çizelge 4.2. Lorenz Tabanlı KFGA Optimizasyon Sonuçları (50 defa koşturma, NP=20, F=1.2 ve CR=0.7)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	0,000000;0,000000	140,66	0,123152	0	1E-12
f2	-19,2085	-0,966604;1,159752	199	0,161413	-19,20850257	1,87E-13
f3	0,557176	-0,059694;-0,139291	125,5	0,096809	0	0,310445
f4	0,000001	1,000088;1,000174	81,6	0,062344	0	1E-12
f5	0,06447	0,467320;0,467320	199	0,16659	0,060247218	1,78E-05
f6	-0,96143	-6,110711;-4,567333	199	0,158009	-0,96353	4,41E-06
f7	0,000001	-0,806498;-0,066571	90,6	0,068343	0	1E-12
f8	-733,74	73,691779;131,571274	199	0,167936	-837,9658	10863,01
f9	-1,03163	0,003593;-0,028506	199	0,15605	-1,031628	0
f10	-10,8333	-0,438851;-0,564237	199	0,153035	-10,8723	0,001524

Şekil 4.2 ve Şekil 4.3'de Ackley ve Himmelblau optimizasyon test problemleri için Lorenz tabanlı KFGA'nın 10 tekrar sonucu elde edilen maliyetleri ve ortalama maliyet değişimleri gösterilmiştir. Her iki şekilden de görüleceği üzere her bir koşturma sonucu elde edilen maliyet değişimleri benzer yörüngeleri takip etmektedir. Ortalama maliyet grafiğine bakıldığında ise Lorenz tabanlı KFGA yapısının minimum noktaya başarı ile gittiği görülmektedir.



Şekil 4.2. Ackley optimizasyon test problemi için Lorenz tabanlı KFGA'nın 10 tekrar sonucu (TS1-TS10) maliyet değişimleri ve ortalama maliyet değişimi (NP=20, F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1)



Şekil 4.3. Himmelblau optimizasyon test problemi için Lorenz tabanlı KFGA'nın 10 tekrar sonucu (TS1-TS10) maliyet değişimleri ve ortalama maliyet değişimi (NP=20, F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1)

4.3 Rössler Tabanlı KFGA Optimizasyon Sonuçları

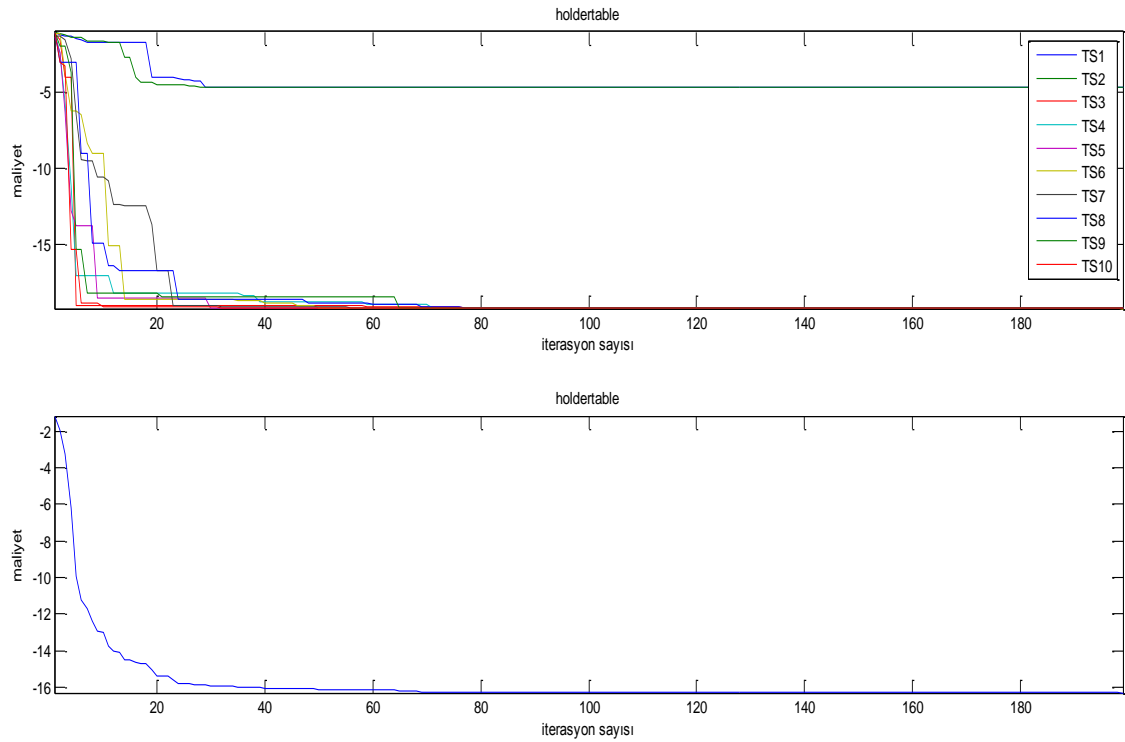
Literatürden alınmış olan on adet optimizasyon test fonksiyonu (bkz. Çizelge 4.1) için DE/rand/1/bin FGA stratejisi kullanılarak, Rössler tabanlı KFGA yapısı ile elde edilen sonuçlar Çizelge 4.3'te verilmiştir. Bu kısımda yapılan optimizasyon uygulamasında yukarıda belirtilen strateji için KFGA yapısı 50 defa koşturulmuştur.

Çizelge 4.3. Rössler Tabanlı KFGA Optimizasyon Sonuçları (50 defa koşma, NP=20, F=1.2 ve CR=0.7)

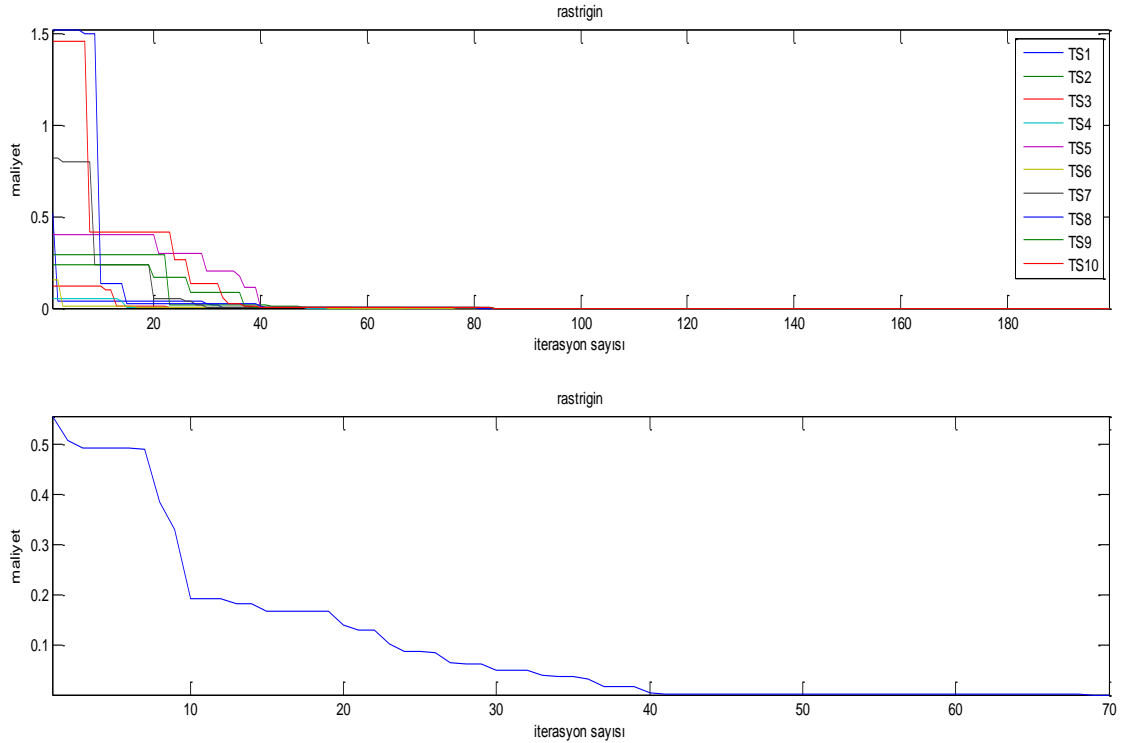
FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	-0,000000;-0,000000	127,36	0,114062	0	1E-12
f2	-16,4118	-0,912011;-0,849705	199	0,165994	-19,20850257	7,821596
f3	0,318387	-0,159192;-0,159189	89,1	0,071072	0	0,10137
f4	0	1,000025;1,000043	67,6	0,055979	0	0
f5	0,06447	0,467320;0,467320	199	0,1732	0,060247218	1,78E-05
f6	-0,84998	0,435369;-1,107981	199	0,165885	-0,96353	0,012895
f7	0	1,543623;0,606838	94,38	0,072846	0	0
f8	-691,097	275,507328;275,507328	199	0,175157	-837,9658	21570,44
f9	-1,03163	-0,014375;0,114025	199	0,159007	-1,031628	0
f10	-10,8679	-0,376771;-0,313977	199	0,162556	-10,8723	1,9E-05

Çizelge 4.3'deki tabloda, MSE değerlerine bakıldığında f2, f3 ve f8 test fonksiyonlarının dışında, diğer test fonksiyonları ile yapılan koşturmaların başarımının yüksek olduğunu söylenebilir. En yüksek başarımlar f4, f7 ve f9 test fonksiyonu için yapılan koşturmalar sonunda sağlanmıştır. f1, f5 ve f10 test fonksiyonlarıyla yapılan koşturmalar da global sonuca yakın değerler elde edilmiştir.

Şekil 4.4 ve Şekil 4.5'de Holdertable ve Rastrigin optimizasyon test problemleri için Rössler tabanlı KFGA'nın 10 tekrar sonucu elde edilen maliyetleri ve ortalama maliyet değişimleri gösterilmiştir. Holdertable test fonksiyonu için iki tekrarda Rössler tabanlı KFGA yapısının lokal minimuma takıldığı, diğer sekiz tekrar için başarılı performans gösterdiği görülmektedir. Rastrigin test fonksiyonu için her bir koşturma sonucu elde edilen maliyet değişimlerinin benzer yörüngeleri takip ettiği Şekil 4.5'den anlaşılmaktadır. Ortalama maliyet grafikleri incelendiğinde, her iki test fonksiyonu için önerilen KFGA yapısının global minimum noktasına başarı ile gittiği görülmektedir.



Şekil 4.4. Holdertable optimizasyon test problemi için Rössler tabanlı KFGA'nın 10 tekrar sonucu (TS1-TS10) maliyet değişimleri ve ortalama maliyet değişimi (NP=20, F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1)



Şekil 4.5. Rastrigin optimizasyon test problemi için Rössler tabanlı KFGA'nın 10 tekrar sonucu (TS1-TS10) maliyet değişimleri ve ortalama maliyet değişimi (NP=20, F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1)

4.4 Farklı KFGA Stratejilerinin Karşılaştırılması (Kaotik Fonksiyon-Lorenz Çekicisi)

Önerilen Lorenz tabanlı KFGA yapısı için en başarılı performansı gösteren stratejinin belirlenmesi için literatürden alınmış olan on adet optimizasyon test fonksiyonu (bkz. Çizelge 4.1) için Çizelge 4.4'te verilen farklı FGA stratejileri kullanılarak algoritma 20'şer defa koşturulmuştur. Lorenz tabanlı KFGA yapısı ile on farklı strateji için elde edilen sonuçlar Çizelge 4.5'de ortalama sonuç ve MSE değerlerine göre, Çizelge 4.6'da ise ortalama iterasyon ve ortalama süre değerlerine göre verilmiştir. Bu kısımda yapılan optimizasyon uygulamasında her bir strateji için önerilen KFGA yapısı 20 defa koşturulmuştur.

Çizelge 4.4. Farklı FGA Stratejileri

ST1	DE/best/1/exp	ST6	DE/best/1/bin
ST2	DE/rand/1/exp	ST7	DE/rand/1/bin
ST3	DE/rand-to-best/1/exp	ST8	DE/rand-to-best/1/bin
ST4	DE/best/2/exp	ST9	DE/best/2/bin
ST5	DE/rand/2/exp	ST10	DE/rand/2/bin

Çizelge 4.5. Lorenz Tabanlı KFGA Optimizasyon Sonuçları (Ortalama Sonuç ve MSE) (20 defa koşma, NP=20, F=1.2 ve CR=0.7)

FN	Global Sonuç	ST1		ST2		ST3		ST4		ST5	
		Ort. Sonuç	MSE	Ort. Sonuç	MSE	Ort. Sonuç	MSE	Ort. Sonuç	MSE	Ort. Sonuç	MSE
f1	0	1,52E+00	2,32E+00	1,00E-06	1,00E-12	1,00E-06	1,00E-12	2,00E-06	4,00E-12	9,00E-06	8,10E-11
f2	-2,E+01	-1,81E+01	1,29E+00	-1,92E+01	3,22E-13	-1,87E+01	2,35E-01	-1,92E+01	2,09E-11	-1,92E+01	7,89E-04
f3	0	1,84E+00	3,39E+00	9,95E-01	9,90E-01	8,96E-01	8,03E-01	1,00E-06	1,00E-12	9,33E-04	8,70E-07
f4	0	8,00E-06	6,40E-11	0,00E+00	0,00E+00	0,00E+00	0,00E+00	1,00E-06	1,00E-12	5,27E-04	2,78E-07
f5	6,02E-02	1,00E-01	1,61E-03	7,16E-02	1,30E-04	8,60E-02	6,63E-04	7,88E-02	3,45E-04	6,45E-02	1,78E-05
f6	-9,64E-01	-9,58E-01	2,76E-05	-9,58E-01	2,76E-05	-9,53E-01	1,04E-04	-9,58E-01	2,76E-05	-9,64E-01	1,00E-10
f7	0	1,11E-04	1,23E-08	0,00E+00	0,00E+00	1,00E-06	1,00E-12	0,00E+00	0,00E+00	9,94E-03	9,87E-05
f8	-8,38E+02	-7,43E+02	8,98E+03	-7,67E+02	5,05E+03	-7,49E+02	7,89E+03	-7,55E+02	6,87E+03	-7,85E+02	2,84E+03
f9	-1,03E+00	-1,03E+00	2,16E-07	-1,03E+00	0,00E+00	-1,03E+00	0,00E+00	-1,03E+00	0,00E+00	-1,03E+00	4,00E-12
f10	-1,09E+01	-1,07E+01	1,66E-02	-1,09E+01	1,37E-04	-1,08E+01	1,20E-03	-1,09E+01	3,90E-05	-1,09E+01	2,97E-04

ST1: best/1/exp, ST2: rand/1/exp, ST3: rand-to-best/1/exp, ST4: best/2/exp, ST5: rand/2/exp

Çizelge 4.5. Lorenz Tabanlı KFGA Optimizasyon Sonuçları (Ortalama Sonuç ve MSE) (20 defa koşma, NP=20, F=1.2 ve CR=0.7) (devamı)

FN	Global Sonuç	ST6		ST7		ST8		S9		ST10	
		Ort. Sonuç	MSE	Ort. Sonuç	MSE	Ort. Sonuç	MSE	Ort. Sonuç	MSE	Ort. Sonuç	MSE
f1	0	1,57E+00	2,46E+00	1,00E-06	1,00E-12	1,00E-06	1,00E-12	3,00E-06	9,00E-12	2,20E-05	4,84E-10
f2	-2,E+01	-1,85E+01	5,25E-01	-1,87E+01	2,35E-01	-1,80E+01	1,46E+00	-1,92E+01	1,34E-09	-1,92E+01	8,88E-05
f3	0	1,35E+00	1,82E+00	4,97E-01	2,47E-01	1,95E-03	3,82E-06	8,95E-01	8,02E-01	5,14E-04	2,64E-07
f4	0	1,40E-05	1,96E-10	0,00E+00	0,00E+00	1,00E-06	1,00E-12	3,60E-05	1,30E-09	6,83E-03	4,67E-05
f5	6,02E-02	8,61E-02	6,70E-04	7,16E-02	1,30E-04	9,32E-02	1,08E-03	7,88E-02	3,45E-04	6,45E-02	1,78E-05
f6	-9,64E-01	-9,59E-01	2,21E-05	-9,58E-01	2,76E-05	-9,64E-01	2,50E-11	-9,64E-01	2,50E-11	-9,58E-01	2,77E-05
f7	0	1,00E-06	1,00E-12	1,00E-06	1,00E-12	1,00E-06	1,00E-12	1,00E-06	1,00E-12	4,33E-03	1,87E-05
f8	-8,38E+02	-7,20E+02	1,40E+04	-7,31E+02	1,14E+04	-7,61E+02	5,93E+03	-7,49E+02	7,89E+03	-7,61E+02	5,93E+03
f9	-1,03E+00	-1,03E+00	9,00E-12	-1,03E+00	0,00E+00	-1,03E+00	0,00E+00	-1,03E+00	0,00E+00	-1,03E+00	4,00E-12
f10	-1,09E+01	-1,06E+01	6,65E-02	-1,09E+01	3,93E-06	-1,08E+01	2,90E-03	-1,09E+01	2,46E-05	-1,09E+01	9,51E-05

ST6: best/1/bin, ST7: rand/1/bin, ST8: rand-to-best/1/bin, ST9: best/2/bin, ST10: rand/2/bin

Çizelge 4.5’de başarıyı yüksek olan (MSE değerleri küçük) değerler **koyu** olarak gösterilmiştir. Tüm stratejiler için 20 koşurma sonunda elde edilen ortalama MSE değerleri incelendiğinde, Lorenz tabanlı KFGA için en başarılı stratejilerin ST2 ve ST7 stratejileri olduğu görülmektedir. En kötü başarıma sahip stratejilerin ise ST1 ve ST6 stratejileri olduğu anlaşılmaktadır.

