KAOTİK KRİL SÜRÜSÜ OPTİMİZASYON YÖNTEMİNİN GELİŞTİRİLMESİ VE SÜSPANSİYON PARÇASININ OPTİMİZASYONU

T.C.
BURSA ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KAOTİK KRİL SÜRÜSÜ OPTİMİZASYON YÖNTEMİNİN GELİŞTİRİLMESİ VE SÜSPANSİYON PARÇASININ OPTİMİZASYONU

Halil BİLAL
0000-0003-0281-9397

Prof. Dr. Ferruh ÖZTÜRK
(Danışman)

DOKTORA TEZİ
OTOMOTİV MÜHENDİSLİĞİ ANABİLİM DALI

BURSA – 2021

Her Hakkı Saklıdır

ÖZET

Doktora Tezi

KAOTİK KRİL SÜRÜSÜ OPTİMİZASYON YÖNTEMİNİN GELİŞTİRİLMESİ VE SÜSPANSİYON PARÇASININ OPTİMİZASYONU

Halil BİLAL

Bursa Uludağ Üniversitesi
Fen Bilimleri Enstitüsü
Otomotiv Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Ferruh ÖZTÜRK

Hayatın her alanında karşılaştığımız optimizasyon problemlerinin çözümü için geliştirilmiş çok çeşitli optimizasyon yöntemleri mevcuttur. Otomotiv sektöründe optimizasyon yöntemlerinin kullanılması ve yeni araç geliştirme süreçlerine dahil edilmesi, rekabetçi bir hal alan sektörde giderek daha da büyük bir önem kazanmaktadır. Bu nedenle mevcut optimizasyon yöntemlerinin performansını artırılması, daha kısa sürede ve daha az kaynak kullanarak global optimuma en yakın sonuca ulaşabilmeleri otomotiv ve diğer endüstriler için oldukça değerlidir.

Bu çalışmada literatürde yer alan yarı sezgisel bir algoritma olan kaotik kril sürüsü optimizasyon yöntemi ele alınmıştır. Kaotik kril sürüsü yöntemi, mevcut kril sürüsü optimizasyon yönteminin kaos teorisi ile birleştirilmiş hibrit bir halidir. Bu sayede, yöntemin global optimumu daha kolay keşfetmesi ve lokal minimumlara sıkışıp kalmaması sağlanmıştır. Literatürdeki mevcut kril sürüsü yöntemine farklı kaotik davranışlar eklenerek güçlendirilen yeni algoritmanın performansı, kıyaslama fonksiyonları ile test edilmiştir. Oldukça fazla sayıda lokal optimumu bulunan bu zorlayıcı fonksiyonlarda bile, test edilen diğer algoritmaların önünde ve global optimuma en yakın sonuçların elde edilmesiyle geliştirilen algoritmanın performansı kanıtlanmıştır. Geliştirilen bu yöntem, araçlarda titreşim sönümleyici olarak kullanılan kauçuk burçların şekil optimizasyonunda kullanılmıştır. Otomotivde yaygın olarak kullanılan kauçuk burçların direngenlikleri kullanıldığı yere göre değişmektedir. Bu nedenle bir burç tasarımından beklenen, istenen boyutlarda kalarak istenen direngenlikleri sağlamasıdır. Parametrik olarak oluşturulan sonlu elemanlar modelinden elde edilen sonuçlara göre amaç fonksiyonuna en uygun tasarım değişkenleri elde edilmiştir.

Geliştirilen yeni yöntem ile elde edilen tasarım, literatürde iyi bilinen diğer algoritmalar ile karşılaştırıldığında, istenen hedefler açısından daha başarılı olduğu görülmüştür.

Anahtar Kelimeler: Optimizasyon, kril sürüsü, kaos, kaotik haritalar, sonlu elemanlar analizi, kauçuk burç, hiperelastisite

2021, XIV + 149 sayfa.

ABSTRACT

PhD Thesis

Development of Chaotic Krill Herd Optimization Method And Optimization of a Suspension Part

Halil BİLAL

Bursa Uludağ University
Graduate School of Natural and Applied Sciences
Department of Automotive Engineering

Supervisor: Prof. Dr. Ferruh ÖZTÜRK

There are various optimization methods developed to solve the optimization problems we encounter in all areas of life. However, the use of optimization methods in the automotive industry and their inclusion in new vehicle development processes are becoming more and more critical in the competitive industry. For this reason, it is essential for automotive and other industries to increase the performance of current optimization methods and to reach the closest global optimum in a shorter time and using fewer resources.

In this study, the chaotic krill herd optimization method, a semi-heuristic algorithm in the literature, is studied. The chaotic krill herd method is a hybrid version of the existing krill herd optimization method combined with chaos theory to increase the effectiveness. As a result, the novel method discovers the global optimum more efficiently and does not get stuck in local minima. The new algorithm's performance, which is more potent by adding different chaotic behaviors to the existing krill herd method in the literature, has been tested with the benchmark functions. Even in these compelling functions with quite a large number of local optima, the performance of the developed algorithm has been proven by obtaining the results closest to the global optimum in front of the other tested algorithms.

The developed method was used in shape optimization of rubber bushings used as vibration absorbers in vehicles. The stiffness of the rubber bushes, which are widely used in automotive, varies according to the place of use. For this reason, a target of the bushing design is to provide the desired stiffness by remaining in the desired dimensions. According to the results obtained from the parametrically created finite element model, the most suitable design variables for the objective function were obtained.

When the design obtained by the new method developed is compared with the other well-known algorithm in the literature, it has been observed that it is more successful in terms of desired goals.

Key words: Optimization, krill herd, chaos, chaotic maps, finite element analysis, rubber bushing, hyperelasticity

2021, XIV + 149 pages.

TEŞEKKÜR

Tez çalışmam sırasında değerli bilgileri ve tecrübeleri ile bana yol gösteren tez danışmanım ve değerli hocam Prof. Dr. Ferruh ÖZTÜRK'e teşekkürlerimi sunarım.

Doğduğum günden beri hep yanımda olan ve desteklerini hiçbir zaman benden esirgemeyen anneme ve ablalarıma teşekkür ederim.

Tez çalışmam nedeniyle kendilerine zaman ayıramadığım eşim ve canım oğluma, gösterdikleri sabır, anlayış ve her türlü yardımları için teşekkür ederim.

Halil BİLAL
29/08/2021

İÇİNDEKİLER

Sayfa

ÖZET.....i
ABSTRACT.....ii
TEŞEKKÜR.....iii
SİMGELER ve KISALTMALAR DİZİNİ.....vii
ŞEKİLLER DİZİNİ.....x
ÇİZELGELER DİZİNİ.....xiv
1. GİRİŞ 1
2. KAYNAK ARAŞTIRMASI.....3
2.1. Yapısal Tasarım Çalışmalarında Kullanılan Optimizasyon Teknikleri.....3
2.2. Kaotik Optimizasyon Teknikleri.....11
2.3. Yapısal Tasarım Çalışmalarında Kullanılmayan Kaotik Yarı Sezgisel Algoritmalar.....13
2.4. Kompleks Doğrusal Olmayan Model İçeren Yapısal Tasarım Çalışmalarında Karşılaşılan Zorluklar.....13
2.5. Kaotik Optimizasyon Tekniğinin Seçimi.....16
3. MATERYAL ve YÖNTEM.....17
3.1. Otomotivde Ürün Geliştirme Süreci ve Optimizasyon.....17
3.1.1. Tasarım Uzayının Belirlenmesi.....20
3.1.2. Topoloji Optimizasyonu.....20
3.1.3. Topolojiye Uygun Endüstriyel Tasarımın Oluşturulması.....21
3.1.4. Şekil ve Boyut Optimizasyonu.....22
3.1.5. Optimize Edilen Şeklin Endüstriyelleştirilmesi.....23
3.1.6. Kabartma Optimizasyonu.....23
3.1.7. Nihai Tasarımın Oluşturulması.....24
3.1.8. Doğrulama Analizleri.....24
3.2. Otomotiv Tasarım ve Optimizasyonunda Karşılaşılan Zorluklar.....24
3.3. Kaotik Optimizasyon tekniği.....26
3.4. Kaotik Sistemlere Giriş.....27
3.4.1. Lojistik Harita.....30
3.4.2. Sinüzoidal Harita.....32

3.4.3. Sinüs Haritası 33
3.4.4. Çadır Haritası 34
3.4.5. Gaus Haritası 36
3.4.6. Chebyshev Haritası 36
3.4.7. Cubic Haritası 37
3.4.8. Intermittency Haritası 38
3.4.9. Neuron Haritası 39
3.4.10. Liebovitch Haritası 40
3.4.11. ICMIC Haritası 41
3.4.12. Singer Haritası 42
3.5. Kaotik Sistemlerin İstatistiksel Değerlendirmeleri 43
3.5.1. Kelebek Etkisi - Başlangıç Koşullarına Hassas Bağımlılık 44
3.5.2. Lyapunov Üsteli 47
3.5.3. Olasılık Fonksiyonu 50
3.6. Biyo-coğrafya Temelli Kril Sürüsü Optimizasyonu 54
3.4.1. Kril Sürüsü Optimizasyonu 64
3.4.2. Kril Sürüsü Algoritması Hareket Süreci 68
3.4.3. Genetik Operatörler 69
3.4.4. Kril Sürüsü Algoritması 71
3.7. Kaotik Kril Sürüsü Optimizasyonu 71
3.8. Önerilen Kaotik Kril Sürüsü Optimizasyonu 76
4. BULGULAR 83
4.1. Kaotik Kril Sürüsü Yönteminin Performans Testleri 83
4.1.1. En Kötüler 84
4.1.2. Standart Sapmalar 87
4.1.3. Ortalamalar 90
4.1.4. En İyiler 93
4.2. Önerilen KKS Algoritmasının Seçimi 102
4.3. Geliştirilen KKS Yöntemi ile Kauçuk Burç Optimizasyonu 109
4.3.1. Optimizasyon Probleminin Kurulması 111
4.3.2. Kauçuk Burç Sonlu Elemanlar Modeli 116
4.3.3. Sınır Koşulu ve Yükler 117

4.3.4. Çözüm ve Sonuçların Değerlendirilmesi 119
4.4. Kauçuk Burç Optimizasyonunda Farklı Optimizasyon Algoritmalarının Karşılaştırılması 122
5. TARTIŞMA ve SONUÇ 127
5.1. KKS Katkıları ve Güçlü Yönleri 127
5.2. Ek Çalışmalar 132
KAYNAKLAR 133

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler	Açıklama
c	Kaotik seri sabiti (Intermittency haritası)
C^{eniyi}	Etki katsayısı
Cr	Çaprazlama olasılık sabiti
C_t	Zaman adımı sabiti
C_{10} , C_{01}	Hiperelastik malzeme modeli sabitleri
d	Kaotik seri sabiti (Intermittency haritası)
D_0	Başlangıçtaki ayrışma (Lyapunov Üsteli)
D^{maks}	Maksimum yayılma hızı
D_t	t. adımdaki ayrışma (Lyapunov Üsteli)
D_i	Fiziksel yayılma
d_s	Hissedilen uzaklık
F_i	Yem arama hareketi
f	Amaç fonksiyonu
g	Eşitsizlik kısıt fonksiyonu
h	Eşitlik kısıt fonksiyonu
I	Döngü Sayısı
I_1 , I_2	Gerinim değişmezleri
I_{maks}	Maksimum döngü sayısı
k_r^H	Hedef radyal direngenlik
k_e^H	Hedef eksenel direngenlik
k_t^H	Hedef torsiyonel direngenlik
k_k^H	Hedef kardanik direngenlik
k_r^i	i. adımda hesaplanan radyal direngenlik
k_e^i	i. adımda hesaplanan eksenel direngenlik
k_t^i	i. adımda hesaplanan torsiyonel direngenlik
k_k^i	i. adımda hesaplanan kardanik direngenlik
$K^{enkötü}$	En kötü amaç fonksiyonu değeri

K^{eniyi}	En iyi amaç fonksiyonu değeri
K_i	i. kril bireyinin amaç fonksiyonu değeri
K_j	j. komşu bireyinin amaç fonksiyonu değeri
LB_j	j. değişkenin alt sınırı
m	Kaotik seri sabiti (Intermittency haritası)
Mu	Mutasyon olasılık sabiti
N^{maks}	Maksimum sebep olunan hız
NN	Toplam komşu sayısını
N_i	Diğer kril bireylerinin sebep olduğu hareket
P	Gerçek değerli bir fonksiyon
r	Ceza parametresi vektörü
UB_j	j. değişkenin üst sınırı
V_f	Yem arama hızını
w_f	Yem arama hareketinin atalet ağırlığı
ω_n	Sebep olunan hareketin atalet ağırlığı
x_n	Kaotik Seri
w_i	i. kısıt fonksiyonuna (h_i) ait ağırlık fonksiyonu
X_i	Krilin pozisyonu
α_i	Sebep olunan hareketin doğrultusu
α	Kaotik seri sabiti (Sinüs haritası)
β_i^{yem}	Yem çekiciliği
β_i^{eniyi}	i. kril bireyinin şu ana kadarki en iyi amaç fonksiyonu değerinin etkisi
γ	Oransallık faktörü (Neuron haritası)
η	Azaltma faktörü (Neuron haritası)
λ	Lyapunov Üsteli
Ψ	Gerinim enerji fonksiyonu
$\Phi(x,r)$	Transformasyon fonksiyonu
δ	Yönlü vektör
ε	Tekillik düzeltme sabiti
μ	Kaotik seri sabiti

Kısaltmalar	Açıklama
Alg.	Algoritma
CKH	Chaotic Krill Herd
DGA	Diferansiyel Gelişim Algoritması
GA	Genetik Algoritma
KH	Krill Herd
KKS	Kaotik Kril Sürüsü
KS	Kril Sürüsü
KSO	Kril Sürüsü Optimizasyonu
PSO	Parçacık Sürüsü Optimizasyonu

ŞEKİLLER DİZİNİ

Sayfa

Şekil 3.1 Aşama Eşiği Yöntem (Cooper 1990)....18
Şekil 3.2 Otomotivde Ürün Geliştirme ve Üretim Süreci Akış Diyagramı (Bhise 2017) 19
Şekil 3.3 Otomotiv ürün geliştirme sürecinde V Modeli (Johanson 2012) 19
Şekil 3.4 Tüm araç alt gövde topoloji optimizasyonu sonucu (Volz 2011)....21
Şekil 3.5 Topoloji optimizasyonu sonrasında elde edilen topoloji ve buna göre oluşturulan tasarım....22
Şekil 3.6 İnce cidarlı yapılarda şekil optimizasyonu (W. Zhang, Zhu, ve Gao 2016)....23
Şekil 3.7 Sıvı Tankı kabartma optimizasyonu (Altair Inc. 2018)....23
Şekil 3.8 Kaotik Haritalamaya bir örnek (Logistic Map (Kaveh Ali 2014))....28
Şekil 3.9 Farklı μ değerleri ile elde edilen Lojistik Harita Serileri (Feng vd. 2017)....31
Şekil 3.10 Lojistik harita fonksiyonu ile üretilen seriye ait serpilme ve çubuk dağılım grafikleri (Feng vd. 2017)....32
Şekil 3.11 Sinüzoidal harita fonksiyonu ile üretilen seriye ait serpilme ve çubuk dağılım grafikleri (Feng vd. 2017)....33
Şekil 3.12 Sinüs harita fonksiyonu kullanılarak farklı α değerleri ile elde edilen kaotik seriler....34
Şekil 3.13 Sinüzoidal harita fonksiyonu ile üretilen seriye ait serpilme ve çubuk dağılım grafikleri (Feng vd. 2017)....34
Şekil 3.14 Çadır harita fonksiyonu (eşitlik 3.8) ile üretilen seriye ait serpilme ve çubuk dağılım grafikleri (Feng vd. 2017)....35
Şekil 3.15 Çadır harita fonksiyonu (eşitlik 3.9) ile üretilen seriye ait serpilme ve çubuk dağılım grafikleri (Feng vd. 2017)....36
Şekil 3.16 Chebyshev harita fonksiyonu ile üretilen seriye ait serpilme ve çubuk dağılım grafikleri (Feng vd. 2017)....37
Şekil 3.17 Cubic harita fonksiyonu ile üretilen seriye ait serpilme ve çubuk dağılım grafikleri (Feng vd. 2017)....38
Şekil 3.18 Intermittency harita fonksiyonu ile üretilen seriye ait serpilme ve çubuk dağılım grafikleri (Feng vd. 2017)....39

Şekil 3.19 Neuron harita fonksiyonu ile üretilen seriye ait serpilme ve çubuk dağılım grafikleri (Feng vd. 2017) 40
Şekil 3.20 Liebovitch harita fonksiyonu ile üretilen seriye ait serpilme ve çubuk dağılım grafikleri (Feng vd. 2017) 41
Şekil 3.21 ICMIC harita fonksiyonu ile üretilen seriye ait serpilme ve çubuk dağılım grafikleri (Feng vd. 2017) 42
Şekil 3.22 Singer harita fonksiyonu ile üretilen seriye ait serpilme ve çubuk dağılım grafikleri 43
Şekil 3.23 x_0=0.4 ve y_0=0.41 başlangıç değerleri ile oluşturulan x ve y kaotik serileri (Feldman 2012) 45
Şekil 3.24 x ve y kaotik seriler arasındaki fark (Feldman 2012) 45
Şekil 3.25 μ=4.0 ve $0.395 < x_0 < 0.405$ için elde edilen 1000 farklı kaotik seri (Feldman 2012) 46
Şekil 3.26 μ=3.2 ve $0.395 < x_0 < 0.405$ için elde edilen 1000 farklı kaotik seri (Feldman 2012) 47
Şekil 3.27 2 zaman serisi arasındaki mutlak fark (Feldman 2012) 49
Şekil 3.28 Normal olasılık dağılım fonksiyonu 51
Şekil 3.29 Lojistik Harita kaotik serisinin olasılık dağılımı 52
Şekil 3.30 Bernauli Shift ve Kent kaotik serilerinin olasılık dağılımı 52
Şekil 3.31 Sinüs haritası kaotik serisinin olasılık dağılımı 52
Şekil 3.32 ICMIC haritası kaotik serisinin olasılık dağılımı 53
Şekil 3.33 Çember haritası kaotik serisinin olasılık dağılımı 53
Şekil 3.34 Chebyshev haritası kaotik serisinin olasılık dağılımı 53
Şekil 3.35 Gauss haritası kaotik serisinin olasılık dağılımı 54
Şekil 3.36 Boru şeklindeki kolon tasarımı (Amir H. Gandomi ve Alavi 2015) 57
Şekil 3.37 3-Çubuk Kafes Tasarımı (Amir H. Gandomi ve Alavi 2015) 57
Şekil 3.38 Hız düşürücü tasarımı (Amir H. Gandomi ve Alavi 2015) 57
Şekil 3.39 Helisel Yay Tasarımı (Amir H. Gandomi ve Alavi 2015) 58
Şekil 3.40 Kaynaklı kiriş optimizasyon problemi (Abdel-Basset vd. 2017) 61
Şekil 3.41 Basınçlı kap problemi (Abdel-Basset vd. 2017) 62
Şekil 3.42 Dişli Grubu Tasarımı (Abdel-Basset vd. 2017) 62

Şekil 3.43 Hissedilen uzaklık ve komşuluk yapısı (Amir Hossein Gandomi ve Alavi 2012), (Gölcük vd. 2014) 66
Şekil 3.44 Kril Sürüsü Algoritmasının basitleştirilmiş Akış Şeması (Amir Hossein Gandomi ve Alavi 2012) 72
Şekil 3.45 Kıyaslama Fonksiyonlar 79
Şekil 3.46 Yukarıdaki harita fonksiyonlarıyla üretilen serilere ait serpilme grafikleri ..80
Şekil 3.47 Yukarıdaki harita fonksiyonlarıyla üretilen serilere ait çubuk dağılım grafikleri 80
Şekil 4.1 Her bir algoritmadan elde edilen ortalamalar (çizgi grafik) 96
Şekil 4.2 Her bir algoritmadan elde edilen ortalamalar (Radar grafik) 97
Şekil 4.3 Her bir algoritmadan elde edilen standart sapmalar (çizgi grafik) 98
Şekil 4.4 Her bir algoritmadan elde edilen standart sapmalar (radar grafik) 99
Şekil 4.5 Her bir algoritmadan elde edilen en iyiler (çizgi grafik) 100
Şekil 4.6 Her bir algoritmadan elde edilen en iyiler (radar grafik) 101
Şekil 4.7 Her bir algoritmadaki Harita fonksiyonlarının ilk 4 'e giriş sayıları 103
Şekil 4.8 Harita Fonksiyonlarının Standart Sapmada İlk 4 'e Giriş Toplamları 104
Şekil 4.9 Harita Fonksiyonlarının Ortalamalarda İlk 4 'e Giriş Toplamları 104
Şekil 4.10 Harita Fonksiyonlarının En İyilerde İlk 4 'e Giriş Toplamları 104
Şekil 4.11 Harita Fonksiyonlarının Standart Sapmada İlk 4 'e Giriş Sayıları 105
Şekil 4.12 Harita Fonksiyonlarının Ortalamalarda İlk 4 'e Giriş Sayıları 105
Şekil 4.13 Harita Fonksiyonlarının En İyilerde İlk 4 'e Giriş Sayıları 105
Şekil 4.14 Algoritma 6'nın Harita Fonksiyonlarının İlk 4 'e Giriş Sayıları (Çubuk grafik) 106
Şekil 4.15 Algoritma 6'nın Harita Fonksiyonlarının İlk 4 'e Giriş Sayıları (Yığın grafik) 106
Şekil 4.16 Geliştirilen KKS ‘nün Diğer Yarı Sezgisel Algoritmalar ile Karşılaştırması, Normalize Edilmiş En Kötü 107
Şekil 4.17 Geliştirilen KKS ‘nün Diğer Yarı Sezgisel Algoritmalar ile Karşılaştırması, Normalize Edilmiş Standart Sapma 108
Şekil 4.18 Geliştirilen KKS ‘nün Diğer Yarı Sezgisel Algoritmalar ile Karşılaştırması, Normalize Edilmiş Ortalama 108

Şekil 4.19 Geliştirilen KKS ‘nün Diğer Yarı Sezgisel Algoritmalar ile Karşılaştırması, Normalize Edilmiş En İyi 109
Şekil 4.20 Titreşim sönümleyici elastik burçlar (Heißing ve Ersoy 2015). 111
Şekil 4.21 Elastik burç yükleme yönleri (Goossens vd. 2017) 111
Şekil 4.22 Elastik Burç Tasarım Değişkenleri 115
Şekil 4.23 Elastik Burç Sonlu Elemanlar Modeli 117
Şekil 4.24 Radyal direngenlik Değişimi 121
Şekil 4.25 Eksenel direngenlik değişimi 121
Şekil 4.26 Torsiyonel direngenlik değişimi 121
Şekil 4.27 Kardanik direngenlik değişimi 121
Şekil 4.28 Değişkenlerin çözüm sırasındaki değişimi 121
Şekil 4.29 Amaç fonksiyonunun optimizasyon süresince değişimi 122
Şekil 4.30 KKS, PSO ve DGA yöntemlerinden elde edilen direngenlik değerleri 124
Şekil 4.31 KKS , PSO ve DGA amaç fonksiyonu değişimi 125
Şekil 4.32 KKS ve DGA optimizasyon problemi çözüm süreleri 125
Şekil 4.33 KKS ve DGA ile optimize edilen tasarım değişkenleri 126
Şekil 5.1 X1 ve X2 değişkenlerinin İterasyon boyunca aldıkları değerler 129
Şekil 5.2 Değişkenlerin aldığı değerlere ait kutu grafiği 130
Şekil 5.3 X1 değerlerinin çubuk dağılım grafiği 131
Şekil 5.4 X2 değerlerinin çubuk dağılım grafiği 131
Şekil 5.5 X1 ve X2 değerlerinin Standart Sapmaları 132

ÇİZELGELER DİZİNİ

Sayfa

Çizelge 2.1 Yapısal tasarım problemlerinde kullanılan yarı sezgisel algoritmalar (Ali Kaveh, Majid, ve Ghazaan 2018)....4
Çizelge 2.2 Çok amaçlı optimizasyonlarda kullanılan yarı sezgisel algoritmalar (Ali Kaveh 2014)....5
Çizelge 2.3 Otomotiv tasarımında karşılaşılan optimizasyon uygulamaları....6
Çizelge 2.4 Optimizasyon Yöntemleri....7
Çizelge 2.5 İşbirlikçi Hibrit Algoritmalar....8
Çizelge 2.6 Tümleşik Hibrit Algoritmalar....9
Çizelge 2.7 Literatürde sıkça karşılaşılan yarı sezgisel optimizasyon yöntemleri....10
Çizelge 2.8 Yarı sezgisel algoritmalarla yapılan çalışma sayıları (Nabaei vd. 2018)....11
Çizelge 2.9 Yarı sezgisel algoritmalar ve kaotik uygulamaları....12
Çizelge 2.10 Yapısal tasarım uygulamalarında kullanılmayan kaotik ve yarı sezgisel algoritmalar....14
Çizelge 3.1 Lyapunov üstelleri....49
Çizelge 3.2 Kaotik Haritaları Olasılık Yoğunluk Fonksiyonları....51
Çizelge 3.3 literatürde karşılaşılan kaotik uygulamalar....75
Çizelge 3.4 Literatürde karşılaşılan kaotik uygulamaların sınıflandırması....75
Çizelge 3.5 Test Fonksiyonları ve Çözümleri....78
Çizelge 3.6 KS algoritmalarında kullanılan sabitler....81
Çizelge 4.1 Mevcut KS yönteminden (Algoritma 1) elde edilen En Kötü Değerler....84
Çizelge 4.2 KKS Wang Algoritmasıyla (Algoritma 2) elde edilen En Kötü Değerler....84
Çizelge 4.3 Önerilen KKS algoritmasıyla (Algoritma 3) elde edilen En Kötü Değerler 84
Çizelge 4.4 Önerilen KKS algoritmasıyla (Algoritma 4) elde edilen En Kötü Değerler 85
Çizelge 4.5 Önerilen KKS algoritmasıyla (Algoritma 5) elde edilen En Kötü Değerler 85
Çizelge 4.6 KKS algoritmasıyla (Algoritma 6) elde edilen En Kötü Değerler....85
Çizelge 4.7 PSO ile elde edilen En Kötü Değerler....86
Çizelge 4.8 GA ile elde edilen En Kötü Değerler....86
Çizelge 4.9 DGA ile elde edilen En Kötü Değerler....86

Çizelge 4.10 Mevcut KS yönteminden (Algoritma 1) elde edilen sonuçların standart sapmaları 87
Çizelge 4.11 KKS Wang algoritmasıyla (Algoritma 2) elde edilen sonuçların standart sapmaları 87
Çizelge 4.12 Önerilen KKS algoritmasıyla (Algoritma 3) elde edilen sonuçların standart sapmaları 87
Çizelge 4.13 Önerilen KKS algoritmasıyla (Algoritma 4) elde edilen sonuçların standart sapmaları 88
Çizelge 4.14 Önerilen KKS algoritmasıyla (Algoritma 5) elde edilen sonuçların standart sapmaları 88
Çizelge 4.15 Önerilen KKS algoritmasıyla (Algoritma 6) elde edilen sonuçların standart sapmaları 88
Çizelge 4.16 PSO ile elde edilen sonuçların standart sapmaları 89
Çizelge 4.17 GA ile elde edilen sonuçların standart sapmaları 89
Çizelge 4.18 DGA ile elde edilen sonuçların standart sapmaları 89
Çizelge 4.19 Mevcut KS yönteminden (Algoritma 1) elde edilen sonuçların ortalamaları 90
Çizelge 4.20 KKS Wang Algoritmasıyla (Algoritma 2) elde edilen sonuçların ortalamaları 90
Çizelge 4.21 Önerilen KKS Algoritmasıyla (Algoritma 3) elde edilen sonuçların ortalamaları 90
Çizelge 4.22 Önerilen KKS Algoritmasıyla (Algoritma 4) elde edilen sonuçların ortalamaları 91
Çizelge 4.23 Önerilen KKS Algoritmasıyla (Algoritma 5) elde edilen sonuçların ortalamaları 91
Çizelge 4.24 Önerilen KKS Algoritmasıyla (Algoritma 6) elde edilen sonuçların ortalamaları 91
Çizelge 4.25 PSO ile elde edilen sonuçların ortalamaları 92
Çizelge 4.26 GA ile elde edilen sonuçların ortalamaları 92
Çizelge 4.27 DGA ile elde edilen sonuçların ortalamaları 92
Çizelge 4.28 Mevcut KS yönteminden (Algoritma 1) elde edilen sonuçların en iyileri. 93

Çizelge 4.29 KKS Wang algoritmasıyla (Algoritma 2) elde edilen sonuçların en iyileri 93
Çizelge 4.30 Önerilen KKS algoritmasıyla (Algoritma 3) elde edilen sonuçların en iyileri 93
Çizelge 4.31 Önerilen KKS algoritmasıyla (Algoritma 4) elde edilen sonuçların en iyileri 94
Çizelge 4.32 Önerilen KKS algoritmasıyla (Algoritma 5) elde edilen sonuçların en iyileri 94
Çizelge 4.33 Önerilen KKS algoritmasıyla (Algoritma 6) elde edilen sonuçların en iyileri 94
Çizelge 4.34 PSO ile elde edilen sonuçların en iyileri.....95
Çizelge 4.35 GA ile elde edilen sonuçların en iyileri 95
Çizelge 4.36 DGA ile elde edilen sonuçların en iyileri 95
Çizelge 4.37 En iyi sonuçlara ulaşan algoritma.....102
Çizelge 4.38 En iyi sonuçlara ulaşan harita fonksiyonu.....102
Çizelge 4.39 Hedef Direngenlik değerleri 114
Çizelge 4.40 Tasarım Değişkenleri Alt ve Üst Limitleri 115
Çizelge 4.41 Optimize edilen tasarım değişkenleri 120
Çizelge 4.42 Optimize edilen direngenlik değerleri 120

1. GİRİŞ

Otomotivde ürün geliştirme süreci, müşterilerin ve onların ihtiyaçlarının belirlenmesiyle başlamaktadır. Ardından bu ihtiyaçların ürün özelliğine bir başka deyişle teknik ifadelere dönüşmesiyle devam eder. Ürün tasarım süreci ve mühendislik çalışmaları ise bu noktadan sonra başlar. Otomotivdeki bu tasarım sürecini en iyi ifade eden ve yaygın olarak kullanılan model, V modeli olarak adlandırılır Burada komple araç seviyesinde belirlenen müşteri beklentileri veya ihtiyaçlar, sırasıyla, sistemlere, oradan alt sistemlere ve oradan da parça seviyesine indirgenir. Ardından en alt seviyedeki parça tasarımı ve doğrulama çalışmaları gerçekleştirilir. Bu kez de en alt seviyedeki parça seviyesinden en üst seviyedeki komple araca kadar olan tüm alt sistem ve sistemler sırasıyla test edilir ve bu sayede ürün doğrulanır.

V modeli ve buna benzer önce tasarla sonra test et şeklindeki ürün geliştirme yöntemleri aşama aşama ilerleyen geleneksel yöntemlerdir. Bu yöntemlerde ürün geliştirme sürecinin ardından doğrulama testleri gerçekleştirilir. Oysa ki yenilikçi yöntemlerde ürün geliştirme süresini kısaltabilmek için bu süreçler iç içe geçmiştir ve birlikte gerçekleşir. Tasarımın ve doğrulamanın birlikte gerçekleştiği bu yöntem, Simülasyon Tabanlı Ürün Tasarımı olarak adlandırılır.

Otomotiv sanayiinde faaliyet gösteren ve tasarım yapan her firma, otomotivdeki artan rekabet koşulları nedeniyle, en güçlü ve en ekonomik ürünü, pazara en kısa sürede sürmek zorundadırlar. Bu zorluğun üstesinden gelebilmeleri için de kısa sürede ürün geliştirebilmelerine imkân veren, Simülasyon Tabanlı Ürün Tasarımı yönteminin kullanımının giderek artığı görülmektedir. Bu yöntemde ürün geliştirmede kullanılabilecek optimizasyon yöntemlerinin neredeyse tamamı kullanılmaktadır.

Optimizasyon yöntemlerinin kullanımındaki bu artış araştırmacıların sürekli olarak yeni optimizasyon algoritmaları geliştirmeleri için oldukça önemli bir motivasyon kaynağıdır. Geliştirilen her bir yeni yöntem ile kurulan optimizasyon probleminin daha kısa sürede, daha az kaynak kullanarak en iyi şekilde çözebilmek amaçlanır.

Bu tez çalışmasında, otomotivde ürün geliştirme sürecinde kullanılabilecek oldukça güçlü bir optimizasyon metodu geliştirilmiştir. Bu yöntem literatürde mevcut olan kaotik kril sürüsü (KKS) yönteminin farklı kaotik davranışlar içeren yeni bir halidir. Algoritmanın geliştirilmesi, kril sürüsü yönteminin farklı adımlarına farklı kaotik davranışlar ekleyerek gerçekleştirilmiştir. Oluşturulan her yeni algoritmanın performansı, çok sayıda lokal optimum içeren test fonksiyonları kullanılarak mevcut optimizasyon algoritmaları ile karşılaştırılarak yapılmıştır. Bu karşılaştırmalarda, yeni kaotik kril sürüsü algoritmalarından mevcut algoritmalara göre performansı en iyi olan algoritma seçilmiştir.

Seçilen KKS algoritması, otomotivde yaygın olarak kullanılan kauçuk burçların tasarım optimizasyonunda kullanılmıştır. Elde edilen sonuçlar diferansiyel gelişim algoritmasından elde edilenler ile karşılaştırılmıştır. Geliştirilen algoritmanın doğrusal olmayan bir yapısal tasarım probleminde dahi kullanılabilmesi hem uygulanabilirliğini kanıtlanmış hem de mevcut yöntemlere göre olan başarısını bir kez daha göstermiştir. Bu çalışma ile geliştirilen algoritma ile literatüre yeni bir yaklaşım kazandırılmıştır.

2. KAYNAK ARAŞTIRMASI

KKS optimizasyon yönteminin geliştirilmesi için yapılan kaynak araştırması aşağıdaki ana konular çerçevesinde yapılmıştır.

- Yapısal Tasarım Çalışmalarında Kullanılan Optimizasyon Teknikleri
- Yapısal Tasarım Çalışmalarında Kullanılan Kaotik Optimizasyon Teknikleri
- Yapısal Tasarım Çalışmalarında Kullanılmayan Kaotik Optimizasyon ve Yarı Sezgisel Algoritmalar
- Kompleks Doğrusal Olmayan Model İçeren Yapısal Tasarım Çalışmalarında Karşılaşılan Zorluklar

2.1. Yapısal Tasarım Çalışmalarında Kullanılan Optimizasyon Teknikleri

Bu kapsamda yapılan kaynak araştırması 3 alt kısımdan oluşmaktadır.

- Gerçek boyutlu yapıların optimizasyonunda kullanılan yarı sezgisel algoritmalar
- Yapıların optimizasyonunda kullanılan yarı sezgisel algoritmalardaki gelişmeler
- Araç süspansiyonlarındaki yapısal optimizasyon çalışmaları

Kaveh ve Ghazaan (2018), gerçek boyutlu yapıların optimizasyonunda kullanılan yarı sezgisel algoritmalar ve bu algoritmalardaki gelişmeleri derledikleri 2 ayrı elektronik kitapta toplamıştır. Yarı sezgisel optimizasyonların gelişim sürecini inceledikleri bu çalışmalarında, kullanılan yarı sezgisel algoritmaları açıklamış, 2 ve 3 boyutlu kafes yapıların titreşim ve dayanım optimizasyonlarındaki performanslarını değerlendirilmiştir. Burada ele alınan optimizasyon yöntemleri Çizelge 2.1'de verilmiştir.

Yarı sezgisel optimizasyon algoritmalarının çok amaçlı optimizasyon problemlerindeki performanslarını önce kısıt olmayan matematik problemleri ile test etmişler, daha sonra da kısıt uygulanan kafes yapılardaki boyut optimizasyon problemlerinde kullanmışlardır. Burada incelenen kafes yapılar, 10, 18 ve 52 kirişli kafesler, Basit mesnetli 37 kirişli düzlemsel kafes, 52 ve 120 kirişli kubbe kafesler, 25 kirişli özel kafes gibi literatürde karşılaşılan kafes yapılardır (Ali Kaveh, Majid, ve Ghazaan 2018), (Ali Kaveh 2014).

Çizelge 2.1 Yapısal tasarım problemlerinde kullanılan yarı sezgisel algoritmalar (Ali Kaveh, Majid, ve Ghazaan 2018)

No	Kısaltma	İngilizce Uzun Adı	Türkçe Uzun Adı
1	CBO	Colliding Bodies Optimization	Çarpışan Cisimler Optimizasyonu
2	ECBO	Enhanced Colliding Bodies Optimization	Geliştirilmiş Çarpışan Cisimler Optimizasyonu
3	VPS	Vibrating Particles System	Titreşen Parçacıklar Sistemi
4	PSO	Particle Swarm Optimization	Parçacık Sürüsü Optimizasyonu
5	CSS	Charged System Search Algorithm	Yüklü Sistem Arama Algoritması
6	MCSS	Magnetic Charged System Search	Manyetik Yüklü Sistem Araması
7	FOF	Field of Forces Optimization	Kuvvetler Alanı Optimizasyonu
8	DEO	Dolphin Echolocation Optimization	Yunus Ekosu ile Yer Tespiti Optimizasyon Algoritması
9	RO	Ray Optimization Algorithm	Işın Optimizasyon Algoritması
10	MBB-BC	Modified Big Bang–Big Crunch Algorithm	Değiştirilmiş Büyük Patlama Büyük Çöküş Algoritması
11	CSO	Cuckoo Search Optimization	Guguk Kuşu Arama Algoritması
12	ICA	Imperialist Competitive Algorithm	Emperyalist Rekabetçi Algoritması

Burada incelenen çok amaçlı optimizasyon problemlerin özellikleri şunlardır (Kahev 2014);

I. Birçok tasarım değişkenin katılması
II. Hedef fonksiyonun, genellikle çok fazla yerel optimumu olan çok modelli (multi-modal) bir fonksiyon olması

III. Hedef fonksiyonun hesaplanabilmesi için yapının oluşturulması ve analizlerin tamamlanmasını gerektirdiği için hesaplama maliyeti oldukça yüksek olması

Çok amaçlı optimizasyon problemlerinde kullanılan yarı sezgisel algoritmalar Çizelge 2.2'de gösterilmiştir.

Çizelge 2.2 Çok amaçlı optimizasyonlarda kullanılan yarı sezgisel algoritmalar (Ali Kaveh 2014)

No	Kısaltma	İngilizce Uzun Adı	Türkçe Uzun Adı
1	MOEA/D	Multi-objective evolutionary algorithm based on decomposition	Ayrıştırmaya dayalı çok amaçlı evrimsel algoritma
2	MOPSO	Multi Objective Particle Swarm Optimization	Çok Amaçlı Parçacık Sürüsü Optimizasyonu
3	MOGA	Multi Objective Genetic Algorithm	Çok Amaçlı Genetik Algoritma
4	NSGA-II	Non-dominated Sorting Genetic Algorithm	Baskın Olmayan Sıralama Genetik Algoritması
5	SPEA2	Strength Pareto Evolutionary Algorithm	Güç Pareto Evrim Algoritma
6	MO-MSCSS	Multi Objective – Charged System Search	Çok Amaçlı – Yüklü Sistem Araması

Degertekin ve Geem (2016), yakın zamanda geliştirilen kendinden uyarlamalı uyum arama (Self-Adaptive Harmony Search Algorithm, SAHSA) ve öğretme-öğrenme temelli (Teaching-Learning Based Optimization, TLBO) optimizasyon gibi yarı sezgisel optimizasyon yöntemlerini incelemiştir. Bu yöntemlerin performansını yapı mühendisliği alanındaki 2 ve 3 boyutlu uzay kafes sistemlerinin doğrusal statik analizlerinde test etmişlerdir. Elde ettikleri sonuçları, uyum arama (Harmony Search, HS), PSO, pasif birleşmeli PSO (particle swarm optimizer with passive congregation, PSOPC), sezgisel PSO (Heuristic Particle Swarm Optimization, HPSO), BB-BC ve Hibrit BB-BC (HBB-BC) ile elde edilenler ile karşılaştırmışlardır.

Çizelge 2.3 Otomotiv tasarımında karşılaşılan optimizasyon uygulamaları

Amaç Fonksiyonu	Değişkeler	Opt. Yöntemi	Referans
Araç konforu	Aktif süspansiyon elamanlarının karakteristikleri	APP	(Das vd. 2018)
Yüksek hızlı tren tasarımda yanal/dikey kararlılığı	Yüksek hızlı tren süspansiyon elemanları karakteristikleri	BPN	(J. Jiang vd. 2018)
Süspansiyon sistemi elastokinematik performansı (toe, kamber ve kaster açıları)	Süspansiyon geometrisi	GA	(Q. Li, Yu, ve Wu 2018)
Kompozit bagaj kapağı burulma direngenliği	Kompozit malzemedeki takviye elemanlarının yönleri	SQP	(C. Wu vd. 2017)
Çok amaçlı Şasi kolu optimizasyonu (Kalan enerji, Tepe çarpışma kuvveti, kütle ve ürün ailesi ceza fonksiyonu (PFPF)	Şasi kolu geometrisi ve kalınlığı	MOGA	(Hosseini vd. 2019)
Ağır ticari araç kabin içi gürültü seviyesi	Panel saclarının kalınlıkları	GA	(Z. Zhang vd. 2018)
Balastlı ve Panelli tren yolları arasındaki geçiş problemi	Demir yollarında kullanılan elastik padlerin direngenlikleri ve sleperler arasındaki mesafe	GA	(Aggestam ve Nielsen 2019)
Kamyon kabini yorulma dayanımı ve güçlü tasarım oluşturma	Kabin taşıyıcı kolonlarının kalınlıkları	MOPSO	(Fang vd. 2015)
Salıncak kolu yapısal dayanımı	Boşaltma deliklerinin çapı	ISO	(Yıldız 2017)
El fren braketi direngenliği	El fren braketi tasarım değişkenleri	GA	(Bilal, Serbest, ve Aydıner 2018)
Arka süspansiyon burulma kirişi yapısal dayanım ve toe açısı	Burulma kirişi tasarım değişkenleri	GA	(Bilal 2018)
Araç gövdesi burulma modu	Gövde sacı kalınlıkları	SQP	(Akdeniz, Yalçın, ve Bilal 2016)
Kauçuk burç direngenlik eğrisi	Kauçuk burç tasarım değişkenleri	DE	(Kaya 2014)
Otomobil parçası kalıbındaki yer değiştirme	Kalıp tasarım değişkenleri	DEBV	(İdris Karen 2011)

Otomotiv tasarımında kullanılan yapısal algoritmalar incelendiğinde hem yapısal hem de süspansiyon elemanlarının karakteristikleriyle yapılan optimizasyon çalışmalarına rastlanmaktadır. Bu çalışmalarda Burada yer alan optimizasyon problemlerindeki amaç fonksiyonları, değişkenler ve kullanılan optimizasyon yöntemleri Çizelge 2.3 'te sunulmuştur. Çizelge 2.3 'te kısaltmaları verilen optimizasyon yöntemlerinin açık adları Çizelge 2.4' te verilmiştir.

Çizelge 2.4 Optimizasyon Yöntemleri

Kısaltma	**İngilizce Açık Adı**	**Türkçe Açık Adı**
GA	Genetic Algorithm	Genetik Algoritma
MOGA	Multi Objective Genetic Algorithm	Çok Amaçlı Genetik Algoritma
MIGA	Multi-Island Genetic Algorithm	Çoklu-Ada Genetik Algoritma
MOPSO	Multi Objective Particle Swarm Optimization	Çok Amaçlı Parçacık Sürüsü Optimizasyonu
APPO	Adaptive Predator-Prey Optimization	Uyarlanabilir Av-Avcı Optimizasyonu
BPN	Back Propagation Neural Network	Geri Yayılımlı Sinir Ağı
SQP	Sequential Quadratic Programing	Sıralı Kuadratik Programlama
ISO	Interior Search Optimization	İç Arama Optimizasyonu
DE	Differential Evaluation	Diferansiyel Gelişim
DEBV	Differential Evaluation using Best Vectors	En İyi Vektörleri Kullanan Diferansiyel Gelişim

Farklı algoritmalara rastlanılmasına rağmen ekseriyetle genetik algoritmanın kullanıldığı görülmüştür. Sonuç olarak incelenen yapısal tasarım problemleri ile yapılan çalışmalarda hem tekil hem de çok amaçlı genetik algoritmalar yaygın olarak kullanılırken, hibrit yöntemlere nadiren rastlanılmıştır. Cevap yüzeyi yaklaşımının da bu çözümlerde yaygın olarak kullanıldığı görülmektedir.

Ting vd. (2015), 2010 yılından sonra geliştirilen yarı sezgisel hibrit optimizasyon algoritmalarını incelemişlerdir. İki farklı optimizasyon yönteminin avantajlı yönlerini

birleştirerek oluşturdukları yeni algoritma hibrit algoritma olarak adlandırılır. Bu hibrit algoritmalar da işbirlikçi (collaborative) ve tümleşik (integrative) olarak iki grupta sınıflandırılırlar (Ting vd. 2015).

İşbirlikçi algoritmalar, iki veya daha fazla algoritma sıralı veya paralel olarak çalışmasından oluşur. Algoritmaların katkı yüzdeleri en basit haliyle yarı yarıya olarak tanımlanabilir. Çok aşamalı işbirlikçi algoritmada, ilk algoritma global optimize edici olarak görev alırken diğeri o noktaya en yakın lokal optimumu bulmak için çalışır. Sıralı işbirlikçilikte, her bir algoritma diğerinden bağımsız olarak çalışır ancak bir çalışmaya başlamak için diğerinin istenilen sayıdaki iterasyonu tamamlamasını bekler. Paralel işbirlikçilikte ise sıralı da olduğu gibi bir bekleme olmaksızın her iki algoritma paralel olarak çalışır, ardından o adımda çıkan popülasyon birleştirilerek bir sonraki adıma geçilir (Ting vd. 2015). İşbirlikçi algoritmalardan bazıları örnek olarak Çizelge 2.5 'te verilmiştir.

Çizelge 2.5 İşbirlikçi Hibrit Algoritmalar

No	**Kısaltma**	**İngilizce Uzun Adı**	**Referans Kaynak**
1	GAAPI	Hybrid ant colony-genetic algorithm	(Ciornei ve Kyriakides 2012)
2	HPSO-BFGS	PSO-Broyden-Fletcher-Goldfarb-Shanno	(S. Li vd. 2011)
3	PSO-ACO	Particle swarm optimization-ant colony optimization	(Shelokar vd. 2007)
4	DE-BBO	Hybrid differential evolution-biogeography based optimization	(Gong, Cai, ve Ling 2011)
5	HABCDE	Hybrid artificial bee colony-differential evolution	(Xiang, Ma, ve An 2014)

Tümleşik hibrit algoritmalarda, ana algoritmanın bazı adımlarına bir diğer algoritma adapte edilir. Bu durumda ikinci algoritmanın katkı yüzdesi yaklaşık olarak %10-20 mertebelerinde olur. Bu kategoride tam ve kısmi manipülasyon olmak üzere 2 yöntem

mevcuttur. Tam manipülasyonda popülasyonun tamamı üzerinde ikinci algoritma ile gerçekleştirilmek istenen müdahale yapılırken, kısmi manipülasyonda sadece bir kısmına müdahale edilir. Tümleşik algoritmalardan bazıları örnek olarak Çizelge 2.6 'da verilmiştir.

Çizelge 2.6 Tümleşik Hibrit Algoritmalar

No	**Kısaltma**	**İngilizce Uzun Adı**	**Referans Kaynak**
1	CPSO	Cellular particle swarm optimization	(Shi vd. 2010)
2	TGA	Taguchi genetic algorithm	(Tsai, Liu, ve Chou 2004)
3	OGA	Orthogonal genetic algorithm	(Leung vd. 2001)
4	HABC	Hybrid artificial bee colony	(Kong, Liu, ve Wang 2013)
5	HMA	Hybrid memetic algorithm	(Y. Li vd. 2014)
6	HCS	Hybrid cuckoo search	(Long vd. 2014)
7	TC-ABC	Taguchi chaos-artificial bee colony	(Tien ve Li 2013)
8	CLA-DE	Cellular learning automata	(Vafashoar, Meybodi, ve Momeni Azandaryani 2012)
9	HPSO	Hybrid particle swarm optimization	(J. Wang 2012)
10	HDE	Hybrid differential evolution	(Yan, Guo, ve Gong 2011)

Literatürde yarı sezgisel algoritmalardaki gelişmeler, onların topolojileri ve diğerlerine göre karşılaştırılmaları inceleyen çeşitli çalışmalar mevcuttur (Nabaei vd. 2018), (Amaran vd. 2016), (Gai vd. 2018), (Ali Kaveh 2014). Bu çalışmalar incelenerek oluşturulan literatürde en çok karşılaşılan yarı sezgisel optimizasyon yöntemleri listesi, Çizelge 2.7 'de verilmiştir.

Çizelge 2.7 Literatürde sıkça karşılaşılan yarı sezgisel optimizasyon yöntemleri

No	Kısaltma	İngilizce Uzun Adı	Türkçe Uzun Adı
1	GA	Genetic Algorithm	Genetik Algoritma
2	PSO	Particle Swarm Optimization	Parçacık Sürüsü Optimizasyonu
3	DE	Differential Evolution Algorithm	Diferansiyel Gelişim Algoritması
4	ACO	Ant Colony Optimization	Karınca Kolonisi Optimizasyonu
5	ABC	Artificial Bee Colony	Yapay Arı Kolonisi
6	GSA	Gravitational Search Algorithm	Yer Çekimi Arama Algoritması
7	CSS	Charged System Search Algorithm	Yüklü Sistem Arama Algoritması
8	MCSS	Magnetic Charged System Search	Manyetik Yüklü Sistem Araması
9	FOF	Field of Forces Optimization	Kuvvetlerin Alanı Optimizasyonu
10	DEO	Dolphin Echolocation Optimization	Yunus Ekosu ile Yer Tespiti Optimizasyon Algoritması
11	RO	Ray Optimization Algorithm	Işın Optimizasyon Algoritması
12	MBB-BC	Modified Big Bang–Big Crunch Algorithm	Değiştirilmiş Büyük Patlama Büyük Çöküş Algoritması
13	CSO	Cuckoo Search Optimization	Guguk Kuşu Arama Algoritması
14	ICA	Imperialist Competitive Algorithm	Emperyalist Rekabetçi Algoritması
15	GWO	Gray Wolf Optimization	Gri Kurt Optimizasyonu
16	SOS	Symbiotic Organisms Search	Simbiyotik Organizmalar Arama
17	HSO	Harmony Search Optimization	Uyum Arama Optimizasyonu
18	KHO	Krill Herd Optimization	Kril Sürüsü Optimizasyonu

Bilgisayar ve elektrik mühendisliği alanında ünlü bir veri tabanı olan IEEE de yapılan sorgulamaya göre yarı sezgisel algoritmaların kullanıldığı çalışma sayıları Çizelge 2.8 ‘de verilmiştir.

Çizelge 2.8 Yarı sezgisel algoritmalarla yapılan çalışma sayıları (Nabaei vd. 2018)

Konferans	Yıl			
	2000 Öncesi	2001-2005	2006-2010	2011-Bugün
GA	5600	7167	18133	8700
PSO	45	710	6564	4173
ABC	0	0	68	347
HBMO	0	0	12	22
GSA	0	0	15	85
HSA	0	4	108	166
Dergi	2000 Öncesi	2001-2005	2006-2010	2011-Bugün
GA	1019	1399	1739	1238
PSO	3	76	409	512
ABC	0	0	2	24
HBMO	0	0	0	2
GSA	0	0	1	20
HSA	0	0	7	17

2.2. Kaotik Optimizasyon Teknikleri

Matematiksel olarak kaos, basit bir deterministtik sistem tarafından üretilen rastgelelik olarak tanımlanır (Ali Kaveh 2014). Rastgelelik, kaotik sistemin başlangıç koşullarına olan duyarlılığının bir sonucudur. Bu duyarlılık, başlangıç değerlerindeki ufak değişikliklerin, kararlı sabit noktalar, periyodik dalgalanmalar, çatallanmalar veya ergodiklik gibi birbirinden tamamen farklı davranışlara sebep olabileceği anlamına gelir. Bununla birlikte kaotik sistemler deterministiktir ve kaos düzenini ifade eder.

Yarı sezgisel algoritmalar, global optimumu ararken keşif ve sömürü olmak üzere genel olarak 2 temel strateji kullanırlar. Keşif, arama uzayındaki en iyi lokal çözüme ulaştırırken, sömürü ise, elde edilen lokal çözümün yakınlarında olabilecek global optimuma ulaştırmaya yarar. Keşif sırasında, keşfedilmemiş bölgelerin keşfedilmesi ve

arama alanının tüm bölgelerinin yeterince araştırıldığından emin olmak gerekirken, sömürme aşamasında, gelecek vaat eden bölgeler daha kapsamlı bir şekilde araştırılır (Ali Kaveh 2014).

Çizelge 2.9 Yarı sezgisel algoritmalar ve kaotik uygulamaları

No	**Kısaltma**	**Genel Kaotik Uygulaması**	**Yapısal Tasarım çalışmalarına ait Yayın**
1	GA	(Cheng vd. 2008)	(P. Guo, Wang, ve Han 2011)
2	PSO	(B. Liu vd. 2005)	(A. Kaveh vd. 2014)
3	DE	(Xiaohui Yuan vd. 2008)	(Tang, Xue, ve Fan 2008)
4	ACO	(Tian ve Jiang 2007)	Bulunamadı
5	ABC	(B. Wu ve Fan 2011)	Bulunamadı
6	GSA	(Gao vd. 2014)	Bulunamadı
7	CSS	(S. Talatahari, Kaveh, ve Sheikholeslami 2011)	(S. Talatahari, Kaveh, ve Sheikholeslami 2012)
8	MCSS	Bulunamadı	Bulunamadı
9	KHO	(Gharavian, Yaghoobi, ve Keshavarzian 2013)	Bulunamadı
10	DEO	Bulunamadı	Bulunamadı
11	RO	Bulunamadı	Bulunamadı
12	MBB-BC	(Rezaee Jordehi 2014)	Bulunamadı
13	CSO	Bulunamadı	Bulunamadı
14	ICA	(Bahrami, Faez, ve Abdechiri 2010)	(Siamak Talatahari, Kaveh, ve Sheikholeslami 2012)
15	GWO	(Kohli ve Arora 2018)	Bulunamadı
16	SOS	(Secui 2016)	Bulunamadı
17	HSO	(Xiaofang Yuan vd. 2014)	Bulunamadı

Kaos ve yarı sezgisel optimizasyon algoritmalarının ortak özelliklerinde ötürü, birlikte kullanılmaları durumunda yarı sezgisel algoritmaların limitlerinin üstesinden gelindiği ve performanslarının iyileştiği görülmektedir. Bu konudaki çalışmalar 2 tipte kategorize edilmiştir. Birinci tip çalışmalar, yarı sezgisel algoritmalardaki parametrelerin rastgele üretilme aşamalarına kaotik sistemlerin eklendiği çalışmalardır. İkinci tip ise, arama davranışını zenginleştirmek ve geleneksel kaotik optimizasyon algoritmaları kullanarak yerel optimum koşullarda sıkışıp kalmamak için yarı sezgisel algoritmalara dahil edilen uygulamalardır (Ali Kaveh 2014).

Aşağıda Çizelge 2.9' da yarı sezgisel algoritmalar ve bu algoritmalardan hangilerinin kaotik sistemler ile birlikte kullanıldıkları listelenmiştir. Kaotik sistem kullanımları hem genel olarak hem de yapısal tasarım olarak iki ayrı grupta değerlendirilmiştir. Her iki grup için de erişilen en eski yayın belirtilmiştir.

2.3. Yapısal Tasarım Çalışmalarında Kullanılmayan Kaotik Yarı Sezgisel Algoritmalar

Bir önceki kısımda sunulan Çizelge 2.9, literatürdeki yapısal tasarım çalışmalarında kullanılan ve kullanılmayan kaotik optimizasyon algoritmalarını göstermektedir. Bunlardan yapısal tasarım çalışmalarında kullanılmayanlar Çizelge 2.10 'da listelenmiştir. Canlı varlıklardan esinlenen algoritmaların büyük bir çoğunluğunun kaotik olarak yapısal tasarım çalışmalarında kullanılmadığı görülmüştür.

2.4. Kompleks Doğrusal Olmayan Model İçeren Yapısal Tasarım Çalışmalarında Karşılaşılan Zorluklar

Yapısal tasarım problemlerindeki doğrusalsızlıkları aşağıdaki 3 ana başlıkta toplanır (Simulia 2014).

- Sınır koşularındaki değişiklikler
- Doğrusal olmayan malzeme davranışı
- Büyük deformasyonlar

Çizelge 2.10 Yapısal tasarım uygulamalarında kullanılmayan kaotik ve yarı sezgisel algoritmalar

No	Kısaltma	Türkçe Uzun Adı
1	DE	Diferansiyel Gelişim Algoritması
2	ACO	Karınca Kolonisi Optimizasyonu
3	ABC	Yapay Arı Kolonisi
4	GSA	Yer Çekimi Arama Algoritması
5	MCSS	Manyetik Yüklü Sistem Araması
6	KHO	Kril Sürüsü Optimizasyonu
7	DEO	Yunus Ekosu ile Yer Tespiti Optimizasyon Algoritması
8	RO	Işın Optimizasyon Algoritması
9	MBB-BC	Değiştirilmiş Büyük Patlama Büyük Çöküş Algoritması
10	CSO	Guguk Kuşu Arama Algoritması
11	GWO	Gri Kurt Optimizasyonu
12	SOS	Simbiyotik Organizmalar Arama
13	HSO	Uyum Arama Optimizasyonu

Sınır koşularındaki değişiklikler ile anlatılmak istenen yapısal tasarım probleminde temasın yer alması. Yapının üzerine gelen yükün büyüklüğüne ve yönüne bağlı olarak yapının temas edip etmeyeceği ve ederse ne kadarlık bir kısmın temas edeceği sürekli değişeceği için bu tipteki yapısal tasarım problemlerinin doğrusal olarak çözülmeleri mümkün değildir.

Elastoplastik malzemeler için malzeme kaynaklı doğrusalsızlıklar, malzemenin kalıcı şekil değiştirmeye başladığı yani plastik bölgeye geçtiği durumlarda gerçekleşir. Bununla birlikte lastikler gibi hiçbir şekilde doğrusal bir davranış göstermeyen malzemelerin kullanıldığı yapısal tasarım problemlerinde ise çözüm en başından itibaren doğrusal değildir.

Son olarak yapısal tasarım probleminin çözümü öncesinde, deforme olmamış boyutlara göre yazılan denge durumu, kuvvet ve moment dağılımları, yapının üzerine gelen yük

altındaki deformasyonu sonrasında değişiklik gösterir. Deformasyonun küçük olduğu durumlarda bu fark ihmal edilebilir. Deformasyonun artmasıyla bu durumun dikkate alınması ihtiyacı doğar. Bu da yapısal tasarım problemini doğrusal olmaktan çıkarır.

Bu tip doğrusal olmayan problemlerde karşılaşılan problemler ise aşağıdaki şekilde gruplanabilir.

- Burkulma veya göçme
- Lokal kararsızlıklar

Burkulma (buckling) veya göçme (collapse) yapının belli bir yük değerinden sonra direnç gösterememesi, üzerine gelen yük azalsa bile şekil değiştirmeye devam etmesidir. Bu duruma gelen doğrusal olmayan bir yapısal tasarım probleminin çözümü normal yöntemlerle mümkün değildir. Bu tip problemlerin çözümü için özel yöntemler geliştirilmiştir. Lokal kararsızlıklar ise lokal burkulma ve malzemedeki kararsızlıklardan dolayı oluşur. Bu gibi durumlarda önceki tip için uygulanan özel çözüm yöntemleri çözümün ilerleyebilmesi için yeterli değildir. Bu tip problemler, komple modele sönüm uygulayarak kararlı hale getirilirler. Modele eklenen viskoz kuvvetler, problemi kararlı hale getirmeye yeterken genel davranışını etkilemeyecek bir büyüklüktedirler (Simulia 2014).

Doğrusal olmayan yapısal tasarım problemindeki burkulma ve lokal kararsızlıkların çözümü için geliştirilen yöntemler çözüm sürelerini uzattıkları gibi çözüm için oluşturulan sonlu elemanlar modellerinin de neredeyse hatasız bir şekilde kurulmalarını zorunlu hale getirir. Modeldeki en küçük bir hata, çözüm sırasındaki zaman adımını küçültmeye başlar. Bu da iterasyon sayısını ve dolayısıyla da çözüm sürelerini uzatmaya başlar. Modelleme hatasından kaynaklanan lokal kararsızlık bulunan bir modelin çözümü saatler sürerken lokal kararsızlığa neden olan durumun ortadan kaldırılmasından sonra aynı sonlu elemanlar modeli dakikalar içerisinde çözüme ulaşabilmektedir.

Bu nedenle yarı sezgisel algoritma gibi çok fazla sayıda deneye ihtiyaç duyacak optimizasyonlarda kullanılacak doğrusal olmayan sonlu elemanlar modellerinin, yukarıda açıklanan problemlerden arındırılmış olması zorunludur. Aksi halde çözüm süreleri gerçekçi olmaktan uzaklaşacaktır.

2.5. Kaotik Optimizasyon Tekniğinin Seçimi

Tez çalışması için kril sürüsü optimizasyon yöntemi seçilmiştir. Literatürde kaotik uygulamalarının az sayıda olması ve otomotivdeki yapısal tasarım problemleri üzerinde sonlu elemanlar analizleri ile birlikte gerçekleştirilmiş bugüne dek yapılan herhangi bir çalışmanın bulunmaması bu yöntemin seçiminde etkili olmuştur.

3. MATERYAL ve YÖNTEM

Tez çalışmasının bu bölümünü otomotivde ürün geliştirme süreci ve üzerinde çalışılacak olan kaotik kril sürüsü (KKS) yöntemi hakkında detaylı bilgilendirmelerin yer aldığı aşağıdaki bölümleri içermektedir.

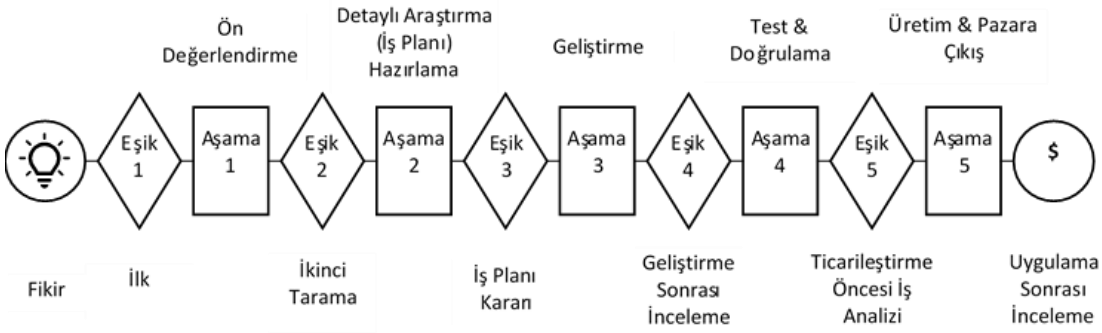
- Otomotivde Ürün Geliştirme Süreci ve Optimizasyon
- Kaotik Optimizasyon Tekniği
- Biyo-Coğrafya Temelli Kril Sürüsü Optimizasyonu
- Kaotik Kril Sürüsü Optimizasyonu

3.1. Otomotivde Ürün Geliştirme Süreci ve Optimizasyon

Yeni ürün geliştirme yeni bir fikrin ortaya çıkmasıyla başlayıp bu fikrin ticarileştirilmiş bir ürüne dönüştürülmesine kadar uzanan bir süreçtir. Bu başlangıç ve sonuç noktaları arasındaki süreç belli bazı aşamalara bölünmüştür (Cengiz, Ayyildiz, ve Kirkbir 2006). Aşamalar sürecin kritik bazı göstergelerinin durulup incelenmesi gereken önemli noktalarıdır. Bir aşamadan sonraki aşamaya geçilip geçilmeyeceği, yani sürece devam kararının veya süreci durdurma kararının verileceği aşama sonu değerlendirme noktalarına eşik adı verilir.

Yeni ürün geliştirme sürecinde aşama eşiği (Stage Gate) kavramı Johnson ve Jones'un 1957 yılında yayınladıkları makalelerinden bugüne kadar birçok araştırmacı tarafından tartışılmış ve farklı versiyonlarıyla güncelleştirilmiştir (Cengiz, Ayyildiz, ve Kirkbir 2006). En yaygın olarak kullanılan haline ise Cooper ve ark. tarafından getirilmiştir. Cooper ve ark. tarafından sunulan, Şekil 3.1 ‘de gösterilen yöntem, aşasıdaki aşama ve eşiklerden oluşmaktadır.

İlk gözlem, pazar değerlendirme ön hazırlığı, teknik değerlendirme ön hazırlığı, detaylı pazar araştırması, finansal analiz, ürün geliştirme, gözden geçirme, test ve doğrulama, ticarileştirme öncesi iş analizi, ürünün üretimine başlama, pazara sunma ve uygulama sonrası incelemedir (Cooper 1990);(1994).

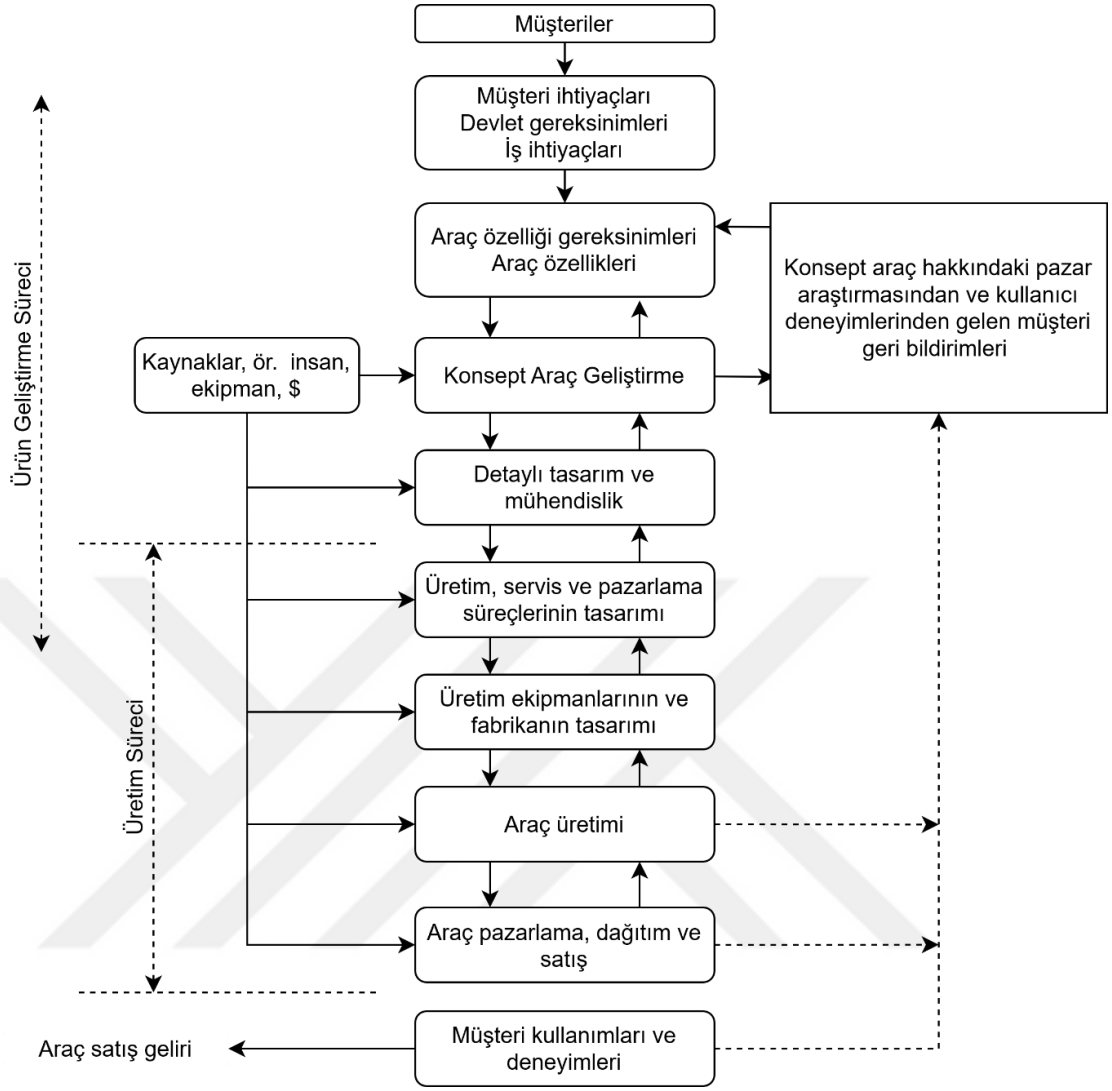


Şekil 3.1 Aşama Eşiği Yöntem (Cooper 1990)

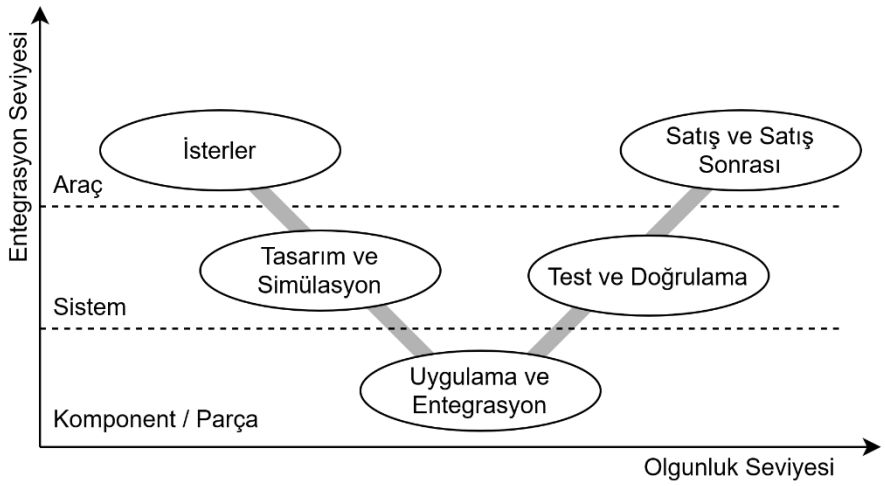
Otomotivde ürün geliştirme sürecini özetleyen akış şeması aşağıda Şekil 3.2' de verilmiştir (Bhise 2017). Burada görüldüğü üzere otomotivde ürün geliştirme süreci müşterilerin ve onların ihtiyaçlarının belirlenmesiyle başlamaktadır. Ardından bu ihtiyaçların ürün özelliğine bir başka deyişle teknik ifadelere dönüşmesiyle devam eder. Ürün tasarım süreci ve mühendislik çalışmaları ise bu noktadan sonra başlar.

Otomotivdeki tasarım sürecini en iyi ifade eden bir diğer model ise V modeli (Şekil 3.3) olarak adlandırılır (Johanson 2012), (Weber 2013), (Bhise 2017). V modeli, ilk olarak 1997 yılında yazılım geliştiricileri için ortaya çıkmıştır (Pfeffer vd. 2019), (Höhn ve Hoeppner 2008). Burada komple araç seviyesinde belirlenen müşteri beklentileri veya ihtiyaçlar, sırasıyla, sistemlere, oradan alt sistemlere ve oradan da parça seviyesine indirgenir. Ardından en alt seviyedeki parça tasarımı ve doğrulama çalışmaları gerçekleştirilir. Bu kez de en alt seviyedeki parça seviyesinden en üst seviyedeki komple araca kadar olan tüm alt sistem ve sistemler sırasıyla test edilir ve bu sayede ürün doğrulanır. V'nin sol tarafı tasarım ve mühendislik faaliyetlerini içerirken sağ tarafı ise, doğrulama, üretim ve montaj süreçlerini içerir (Bhise 2017).

Şu ana kadar önerilen tüm ürün geliştirme yöntemleri aşama aşama ilerleyen geleneksel ürün geliştirme yöntemleridir. Bu yöntemlerde ürün geliştirme sürecinin ardından doğrulama testleri gerçekleştirilir.



Şekil 3.2 Otomotivde Ürün Geliştirme ve Üretim Süreci Akış Diyagramı (Bhise 2017)



Şekil 3.3 Otomotiv ürün geliştirme sürecinde V Modeli (Johanson 2012)

Oysa ki yenilikçi yöntemlerde bu süreçler iç içe geçmiştir ve birlikte gerçekleşir (Goelke 2017) . Bu yöntem, Simülasyon Tabanlı Ürün Tasarımı olarak adlandırılır. Simülasyon tabanlı ürün tasarımı aşağıdaki aşamalardan oluşur.

- Tasarım uzayının belirlenmesi
- Topoloji optimizasyonu
- Topolojiye uygun endüstriyel tasarımın oluşturulması
- Şekil Optimizasyonu
- Optimize edilen şeklin endüstriyelleştirilmesi
- Kabartma optimizasyonu
- Nihai tasarımın oluşturulması
- Doğrulama analizleri

3.1.1. Tasarım Uzayının Belirlenmesi

Bu aşamada tasarlanacak parça için kullanılabilecek boşluk başka bir deyişle geometrik yer oluşturulur. Bu boşluk belirlenirken çevre parçaları ile olan ilişi ve yakınlığa dikkat edilir. İleriki aşamalarda şekillenecek tasarım mutlak suretle kendine ayrılan bu alanın içerisinde kalacaktır.

3.1.2. Topoloji Optimizasyonu

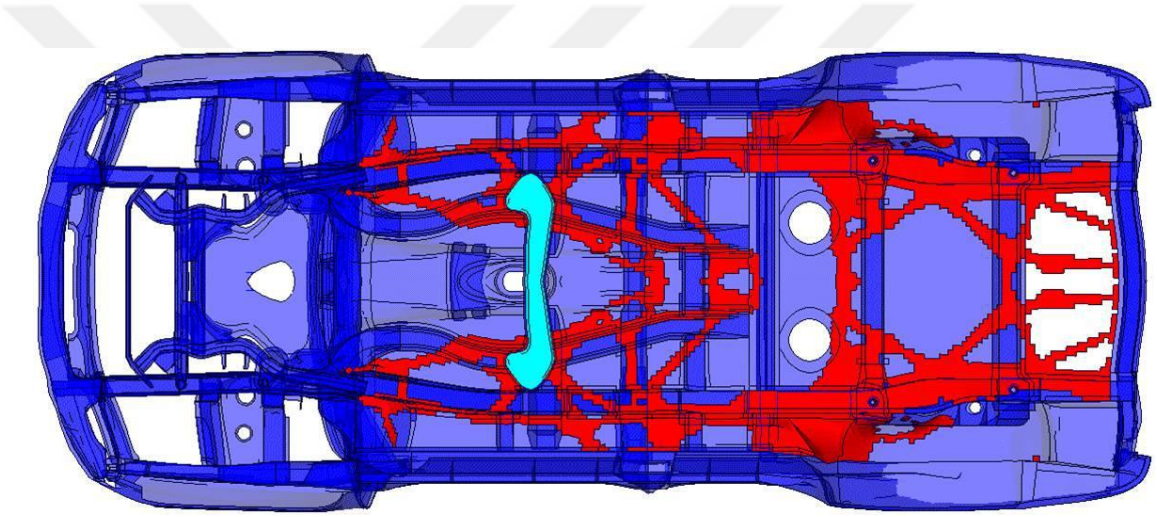
Topoloji optimizasyonunun amacı ürün geliştirme sürecinin çok erken safhalarında ürün ile ilgili temel boyutların belirlenmesinin ardından en uygun konseptin ve buna bağlı olarak ana yük yollarının belirlenmesidir (Duddeck 2012). Belirlenen tasarım uzayı ve parçanın kendisinden beklenen performans hedefleri ile gerçekleştirilen bir optimizasyon sürecidir. Bu aşamanın ardından belirlenen boşlukta istenen hedeflere ve amaca en uygun topoloji elde edilir (Bendsoe ve Kikuchi 1988), (W. Zhang, Zhu, ve Gao 2016), (Li Zhaokun ve Zhang Xianmin 2018).

Komple araç çarpışma analizleri gibi, büyük deformasyonların ve yüksek dereceden doğrusalsızlıkların olduğu uzun süreler alan optimizasyon problemlerinin çözümünde

ekseriyetle kullanılan yöntemler aşağıda listelenmiştir (Duddeck 2012), (Aulig vd. 2016). Bunlar;

- Matematiksel olarak tanımlanan nodal eşdeğer statik yükler (Equivalent Static Loads, ESL)
- Hibrit hücresel otomatlar
- Tek bir statik yük
- Diğer yöntemler

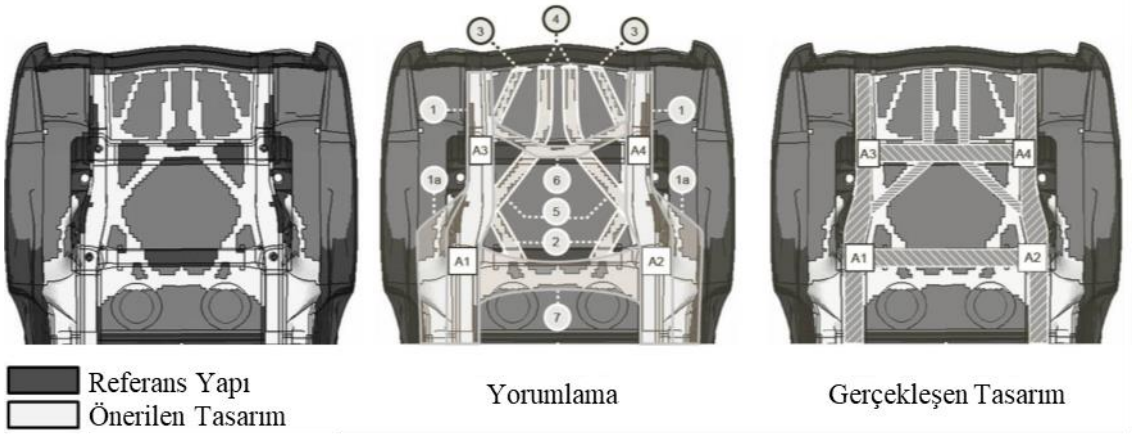
Şekil 3.4' te otomotiv alt gövdesi üzerinde gerçekleştirilen bir topoloji optimizasyonu örneği gösterilmiştir.



Şekil 3.4 Tüm araç alt gövde topoloji optimizasyonu sonucu (Volz 2011)

3.1.3. Topolojiye Uygun Endüstriyel Tasarımın Oluşturulması

Topoloji optimizasyonu sırasında üretim kısıtlarını da tanımlamak günümüz şartlarında mümkündür. Oluşturulan tasarımın az sayıda üretilecek olması durumunda, 3 boyutlu yazıcılarda doğrudan üretilebilir. Ancak, çok sayıda seri üretimi yapılacak tasarımlar için, topoloji optimizasyonu sonrasında elde edilen şekil, seri üretim şartlarına uygun değildir. Bu nedenle bu şeklin seri üretim şartlarına uygun olacak şekilde yeniden 3 boyutlu olarak tasarlanması gerekir.



Şekil 3.5 Topoloji optimizasyonu sonrasında elde edilen topoloji ve buna göre oluşturulan tasarım

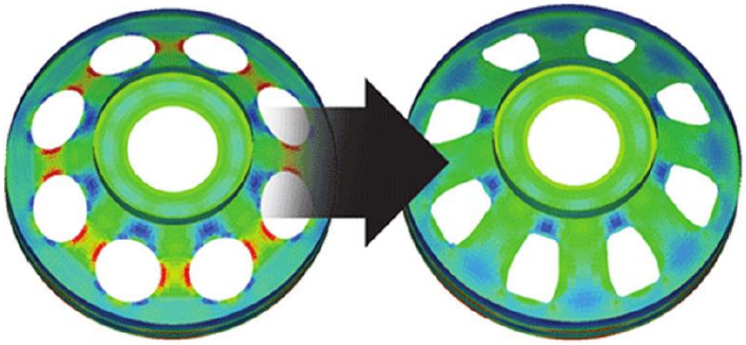
3.1.4. Şekil ve Boyut Optimizasyonu

Bu aşamada seri üretim şartlarına uygun olarak oluşturulan tasarımın şekli değiştirilerek istenilen kısıtları sağlayan ve amaca daha uygun bir tasarım elde edilmeye çalışılır (Bendsoe ve Kikuchi 1988). Örnek olarak daire şeklinde oluşturulan bir boşaltma yumuşatılarak istenilen amaca daha uygun bir tasarım elde edilebilir (Li Zhaokun ve Zhang Xianmin 2018), (W. Zhang, Zhu, ve Gao 2016). Bununla birlikte sac parçaların kullanıldığı tasarımlarda optimizasyon sürecine kalınlık da katılarak optimum şekil ve kalınlığın birlikte oluşturulması sağlanır.

Şekil optimizasyonu günümüzde kullanılan ticari yazılımlarda 2 şekilde gerçekleştirmek mümkündür (Altair Inc. 2018). Bunlar;

- Serbest şekil yöntemi
- Şekil dönüştürme yöntemi (morph)

Aşağıda Şekil 3.6'da ince cidarlı bir yapının delik bölgesinde gerçekleştirilen bir şekil optimizasyonu gösterilmiştir.



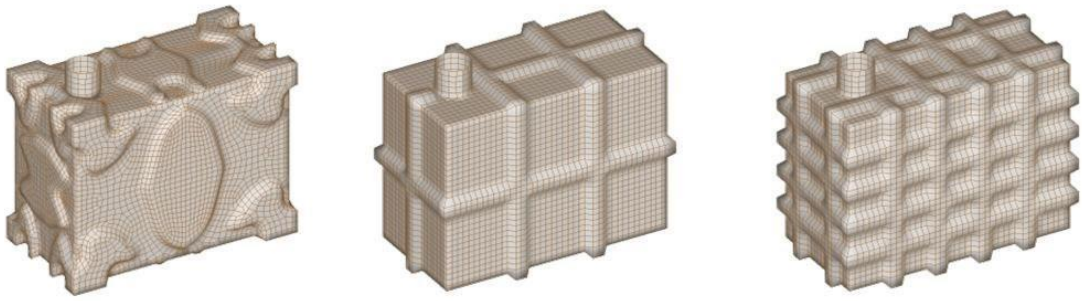
Şekil 3.6 İnce cidarlı yapılarda şekil optimizasyonu (W. Zhang, Zhu, ve Gao 2016)

3.1.5. Optimize Edilen Şeklin Endüstriyelleştirilmesi

Topoloji optimizasyonu sonrasında olduğu gibi yine şekil optimizasyonu sonrasında da elde edilen şeklin seri üretim şartlarına uygunluğu kontrol edilmeli ve tasarım buna göre yenilenmelidir.

3.1.6. Kabartma Optimizasyonu

Kabartma (bead) ya da başka bir deyişle topoğrafya (topography) optimizasyonu genellikle kabuk ve ince cidarlı yapılara onları güçlendirmek için uygulanır. Sonrasında kabuk yapılar üzerinde optimize edilmiş bir kabartma formu elde edilir. Aşağıda, Şekil 3.7 'de, bir sıvı tankında yapılan kabartma optimizasyonu gösterilmiştir (Altair Inc. 2018). Burada, kabartma miktarını artırmanın yapının direngenliğini artırmadığı görülmektedir.



Maksimum deformasyon 7.54 mm Maksimum def. 10.8 mm Maksimum def. 13.9 mm

Şekil 3.7 Sıvı Tankı kabartma optimizasyonu (Altair Inc. 2018)

3.1.7. Nihai Tasarımın Oluşturulması

Kabartma optimizasyonu sonrasında ortaya çıkan form, üretilebilirlik şartlarına göre yeniden 3 boyutlu olarak modellenir.

3.1.8. Doğrulama Analizleri

Oluşturulan nihai tasarım ile parçanın kendisinden beklenen performans hedefleri son olarak kontrol edilir. Bu analizlerin ardından tasarlanan parçaya ait doğrulama süreci de tamamlanmış olur.

3.2. Otomotiv Tasarım ve Optimizasyonunda Karşılaşılan Zorluklar

Otomotiv parçalarında, kompleks geometrileri nedeniyle boyut optimizasyonu yöntemi çok sıklıkla kullanılan bir yöntem değildir. Bununla birlikte topoloji, topoğrafya ve şekil dönüştürme yöntemleri daha fazla kullanılır. Ancak kalınlığın boyut parametresi olarak kullanıldığı boyut optimizasyon uygulamalarına da literatürde rastlanır. Her iki durumda dahi parçaların geometrilerinin karmaşıklığı geometriyi kontrol eden parametre sayısının fazlalığına neden olacaktır. Komple araç modelleri üzerinden yapılan optimizasyon çalışmalarında ise parça sayısının fazlalığı nedeniyle optimize edilecek parametre sayısında artışa neden olacaktır. Bu da optimizasyon çözümünün global optimumu bulmasında hem zora sokacaktır hem de çözüm süresini oldukça uzatacaktır.
Otomotiv sektöründeki optimizasyon çalışmalarında parçaların kompleks geometride olmalarının yanında karşılaşılan diğer zorluklar ise şunlardır;

- Doğrusal olmayan çözümler
    - Aşırı deformasyon
    - İçerisinde temasın olması nedeniyle değişken sınır koşulları
    - Plastikler, elastomerler veya plastik bölgeye giren her türlü malzeme davranışı
- Kısıt sayısının çokluğu nedeniyle her bir deney için birden fazla modelin çözümüne ihtiyaç duyulması
- Amaç sayısının çokluğu

Bütün bu zorluklar öncelikle her bir adım ve her bir deneme için harcanacak çözüm sürelerinin artışına neden olacaktır (Ali Kaveh 2014). Çözüm sürelerinin yüksek olması nedeniyle az sayıda tekrarla çözüm üretebilen ve genellikle türev tabanlı optimizasyon yöntemleri olarak adlandırılan optimizasyon yöntemleri kullanılır. Bu durumda da elde edilen çözümün global optimum noktasına ait olduğunu söyleyebilmek oldukça zordur (J. S. Arora 2017). Global optimumu elde edebilmek için, genetik algoritma (GA) ve parçacık sürüsü (PSO) gibi yarı sezgisel olarak adlandırılan çok sayıda optimizasyon yöntemleri geliştirilmiştir. Hatta farklı yöntemlerin bir arada kullanıldığı hibrit optimizasyon yöntemleri de günümüzde araştırmacılar tarafından geliştirilmeye devam etmektedir (Ali Kaveh 2014). Ancak bu yöntemler de çok sayıda denemeye ve adıma ihtiyaç duyarlar. Bu da çözüm süresinin daha da artmasına neden olur.

Nabaei ve ark. mühendislik problemlerinde kullanılan birçok sezgisel algoritmanın topolojilerini incelemiş ve performanslarını birbirleriyle kıyaslamışlardır. Son olarak da gelecekte yapılacak çalışmalar için aşağıdaki öngörülerde bulunmuşlardır (Nabaei vd. 2018).

- Gerçek zamanlı problemlerin çözümü için hızlı ve güçlü optimizasyon yöntemleri
- Ekstra operatör kullanmayan sadece denklemlerin modifikasyonuyla lokal ve global optimumların bulunmasının iyileştirilmesinde yenilikçi yaklaşımlar
- Kaos ve hibrit gibi algoritmalar gibi harici operatörlerin kullanılması
- Hesaplama zamanının azaltılması için kullanışlı matematiksel operatörlerin bulunması ve uygulanmasına devam edilmesi

Üzerinde çalışılan mühendislik probleminde kullanılan optimizasyon yöntemi, az sayıda deney ile hızlı bir şekilde çözüme ulaşabilmelidir. Yukarıda bahsedilen doğrusalsızlıklar, farklı kısıtlar ve hedefler için birden fazla çözülmesi gereken sayısal modelin varlığı nedeniyle optimizasyon yönteminin etkinliği kısa bir sürede çözüme ulaşabilmek için büyük önem taşıyacaktır. Aksi taktirde gerçek endüstriyel problemlerde kullanım şansı bulması oldukça zordur.

Bununla birlikte kullanılan optimizasyon yöntemi, elde edilen çözümün, lokal bir minimum noktası değil de global minimum olduğunu garanti edebilmelidir. Bu sayede en doğru çözüme ulaşılmış olunacaktır.