Çizelge 4.6. Lorenz Tabanlı KFGA Optimizasyon Sonuçları (Ortalama İterasyon ve Ortalama Süre) (20 defa koşma, NP=20, F=1.2 ve CR=0.7)

FN	ST1		ST2		ST3		ST4		ST5	
	Ort. İter.	Ort. Süre(sn)	Ort. İter.	Ort. Süre(sn)	Ort. İter.	Ort. Süre(sn)	Ort. İter.	Ort. Süre(sn)	Ort. İter.	Ort. Süre(sn)
f1	119,80	0,135	143,15	0,151	113,70	0,118	195,20	0,209	198,85	0,209
f2	199,00	0,197	199,00	0,192	199,00	0,188	199,00	0,189	199,00	0,190
f3	121,90	0,117	129,35	0,119	193,75	0,178	165,50	0,150	197,30	0,177
f4	68,80	0,066	84,75	0,079	74,50	0,071	151,65	0,140	189,95	0,173
f5	199,00	0,200	199,00	0,201	199,00	0,200	199,00	0,206	199,00	0,200
f6	199,00	0,195	199,00	0,191	199,00	0,189	199,00	0,189	199,00	0,186
f7	65,60	0,061	90,85	0,082	91,35	0,084	108,00	0,096	180,05	0,160
f8	199,00	0,198	199,00	0,207	199,00	0,204	199,00	0,198	199,00	0,195
f9	199,00	0,189	199,00	0,187	199,00	0,191	199,00	0,186	199,00	0,183
f10	199,00	0,189	199,00	0,182	199,00	0,186	199,00	0,188	199,00	0,181

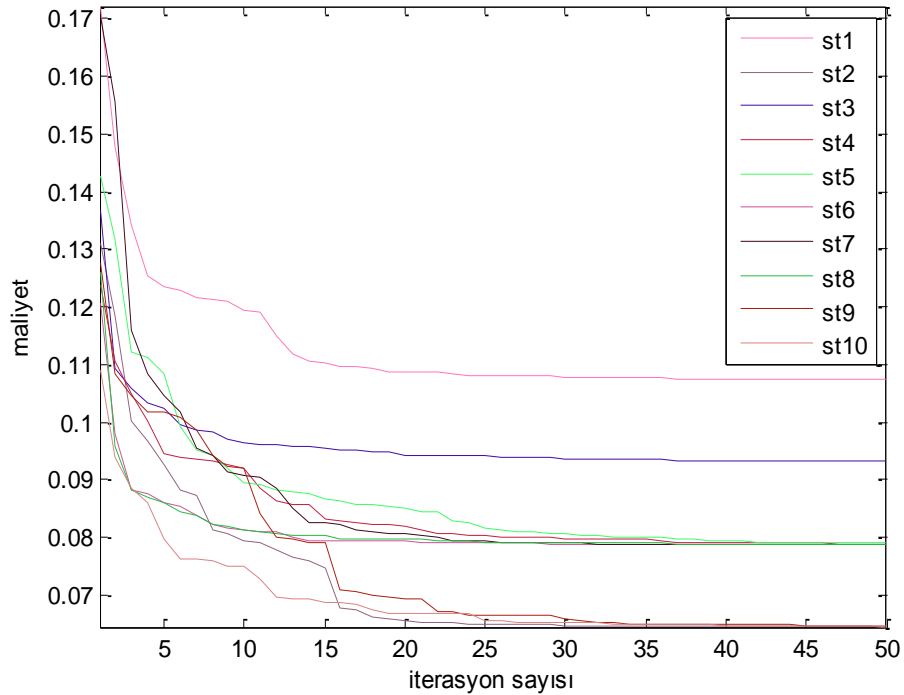
ST1: best/1/exp, ST2: rand/1/exp, ST3: rand-to-best/1/exp, ST4: best/2/exp, ST5: rand/2/exp

Çizelge 4.6. Lorenz Tabanlı KFGA Optimizasyon Sonuçları (Ortalama İterasyon ve Ortalama Süre) (20 defa koşma, NP=20, F=1.2 ve CR=0.7) (devamı)

FN	ST6		ST7		ST8		S9		ST10	
	Ort. İter.	Ort. Süre(sn)	Ort. İter.	Ort. Süre(sn)	Ort. İter.	Ort. Süre(sn)	Ort. İter.	Ort. Süre(sn)	Ort. İter.	Ort. Süre(sn)
f1	131,65	0,124	143,20	0,127	113,50	0,101	197,45	0,174	199,00	0,177
f2	199,00	0,164	199,00	0,164	199,00	0,162	199,00	0,161	199,00	0,165
f3	154,95	0,123	123,50	0,097	186,70	0,144	156,50	0,121	195,40	0,151
f4	55,80	0,046	86,85	0,069	79,95	0,064	157,30	0,122	196,15	0,152
f5	199,00	0,176	199,00	0,172	199,00	0,173	199,00	0,171	199,00	0,176
f6	199,00	0,163	199,00	0,163	199,00	0,166	199,00	0,164	199,00	0,162
f7	59,90	0,047	86,05	0,067	84,90	0,065	109,10	0,084	181,45	0,139
f8	199,00	0,180	199,00	0,173	199,00	0,173	199,00	0,174	199,00	0,171
f9	199,00	0,159	199,00	0,157	199,00	0,161	199,00	0,160	199,00	0,158
f10	199,00	0,160	199,00	0,159	199,00	0,154	199,00	0,158	199,00	0,155

ST6: best/1/bin, ST7: rand/1/bin, ST8: rand-to-best/1/bin, ST9: best/2/bin, ST10: rand/2/bin

Ortalama sürelerle göre en iyi başarımlar binom çaprazlama stratejilerine aittir. Ortalama iterasyon açısından en iyi başarımlar, en az iterasyon değerine sekiz defa ulaşan ST6 stratejisidir, en kötü başarımlar ise en az iterasyon değerine beşer defa ulaşan ST7 ve ST9 stratejileridir. Şekil 4.6'da Guinta optimizasyon test problemi için Lorenz tabanlı KFGA'nın farklı stratejilerindeki maliyet değişimleri görülmektedir.



Şekil 4.6. Guinta optimizasyon test problemi için Lorenz tabanlı KFGA'nın farklı stratejilerinde maliyet değişimleri (NP=20, F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, tekrar sayısı=20)

4.5 Farklı KFGA Stratejilerinin Karşılaştırılması (Kaotik Fonksiyon-Rössler Çekicisi)

Önerilen Rössler tabanlı KFGA yapısı için en başarılı performansı gösteren stratejinin belirlenmesi için literatürden alınmış olan on adet optimizasyon test fonksiyonu (bkz. Çizelge 4.1) için Çizelge 4.4'te verilen farklı FGA stratejileri kullanılarak algoritma 20'şer defa koşturulmuştur. Rössler tabanlı KFGA yapısı ile on farklı strateji için elde edilen sonuçlar, Çizelge 4.7'de ortalama sonuç ve MSE değerlerine göre, Çizelge 4.8'de ise ortalama iterasyon ve ortalama süre değerlerine göre verilmiştir. Bu kısımda yapılan optimizasyon uygulamalarında her bir strateji için KFGA yapısı 20 defa koşturulmuştur.

Çizelge 4.7'de MSE değerlerine bakıldığında en başarılı, en düşük MSE değerlerine sahip (adet bazında), ST8 (beş tane en küçük MSE değeri), daha sonra ise ST2, ST4, ST7, ST9 ve ST10 stratejileri (dört tane en küçük MSE değeri), görülmektedir. En kötü koşturma başarımı ise ST1 ve ST6 stratejilerinde olduğu görülmektedir. f9 test fonksiyonunda ST6 stratejisi hariç bütün stratejilerde en küçük MSE değerlerini aldığı görülmektedir.

Çizelge 4.7. Rössler Tabanlı KFGA Optimizasyon Sonuçları (Ortalama Sonuç ve MSE) (20 defa koşma, NP=20, F=1.2 ve CR=0.7)

FN	Global Sonuç	ST1		ST2		ST3		ST4		ST5	
		Ort. Sonuç	MSE	Ort. Sonuç	MSE	Ort. Sonuç	MSE	Ort. Sonuç	MSE	Ort. Sonuç	MSE
f1	0	5,46E-04	2,98E-07	1,00E-06	1,00E-12	1,00E-06	1,00E-12	1,00E-06	1,00E-12	4,00E-06	1,60E-11
f2	-2,E+01	-1,16E+01	5,82E+01	-1,51E+01	1,70E+01	-1,21E+01	5,04E+01	-1,92E+01	6,86E-05	-1,91E+01	3,72E-03
f3	0	3,98E-01	1,58E-01	3,98E-01	1,58E-01	7,96E-01	6,34E-01	2,49E-01	6,19E-02	2,00E-01	3,98E-02
f4	0	0,00E+00	0,00E+00	1,00E-06	1,00E-12	1,00E-06	1,00E-12	0,00E+00	0,00E+00	0,00E+00	0,00E+00
f5	6,02E-02	6,45E-02	1,78E-05	6,45E-02	1,78E-05	6,45E-02	1,78E-05	6,45E-02	1,78E-05	6,45E-02	1,78E-05
f6	-9,64E-01	-7,77E-01	3,48E-02	-8,58E-01	1,11E-02	-8,04E-01	2,54E-02	-9,21E-01	1,85E-03	-8,93E-01	4,94E-03
f7	0	3,10E-05	9,61E-10	0,00E+00	0,00E+00	1,00E-06	1,00E-12	1,00E-06	1,00E-12	5,91E-03	3,49E-05
f8	-8,38E+02	-5,54E+02	8,08E+04	-7,20E+02	1,40E+04	-5,48E+02	8,42E+04	-6,84E+02	2,37E+04	-7,49E+02	7,99E+03
f9	-1,03E+00	-1,03E+00	3,60E-11	-1,03E+00	0,00E+00	-1,03E+00	0,00E+00	-1,03E+00	0,00E+00	-1,03E+00	0,00E+00
f10	-1,09E+01	-1,08E+01	1,58E-03	-1,09E+01	6,28E-05	-1,09E+01	3,16E-05	-1,09E+01	5,11E-05	-1,09E+01	1,81E-04
ST1: best/1/exp, ST2: rand/1/exp, ST3: rand-to-best/1/exp, ST4: best/2/exp, ST5: rand/2/exp											

Çizelge 4.7. Rössler Tabanlı KFGA Optimizasyon Sonuçları (Ortalama Sonuç ve MSE) (20 defa koşma, NP=20, F=1.2 ve CR=0.7) (devamı)

		ST6		ST7		ST8		S9		ST10	
FN	Global Sonuç	Ort. Sonuç	MSE	Ort. Sonuç	MSE	Ort. Sonuç	MSE	Ort. Sonuç	MSE	Ort. Sonuç	MSE
f1	0	1,00E-06	1,00E-12	1,00E-06	1,00E-12	1,00E-06	1,00E-12	1,00E-06	1,00E-12	1,00E-05	1,00E-10
f2	-2,E+01	-1,85E+01	5,25E-01	-1,67E+01	6,25E+00	-1,26E+01	4,33E+01	-1,92E+01	3,43E-09	-1,92E+01	1,67E-03
f3	0	1,35E+00	1,82E+00	2,98E-01	8,91E-02	2,99E-01	8,95E-02	1,99E-01	3,96E-02	9,95E-02	9,90E-03
f4	0	1,40E-05	1,96E-10	1,00E-06	1,00E-12	0,00E+00	0,00E+00	1,00E-06	1,00E-12	0,00E+00	0,00E+00
f5	6,02E-02	8,61E-02	6,70E-04	6,45E-02	1,78E-05	6,45E-02	1,78E-05	6,45E-02	1,78E-05	6,45E-02	1,78E-05
f6	-9,64E-01	-9,59E-01	2,21E-05	-8,31E-01	1,75E-02	-8,04E-01	2,54E-02	-9,07E-01	3,20E-03	-9,07E-01	3,21E-03
f7	0	1,00E-06	1,00E-12	1,00E-06	1,00E-12	0,00E+00	0,00E+00	1,00E-06	1,00E-12	2,17E-03	4,70E-06
f8	-8,38E+02	-7,20E+02	1,40E+04	-7,08E+02	1,70E+04	-5,71E+02	7,10E+04	-7,20E+02	1,40E+04	-7,76E+02	3,78E+03
f9	-1,03E+00	-1,03E+00	9,00E-12	-1,03E+00	0,00E+00	-1,03E+00	0,00E+00	-1,03E+00	0,00E+00	-1,03E+00	1,00E-12
f10	-1,09E+01	-1,06E+01	6,65E-02	-1,09E+01	8,86E-06	-1,09E+01	6,35E-05	-1,09E+01	1,31E-04	-1,09E+01	4,12E-04
ST6: best/1/bin, ST7: rand/1/bin, ST8: rand-to-best/1/bin, ST9: best/2/bin, ST10: rand/2/bin											

Çizelge 4.8’de ortalama süreye göre en iyi başarımlar, en kısa süre değerlerine göre beş fonksiyonda ulaşan ST7 ve dört fonksiyonda ulaşan ST8 stratejisidir. Ortalama iterasyon açısından en iyi başarımlar, en az iterasyon değerine sekiz defa ulaşan ST1 stratejisidir, en kötü başarımlar ise en az iterasyon değerine iki defa ulaşan ST8 stratejisidir.

Çizelge 4.8. Rössler Tabanlı KFGA Optimizasyon Sonuçları (Ortalama İterasyon ve Ortalama Süre) (20 defa koşma, NP=20, F=1.2 ve CR=0.7)

	ST1		ST2		ST3		ST4		ST5	
FN	Ort. İter.	Ort. Süre(sn)	Ort. İter.	Ort. Süre(sn)	Ort. İter.	Ort. Süre(sn)	Ort. İter.	Ort. Süre(sn)	Ort. İter.	Ort. Süre(sn)
f1	107,9	0,116	129	0,137	100	0,116	182	0,207	195	0,202
f2	199	0,195	199	0,194	199	0,209	199	0,216	199	0,214
f3	80,2	0,078	97,4	0,093	143	0,142	129	0,13	153	0,14
f4	49,6	0,05	78	0,077	55	0,058	102	0,105	134	0,126
f5	199	0,214	199	0,207	199	0,209	199	0,219	199	0,203
f6	199	0,2	199	0,192	199	0,219	199	0,216	199	0,194
f7	57,9	0,055	118	0,112	60,7	0,06	119	0,118	199	0,184
f8	199	0,212	199	0,21	199	0,222	199	0,214	199	0,203
f9	199	0,209	199	0,2	199	0,215	199	0,198	199	0,185
f10	199	0,197	199	0,207	199	0,211	199	0,215	199	0,185

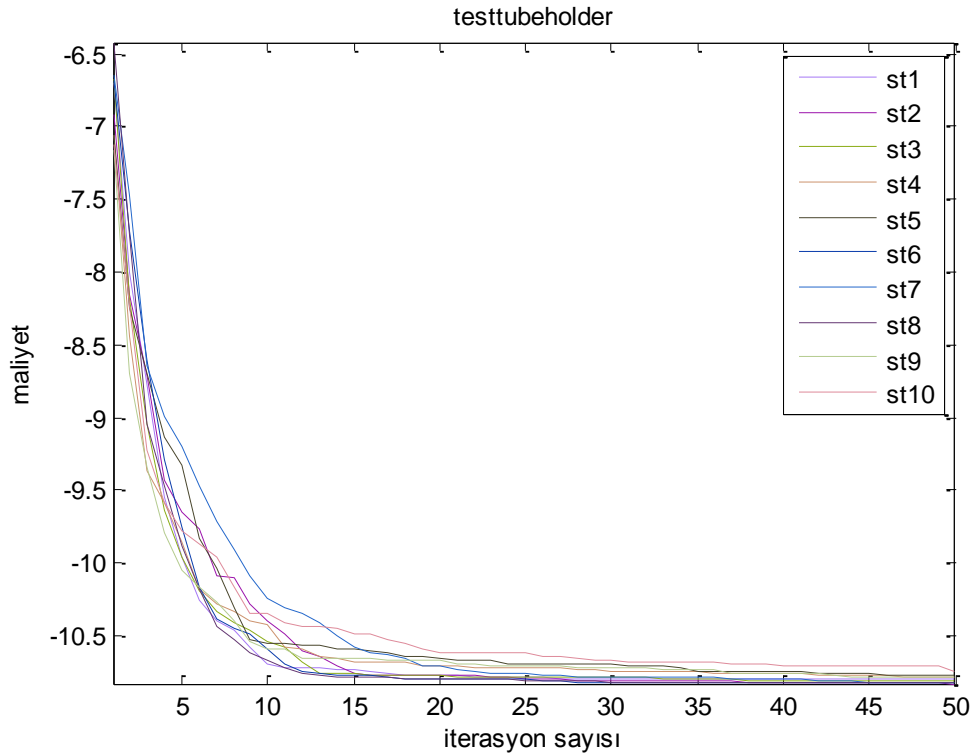
ST1: best/1/exp, ST2: rand/1/exp, ST3: rand-to-best/1/exp, ST4: best/2/exp, ST5: rand/2/exp

Çizelge 4.8. Rössler Tabanlı KFGA Optimizasyon Sonuçları (Ortalama İterasyon ve Ortalama Süre) (20 defa koşma, NP=20, F=1.2 ve CR=0.7)(devamı)

	ST6		ST7		ST8		ST9		ST10	
FN	Ort. İter.	Ort. Süre(sn)	Ort. İter.	Ort. Süre(sn)	Ort. İter.	Ort. Süre(sn)	Ort. İter.	Ort. Süre(sn)	Ort. İter.	Ort. Süre(sn)
f1	111,3	0,104	125	0,115	99,8	0,092	183	0,168	197	0,183
f2	199	0,179	199	0,164	199	0,166	199	0,164	199	0,167
f3	94,75	0,079	88,1	0,071	108	0,087	121	0,104	158	0,129
f4	52	0,044	67,8	0,055	53,1	0,044	96,4	0,079	128	0,106
f5	199	0,18	199	0,175	199	0,176	199	0,175	199	0,176
f6	199	0,169	199	0,167	199	0,166	199	0,167	199	0,168
f7	59,85	0,049	93,5	0,074	55,4	0,044	126	0,1	179	0,142
f8	199	0,182	199	0,174	199	0,18	199	0,178	199	0,178
f9	199	0,165	199	0,16	199	0,165	199	0,169	199	0,16
f10	199	0,174	199	0,159	199	0,161	199	0,162	199	0,163

ST6: best/1/bin, ST7: rand/1/bin, ST8: rand-to-best/1/bin, ST9: best/2/bin, ST10: rand/2/bin

Şekil 4.7’de Testtubeholder optimizasyon test problemi için Rössler tabanlı KFGA'nın farklı stratejilerindeki maliyet değişimleri görülmektedir.