3.3. Kaotik Optimizasyon tekniği

Kuşların kolektif davranışları, arıların arama faaliyeti veya karıncaların kooperatif davranışları gibi doğadaki karmaşık biyolojik olaylar aslında göreceli olarak basit ve başlangıç koşullarına duyarlı doğrusal olmayan davranışa sahip olan kuralların bir sonucudur. Bu tür sistemler genellikle "deterministik doğrusal olmayan sistemler" olarak bilinir ve buna karşılık gelen teori de "kaos teorisi" olarak adlandırılır. Böylece rastlantısal veya rastgele görünen gerçek dünya sistemleri, doğrusal olmayan deterministik ve kaotik bir davranış olarak ifade edilebilir. Kaos ve rastgele sinyaller uzun vadeli tahmin edilemeyen düzensiz davranış özelliği gösterseler de kaotik haritalar ve programlama dillerindeki birçok rasgele üretici aslında deterministiktir. Ancak kaos düzensizlikten doğma düzenine yardımcı olabilir (Ali Kaveh 2014).

Benzer şekilde, birçok yarı sezgisel optimizasyon algoritması, düzenin düzensizlikten doğduğu biyolojik sistemlerden ilham almaktadır. Bu durumlarda bozukluk genellikle hem örgütlenmemiş örüntüleri hem de düzensiz davranışları gösterir; oysa düzen, öz-örgütlenme ve evrimin bir sonucudur ve genellikle bir hastalık durumundan veya asimetrinin varlığından kaynaklanır. Öz organizasyon ve evrim, birçok yarı sezgisel optimizasyon tekniğinin iki anahtar faktörüdür. Kaos ve optimizasyon algoritmaları arasındaki bu ortak özellikler nedeniyle, bu kavramların aynı anda kullanılması optimizasyon algoritmalarının performansını iyileştirebilir (Sheikholeslami ve Kaveh 2013). Görünüşte, bu kombinasyonun yararları, diğer rastlantısal optimizasyonlar için de geçerlidir ve deneysel çalışmalar bunu doğrulamıştır. Bununla birlikte, bu durum matematiksel olarak henüz kanıtlanmamıştır (Tavazoei ve Haeri 2007).

Son zamanlarda, kaos ve yarı sezgisellik farklı çalışmalarda birleştirilmiştir. Eserlerden bazıları, yarı sezgisel algoritmalardaki kaotik davranışları göstermeyi amaçlarken, bazılarında ise, yarı sezgisel algoritmaların sınırlarının üstesinden gelmek için kaos

kullanılmıştır. Bu nedenle önceki araştırmalar iki tipte sınıflandırılabilir (Kaveh Ali 2014).

Birinci tipte, yarı sezgisel algoritmada rasgele sayı üreteci yerine kaos eklenir. Yani, yarı sezgisel denklemlerdeki parametrelerin değerlerini kontrol etmek için kullanılır. Yarı sezgisel algoritmaların yakınsama özellikleri, çözüm sırasında operatörleri üzerinde uygulanan rasgele diziye yakından bağlıdır. Deneyimler, farklı rasgele sayılarla bazı optimizasyonlara başlandığında elde edilen sonuçların birbirlerine çok yakın ancak eşit olmadığını, bununla birlikte aynı optimum değere ulaşmak için farklı sayıda nesiller gerektirebileceğini göstermektedir. En çok kullanılan yarı sezgisel araçlarının dayandığı rasgele sayı üretme algoritmaları genellikle ki-kare ya da normallik gibi bazı istatistiksel testleri yerine getirir. Bununla birlikte, belirli bir rasgele sayı üretecinin seçimine bağlı olarak, yarı sezgisel algoritmaların performans endekslerinin iyileştirilmesini garanti eden bir analitik sonuç yoktur (Kaveh Ali 2014).

İkinci tipte, arama davranışını zenginleştirmek ve yerel optimum koşullarda sıkışıp kalmamak için yarı sezgisel algoritmalara kaotik arama dahil edilmiştir. Kaos değişkenlerinin avantajlarına dayanan rastlantısal bir arama tekniği olan geleneksel bir kaos optimizasyon algoritması (COA) geliştirilmiştir. COA 'nın basit felsefesi iki ana aşamadan oluşur. Bunlar; kaotik alandan çözüm alanına eşleştirme ve ardından rasgele arama yerine kaotik dinamikleri kullanarak en uygun bölgeleri aramadır (Liu, et al. 2005). Bununla birlikte, COA 'nın bazı dezavantajları da vardır. Örneğin, büyük ölçekli optimizasyon problemlerinde, algoritmanın etkinliği çok düşük olacaktır ve COA, genel olarak optimum seviyeye ulaşmak için sıklıkla çok sayıda iterasyona ihtiyaç duyar.

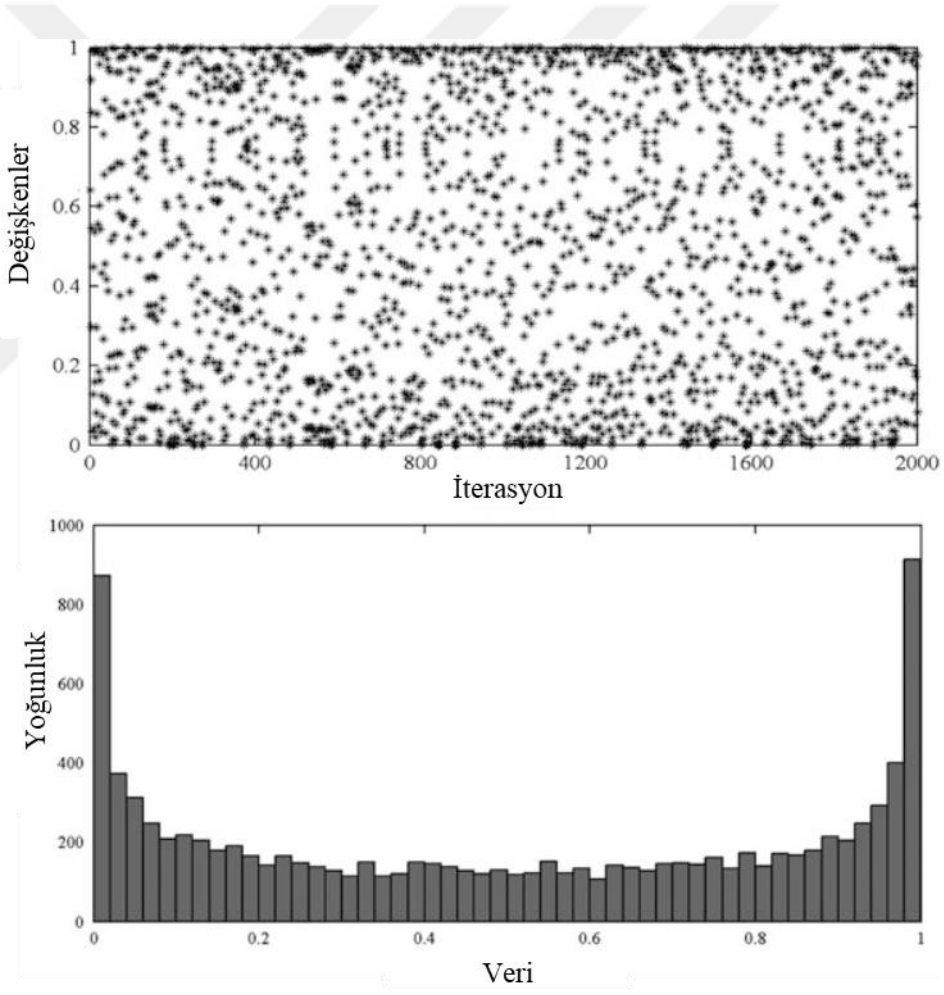
3.4. Kaotik Sistemlere Giriş

Matematikte kaos basit tanımlı sistemler tarafından üretilen “rastgelelik” olarak tanımlanmaktadır. Rastgele olma, kaotik sistemlerin başlangıç koşullarına duyarlı olmasının bir sonucudur. Yani, veri üreten fonksiyonun parametrelerindeki veya verilerin başlangıç değerlerindeki küçük değişikliklerin, kararlı sabit noktalar, periyodik

salınımlar, çatallanmalar ve ergodiklik gibi oldukça farklı davranışlara yol açtığı anlamına gelir (Kaveh Ali 2014).

Ancak, kaotik sistemler deterministik olduğundan, kaos düzen anlamına da gelir. Bir sistemin, düzenli bir periyodik sistemden karmaşık bir kaotik sisteme dönüşümü, sadece kontrol parametrelerinden birini değiştirerek yapabilir. Ayrıca kaotik bir hareket, belirli bir bölgedeki her durumda kendi sistemine göre geçebilir ve her durum yalnızca bir kez elde edilir (S. Liu ve Hou 2002).

Kaotik haritalamanın bir örneği Şekil 3.8' da gösterilmiştir.



Şekil 3.8 Kaotik Haritalamaya bir örnek (Logistic Map (Kaveh Ali 2014))

Ayrık zamanlı bir seri göz önüne alındığında, Li-Yorke anlamında kaos tanımlanabilir. Bir boyutlu yinelenmiş harita, bir değişkenli bir fonksiyon olarak tanımlanabilir ve aşağıdaki şekli alır.

$$x_{t+1} = F(x_t) \tag{3.1}$$

Burada $x_t \in R^n, t = 1,2,3, \ldots$ ve F ise R^n 'in bir fonksiyonudur. $F^{(p)}$, F 'nin p > 0 kez üssü olarak tanımlandığında, $F^{(p)}(x) = x$ ancak, tüm $k \leq p$ için $F^{(k)}(x) \neq x$ koşulunu sağlayan x noktası, F'nin p-periyodik noktası olarak adlandırılır. Özellikle, $F(x) = x$ koşulunu sağlayan x noktası F 'nin sabit noktası olarak adlandırılır. X noktasının ε komşusu $N_\varepsilon(x)$, şu şekilde tanımlanır.

$$N_\varepsilon(x) = \{y \in R^n | \|x - y\| \leq \varepsilon\} \tag{3.2}$$

Burada $\|.\|$, R^n içinde Euclidean normunu ifade eder. Li-Yorke 'un kaos tanımı şu şekilde denkleme eklenir (T. Y. Li ve Yorke 1975)

Tanım 1: Ayrık zamanlı bir seri aşağıdaki koşulları sağlıyorsa kaotik olarak adlandırılır (Kaveh Ali 2014).

1. Herhangi bir p ≥ N için F'nin p-periyodik noktası olduğu pozitif bir N sabiti vardır.
2. Herhangi bir periyodik F noktası içermeyen ve aşağıdaki koşulları sağlayan, sayılamayan bir $S \subset R^n$ kümesi vardır.
   a. $F(S) \subset S$
   b. Her $x, y \in S (x \neq y)$ için

      $$\lim_{n\to\infty} sup\left\|F^{(n)}(x) - F^{(n)}(y)\right\| > 0,$$

      ve herhangi bir $x \in S$, F'nin herhangi bir y periyot noktası için

      $$\lim_{n\to\infty} sup\left\|F^{(n)}(x) - F^{(n)}(y)\right\| > 0,$$

   c. Herhangi bir x, y ∈ S_0 için, sayılamayan bir $S_0 \subset S$ alt kümesi vardır

      $$\lim_{n\to\infty} inf\left\|F^{(n)}(x) - F^{(n)}(y)\right\| = 0$$

Yukarıdaki tanımda yer alan S setine, karıştırılmış set denir.

Sistemdeki ani reaksiyon (Snap-Back) kovucusu adı verilen sabit bir noktanın varlığı, sistemin Li-Yorke anlamında kaotik olduğu anlamına gelir (Tatsumi, Obita, ve Tanino 2009). Bu nedenle, bir sistem sonsuz sayıda periyotları göreceli olarak büyük periyodik yörüngeleri içeriyorsa kaotiktir. Bu tanım esasen 1964 yılında Rus matematikçi Sarkovski 'nin ispatladığı Sarkovski teoreminin bir sonucudur. Ancak, çağdaş bilimsel anlamıyla kaos kelimesi ilk olarak Li ve Yorke tarafından sunulan bir makalede yer almıştır (T. Y. Li ve Yorke 1975), (Hilborn 1994).

Kaotik bir harita rastgele bir sayı dizisi için yayılı spektrum dizisi olarak kullanılabilir. Kaotik dizilerin kolay ve hızlı bir şekilde oluşturulabilmesi nedeniyle uzun dizilerin depolanmasına gerek yoktur. Oldukça uzun diziler için sadece birkaç fonksiyon (kaotik haritalar) ve birkaç parametreye (başlangıç koşulları) ihtiyaç duyulur. Ayrıca, başlangıç koşulları değiştirilerek çok sayıda farklı seri üretilebilir. Ek olarak, bu diziler deterministik ve yeniden elde edilebilir serilerdir. Kaotik dizilerin seçimi, yayılı spektrum karakteristiklerine ve ergodiklik özelliklerine karşılık gelen tahmin edilemezlikleri ile teorik olarak haklı gösterilebilir (Heidari-Bateni ve Mcgillem 1994). Bu nedenle, rastgele bir sayı gerektiğinde, çalışmanın ilk adımında, seçilen bir kaotik harita ile rastgele bir başlangıç koşulları kullanılarak tek bir adımda üretilebilir.

Literatür, kaotik zaman serisi dizileri bakımından oldukça zengindir. Bu haritalardan en etkili olduğu belirtilen (Feng vd. 2017) ve literatürde sıklıkla karşılan haritalar sonraki alt bölümlerde listelenmiştir.

3.4.1. Lojistik Harita

Bu harita fonksiyonu, ilk olarak Robert May tarafından 1976 yılında biyolojik popülasyonun doğrusal olmayan dinamikliği ve bunun neden olduğu kaotik davranışı ispatlamak için geliştirilmiştir (May 1976). Bu fonksiyon aynı zamanda Kaotik Optimizasyon algoritmaları içinde en fazla kullanılanlardandır ve aşağıdaki şekilde tanımlanır.

$$x_{n+1} = \mu x_n(1 - x_n) \tag{3.3}$$

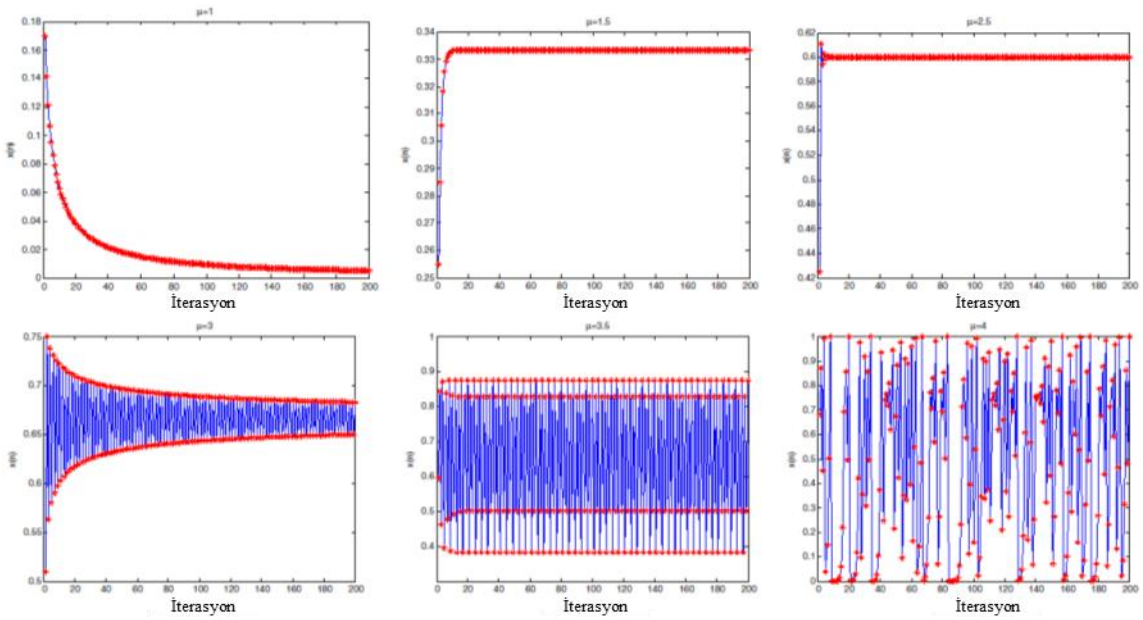
Bu denklemde, n iterasyon sayısını, x_n n'inci kaotik sayıyı temsil etmektedir. μ ise 0ile 4 arasında ($0 < \mu < 4$) sabit bir sayıdır. μ ye farklı değerler verildiğinde farklı kaotik seriler elde edilecektir. Farklı μ değerleri ile elde edilen kaotik seriler **Şekil 3.9**'da gösterilmiştir. Simülasyon sonuçları $\mu = 4$ olması durumunda en iyi kaotik seriye ulaşıldığını göstermiştir (Feng vd. 2017).

(3.3) numaralı denklemde $x_{n-1} = 0.25$ için $x_n = 4 \times 0.25 \times (1 - 0.25) = 0.75$ olur. Ardından $x_{n+1} = 4 \times 0.75 \times (1 - 0.75) = 0.75$ değeri elde edilir. Bu durum sonraki iterasyonlarda da bu şekilde ilerleyerek 0.75 sabit sayısını vermeye devam eder. Aynı durum $x_{n-1} = 0.5\ , 0.75$ değerleri için de geçerlidir. Bu durumlarda elde edilen diziler kaotik değildir.

Bu durumdan kurtulabilmek için ilk başlangıç x değeri aşağıdaki şekilde değiştirilir.

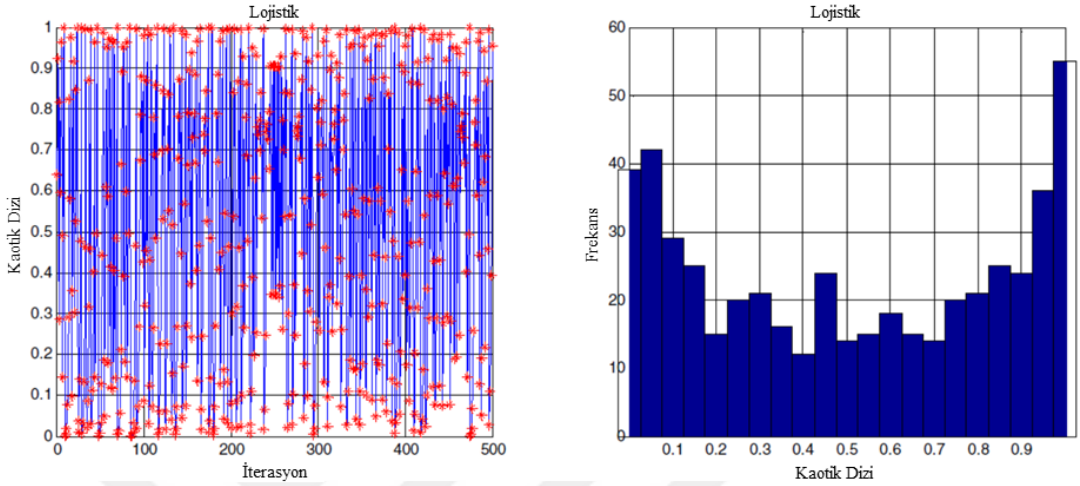
$$x_n = x_n + 0.1 \times r \tag{3.4}$$

Buradaki r parametresi 0 ile 1 arasındaki rastgele bir sayıdır. Denklem (3.4) başlangıç $x_n \in (0.25,0.5,0.75)$ değerleri için bile kaotik seri üretebilmeyi garanti edecektir.



Şekil 3.9 Farklı μ değerleri ile elde edilen Lojistik Harita Serileri (Feng vd. 2017)

Lojistik harita fonksiyonunun 500 iterasyonu sonrasında elde edilen kaotik seriye ait serpilme ve çubuk dağılım grafikleri Şekil 3.10 ‘de gösterilmiştir.



Şekil 3.10 Lojistik harita fonksiyonu ile üretilen seriye ait serpilme ve çubuk dağılım grafikleri (Feng vd. 2017)

3.4.2. Sinüzoidal Harita

Bu fonksiyon da May tarafından 1976 yılında geliştirilmiştir ve aşağıdaki şekilde ifade edilir (May 1976).

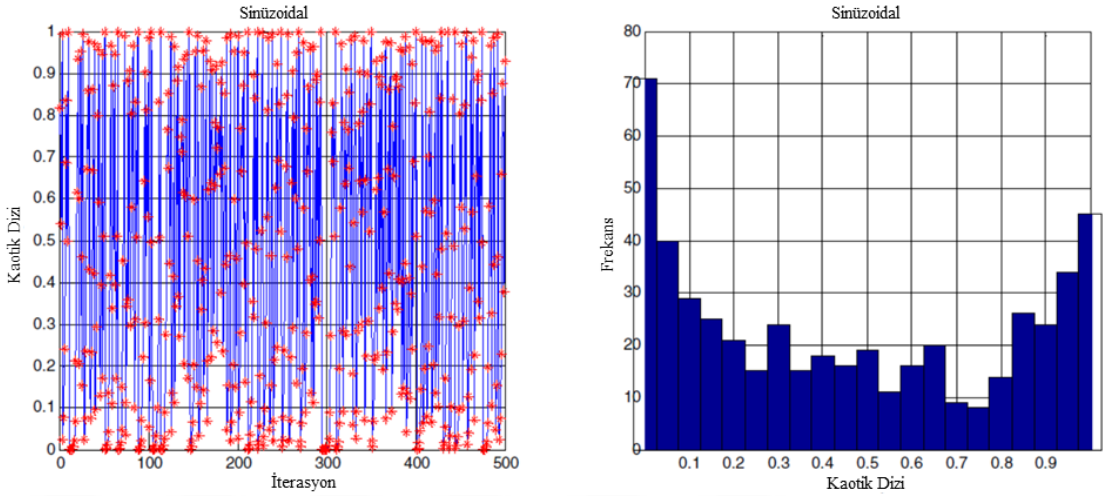
$$x_{n+1} = \alpha x_n^2 Sin(\pi x_n)\ ,\ x_n \in (0,1) \tag{3.5}$$

$a = 2.3\ ve\ x_0 = 0.7$ için aşağıdaki basit formu alır.

$$x_{n+1} = Sin(\pi x_n)\ ,\ \ x_n \in (0,1) \tag{3.6}$$

Bu fonksiyon (0,1) aralığında kaotik bir seri üretir. $x_n = 0.5$ için serinin ilk terimi hariç diğer tüm elemanlarının 0 olduğu görülür. Bu durum da yine eşitlik (3.4) kullanılarak giderilebilir (Feng vd. 2017).

Lojistik harita fonksiyonunun 500 iterasyonu sonrasında elde edilen kaotik seriye ait serpilme ve çubuk dağılım grafikleri Şekil 3.11'de gösterilmiştir.



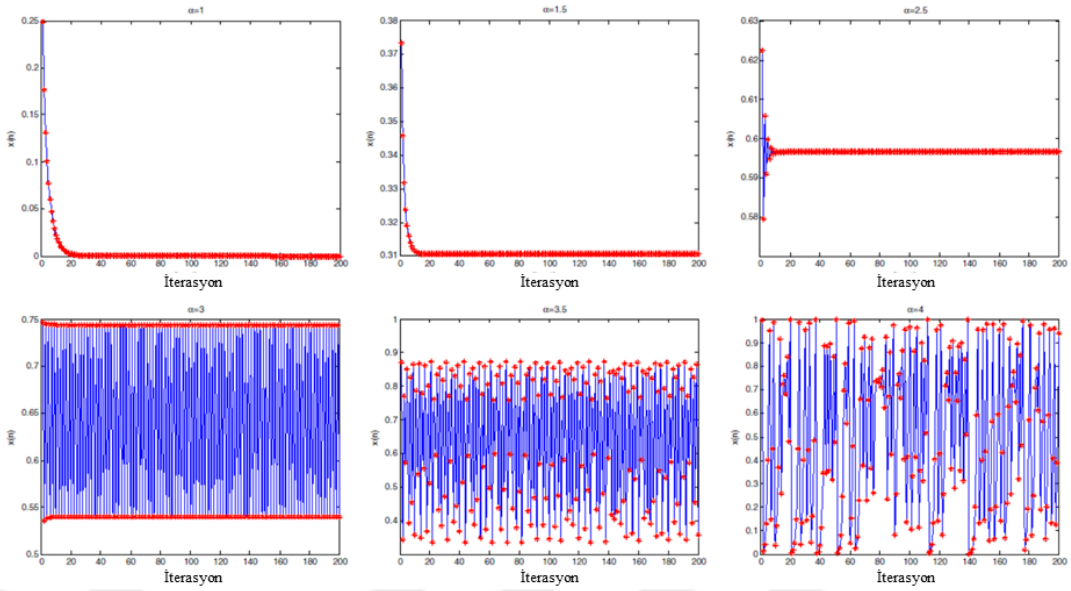
Şekil 3.11 Sinüzoidal harita fonksiyonu ile üretilen seriye ait serpilme ve çubuk dağılım grafikleri (Feng vd. 2017)

3.4.3. Sinüs Haritası

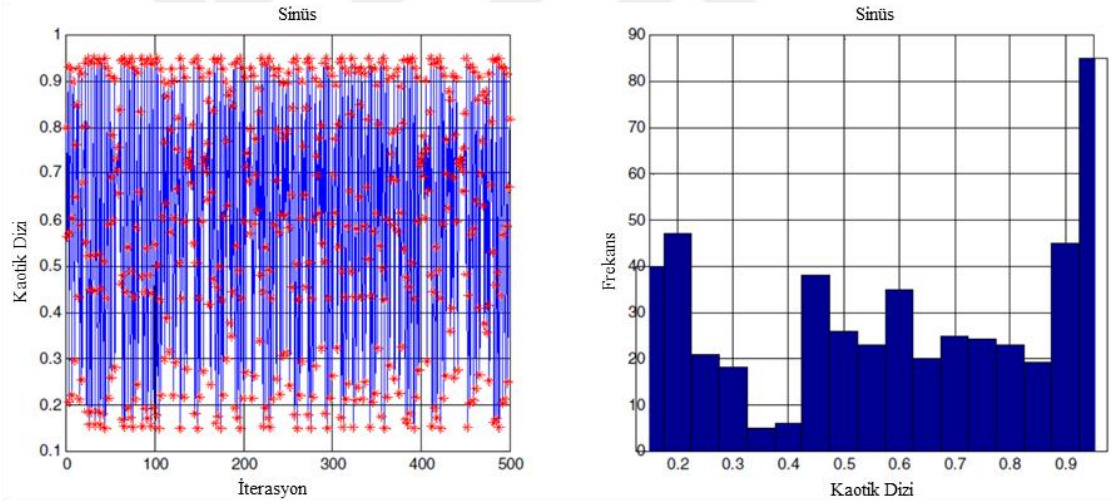
Sinüs harita fonksiyonu tek modlu bir fonksiyondur ve aşağıdaki şekilde ifade edilir (Tavazoei ve Haeri 2007).

$$x_{n+1} = \frac{\alpha}{4} Sin(\pi x_n) \; , \; 0 < \alpha \leq 4 \qquad (3.7)$$

Sinüs haritası, sinüzoidal ile oldukça benzerdir. $\alpha = 4$ için sinüzoidal harita ile aynı formda olacaktır. Bununla birlikte sinüs haritası ile elde edilen kaotik seri 0 ile 1 aralığında değil de 0 ile $\alpha/4$ aralığında olacaktır. Sinüs harita fonksiyonu kullanılarak farklı α değerleri ile {1, 1.5, 2, 2.5, 3, 3.5, 4} elde edilen kaotik seriler Şekil 3.12'te gösterilmiştir. Şekil 3.12 'de $\alpha = 1, 1.5, 2, 2.5 \; ve \; 3$ için elde edilen serilerin bir veya iki sabit değerlerden oluştuğu görülür ve bu durum kaosun ergodiklik özelliği ile uyuşmaz. $\alpha = 3.5 \; ve \; 4$ ile elde edilen serilerin ise oldukça başarılı kaotik özellik gösterdiği görülür. Sinüs haritasının $\alpha = 3.8$ için 500 iterasyonu sonrasında elde edilen kaotik seriye ait serpilme ve çubuk dağılım grafikleri Şekil 3.13'da gösterilmiştir.



Şekil 3.12 Sinüs harita fonksiyonu kullanılarak farklı α değerleri ile elde edilen kaotik seriler



Şekil 3.13 Sinüzoidal harita fonksiyonu ile üretilen seriye ait serpilme ve çubuk dağılım grafikleri (Feng vd. 2017)

3.4.4. Çadır Haritası

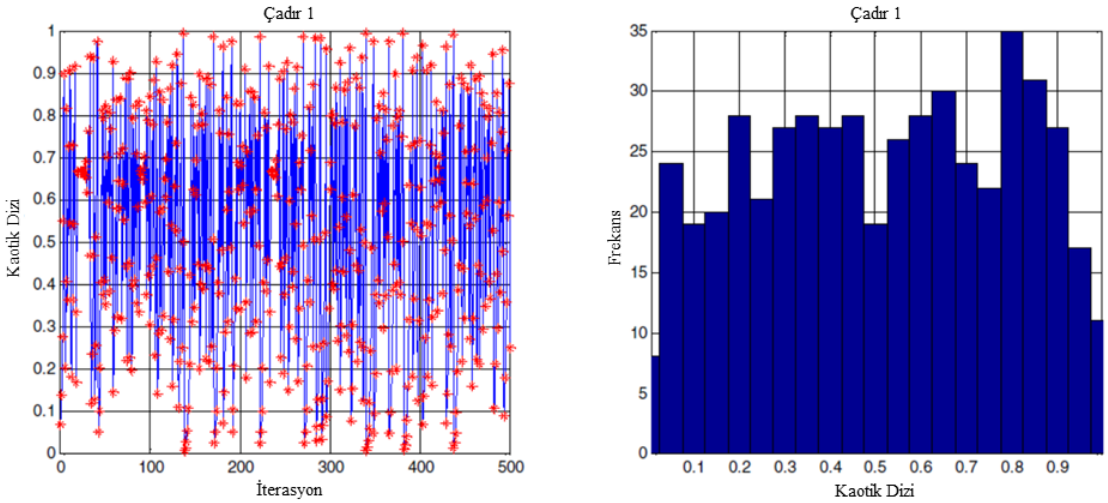
Lojistik haritaya benzeyen bir kaotik harita fonksiyonudur. (0,1) arasında kaotik seri üreten bu harita aşağıdaki gibi ifade edilir.

$$x_{n+1} = \begin{cases} \mu x_n & 0 \leq x_n \leq 1/2 \\ \mu\,(1 - x_n) & 1/2 \leq x_n \leq 1 \end{cases}, \mu = 2 \quad (3.8)$$

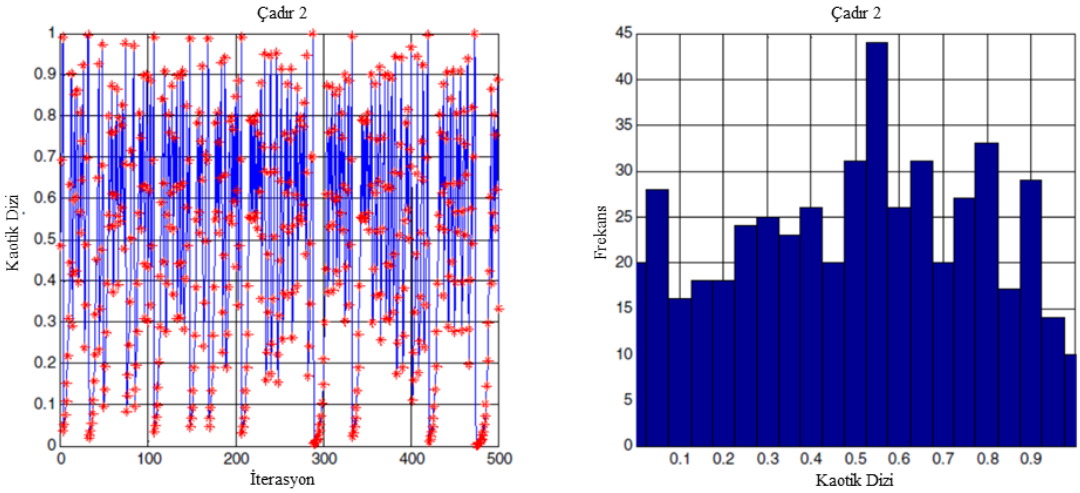
Çadır haritasının bir diğer gösterimi ise aşağıdaki gibidir (Peitgen, Jürgens and Saupe 2004)

$$x_{n+1} = \begin{cases} x_n/0.7 & x_n < 0.7 \\ 10/3x_n(1 - x_n) & değilse \end{cases} \quad (3.9)$$

Eşitlik 3.8 ve 3.9 ile 500 iterasyon ile elde edilen kaotik serilere ait serpilme diyagramı ve çubuk dağılımları Şekil 3.14 ve Şekil 3.15'da gösterilmiştir. Her iki eşitlikten de elde edilen seriler benzer kaotik davranış göstermektedirler (Feng vd. 2017).



Şekil 3.14 Çadır harita fonksiyonu (eşitlik 3.8) ile üretilen seriye ait serpilme ve çubuk dağılım grafikleri (Feng vd. 2017)



Şekil 3.15 Çadır harita fonksiyonu (eşitlik 3.9) ile üretilen seriye ait serpilme ve çubuk dağılım grafikleri (Feng vd. 2017)

3.4.5. Gaus Haritası

Gauss haritası literatürde test amaçlı kullanılmıştır ve aşağıdaki şekilde tanımlanır. Bu harita fonksiyonu da (0,1) arasında kaotik seri üretir (Peitgen, Jürgens, ve Saupe 2004).

$$x_{n+1} = \begin{cases} 0 & x_n = 0 \\ \mu/x_n mod(1) & değilse \end{cases} \quad (3.10)$$

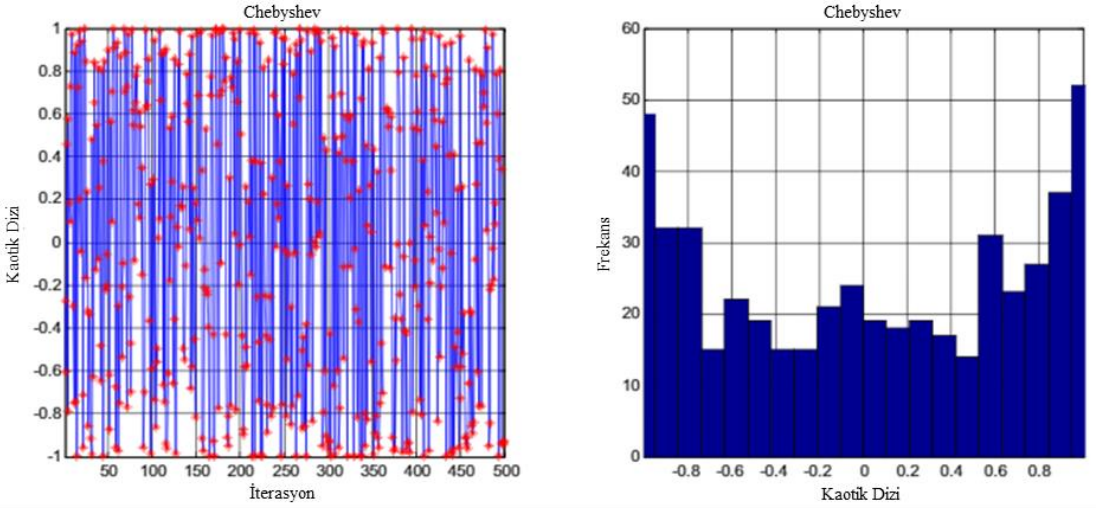
3.4.6. Chebyshev Haritası

Chebyshev kaotik haritası yaygın bir simetrik alan haritasıdır. Genel olarak, sinir ağları, dijital haberleşme ve güvenlik problemlerinde kullanılır. Chebyshev haritası (-1,1) arasında kaotik seri üretir ve aşağıdaki şekilde ifade edilir (Feng vd. 2017), (Xiaofang Yuan vd. 2014), (Tavazoei ve Haeri 2007).

$$x_{n+1} = cos(\mu\, cos^{-1} x_n) \quad , \qquad 0 < \mu \quad (3.11)$$

$\mu = 8$ ve $x_n = 0.5$ olması durumunda yukarıdaki eşitlik sonraki tüm değerleri -0.5 olan bir seri üretir. Bu durumu gidermek için eşitlik (3.3.4) kullanılabilir.

Eşitlik (3.11) ile 500 iterasyon sonrasında elde edilen kaotik seriye ait serpilme diyagramı ve çubuk dağılımları Şekil 3.16' da gösterilmiştir.



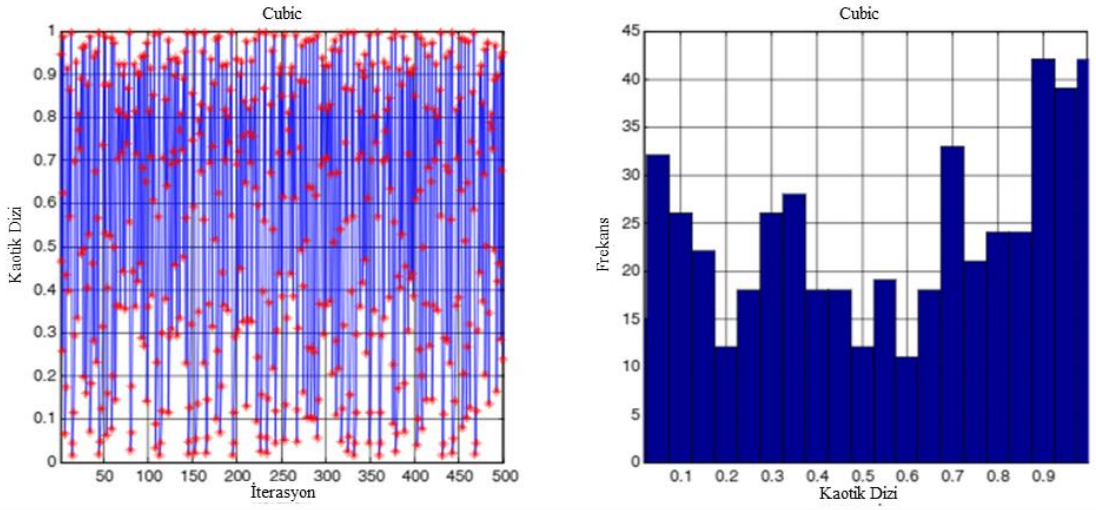
Şekil 3.16 Chebyshev harita fonksiyonu ile üretilen seriye ait serpilme ve çubuk dağılım grafikleri (Feng vd. 2017)

3.4.7. Cubic Haritası

Cubic haritası, şifreleme gibi çeşitli uygulamalarda yaygın olarak kullanılan ve aşağıdaki şekilde ifade edilen bir kaotik seri üretim fonksiyonudur (Baykasoglu 2012), (Feng vd. 2017), (Xiaofang Yuan vd. 2014).

$$x_{n+1} = \rho x_n (1 - x_n^2) \quad , \qquad x_n \in (0,1) \tag{3.12}$$

Cubic harita fonksiyonu $\rho = 2.59$ için (0,1) arasında kaotik seri üretir. Bu fonksiyon ile 500 iterasyon sonrasında elde edilen kaotik seriye ait serpilme diyagramı ve çubuk dağılımları Şekil 3.17' da gösterilmiştir.



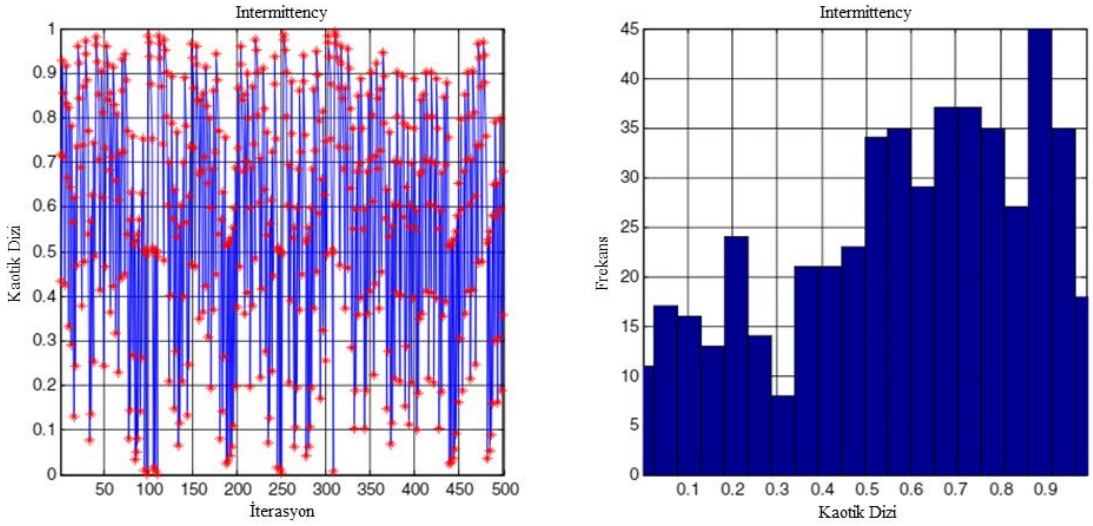
Şekil 3.17 Cubic harita fonksiyonu ile üretilen seriye ait serpilme ve çubuk dağılım grafikleri (Feng vd. 2017)

3.4.8. Intermittency Haritası

Intermittency haritası, parçalı doğrusal dilimlerin doğrusal olmayan bir dilim ile değiştirildiği Bernoulli Shift haritasının bir uzantısıdır ve aşağıdaki şekilde tanımlanır (Fister vd. 2015), (A. H. Gandomi vd. 2013), (Tavazoei ve Haeri 2007), (Feng vd. 2017).

$$x_{n+1} = \begin{cases} \varepsilon + x_n + c x_n^m & 0 < x_n \leq d \\ \dfrac{x_n - d}{1 - d} & d < x_n \leq 1 \end{cases} \tag{3.13}$$

Burada $d \in (0,1)$, $c \ll d$, $c = \frac{1-\epsilon-d}{1-d^m}$ ve m genellikle 2 ye eşitlenen bir sabittir. $m = 2$, $d = 0.5$ ve $\epsilon = 0.49$ ile oluşturulacak kaotik seri (0,1) arasındadır. Bu fonksiyon ile 500 iterasyon sonrasında elde edilen kaotik seriye ait serpilme diyagramı ve çubuk dağılımları Şekil 3.18 ‘de gösterilmiştir.



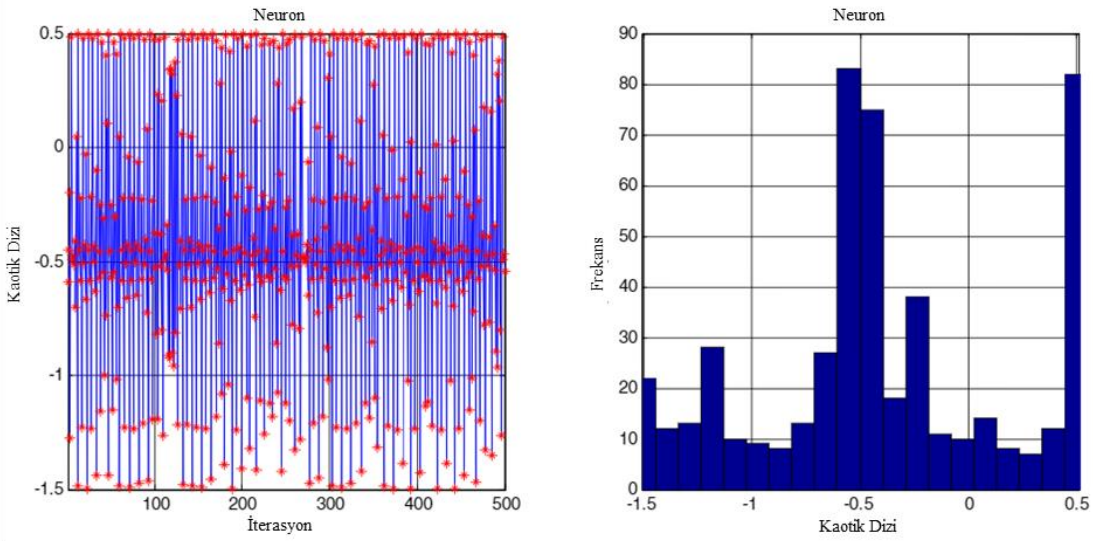
Şekil 3.18 Intermittency harita fonksiyonu ile üretilen seriye ait serpilme ve çubuk dağılım grafikleri (Feng vd. 2017)

3.4.9. Neuron Haritası

Neuron haritası doğrusal olmayan geri bildirimli bir kaotik haritadır. Kaotik bir yapı oluşturabilmek için hiperbolik tanjant fonksiyonu ile üssel fonksiyon bu harita aşağıdaki şekilde tanımlanır (A. H. Gandomi vd. 2013), (Mingjun ve Huanwen 2004), (J. Zhang, Yang, ve Zhang 2009).

$$x_{n+1} = \eta - 2tanh(\gamma)exp(-3x_n^2) \tag{3.14}$$

Burada η azaltma faktörü $(0 \leq \eta \leq 1)$ ve γ oransallık faktörü olarak adlandırılır. $\eta = 0.5$ ve $\gamma = 5$ ile elde oluşturulan kaotik seri (-1.5,0.5) arasındadır. Bu fonksiyon ile 500 iterasyon sonrasında elde edilen kaotik seriye ait serpilme diyagramı ve çubuk dağılımları Şekil 3.19'de gösterilmiştir.



Şekil 3.19 Neuron harita fonksiyonu ile üretilen seriye ait serpilme ve çubuk dağılım grafikleri (Feng vd. 2017)

3.4.10. Liebovitch Haritası

Liebovitch haritası, Liebovitch ve Toth tarafından önerilen bir diğer parçalı doğrusal harita örneğidir. Bu harita, aralıksız üst üste binen alt bölgelerdeki üç parçalı doğrusal bölümden oluşmaktadır (Feldman 2012) (Fister, et al. 2015) (Gandomi, et al. 2013) (Feng, et al. 2017) (Tavazoei and Haeri 2007).

$$x_{n+1} = \begin{cases} \alpha x_n & 0 < x_n \leq d_1 \\ \dfrac{d_2 - x_n}{d_2 - d_1} & d_1 < x_n \leq d_2 \\ 1 - \beta(1 - x_n) & d_2 < x_n \leq 1 \end{cases} \quad (15)$$

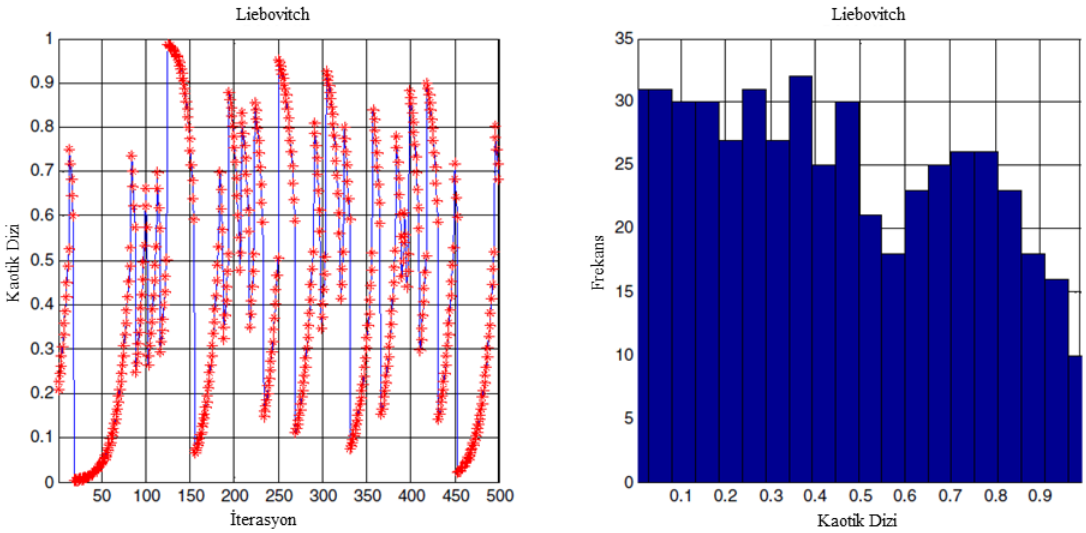
Burada $d_1, d_2 \in (0,1)$, $d_1 < d_2$, $\alpha < \beta$ ve

$$\alpha = \frac{d_2}{d_1}\left(1 - (d_2 - d_1)\right)$$

$$\beta = \frac{1}{d_2 - 1}\left((d_2 - 1) - d_1(d_2 - d_1)\right)$$

Burada $d_1 \; ve \; d_2$, alt aralığın 2 uç noktası iken $\alpha \; ve \; \beta$ sırasıyla aktif ve pasif durumlara karşılık gelen doğrusal haritanın eğimi olarak tanımlanır. $d_1 \; ve \; d_2$ parametrelerinin,

eşitlik 3.15 ile üretilen kaotik seri üzerinde belirgin bir etkisi vardır. Uygun belirlenmeyen $d_1 \; ve \; d_2$ değerleri ile elde edilen seriler kötü veya kaotik olmayan diziler olacaktır. $d_1 = 0.5$, $d_2 = 0.6$, $\alpha = 1.08$ ve $\beta = 1.125$ değerleri ile elde edilen kaotik dizi (0,1) aralığındadır. Bu kaotik seriye ait serpilme diyagramı ve çubuk dağılımları Şekil 3.20' de gösterilmiştir.



Şekil 3.20 Liebovitch harita fonksiyonu ile üretilen seriye ait serpilme ve çubuk dağılım grafikleri (Feng vd. 2017)

3.4.11. ICMIC Haritası

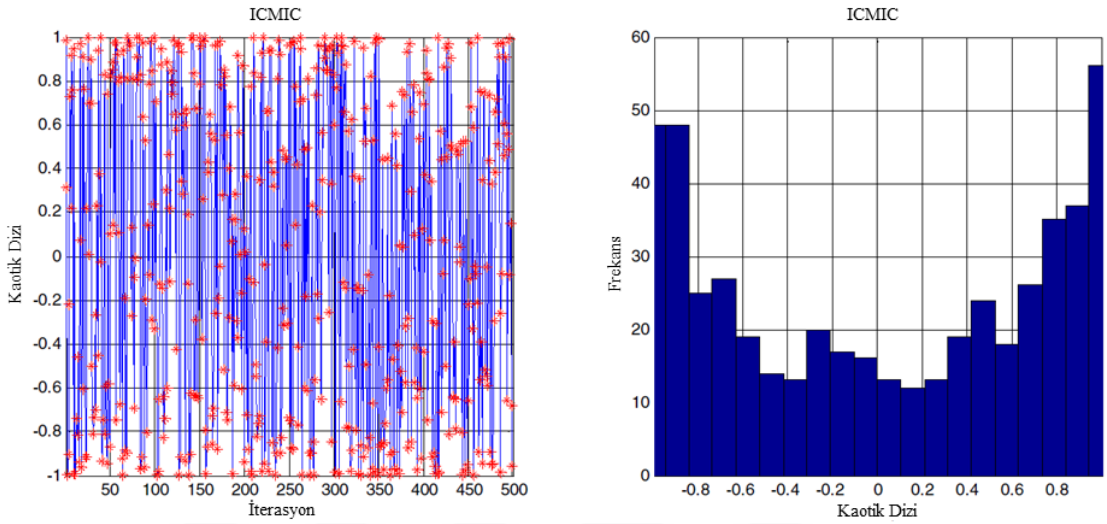
Sonsuz çöküşlerle yinelenen kaotik harita ICMIC harita fonksiyonu, (1,1) aralığında kaotik diziler oluşturur ve aşağıdaki şekilde tanımlanır (Feng vd. 2017), (Tavazoei ve Haeri 2007), (Xiaofang Yuan vd. 2014).

$$x_{n+1} = sin\left(\frac{\alpha}{x_n}\right) \; , \alpha \; \in (0, \infty), \;\; x_n \in (-1,1) \tag{3.16}$$

Bazı kaynaklarda şu şekilde de tanımlandığı görülür (May 1976), (Fister vd. 2015), (A. H. Gandomi vd. 2013).

$$x_{n+1} = sin\left(\frac{\alpha\pi}{x_n}\right) \; , \alpha \; \in (0,1), \;\; x_n \in (-1,1) \tag{3.17}$$

Benzetim çalışmaları, eşitlik 3.3.16 ile elde edilen kaotik dizilerin α sabitinden çok az etkilendiklerini, eşitlik 3.3.17 'nin de $\alpha > 0.6$ olması durumunda iyi bir kaotik dizi üretebildiğini göstermiştir (Feng vd. 2017). Bu fonksiyon ile 500 iterasyon sonrasında elde edilen kaotik seriye ait serpilme diyagramı ve çubuk dağılımları Şekil 3.21 'te gösterilmiştir.



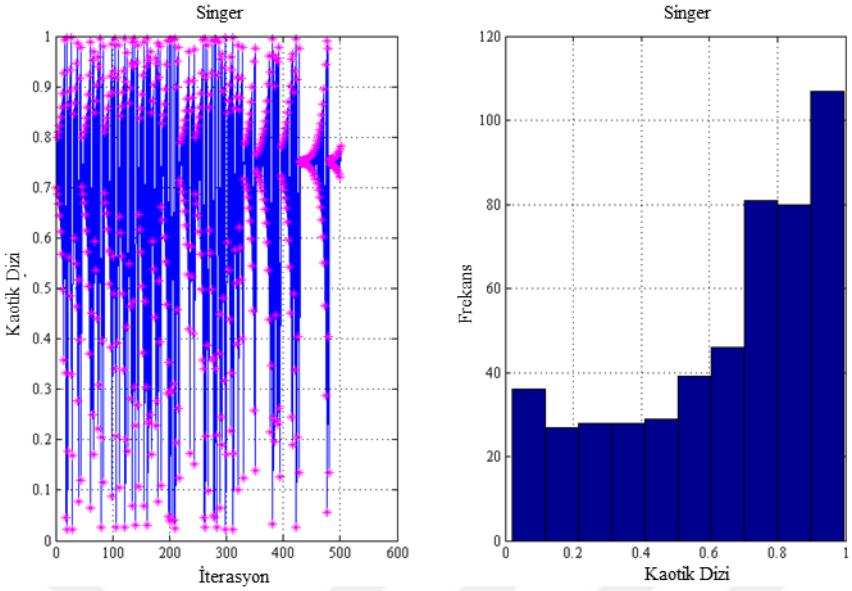
Şekil 3.21 ICMIC harita fonksiyonu ile üretilen seriye ait serpilme ve çubuk dağılım grafikleri (Feng vd. 2017)

3.4.12. Singer Haritası

Bir boyutlu kaotik Singer haritası şu şekilde tanımlanır.

$$x_{n+1} = \mu(7.86x_n - 23.31x_n^2 + 28.75x_n^3 - 13.302875x_n^4) \quad , \qquad (3.18)$$
$$x_n \in (0,1)$$
$$\mu \in (0.9, 1.08)$$

Bu fonksiyon ile 500 iterasyon sonrasında elde edilen kaotik seriye ait serpilme diyagramı ve çubuk dağılımları Şekil 3.22 'de gösterilmiştir.



Şekil 3.22 Singer harita fonksiyonu ile üretilen seriye ait serpilme ve çubuk dağılım grafikleri

3.5. Kaotik Sistemlerin İstatistiksel Değerlendirmeleri

Aşağıdaki 4 koşulu sağlayan dinamik sistemlerin kaotik bir davranışa sahip oldukları söylenebilir (Feldman 2012).

- Dinamik kural tanımlıdır.
- Yörüngeler periyodik değildir.
- Yörüngeler sınırlıdır.
- Dinamik sistem başlangıç koşullarına karşı duyarlıdır.

Dinamik kural ile anlatılmak istenen bu sisteme ait fonksiyondur. Bir önceki bölümde bu amaçla kullanılan farklı fonksiyonlar tanımlanmıştır.

Yörüngelerin periyodik olmaması da fonksiyonun sürekli aynı değerleri üretmemesi gerekliliğini tarif etmektedir. Yani kullanılan dinamik sistem, ergodiklik olarak adlandırılan sürekli farklı sayılar üretebilen bir özelliğe sahip olmalıdır.

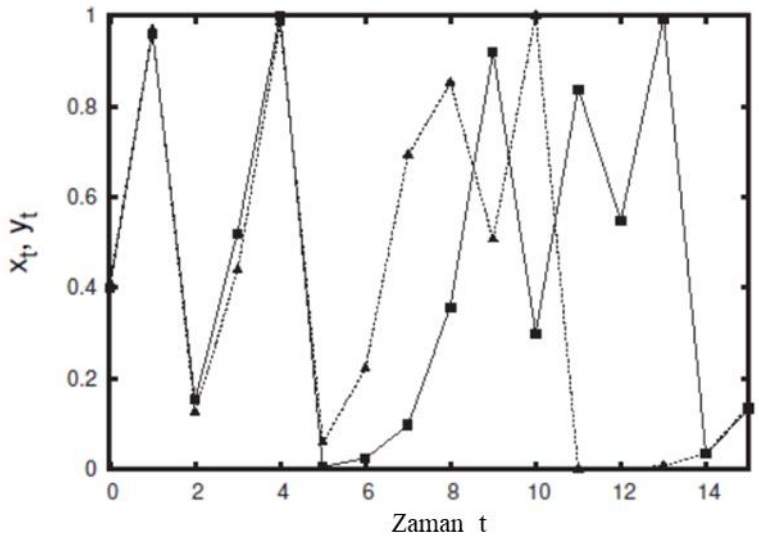
Yörüngelerin sınırlı olması kullanılan fonksiyonun sürekli belli bir aralıkta sayı üretebilmesini belirtir. Yani üretilen sayı aralığı sürekli artmamalıdır.

Sonraki kısımda başlangıç koşullarına hassas bağımlılık ve kaotik sistemlerin istatistiksel kararlılıkları detaylı olarak incelenecektir.

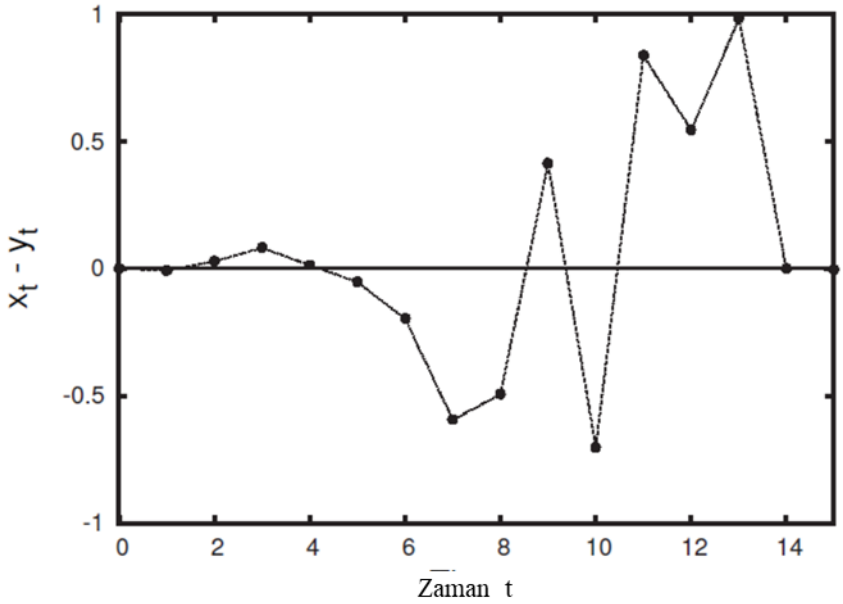
3.5.1. Kelebek Etkisi - Başlangıç Koşullarına Hassas Bağımlılık

Başlangıç koşullarına hassas bağımlılık (Sensitive Dependence on Initial Condition, SDIC) ise başlangıç koşullarındaki ufak değişikliklerin dinamik dizinin yörüngesinde büyük etkiye yol açması olarak tanımlanır. Bu özellik yaygın haliyle kelebek etkisi olarak da bilinir (Feldman 2012), (B. Liu vd. 2005), (Yang, Liu, ve Zhou 2014).

Lojistik harita fonksiyonu ile birbirine çok yakın iki başlangıç değeri için oluşturulan 2 farklı kaotik serinin serpinti grafiği Şekil 3.23 ’te gösterilmiştir. Burada x serisinin başlangıç değeri x_0=0.4 iken y serisi için başlangıç değeri y_0=0.41 alınmıştır. Her iki seride de Lojistik harita fonksiyonu sabiti μ=4 alınmıştır. Şekil 3.23 incelendiğinde, 4 ve 5. adımlara kadar serilerin benzer değerleri ürettiği görülürken 6. ve 7. adımlardan sonra oldukça farklı değerlere sahip oldukları görülür. Şekil 3.24 ‘da ise bu 2 seri arasındaki fark gösterilmiştir. Burada olduğu gibi, bir kaotik fonksiyonun, birbirine çok yakın başlangıç değerleriyle, az sayıdaki adımdan sonra birbirinden tamamen farklı değerler üretebilmesi başlangıç koşullarına hassas bağımlılık olarak adlandırılır.



Şekil 3.23 x_0=0.4 ve y_0=0.41 başlangıç değerleri ile oluşturulan x ve y kaotik serileri (Feldman 2012)



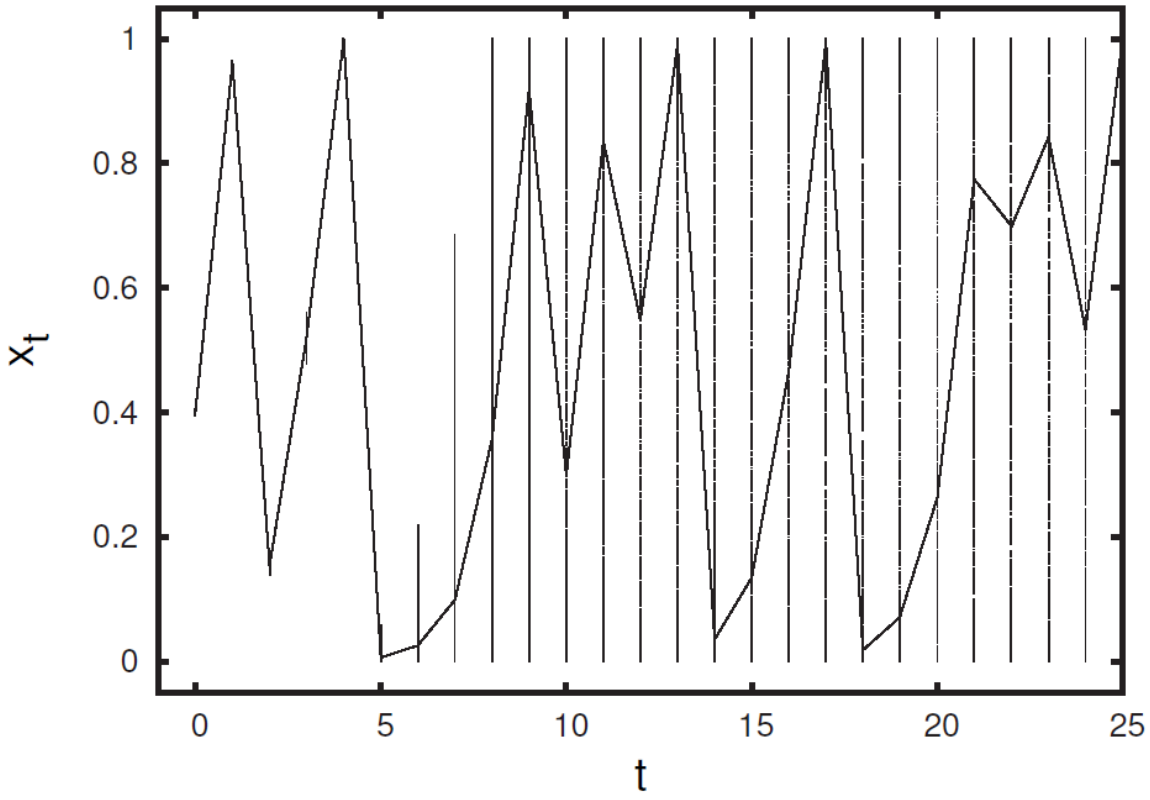
Şekil 3.24 x ve y kaotik seriler arasındaki fark (Feldman 2012)

Başlangıç koşullarına hassas bağımlılığın matematiksel olarak tanımlanabilmesi için bazı matematiksel ifadeler gereklidir (Feldman 2012). Belirlenen kaotik fonksiyon f ile x_0 ve y_0 başlangıç değerleri kullanılarak x ve y olmak üzere 2 farklı kaotik seri üretilir. Burada kullanılan kaotik fonksiyonun başlangıç koşullarına hassas bağımlığının kabulü için, her bir adımdaki x ve y değerlerinin farkı ε değerinin, n adım sonrasında istenilen bir δ

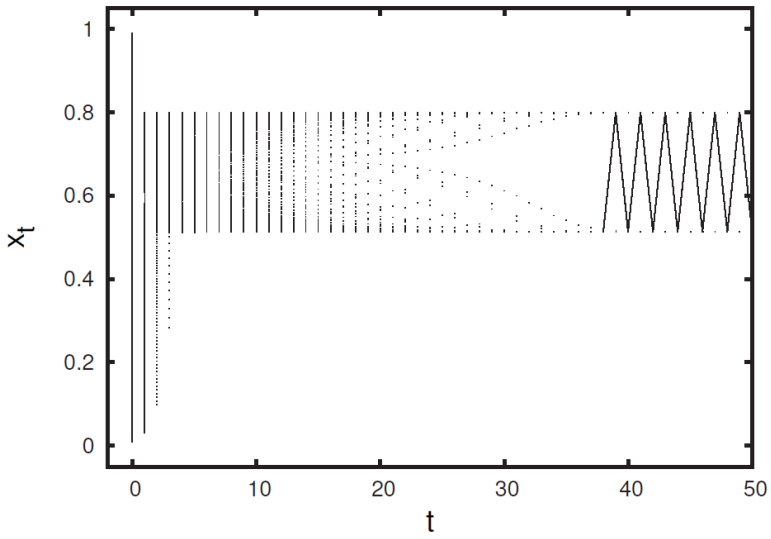
değerinden büyük olması gerekir. Yani başlangıç koşullarına hassas bağımlılık aşağıdaki şekilde ifade edilir.

$$|x_n - y_n| > \delta \tag{3.19}$$

Başlangıç koşullarına olan duyarlılığı göstermek için yine lojistik harita fonksiyonu ile 0.395 ve 0.405 arasında değişen başlangıç değerleri ile 1000 farklı kaotik seri üretilmiştir. , μ=4.0 için x_0=4 serisi ve diğer serilerin bu seriden olan uzaklıklarını gösteren serpinti grafiği Şekil 3.25 'de, μ=3.2 için elde edilen serpinti grafiği ise Şekil 3.26 'de gösterilmiştir. Şekil 3.25 'de 8. iterasyon sonrasında elde edilen değerlerin birbirinden tamamen farklılaşabildiği açıkça görülürken, Şekil 3.26 'deki değerlerin 0.5 ile 0.8 arasında sıkıştığı görülmektedir.



Şekil 3.25 μ=4.0 ve 0.395 < x_0 < 0.405 için elde edilen 1000 farklı kaotik seri (Feldman 2012)



Şekil 3.26 μ=3.2 ve 0.395 < x_0 < 0.405 için elde edilen 1000 farklı kaotik seri (Feldman 2012)

3.5.2. Lyapunov Üsteli

Başlangıç koşullarına duyarlı dinamik sistemlerin ne kadar duyarlı oldukları kaç iterasyon sonrasında birbirlerinden ayrıştıkları dinamik fonksiyonların karakteristik özelliklerinden biridir. Bu özellik farklı kaotik fonksiyonların karşılaştırılmalarında da kullanılır (Feldman 2012).

Birbirine yakın iki farklı x_0 ve y_0 başlangıç değerleri ile üretilen x ve y kaotik serileri incelendiğinde başlangıçtaki ayrışma Do şu şekilde tanımlanır.

$$D_0 = |x_0 - y_0| \tag{3.20}$$

t iterasyon sonra oluşan farklılık D_t ise

$$D_t = |x_t - y_t| \tag{3.21}$$

Şeklinde yazılır. Dinamik sistemin Şekil 3.25 'deki gibi başlangıç koşullarına duyarlı olması durumunda D_t değeri hızla büyüyecektir. Birçok dinamik sistem için fark

fonksiyonun başlangıç adımları için aşağıdaki gibi yazılabilen üstel bir davranış gösterdiği görülmüştür (Feldman 2012).

$$D_t \approx D_0 2^{\lambda t} \tag{3.22}$$

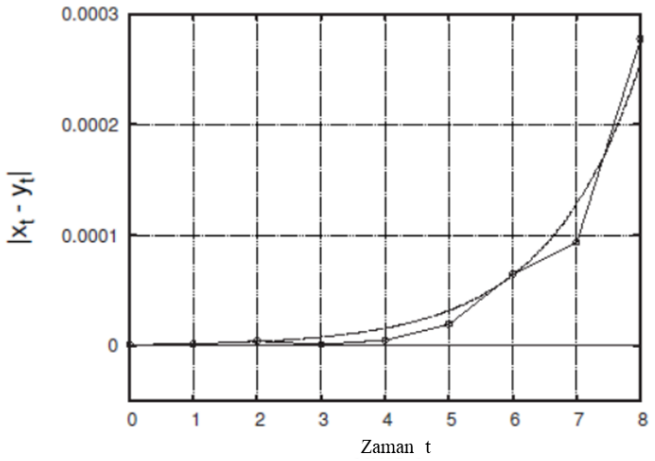
Bu fonksiyondaki λ 'ya, Lyapunov üsteli denir. λ Sıfırdan büyük olduğunda $2^{\lambda t}$ 'nin değeri de t 'nin artmasıyla giderek büyüyecektir. Tersi durumda yani, λ sıfırdan küçük olduğunda, $2^{\lambda t}$ 'nin değeri de t 'nin artmasıyla giderek küçülecektir. Bu durumda $\lambda > 0$ ise yörüngeler birbirlerinden ayrışır. Büyük Lyapunov üsteli λ, yörüngelerin birbirlerinden daha hızlı ayrıştığı ve dinamik sistemin başlangıç değerlerine daha hassas olduğu anlamına gelmektedir. Dinamik sistemler belli bir bant içinde oldukları için fark fonksiyonu sürekli artmaz. Bu nedenle eşiklik 3.22 sadece başlangıç değerleri için ve yaklaşık eşit olarak tanımlanmıştır (Feldman 2012).

Somut bir örnek olarak λ=1 olması durumunda eşiklik 3.22 şu şekle gelir.

$$D_t \approx D_0 2^t \tag{3.23}$$

Bu da, farkın her bir adımda 2 ye katlanacağı anlamına gelir. Sadece 8 adım sonra 2 yörünge arasındaki fark 2^8 = 256 kat artar. Bu durum Şekil 3.27 'da gösterilmiştir. Buradaki fark grafiği, Lojistik fonksiyon ile μ=4.0, x_0=0.3 ve y_0 = 0.300001 durumunda elde edilmiştir.

Lyapunov üsteli, dinamik sistemlerin başlangıç koşullarına olan duyarlılığının göstergesi olarak yaygın olarak kullanılır (Feldman 2012), (Feng vd. 2017), (Yang, Liu, ve Zhou 2014).



Şekil 3.27 2 zaman serisi arasındaki mutlak fark (Feldman 2012)

Farklı kaotik fonksiyonların Lyapunov üstelleri Çizelge 3.1 ’de gösterilmiştir (Yang, Liu ve Zhou 2014).

Çizelge 3.1 Lyapunov üstelleri

Kaotik Harita Fonksiyonu	Eşitlik	Kabuller	LÜ Değeri
Lojistik	$LE = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \ln \lvert f'(z_i) \rvert$	μ=4 için	0.6931
Kent	$LE = -\beta \log \beta - (1-\beta) \log(1-\beta)$	β = 0.4	0.6730
Bernoulli shift	$LE = -\beta \log \beta - (1-\beta) \log(1-\beta)$	λ=0.4	0.6730
Sinüs	$LE = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \ln \lvert f'(z_i) \rvert$	a=4	0.6882
ICMIC		a=4	1.6519
Çember		Ω= 0.5, K = 2	0.3556
Chebyshev		k=2	0.6932
Gaus		μ=4	2.3779

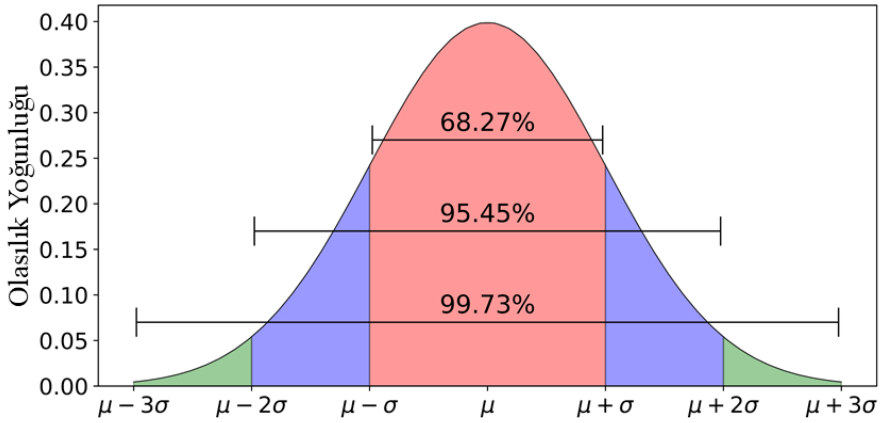
3.5.3. Olasılık Fonksiyonu

Olasılık, her basit ya da bileşik olaya sıfır ila bir arasında bir reel sayı karşılık getiren fonksiyondur. Bir rastgele değişkenin aritmetik ortalama, varyans, asimetri ölçüsü ve basıklık ölçüsü gibi bazı önemli karakteristiklerin belirlenmesinde olasılık fonksiyonu ve olasılık yoğunluk fonksiyonunun (Probability Distribution, Probability Density) bilinmesi gerekir. Olasılık yoğunluk fonksiyonları, rastgele değişkenin alacağı değerlerin olasılıklarını ifade ederler. Bu fonksiyonlar, bağlı bulundukları rastgele değişkenin türüne göre `kesikli' ve 'sürekli' nitelik kazanırlar. Bu fonksiyonlardan bazıları şunlardır (Özdemir 2000).

- Binom dağılım
- Poisson dağılım
- Hipergeometrik
- Normal
- Standart normal
- Weibull

Ancak birçok uygulamalı bilim dalında gözlemlenen sürekli rastgele değişkenlerin büyük bir çoğunluğunun frekans bölünmeleri yaklaşık olarak çan eğrisi görünümünde olup, “normal olasılık yoğunluk fonksiyonu” adı verilen bir fonksiyonla ifade edilmektedir. Normal olasılık fonksiyonu gerek teoride gerekse uygulamada en çok kullanılan bir matematiksel model durumundadır (Özdemir 2000).

Normal dağılım grafiği aşağıda Şekil 3.28 ‘de gösterilmiştir. Buna göre örneklenen kütledeki bağımsız değişkenlerin %68,27 ‘si ortala değer μ ’den 1 σ (standart sapma) kadar küçük ve büyük değerlerden oluşur. Ortalama değer μ ’den 2 σ (standart sapma) kadar küçük ve büyük değerlerin sayısı ise %95,45 ‘tir. Bu durum aynı zamanda 68-95-99.7 kuralı olarak da bilinir (Özdemir 2000).



Şekil 3.28 Normal olasılık dağılım fonksiyonu

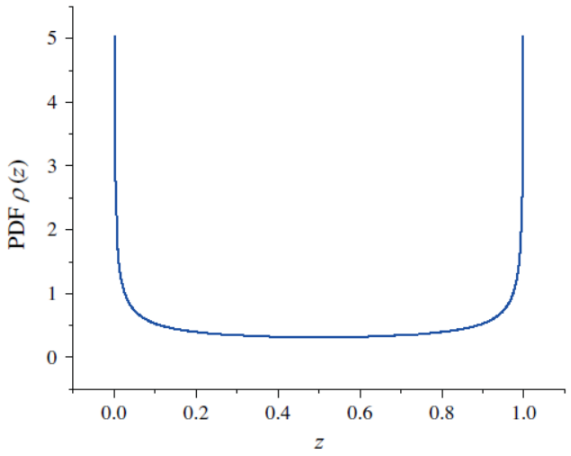
Normal dağılım fonksiyonu eşitlik (3.24) ile tanımlanmıştır.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{3.24}$$

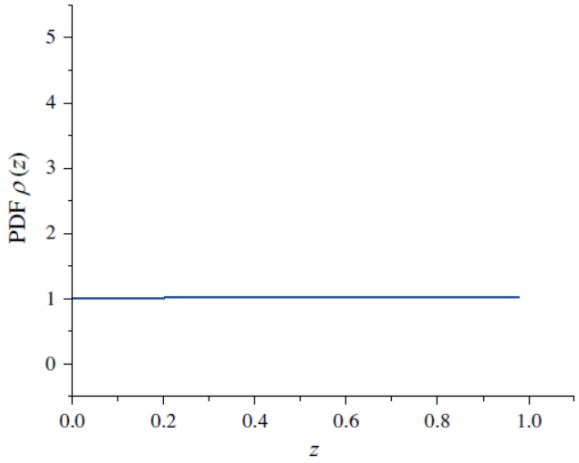
Önceki bölümde bahsedilen her bir kaotik fonksiyonun olasılık yoğunluk fonksiyonları Çizelge 3.2 ‘de belirtilmiş ve her bir fonksiyona ait dağılım grafikleri, Şekil 3.29 - Şekil 3.35 ‘da gösterilmiştir.

Çizelge 3.2 Kaotik Haritaları Olasılık Yoğunluk Fonksiyonları

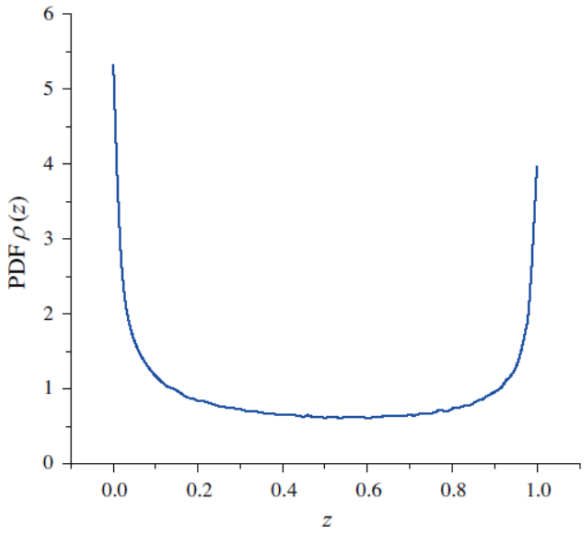
Kaotik Harita Fonksiyonu	Eşitlik	Kabul
Lojistik	$\rho(z) = \frac{1}{\pi\sqrt{z(1-z)}}$	μ=4
Kent ve Bernauli Shift	$\rho(z) = 1$	Her durumda
Sinüs Haritası	$\rho(z) = \frac{1}{\pi\sqrt{z(1-z)}}$	a=4
ICMIC		a=4
Çember		Ω= 0.5, K = 2
Chebyshev	$\rho(z) = \frac{1}{\pi\sqrt{1-z^2}}$	k=2
Gaus		μ=1



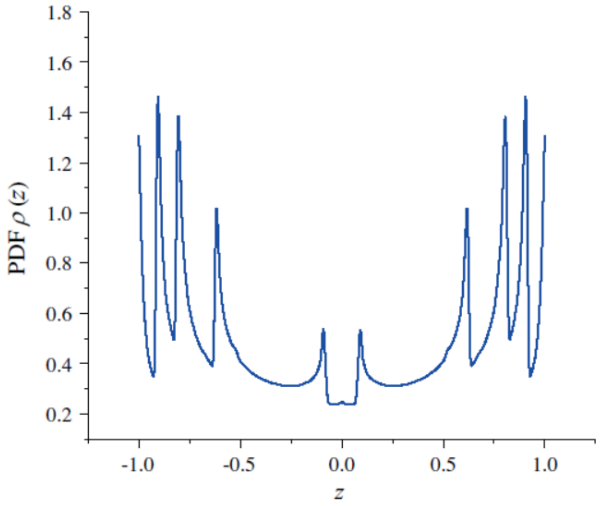
Şekil 3.29 Lojistik Harita kaotik serisinin olasılık dağılımı



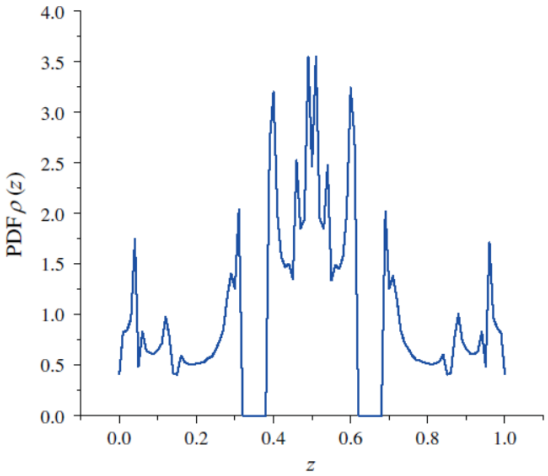
Şekil 3.30 Bernauli Shift ve Kent kaotik serilerinin olasılık dağılımı



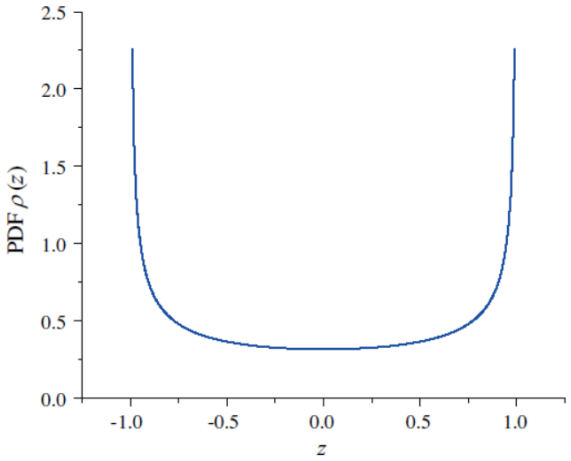
Şekil 3.31 Sinüs haritası kaotik serisinin olasılık dağılımı



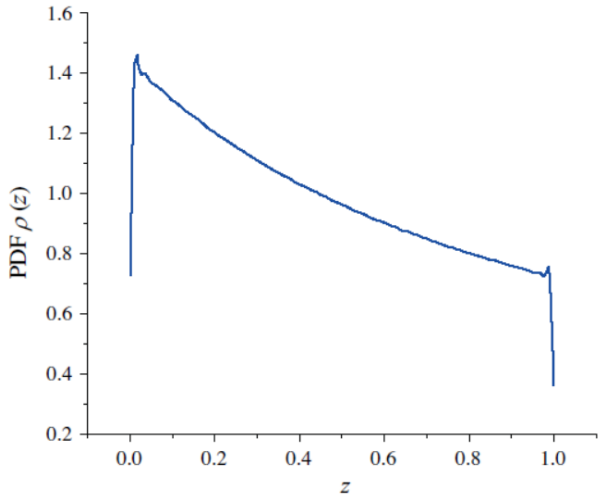
Şekil 3.32 ICMIC haritası kaotik serisinin olasılık dağılımı



Şekil 3.33 Çember haritası kaotik serisinin olasılık dağılımı



Şekil 3.34 Chebyshev haritası kaotik serisinin olasılık dağılımı



Şekil 3.35 Gauss haritası kaotik serisinin olasılık dağılımı

3.6. Biyo-coğrafya Temelli Kril Sürüsü Optimizasyonu

Optimizasyon problemlerinde kullanılan çeşitli yarı sezgisel algoritmalar önceki bölümlerde anlatılmıştır. Bu kısımda bu algoritmalardan biri olan Kril sürüsü optimizasyonu (KSO) algoritması anlatılacaktır.

İlk olarak Gandomi ve Alavi tarafından sunulan kril sürüsü algoritması, Antarktika'da yaşayan kril canlılarının beslenme davranışlarından esinlenilerek geliştirilmiş sürü temelli yarı sezgisel bir algoritmadır (Gandomi ve Alavi 2012). Gandomi ve Alavi, bu yöntemin performansını literatürde sıkça kullanılan test fonksiyonları üzerinde test etmişleri elde ettikleri sonuçları literatürde iyi bilinen 20 optimizasyon yöntemi ile karşılaştırmışlardır. Çaprazlama ve mutasyon operatörlerinin etkilerinin de incelendiği bu çalışmada sadece çaprazlamanın yer aldığı algoritmanın en başarılı sonuçlar verdiği görülmüştür. (Gandomi ve Alavi 2012).

Wang ve ark., global optimizasyon problemleri için benzetimli tavlama temelli kril sürüsü algoritması (KS) geliştirmişlerdir (G.-G. Wang vd. 2013). Test fonksiyonları üzerinden yaptıkları çalışmalarında, iyileştirilmiş yarı sezgisel KS yöntemi standart kril sürüsü yönteminden ve diğer optimizasyon yöntemlerinden daha başarılı olduğunu belirtmişlerdir.

Bu çalışmanın ardından KSO algoritması literatürde farklı uygulamalarda test edilmeye başlamıştır. Alikhani ve ark. Oransal-integral-türevsel (PID) kontrol devrelerinin ayarlanmasında bu yöntemi kullanmışlar, farklı fabrika tiplerindeki kontrolörlerde ve bilinmeyen tesislerdeki çevrimiçi ayar ve ayar denetleyicileri için de uygun olduğunu belirtmişlerdir (Alikhani vd. 2013).

Gölcük ve ark., bu yöntemi kombinatoryal optimizasyon problemlerinden biri olan atölye tipi çizelgeleme problemleri üzerinde test etmiş ve iyi sonuçlar aldıklarını belirtmişlerdir (Gölcük vd. 2014).

Fattahi ve ark., bu yöntemin kritik önem taşıyan keşif ve sömürme özelliklerindeki başarısını artırmak için yeni bir yöntem önermişlerdir. Bu yöntem her adımdaki problem çözme ilerlemesine göre keşif ve sömürü katılım miktarını dinamik olarak değiştirebilen bulanık bir yöntemdir. Farklı test fonksiyonları ile test edilen bulanık KSO algoritmasının yüksek performans gösterdiği belirtilmiştir (Fattahi, Bidar, ve Kanan 2016).

Wang ve ark., KSO algoritmasının global nümerik optimizasyon problemlerinin çözümündeki performansı arttırmak amacıyla yeni bir hibrit diferansiyel evrim operatörü eklemiştir. Tanıtılan HDE operatörü yoğunlaşmaya ilham verir ve krilin tanımlanan bölgede yerel arama yapmasına izin verir (L. Guo vd. 2014), (G. Wang vd. 2014).

Guo ve ark., hareket hesaplama işlemi sırasında üst kril arasında bilgi alışverişine imkân vererek daha iyi aday çözümler üretmeyi başarmıştır. Ayrıca, önerilen iyileştirilmiş KSO yöntemi KS hareket hesaplamasını güncellemek için yeni bir Lévy uçuş dağılımı ve elitizm şeması kullanır. Bu yeni yarı sezgisel yaklaşım, temel KSO algoritmasının sağlamlığını korurken global yakınsama hızını da hızlandırmaktadır (L. Guo vd. 2014)

Li ve ark., Kril sürüsünün özgürlüğünü ve belirsiz bireysel davranışını simüle etmek için, karşıt tabanlı öğrenme (Opposition Based Learning) KTÖ stratejisini ve serbest arama operatörünü KSO algoritmasına ekleyerek yeni bir karşıt tabanlı serbest aramalı KSO

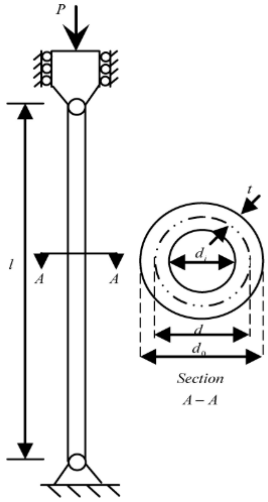
(SAKSO) algoritması geliştirmişlerdir (L. Li, Zhou, ve Xie 2014) . SAKSO 'de her bir kril birey, kendi algısına ve faaliyetlerinin kapsamına göre arama yapabilir. Serbest arama stratejisi, bireyleri yerel optimal çözümde sıkışıp kalmaktan kaçınmaya teşvik eder. Böylece kril popülasyonunun çeşitliliği ve keşif kabiliyeti artırılmıştır. On dört karşılaştırma fonksiyonundan elde edilen deney sonuçları, önerilen algoritmanın hem düşük boyutlu hem de yüksek boyutlu durumlarda etkili ve uygulanabilir olabileceğini göstermiştir. SAKSO 'nin yakınsama hızı ve hassasiyeti karşılaştırılan diğer algoritmalara (parçacık sürüsü optimizasyonu, diferansiyel evrim optimizasyonu, kril sürüsü, harmoni arama, serbest arama ve yarasa arama; PSO, DE, KH, HS, FS ve BA) kıyasla, daha iyi bir optimizasyon performansı ve sağlamlığı göstermiştir (L. Li, Zhou, ve Xie 2014).