Şekil 4.7. Testtubeholder optimizasyon test problemi için Rössler tabanlı KFGA'nın farklı stratejilerinde maliyet değişimleri (NP=20, F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, tekrar sayısı=20)

4.6 Farklı Popülasyon Değerleri için Lorenz Tabanlı KFGA'nın Performansı

Farklı popülasyon değerleri (NP) için önerilen Lorenz tabanlı KFGA yapısının performansını incelemek için literatürden alınmış olan on adet optimizasyon test fonksiyonu (bkz. Çizelge 4.1) ile DE/rand/1/bin FGA stratejisi kullanılarak, elde edilen sonuçlar aşağıdaki çizelgelerde (Çizelge 4.9-4.13) verilmiştir. Bu kısımda yapılan optimizasyon uygulamalarında NP için beş farklı değer (10, 20, 30, 40 ve 50) kullanılarak yukarıda belirtilen strateji için KFGA yapısı 20 defa koşturulmuştur.

Çizelge 4.9. Lorenz Tabanlı KFGA Optimizasyon Sonuçları¹ (NP=10)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,985004	-1,049972;-1,049972	164	0,08598	0	0,970233
f2	-17,0173	-7,579099;-9,351745	199	0,101158	-19,20850257	4,801469
f3	6,715957	-1,193975;-1,243702	185,85	0,106723	0	45,10408
f4	0,000573	0,993927;0,988426	103,35	0,052269	0	3,28E-07
f5	0,150574	0,027124;0,027122	199	0,104813	0,060247218	0,008159
f6	-0,91618	-8,232451;-8,232451	199	0,100175	-0,96353	0,002242
f7	0	-3,779320;-3,283187	77,6	0,041381	0	0
f8	-601,089	-302,52493;-302,52493	199	0,104912	-837,9658	56110,57
f9	-1,03162	0,027254;-0,213728	199	0,099967	-1,031628	6,4E-11
f10	-10,7032	-2,201647;-2,500629	199	0,09719	-10,8723	0,028583

Çizelge 4.10. Lorenz Tabanlı KFGA Optimizasyon Sonuçları² (NP=20)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	0,000000;0,000000	142,85	0,128005	0	1E-12
f2	-19,2085	-0,000055;-2,899356	199	0,166005	-19,20850257	3,22E-13
f3	0,994956	-0,149247;-0,149249	130,65	0,101536	0	0,989937
f4	0,000001	0,999906;0,999805	83,4	0,066221	0	1E-12
f5	0,06447	0,467320;0,467320	199	0,170836	0,060247218	1,78E-05
f6	-0,95827	-1,772151;-2,736773	199	0,163877	-0,96353	2,76E-05
f7	0	-1,172745;-1,049122	88,55	0,068794	0	0
f8	-766,903	203,920642;203,920642	199	0,171404	-837,9658	5049,954
f9	-1,03163	0,008984;-0,071266	199	0,156716	-1,031628	0
f10	-10,8586	-0,157062;0,000607	199	0,158067	-10,8723	0,000187

Çizelge 4.11. Lorenz Tabanlı KFGA Optimizasyon Sonuçları³ (NP=30)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	0,000000;0,000000	138,2	0,175937	0	1E-12
f2	-19,2085	0,805516;-0,966471	199	0,228484	-19,20850257	1,8672E-13
f3	0,000001	-0,000007;0,000003	110,7	0,121501	0	1E-12
f4	0	0,999976;0,999954	75,7	0,084811	0	0
f5	0,06447	0,467320;0,467320	199	0,240961	0,060247218	1,7832E-05
f6	-0,96354	-3,858463;-2,893855	199	0,230232	-0,96353	2,5E-11
f7	0	2,380517;1,086888	88,15	0,094096	0	0
f8	-826,122	384,794062;384,794062	199	0,243051	-837,9658	140,276996
f9	-1,03163	0,017968;-0,142531	199	0,219591	-1,031628	0
f10	-10,8723	0,785301;-0,000001	199	0,218981	-10,8723	0

Çizelge 4.12. Lorenz Tabanlı KFGA Optimizasyon Sonuçları⁴ (NP=40)

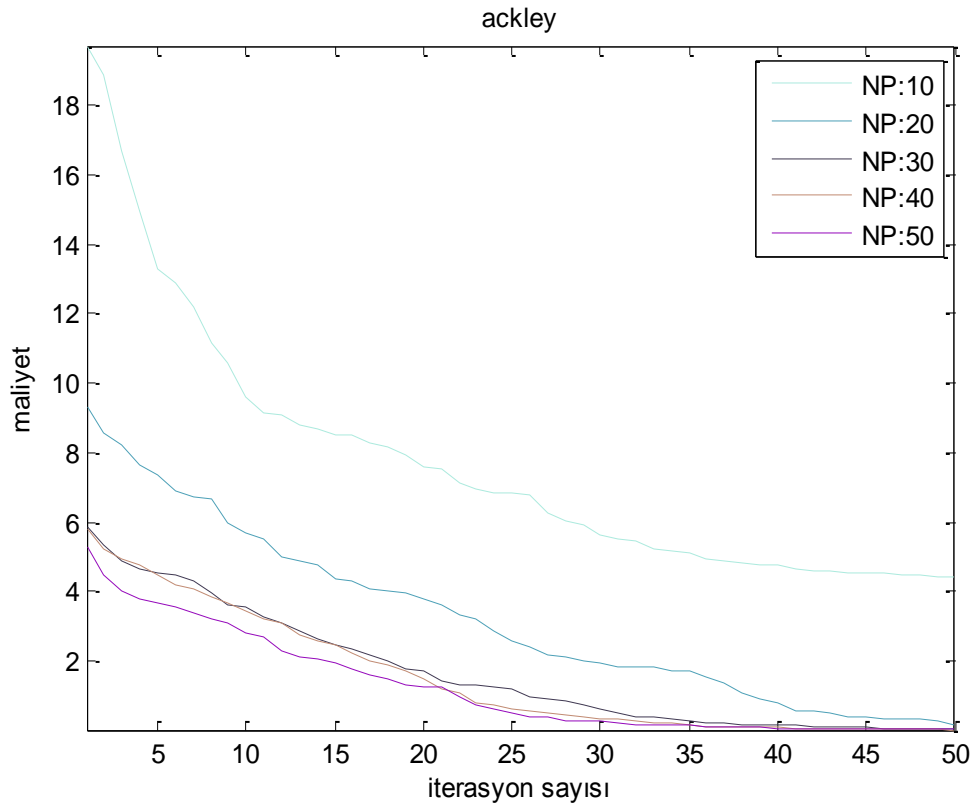
FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	-0,000000;-0,000000	137,3	0,219377	0	1E-12
f2	-19,2085	-0,805483;0,966421	199	0,287711	-19,20850257	3,22E-13
f3	0,994956	-0,149247;-0,149249	130,65	0,101536	0	0,989937
f4	0	1,000060;1,000125	74,3	0,106771	0	0
f5	0,06447	0,467320;0,467320	199	0,31155	0,060247218	1,78E-05
f6	-0,96354	-0,964604;-0,964609	199	0,289531	-0,96353	2,5E-11
f7	0	2,999999;1,999996	88	0,121167	0	0
f8	-837,966	420,968746;420,968746	199	0,307824	-837,9658	6,25E-10
f9	-1,03163	-0,017968;0,142531	199	0,281874	-1,031628	0
f10	-10,8723	-0,000000;0,000009	199	0,284591	-10,8723	0

Çizelge 4.13. Lorenz Tabanlı KFGA Optimizasyon Sonuçları⁵ (NP=50)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	-0,000000;0,000000	130,15	0,252652	0	1E-12
f2	-19,2085	-4,027524;-0,966437	199	0,351459	-19,20850257	3,22E-13
f3	0,000001	0,000001;0,000001	102,55	0,174464	0	1E-12
f4	0,000001	1,000214;1,000398	71,75	0,122815	0	1E-12
f5	0,06447	0,467320;0,467320	199	0,374695	0,060247218	1,78E-05
f6	-0,96354	-0,964624;-1,929236	199	0,351361	-0,96353	2,5E-11
f7	0	3,029236;1,807604	83,15	0,139091	0	0
f8	-837,966	420,968746;420,968746	199	0,374665	-837,9658	6,25E-10
f9	-1,03163	0,000000;0,000000	199	0,340672	-1,031628	0
f10	-10,8723	-0,314132;0,000003	199	0,339502	-10,8723	0

Çizelgelerdeki sonuçlar incelendiğinde popülasyon büyüklüğünün 30 değerinden sonraki değerlerde optimizasyon başarımını sağladığı ve sabitlendiği anlaşılmaktadır. Beklenildiği üzere popülasyon büyüklüğü arttığında ortalama sürelerin de arttığı görülmektedir. Dolayısıyla gerçek zamanlı optimizasyon problemlerinde, hızın önemli olduğu sistemlerde popülasyon boyunu yaklaşık olarak problem boyutunun 15 katı ($NP=D \times 15$) olarak almanın uygun olduğu anlaşılmıştır.

Şekil 4.8’de Ackley optimizasyon test problemi için Lorenz tabanlı KFGA'nın farklı popülasyon sayıları için elde edilen maliyet değişimleri görülmektedir. Şekilde popülasyon büyüklüğünün 10 ve 20 olduğu değerlerde başarımının diğer değerlere göre az olduğu görülmektedir. Diğer popülasyon büyüklüklerinde 30. iterasyondan sonra başarımın arttığı ve yaklaşık olarak aynı olduğu anlaşılmaktadır.



Şekil 4.8. Ackley optimizasyon test problemi için Lorenz tabanlı KFGA'nın farklı popülasyon sayılarında maliyet değişimleri ($F=1.2$, $CR=0.7$, $itermax=200$, $\Delta=1e-6$, strateji:DE/rand/1, tekrar sayısı=20)

4.7 Farklı Popülasyon Değerleri için Rössler Tabanlı KFGA'nın Performansı

Farklı popülasyon değerleri (NP) için önerilen Rössler tabanlı KFGA yapısının performansını incelemek için literatürden alınmış olan on adet optimizasyon test fonksiyonu (bkz. Çizelge 4.1) ile DE/rand/1/bin FGA stratejisi kullanılarak, elde edilen sonuçlar aşağıdaki çizelgelerde (Çizelge 4.14-4.18) verilmiştir. Bu kısımda yapılan optimizasyon uygulamalarında NP için beş farklı değer (10, 20, 30, 40 ve 50) kullanılarak yukarıda belirtilen strateji için KFGA yapısı 20 defa koşturulmuştur.

Çizelge 4.14. Rössler Tabanlı KFGA Optimizasyon Sonuçları¹ (NP=10)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	-0,000000;0,000000	133,25	0,071594	0	1E-12
f2	-3,73847	-1,494333;-1,488160	199	0,099075	-19,20850257	239,3218
f3	0,49748	-0,248742;-0,248742	96,8	0,050051	0	0,247486
f4	0,000001	1,000135;1,000257	77,3	0,039432	0	1E-12
f5	0,06447	0,467320;0,467320	199	0,1076	0,060247218	1,78E-05
f6	-0,69249	-0,358942;-1,361416	199	0,102753	-0,96353	0,073464
f7	0,000001	0,966188;0,415052	87,95	0,043153	0	1E-12
f8	-451,654	76,428161;87,724618	199	0,114272	-837,9658	149236,6
f9	-1,03163	-0,026965;0,213890	199	0,09906	-1,031628	1E-12
f10	-10,8405	-0,474464;-0,951933	199	0,098477	-10,8723	0,001012

Çizelge 4.15. Rössler Tabanlı KFGA Optimizasyon Sonuçları² (NP=20)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	0,000000;-0,000000	124,5	0,115411	0	1E-12
f2	-16,7085	-3,376478;-1,620167	199	0,163982	-19,20850257	6,249968
f3	0,298488	-0,149247;-0,149244	88,1	0,070602	0	0,089095
f4	0,000001	0,999960;0,999932	67,75	0,055391	0	1E-12
f5	0,06447	0,467320;0,467320	199	0,17475	0,060247218	1,78E-05
f6	-0,83118	-3,398307;-1,469006	199	0,167249	-0,96353	0,017516
f7	0,000001	1,692798;1,264057	93,5	0,074363	0	1E-12
f8	-707,679	275,634442;275,634442	199	0,174374	-837,9658	16974,58
f9	-1,03163	0,026952;-0,213797	199	0,16036	-1,031628	0
f10	-10,8693	-0,471082;-0,471078	199	0,15935	-10,8723	8,86E-06

Çizelge 4.16. Rössler Tabanlı KFGA Optimizasyon Sonuçları³ (NP=30)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	-0,000000;0,000000	129,65	0,166915	0	1E-12
f2	-19,2085	0,805455;-0,966422	199	0,233398	-19,20850257	3,22E-13
f3	0,000001	-0,000008;-0,000002	76,4	0,086662	0	1E-12
f4	0	0,999930;0,999858	63,45	0,073158	0	0
f5	0,06447	0,467320;0,467320	199	0,247376	0,060247218	1,78E-05
f6	-0,94203	-3,544317;-1,615063	199	0,229363	-0,96353	0,000462
f7	0,000001	0,724668;0,792287	85,15	0,094889	0	1E-12
f8	-737,29	302,924347;302,924347	199	0,245764	-837,9658	10135,63
f9	-1,03163	-0,000000;-0,000000	199	0,230134	-1,031628	0
f10	-10,8723	0,157393;0,000198	199	0,226293	-10,8723	4E-10

Çizelge 4.17. Rössler Tabanlı KFGA Optimizasyon Sonuçları⁴ (NP=40)

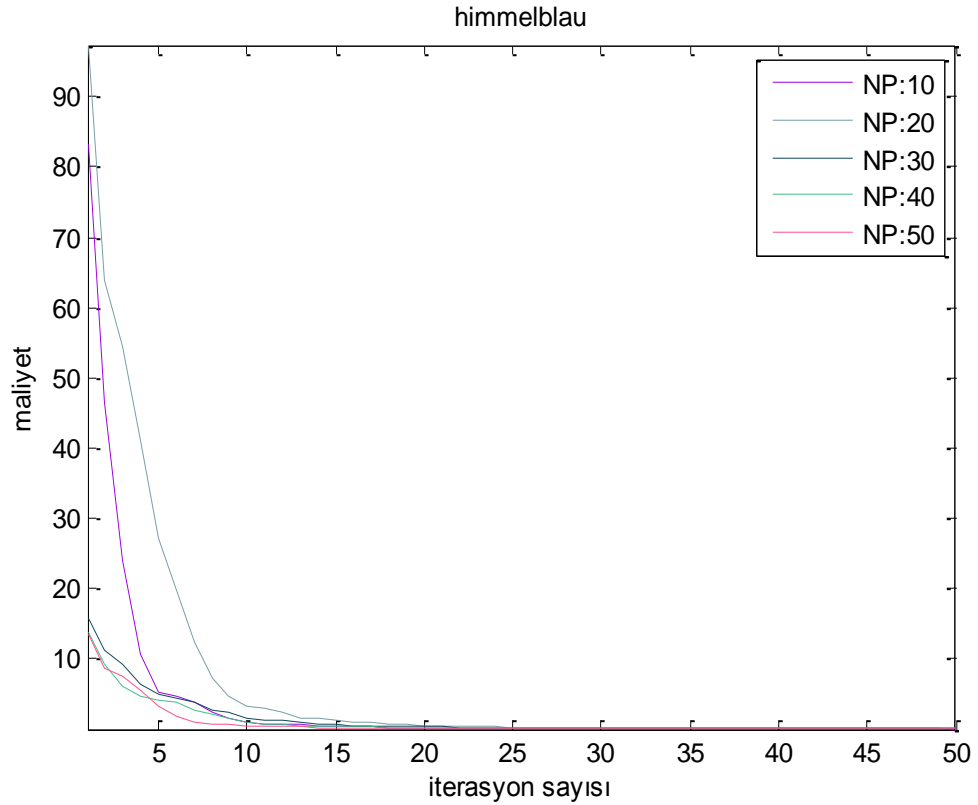
FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	-0,000000;-0,000000	125,05	0,209382	0	1E-12
f2	-19,2085	-5,638512;-5,798753	199	0,298631	-19,20850257	1,87E-13
f3	0	-0,000000;-0,000000	83,55	0,120034	0	0
f4	0,000001	0,999809;0,999628	64,85	0,095458	0	1E-12
f5	0,06447	0,467320;0,467320	199	0,315622	0,060247218	1,78E-05
f6	-0,94203	-2,579689;-3,544306	199	0,2943	-0,96353	0,000462
f7	0,000001	2,070788;0,630278	100,2	0,140695	0	1E-12
f8	-784,667	321,329473;321,329473	199	0,311018	-837,9658	2840,716
f9	-1,03163	-0,000000;0,000000	199	0,284682	-1,031628	0
f10	-10,8723	-0,314104;-0,000067	199	0,283565	-10,8723	0

Çizelge 4.18. Rössler Tabanlı KFGA Optimizasyon Sonuçları⁵ (NP=50)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	0,000000;-0,000000	131	0,256053	0	1E-12
f2	-19,2085	-4,833027;-3,865835	199	0,354323	-19,20850257	3,22E-13
f3	0,000001	0,000006;-0,000001	82,65	0,142451	0	1E-12
f4	0,000001	1,000144;1,000299	63,1	0,111292	0	1E-12
f5	0,06447	0,467320;0,467320	199	0,379919	0,060247218	1,78E-05
f6	-0,94203	-4,508941;-6,438157	199	0,357066	-0,96353	0,000462
f7	0	2,012318;1,015137	94	0,156622	0	0
f8	-796,511	357,504157;357,504157	199	0,379584	-837,9658	1718,478
f9	-1,03163	-0,017968;0,142531	199	0,345004	-1,031628	0
f10	-10,8723	0,000102;-0,000031	199	0,344797	-10,8723	1E-12

Çizelgelerdeki sonuçlar incelendiğinde popülasyon büyüklüğünün Lorenz kaotik sistemine benzer şekilde 30 değerinden sonraki değerlerde optimizasyon başarımını sağladığı ve sabitlendiği anlaşılmaktadır. Beklenildiği üzere popülasyon büyüklüğü arttığında ortalama sürelerin de arttığı görülmektedir. Dolayısıyla gerçek zamanlı optimizasyon problemlerinde, hızın önemli olduğu sistemlerde popülasyon boyunu yaklaşık olarak problem boyutunun 15 katı ($NP=D \times 15$) olarak almanın uygun olduğu anlaşılmıştır.

Şekil 4.9'da Himmelblau optimizasyon test problemi için Rössler tabanlı KFGA'nın farklı popülasyon sayılarında maliyet değişimleri görülmektedir. Şekilde popülasyon büyüklüğünün 10 ve 20 olduğu değerlerde başarımının diğer değerlere göre az olduğu görülmektedir. Diğer popülasyon büyüklüklerinde 20. iterasyondan sonra başarımın arttığı ve yaklaşık olarak aynı olduğu anlaşılmaktadır.



Şekil 4.9. Himmelblau optimizasyon test problemi için Rössler tabanlı KFGA'nın farklı popülasyon sayılarında maliyet değişimleri (F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1, tekrar sayısı=20)

4.8 KFGA Parametrelerinin İncelenmesi

KFGA parametrelerinden skala faktörü (F) ve çaprazlama sabitinin (CR) farklı değerleri için literatürden alınmış olan on adet optimizasyon test fonksiyonu (bkz. Çizelge 4.1) için DE/rand/1/bin stratejisi kullanılmıştır.

4.8.1 Lorenz tabanlı KFGA yapısında skala faktörünün (F) etkisi

Literatürden alınmış olan on adet optimizasyon test fonksiyonu (bkz. Çizelge 4.1) için DE/rand/1/bin FGA stratejisi kullanılarak, Lorenz tabanlı KFGA yapısı ile elde edilen sonuçlar aşağıdaki çizelgelerde (Çizelge 4.19 - 4.23) verilmiştir. Bu kısımda yapılan optimizasyon uygulamasında skala faktörü (F) için farklı değerler (0.6, 0.8, 1.0, 1.2, 1.4) kullanılarak yukarıda belirtilen strateji için KFGA yapısı 20 defa koşturulmuştur. Şekil 4.10'da Rastrigin optimizasyon test problemi için Lorenz tabanlı KFGA'nın farklı skala değerlerinde (F) maliyet değişimleri görülmektedir.

Çizelge 4.19. Lorenz Tabanlı KFGA Optimizasyon Sonuçları¹ (F=0.6)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,987723	-1,099977;-1,099977	79,7	0,072995	0	0,975597
f2	-18,2381	-0,809758;0,318509	199	0,165579	-19,20850257	0,941642
f3	1,094452	-0,248739;-0,149246	83,25	0,067409	0	1,197825
f4	0,06015	0,929339;0,918723	62,15	0,054597	0	0,003618
f5	0,078821	0,393954;0,393954	199	0,18294	0,060247218	0,000345
f6	-0,94775	-6,281078;-5,316462	199	0,173363	-0,96353	0,000249
f7	0,000001	-3,196831;-1,415439	51,95	0,042379	0	1E-12
f8	-731,371	95,396589;95,396590	199	0,181175	-837,9658	11362,39
f9	-0,98339	-0,018693;0,128254	199	0,167652	-1,031628	0,002327
f10	-10,732	-0,444154;-0,781932	199	0,166278	-10,8723	0,019696

Çizelge 4.20. Lorenz Tabanlı KFGA Optimizasyon Sonuçları² (F=0.8)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	0,000000;0,000000	93,2	0,085186	0	1E-12
f2	-18,2381	-0,809758;1,284968	199	0,165951	-19,20850257	0,941642
f3	1,1442	-0,198979;-0,248740	111,3	0,091135	0	1,309194
f4	0,000001	0,999981;0,999946	56,4	0,047117	0	1E-12
f5	0,085996	0,320588;0,393954	199	0,182799	0,060247218	0,000663
f6	-0,95671	-3,541979;-3,703262	199	0,168988	-0,96353	4,66E-05
f7	0	-2,199399;-1,136070	62,85	0,050951	0	0
f8	-737,293	95,396590;131,571274	199	0,179883	-837,9658	10134,97
f9	-1,03163	-0,026953;0,213797	199	0,163147	-1,031628	0
f10	-10,8366	-0,313018;0,157888	199	0,166046	-10,8723	0,001275

Çizelge 4.21. Lorenz Tabanlı KFGA Optimizasyon Sonuçları³ (F=1.0)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,987723	-1,099977;-1,099977	118,6	0,107255	0	0,975597
f2	-19,2085	-2,416545;-2,899378	199	0,166242	-19,20850257	3,22E-13
f3	0,895461	-0,149244;-0,149232	79,65	0,064163	0	0,80185
f4	0,000003	0,999650;0,999308	69,65	0,057644	0	9E-12
f5	0,071647	0,393954;0,467505	199	0,172931	0,060247218	0,00013
f6	-0,9567	-4,507646;-3,705266	199	0,171479	-0,96353	4,66E-05
f7	0,000002	-0,940963;-0,720931	71,05	0,055502	0	4E-12
f8	-743,215	131,571274;131,571274	199	0,175664	-837,9658	8977,694
f9	-1,03163	-0,026953;0,213795	199	0,161453	-1,031628	0
f10	-10,8474	0,314860;-0,313402	199	0,159971	-10,8723	0,000619

Çizelge 4.22. Lorenz Tabanlı KFGA Optimizasyon Sonuçları⁴ (F=1.2)

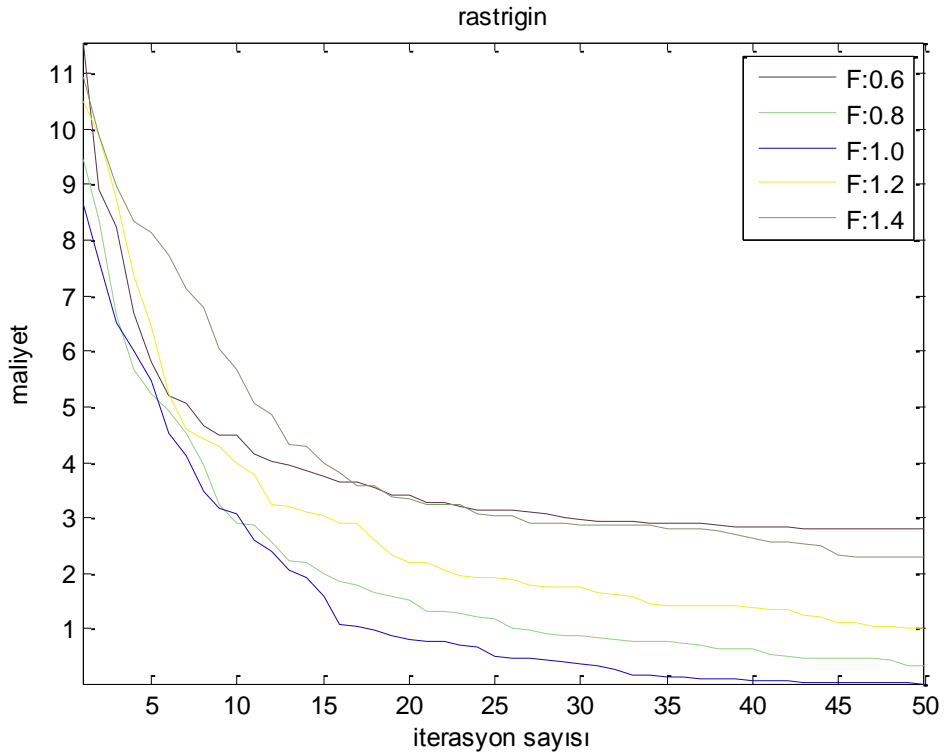
FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	-0,000000;-0,000000	142,5	0,131063	0	1E-12
f2	-19,2085	-2,416515;-0,966443	199	0,171742	-19,20850257	3,22E-13
f3	0,447731	0,000009;-0,149239	120,8	0,099949	0	0,200463
f4	0	0,999958;0,999924	78,1	0,063402	0	0
f5	0,06447	0,467320;0,467320	199	0,18205	0,060247218	1,78E-05
f6	-0,95827	-4,665997;-6,595243	199	0,167801	-0,96353	2,76E-05
f7	0	1,024677;0,030247	90,3	0,068559	0	0
f8	-737,293	95,396590;131,571274	199	0,170625	-837,9658	10134,97
f9	-1,03163	-0,035935;0,285063	199	0,157093	-1,031628	0
f10	-10,7994	-1,097064;-0,469533	199	0,152594	-10,8723	0,00532

Çizelge 4.23. Lorenz Tabanlı KFGA Optimizasyon Sonuçları⁵ (F=1.4)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	0,000000;0,000000	178,15	0,162584	0	1E-12
f2	-19,2085	0,000041;-1,932946	199	0,164151	-19,20850257	3,22E-13
f3	1,159187	-0,200931;-0,248978	178,7	0,141204	0	1,343715
f4	0	0,999988;0,999988	112,55	0,092799	0	0
f5	0,06447	0,467320;0,467320	199	0,17532	0,060247218	1,78E-05
f6	-0,96354	-3,858371;-3,858475	199	0,165659	-0,96353	2,5E-11
f7	0	0,009725;1,218675	106,1	0,077994	0	0
f8	-760,981	167,745961;203,920640	199	0,172342	-837,9658	5926,681
f9	-1,03163	-0,008983;0,071265	199	0,160753	-1,031628	0
f10	-10,8306	-0,470473;-0,469108	199	0,160911	-10,8723	0,001736

Çizelgelerdeki sonuçlar incelendiğinde, skala faktörünün 1.0 değerinde başarımının yüksek olduğu, skala faktörünün 0.6 değerinde diğerlerine göre en az başarıma sahip olduğu görülmektedir.

Şekil 4.10'da skala faktörünün 1.0 olduğu değerinde başarımın diğer skala faktörlerine göre yüksek olduğu görülmektedir. 40. iterasyondan sonra minimum değerine ulaşmaktadır.



Şekil 4.10. Rastrigin optimizasyon test problemi için Lorenz tabanlı KFGA'nın farklı skala değerlerinde (F) maliyet değişimleri (NP=20, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1, tekrar sayısı=20)

4.8.2 Lorenz tabanlı KFGA yapısında çaprazlama sabiti (CR) etkisi

Literatürden alınmış olan on adet optimizasyon test fonksiyonu (bkz. Çizelge 4.1) için DE/rand/1/bin FGA stratejisi kullanılarak, Lorenz tabanlı KFGA yapısı ile elde edilen sonuçlar aşağıdaki çizelgelerde (Çizelge 4.24 - 4.28) verilmiştir. Bu kısımda yapılan optimizasyon uygulamasında CR için farklı değerler (0.3, 0.4, 0.5, 0.6, 0.7) kullanılarak yukarıda belirtilen strateji için KFGA yapısı 20 defa koşturulmuştur.

Şekil 4.11'de Testtubeholder optimizasyon test problemi için Lorenz tabanlı KFGA'nın farklı çaprazlama olasılık değerlerinde (CR) maliyet değişimleri görülmektedir.