Hafez ve ark., maymun algoritması ve KSO temelli hibrit bir özellik seçim sistemi (MAKSO) geliştirmişlerdir. Tavuk hareketinden ilham alan yeni bir evrimsel hesaplama tekniği olan MAKSO algoritması, en uygun çözümü hızlı bir şekilde bulmak için keşif ve sömürüyü uyarlamalı olarak dengeler. 18 veri setinde test edilen bu önerilen sistemin, yaygın olarak kullanılan parçacık sürüsü optimizasyonu (PSO) ve genetik algoritma (GA) yöntemlerine göre daha başarılı olduğu görülmüştür (Hafez vd. 2015), (Ahmed, Hafez, ve Hassanien 2015).

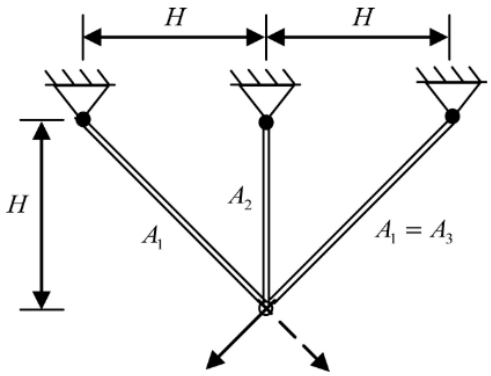
Gandomi ve ark. Kendi geliştirdikleri KSO algoritmasını mühendislikteki bilinen 6 optimizasyon problemine uygulamıştır. Bunlar ;

- Boru şeklindeki kolon tasarımı (Şekil 3.36)
- 3-Çubuk Kafes Tasarımı (Şekil 3.37)
- Hız düşürücü tasarımı (Şekil 3.38)
- Helisel Yay Tasarımı (Şekil 3.39)

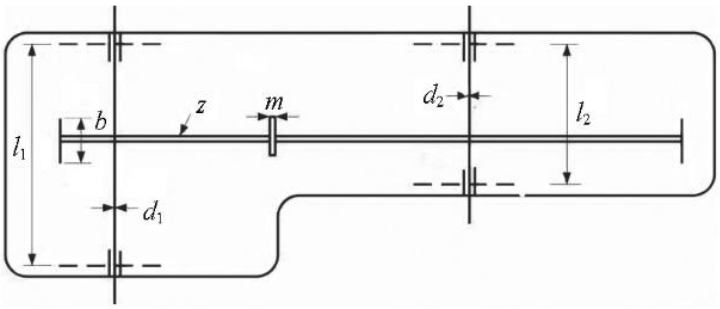
problemleridir. Karşılaştırmalar, KSO tarafından elde edilen sonuçların mevcut yöntemlerle elde edilen en iyi çözümlerden daha iyi olduğunu göstermiştir (Amir H. Gandomi ve Alavi 2015).



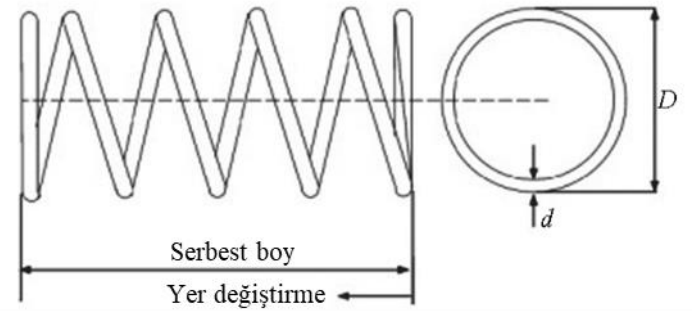
Şekil 3.36 Boru şeklindeki kolon tasarımı (Amir H. Gandomi ve Alavi 2015)



Şekil 3.37 3-Çubuk Kafes Tasarımı (Amir H. Gandomi ve Alavi 2015)



Şekil 3.38 Hız düşürücü tasarımı (Amir H. Gandomi ve Alavi 2015)



Şekil 3.39 Helisel Yay Tasarımı (Amir H. Gandomi ve Alavi 2015)

Vincylloyd ve Anand, kablosuz iletişim sistemlerinde, konum güncelleme (LU) ve sayfalama maliyetlerinin toplamından oluşan ağ maliyetlerinin optimizasyonunda KSO algoritmasını kullanmışlardır. Kullanıcıların çeşitli hareketlilik modellerine ve ağ mimarisine dayanarak, düşük oranlı (LR) alanların tasarımı kombinatoryal bir optimizasyon problemi olarak formüle edilmiştir (Vincylloyd ve Anand 2015).

Kumar ve ark., Birleşik Güç Akış Kontrolörü (UPFC) 'nün yer aldığı bir IEEE 14 veri yolu test sistemindeki gerçek güç kayıplarını ve voltaj saplamalarını en aza indirgemek için KSO algoritmasını kullanmışlardır. Elde ettikleri sonuçları genetik algoritma kullanarak elde edilenler ile karşılaştırarak KSO algoritmasının başarısını ortaya koymuşlardır (Kumar, Suryakalavathi, ve Kumar 2015).

Chaturverdi ve ark., GA, PS ve KSO optimizasyon yöntemlerinin performanslarını literatürdeki test fonksiyonları kullanarak test etmişlerdir. Çözüm süresi, elde edilen sonuçların ortalama ve standart sapma değerleri gibi istatistiksel veriler kullanarak algoritmaların verimliliklerini karşılaştırmışlardır (Chaturvedi, Pragya, ve Verma 2015).

Agrawal ve ark., orijinal KSO algoritmasına komşu uzaklık kavramını da dahil ederek geliştirdikleri algoritmalarını, test fonksiyonları ile test etmişlerdir (Agrawal, Pandit, ve Dubey 2016).

Arora ve ark., KSO algoritmasını, dalgacık mutasyonlu hibrit parçacık sürüsü (HPSOWM) ve örümcek maymunu algoritmalarıyla karşılaştırmışlardır. Test

fonksiyonları kullandıkları bu çalışmada, farklı yöntemlerden elde edilen sonuçların istatistiksel olarak farklılığının gösterilmesi için T testi yöntemini kullanmışlardır (V. Arora, Sood, ve Keshari 2016).

Jensi ve Jiji, zayıf sömürü kabiliyeti nedeniyle düşük olan performansını iyileştirmek için ekledikleri küresel arama operatörü ile, tanımlanmış arama bölgesine ilave olarak global yeni bölgelerin de keşfedilebilmesini sağlamışlardır. Böylece yeni kril bireyleri en keşfedilen en iyi bölgeye hareket ederler. Önerilen yöntemin literatürdeki diğer 13 optimizasyon algoritmasına göre üstünlüğünü, iyi bilinen 26 test fonksiyonu ile karşılaştırarak kanıtlamışlardır (Jensi ve Jiji 2016).

Jiang ve ark., sualtı kablosuz sensor ağlarındaki enerji tüketimini ve yığın (cluster) yönlendirme algoritmalarındaki ana yığınların yüklerinin optimizasyonunda KSO algoritmasını kullanmıştır. Çalışmalarında kullandıkları bu yöntem ile yığın enerji tüketiminin azaltıldığını, ağ enerji tüketiminin dengelenebildiğini ve ağ ömrünüm uzatılabileceğini göstermişlerdir(P. Jiang vd. 2016).

Sun ve ark., katılımcı ortamla doldurulan iki boyutlu bir ışınım muhafazasının ters geometri tasarımını çözmek için KSO algoritmasını kullanmıştır. Elde edilecek optimum tasarımda, tasarım yüzeyi üzerinde düzgün bir ışınım ısı akısı dağılımının elde edilmesi amaçlanmıştır. KSO algoritmasının, mikro genetik algoritma ve parçacık sürüsü optimizasyonuna göre daha iyi bir performans sergilediği belirtilmiştir (Sun vd. 2016).

Wang ve ark. standart KSO optimizasyon yönteminin global ve yerel arama yeteneklerinden tam olarak yararlanmak için, keşif ve sömürme adımlarını içeren, çok aşamalı kril sürüsü (MSKH) algoritmasını geliştirmişlerdir. Keşif aşaması standart KSO yöntemindeki gibiyken, sömürme fazında, optimizasyon probleminin çözümü sonrasında global optimuma ulaşmayı sağlamak ve bu yöntemin güvenilirliğini artırma için yerel mutasyon ve çaprazlama operatörleri üzerine yoğunlaşmışlardır. Bununla birlikte en iyi çözümü garanti edebilmek için bu yöntemin içerisine elitizm metodunu da eklemişlerdir (G.-G. Wang vd. 2016).

Wang ve ark., bu yöntemi elektronik burunlarla gaz tanınmasının geliştirilmesinde kullanmışlardır (L. Wang vd. 2016). Daha doğru bir kril davranışı elde edebilmek için güncellenen bir çaprazlama operatörü eklemişlerdir. Geliştirilmiş KSO algoritmasının, diğer optimizasyon algoritmaları ile karşılaştırılmasıyla elde edilen sonuçlar bu yöntemin oldukça başarılı olduğunu göstermiştir (L. Wang vd. 2016)

Rani ve Ramyachitra, kansere neden olan bir genin veya gen listesinin saptanmasında, kanser özelliklerinin hesaplama açısından oldukça zor olan gen anlatımlarının mikro dizi analizler yöntemi ile belirlenmesinde ve sınıflandırılmasında KSO algoritmasını kullanmıştır. Bu yöntem ve diğer iyi bilinen farklı optimizasyon yöntemleri 10 farklı mikro dizi kanser veri seti ile test edilmiştir. Sonuç olarak, önerilen yöntemin çoğu veri kümesi için % 100 sonuç doğruluğu ile mevcut diğer yöntemlerden daha iyi sonuç verdiği görülmüştür (Rani ve Ramyachitra 2017).

Strumberger ve ark., KSO algoritmasını sınırlandırılmış büyük ölçekli optimizasyon problemlerine uygulamıştır. Standart fonksiyonları ile bu yöntem ile temel ateş böceği yöntemini karşılaştırmışlardır. Ardından orijinal KSO algoritmasına ateşböceği yönteminin arama denklemini uygulayarak melez KSO algoritmasını geliştirmişlerdir. Bu yeni yöntemin, kıyaslama fonksiyonlarından elde edilen sonuçları da geliştirdiği görülmüştür (Strumberger vd. 2017).

Resma ve Nair, görüntü segmentasyonu probleminin çözümünde, yarı sezgisel KSO algoritması kullanan yeni birçok düzeyli eşik algoritması önermiştir. Optimum eşik değerleri, Kapur veya Otsu 'nun amaç fonksiyonlarının KSO algoritması ile maksimizasyonu ile elde edilmiştir. Yöntemin verimliliği, çeşitli kıyaslama görüntüleri kullanılarak, bakteri toplama, PSO ve GA gibi iyi bilinen diğer algoritmalardan elde edilen sonuçlar ile karşılaştırılarak ispat edilmiştir (Baby Resma ve Nair 2018).

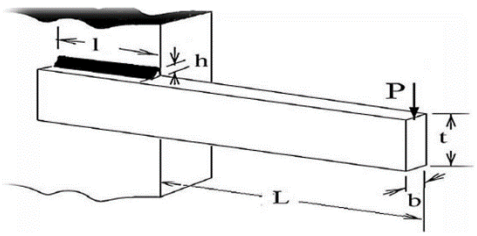
Fathy ve Abdelaziz, mikro şebekenin kurulu yenilenebilir enerji kaynakları (RES) ile çalıştırılmasının optimizasyonunda KSO ve karınca aslanı yöntemleri kullanmıştır. Bu yöntemi diğer algoritmalar ile karşılaştırdıkları çalışmalarında en iyi sonuçları KSO algoritması ile elde ettiklerini belirtmişlerdir (Fathy ve Abdelaziz 2018).

Kesavaraja ve Shenbagavalli, internet kullanıcılarının, bulut hizmeti deneyimini geliştirmek için sanal makine (VM) tahsis politikasında melez KSO algoritmasını kullanan yeni bir yöntem geliştirmiştir. Önerilen optimize edilmiş sanal makine tahsis politikası, daha iyi bir kalite deneyimi için kesintisiz erişim verimi sağladığından eksantrik efektini ve tıkanma etkisini ortadan kaldırır. KSO algoritmasının performansı, PSO, KKO, GA ve benzetimli tavlama yöntemleri ile test edilmiştir (Kesavaraja ve Shenbagavalli 2018).

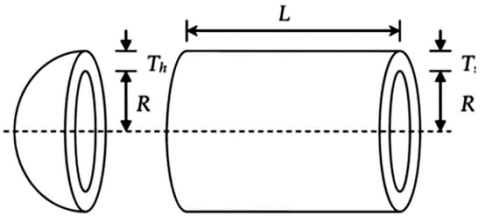
Abdel-Basset ve ark., yapısal optimizasyon problemlerinin çözümü için melez bir KSO algoritması sunmuştur. Geliştirilmiş KSO algoritması (Guguklu Arama Kril Sürüsü Optimizasyonu, GAKSO) guguklu arama (Cuckoo Search, CS) algoritmasındaki KU/KA operatörünün KSO içerisine entegre edilmesiyle elde edilmiştir. (Abdel-Basset vd. 2017). Burada çalışılan mühendislik problemleri şunlardır.

- Kaynaklı kiriş optimizasyon problemi (Şekil 3.40)
- Basınçlı kap problemi (Şekil 3.41)
- 3 Çubuk kafes tasarımı
- Dişli grubu tasarımı (Şekil 3.42)
- Hız düşürücü tasarımı

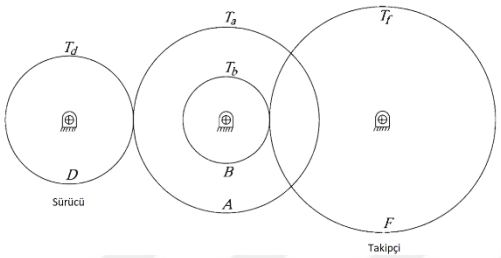
Önerilen bu yöntemin performansının mevcut KSO ve CS yöntemlerinden daha iyi oldukları görülmüştür (Abdel-Basset vd. 2017).



Şekil 3.40 Kaynaklı kiriş optimizasyon problemi (Abdel-Basset vd. 2017)



Şekil 3.41 Basınçlı kap problemi (Abdel-Basset vd. 2017)



Şekil 3.42 Dişli Grubu Tasarımı (Abdel-Basset vd. 2017)

Ashrafzadeh ve ark., günlük kap buharlaşmasının tahmini için, çok katmanlı algılayıcı (MLP) ve KSO algoritmalarına dayalı entegre bir veri zekâsı modeli sunmuştur (MLP-KH). İran'ın kuzey bölgesindeki iki meteoroloji istasyonundan toplanan günlük klimatoloji bilgileri, önerilen modelin klasik MLP ile karşılaştırmasında kullanılmıştır. Her bir yöntem ile elde edilen kare ortalamalarının karekköküne göre yapılan değerlendirmede önerilen yöntemin daha başarılı olduğu görülmüştür (Ashrafzadeh vd. 2019).

Bhowmik ve Malathi, Bilişsel radyo ağlarında etkili veri paylaşımı için KS balina optimizasyon algoritmasını önermiştir. Önerilen bu yöntem ile bilişsel radyo ağlarındaki yanlış alarm algılama oranının iyileştirilmesi hedeflenmiştir. KS balina optimizasyon yöntemi ile elde edilen sonuçların, genelleştirilmiş olasılık oranı, maksimum-minimum öz değer bulma ve maksimum öz değer bulma yöntemlerinden elde edilenlerden daha başarılı olduğu görülmüştür (Bhowmik ve Malathi 2019).

Zhihui ve ark., arz ve talep ile Kooperatif sinyal kontrol yönteminin (Cooperative Control with Supply and Demand, CCSD) gerçek zamanlı kapsamlı aramadaki zayıflıklarını iyileştirmek için KSO yöntemini önermiştir. KH-CCSD olarak adlandırılan bu yöntemin

karşılaştırılan diğer yöntemlere göre daha başarılı sonuçlar verdikleri görülmüştür (Zhihui vd. 2019).

Yukarıda anlatılan çalışmalara ek olarak Gharavian ve ark., Wang ve ark., Mahdi ve ark. ile Bentouati ve ark. KSO optimizasyon yöntemini kaotik fonksiyonlarla birleştirerek Kaotik KSO (KKSO) algoritmasını geliştirmişlerdir (Gharavian, Yaghoobi, ve Keshavarzian 2013), (B. Liu vd. 2005), (G.-G. Wang, Hossein Gandomi, ve Hossein Alavi 2013), (G. G. Wang vd. 2014), (Bentouati, Chettih, ve El-Sehiemy 2017).

Gharavian ve ark. global optimizasyon problemlerini iyileştirmek için kaos teorisi ile KSO algoritmasını birleştirmiştir. KSO içerisindeki rastgele olarak belirlenen fiziksel yayılma adımı bu çalışmada lojistik haritalama fonksiyonu ile değiştirilerek mevcut algoritmanın kaos tabanlı olması sağlanmıştır. Bu çalışmada kaotik haritalama fonksiyonlarından sadece lojistik harita kullanılmıştır. Literatürdeki çeşitli kıyaslama fonksiyonları ile test edilen KKSO yönteminin performansının, mevcut KSO 'ye göre daha iyi olduğu belirtilmiştir (Gharavian, Yaghoobi, ve Keshavarzian 2013).

Wang ve ark. sınırlı zaman gereksinimlerinde KSO algoritmasının performansını arttırmak için kaotik KSO algoritmasını sunmuştur. Bu yöntemi 32 tane kıyaslama fonksiyonunda ve dişli grubu tasarımı probleminde test etmiştir.

$$x_i^{t+1}(j) = (1-\beta)x_i^{ft}(j) + \beta g^* + \alpha r \tag{3.25}$$

Eşitlik 3.25 'te β, α ve g^* parametreleri sırasıyla, çekim parametresi, kontrol parametresi ve mevcut en iyi global pozisyon olarak adlandırılırlar. r ise 0 ile 1 arasında değişen rastlantısal bir sayıdır. Bu eşitlikteki ana parametre, hız vektörünü ölçekleyen çekim parametresi β ' dır ve global arama yeteneğini etkilemektedir (G.-G. Wang, Hossein Gandomi, ve Hossein Alavi 2013). Bu çalışmada, yem arama aktivitesi içindeki çekim parametresi β, sabit bir değer almak yerine kaotik haritalama fonksiyonları ile tanımlanarak mevcut KSO algoritmasının kaos yöntemi ile birleştirilmesi sağlanmıştır. Çekim parametresinin kaotik olarak belirlenmesi lokal minimuma takılma problemini giderirken, global optimuma hızlı bir şekilde ulaşma imkanı da sağlayacaktır Test

fonksiyonlarıyla yapılan karşılaştırmalardan elde edilen sonuçların, standart KSO yönteminden elde edilenlere göre daha iyi olduğu ya da en azından daha rekabetçi oldukları görülmüştür (G.-G. Wang, Hossein Gandomi, ve Hossein Alavi 2013), (G. G. Wang vd. 2014). Wang ve ark. genetik operatörlerden elitizm yöntemini de sonraki çalışmalarında kaos ile birlikte kullanmışlardır(G. G. Wang vd. 2014).

Bentonuati ve ark. KSO algoritmasındaki atalet ağırlığı parametresini sabit bir değer almak yerine kaotik haritalama fonksiyonlar ile çözüm süresince kaotik olarak değiştirmişlerdir. Buna ek olarak orijinal algoritmadan farklı olarak elitizm yöntemi de dahil edilmiştir. Önerdikleri bu algoritmayı ekonomik dağıtım probleminin çözümünde kullanmış, elde ettikleri sonuçları literatürdeki diğer algoritmalardan elde edilenler ile karşılaştırmışlardır. Yeni algoritmanın performansının oldukça başarılı olduğu belirtilmiştir (Bentouati, Chettih, ve El-Sehiemy 2017).

3.4.1. Kril Sürüsü Optimizasyonu

Kril sürüsü algoritması sürünün başka canlılar tarafından avlandığı ve sürüdeki ortalama kril yoğunluğunun azalarak yemek kaynağından uzaklaşılan durumu başlangıç aşaması olarak kabul eder. Doğal bir sistemde, her bireyin uygunluk değeri, kril sürüsünün en yüksek yoğunluk noktası ve yemeğin bulunduğu noktalara uzaklığın bir birleşimi şeklinde hesaplanır. Böylece uygunluk, amaç fonksiyonunun değeri olmaktadır. Kril bireylerinin iki boyutlu yüzeyde zamana bağlı pozisyonları üç temel eylem sonunda gerçekleşmektedir (Hofmann vd. 2004) :

- Diğer kril bireylerinin sebep olduğu hareket (Movement induced by other krill individuals)
- Yem arama aktivitesi (Foraging activity)
- Fiziksel/Rastgele yayılma (Random diffusion)

n boyutlu karar uzayında zamana bağlı pozisyonlar, X_i için ifade edilen Lagrange modeli şu şekilde tanımlanır;

$$\frac{dX_i}{dt} = N_i + F_i + D_i \tag{3.26}$$

Burada; N_i diğer kril bireylerinin sebep olduğu hareketi, F_i yem arama hareketini ve D_i kril bireyinin fiziksel yayılmasını ifade etmektedir.

Diğer kril bireylerinin sebep olduğu hareket :

Her bir kril için, diğerlerinin sebep olduğu hareket aşağıdaki şekilde ifade edilir.

$$N_i^{yeni} = N^{maks}\alpha_i + \omega_n N_i^{eski} \tag{3.27}$$

$$\alpha_i = \alpha_i^{lokal} + \alpha_i^{hedef} \tag{3.28}$$

Burada; N^{maks}, maksimum sebep olunan hız, 0.01 (m/s) olarak alınmıştır. α_i sebep olunan hareketin doğrultusu, ω_n sebep olunan hareketin atalet ağırlığı, N_i^{eski} son sebep olunan hareket, α_i^{lokal} komşu bireyler tarafından sağlanan lokal etkiler ve α_i^{hedef} en iyi kril bireyi tarafından sağlanan hedef yönün etkisidir. Kril bireyinin hareketine komşu krillerin etkisi eşitlik 3.29 - 3.31 ile ifade edilir.

$$\alpha_i^{lokal} = \sum_{j=1}^{NN} \hat{K}_{i,j}\hat{X}_{i,j} \tag{3.29}$$

$$\hat{X}_{i,j} = \frac{x_j - x_i}{\left\|x_j - x_i\right\| + \varepsilon} \tag{3.30}$$

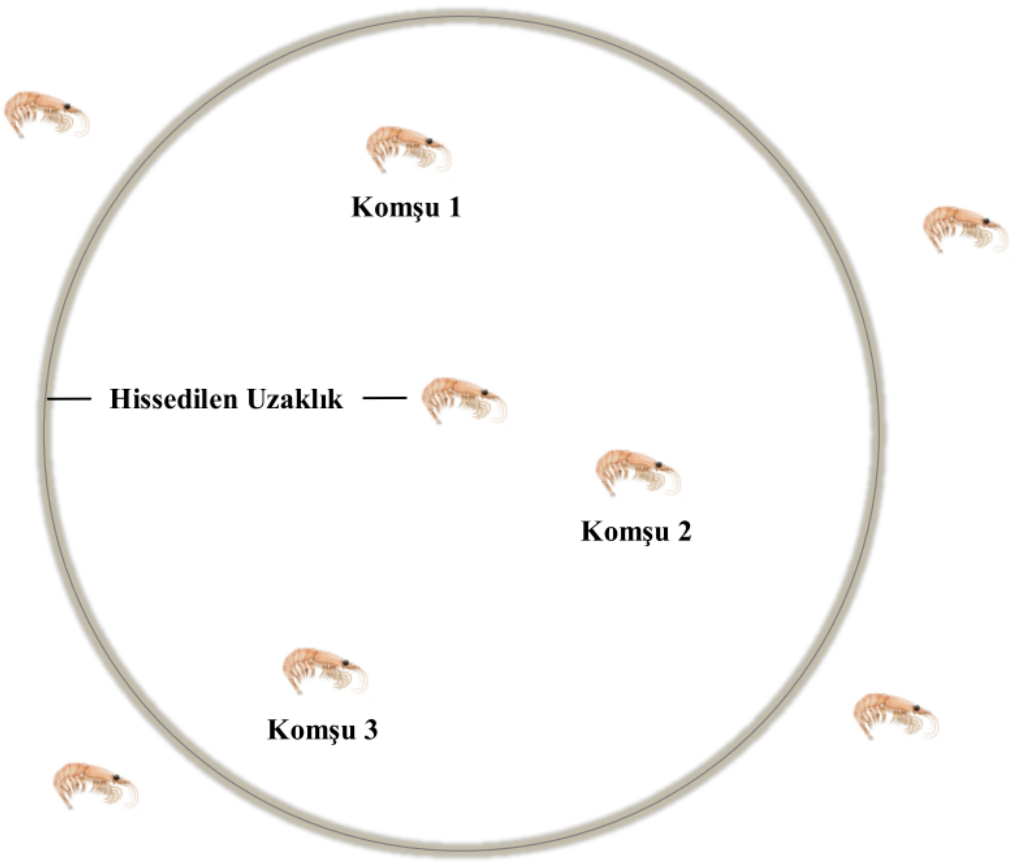
$$\hat{K}_{i,j} = \frac{K_i - K_j}{K^{enkötü} - K^{eniyi}} \tag{3.31}$$

Burada $K^{enkötü}$ ve K^{eniyi} kril bireylerinin şimdiye kadarki en kötü ve en iyi uygunluk değerlerini ifade etmektedir. K_i , i. kril bireyinin amaç fonksiyonu değerini temsil ederken, K_j, j. komşu bireyin amaç fonksiyonu değeridir. Her bir kril bireyinin pozisyonu

X ile ifade edilirken, NN toplam komşu sayısını göstermektedir. Çözüm sırasında yaşanacak herhangi bir tekilliğe (singularity) izin vermemek için küçük bir ε değeri kullanılmıştır.

3.29 - 3.31 nolu eşitliklerin sağ tarafları, birim vektörler ve normalize edilmiş uygunluk değerlerini içerir. Bu vektörler farklı komşular tarafından uyarılan yönleri gösterirken, her bir değer ise komşunun etkisini ifade eder. Komşu vektörleri, normalize edilmiş değer negatif veya pozitif olabileceğinden çekici veya itici olabilir.

Komşu seçim işlemi, d_s ile ifade edilen hissedilen uzaklık (sensing distance) baz alınarak, Şekil 3.43 'deki gibi yapılmaktadır.



Şekil 3.43 Hissedilen uzaklık ve komşuluk yapısı (Amir Hossein Gandomi ve Alavi 2012), (Gölcük vd. 2014)

$$d_{s,i} = \frac{1}{5N}\sum_{j=1}^{N}\left\|X_i - X_j\right\| \tag{3.32}$$

Burada $d_{s,i}$, i. kril bireyinin hissedilen uzaklığını gösterirken, N toplam kril bireyi sayısını vermektedir. Eşitlik 3.3.32'e göre eğer iki kril bireyi arasındaki uzaklık d_s 'den küçükse, kril bireylerinin komşu olduğu sonucu çıkartılmaktadır. En iyi amaç fonksiyonu değerine sahip olan kril bireyinin i. kril bireyi üzerine olan etkisi şu şekilde modellenmektedir:

$$\alpha_i^{hedef} = C^{eniyi}\widehat{K}_{i,eniyi}\widehat{X}_{i,eniyi} \tag{3.33}$$

C^{eniyi} , etki katsayısı olmak üzere şu şekilde tanımlanır.

$$C^{eniyi} = 2\left(rand + \frac{I}{I_{maks}}\right) \tag{3.34}$$

Burada *rand* terimi 0 ile 1 arasında rastgele üretilen bir sayı, I döngü sayısı ve I_{maks} ise maksimum döngü sayısıdır.

Yem Arama Aktivitesi :

Yem arama aktivitesi iki temel kavramla ilişkilidir. Bunlardan ilki yem lokasyonu, ikincisiyse önceki tecrübedir. Bu hareket i. kril bireyi için şu şekilde tanımlanmaktadır:

$$\mathrm{F}_i = \mathrm{V}_f\beta_i + \mathrm{w}_f F_i^{eski} \tag{3.35}$$

$$\beta_i = \beta_i^{yem} + \beta_i^{eniyi} \tag{3.36}$$

Burada V_f yem arama hızını göstermektedir ve Price 'a dayanarak 0.02 (m/s) olarak alınmıştır (H. J. Price 1989). w_f 0-1 arasında yem arama hareketinin atalet ağırlığı, F_i^{eski} son yem arama hareketi, β_i^{yem} yem çekiciliği β_i^{eniyi} i. kril bireyinin şu ana kadarki en iyi amaç fonksiyonu değerinin etkisidir. Her bir iterasyonda yemek merkezi şu şekilde tanımlanır:

$$x^{yem} = \frac{\sum_{i=1}^{N}(1 / K_i)X_i}{\sum_{i=1}^{N}(1 / K_i)} \tag{3.37}$$

Böylece i. kril bireyi için yem çekiciliği şu şekilde ifade edilir:

$$\beta_i^{yem} = C^{yem}\widehat{K}_{i,yem}\widehat{X}_{i,yem} \tag{3.38}$$

$$C^{yem} = 2\left(1 - \frac{I}{I_{maks}}\right) \tag{3.39}$$

i. kril bireyine ait en iyi amaç fonksiyonu değerinin etkisi şu şekilde modellenmektedir:

$$\beta_i^{eniyi} = \widehat{K}_{i,eniyi}\widehat{X}_{i,eniyi} \tag{3.40}$$

Burada $K_{i,eniyi}$ daha önceden ziyaret edilmiş en iyi pozisyona ait uygunluk değeridir.

Fiziksel Yayılma :

Rastlantısal bir süreç olan fiziksel yayılma şu şekilde formüle edilmektedir.

$$D_i = D^{\text{maks}}\left(1 - \frac{I}{I_{maks}}\right)\delta \tag{3.41}$$

Burada maksimum yayılma hızı D^{maks}, $D^{\text{maks}} \in [0.002,0.010]$ (m/s) ve yönlü vektör δ , -1 ile 1 arasında olan rastlantısal bir sabittir. Çözümün ilerleyen aşamalarında bu rastlantısal etkinin azaltılması için $\left(1 - \frac{I}{I_{maks}}\right)$ terimi kullanılmıştır.

3.4.2. Kril Sürüsü Algoritması Hareket Süreci

Genel olarak, tanımlanan hareketler genellikle kril bireyinin konumunu, en iyi uygunluk değerine doğru değiştirir. Yem aram hareketi ve diğer kril bireylerin neden olduğu hareket iki küresel ve iki yerel strateji içerir. Bunlar paralel olarak çalışır ve bu da KSO güçlü bir

algoritma yapar. i. kril bireyi için bu hareketlerin formülasyonlarına göre, yukarıda belirtilen her bir etkili faktörün (K_j; K^{eniyi}; K^{yem} veya K_i^{eniyi}) uygunluk değeri, i krilinin uygunluğundan daha iyi (daha az) ise, çekici bir etki, aksi takdirde itici bir etkiye sahiptir. Yukarıdaki formülasyonlarda, daha iyi bir uygunluğun, kril bireyin hareketinde daha etkili olduğu da açıktır. Bu yöntemde fiziksel yayılma rastgele bir arama yapar. Bir krilin t ila $t + \Delta t$ zaman aralığındaki konum vektörü, hareketin farklı etkili parametreleri kullanılarak aşağıdaki şekilde ifade edilir.

$$X_i(\, t + \Delta t\,) = X_i(\, t\,) + \Delta t \frac{dX_i}{dt} \tag{3.42}$$

Δt 'nin en önemli sabitlerden biri olduğu ve optimizasyon problemine göre dikkatle ayarlanması gerektiği unutulmamalıdır. Bunun nedeni, bu parametrenin hız vektörünün ölçek faktörü olarak çalışmasıdır. Δt tamamen arama alanına bağlıdır ve basitçe aşağıdaki formülden elde edilebilir.

$$\Delta t = C_t \sum_{j=1}^{NV} UB_j - LB_j \tag{3.43}$$

Burada NV toplam değişken sayısını, LB_jve UB_j j. değişkene (j =1,2, ..., NV), ait alt ve üst limitleri, C_t ise 0 ila 2 arasındaki sabit bir sayıyı göstermektedir. Küçük C_t değerleri kril bireylerinin alanı dikkatli bir şekilde aramasını sağlar.

3.4.3. Genetik Operatörler

Algoritmanın performansını artırmak için KSO yönteminin içerisine genetik üreme mekanizmaları eklenmiştir. Bunlar, klasik türevsel evrim (DE) algoritmasından esinlenen çaprazlama ve mutasyon mekanizmalarıdır.

Çaprazlama :

Çaprazlama operatörleri ilk olarak genetik algoritmalarda global optimuma ulaşma etkinliğini artırmak için kullanılmıştır. Çaprazlamanın vektörize versiyonu ise GA ‘nın

ayrı bir türü olan diferansiyel algoritmada kullanılır. KSO yönteminde adaptif vektörize çaprazlama şeması kullanılmıştır (Amir Hossein Gandomi ve Alavi 2012).

Binom ve üstel olarak uygulanabilen çaprazlama, çaprazlama olasılık sabiti Cr ile kontrol edilir. Binom şeması, d bileşenlerinin veya değişkenlerin / parametrelerin her birinde çaprazlama gerçekleştirir. 0 ile 1 arasında eşit dağıtılmış rasgele bir sayı üreterek, X_i, $x_{i,m}$ 'in m'inci bileşeni şu şekilde manipüle edilir:

$$x_{i,m} = \begin{cases} X_{r,m} & rand_{i,m} < Cr \\ X_{i,m} & değilse \end{cases} \tag{3.44}$$

$$Cr = 0.2\ \widehat{K}_{i,eniyi} \tag{3.45}$$

Burada $r \in \{1,2,\ldots,i-1,i+1,\ldots,N\}$. Bu yeni çaprazlama olasılığını kullanarak, global en iyinin çaprazlama olasılığı sıfıra eşitlenir ve uygunluğu azaldıkça artar. Böylece uygunluk değeri düşük olan bireyler uygunluk değeri daha iyi olanların değerlerini alma olasılığı artmış olur.

Mutasyon :

Mutasyon, evrimsel strateji (ES) ve DE gibi evrimsel algoritmalarda önemli bir rol oynar. Mutasyon, mutasyon olasılığı sabiti, Mu ile kontrol edilir. Burada kullanılan uyarlanabilen mutasyon şeması şu şekilde ifade edilir:

$$x_{i,m} = \begin{cases} x_{gbes,m} + \mu\left(x_{p,m} - x_{q,m}\right) & rand_{i,m} < Mu \\ x_{i,m} & değilse \end{cases} \tag{3.46}$$

$$Mu = 0.05\ /\ \widehat{K}_{i,eniyi} \tag{3.47}$$

Burada $p,q \in \{1,2,\ldots,i-1,i+1,\ldots,K\}$ ve μ 0 ile 1 arasında bir sayıdır. $\widehat{K}_{i,eniyi}$ ataması ise $K_i - K^{eniyi}$ ile hesaplanır. Bu yeni mutasyon olasılığını kullanarak, global en iyinin mutasyon olasılığı sıfıra eşitlenir ve uygunluğu azaldıkça artar.

3.4.4. Kril Sürüsü Algoritması

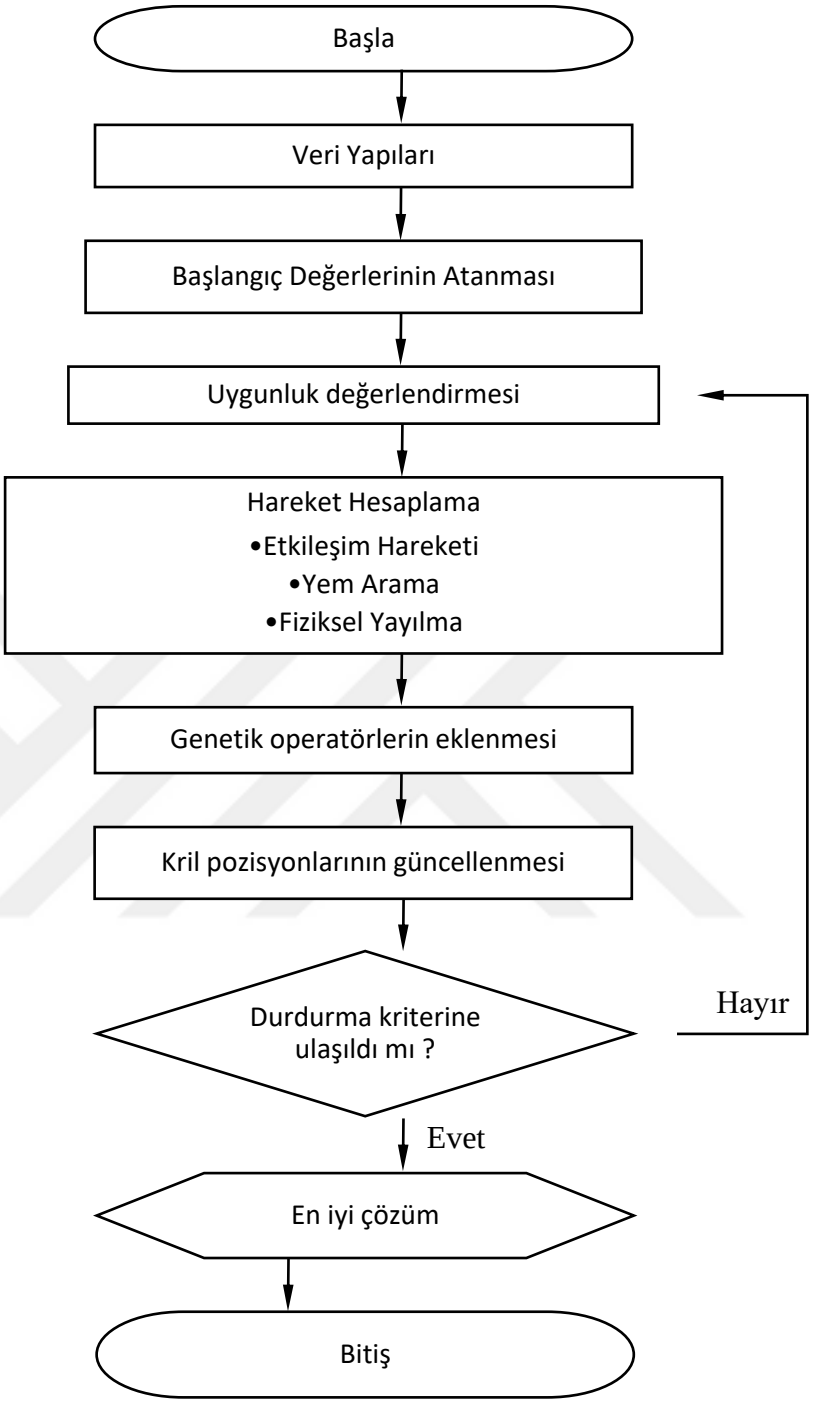
Kril Sürüsü (KS) algoritması şu adımlardan oluşur:

i. Veri Yapıları: Basit sınırların tanımlanması, algoritma parametrelerinin belirlenmesi vb.
ii. Başlatma: Arama alanında ilk popülasyonu rastgele oluşturun.
iii. Kondisyon değerlendirmesi: Her bir krill bireyin pozisyonuna göre değerlendirilmesi.
iv. Hareket hesaplama:
    - Diğer bireylerin varlığından kaynaklanan hareket
    - Yem arama hareketi
    - Fiziksel difüzyon
v. Genetik operatörlerin uygulanması
vi. Güncelleme: Arama alanındaki kril bireyinin konumunun güncellemesi
vii. Tekrarlama: durdurma kriterlerine ulaşılana kadar III. adıma gidin.
viii. Son

Bu algoritmanın akış şeması Şekil 3.44 ‘te gösterilmiştir.

3.7. Kaotik Kril Sürüsü Optimizasyonu

Bir önceki bölümde anlatılan çalışmalara ek olarak Gharavian ve ark., Wang ve ark., Mahdi ve ark. ile Bentouati ve ark. KSO optimizasyon yöntemini kaotik fonksiyonlarla birleştirerek Kaotik KSO (KKSO) algoritmasını geliştirmişlerdir (Gharavian, Yaghoobi, ve Keshavarzian 2013), (B. Liu vd. 2005), (G.-G. Wang, Hossein Gandomi, ve Hossein Alavi 2013), (G. G. Wang vd. 2014), (Bentouati, Chettih, ve El-Sehiemy 2017).



Şekil 3.44 Kril Sürüsü Algoritmasının basitleştirilmiş Akış Şeması (Amir Hossein Gandomi ve Alavi 2012)

Gharavian ve ark. global optimizasyon problemlerini iyileştirmek için kaos teorisi ile KSO algoritmasını birleştirmiştir. KSO' nun fiziksel yayılma adımında rastgele olarak belirlenen sabit sayı bu çalışmada lojistik haritalama fonksiyonu ile değiştirilerek mevcut algoritmanın kaos tabanlı olması sağlanmıştır. Bu çalışmada kaotik haritalama fonksiyonlarından sadece lojistik harita kullanılmıştır. Literatürdeki çeşitli kıyaslama fonksiyonları ile test edilen KKSO yönteminin performansının, mevcut KSO 'ye göre daha iyi olduğu belirtilmiştir (Gharavian, Yaghoobi, ve Keshavarzian 2013).

Wang ve ark., KSO algoritmasının performansını arttırmak için bu algoritmanın içerisine farklı kaotik yaklaşımlar eklemiştir. İlk çalışmalarında, orijinal KS algoritmasında başlangıç değeri 0.9 olup zamanla azalan atalet ağırlığı, w, kullanmak yerine, kaotik harita fonksiyonları ile hesaplamayı tercih etmişlerdir (G. G. Wang vd. 2014). Bir diğer çalışmalarında orijinal KSO yönteminin eşitlik 18'inde yer alan konum vektörü hesabına da müdahale ederek (Eşitlik 48) ikinci bir kaotik davranış eklemişlerdir. Bu yöntemi 32 tane kıyaslama fonksiyonu ile test ettikten sonra dişli grubu tasarımı probleminde uygulamışlardır.

$$x_i^{t+1}(j) = (1-\beta)x_i^{ft}(j) + \beta g^* + \alpha r \tag{3.48}$$

Eşitlik 3.48 'de β, α ve g^* parametreleri sırasıyla, çekim parametresi, kontrol parametresi ve mevcut en iyi global pozisyon olarak adlandırılırlar. r ise 0 ile 1 arasında değişen rastlantısal bir sayıdır. Bu eşitlikteki ana parametre, hız vektörünü ölçekleyen çekim parametresi β ' dır ve global arama yeteneğini etkilemektedir (G.-G. Wang, Hossein Gandomi, ve Hossein Alavi 2013). Bu çalışmada, yem arama aktivitesi içindeki çekim parametresi β, sabit bir değer almak yerine kaotik haritalama fonksiyonları ile tanımlanarak mevcut KSO algoritmasının kaos yöntemi ile birleştirilmesi sağlanmıştır. Çekim parametresinin kaotik olarak belirlenmesi lokal minimuma takılma problemini giderirken, global optimuma hızlı bir şekilde ulaşma imkanı da sağlayacaktır Test fonksiyonlarıyla yapılan karşılaştırmalardan elde edilen sonuçların, standart KSO yönteminden elde edilenlere göre daha iyi olduğu ya da en azından daha rekabetçi oldukları görülmüştür (G.-G. Wang, Hossein Gandomi, ve Hossein Alavi 2013), (G. G.

Wang vd. 2014). Wang ve ark. genetik operatörlerden elitizm yöntemini de sonraki çalışmalarında kaos ile birlikte kullanmışlardır(G. G. Wang vd. 2014).