Çizelge 4.24. Lorenz Tabanlı KFGA Optimizasyon Sonuçları¹ (CR=0.3)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000027	0,000001;-0,000003	197,9	0,178648	0	7,29E-10
f2	-18,7233	-2,418094;-1,773983	199	0,166	-19,20850257	0,235446
f3	0	-0,000001;0,000002	148,65	0,123565	0	0
f4	0,000351	1,001963;1,004246	194,95	0,15904	0	1,23E-07
f5	0,071646	0,393954;0,467318	199	0,178839	0,060247218	0,00013
f6	-0,95827	-1,772180;3,050828	199	0,169849	-0,96353	2,76E-05
f7	0,001483	-1,395275;-0,161221	150,5	0,118546	0	2,2E-06
f8	-755,059	131,571774;203,919014	199	0,176888	-837,9658	6873,548
f9	-1,03165	0,026938;-0,213790	199	0,16171	-1,031628	4,84E-10
f10	-10,8596	-0,471030;-0,156530	199	0,163777	-10,8723	0,000161

Çizelge 4.25. Lorenz Tabanlı KFGA Optimizasyon Sonuçları² (CR=0.4)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	0,000000;-0,000000	174,3	0,155956	0	1E-12
f2	-19,2085	-4,027566;-0,966557	199	0,201862	-19,20850257	5,73E-11
f3	0,895467	-0,149198;-0,149238	153,5	0,123754	0	0,801861
f4	0,000023	1,000208;1,000305	143,15	0,117126	0	5,29E-10
f5	0,093172	0,320588;0,320588	199	0,174635	0,060247218	0,001084
f6	-0,95671	-6,597135;-1,612832	199	0,166936	-0,96353	4,66E-05
f7	0,000001	-1,038335;-1,362292	96,85	0,080783	0	1E-12
f8	-772,825	203,920383;240,096845	199	0,178215	-837,9658	4243,365
f9	-1,03163	0,026938;-0,213790	199	0,161705	-1,031628	0
f10	-10,8606	0,156942;-0,313200	199	0,160335	-10,8723	0,000137

Çizelge 4.26. Lorenz Tabanlı KFGA Optimizasyon Sonuçları³ (CR=0.5)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	0,000000;-0,000000	149,5	0,130678	0	1E-12
f2	-18,7233	-0,002128;-0,807202	199	0,163516	-19,20850257	0,235411
f3	0,945209	-0,149243;-0,198998	132,1	0,105004	0	0,89342
f4	0	0,999971;0,999946	83,2	0,067875	0	0
f5	0,06447	0,467320;0,467320	199	0,174473	0,060247218	1,78E-05
f6	-0,95827	-3,701385;-3,701388	199	0,166695	-0,96353	2,76E-05
f7	0,000001	-1,463045;-0,992564	93,55	0,073427	0	1E-12
f8	-725,449	59,221905;95,396589	199	0,170919	-837,9658	12659,95
f9	-1,03163	0,008985;-0,071266	199	0,161303	-1,031628	0
f10	-10,8596	0,157106;-0,470491	199	0,161145	-10,8723	0,000161

Çizelge 4.27. Lorenz Tabanlı KFGA Optimizasyon Sonuçları⁴ (CR=0.6)

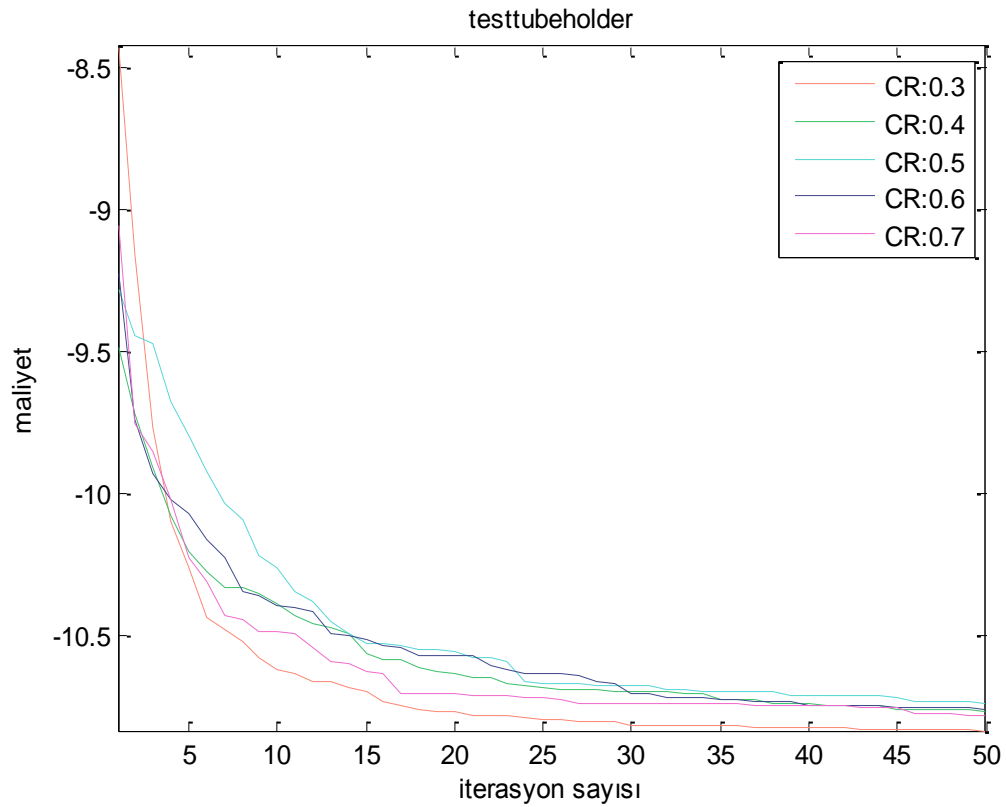
FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	0,000000;0,000000	142,8	0,128058	0	1E-12
f2	-18,7233	-4,835142;-1,773615	199	0,162739	-19,20850257	0,235411
f3	0,497478	-0,099495;-0,000010	119,35	0,095023	0	0,247484
f4	0	1,000123;1,000250	87,35	0,071202	0	0
f5	0,06447	0,467320;0,467320	199	0,172707	0,060247218	1,78E-05
f6	-0,95827	-2,736767;-5,630628	199	0,161956	-0,96353	2,76E-05
f7	0	-1,250684;-1,177404	95,25	0,073259	0	0
f8	-772,825	203,920642;240,095326	199	0,173101	-837,9658	4243,364
f9	-1,03163	0,008984;-0,071266	199	0,159671	-1,031628	0
f10	-10,8584	0,156677;-0,157708	199	0,158677	-10,8723	0,000192

Çizelge 4.28. Lorenz Tabanlı KFGA Optimizasyon Sonuçları⁵ (CR=0.7)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	-0,000000;-0,000000	142,5	0,131063	0	1E-12
f2	-19,2085	-2,416515;-0,966443	199	0,171742	-19,20850257	3,22E-13
f3	0,447731	0,000009;-0,149239	120,8	0,099949	0	0,200463
f4	0	0,999958;0,999924	78,1	0,063402	0	0
f5	0,06447	0,467320;0,467320	199	0,18205	0,060247218	1,78E-05
f6	-0,95827	-4,665997;-6,595243	199	0,167801	-0,96353	2,76E-05
f7	0	1,024677;0,030247	90,3	0,068559	0	0
f8	-737,293	95,396590;131,571274	199	0,170625	-837,9658	10134,97
f9	-1,03163	-0,035935;0,285063	199	0,157093	-1,031628	0
f10	-10,7994	-1,097064;-0,469533	199	0,152594	-10,8723	0,00532

Çizelgelerdeki sonuçlar incelendiğinde, çaprazlama olasılık değerinin 0.3 değeri için başarımının yüksek olduğu, diğer çaprazlama olasılık değerlerinin birbirlerine yakın başarıma sahip olduğu görülmektedir.

Şekil 4.11'de çaprazlama olasılık değerinin 0.3 olduğu değerlerde başarımın diğer skala faktörlerine göre yüksek olduğu görülmektedir. 40. iterasyondan sonra minimum değerine ulaşmaktadır.



Şekil 4.11. Testtubeholder optimizasyon test problemi için Lorenz tabanlı KFGA'nın farklı Çaprazlama olasılık değerlerinde (CR) maliyet değişimleri (NP=20, F=1.2, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1, tekrar sayısı=20)

4.8.3 Rössler tabanlı KFGA yapısında skala faktörünün (F) etkisi

Literatürden alınmış olan on adet optimizasyon test fonksiyonu (bkz. Çizelge 4.1) için DE/rand/1/bin FGA stratejisi kullanılarak, Rössler tabanlı KFGA yapısı ile elde edilen sonuçlar aşağıdaki çizelgelerde verilmiştir. Bu kısımda yapılan optimizasyon uygulamasında F için farklı değerler (0.6, 0.8, 1.0, 1.2, 1.4) kullanılarak yukarıda belirtilen strateji için KFGA yapısı 20 defa koşturulmuştur.

Şekil 4.12'de Holdertable optimizasyon test problemi için Rössler tabanlı KFGA'nın skala faktörü değerlerinde (F) maliyet değişimleri görülmektedir.

Çizelge 4.29. Rössler Tabanlı KFGA Optimizasyon Sonuçları¹ (F=0.6)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	-0,000000;0,000000	67,25	0,062328	0	1E-12
f2	-3,37055	-0,216216;-0,683774	199	0,165972	-19,20850257	250,8408
f3	0,298488	-0,149254;-0,149247	58,95	0,046903	0	0,089095
f4	0,130051	0,715923;0,558382	185,55	0,156995	0	0,016913
f5	0,06447	0,467320;0,467320	199	0,174655	0,060247218	1,78E-05
f6	-0,69778	-1,475114;-1,151312	199	0,170525	-0,96353	0,070623
f7	1,424525	-0,268686;-0,499172	72,65	0,06121	0	2,029271
f8	-454,999	56,610419;56,610419	199	0,17499	-837,9658	146663,9
f9	-1,03159	0,044235;-0,356103	199	0,163137	-1,031628	1,68E-09
f10	-10,8673	-0,157022;-0,470949	199	0,164238	-10,8723	2,45E-05

Çizelge 4.30. Rössler Tabanlı KFGA Optimizasyon Sonuçları² (F=0.8)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	0,000000;0,000000	82,55	0,077475	0	1E-12
f2	-10,3883	-1,251755;-1,479289	199	0,164208	-19,20850257	77,79677
f3	0,000001	-0,000007;-0,000027	43,1	0,035094	0	1E-12
f4	0	0,999764;0,999537	59,35	0,049147	0	0
f5	0,06447	0,467320;0,467320	199	0,176969	0,060247218	1,78E-05
f6	-0,81993	-2,097383;-4,026616	199	0,163786	-0,96353	0,020622
f7	0	-0,050671;-0,377446	64,35	0,04904	0	0
f8	-607,004	112,530579;76,355895	199	0,173922	-837,9658	53343,12
f9	-1,03163	0,035937;-0,285063	199	0,159263	-1,031628	0
f10	-10,8654	-0,628048;-0,470949	199	0,160842	-10,8723	4,81E-05

Çizelge 4.31. Rössler Tabanlı KFGA Optimizasyon Sonuçları³ (F=1.0)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	-0,000000;0,000000	101,1	0,092055	0	1E-12
f2	-12,5244	-2,394777;-0,188824	199	0,162777	-19,20850257	44,67733
f3	0,298488	-0,149260;-0,149244	76,4	0,063247	0	0,089095
f4	0	1,000051;1,000099	61,9	0,052347	0	0
f5	0,06447	0,467320;0,467320	199	0,175498	0,060247218	1,78E-05
f6	-0,80467	0,460165;0,130600	199	0,164869	-0,96353	0,025236
f7	0	1,838950;0,302046	66,85	0,054866	0	0
f8	-707,679	239,459758;311,809126	199	0,175357	-837,9658	16974,58
f9	-1,03163	0,008986;-0,071265	199	0,159551	-1,031628	0
f10	-10,8693	-0,628098;-0,470899	199	0,161462	-10,8723	8,83E-06

Çizelge 4.32. Rössler Tabanlı KFGA Optimizasyon Sonuçları⁴ (F=1.2)

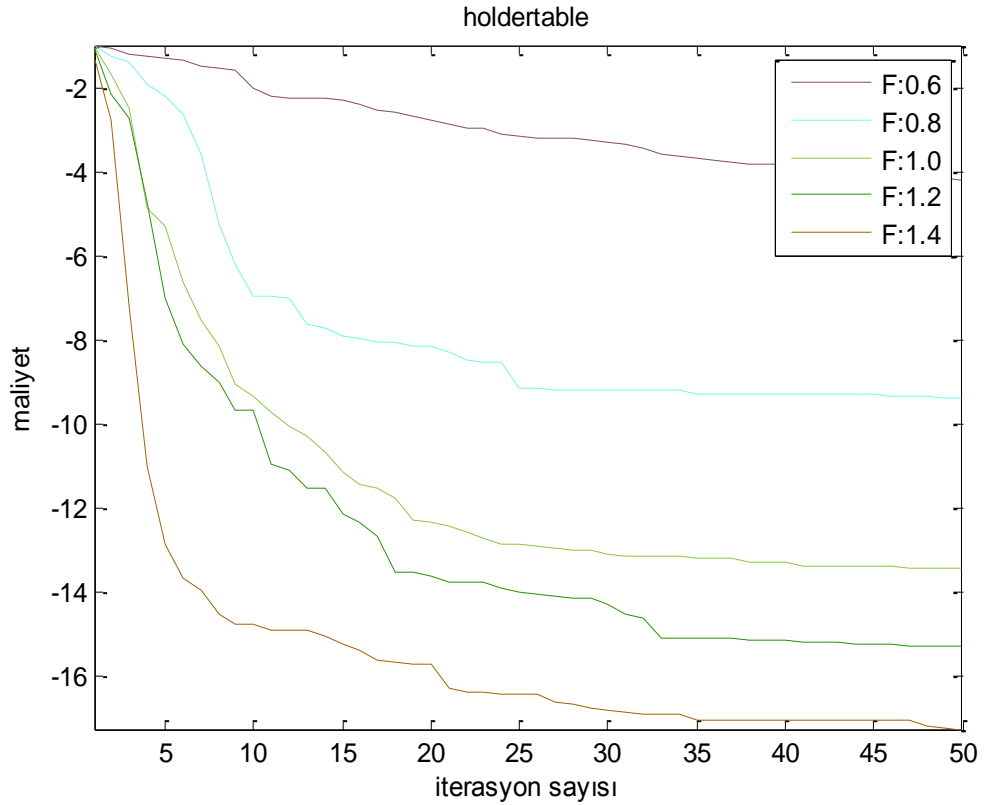
FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	-0,000000;0,000000	128	0,116546	0	1E-12
f2	-16,8849	-0,455163;1,449692	199	0,164648	-19,20850257	5,398915
f3	0	-0,000004;-0,000006	73,05	0,058943	0	0
f4	0	1,000058;1,000120	64,75	0,051924	0	0
f5	0,06447	0,467320;0,467320	199	0,173786	0,060247218	1,78E-05
f6	-0,81762	-0,986759;-0,986761	199	0,162124	-0,96353	0,021291
f7	0	-0,214300;0,128188	98,05	0,077524	0	0
f8	-743,212	302,924347;339,099031	199	0,17758	-837,9658	8978,312
f9	-1,03163	0,026952;-0,213796	199	0,161074	-1,031628	0
f10	-10,8636	-0,309027;-0,314096	199	0,160231	-10,8723	7,56E-05

Çizelge 4.33. Rössler Tabanlı KFGA Optimizasyon Sonuçları⁵ (F=1.4)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	-0,000000;0,000000	155,65	0,137068	0	1E-12
f2	-19,2083	-2,417360;1,934076	199	0,16378	-19,20850257	5,98E-08
f3	0	0,000005;0,000003	85,9	0,066357	0	0
f4	0,000001	1,000021;1,000033	88,85	0,072552	0	1E-12
f5	0,06447	0,467320;0,467320	199	0,172404	0,060247218	1,78E-05
f6	-0,87188	-1,951293;-4,845262	199	0,165123	-0,96353	0,008399
f7	0,000001	1,983094;1,207496	117,4	0,090793	0	1E-12
f8	-719,523	311,809139;311,809142	199	0,17007	-837,9658	14028,68
f9	-1,03163	-0,035936;0,285062	199	0,158554	-1,031628	0
f10	-10,8632	0,313660;0,317442	199	0,160768	-10,8723	8,26E-05

Çizelgelerdeki sonuçlar incelendiğinde, skala faktörünün 1.4 değerinde başarımının yüksek olduğu, skala faktörünün 0.6 değerinde diğerlerine göre en az başarıma sahip olduğu görülmektedir.

Şekil 4.12’de skala faktörünün 1.4 olduğu değerinde başarımın diğer skala faktörlerine göre yüksek olduğu, 0.6 olduğu değerinde düşük olduğu görülmektedir. 45. iterasyondan sonra minimum değerine ulaşmaktadır



Şekil 4.12. Holdertable optimizasyon test problemi için Rössler tabanlı KFGA'nın farklı Skala değerlerinde (F) maliyet değişimleri (NP=20, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1, tekrar sayısı=20)

4.8.4 Rössler tabanlı KFGA yapısında çaprazlama sabiti (CR) etkisi

Literatürden alınmış olan on adet optimizasyon test fonksiyonu (bkz. Çizelge 4.1) için DE/rand/1/bin FGA stratejisi kullanılarak, Rössler tabanlı KFGA yapısı ile elde edilen sonuçlar aşağıdaki çizelgelerde (Çizelge 4.34 - 4.38) verilmiştir. Bu kısımda yapılan optimizasyon uygulamasında CR için farklı değerler (0.3, 0.4, 0.5, 0.6, 0.7) kullanılarak yukarıda belirtilen strateji için KFGA yapısı 20 defa koşturulmuştur.

Şekil 4.13'de Guinta optimizasyon test problemi için Rössler tabanlı KFGA'nın farklı Çaprazlama olasılık değerlerinde (CR) maliyet değişimleri görülmektedir.

Çizelge 4.34. Rössler Tabanlı KFGA Optimizasyon Sonuçları¹ (CR=0.3)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000005	-0,000001;0,000000	183,75	0,168741	0	2,5E-11
f2	-15,8237	0,983469;1,287068	199	0,169372	-19,20850257	11,45722
f3	0,049748	-0,049739;0,000010	121,7	0,09834	0	0,002475
f4	0,000004	0,999716;0,999403	183,25	0,168224	0	1,6E-11
f5	0,06447	0,467320;0,467320	199	0,187161	0,060247218	1,78E-05
f6	-0,82324	0,292033;-0,672559	199	0,172796	-0,96353	0,019681
f7	0,000244	2,826447;1,286760	180,85	0,141674	0	5,95E-08
f8	-752,095	339,098947;294,039791	199	0,180283	-837,9658	7373,749
f9	-1,03163	0,008979;-0,071259	199	0,165731	-1,031628	0
f10	-10,8693	-0,156940;-0,470932	199	0,166904	-10,8723	8,83E-06

Çizelge 4.35. Rössler Tabanlı KFGA Optimizasyon Sonuçları² (CR=0.4)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	-0,000000;0,000000	140,75	0,125479	0	1E-12
f2	-17,7791	-1,755355;1,446548	199	0,167481	-19,20850257	2,04332
f3	0,24874	-0,149227;-0,099506	100,45	0,080771	0	0,061872
f4	0	1,000019;1,000045	129,55	0,103618	0	0
f5	0,06447	0,467320;0,467320	199	0,175136	0,060247218	1,78E-05
f6	-0,86626	-2,265467;0,628311	199	0,166257	-0,96353	0,009462
f7	0,000001	2,477949;1,728296	136,2	0,110183	0	1E-12
f8	-698,796	275,634440;284,519220	199	0,178526	-837,9658	19368,28
f9	-1,03163	-0,008975;0,071272	199	0,165927	-1,031628	0
f10	-10,8693	-0,785217;-0,470950	199	0,165094	-10,8723	8,83E-06

Çizelge 4.36. Rössler Tabanlı KFGA Optimizasyon Sonuçları³ (CR=0.5)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	0,000000;0,000000	127,35	0,116761	0	1E-12
f2	-16,6811	1,434479;-1,791570	199	0,165971	-19,20850257	6,387915
f3	0,397984	-0,198987;-0,198992	99,95	0,079861	0	0,158391
f4	0,000001	0,999809;0,999624	106,2	0,086577	0	1E-12
f5	0,06447	0,467320;0,467320	199	0,175294	0,060247218	1,78E-05
f6	-0,83259	-1,637268;-0,186087	199	0,168242	-0,96353	0,017145
f7	0,000001	1,102573;0,158570	113,15	0,089056	0	1E-12
f8	-701,757	275,634442;239,459758	199	0,176177	-837,9658	18552,74
f9	-1,03163	-0,000000;-0,000000	199	0,16154	-1,031628	0
f10	-10,8683	0,000158;-0,627930	199	0,15865	-10,8723	1,57E-05

Çizelge 4.37. Rössler Tabanlı KFGA Optimizasyon Sonuçları⁴ (CR=0.6)

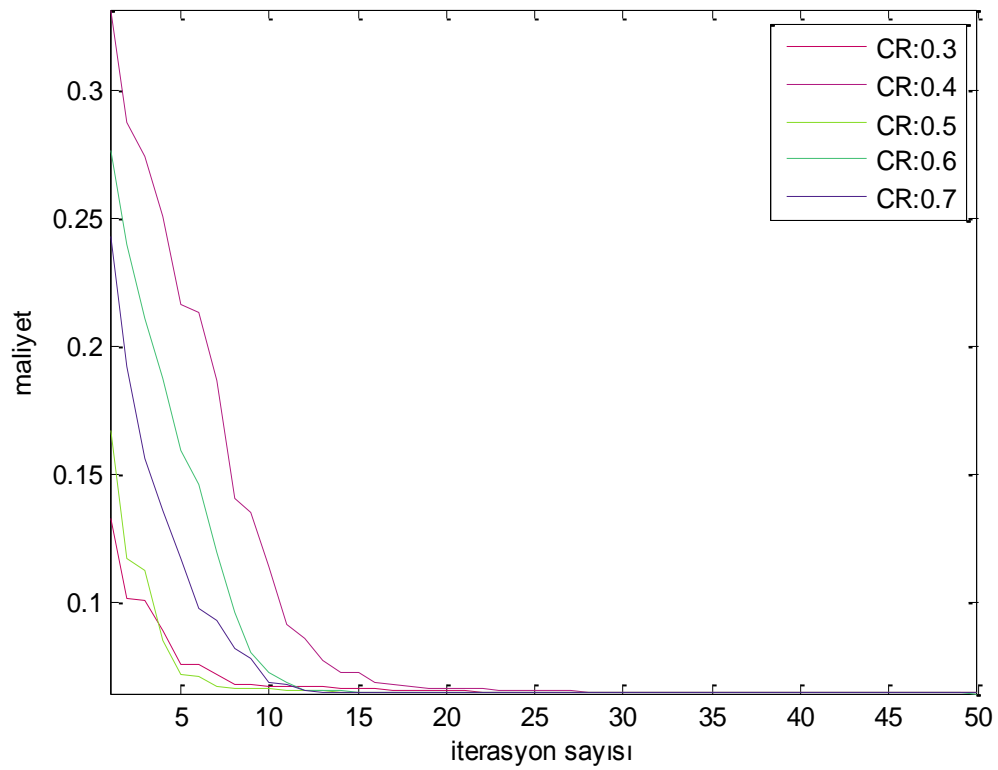
FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	0,000000;0,000000	130	0,116924	0	1E-12
f2	-14,7846	-1,568962;-5,657505	199	0,164862	-19,20850257	19,57066
f3	0,298488	-0,149242;-0,149255	93,5	0,073603	0	0,089095
f4	0,000001	0,999944;0,999892	80,3	0,063113	0	1E-12
f5	0,06447	0,467320;0,467320	199	0,172957	0,060247218	1,78E-05
f6	-0,86626	-1,300915;0,628318	199	0,165232	-0,96353	0,009462
f7	0,000001	1,451296;-0,282772	105,8	0,085342	0	1E-12
f8	-731,368	266,749663;302,924347	199	0,171325	-837,9658	11363,09
f9	-1,03163	-0,035936;0,285063	199	0,159943	-1,031628	0
f10	-10,8656	0,153365;-0,469774	199	0,15834	-10,8723	4,52E-05

Çizelge 4.38. Rössler Tabanlı KFGA Optimizasyon Sonuçları⁵ (CR=0.7)

FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre(sn)	Global Sonuç	MSE
f1	0,000001	-0,000000;0,000000	128	0,116546	0	1E-12
f2	-16,8849	-0,455163;1,449692	199	0,164648	-19,20850257	5,398915
f3	0	-0,000004;-0,000006	73,05	0,058943	0	0
f4	0	1,000058;1,000120	64,75	0,051924	0	0
f5	0,06447	0,467320;0,467320	199	0,173786	0,060247218	1,78E-05
f6	-0,81762	-0,986759;-0,986761	199	0,162124	-0,96353	0,021291
f7	0	-0,214300;0,128188	98,05	0,077524	0	0
f8	-743,212	302,924347;339,099031	199	0,17758	-837,9658	8978,312
f9	-1,03163	0,026952;-0,213796	199	0,161074	-1,031628	0
f10	-10,8636	-0,309027;-0,314096	199	0,160231	-10,8723	7,56E-05

Çizelgelerdeki sonuçlar incelendiğinde, çaprazlama olasılık değerinin 0.5 değerinde başarımının yüksek olduğu görülmektedir.

Şekil 4.13'te çaprazlama olasılık değerinin 0.5 olduğu değerlerde başarımın diğer skala faktörlerine göre yüksek olduğu görülmektedir. 10. iterasyondan sonra minimum değerine ulaşmaktadır.

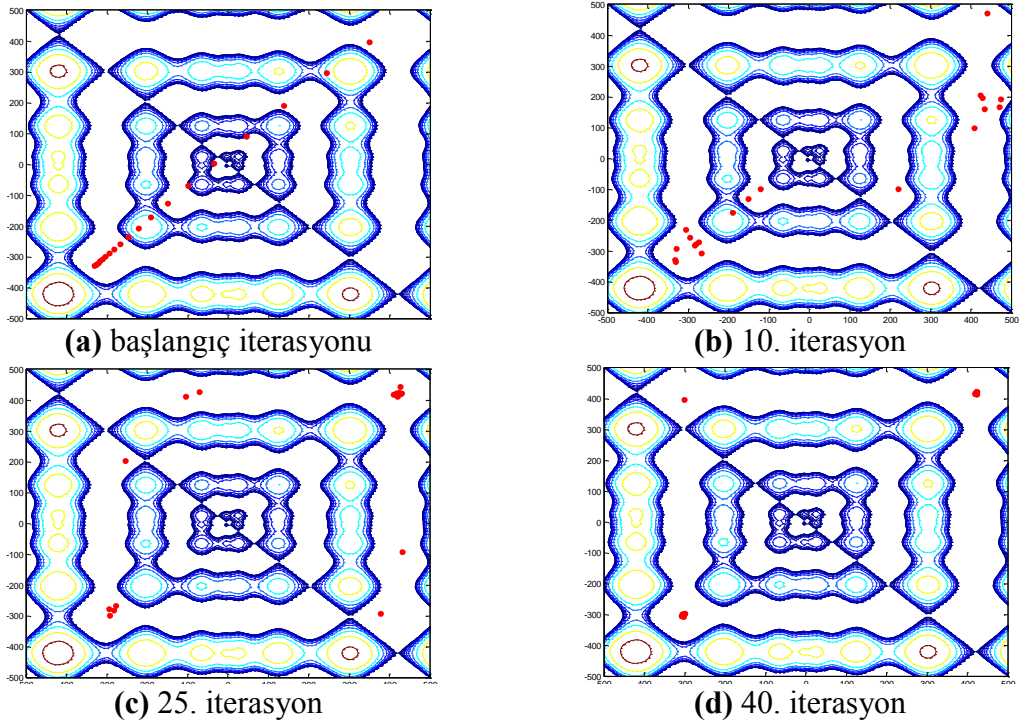


Şekil 4.13. Guinta optimizasyon test problemi için Rössler tabanlı KFGA'nın farklı Çaprazlama olasılık değerlerinde (CR) maliyet değişimleri (NP=20, F=1.2, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1, tekrar sayısı=20)

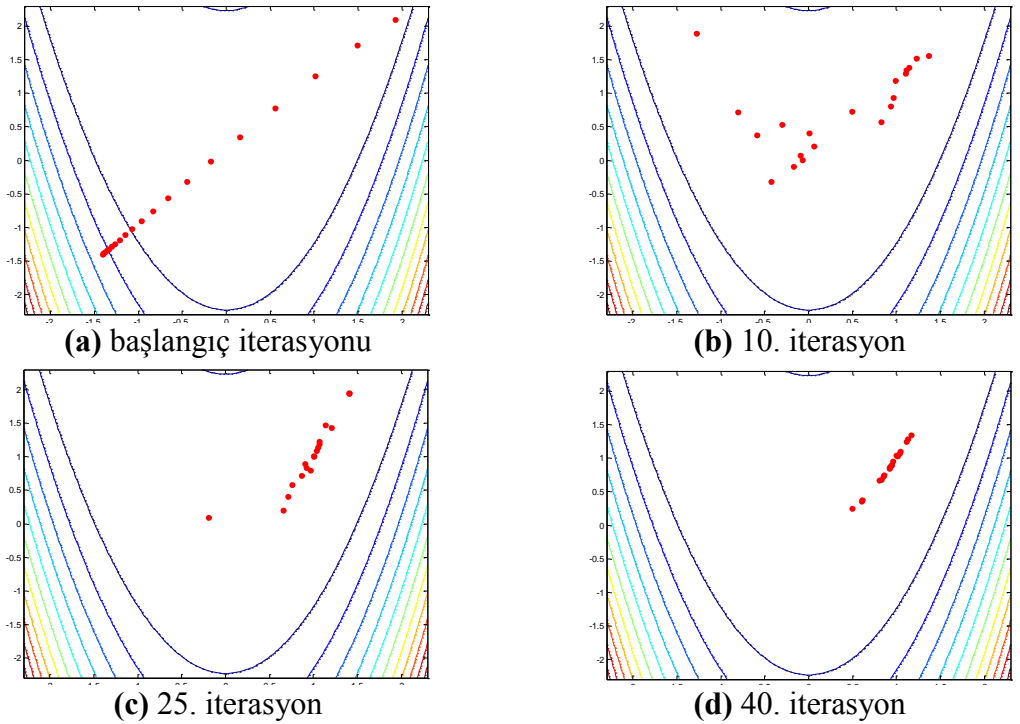
4.9 KFGA Yapısında Çeşitlilik (Diversity) İncelenmesi

Optimizasyon problemlerinin çözümünde, olurlu çözüm bölgesinde, adayların birbirine çok yakın olmaması çeşitliliği (diversity) gösteren bir unsurdur. Çeşitlilik her iterasyon sonrası üretilen yeni aday çözümler yardımı ile lokal çözümlerden kurtulmayı sağlar. Dolayısıyla çözümün belirli noktalarda takılmasını engeller. Çeşitlilik, optimizasyonun başarımını gösteren bir faktördür.

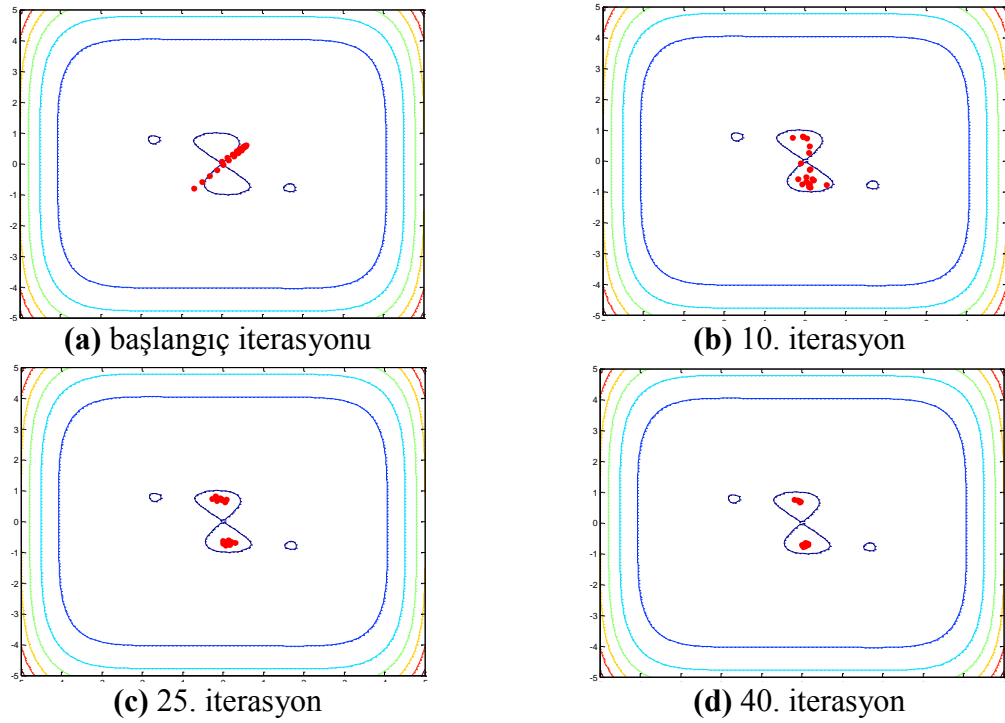
Aşağıdaki şekillerde (Şekil 4.14-4.17) dört farklı optimizasyon test fonksiyonu için başlangıç iterasyonunda, 10, 25 ve 40. iterasyonlarda popülasyon içerisindeki aday çözümlerin dağılımları görülmektedir. Başlangıç iterasyonları ve 10. iterasyonlarda adayların dağılımlarının çeşitli noktalarda yer aldığı ve bir nokta üzerinde toplanmadığı görülmektedir. 25. iterasyonda aday çözümlerin global çözüm noktasına/noktalarına doğru hareket ettiği; ama bunun yanı sıra çeşitliliğin de devam ettiği görülmektedir. 40. iterasyon sonunda ise adayların belirli noktalarda kümelenmesine rağmen kısmen çeşitliliğin sağlandığı görülmektedir.