Bentonuati ve ark. KSO algoritmasındaki atalet ağırlığı parametresini sabit bir değer almak yerine kaotik haritalama fonksiyonlar ile çözüm süresince kaotik olarak değiştirmişlerdir. Buna ek olarak orijinal algoritmadan farklı olarak elitizm yöntemi de dahil edilmiştir. Önerdikleri bu algoritmayı ekonomik dağıtım probleminin çözümünde kullanmış, elde ettikleri sonuçları literatürdeki diğer algoritmalardan elde edilenler ile karşılaştırmışlardır. Yeni algoritmanın performansının oldukça başarılı olduğu belirtilmiştir (Bentouati, Chettih, ve El-Sehiemy 2017).

Mevcut kril sürüsü optimizasyonunda rastgele belirlenen sayıların kullanıldığı adımlar şunlardır;

1. Başlangıç popülasyonun oluşturulması
2. En iyi kril bireyinin etkisini, α_i^{hedef} , hesabındaki etki katsayısı, C^{eniyi} hesabı (Eşitlik 3.33)
3. Fiziksel yayılma adımında yönlü vektör, δ (Eşitlik 3.41)
4. Çaprazlama Adımı (Eşitlik 3.44)

Orijinal KSO ’nın literatürde karşılaşılan kaotik uygulamaları ve bunların sınıflandırmaları Çizelge 3.3 ve Çizelge 3.4’de gösterilmiştir. Çizelge 3.3’de numaralandırılarak kullanılan değişikliklerin yapıldığı noktalar şunlardır.

1. Diğer kril bireylerinin sebep olduğu hareket
2. Yem arama hareketi
3. Fiziksel yayılma hareketi
4. Hareket Sonrası

Çizelge 3.3 literatürde karşılaşılan kaotik uygulamalar

Yayın	**Değişliğin Yapıldığı Yer**	**Değiştirilen Parametre**	**Denenen Harita Fonksiyonu Sayısı**	**Önerilen Harita Fonksiyonu**
(Gharavian, Yaghoobi, ve Keshavarzian 2013).	3	Yönlü vektör, δ	1	Logistic
(G. G. Wang vd., 2014)	1, 2	Atalet ağırlığı, w	10	Singer
(G.-G. Wang, Hossein Gandomi, ve Hossein Alavi 2013)	1, 2, 4	Atalet ağırlığı, w Konum vektörü, X	13	Singer
(Bentouati, Chettih, ve El-Sehiemy 2017)	1, 2	Atalet ağırlığı, w	12	Singer

Çizelge 3.4 Literatürde karşılaşılan kaotik uygulamaların sınıflandırması

Yayın	**Diğer krillerin sebep olduğu hareket**	**Yem Arama**	**Fiziksel Yayılma**	**Diğer**	**Değiştirilen Parametre**
(Gharavian, Yaghoobi, ve Keshavarzian 2013).			x		Yönlü vektör, δ
(G. G. Wang vd., 2014)	x	x			Atalet ağırlığı, w
(G.-G. Wang, Hossein Gandomi, ve Hossein Alavi 2013)	x	x		x	Çekim parametresi, β
(Bentouati, Chettih, ve El-Sehiemy 2017)	x	x			Atalet ağırlığı, w

Çizelge 3.4 ‘e göre literatürde yer alan KKS uygulamaların daha çok yem arama aktivitesi üzerinde yoğunlaştığını söylemek mümkündür.

3.8. Önerilen Kaotik Kril Sürüsü Optimizasyonu

Literatürdeki kaotik KSO çalışmaları incelendiğinde aşağıdaki süreçlerde herhangi bir kaotik yaklaşımın uygulanmadığı görülmüştür.

1. Başlangıç popülasyonun oluşturulması
2. En iyi kril bireyinin etkisini, α_i^{hedef} , hesabındaki etki katsayısı, C^{eniyi} hesabı
3. Çaprazlama Adımı

Bu nedenle bu çalışmada yukarıdaki adımlara da kaotik harita fonksiyonlarının eklendiği yeni algoritmalar oluşturulmuştur. Önerilen algoritmalar ile yapılan çalışmalar şunlardır.

1. Farklı harita fonksiyonlarının literatürdeki kıyaslama fonksiyonları ile test edilmesi.
2. En iyi seçilen harita fonksiyonu ile elde edilen sonuçların aşağıda belirtilen literatürdeki mevcut optimizasyon algoritmaları ile karşılaştırılması.
    a. KSO
    b. Kaotik KSO (Wang, 2014)
    c. DGA
    d. PSO
3. Geliştirilen kaotik KSO yöntemi ile süspansiyon burcunun geometrik optimizasyonu.

1. kısımda test edilen harita fonksiyonları şunlardır:

1. Chebyshev
2. Circle
3. Gauss/mouse
4. Iterative

5. Logistic
6. Piecewise
7. Sine
8. Singer
9. Sinusoidal
10. Tent
11. Cubic
12. Intermittency
13. Liebovitch
14. ICMIC

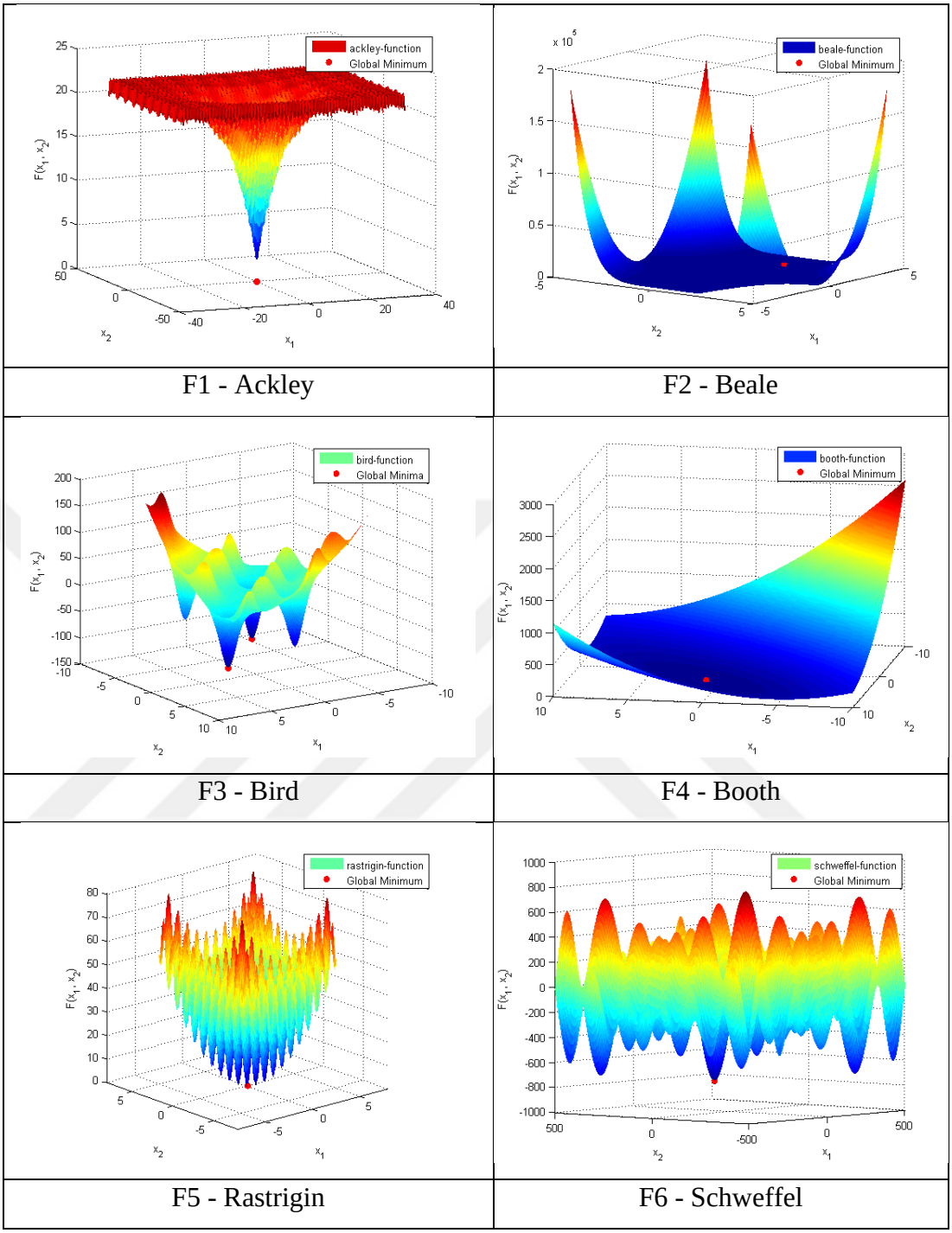
Yukarıda belirtilen harita fonksiyonları kullanılarak önerilen optimizasyon algoritmalarının test edileceği kıyaslama fonksiyonları ise aşağıda listelenmiştir.

1. Ackley
2. Beale
3. Bird
4. Booth
5. Rastrigin
6. Schweffel

Bu fonksiyonların denklemleri, değişken aralıkları, global minimumları ve global minimum noktası Çizelge 3.5 ‘te, grafiksel gösterimleri ise Şekil 3.45’te gösterilmiştir. Test edilen fonksiyonların grafiksel gösterimlerinden de açıkça görüleceği üzere 2 ve 4 numaralı Beale ve Booth fonksiyonları dışında diğer tüm fonksiyonların çok sayıda yerel minimumları bulunmaktadır. Yapılan deneme çözümlerinde test edilen fonksiyonlar 2 değişkenli olarak ele alınmıştır.

Çizelge 3.5 Test Fonksiyonları ve Çözümleri

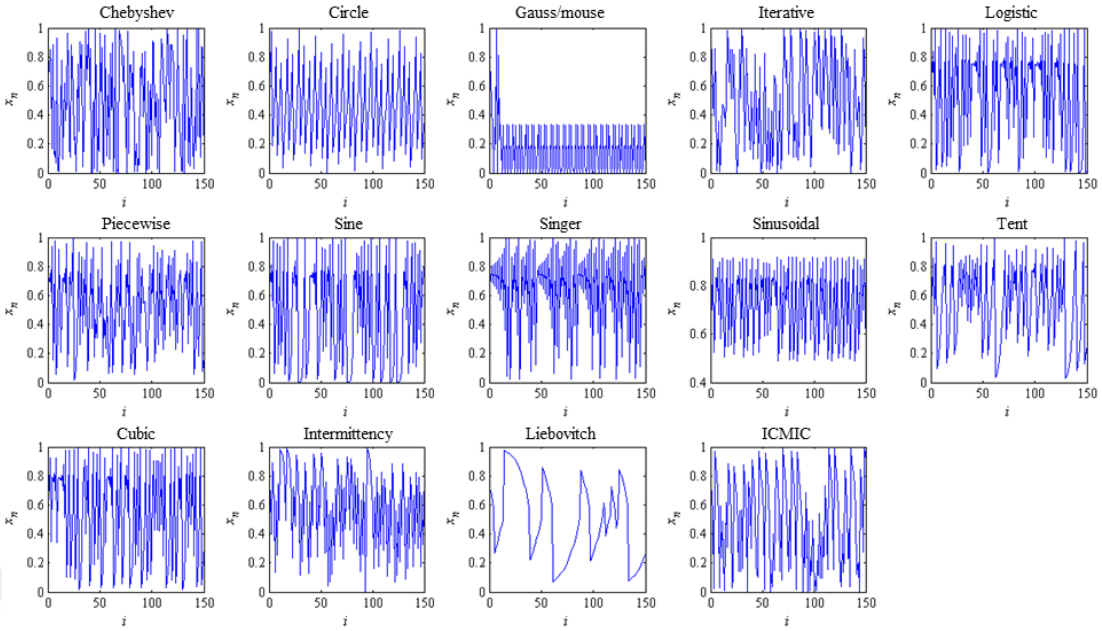
Fonksiyon	**Denklem**	**Aralık**	**Min.**	**Çözüm**
Ackley	$f(X) = -20exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - exp(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)) + 20 + e$	$-32 \leq x_i \leq 32$	0	(0,0)
Beale	$f(X) = [1.5 - x_1(1 - x_2)]^2 + [2.25 - x_1(1 - x_2^2)]^2 + [2.625 - x_1(1 - x_2^3)]^2$	$-4.5 \leq x_i \leq 4.5$	0	(3,0.5)
Bird	$f(X) = (x_1 - x_2)^2 + \sin(x_1) \cdot e^{[1-\cos(x_2)]^2} + \cos(x_2) \cdot e^{[1-\sin(x_1)]^2}$	$-2\pi \leq x_i \leq 2\pi$	−106.7645367198034	(4.701055751981055,3.15294601960139 1),(−1.582142172055011,−3.130246799635430)
Booth	$f(X) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	$-10 \leq x_i \leq 10$	0	(0,0)
Rastrigin	$f(X) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$-5.12 \leq x_i \leq 5.12$	0	(0,0)
Schweffel	$f(X) = \sum_{i=1}^{n} (-x_i sin(\sqrt{\lvert x_i \rvert})) + \alpha \cdot n$ $\alpha = 418.982887$	$-512 \leq x_i \leq 512$	0	(420.968746, 420.968746)



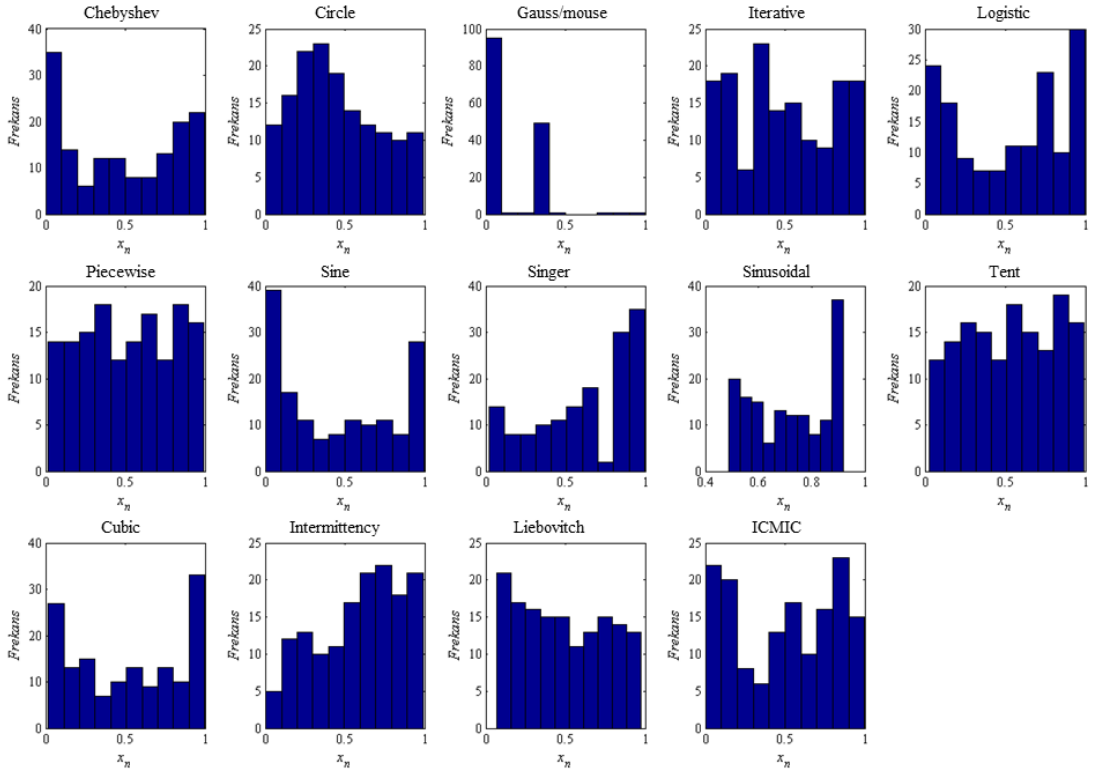
F1 - Ackley	F2 - Beale
F3 - Bird	F4 - Booth
F5 - Rastrigin	F6 - Schweffel

Şekil 3.45 Kıyaslama Fonksiyonlar

Yukarıda belirtilen harita fonksiyonlarıyla üretilen serilere ait serpilme grafikleri Şekil 3.46'de, çubuk dağılım grafikleri ise Şekil 3.47 'te gösterilmiştir.



Şekil 3.46 Yukarıdaki harita fonksiyonlarıyla üretilen serilere ait serpilme grafikleri



Şekil 3.47 Yukarıdaki harita fonksiyonlarıyla üretilen serilere ait çubuk dağılım grafikleri

Önerilen kaotik kril sürüsü optimizasyonunda kullanılan sabitler ve değerleri aşağıda Çizelge 3.6 'da gösterilmiştir. Bu değerler literatürde yer aldığı haliyle kullanılmıştır (G. Wang vd. 2014)(Amir Hossein Gandomi ve Alavi 2012), (G. G. Wang vd. 2014).

Çizelge 3.6 KS algoritmalarında kullanılan sabitler

Sabit	**Açıklama**	**Değer**
N^{maks}	Maksimum sebep olunan hız	0.01
V_f	Yem arama hızı	0.02
w_f	Atalet ağırlığı	0.9 – 0.1
D^{maks}	Maksimum yayılma hızı	0.005
C_t	Zaman adımı katsayısı	0.5
NK	Popülasyondaki kril sayısı	50
MI	Maksimum iterasyon sayısı	50

Önerilen algoritmanın testinde yukarıda tanımlanan her bir test fonksiyonu her bir harita fonksiyonu ile ayrı ayrı koşturulmuştur. Yarı sezgisel algoritmaların doğrulukları tek bir çözüm ile ispatlanamayacağı her bir çözüm seti 100 kez tekrarlanmıştır. Elde edilen sonuçların istatistiksel olarak değerlendirmeleri aşağıda gösterilmiştir.

Mevcut KS yönteminde rastgele olarak belirlenen sayıların kullanıldığı; başlangıç popülasyonun oluşturulması ve etki katsayısı C^{eniyi} hesaplanması adımlarında kaotik harita fonksiyonları ile üretilen sayılar kullanılmıştır. Yapılan çalışmalarda hem birlikte kullanılmasının hem de ayrı yarı kullanılmasının etkileri farklı harita fonksiyonları ile birlikte araştırılmıştır. Yukarıda belirtilen test fonksiyonları aşağıdaki algoritmalar ile test edilmiştir.

1. Mevcut KS algoritması. (Algoritma 1)
2. Mevcut Wang algoritması. (Algoritma 2)
3. Wang + Sadece başlangıç popülasyonunun pozisyonlarının kaotik olduğu KKS (Algoritma 3 - Önerilen 1)
4. Wang + Sadece C^{eniyi} hesabının kaotik olduğu KKS (Algoritma 4 - Önerilen 2)

5. Wang + Her iki adımın da kaotik olduğu KKS (Algoritma 5 - Önerilen 3)
6. Başlangıç popülasyonunun pozisyonlarının ve C^{eniyi} hesabının kaotik olduğu KKS (Algoritma 6 - Önerilen 4)
7. Parçacık Sürüsü Optimizasyonu, PSO
8. Genetik Algoritma, GA
9. Diferansiyel Gelişim Algoritması, DGA

Önerilen kaotik kril sürüsü algoritması, çok paradigmalı sayısal hesaplama yazılımı olan Matlab yazılımı ile programlanmıştır. Bu programın yazımında Wang ve ark. (2014) tarafından geliştirilen Matlab kodundan yararlanılmıştır (G. G. Wang vd. 2014). Yukarıda tanımlanan algoritmalar Matlab ile yazılan kodun içinde değişkenler ile kontrol edilerek istenilen algoritmanın çalıştırılması sağlanmıştır.

Gandomi ve ark. KS optimizasyonunda elitizm yer almamasına rağmen Wang 'a ait KKS algoritmasında elitizm kullanılmıştır. Bu nedenle geliştirilen yeni algoritmalarda da elitizm kullanılmaya devam edilmiştir.

Karşılaştırmalarda kullanılan DGA, PSO ve GA algoritmalarında sırasıyla Mostapha Kalami Heris (2015), Mahamad Nabab Alam (2016) ve Seyedali Mirjalili (2021) tarafından geliştirilen Matlab kodları kullanılmıştır (Heris 2015), (Alam 2016), (Mirjalili 2021).

4. BULGULAR

Bu bölümde bir önceki bölümde önerilen KKS algoritmalarının performans testlerinden elde edilen sonuçlar raporlanmıştır. Elde edilen sonuçlar öncelikle hem önerilen farklı algoritmalar ile hem de literatürdeki mevcut KS algoritmaları (kaotik olan ve almayan) ile karşılaştırılmıştır. Bu karşılaştırmalarda denenen her bir kaotik harita fonksiyonundan elde edilen sonuçlara da yer verilmiştir. Bu sonuçlara göre en başarılı algoritma ve harita fonksiyonu seçilmiş, daha sonra seçilen bu algoritmanın performansı; PSO, GA ve DGA ile de karşılaştırılmıştır. Ardından da seçilen algoritma, endüstriyel bir problem olan kauçuk burç optimizasyonu üzerinde uygulanmıştır.

4.1. Kaotik Kril Sürüsü Yönteminin Performans Testleri

Algoritmaların istatistiksel karşılaştırmalarında, her biri için yapılan 100 ayrı çözümden elde edilen aşağıdaki parametreler kullanılmıştır. Elde edilen sonuçlar Çizelge, çizgi grafik ve radar grafik olarak raporlanmıştır.

- En kötüler (**Çizelge 4.1** - **Çizelge 4.9**)
- Standart sapmalar (**Çizelge 4.10** - **Çizelge 4.18**)
- Ortalamalar (**Çizelge 4.19** - **Çizelge 4.27**)
- En iyiler (**Çizelge 4.28** - **Çizelge 4.36**)

Her bir algoritmadan elde edilen; ortalamalar, Şekil 4.2 ve Şekil 4.2'de, standart sapmalar, Şekil 4.3 ve Şekil 4.4'te, en iyiler ise Şekil 4.5 ve Şekil 4.6 'da çizgi ve radar grafik olarak karşılaştırılmıştır.

4.1.1. En Kötüler

Çizelge 4.1 Mevcut KS yönteminden (Algoritma 1) elde edilen En Kötü Değerler

	Değer
F1	2.00E-07
F2	1.35E+00
F3	-5.67E+01
F4	1.25E-07
F5	3.48E+00
F6	2.74E+02

Çizelge 4.2 KKS Wang Algoritmasıyla (Algoritma 2) elde edilen En Kötü Değerler

	Değer
F1	2.76E-04
F2	9.56E-01
F3	-8.73E+01
F4	1.08E-06
F5	5.00E+00
F6	2.96E+02

Çizelge 4.3 Önerilen KKS algoritmasıyla (Algoritma 3) elde edilen En Kötü Değerler

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	En İyi Harita
F1	2.96E-04	6.33E-04	3.92E-04	2.78E-04	1.05E-03	6.16E+00	2.49E-04	6.63E-04	2.00E+01	2.99E-04	1.10E-03	1.36E-03	1.96E+01	1.22E-03	7
F2	2.39E+00	2.03E+00	1.63E+00	2.71E+00	1.55E+00	1.00E+00	1.26E+00	7.76E-01	9.79E+00	1.12E+00	1.22E+00	9.76E-01	9.39E+00	1.77E+00	8
F3	-5.47E+01	-4.84E+01	-8.73E+01	-8.73E+01	-8.50E+01	-8.73E+01	-8.15E+01	-7.71E+01	1.49E+00	-8.73E+01	-8.26E+01	-8.73E+01	1.49E+00	-8.73E+01	10
F4	1.79E-06	3.93E-07	7.11E-06	1.84E-06	1.83E-06	6.94E-07	5.04E-06	4.87E-07	2.97E+01	2.09E-07	2.08E-06	3.09E-06	1.63E-02	6.53E+00	10
F5	4.97E+00	4.97E+00	5.69E+00	3.98E+00	4.97E+00	3.98E+00	3.98E+00	8.95E+00	4.97E+01	2.57E+00	1.36E+01	4.97E+00	1.29E+01	4.97E+00	10
F6	2.32E+02	3.36E+02	2.97E+02	2.17E+02	1.79E+02	2.96E+02	1.58E+02	2.17E+02	2.30E+02	2.37E+02	1.18E+02	2.21E+02	4.15E+02	3.27E+02	11

Çizelge 4.4 Önerilen KKS algoritmasıyla (Algoritma 4) elde edilen En Kötü Değerler

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	En İyi Harita
F1	4.60E-04	3.54E-04	5.19E-04	5.52E+00	2.52E-04	6.41E+00	2.03E-04	2.81E-04	1.98E-04	1.26E-04	2.60E-04	6.58E+00	1.40E-04	4.69E-04	10
F2	1.25E+00	9.96E-01	1.06E+00	2.27E+00	1.14E+00	9.48E-01	9.07E-01	1.62E+00	1.93E+00	2.00E+00	9.89E-01	1.36E+00	2.52E+00	1.49E+00	7
F3	-8.73E+01	-7.71E+01	-8.73E+01	-8.47E+01	-8.73E+01	-7.51E+01	-8.73E+01	-8.73E+01	-8.73E+01	-8.03E+01	-5.83E+01	-7.85E+01	-8.73E+01	-8.25E+01	5
F4	9.94E-07	1.94E-06	1.10E-05	1.23E-06	1.57E+00	1.69E-06	1.16E-06	1.56E+00	9.65E-07	3.39E-06	6.82E-06	3.32E-06	4.09E-07	8.30E-07	13
F5	1.99E+00	4.97E+00	1.99E+00	3.98E+00	3.98E+00	3.98E+00	1.99E+00	4.03E+00	1.99E+00	1.99E+00	4.97E+00	1.99E+00	1.99E+00	3.71E+00	3
F6	2.37E+02	2.81E+02	2.96E+02	2.37E+02	2.39E+02	2.37E+02	2.37E+02	2.37E+02	2.33E+02	2.59E+02	2.37E+02	2.37E+02	2.37E+02	2.37E+02	9

Çizelge 4.5 Önerilen KKS algoritmasıyla (Algoritma 5) elde edilen En Kötü Değerler

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	En İyi Harita
F1	2.18E-04	2.66E-02	2.65E-04	2.73E-04	2.72E-03	7.62E-04	2.95E+00	3.04E-03	2.00E+01	7.10E-04	2.92E-04	5.54E-04	1.98E+01	3.30E-04	1
F2	1.66E+00	1.39E+00	1.44E+00	1.63E+00	1.04E+00	1.13E+00	1.66E+00	1.76E+00	9.76E+00	1.00E+00	1.23E+00	3.57E+00	3.65E+00	1.06E+00	10
F3	-6.20E+01	-4.84E+01	-8.73E+01	-8.73E+01	-6.90E+01	-8.73E+01	-8.16E+01	-8.73E+01	1.50E+00	-8.73E+01	-8.79E+01	-8.73E+01	-4.84E+01	-8.73E+01	11
F4	6.18E-06	3.32E-06	4.97E-06	4.42E-06	7.54E-06	1.18E-06	2.67E-06	2.86E+01	1.46E+02	6.47E-07	1.72E-06	8.59E-01	4.54E-05	3.94E-07	14
F5	1.82E+01	4.97E+00	1.99E+00	1.46E+01	3.98E+00	1.99E+00	1.06E+01	8.95E+00	4.97E+01	3.98E+00	3.98E+00	4.97E+00	1.29E+01	4.97E+00	3
F6	2.05E+02	3.56E+02	3.38E+02	2.96E+02	1.78E+02	2.37E+02	1.60E+02	1.90E+02	2.30E+02	2.37E+02	1.63E+02	2.33E+02	4.15E+02	2.33E+02	7

Çizelge 4.6 KKS algoritmasıyla (Algoritma 6) elde edilen En Kötü Değerler

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	En İyi Harita
F1	2.18E-04	9.93E-04	1.56E-05	3.59E-05	1.35E-03	9.09E-04	9.71E-05	1.89E-04	2.00E+01	1.02E-05	4.13E-05	1.47E-03	4.12E-04	2.21E-04	10
F2	7.82E-01	8.42E-01	7.62E-01	7.62E-01	7.64E-01	7.62E-01	7.89E-01	9.50E+01	9.52E+00	8.11E-01	7.62E-01	7.62E-01	9.39E+00	7.62E-01	6
F3	-87.31088273	-8.73E+01	-8.73E+01	-8.73E+01	-8.73E+01	-8.73E+01	-8.73E+01	-8.73E+01	1.49E+00	-8.73E+01	-8.73E+01	-8.73E+01	1.49E+00	-8.73E+01	1
F4	9.52E-09	3.90E-08	1.00E-08	4.76E-09	2.86E-08	4.75E-09	4.72E-08	3.81E-08	1.46E+02	5.24E-09	6.00E-08	4.09E-09	1.52E-08	1.08E-08	12
F5	1.99E+00	1.99E+00	9.95E-01	1.99E+00	9.95E-01	9.95E-01	9.95E-01	9.95E-01	2.49E+01	9.95E-01	9.95E-01	1.99E+00	1.59E+01	9.95E-01	7
F6	1.18E+02	2.37E+02	2.37E+02	2.30E+02	1.18E+02	2.96E+02	1.18E+02	2.17E+02	2.30E+02	2.37E+02	1.18E+02	2.17E+02	3.36E+02	2.37E+02	1

Çizelge 4.7 PSO ile elde edilen En Kötü Değerler

	Değer
F1	3.97E-02
F2	7.62E-01
F3	-1.07E+02
F4	1.04E-04
F5	9.98E-01
F6	2.30E+02

Çizelge 4.8 GA ile elde edilen En Kötü Değerler

	Değer
F1	3.95E+00
F2	3.54E+00
F3	-4.65E+01
F4	2.77E+01
F5	1.39E+01
F6	4.62E+02

Çizelge 4.9 DGA ile elde edilen En Kötü Değerler

	Değer
F1	9.68E-04
F2	6.34E-07
F3	-1.07E+02
F4	3.45E-07
F5	1.24E+00
F6	1.15E+02

4.1.2. Standart Sapmalar

Çizelge 4.10 Mevcut KS yönteminden (Algoritma 1) elde edilen sonuçların standart sapmaları

	Değer
F1	2.14E-08
F2	3.15E-01
F3	6.80E+00
F4	1.28E-08
F5	6.80E-01
F6	7.80E+01

Çizelge 4.11 KKS Wang algoritmasıyla (Algoritma 2) elde edilen sonuçların standart sapmaları

	Değer
F1	4.10E-05
F2	2.61E-01
F3	5.18E+00
F4	1.50E-07
F5	7.30E-01
F6	7.48E+01

Çizelge 4.12 Önerilen KKS algoritmasıyla (Algoritma 3) elde edilen sonuçların standart sapmaları

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	En İyi Harita
F1	3.81E-05	1.18E-04	5.04E-05	5.23E-05	1.15E-04	6.16E-01	4.48E-05	1.16E-04	9.73E+00	4.27E-05	1.53E-04	1.40E-04	1.96E+00	1.26E-04	1
F2	3.95E-01	4.33E-01	3.12E-01	3.99E-01	3.52E-01	3.01E-01	3.07E-01	1.63E-01	3.62E+00	2.43E-01	2.99E-01	1.42E-01	9.91E-01	3.61E-01	12
F3	7.96E+00	9.95E+00	4.98E+00	4.01E+00	5.82E+00	3.92E+00	4.48E+00	3.53E+00	4.53E+01	4.43E+00	5.47E+00	4.69E+00	1.40E+01	3.97E+00	8
F4	2.37E-07	5.76E-08	7.85E-07	2.93E-07	2.27E-07	8.58E-08	6.89E-07	5.64E-08	4.01E+00	2.48E-08	2.86E-07	3.12E-07	1.63E-03	6.53E-01	10
F5	9.78E-01	1.38E+00	8.03E-01	7.28E-01	8.08E-01	6.64E-01	8.89E-01	1.62E+00	1.17E+01	6.04E-01	1.59E+00	9.63E-01	2.38E+00	8.85E-01	10
F6	6.20E+01	8.52E+01	7.97E+01	6.05E+01	5.05E+01	7.34E+01	5.52E+01	4.51E+01	7.91E+01	7.47E+01	4.48E+01	7.36E+01	1.05E+02	7.50E+01	11

Çizelge 4.13 Önerilen KKS algoritmasıyla (Algoritma 4) elde edilen sonuçların standart sapmaları

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	En İyi Harita
F1	4.73E-05	4.68E-05	7.27E-05	5.52E-01	3.08E-05	6.41E-01	2.98E-05	3.43E-05	2.29E-05	1.98E-05	3.22E-05	6.58E-01	2.54E-05	5.66E-05	10
F2	3.35E-01	2.87E-01	2.44E-01	4.32E-01	2.77E-01	2.49E-01	2.44E-01	3.17E-01	3.29E-01	3.99E-01	2.49E-01	3.17E-01	3.56E-01	3.08E-01	7
F3	5.26E+00	5.70E+00	3.41E+00	5.22E+00	2.83E+00	5.27E+00	4.81E+00	4.41E+00	4.57E+00	5.45E+00	5.85E+00	4.54E+00	4.25E+00	5.13E+00	5
F4	1.18E-07	2.75E-07	1.12E-06	1.40E-07	1.57E-01	2.13E-07	1.67E-07	1.56E-01	1.13E-07	3.41E-07	7.02E-07	3.36E-07	5.13E-08	1.12E-07	13
F5	6.28E-01	7.27E-01	5.36E-01	6.69E-01	7.24E-01	6.72E-01	6.00E-01	6.97E-01	5.96E-01	6.54E-01	7.01E-01	6.42E-01	6.16E-01	6.81E-01	3
F6	6.67E+01	6.99E+01	7.32E+01	7.15E+01	7.75E+01	7.08E+01	7.98E+01	7.14E+01	6.80E+01	7.47E+01	6.47E+01	7.82E+01	6.69E+01	7.41E+01	11

Çizelge 4.14 Önerilen KKS algoritmasıyla (Algoritma 5) elde edilen sonuçların standart sapmaları

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	En İyi Harita
F1	3.38E-05	2.66E-03	4.11E-05	4.17E-05	2.73E-04	8.07E-05	2.95E-01	3.45E-04	9.61E+00	7.82E-05	4.44E-05	7.12E-05	3.23E+00	5.44E-05	1
F2	3.27E-01	4.28E-01	3.57E-01	3.82E-01	3.00E-01	2.70E-01	3.35E-01	3.38E-01	3.80E+00	2.09E-01	3.16E-01	4.10E-01	5.26E-01	2.87E-01	10
F3	6.71E+00	1.34E+01	2.06E+00	3.36E+00	4.78E+00	3.22E+00	4.46E+00	1.97E+00	4.09E+01	4.53E+00	1.97E+00	4.65E+00	1.06E+01	4.28E+00	8
F4	8.27E-07	3.91E-07	5.01E-07	4.47E-07	9.39E-07	1.38E-07	3.40E-07	2.86E+00	2.00E+01	8.86E-08	2.68E-07	8.59E-02	4.54E-06	6.68E-08	14
F5	1.90E+00	1.46E+00	6.08E-01	1.54E+00	8.77E-01	6.28E-01	1.38E+00	1.79E+00	1.28E+01	6.69E-01	7.31E-01	8.67E-01	1.84E+00	9.88E-01	3
F6	5.41E+01	9.40E+01	8.54E+01	6.39E+01	5.44E+01	7.02E+01	5.30E+01	3.99E+01	8.24E+01	7.19E+01	5.05E+01	6.56E+01	9.80E+01	6.77E+01	8

Çizelge 4.15 Önerilen KKS algoritmasıyla (Algoritma 6) elde edilen sonuçların standart sapmaları

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	En İyi Harita
F1	2.18E-05	8.44E-08	3.39E-08	5.90E-08	1.72E-08	9.45E-08	4.87E-08	1.65E-05	9.47E+00	1.91E-08	1.35E-08	2.24E-07	2.51E+00	1.16E-07	11
F2	1.51E-01	5.09E-01	3.36E-01	3.37E-01	4.16E-01	3.15E-01	4.42E-01	3.63E-01	3.83E+00	3.88E-01	4.81E-01	4.20E-01	4.43E-01	3.34E-01	1
F3	1.945361797	1.08E+01	4.38E+00	3.51E+00	5.53E+00	4.31E+00	8.95E+00	1.98E+00	4.91E+01	5.02E+00	6.80E+00	5.54E+00	1.44E+01	5.57E+00	1
F4	1.30E-09	2.03E-08	5.59E-09	1.85E-08	2.01E-08	9.47E-10	5.18E-09	3.59E-09	8.06E+00	2.61E-09	7.65E-09	6.24E-10	2.39E-08	6.62E-09	12
F5	3.88E-01	1.51E+00	7.35E-01	7.29E-01	1.05E+00	5.99E-01	1.00E+00	1.65E+00	1.24E+01	6.15E-01	7.04E-01	1.17E+00	2.12E+00	8.73E-01	1
F6	4.67E+01	8.12E+01	6.88E+01	6.58E+01	5.44E+01	6.58E+01	5.41E+01	4.11E+01	8.12E+01	7.36E+01	5.78E+01	6.17E+01	1.01E+02	6.84E+01	8

Çizelge 4.16 PSO ile elde edilen sonuçların standart sapmaları

	Değer
F1	7.25E-03
F2	7.62E-02
F3	7.17E-05
F4	1.49E-05
F5	1.02E-01
F6	6.35E+01

Çizelge 4.17 GA ile elde edilen sonuçların standart sapmaları

	Değer
F1	9.87E-01
F2	4.50E-01
F3	1.45E+01
F4	6.40E+00
F5	2.89E+00
F6	8.98E+01

Çizelge 4.18 DGA ile elde edilen sonuçların standart sapmaları

	Değer
F1	2.14E-04
F2	1.93E-08
F3	2.06E-02
F4	5.72E-08
F5	1.31E-01
F6	3.77E+01

4.1.3. Ortalamalar

Çizelge 4.19 Mevcut KS yönteminden (Algoritma 1) elde edilen sonuçların ortalamaları

	Değer
F1	5.84E-09
F2	1.51E-01
F3	-1.05E+02
F4	2.00E-09
F5	6.41E-01
F6	8.10E+01

Çizelge 4.20 KKS Wang Algoritmasıyla (Algoritma 2) elde edilen sonuçların ortalamaları

	Değer
F1	1.42E-05
F2	1.14E-01
F3	-1.05E+02
F4	3.74E-08
F5	6.95E-01
F6	7.61E+01

Çizelge 4.21 Önerilen KKS Algoritmasıyla (Algoritma 3) elde edilen sonuçların ortalamaları

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	En İyi Harita
F1	1.52E-05	4.13E-05	2.13E-05	2.55E-05	3.62E-05	6.17E-02	2.35E-05	7.16E-05	9.16E+00	1.64E-05	4.10E-05	3.94E-05	1.96E-01	3.00E-05	1
F2	2.11E-01	2.02E-01	1.39E-01	1.80E-01	2.27E-01	1.54E-01	1.54E-01	4.54E-02	2.10E+00	9.15E-02	1.55E-01	3.58E-02	3.15E-01	1.98E-01	12
F3	-1.04E+02	-1.01E+02	-1.05E+02	-1.06E+02	-1.04E+02	-1.06E+02	-1.06E+02	-1.06E+02	-6.78E+01	-1.06E+02	-1.05E+02	-1.05E+02	-1.02E+02	-1.06E+02	8
F4	7.34E-08	2.20E-08	1.32E-07	1.03E-07	7.37E-08	2.69E-08	2.06E-07	1.75E-08	7.08E-01	7.52E-09	7.29E-08	4.82E-08	1.63E-04	6.53E-02	10
F5	8.47E-01	1.68E+00	5.82E-01	6.33E-01	7.73E-01	5.93E-01	8.51E-01	1.38E+00	1.46E+01	4.77E-01	9.76E-01	9.63E-01	1.58E+00	8.87E-01	10
F6	4.87E+01	1.43E+02	9.93E+01	5.24E+01	2.91E+01	7.93E+01	3.97E+01	1.70E+01	1.14E+02	7.28E+01	2.10E+01	5.46E+01	1.66E+02	7.69E+01	8

Çizelge 4.22 Önerilen KKS Algoritmasıyla (Algoritma 4) elde edilen sonuçların ortalamaları

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	En İyi Harita
F1	1.30E-05	1.85E-05	2.32E-05	5.52E-02	1.32E-05	6.41E-02	1.65E-05	1.34E-05	8.83E-06	1.13E-05	1.53E-05	6.58E-02	1.34E-05	1.80E-05	9
F2	1.68E-01	1.35E-01	1.04E-01	2.12E-01	1.27E-01	9.48E-02	9.78E-02	1.34E-01	1.27E-01	1.95E-01	9.67E-02	1.51E-01	1.27E-01	1.16E-01	6
F3	-1.05E+02	-1.05E+02	-1.06E+02	-1.05E+02	-1.06E+02	-1.05E+02	-1.05E+02	-1.05E+02	-1.05E+02	-1.05E+02	-1.06E+02	-1.06E+02	-1.06E+02	-1.05E+02	3
F4	3.67E-08	6.98E-08	1.63E-07	4.18E-08	1.57E-02	5.97E-08	3.97E-08	1.56E-02	2.99E-08	5.41E-08	9.83E-08	5.05E-08	1.64E-08	3.30E-08	13
F5	5.00E-01	5.58E-01	4.65E-01	5.46E-01	5.56E-01	7.29E-01	5.06E-01	6.41E-01	4.97E-01	5.98E-01	6.21E-01	5.61E-01	5.81E-01	6.14E-01	3
F6	6.98E+01	6.58E+01	7.47E+01	5.74E+01	7.35E+01	7.17E+01	7.25E+01	8.62E+01	6.98E+01	7.77E+01	7.57E+01	8.64E+01	5.88E+01	7.24E+01	4

Çizelge 4.23 Önerilen KKS Algoritmasıyla (Algoritma 5) elde edilen sonuçların ortalamaları

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	En İyi Harita
F1	1.53E-05	3.00E-04	1.97E-05	1.70E-05	4.34E-05	2.17E-05	2.96E-02	1.10E-04	1.15E+01	2.27E-05	2.04E-05	2.76E-05	5.66E-01	2.28E-05	1
F2	1.45E-01	2.35E-01	1.90E-01	2.13E-01	1.59E-01	1.19E-01	1.91E-01	1.44E-01	2.56E+00	6.82E-02	1.65E-01	1.14E-01	2.84E-01	1.38E-01	10
F3	-1.05E+02	-9.95E+01	-1.06E+02	-1.06E+02	-1.06E+02	-1.06E+02	-1.05E+02	-1.07E+02	-7.96E+01	-1.06E+02	-1.07E+02	-1.05E+02	-1.02E+02	-1.06E+02	8
F4	1.95E-07	8.07E-08	7.58E-08	6.53E-08	2.25E-07	3.52E-08	9.67E-08	2.86E-01	4.38E+00	3.04E-08	9.03E-08	8.59E-03	5.46E-07	2.33E-08	14
F5	1.00E+00	1.56E+00	4.75E-01	7.55E-01	7.76E-01	5.95E-01	1.09E+00	1.64E+00	1.46E+01	5.85E-01	6.70E-01	9.47E-01	1.33E+00	1.10E+00	3
F6	3.33E+01	1.38E+02	1.09E+02	4.93E+01	3.61E+01	7.46E+01	3.67E+01	1.42E+01	1.16E+02	6.64E+01	2.95E+01	4.05E+01	1.55E+02	6.37E+01	8

Çizelge 4.24 Önerilen KKS Algoritmasıyla (Algoritma 6) elde edilen sonuçların ortalamaları

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	En İyi Harita
F1	2.23E-06	2.46E-08	9.73E-09	1.39E-08	6.17E-09	1.77E-08	1.38E-08	1.70E-06	1.22E+01	5.98E-09	5.32E-09	4.69E-08	3.56E-01	2.78E-08	11
F2	3.20E-02	2.88E-01	1.78E-01	2.11E-01	2.21E-01	1.70E-01	2.25E-01	1.35E-01	2.42E+00	1.31E-01	2.09E-01	1.58E-01	2.45E-01	1.66E-01	1
F3	-1.07E+02	-9.99E+01	-1.06E+02	-1.06E+02	-1.05E+02	-1.06E+02	-1.04E+02	-1.07E+02	-6.81E+01	-1.06E+02	-1.04E+02	-1.05E+02	-1.02E+02	-1.05E+02	1
F4	4.02E-10	4.05E-09	8.40E-10	3.53E-09	3.12E-09	3.39E-10	1.36E-09	1.07E-09	1.62E+00	7.47E-10	2.38E-09	3.21E-10	3.76E-09	1.17E-09	12
F5	1.60E-01	1.75E+00	6.63E-01	6.76E-01	1.09E+00	5.67E-01	9.99E-01	1.68E+00	1.48E+01	5.14E-01	7.55E-01	1.20E+00	1.40E+00	9.96E-01	1
F6	2.32E+01	1.26E+02	9.55E+01	5.39E+01	3.63E+01	5.78E+01	3.93E+01	1.15E+01	1.27E+02	6.68E+01	3.88E+01	4.11E+01	1.66E+02	7.29E+01	8