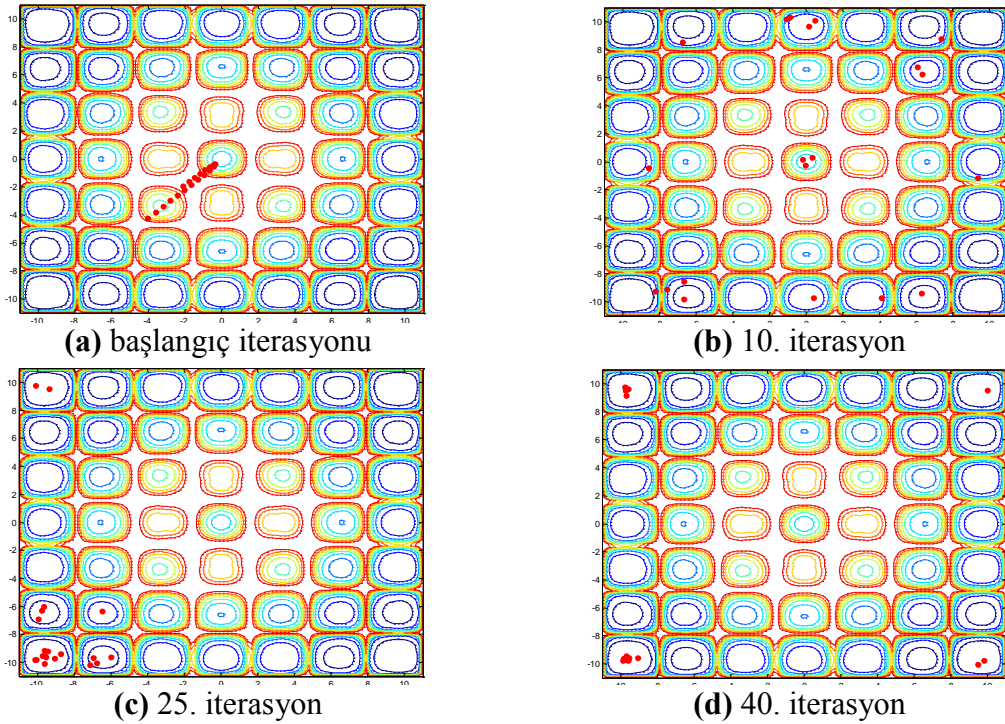
Şekil 4.14. Schwefel optimizasyon test problemi için Lorenz tabanlı KFGA'nın farklı iterasyonlardaki popülasyonlarının değişimi (NP=20, F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1)



Şekil 4.15. Rosenbrock optimizasyon test problemi için Lorenz tabanlı KFGA'nın farklı iterasyonlardaki popülasyonlarının değişimi (NP=20, F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1)



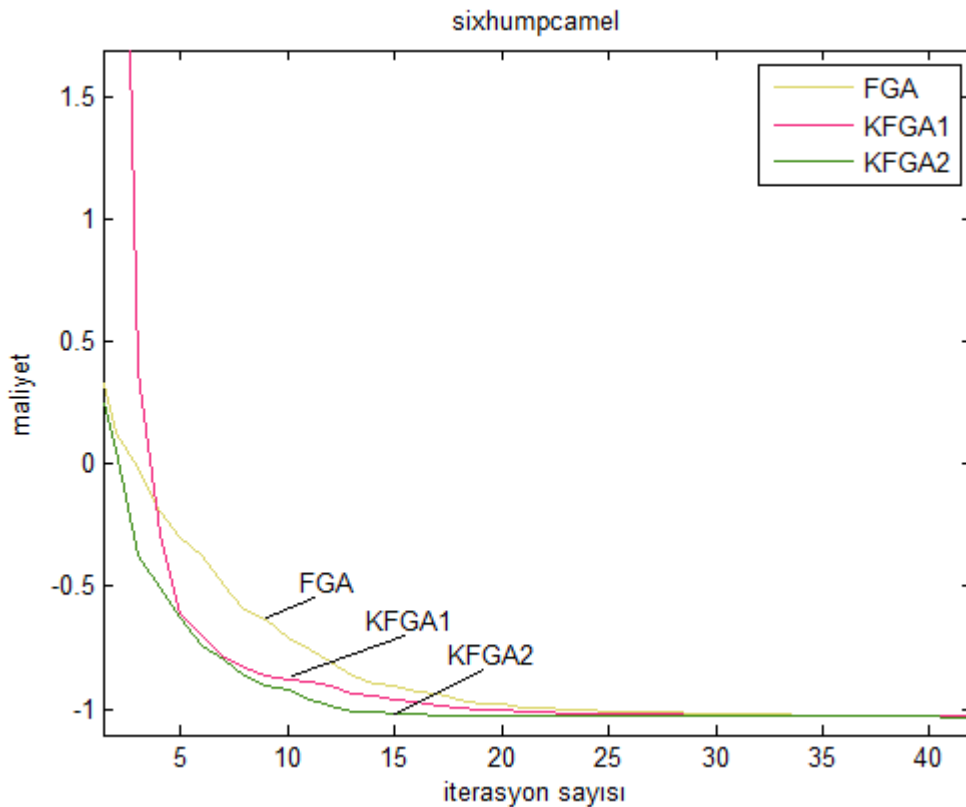
Şekil 4.16. Sixhumpcamel optimizasyon test problemi için Rössler tabanlı KFGA'nın farklı iterasyonlardaki popülasyonlarının değişimi (NP=20, F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1)



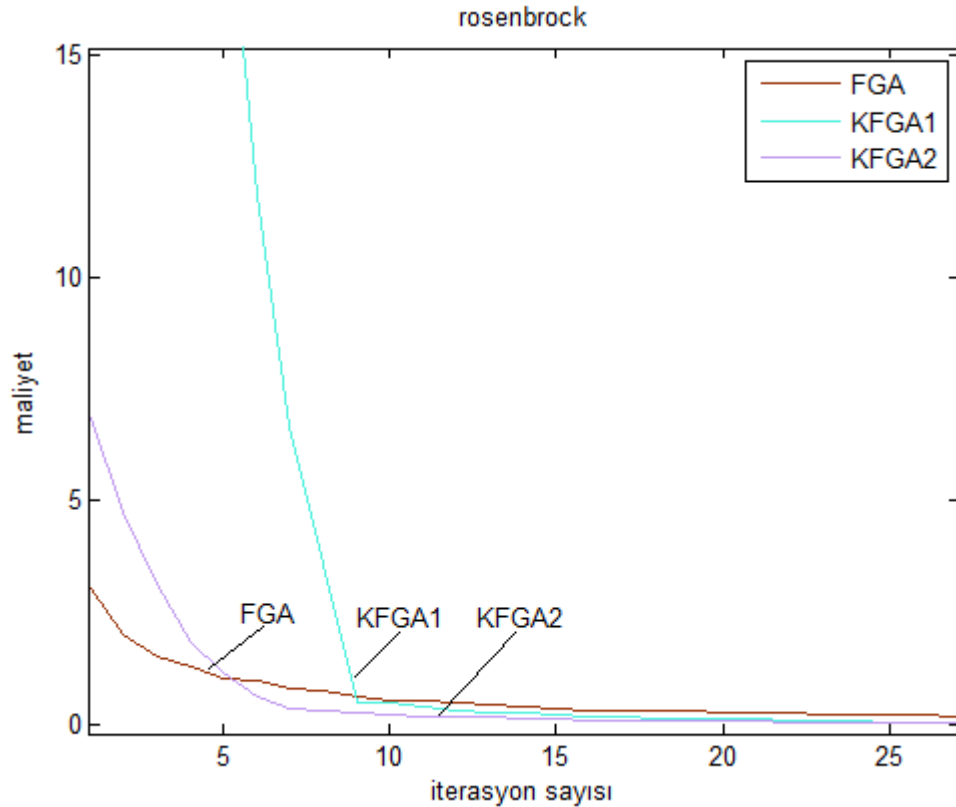
Şekil 4.17. Penholder optimizasyon test problemi için Rössler tabanlı KFGA'nın farklı iterasyonlardaki popülasyonlarının değişimi (NP=20, F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1)

4.10 Klasik FGA ile KFGA Performanslarının Karşılaştırılması

Önerilen iki KFGA yapıları ile klasik FGA yapısının performanslarını karşılaştırmak için literatürden alınmış olan on adet optimizasyon test fonksiyonu (bkz. Çizelge 4.1) ile DE/rand/1/bin FGA stratejisi kullanılarak algoritmalar 50'şer defa koşturulmuştur. Şekil 4.18'de Sixhumpcamel optimizasyon test problemi için Lorenz tabanlı FGA (KFGA1), Rössler tabanlı FGA (KFGA2) yapıları ile klasik FGA yapısının ortalama maliyet değişim sonuçları bir arada gösterilmiştir. Şekil 4.19'da ise Rosenbrock optimizasyon test problemi için Lorenz tabanlı FGA (KFGA1), Rössler tabanlı FGA (KFGA2) yapıları ile klasik FGA yapısının ortalama maliyet değişim sonuçları bir arada gösterilmiştir. Çizelge 4.38'de tüm test fonksiyonları için KFGA yapıları ve FGA yapısının optimizasyon sonuçları verilmiştir.



Şekil 4.18. Sixhumpcamel optimizasyon test problemi için FGA ile Lorenz (KFGA1) ve Rössler tabanlı (KFGA2) yapılarının sonuçları (NP=20, F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1, tekrar sayısı=50)



Şekil 4.19. Rosenbrock optimizasyon test problemi için FGA ile Lorenz (KFGA1) ve Rössler tabanlı (KFGA2) yapılarının sonuçları (NP=20, F=1.2, CR=0.7, itermax=200, $\Delta=1e-6$, strateji:DE/rand/1, tekrar sayısı=50)

Çizelge 4.38. Klasik FGA ile Lorenz tabanlı KFGA (KFGA1) ve Rössler tabanlı KFGA (KFGA2) performanslarının karşılaştırılması

FN	Global Sonuç	FGA				KFGA1				KFGA2			
		Ortalama Sonuç	Ort. İter.	Ortalama Süre(sn)	MSE	Ortalama Sonuç	Ort. İter.	Ortalama Süre(sn)	MSE	Ortalama Sonuç	Ort. İter.	Ortalama Süre(sn)	MSE
f1	0	0,000001	151,78	0,135307	1E-12	0,000001	140,66	0,123152	1E-12	0,000001	127,36	0,114062	1E-12
f2	-19,2085	-19,2085	199	0,16384	1,87E-13	-19,2085	199	0,161413	1,87E-13	-16,4118	199	0,165994	7,821596
f3	0	0	109,12	0,087826	0	0,557176	125,5	0,096809	0,310445	0,318387	89,1	0,071072	0,10137
f4	0	0,000001	150,02	0,1232	1E-12	0,000001	81,6	0,062344	1E-12	0	67,6	0,055979	0
f5	0,060247	0,06447	199	0,176007	1,78E-05	0,06447	199	0,16659	1,78E-05	0,06447	199	0,1732	1,78E-05
f6	-0,96353	-0,96354	199	0,167491	2,5E-11	-0,96143	199	0,158009	4,41E-06	-0,84998	199	0,165885	0,012895
f7	0	0,000002	153,24	0,120754	4E-12	0,000001	90,6	0,068343	1E-12	0	94,38	0,072846	0
f8	-837,966	-837,966	199	0,172538	6,25E-10	-733,74	199	0,167936	10863,01	-691,097	199	0,175157	21570,44
f9	-1,03163	-1,03163	199	0,158337	0	-1,03163	199	0,15605	0	-1,03163	199	0,159007	0
f10	-10,8723	-10,8719	199	0,157705	1,57E-07	-10,8333	199	0,153035	0,001524	-10,8679	199	0,162556	1,9E-05

Şekil 4.18’de KFGA2 başlangıç iterasyondan yaklaşık 25.iterasyona kadar en iyi başarıma sahip olduğu görülmektedir. FGA, 5. iterasyondan yaklaşık 25.iterasyona kadar en zayıf başarıma sahiptir. Yaklaşık 25. İterasyondan sonra FGA, KFGA1 ve

KFGA2 aynı başarıyı göstermektedir.

Şekil 4.19'da KFGA2, 5 ve 10. İterasyon arasında FGA'dan daha yüksek başarıya ulaşmakta ve diğer iterasyon süresince FGA'dan daha yüksek başarıyla devam etmekte olduğu görülmektedir. 15. iterasyondan sonra KFGA1 ve KFGA2 hemen hemen aynı başarıyla devam etmektedir.

Çizelge 4.38'de yüksek başarı (MSE değerleri küçük), en iyi ortalama iterasyon ve en iyi ortalama süre değerleri **koyu** olarak gösterilmiştir. Çizelgedeki tablo incelendiğinde, ortalama iterasyon değerlerine göre en iyi algoritmanın KFGA2, en kötü algoritmanın ise FGA olduğu görülmektedir. Ortalama süreler baz alınarak karşılaştırma yapıldığında ise en iyi sürelerin KFGA1'e ait olduğu, ikinci en iyi sürelerin ise KFGA2'ye ait olduğu verilmiştir. FGA'nın en iyi sürede sonuca ulaştığı koşturma bulunmamaktadır. MSE değerlerine bakıldığında FGA'nın en iyi başarıya sahip olduğu, KFGA2'nin ikinci iyi başarıya sahip olduğu, en az başarımın KFGA1'de olduğu görülmektedir.

Başarımlarını ortalama süre ile birlikte irdelediğimizde MSE değerlerine göre FGA'nın daha başarılı gözükmesine rağmen ortalama sürelerde FGA'nın hiç en iyi değere sahip olmamasının yanı sıra bazı durumlarda en iyi ortalama süresinin iki kat süresine ulaşması gerçek zamanlı sistemlerde çok yavaş çözüme ulaşmasına sebep olacaktır.

5. SONUÇ

Yapılan bu tez çalışmasında optimizasyon problemlerinin çözümünde en çok kullanılan evrimsel algoritmalarından FGA'nın temelini oluşturan rastgele sayı üretim süreci yerine, kaotik sistem tabanlı sayı üretici geliştirilerek, kaotik tabanlı FGA yapısı önerilmiştir. Kaotik sistem olarak Lorenz ve Rössler çekici sistemleri kullanılmıştır. Önerilen kaotik tabanlı FGA'nın performansı literatürden alınan on optimizasyon test problemleri için koşturularak FGA'nın farklı stratejileri, farklı parametreleri için benzetim çalışmaları yapılmıştır. Klasik FGA ile önerilen KFGA (Kaotik Tabanlı Diferansiyel Evrimsel Gelişim Algoritması)'nın karşılaştırılması yapılmış, önerilen kaotik yapının avantaj ve dezavantajları vurgulanmıştır.

Önerilen KFGA yapılarının performanslarının incelenmesi için literatürden on optimizasyon test problemleri alınarak benzetim çalışmaları yapılmıştır. Lorenz tabanlı KFGA ve Rössler tabanlı KFGA, DE/rand/1/bin FGA stratejisi için parametre değerleri NP (Popülasyon değeri) için 20, maksimum iterasyon sayısı için 200, F (skala faktörü) için 1.2 ve CR (çaprazlama sabiti) için 0.7 alınarak 50'şer defa koşturulmuştur. Lorenz tabanlı KFGA yapısının koşturma sonunda elde edilen ortalama sürelerine bakıldığında, genel olarak sürelerin kısa olduğu ve gerçek zamanlı uygulamalar için önerilen algoritmanın kullanılabilir olduğu görülmüştür. Rössler tabanlı KFGA yapısının koşturma sonunda Holdertable, Rastrigin ve Schwefel test fonksiyonlarının dışında başarımının yüksek olduğu görülmüştür.

Sonraki adımlarda parametre değerleri değiştirilerek başarımlar üzerindeki etkileri gözlenmiştir. Bu parametre değişimleri; farklı stratejiler kullanmak, NP (Popülasyon sayısı) değerini değiştirmek, farklı F (skala faktörü) değeri ve farklı CR (çaprazlama sabiti) değeri kullanmak şeklindedir. On farklı strateji için Lorenz tabanlı KFGA ve Rössler tabanlı KFGA, NP için 20, maksimum iterasyon sayısı için 200, F için 1.2 ve CR için 0.7 alınarak 20'şer defa koşturulmuştur.

Lorenz tabanlı KFGA için tüm stratejilerde 20 koşturma sonunda elde edilen ortalama MSE değerleri incelendiğinde, Lorenz tabanlı KFGA için en başarılı stratejilerin ST2 (DE/rand/1/exp) ve ST7 (DE/rand/1/bin) stratejileri olduğu görülmüştür. En kötü başarıma sahip stratejilerin ise ST1 (DE/best/1/exp) ve ST6

(DE/best/1/bin) stratejileri olduğu anlaşılmıştır. Ortalama sürelerle göre en iyi başarımlar binom çaprazlama stratejilerine aittir. Ortalama iterasyon açısından en iyi başarımlar, en az iterasyon değerine sekiz defa ulaşan ST6 (DE/best/1/bin) stratejisidir, en kötü başarımlar ise en az iterasyon değerine beşer defa ulaşan ST7 (DE/rand/1/bin) ve ST9 (DE/best/2/bin) stratejileridir.

Rössler tabanlı KFGA için tüm stratejilerde 20 koşturma sonunda elde edilen ortalama MSE değerleri incelendiğinde, Rössler tabanlı KFGA için en başarılı stratejinin ST8 (DE/rand-to-best/1/bin) olduğu, daha sonra ise ST2 (DE/rand/1/exp), ST4 (DE/best/2/exp), ST7 (DE/rand/1/bin), ST9 (DE/best/2/bin) ve ST10 (DE/rand/2/bin) stratejileri olduğu görülmüştür. En kötü koşturma başarımları ise ST1 (DE/best/1/exp) ve ST6 (DE/best/1/bin) stratejilerinde olduğu görülmüştür. f9 test fonksiyonunda ST6 (DE/best/1/bin) stratejisi hariç bütün stratejilerde en küçük MSE değerlerini aldığı görülmüştür. Ortalama süreye göre en iyi başarımlar, en kısa süre değerlerine göre beş fonksiyonda ulaşan ST7 ve dört fonksiyonda ulaşan ST8 stratejisidir. Ortalama iterasyon açısından en iyi başarımlar, en az iterasyon değerine sekiz defa ulaşan ST1 stratejisidir, en kötü başarımlar ise en az iterasyon değerine iki defa ulaşan ST8 stratejisidir.

Lorenz tabanlı KFGA ve Rössler tabanlı KFGA, farklı NP değerleri için DE/rand/1/bin stratejisi kullanılarak maksimum iterasyon sayısı için 200, F için 1.2 ve CR için 0.7 alınarak 20'şer defa koşturulmuştur. Hem Lorenz tabanlı KFGA için hem Rössler tabanlı KFGA için popülasyon büyüklüğünün 30 değerinden sonraki değerlerde optimizasyon başarımlarını sağladığı ve sabitlendiği anlaşılmıştır. Beklenildiği üzere popülasyon büyüklüğü arttırıldığında ortalama sürelerin de arttığı görülmüştür. Dolayısıyla gerçek zamanlı optimizasyon problemlerinde, hızın önemli olduğu sistemlerde popülasyon boyunu yaklaşık olarak problem boyutunun 15 katı ($NP=D \times 15$) olarak almanın uygun olduğu anlaşılmıştır.