Çizelge 4.25 PSO ile elde edilen sonuçların ortalamaları

	Değer
F1	7.80E-03
F2	7.63E-03
F3	-1.07E+02
F4	8.37E-06
F5	1.67E-02
F6	4.82E+01

Çizelge 4.26 GA ile elde edilen sonuçların ortalamaları

	Değer
F1	1.82E+00
F2	2.83E-01
F3	-8.68E+01
F4	6.13E+00
F5	5.79E+00
F6	1.85E+02

Çizelge 4.27 DGA ile elde edilen sonuçların ortalamaları

	Değer
F1	2.05E-04
F2	8.61E-09
F3	-1.07E+02
F4	1.13E-08
F5	3.56E-02
F6	1.36E+01

4.1.4. En İyiler

Çizelge 4.28 Mevcut KS yönteminden (Algoritma 1) elde edilen sonuçların en iyileri

	Değer
F1	1.70E-11
F2	3.56E-13
F3	-1.07E+02
F4	4.56E-14
F5	2.23E-12
F6	1.10E-10

Çizelge 4.29 KKS Wang algoritmasıyla (Algoritma 2) elde edilen sonuçların en iyileri

	Değer
F1	1.13E-09
F2	2.31E-12
F3	-1.07E+02
F4	7.21E-13
F5	1.12E-12
F6	8.56E-10

Çizelge 4.30 Önerilen KKS algoritmasıyla (Algoritma 3) elde edilen sonuçların en iyileri

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	En İyi Harita
F1	6.01E-10	1.05E-09	2.13E-09	1.41E-09	5.18E-10	4.61E-10	4.81E-10	2.89E-08	1.99E-09	3.96E-10	6.82E-10	6.29E-09	1.17E-09	2.44E-10	14
F2	8.14E-11	4.30E-14	1.36E-11	4.63E-12	7.36E-13	6.61E-13	5.85E-12	9.12E-13	1.35E-13	2.06E-12	3.99E-12	2.92E-12	1.24E-13	3.22E-11	2
F3	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	3
F4	2.44E-13	9.76E-13	3.88E-13	3.41E-13	1.53E-12	2.34E-13	5.13E-13	8.57E-14	8.55E-13	1.16E-13	2.58E-12	4.86E-14	3.57E-13	1.85E-13	12
F5	2.35E-11	8.46E-13	4.40E-11	1.11E-12	5.79E-10	2.16E-10	2.58E-10	2.11E-10	9.95E-01	4.02E-11	4.72E-10	1.81E-09	8.42E-11	5.79E-11	2
F6	1.01E-10	7.78E-09	8.91E-10	6.00E-11	1.96E-10	1.10E-10	7.15E-10	1.14E-12	2.16E-11	5.55E-10	3.29E-10	1.82E-12	2.47E-10	1.30E-11	8

Çizelge 4.31 Önerilen KKS algoritmasıyla (Algoritma 4) elde edilen sonuçların en iyileri

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	En İyi Harita
F1	3.16E-10	1.39E-09	6.70E-10	1.54E-09	4.93E-11	3.41E-10	2.68E-10	4.32E-10	2.20E-10	4.03E-10	3.45E-11	5.91E-10	5.06E-10	2.16E-10	11
F2	2.94E-11	2.12E-11	4.05E-12	1.01E-12	1.06E-11	1.04E-12	3.93E-12	2.61E-12	1.34E-12	1.34E-11	3.45E-13	1.44E-12	1.63E-12	3.41E-12	11
F3	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	4
F4	1.01E-13	1.48E-13	8.47E-14	6.21E-13	6.78E-13	1.35E-13	1.51E-13	7.17E-14	4.61E-13	1.41E-12	8.88E-14	5.42E-13	2.49E-13	3.21E-15	14
F5	5.01E-11	8.19E-12	8.55E-11	7.22E-12	9.99E-12	2.70E-10	2.16E-12	6.08E-13	3.15E-10	1.48E-11	2.04E-11	3.19E-11	6.04E-12	4.37E-11	8
F6	1.50E-09	1.00E-09	5.00E-12	1.24E-09	7.62E-11	3.52E-09	5.00E-10	1.65E-09	1.05E-10	1.82E-10	1.09E-09	4.21E-09	3.20E-10	4.79E-09	3

Çizelge 4.32 Önerilen KKS algoritmasıyla (Algoritma 5) elde edilen sonuçların en iyileri

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	En İyi Harita
F1	9.90E-10	5.16E-10	1.07E-10	1.99E-10	9.90E-10	1.02E-09	7.64E-10	2.85E-08	1.27E-09	4.64E-09	9.19E-10	3.87E-10	7.60E-10	4.08E-10	3
F2	9.41E-13	1.23E-12	1.13E-13	3.94E-13	2.50E-12	2.18E-12	2.02E-10	1.88E-12	2.16E-14	7.98E-13	5.31E-12	1.67E-12	4.04E-13	7.26E-12	9
F3	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	2
F4	1.64E-12	7.04E-13	6.93E-13	2.12E-12	9.24E-12	2.66E-17	1.98E-12	5.88E-15	3.50E-14	1.57E-14	3.11E-13	7.40E-14	6.32E-14	1.07E-12	6
F5	1.50E-12	4.53E-12	1.06E-11	2.45E-11	1.15E-10	6.29E-11	1.16E-09	1.94E-10	1.07E-06	3.69E-12	7.30E-11	2.21E-10	2.94E-09	3.25E-09	1
F6	5.15E-10	6.27E-09	5.67E-10	3.41E-11	1.14E-10	6.76E-11	8.28E-10	9.32E-12	3.64E-12	1.06E-09	7.54E-10	2.34E-10	5.68E-10	5.67E-11	9

Çizelge 4.33 Önerilen KKS algoritmasıyla (Algoritma 6) elde edilen sonuçların en iyileri

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	En İyi Harita
F1	1.05E-10	2.87E-11	1.46E-11	9.70E-11	5.87E-11	4.99E-11	2.91E-11	2.31E-11	6.48E-11	1.27E-11	2.60E-11	5.94E-11	2.95E-11	4.56E-11	10
F2	1.73E-13	6.34E-14	6.94E-14	1.92E-13	6.95E-14	5.29E-14	6.25E-13	1.91E-16	4.34E-14	1.48E-13	3.62E-14	3.03E-14	8.21E-17	1.28E-14	13
F3	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	-1.07E+02	1
F4	7.89E-15	2.35E-15	2.35E-15	5.21E-17	1.91E-16	6.12E-16	3.87E-15	5.82E-15	1.49E-14	4.44E-15	9.13E-16	2.03E-15	5.74E-17	7.95E-16	4
F5	2.44E-12	5.97E-13	5.01E-13	4.26E-14	2.42E-13	7.11E-15	1.10E-13	4.48E-13	2.13E-14	2.00E-12	7.96E-13	2.98E-12	8.53E-14	1.97E-12	6
F6	6.82E-13	1.73E-11	7.00E-11	8.00E-11	4.77E-12	1.02E-12	2.27E-13	1.14E-13	7.96E-13	2.27E-13	3.41E-13	5.00E-12	1.13E-11	9.00E-11	8

Çizelge 4.34 PSO ile elde edilen sonuçların en iyileri

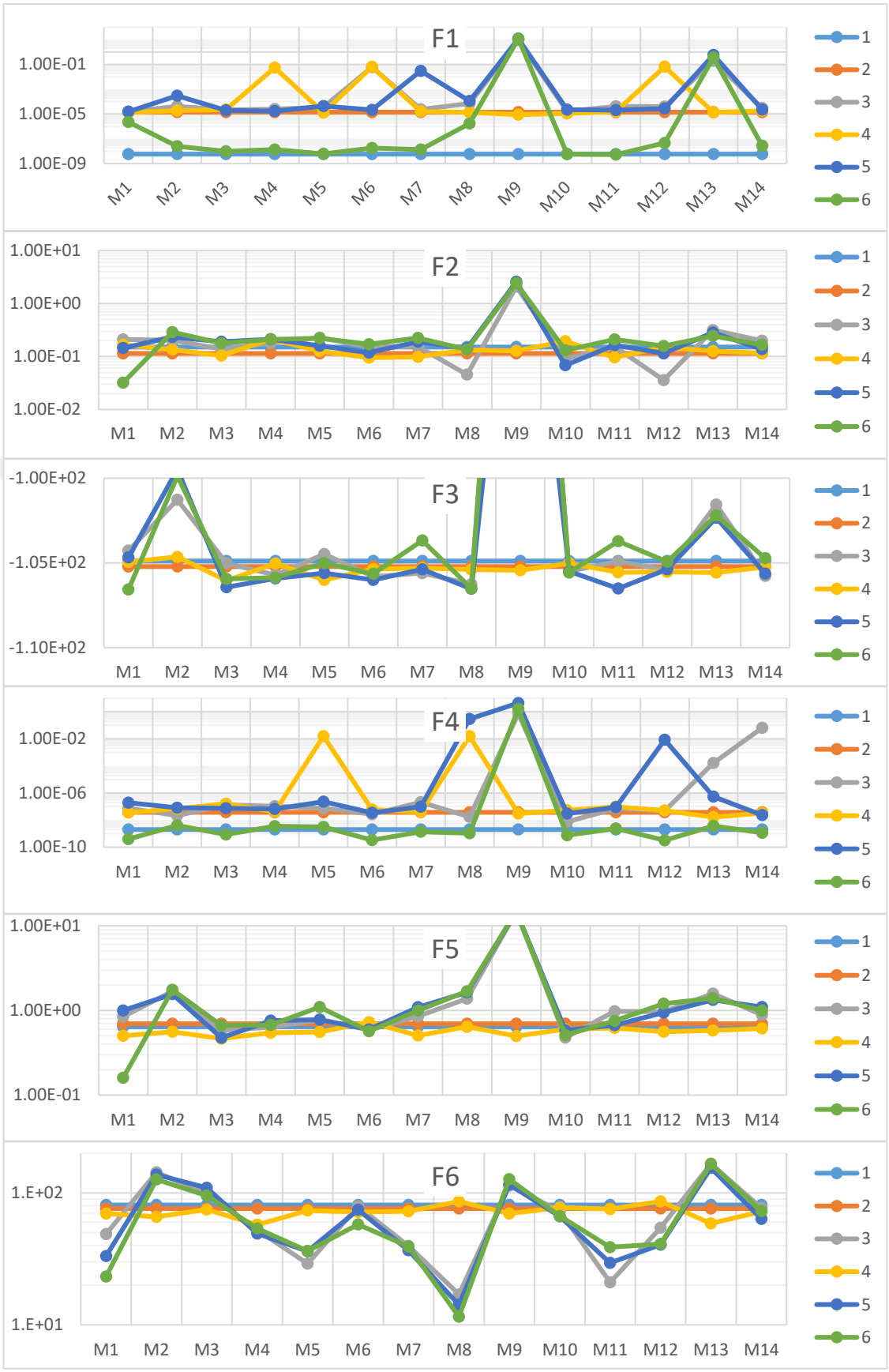
	Değer
F1	2.01E-04
F2	7.68E-10
F3	-1.07E+02
F4	8.88E-09
F5	1.73E-08
F6	3.55E-07

Çizelge 4.35 GA ile elde edilen sonuçların en iyileri

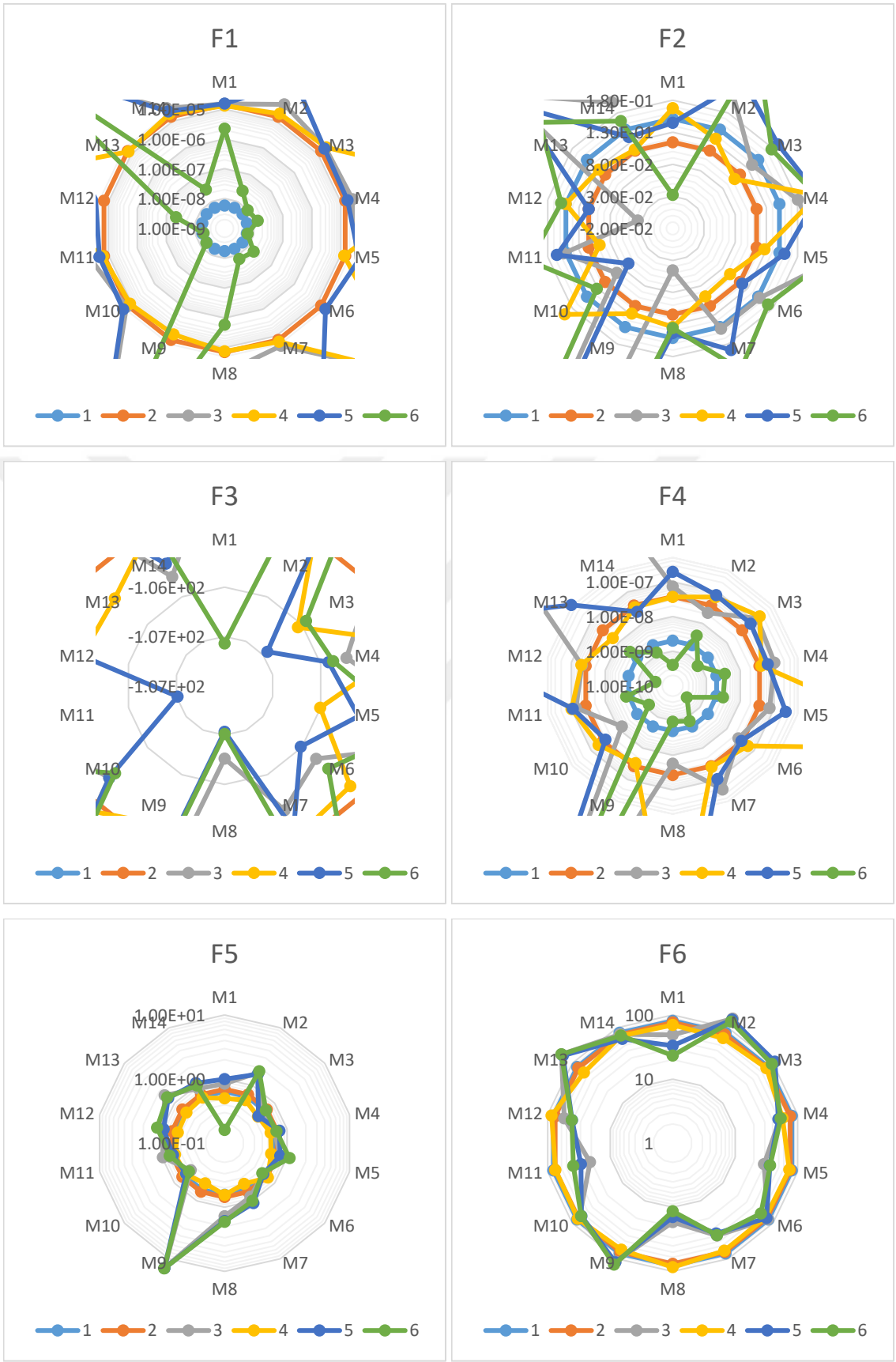
	Değer
F1	9.60E-02
F2	1.40E-02
F3	-1.07E+02
F4	8.66E-02
F5	5.97E-01
F6	1.00E+01

Çizelge 4.36 DGA ile elde edilen sonuçların en iyileri

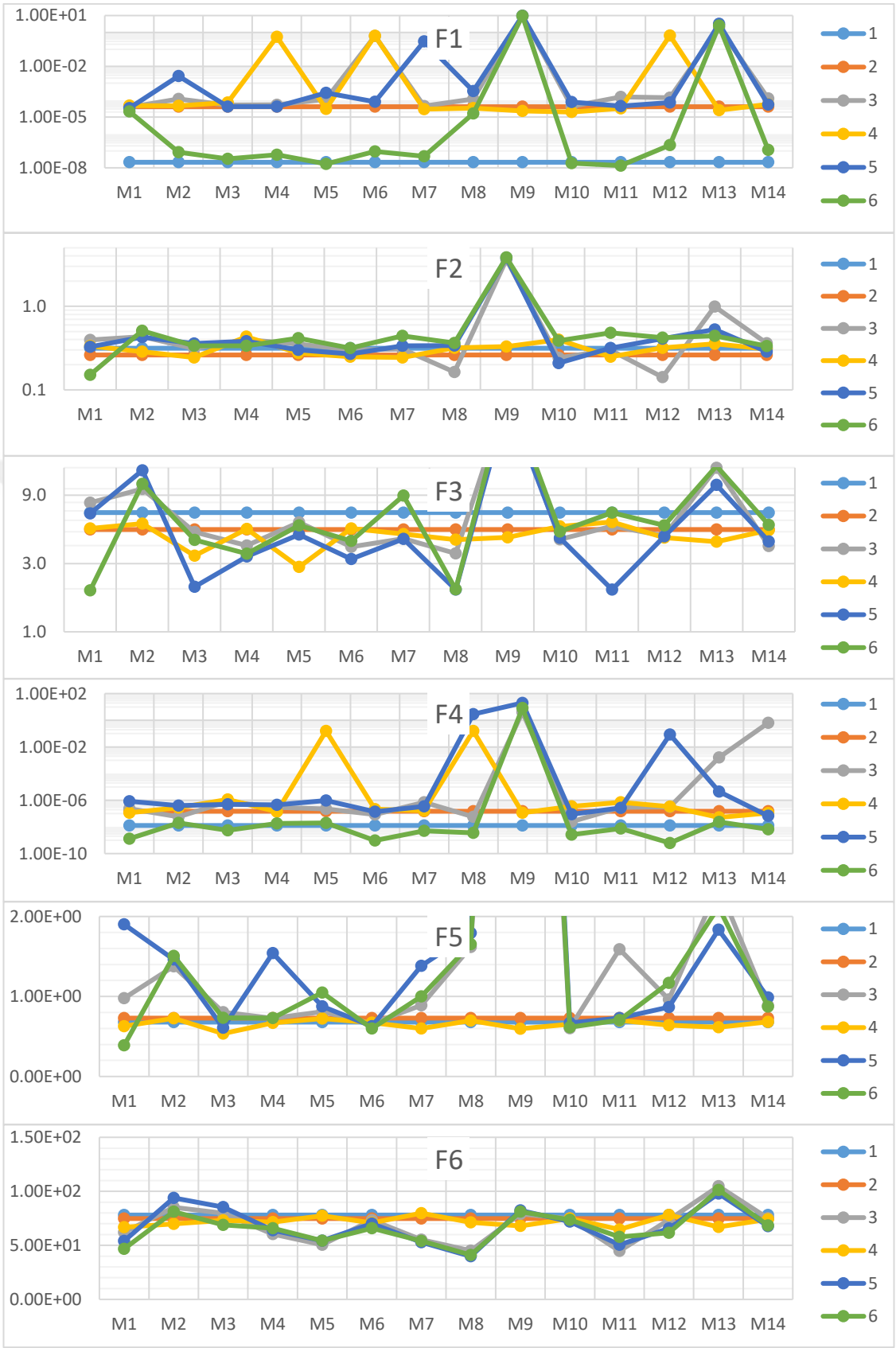
	Değer
F1	8.07E-06
F2	1.20E-11
F3	-1.07E+02
F4	3.00E-12
F5	1.90E-07
F6	9.52E-08



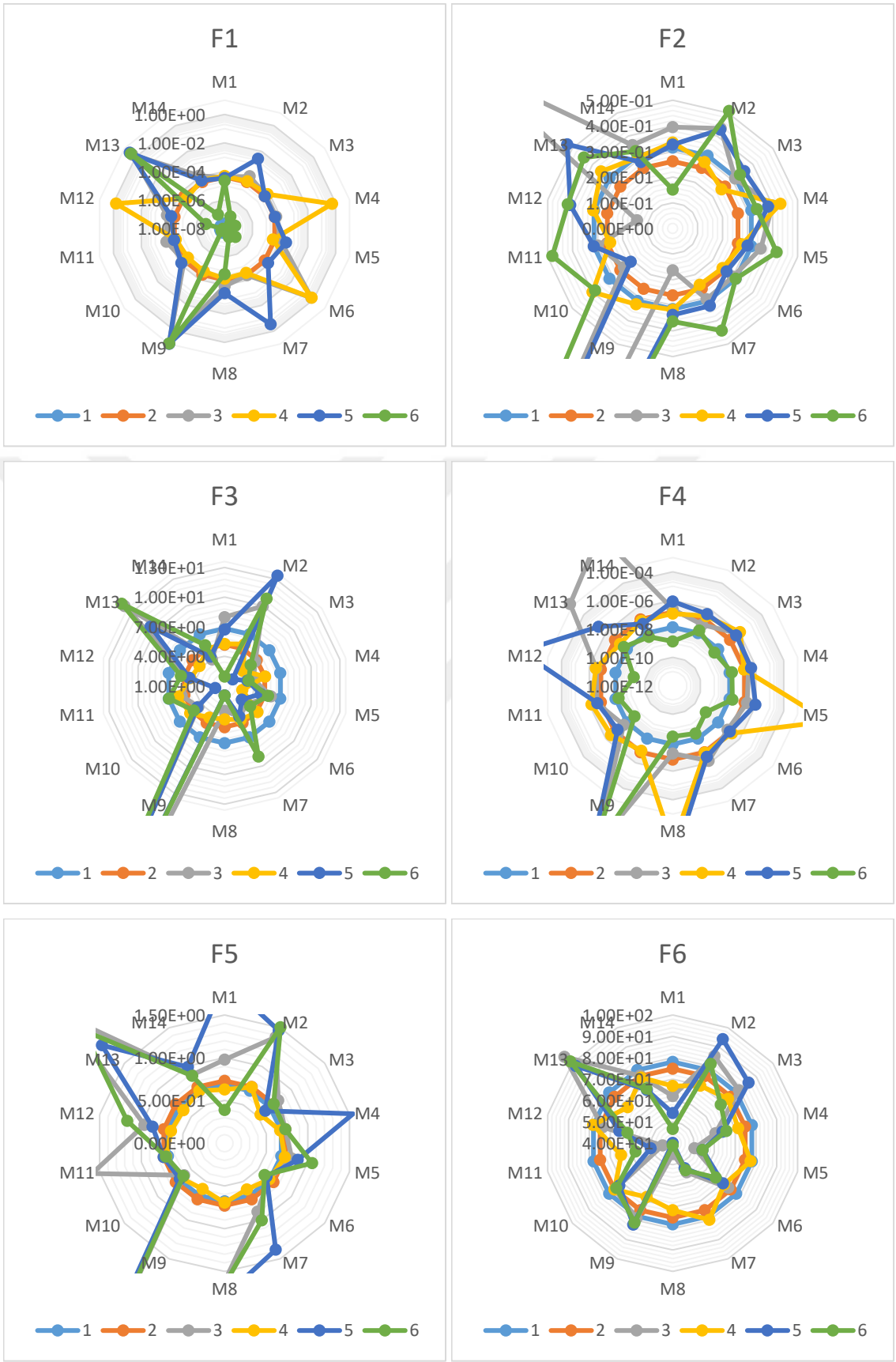
Şekil 4.1 Her bir algoritmadan elde edilen ortalamalar (çizgi grafik)



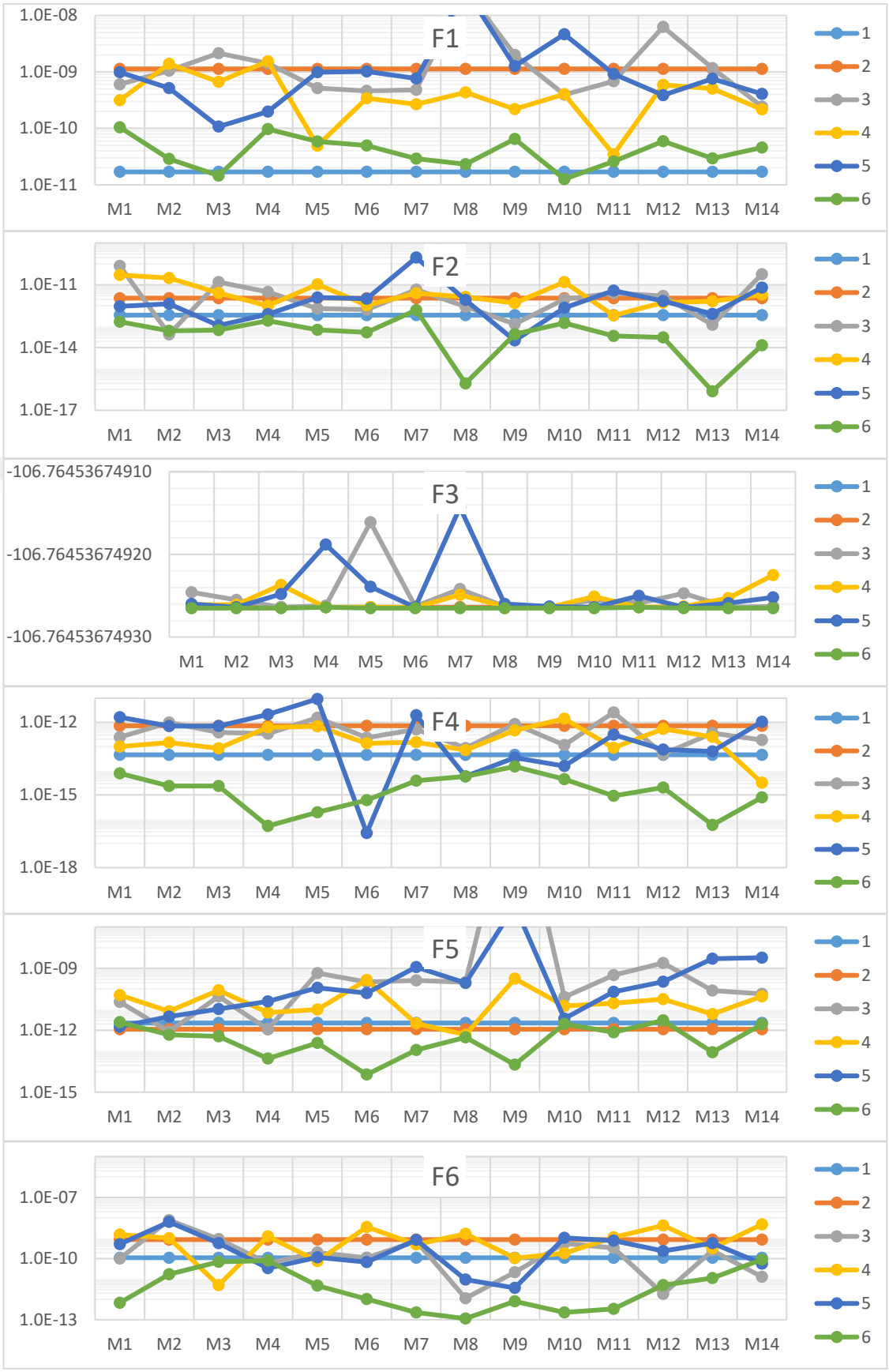
Şekil 4.2 Her bir algoritmadan elde edilen ortalamalar (Radar grafik)



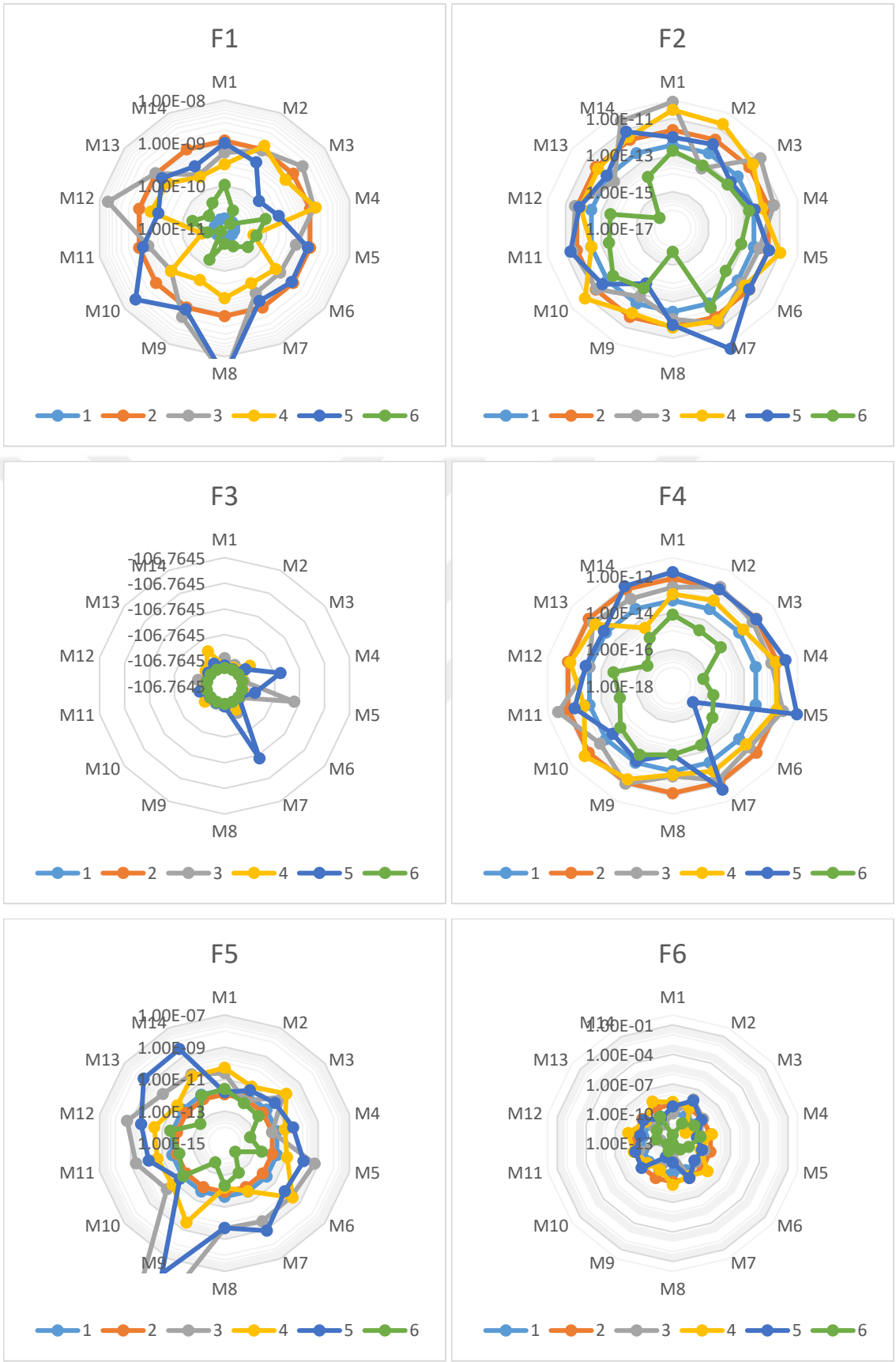
Şekil 4.3 Her bir algoritmadan elde edilen standart sapmalar (çizgi grafik)



Şekil 4.4 Her bir algoritmadan elde edilen standart sapmalar (radar grafik)



Şekil 4.5 Her bir algoritmadan elde edilen en iyiler (çizgi grafik)



Şekil 4.6 Her bir algoritmadan elde edilen en iyiler (radar grafik)

4.2. Önerilen KKS Algoritmasının Seçimi

Elde edilen sonuçların yer aldığı Şekil 4.1 - Şekil 4.6 incelenerek; ortalama, standart sapma ve en iyi değerlere göre her bir test fonksiyonu için en iyi sonuca ulaşan algoritma ve harita fonksiyonları, sırasıyla Çizelge 4.37 ve Çizelge 4.38 ‘de sunulmuştur.

Çizelge 4.37 En iyi sonuçlara ulaşan algoritma

	Standart Sapma	Ortalama	En iyi
F1	Alg.6	Alg.6	Alg.6
F2	Alg.6	Alg.6	Alg.6
F3	Alg.6	Alg.6	Alg.6
F4	Alg.6	Alg.6	Alg.5
F5	Alg.6	Alg.6	Alg.6
F6	Alg.6	Alg.6	Alg.6

Çizelge 4.38 En iyi sonuçlara ulaşan harita fonksiyonu

	Standart Sapma	Ortalama	En iyi
F1	M11	M11	M10
F2	M1	M1	M13
F3	M1	M1	M1
F4	M12	M12	M4
F5	M1	M1	M6
F6	M8	M8	M8

Çizelge 4.37’ deki duruma göre, Algoritma 6 ‘nın hem önerilen hem de literatürdeki mevcut KS ve KKS algoritmalarına göre çok daha başarılı olduğu açıkça görülmektedir. Bu başarı önerilen algoritmanın, literatürdeki hem orijinal hem de kaotik kril sürüsü optimizasyon yöntemine göre olan üstünlüğünü göstermektedir.

Bununla birlikte harita fonksiyonlarının genel performansını karşılaştırabilmek için ilk 4‘e kaç kez girdikleri sayılmıştır. Sayım sonrasında elde edilen sonuçlar, Şekil 4.7’ de gösterilmiştir.

Harita	Alg. 3	Alg. 4	Alg. 5	Alg. 6
1	1	2	2	5
2	1	0	0	0
3	2	3	3	2
4	2	0	1	1
5	1	1	1	2
6	3	1	4	4
7	2	3	1	1
8	4	1	2	2
9	0	4	0	0
10	4	1	3	3
11	2	2	5	2
12	1	0	0	1
13	0	5	0	0
14	1	1	2	1

a. Standart Sapma

Harita	Alg. 3	Alg. 4	Alg. 5	Alg. 6
1	1	3	2	5
2	1	1	0	0
3	3	3	3	3
4	2	1	2	1
5	1	2	1	2
6	3	1	4	2
7	2	2	0	0
8	4	0	2	3
9	0	4	0	0
10	4	1	3	4
11	1	2	4	2
12	1	0	1	2
13	0	3	0	0
14	1	1	2	0

b. Ortalama

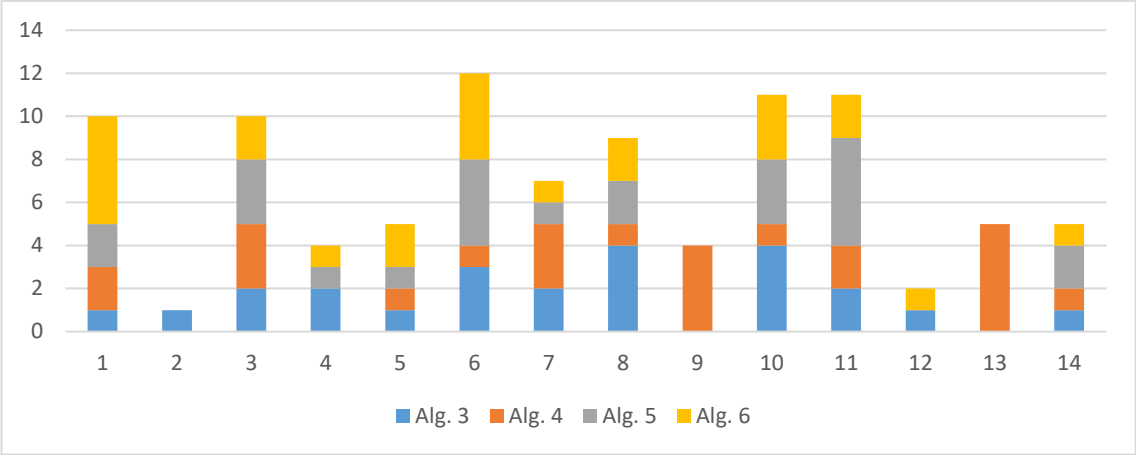
Harita	Alg. 3	Alg. 4	Alg. 5	Alg. 6
1	1	0	1	4
2	2	0	5	0
3	3	2	3	1
4	1	6	3	2
5	0	2	0	1
6	3	1	1	2
7	1	1	0	2
8	2	2	2	3
9	2	3	3	1
10	3	1	2	1
11	0	3	0	2
12	2	0	1	1
13	1	1	1	3
14	3	2	2	1

c. En İyi

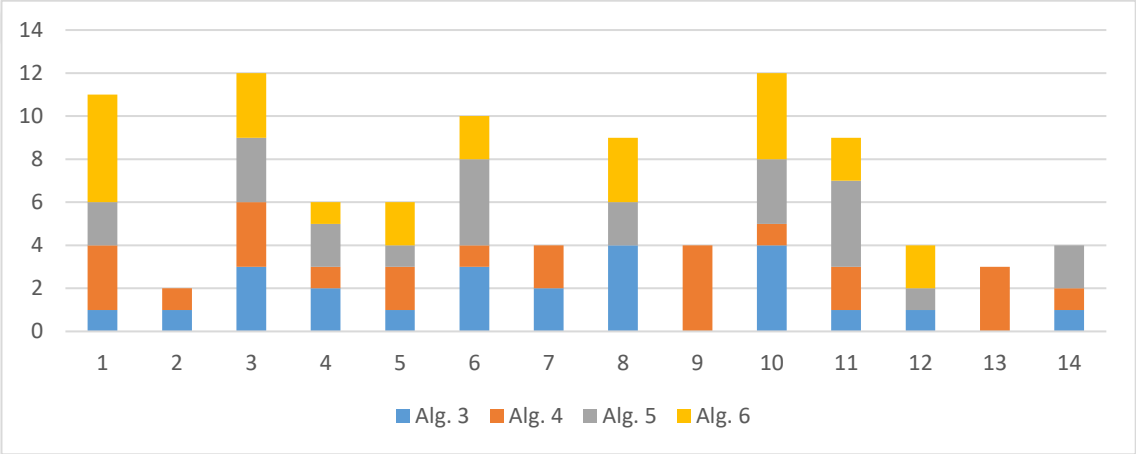
Şekil 4.7 Her bir algoritmadaki Harita fonksiyonlarının ilk 4 'e giriş sayıları

Çizelge 4.38 ‘e göre standart sapma ve ortalama değer için 1 nolu harita fonksiyonun (Chebyshev) daha başarılı olduğu açıkça görülmektedir. Buna karşılık, en iyi değerlere bakıldığında, her bir fonksiyonda başka bir harita fonksiyonun daha başarılı olduğu görülmüştür.

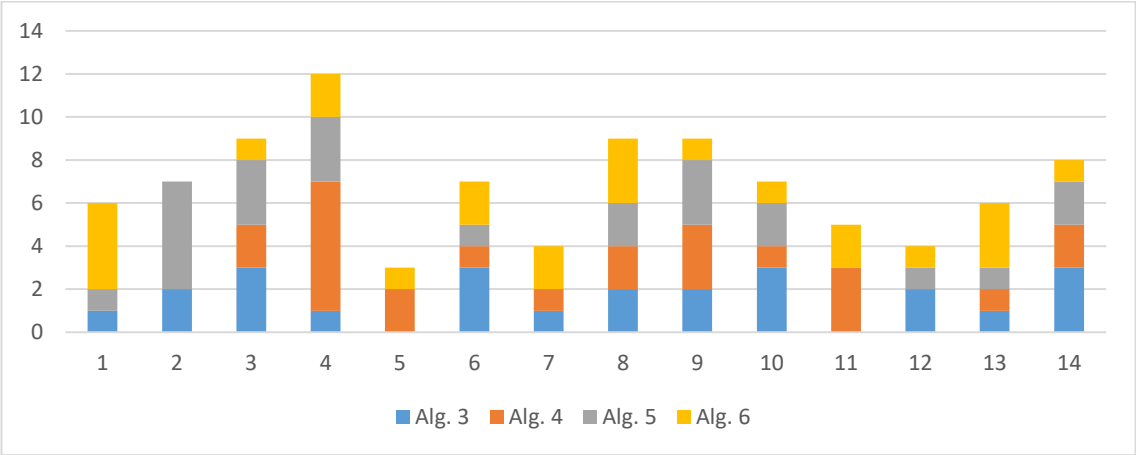
Algoritmaların ve harita fonksiyonlarının performanslarının daha iyi incelenebilmeleri için, her bir harita fonksiyonun her bir algoritmadaki ilk 4 ‘e giriş sayılarının, yığın ve çubuk grafikleri oluşturulmuştur (Şekil 4.8 - Şekil 4.13). Hangi algoritmanın hangi harita fonksiyonu ile daha başarılı oldukları bu grafiklerden okunabilir.