Lorenz tabanlı KFGA ve Rössler tabanlı KFGA, farklı F değerleri için DE/rand/1/bin stratejisi kullanılarak NP için 20, maksimum iterasyon sayısı için 200 ve CR için 0.7 alınarak 20'şer defa koşturulmuştur. Lorenz tabanlı KFGA'da skala faktörünün 1.0 değerinde başarımlarının yüksek olduğu, skala faktörünün 0.6 değerinde diğerlerine göre en az başarımlara sahip olduğu görülmüştür. Rössler tabanlı KFGA'da ise

skala faktörünün 1.4 değerinde başarımının yüksek olduğu, skala faktörünün 0.6 değerinde diğerlerine göre en az başarıma sahip olduğu görülmüştür.

Lorenz tabanlı KFGA ve Rössler tabanlı KFGA, farklı CR değerleri için DE/rand/1/bin FGA stratejisi kullanılarak NP için 20, maksimum iterasyon sayısı için 200 ve F için 1.2 alınarak 20'şer defa koşturulmuştur. Lorenz tabanlı KFGA'da çaprazlama olasılık değerinin 0.3 değeri için başarımının yüksek olduğu, diğer çaprazlama olasılık değerlerinin birbirlerine yakın başarıma sahip olduğu görülmüştür. Rössler tabanlı KFGA'da ise çaprazlama olasılık değerinin 0.5 değerinde başarımının yüksek olduğu görülmüştür.

Farklı iterasyon sayıları için KFGA yapısında çeşitlilik (diversity) kavramı incelenmiştir. Çeşitlilik incelemesinde dört farklı optimizasyon test fonksiyonu Lorenz tabanlı ve Rössler tabanlı KFGA için başlangıç iterasyonunda, 10, 25 ve 40. iterasyonlarda popülasyon içerisindeki aday çözümlerin dağılımları elde edilmiş, başlangıç iterasyonları ve 10. iterasyonlarda adayların dağılımlarının çeşitli noktalarda yer aldığı ve bir nokta üzerinde toplanmadığı görülmüştür. 25. iterasyonda aday çözümlerin global çözüm noktasına/noktalarına doğru hareket ettiği; ama bunun yanı sıra çeşitliliğin de devam ettiği görülmüştür. 40. iterasyon sonunda ise adayların belirli noktalarda kümelenmesine rağmen kısmen çeşitliliğin sağlandığı görülmüştür.

Son olarak klasik FGA ile Lorenz tabanlı ve Rössler tabanlı KFGA performanslarının karşılaştırılması yapılmıştır. Bu karşılaştırma sonucunda ortalama iterasyon değerlerine göre en iyi başarıma sahip algoritmanın KFGA2, en az başarıma sahip algoritmanın ise FGA olduğu görülmüştür. Ortalama süreler baz alınarak karşılaştırma yapıldığında ise en iyi sürelerin KFGA1'e ait olduğu, ikinci en iyi sürelerin ise KFGA2'ye ait olduğu verilmiştir. FGA'nın en iyi sürede sonuca ulaştığı koşturma bulunmamaktadır. MSE değerlerine bakıldığında FGA'nın en iyi başarıma sahip olduğu, KFGA2'nin ikinci iyi başarıma sahip olduğu, en az başarımın KFGA1'de olduğu görülmüştür.

Başarım değerlerini ortalama süre ile birlikte irdelediğimizde MSE değerlerine göre FGA'nın daha başarılı gözükmesine rağmen ortalama sürelerde FGA'nın en iyi değere sahip olmamasının yanı sıra bazı durumlarda en iyi ortalama süresinin iki kat süresine

ulaşması gerçek zamanlı sistemlerde çok yavaş çözüme ulaşmasına sebep olacağı kanaatine varılmıştır.

Bundan sonraki çalışmalarda bu tezde yapılan Lorenz tabanlı ve Rössler tabanlı KFGA optimizasyon testlerinin ve kıyaslamaların Chua tabanlı ve Mackey-Glass tabanlı KFGA için yapılması ve burada bahsedilen kaotik tabanlı KFGA yapılarının gerçek zamanlı sistemlerde uygulamalarının yapılması planlanmaktadır.

KAYNAKLAR

- Alligood, K.T., Sauer, T.D., Yorke, J.A., "Chaos", New York, 603, (1997).
- Arslan, M., " Diferansiyel evrim algoritması yardımıyla asenkron motor parametrelerinin belirlenmesi", Yüksek Lisans Tezi , **Selçuk Üniversitesi Fen Bilimleri Enstitüsü**, Konya , 2010.
- Baker, G.L., Gollub, J.P., "Chaotic Dynamics an Introduction", **Cambridge University Pres.**, Cambridge, 179, 1990.
- Becerra, R. L., Coello, C. A., "Cultured differential evolution for constrained optimization", **Computer Methods in Applied Mechanics and Engineering**, 195:4303-4322 (2006).
- Bergey, P., Ragsdale, C., "Modified Differential Evolution: A Greedyrandom Strategy For Genetic Recombination", **Omega**, 33 : 255-265 (2005).
- Bingham, D., Surjanovic, S., "Test Functions and Datasets, Optimization Test Functions", <http://www.sfu.ca/~ssurjano/optimization.html>, 2014.
- Brest, J., Greiner, S., Boskovic, B., Mernik, M., & Zumer, V., "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems", **IEEE Trans. Evol. Comput.**, 10(6), 646-657 (2006).
- Canan, S., "Kaos, karmaşıklık bilimi ve yeni bilimsel anlayışlar – 1", <http://www.sinancanan.net.tr/2011/01/kaos-karmasklk-bilimi-ve-yeni-bilimsel-22.html>, 2014
- Çelik. Y., "Optimizasyon problemlerinde bal arıları evlilik optimizasyonu algoritmasının (marriage in honey bee optimization-MBO) performansının geliştirilmesi", Doktora tezi, **Selçuk Üniversitesi Fen Bilimleri Enstitüsü**, Konya, 2013.
- Erdemir, G., "Yapı Sistem Analizi", **Afyon Kocatepe Üniversitesi , Fen bilimleri enstitüsü**, Yapı Eğitimi Bölümü ders sunumu, s.14 (2008).
- Ertokatlı, C. T."Borsa endeks getirilerinin kaotik süreçleri : deneysel bir sına", Doktora Tezi, **Gebze Yüksek Teknoloji Enstitüsü Sosyal Bilimler Enstitüsü**, Gebze, 2013.
- Eser, M., Yüzgeç, U., "Kaotik Tabanlı Diferansiyel (Farksal) Gelişim Algoritması", **ISITES 2014- 2. Uluslararası Mühendislik Ve Bilim Alanında Yenilikçi Teknolojiler Sempozyumu**, Karabük, 2014 (Basım aşamasında).

KAYNAKLAR DİZİNİ (devam ediyor)

- Gleick J., “Kaos” *Tübitak Yayınları*, Gebze, (2008).
- Gündüz, G., “Kargaşa Kaos ve Şekil Oluşumları”, **METU Press**, Ankara, 538-56 (2002).
- Kennedy, M. P. and Kolumban, G., “Digital Communications Using Chaos”, **McGraw Hill**, New York, 477-499 (1999).
- Gürarşlan, G., ” Yeraltısuyu kirletici kaynak yerlerinin ve boşalım geçmişlerinin diferansiyel gelişim algoritması kullanılarak belirlenmesi”, Doktora Tezi, *Pamukkale Üniversitesi Fen Bilimleri Enstitüsü*, Denizli, 2011.
- Hiçdurmaz, B. “Yüksek hızlı dwdm sistemlerinde optik fiberdeki fwm ve ase etkisi altında iletim performansının analizi ve optimizasyonu”, Doktora Tezi, *Uludağ Üniversitesi Fen Bilimleri Enstitüsü*, Bursa, 2013.
- Holmes, PJ., “Poincare celestial mechanics, dynamical-systems theory and chaos”, *Phys. Rep.*, 193(3):138-163 (1990).
- Hrstka, O., Kucerova, A., “Improvements of Real Coded Genetic Algorithms Based On Differential Operators Preventing Premature Convergence”, *Advances in Engineering Software*, 35 : 237–246 (2004).
- İskurt, F.T., “Elektronik sistemlerde kaotik sinyallerin aktif kontrolü”, Yüksek Lisans Tezi, *Sakarya Üniversitesi Fen Bilimleri Enstitüsü*, Sakarya, 2012.
- Kennedy MP., Chua LO., “Van Der Pol and Chaos”, *IEEE Trans. Circuit Syst., CAS-33:974-980* (1986).
- Karaboğa, D., Yapay Zeka Optimizasyon Algoritmaları, *Nobel Yayın Dağıtım*, Ankara, 2011.
- Kavak, N..”Kaotik yapay sinir ağlarının analizi ve sistem modelleme”, Yüksek Lisans Tezi, *Firat Üniversitesi Fen Bilimleri Enstitüsü*, Elazığ, 2009.
- Kennedy, MP.,” Experimental Chaos from Autonomous Electronic Circuits”, *Phil. Trans. R. Soc.*, London, A(353):13-32 (1995).
- Keskintürk, T., “Diferansiyel gelişim algoritması”, *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, Sayı: 9: 85-99 (2006).
- Keunen, R.W.M., Vliegen, J.H.R., Stam, C.J. ve Tavy D.L.J., “Nonlinear Transcranial Doppler Analysis Demonstrates Age-Related Changes of Cerebral Hemodynamics”, *Ultrasound Med. & Biol.*, 22: 383-390, 1996.

KAYNAKLAR DİZİNİ (devam ediyor)

- Koyuncu C.A., “Farksal Gelişim Algoritmasının İncelenmesi ve İşaret Kestiriminde Kullanılması”, Yüksek Lisans Tezi, *Erciyes Üniversitesi Fen Bilimleri Enstitüsü*, Kayseri, 2006.
- Kurban, T., “Video ve gps verilerine dayalı harita üretimi”, Yüksek Lisans Tezi , *Erciyes Üniversitesi Fen Bilimleri Enstitüsü*, Kayseri, 2008.
- Li T., Yorke, J.A.,” Period three implies chaos”, *Amer. Math. Monthly*, 82:985–992 (1975).
- Lorenz, E. N., “Deterministic nonperiodic flow”, *J. Atmos. Sci.*, 20:130–141 (1963).
- Lin, Y.C., Hwang, K.S., Wang, F.S., “A Mixed-Coding Scheme of Evolutionary Algorithms to Solve Mixed-Integer Nonlinear Programming Problems”, *Computers and Mathematics with Applications*, 47, 1295-1307 (2004).
- Mackey, M., Glass, L., “Oscillation and Chaos in Physiological Control Systems”, *Science New Series*, 197:287-289 (1977).
- Muralı, K., Lakshmanan, M., Chua LO.,” Controlling and Synchronization of Chaos in The Simplest Dissipative Nonautonomous Circuit”, *International J. of Bifurcation&Chaos*, 5:563-571 (1995).
- Moon, F.C., “Chaotic Vibrations: An Introduction for Applied Scientists and Engineers”, *John Wiley & Sons*, New York, (1987).
- Pehlivan, İ. “Yeni kaotik sistemler: Elektronik devre gerçeklemeleri, senkronizasyon ve güvenli haberleşme uygulamaları”, Doktora Tezi, *Sakarya Üniversitesi Fen Bilimleri Enstitüsü*, Sakarya, 2007.
- Rahnamayan, S., Tizhoosh, H. R., & Salama, M. M. A., “Opposition-based differential evolution algorithms”, *In Proc. IEEE Congr. Evol. Comput., Vancouver*, BC, Canada, 2010-201 (2006).
- Ruelle, D., “Raslantı ve Kaos” *Tübitak Yayınları*, Gebze, (2010).
- Rössler OE., “An equation for continuous chaos”, *Phys. Lett. A*, 57:397–398 (1976).
- Sun, J., Zhang, Q., Tsang, E., “DE/EDA: A New Evolutionary Algorithm For Global Optimization”, *Information Sciences*, 169: 249-262 (2005).
- Strogatz, S.H., “Nonlinear Dynamics and Chaos with Application to Physics, Biology, Chemistry and Engineering”, *Perseus Books Publishing*, Massachusetts, A.B.D., (1994).

KAYNAKLAR DİZİNİ (devam ediyor)

- Terzi, Ü. “Gezgin satıcı problemi için diferansiyel gelişim algoritması tabanlı bir metasezgisel önerisi”, Doktora Tezi, *Kocaeli Üniversitesi Fen Bilimleri Enstitüsü*, Kocaeli, 2009.
- Tizhoosh, H. R., “Opposition-based learning: A new scheme for machine intelligence”, *in Proc. Int. Conf. Comput. Intell. Modeling Control and Autom.*, Vienna, Austria, 695-701 (2005).
- Uyaroğlu, Y., Gündüz, S. , Yığınç, İ.H. ve Keskin, H., “Kaos Teorisindeki Lorenz Eşitliklerinin Matlab Ve Simulink Ortamında Benzetimi ile Karakterize Edilmesi”, *5. Uluslararası İleri Teknolojiler Sempozyumu (IATS'09)*, Karabük, (2009).
- Uyaroğlu Y., Pehlivan, İ. , “A New Chaotic Attractor From General Lorenz System”, *Turk J Elec Eng & Comp Sci*, 18 (2010).
- Uzun, Y.H.İ., “Makine mühendisliğinde kullanılan optimizasyon tekniklerinin incelenmesi”, Yüksek Lisans Tezi, *Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul, 2006.
- Williams, G.P., “Chaos Theory Tamed”, *Taylor & Francis Publisher*, London, 1997.
- Yardım, F.E., Afacan, E., "Lorenz-Tabanlı Diferansiyel Kaos Kaydırmalı Anahtarlama (DCSK) Modeli Kullanılarak Kaotik Bir Haberleşme Sisteminin Simülasyonu", *Gazi Üniv. Müh. Mim. Fak. Der.*, 25:101-110 (2010).
- Yılmaz, D., Güler, N.F.,” Kaotik Zaman Serilerinin Analizi Üzerine Bir Araştırma”, *Gazi Üniv.Müh. Mim. Fak. Der.*, 21: 759-779 (2006).
- Yılmaz, S., “Yarasa algoritmasının unimodal, multimodal ve kaydırılmış sayısal optimizasyon problemleri (cec05) üzerinde geliştirilmesi”, Yüksek Lisans Tezi, *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü*, Isparta, 2014.
- Yiğit, E.. ”Güç trafolarında ferrozonans olayların kaotik analizi”, Yüksek Lisans Tezi, *Sakarya Üniversitesi Fen Bilimleri Enstitüsü*, Sakarya, 2009.
- Yiğit, N., “Farksal gelişim algoritması kullanılarak adaptif gürültü giderici”, Yüksek Lisans Tezi, *Erciyes Üniversitesi Fen Bilimleri Enstitüsü*, Kayseri, 2007.
- Yüzgeç, U., “Kendinden uyarlanabilir karşıtlık tabanlı farksal gelişim algoritması”, *Bilecik Üniv.*, BAP Sonuç Raporu, *Bilecik*, 2011.
- Zaharie, D., “Critical Values for Control Parameters of Differential Evolution Algorithms”, *Proceedings of Mendel 2002, 8th International Conference on Soft Computing*, Brno,62-67, (2002)

ÖZGEÇMİŞ



Kişisel Bilgiler

Adı Soyadı : Mehmet ESER
Doğum Yeri ve Tarihi : Bozüyük-1974

Eğitim Durumu

Lisans Öğrenimi : Fırat Üniv. TEF Elektronik ve Bilg. Eğt.
Bildiği Yabancı Diller : İngilizce

İş Deneyimi

Çalıştığı Kurumlar : MEB
Dumlupınar Üniv.
Anadolu Üniv.
Bilecik Şeyh Edebali Üniv.

Diğer / Akademik Çalışmalar

- [1] Eser M., Yüzgeç U., "Kaotik Tabanlı Diferansiyel (Farksal) Gelişim Algoritması", *2nd International Symposium on Innovative Technologies in Engineering and Science, ISITES 2014*, s.201-210, Karabük, 18-20 Haziran 2014.
- [2] İmal, N., Eser M., "Bilişim Saldırganlığı, Nedenleri Ve Sınıflandırılması", *3.Ağ Ve Bilgi Güvenliği Ulusal Sempozyumu*, Ankara, 5-6 Şubat 2010
- [3] İmal, N., Eser M., "Programlama Dili Öğrenmedeki Zorluklar ve Çözüm Yaklaşımları", *İv. Elektrik Elektronik Bilgisayar Biyomedikal Mühendislikleri Eğitimi Sempozyumu*, Eskişehir, 22-24 Ekim 2009

İletişim

Adres : Bilecik Üniversitesi, MYO, Bilg.Prog.P.
Gülümbe Kampüsü, BİLECİK
Tel: : 0228.2141609
E-Posta Adresi : mehmet.eser@bilecik.edu.tr

Tarih: / /

İmza