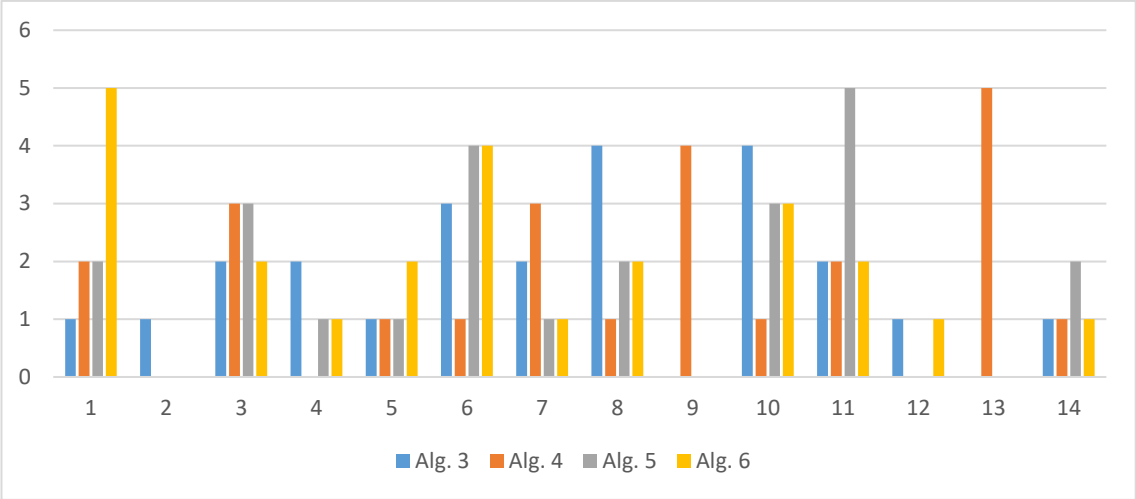
Şekil 4.8 Harita Fonksiyonlarının Standart Sapmada İlk 4 'e Giriş Toplamları



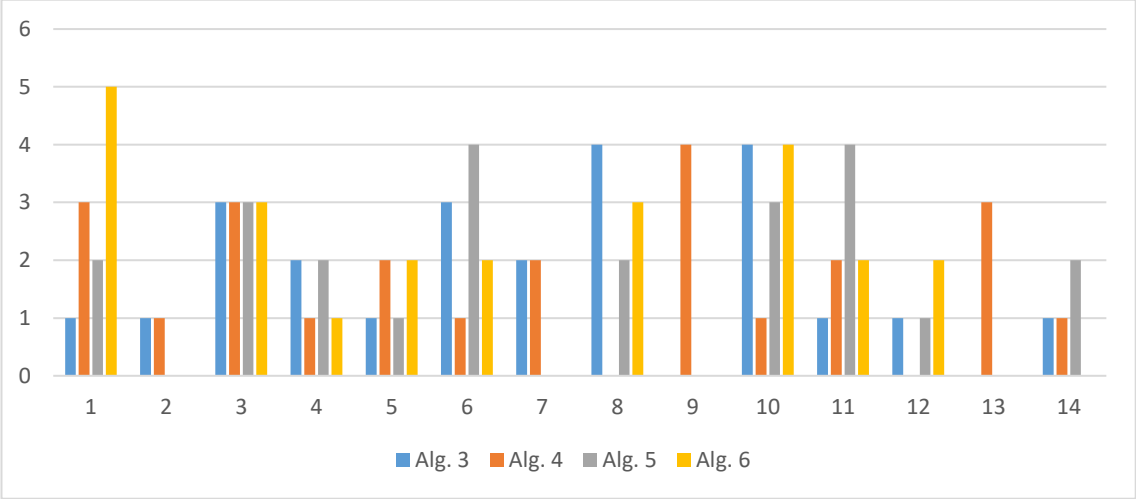
Şekil 4.9 Harita Fonksiyonlarının Ortalamalarda İlk 4 'e Giriş Toplamları



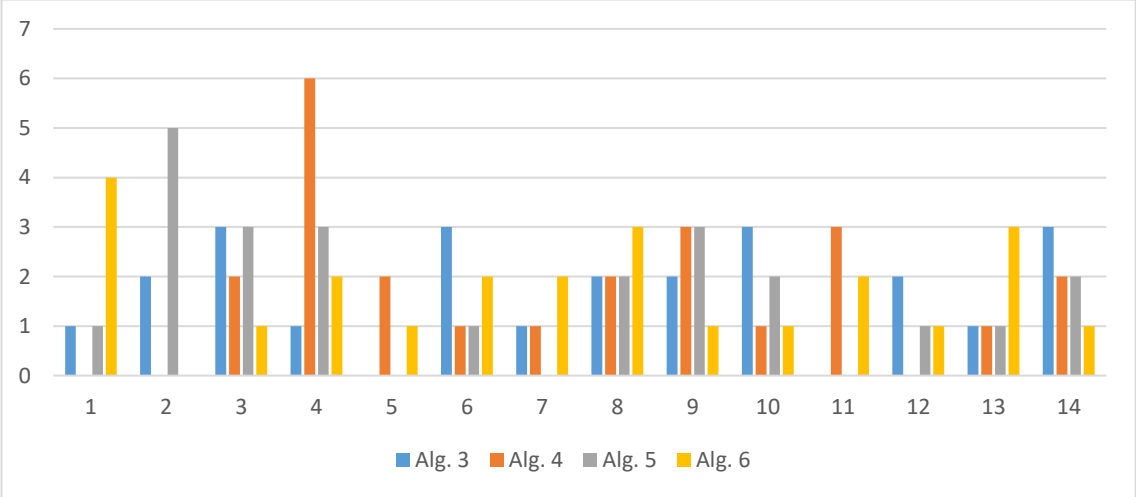
Şekil 4.10 Harita Fonksiyonlarının En İyilerde İlk 4 'e Giriş Toplamları



Şekil 4.11 Harita Fonksiyonlarının Standart Sapmada İlk 4 'e Giriş Sayıları



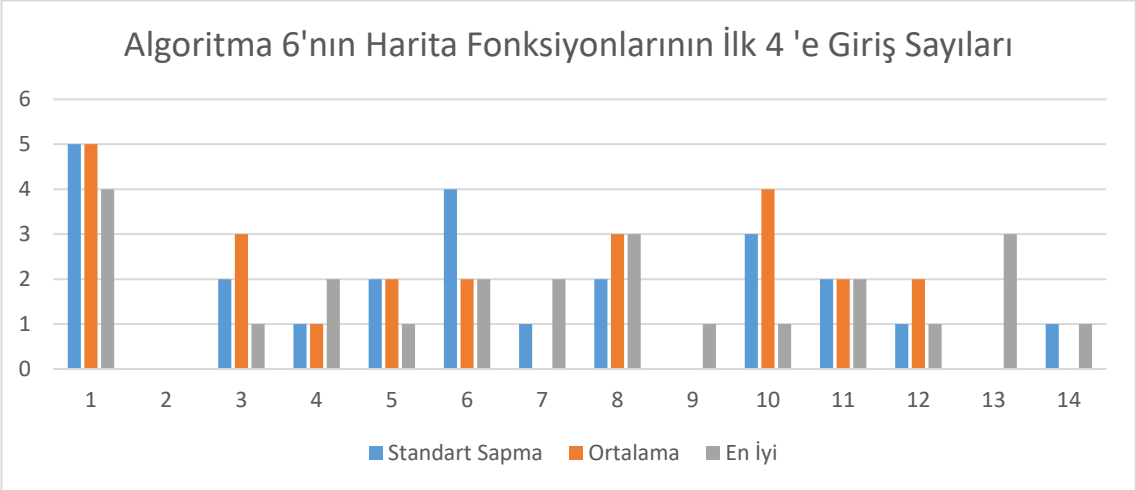
Şekil 4.12 Harita Fonksiyonlarının Ortalamalarda İlk 4 'e Giriş Sayıları



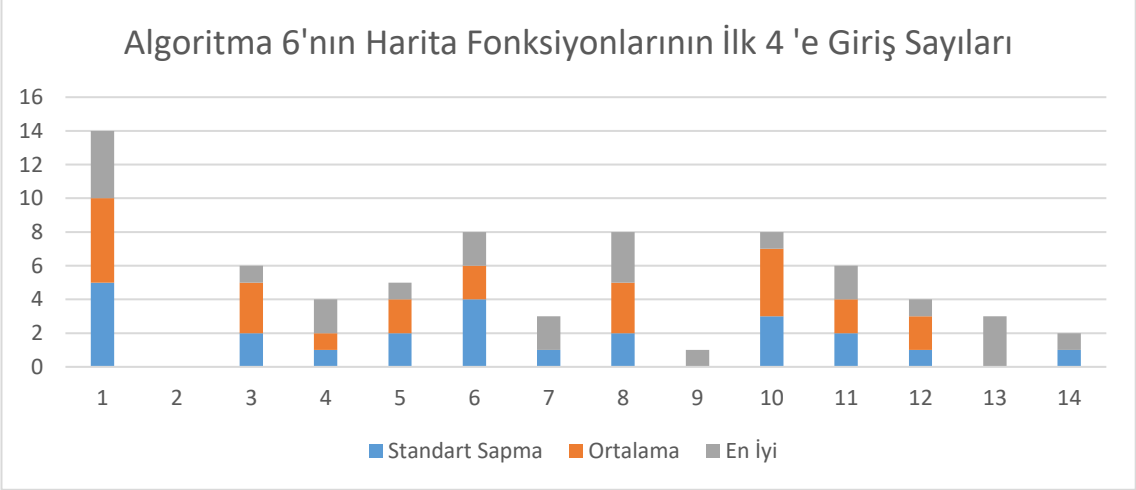
Şekil 4.13 Harita Fonksiyonlarının En İyilerde İlk 4 'e Giriş Sayıları

Bununla birlikte en iyi sonucu veren Algoritma 6'nın her bir harita fonksiyonu ile ilk 4'e giriş sayıları, Şekil 4.14 ve Şekil 4.15 'de çubuk ve yığın grafik olarak incelenmiştir. Şekil 4.15'de görüldüğü üzere, standart sapma, ortalama ve en iyi değerlere göre ilk 4'e giriş sayıları üst üste bindirildiğinde 1 nolu Chebyshev harita fonksiyonun toplam başarısının diğerlerinden net bir şekilde ayrıldığı görülmektedir. Bu algoritmadaki harita fonksiyonlarının başarı sıralaması aşağıdaki gibidir.

1. Chebyshev
2. Cubic
3. Piecewise
4. Gauss/mouse ve Singer



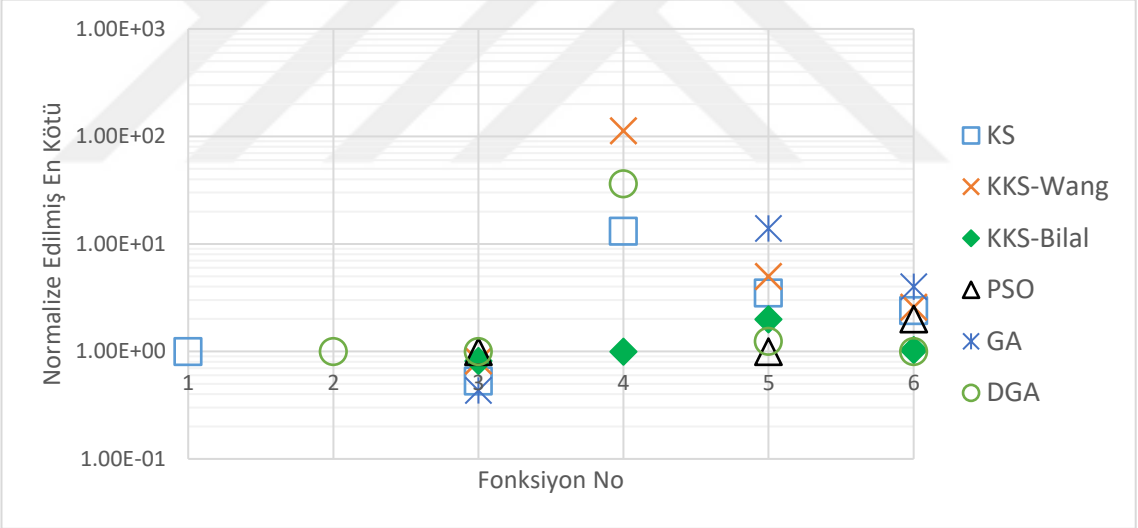
Şekil 4.14 Algoritma 6'nın Harita Fonksiyonlarının İlk 4 'e Giriş Sayıları (Çubuk grafik)



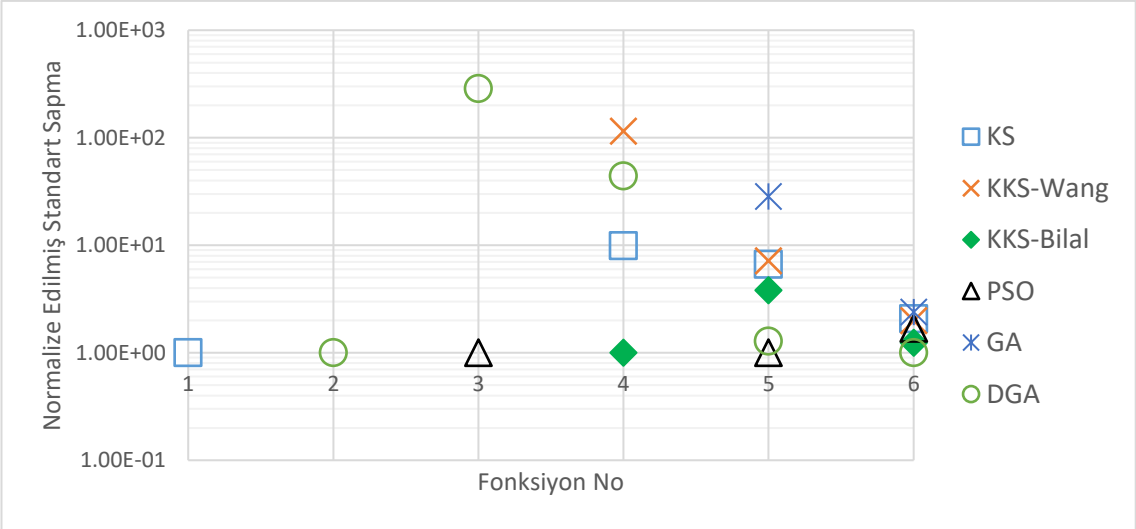
Şekil 4.15 Algoritma 6'nın Harita Fonksiyonlarının İlk 4 'e Giriş Sayıları (Yığın grafik)

Buradan elde edilen sonuçlara göre atalet ağırlığı (w_f) in kaotik olmadığı, başlangıç pozisyonlarının ve en iyi bireyin etkisinin hesaplandığı adımdaki C^{eniyi} katsayısının hesabının kaotik olduğu algoritma (Algoritma 6) en iyi sonucu veren yöntem olarak seçilmiştir. Kaotik alanlarda kullanılacak harita fonksiyonu ise Chebyshev fonksiyonu olmalıdır.

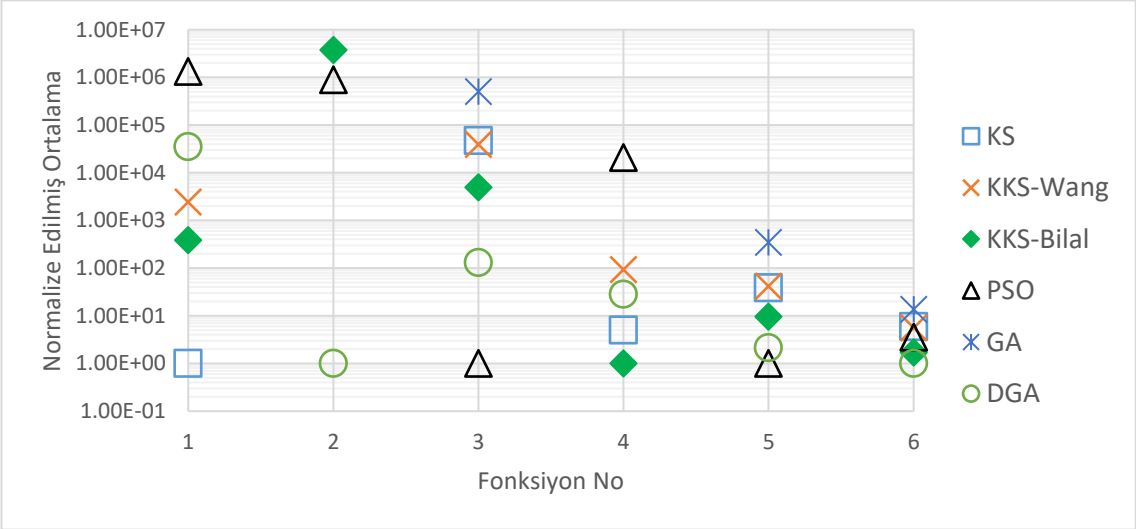
Bu algoritma ile elde edilen sonuçların mevcut kaotik olmayan ve olan KS optimizasyonları ile olan karşılaştırmalarına ek olarak, PSO, GA ve DGA ile olan karşılaştırmaları da Şekil 4.16 - Şekil 4.19 'da gösterilmiştir. Burada yer alan veriler; yapılan çözümlerde en iyi sonucu veren algoritmanın değerine göre normalize edilmiş, en kötü, standart sapma, ortalama ve en iyi değerleridir. Bu grafiklerde, 1'e en yakın sonuçları veren algoritmalar en başarılı algoritmadır.



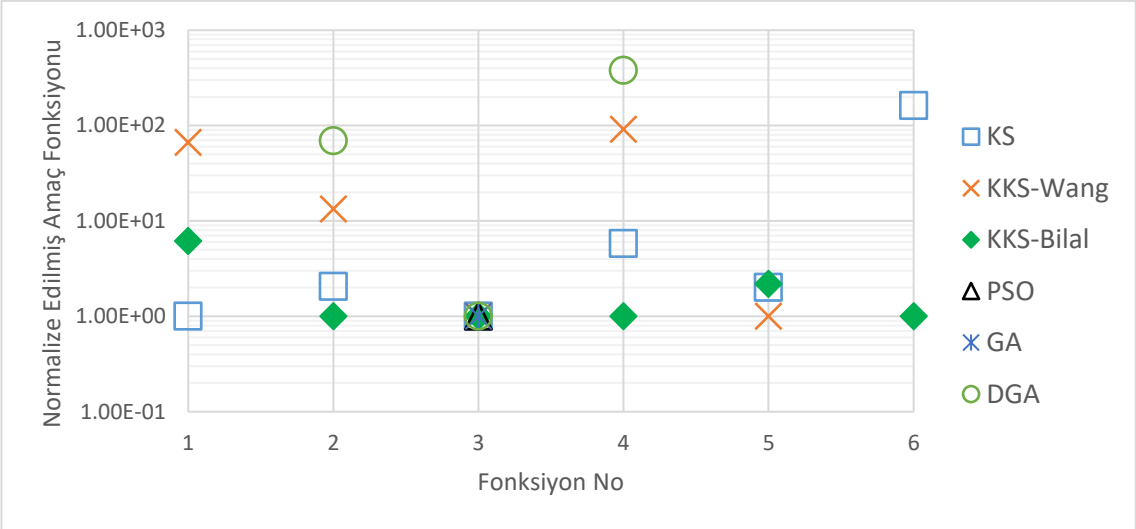
Şekil 4.16 Geliştirilen KKS 'nün Diğer Yarı Sezgisel Algoritmalar ile Karşılaştırması, Normalize Edilmiş En Kötü



Şekil 4.17 Geliştirilen KKS 'nün Diğer Yarı Sezgisel Algoritmalar ile Karşılaştırması, Normalize Edilmiş Standart Sapma



Şekil 4.18 Geliştirilen KKS 'nün Diğer Yarı Sezgisel Algoritmalar ile Karşılaştırması, Normalize Edilmiş Ortalama



Şekil 4.19 Geliştirilen KKS 'nün Diğer Yarı Sezgisel Algoritmalar ile Karşılaştırması, Normalize Edilmiş En İyi

Yukarıda yapılan değerlendirmelere göre, geliştirilen KKS optimizasyon yönteminin sadece KS optimizasyon yönteminden değil, literatürdeki diğer iyi bilinen yarı sezgisel algoritmalara göre de oldukça başarılı ve rekabetçi olduğu görülmüştür.

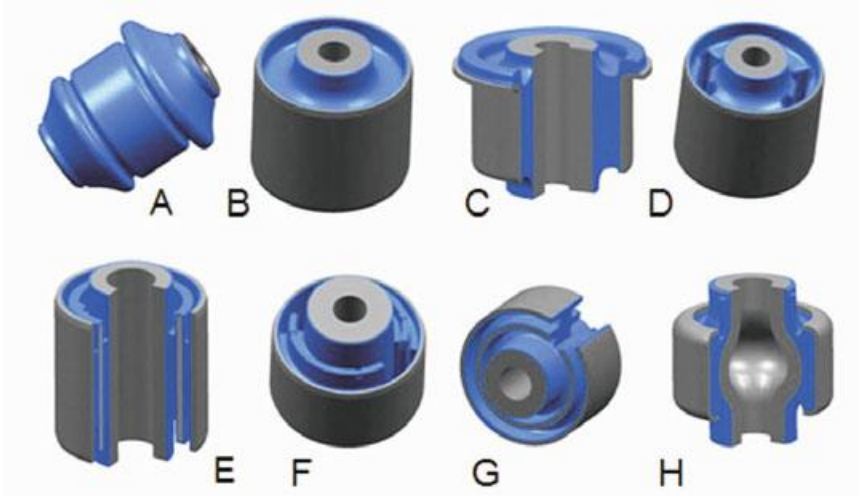
4.3. Geliştirilen KKS Yöntemi ile Kauçuk Burç Optimizasyonu

Salıncak kolları, amortisörler ve motor traversi gibi şasi ve süspansiyon bileşenlerini monte etmek için standart kauçuk burçlar kullanılır (Heißing ve Ersoy 2015). Burçlar ya da kauçuk burçlar bir çeşit titreşim sönümleyicidir. İki parçayı birbirine bağlayan kauçuk burçlar, üzerinden geçen enerjiyi sönümleyerek diğerine aktarır. Yaygın olarak araç süspansiyon sistemlerinde kullanılan kauçuk burçlar, iki parçayı birbirine bağlarken bir miktar da hareket serbestliği sağlar. Bu hareket, süspansiyon parçalarının serbestçe hareket etmesine izin verirken, örneğin büyük bir tümsek üzerinden geçerken gürültünün ve küçük titreşimlerin aracın şasisine aktarılmasını en aza indirir ("Bushing (isolator) - Wikipedia" y.y.).

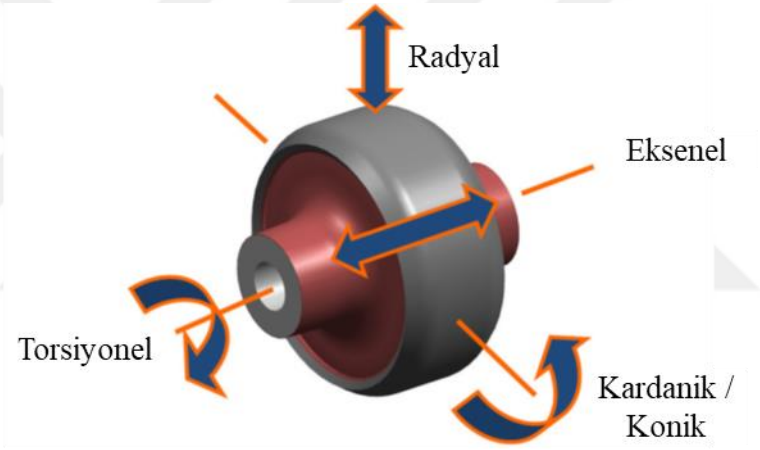
Bu işlevler için hidrolik burçlar veya kayar burçlar da kullanılabilir. Kayar burçlar, sönümlemenin yanı sıra serbest dönüşe izin vermek için kauçuk ile montaj yüzeyi arasında kayan bir yatağa sahiptir (Heißing ve Ersoy 2015).

Farklı kauçuk burç türlerinden bazıları Şekil 4.20' de görülebilir. Spesifik uygulamanın gereksinimlerine bağlı olarak çok çeşitli farklı çözümler mevcuttur. Tek borulu burç olarak da adlandırılan A tasarımı, en basit ve en ucuz çözümü temsil eder. Bu tip bir burç, radyal ön gerilim ve entegre eksenel tutma halkaları ile yerinde tutulduğu bir süspansiyon bağlantısı veya kontrol kolundaki silindirik bir deliğe bastırılmak üzere tasarlanmıştır. Tasarım B, bileşenin radyal ve eksenel direngenlik oranlarının birbirine uyacak şekilde daha doğru ayarlanmasına olanak tanır. Dış borunun çapını ve elastomer iç burcun yükseklik-genişlik oranını değiştirerek, radyal/eksenel direngenlik oranı 1 : 4 ile 1: 10 arasında bir değere getirilebilir. Tasarım D, yükleme yönüne bağlı olarak farklı radyal direngenlik oranlarına neden olan böbrek şeklinde boşluklara sahiptir. E, F ve G tasarımları, ara kovanlı burçlara örnektir. E ve F tasarımları, üretim sürecinde kalibre edilir ve gerekirse iç boru genişletilerek ayarlanır. Tasarım G üzerindeki ön yük, montaj sırasında eklenir. Tasarım H, radyal yönde yüksek bir direngenlik oranını korurken, büyük kardanik yüklere ve yer değiştirmelere dayanacak şekilde geliştirilmiş bir kauçuk burç örneğidir. Gösterilen örnekte, ağırlığı azaltmak için oyuk bilyeli bir iç boru kullanılmıştır (Heißing ve Ersoy 2015).

Elastik burçlar için performans kriterleri radyal, eksenel, torsiyonel ve kardanik yönlerdeki direngenlik değerleridir. Bu yönler Şekil 4.21 'de gösterilmiştir. Farklı yönlerdeki direngenliğini belirten bu direngenlik değerleri burcun kullanıldığı yere göre değişiklik göstermektedir. Örnek olarak bir araç süspansiyonundaki salıncak kolunda yer alan elastik burçların karakteristikleri süspansiyonun elastokinematik davranışını etkiler. Bu da komple aracın yol tutuşu ve konforu üzerinde önemli bir etkiye sahiptir. Bu nedenle elastik burçların istenilen direngenlikleri kullanılacakları yerdeki görevlerine göre belirlenir. Direngenlik hedefleri belirlendikten sonraki süreç, bu hedefleri sağlayabilen elastik burç geometrisini ve malzemesini tasarlayabilmektir.



Şekil 4.20 Titreşim sönümleyici elastik burçlar (Heißing ve Ersoy 2015).



Şekil 4.21 Elastik burç yükleme yönleri (Goossens vd. 2017)

4.3.1. Optimizasyon Probleminin Kurulması

Bu çalışmada istenilen direngenlik hedeflerini sağlayabilen elastik burç optimizasyonu gerçekleştirilmiştir. Tasarımdan istenilen hedef değerlerini bir optimizasyon problemine dönüştürmek için Radyal direngenlik, f amaç fonksiyonunda, eksenel, torsiyonel ve kardanik direngenlik değerleri ise h_i kısıt fonksiyonlarında kullanılmak üzere seçilmiştir. Bu tercih optimize edilmek istenen hedef parametresinin önemine göre farklılık gösterebilir. Bu durumda f amaç fonksiyonu şu şekilde yazılır.

$$f \ \min \quad \left|\frac{k_r^H - k_r^i}{k_r^H}\right| \tag{4.1}$$

Eşitlik kısıt fonksiyonları ise aşağıdaki şekli alır.

$$h_1 = \left|\frac{k_e^H - k_e^i}{k_e^H}\right| = 0 \tag{4.2}$$

$$h_2 = \left|\frac{k_t^H - k_t^i}{k_t^H}\right| = 0 \tag{4.3}$$

$$h_3 = \left|\frac{k_k^H - k_k^i}{k_k^H}\right| = 0 \tag{4.4}$$

Burada ;

k_r^H Hedef radyal direngenlik

k_e^H Hedef eksenel direngenlik

k_t^H Hedef torsiyonel direngenlik

k_k^H Hedef kardanik direngenlik

k_r^i i. adımda hesaplanan radyal direngenlik

k_e^i i. adımda hesaplanan eksenel direngenlik

k_t^i i. adımda hesaplanan torsiyonel direngenlik

k_k^i i. adımda hesaplanan kardanik direngenlik

Eşitlik 4.1 - 4.4 ile verilen kısıt içeren optimizasyon problemini kısıt içermeyen hale dönüştürmek gereklidir. Kısıt içeren fonksiyonları kısıt içermeyecek hale dönüştüren yöntemler genel olarak sıralı kısıtsız minimizasyon tekniği olarak adlandırılır. Bu yöntemde amaç fonksiyonu, ceza parametreleri ile cezalandırılan kısıt fonksiyonları ile birleştirilerek kompozit bir fonksiyon haline dönüşür (J. S. Arora 2017).

f amaç fonksiyonu, h eşitlik ve g eşitsizlik kısıt fonksiyonları için optimizasyon problemi aşağıdaki şekilde yazılır.

$$f = f(x) \tag{4.5}$$

$$h_i(x) = 0; i = 1\ 'den\ p'ye \tag{4.6}$$

$$g_i(x) \leq 0; i = 1\ 'den\ m'ye \tag{4.7}$$

Burada p eşitlik kısıt sayısını, m ise eşitsizlik kısıt sayısını belirtmektedir. Yukarıdaki amaç ve kısıt fonksiyonları birleştirildiğinde aşağıdaki transformasyon fonksiyonu haline gelir.

$$\Phi(x, r) = f(x) + P(h(x), g(x), r) \tag{4.8}$$

Burada r, ceza parametresi vektörüdür. P ise kısıt fonksiyonlarına ceza verme eylemi r tarafından kontrol edilen gerçek değerli bir fonksiyondur. P fonksiyonu, kullanılan dönüşüm yöntemlerine göre farklı haller alabilmektedir. Bu çalışmada tercih edilen ceza fonksiyonu yöntemine göre P fonksiyonu aşağıdaki hali alır.

$$\begin{aligned} P(h(x), g(x), r) &= r\left\{\sum_{i=1}^{p}[h_i(x)]^2 + \sum_{i=1}^{m}[g_i^+(x)]^2\right\};\ g_i^+(x) \\ &= max\big(0, g_i(x)\big) \end{aligned} \tag{4.9}$$

r ceza parametresi 0 ‘dan büyük skalar bir değerdir. Bu yönteme göre belirlenen bir r değeri ile optimizasyon gerçekleştirilir ve elde edilen çözümün optimum olup olmadığı kontrol edilir. Eğer değilse r parametresi küçültülerek optimizasyon tekrar edilir. Bu işlem optimum nokta elde edilinceye kadar tekrar edilir(J. S. Arora 2017).

Bu uygulamada ceza parametresi 0.3 olarak alınmış ve optimizasyon tek bir çözüm sonrasında sonlandırılmıştır. Yani küçültülmüş yeni bir r değeri ile tekrar edilmemiştir. Değiştirilmeme se sebepleri şunlardır;

- Kullanılan KKS optimizasyon yöntemi yarı sezgisel bir algoritma olması nedeniyle çok sayıda deney ve iterasyona ihtiyaç duymaktadır.

- Amaç fonksiyonu sonlu elemanlar yöntemi kullanılarak hesaplanacağı için her bir bireye ait çözüm süresi uzundur.
- Bununla birlikte elde edilen çözüm, sezgisel algoritmaların doğası gereği lokal optimuma takılıp kalmak yerine global optimuma ulaşabilmektedirler.

Ek olarak her bir kısıt fonksiyonun önem derecelerini kontrol edebilmek için skalar ağırlık fonksiyonu yöntemi kullanılmıştır. Buna göre ;

$$h(x) = \sum_{i=1}^{m} w_i h_i(x) \quad , \quad \sum_{i=1}^{m} w_i = 1 \tag{4.10}$$

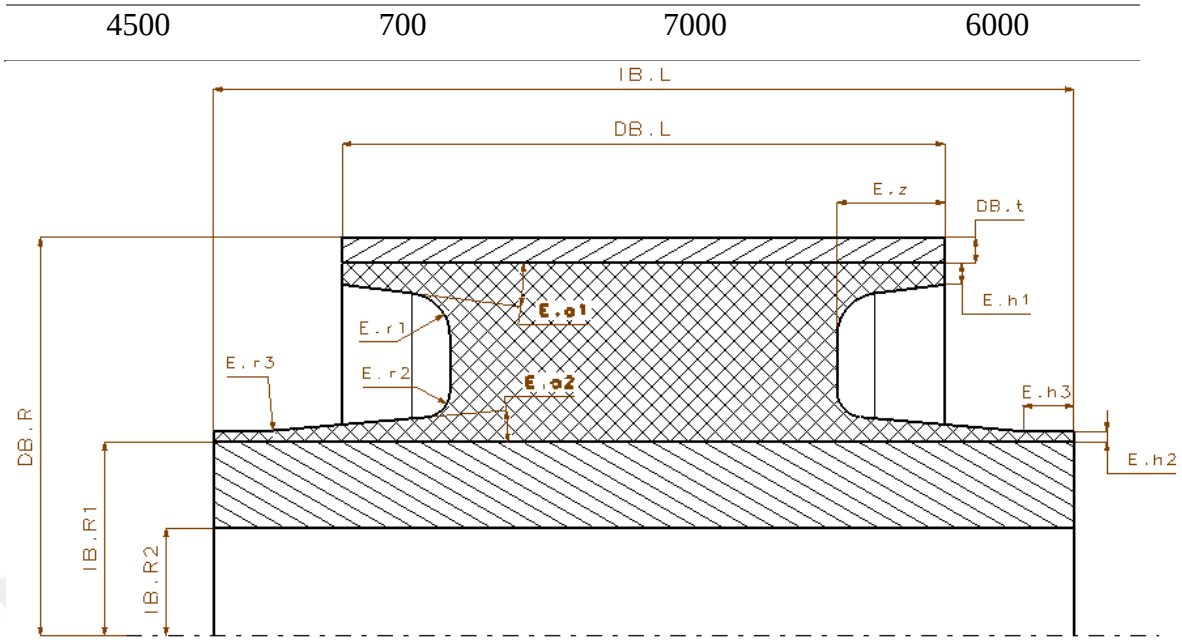
Burada w_i, i. kısıt fonksiyonuna (h_i) ait ağırlık fonksiyonudur ve tüm ağırlık fonksiyonlarının toplamı 1 olmalıdır. Bu çalışmada tüm kısıt fonksiyonları için eşit ağırlık değerleri kullanılmıştır. Buna göre ağırlıklandırılmış ve dönüştürülmüş hedef fonksiyonu aşağıdaki şekli alır.

$$\Phi = \left|\frac{k_r^H - k_r^i}{k_r^H}\right| + 0.3 * \left(0.33 * \left|\frac{k_e^H - k_e^i}{k_e^H}\right| + 0.33 * \left|\frac{k_t^H - k_t^i}{k_t^H}\right| + 0.34 * \left|\frac{k_k^H - k_k^i}{k_k^H}\right|\right) \tag{4.11}$$

Bu çalışmada kullanılacak hedef direngenlik değerleri Çizelge 4.39 ‘de, optimize edilecek tasarıma ait tasarım değişkenleri ise Şekil 4.22 ‘de gösterilmiştir. Tasarım değişkenlerine ait alt ve üst limit bilgileriyse Çizelge 4.40’de verilmiştir.

Çizelge 4.39 Hedef Direngenlik değerleri

Radyal [N/mm] k_r^H	**Eksenel [N/mm]** k_e^H	**Torsiyonel [Nmm/°]** k_t^H	**Kardanik [Nmm/°]** k_k^H



Şekil 4.22 Elastik Burç Tasarım Değişkenleri

Çizelge 4.40 Tasarım Değişkenleri Alt ve Üst Limitleri

No	Tasarım Parametreleri	Alt Limit	Üst Limit
1	Sertlik (ShA)	50	70
2	İç Kenar Açısı (E,a2)	5	8
3	Oyuk Derinliği (E,z)	3	9
4	Dış Kenar (E,h1)	0.5	1
5	Çaplama (mm)	-1.2	-0.2
6	İç Boru Set (E,h3)	3	5.0
7	İç Yarı Çap (IB,D1)	18	21.5
8	Dış Boru Boy	45	60
9	İç Boru Boy	65	75
10	Dış Yarı Çap (DB,D)	30	40

Oluşturulan optimizasyon probleminde, her bir popülasyonda 50 birey kullanılırken optimizasyon süreci 50 iterasyon ile sınırlandırılmıştır.

4.3.2. Kauçuk Burç Sonlu Elemanlar Modeli

Kauçuk burç direngenlik optimizasyonu için kullanılacak sonlu elemanlar analizleri Abaqus ticari yazılımı ile gerçekleştirilmiştir. Şekil 4.22 ‘de verilen geometri Abaqus CAE yazılımı içerisinde parametrik olarak oluşturulmuştur. Çözümde kullanılacak sonlu elemanlar modelinde, birinci dereceden 8 düğümlü altı yüzlü tuğla (hexa) eleman ve 6 düğümlü 5 yüzlü üçgen prizma elemanlar tercih edilmiştir. Elastik burçlarda kullanılan malzeme kauçuk olduğundan hiperelastik malzeme modeli kullanılmıştır. Abaqus içerisinde hiperelastik malzeme modeli ile hibrit elemanların kullanılması zorunlu olduğu için çözüm ağında hibrit elemanlar kullanılmıştır.

Sonlu elemanlar analizlerindeki hiperelastik malzemeler, Mooney-Rivlin, Ogden, Yeoh ve Neo-Hookean gibi farklı farklı malzeme modelleri kullanılarak modellenebilir (Güven, Yavuz Erkek, ve Kaya 2014), (Kaya 2014), (Güven 2014), (Mehmet Ali ÖZCAN 2016). Kauçuk malzemenin davranışı kullanılan hamurun karışımına ve vulkanizasyon işlemindeki üretim parametrelerine bağlı olarak değişiklik göstermektedir. Bu nedenle öncelikle parça üzerinde kullanılacak kauçuk hamuru ile malzeme testleri yapılır. Bu testler aşağıda listelenmiştir (Mehmet Ali ÖZCAN 2016)

- Tek eksenli çekme
- Düzlemsel çekme
- Çift Eksenli eş çekme
- Basma

Yukarıda belirtilen testlerden elde edilen sonuçlarına en uygun malzeme modeli belirlenir ve bu modele ait parametreler sonlu elemanlar yazılımına girilir. Bu çalışmada Mooney-Rivlin malzeme modelini tercih edilmiştir. Mooney-Rivlin modeline göre gerinim enerji fonksiyonu, Ψ aşağıdaki şekilde yazılır (Dal, Açıkgöz, ve Badienia 2021).

$$\Psi = C_{10}(I_1 - 3) + C_{01}(I_2 - 3) \tag{4.12}$$

Burada C_{10} ve C_{01} malzeme sabitleri, I_1 ve I_2 ise gerinim değişmezleridir. Kullanılacak lastik hamuruna ait C_{10} ve C_{01} malzeme sabitlerinin belirlenebilmesi için yukarıda anlatıldığı üzere malzeme testleri yapılmalıdır. Bu çalışmada her bir hamur sertliğine karşılık gelecek Mooney-Rivlin parametreleri için literatürde bulunan değerler kullanılmıştır (Altidis ve Warner 2005).

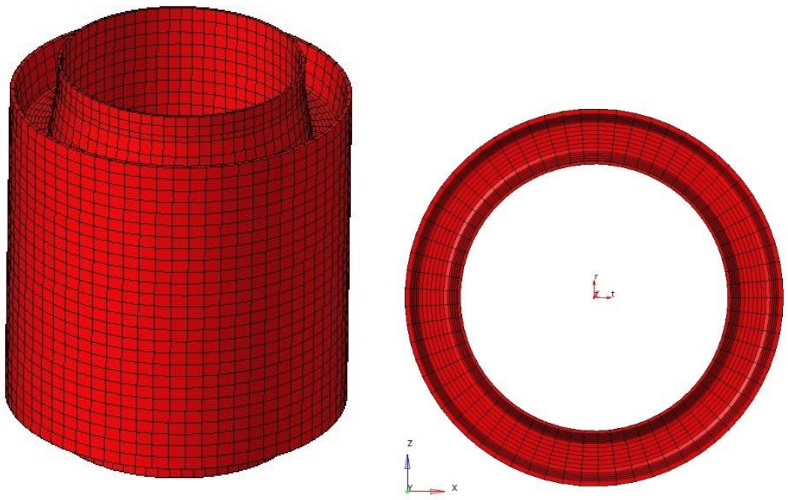
Lastiğe ait termal büzülmelerin de hesaplanabilmesi için termal genleşme katsayısı 6.5 10^{-4} mm/mmK alınmıştır.

Oluşturulan sonlu elemanlar modeli her bir kril bireyi için farklılık gösterse de mertebe olarak 15000 eleman ve 15000 düğümden oluşmaktadır ve Şekil 4.23 'te gösterilmiştir.

4.3.3. Sınır Koşulu ve Yükler

Elastik burç direngenlik analizleri aşağıdaki çözüm adımlarından oluşmaktadır.

- Termal Büzülme
- Çaplama
- Radyal Direngenlik
- Eksenel Direngenlik
- Torsiyonel Direngenlik
- Kardanik Direngenlik



Şekil 4.23 Elastik Burç Sonlu Elemanlar Modeli

4.3.3.1. Termal Büzülme

Elastik burcun lastik kalıbı içerisinde üretimi yüksek sıcaklıkta gerçekleşir. Kalıptan çıkan sıcak lastik oda sıcaklığına kadar soğurken termal büzülmeye uğrar ve şekil değiştirir. Bu şekil değişikliğin direngenlikler üzerinde etkisi vardır ve bu nedenle analiz sürecine dahil edilirler.

Vulkanizasyon kalıbından çıkan elastik burcun kalıp içerisindeki sıcaklığı olan 150 °C, lastik üzerine başlangıç sıcaklığı olarak tanımlanmıştır. Ardından ilk çözüm adımında lastiğe ait tüm düğümlere sıcaklık sınır koşulu olarak 25 °C sıcaklık tanımlanmıştır. Lastiğin vulkanize olduğu iç ve dış boru yüzeyleri ise, metalin termal büzülmesi ihmal edilebileceğinden, tüm serbestlik derecelerinde tutulmuştur.

4.3.3.2. Çaplama

Ön sıkışma olarak da adlandırılan bu adımda, vulkanizasyon kalıbından çıkan elastik burcun dış borusu belli oranda sıkıştırılarak kalıcı şekil değişikliğine uğratılır. Bu işlem direngenlikleri çeşitli oranlarda değiştirdiği gibi burç ömrü üzerinde de etkilidir. Direngenliği optimize edilecek elastik burca uygulanacak çaplama (calibration) miktarı bu çalışmadaki tasarım değişkenlerinden biridir.

Çaplama işleminde iç boru yüzeyindeki düğümler her yönde tutulmuştur. Dış boruya vulkanize olan düğümler ise Radyal yönde çaplama miktarı kadar hareket ettirilmişlerdir. Radyal yön dışındaki diğer yönler sabit tutulmuştur.

4.3.3.3. Radyal Direngenlik

İçi dolu bir elastik burcun her iki radyal yöndeki direngenliği aynıdır. Şekil 4.23 teki koordinat sistemine göre elastik burcun Radyal direngenliği X ve Z yönlerindeki direngenlikleridir. Bu yönlerdeki direngenliği hesaplayabilmek için dış boru yüzeyindeki düğümler sabit tutulurken iç boru yüzeylerinin bağlı olduğu kaskatı elemanın merkezdeki bağımsız düğümü X yönünde 0.1 mm hareket ettirilmiştir. Analiz sonrasında bu

düğümden okunan X yönündeki reaksiyon kuvveti, uygulanan yer değiştirme miktarına bölünerek radyal direngenlik değeri hesaplanmıştır.

4.3.3.4. Eksenel Direngenlik

Merkezdeki bağımsız düğüm, elastik burcun ekseni yönünde, Şekil 4.23'e göre Y yönünde 0.5 mm ötelenmiştir. Analiz sonrasında bu düğümden okunan Y yönündeki reaksiyon kuvveti, uygulanan yer değiştirme miktarına bölünerek eksenel direngenlik değeri hesaplanmıştır.

4.3.3.5. Torsiyonel Direngenlik

Merkezdeki bağımsız düğüm elastik burcun ekseni, Şekil 4.23'e göre Y ekseni etrafında 1 derecelik bir dönme hareketine zorlanmıştır. Analiz sonrasında bu düğümden okunan Y ekseni etrafındaki reaksiyon momenti, uygulanan dönme miktarına bölünerek torsiyonel direngenlik değeri hesaplanmıştır.

4.3.3.6. Kardanik Direngenlik

Merkezdeki bağımsız düğüm elastik burcun radyal ekseni, Şekil 4.23'e göre X ekseni etrafında 1 derecelik bir dönme hareketine zorlanmıştır. Analiz sonrasında bu düğümden okunan X ekseni etrafındaki reaksiyon momenti, uygulanan dönme miktarına bölünerek kardanik direngenlik değeri hesaplanmıştır.

4.3.4. Çözüm ve Sonuçların Değerlendirilmesi

Çözümler Windows 10 işletim sitemine sahip, işlemcisi 4 çekirdekli Intel i7-8650U olan ve 16GB hafızası olan bir bilgisayarda koşturulmuştur. Her bir çözüme ait çözüm süresi, oluşan modelin büyüklüğüne göre değişiklik gösterse de 50 saniye mertebelerindedir. Kurulan optimizasyon probleminin çözümü 24 saat sürmüştür.

Optimize edilen tasarım değişkenleri ve bu parametreler ile elde edilen direngenlik değerleri Çizelge 4.41 ve Çizelge 4.42 ‘de verilmiştir.

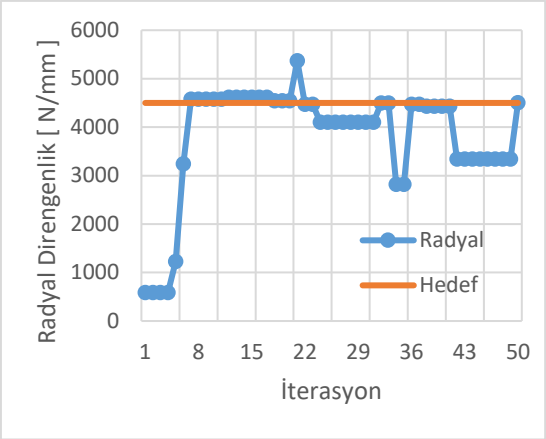
Çizelge 4.41 Optimize edilen tasarım değişkenleri

No	Tasarım Parametreleri	Alt Limit	Üst Limit	Optimize edilen
1	Sertlik (ShA)	51	70	56
2	İç Kenar Açısı	5.0	15.0	8,26
3	Oyuk Derinliği	5.0	15.0	12,25
4	Dış Kenar	0.5	1.0	0,80
5	Çaplama	-1.2	0.0	-0,96
6	İç Boru Set	2.0	7.0	5,79
7	İç Yarı Çap	14.0	24.0	23,98
8	Dış Boru Boy	45.0	60.0	54,92
9	İç Boru Boy	65.0	75.0	74,09
10	Dış Yarı Çap	32.0	45.0	32,52

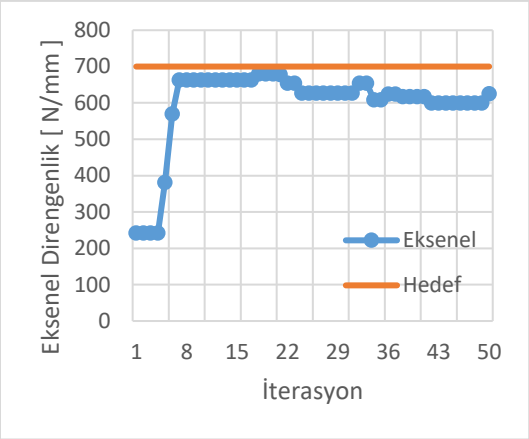
Çizelge 4.42 Optimize edilen direngenlik değerleri

	Radyal, k_r^H [N/mm]	**Eksenel, k_e^H [N/mm]**	**Torsiyonel, k_t^H [Nmm/°]**	**Kardanik, k_k^H [Nmm/°]**
Hedef	4500	700	7000	6000
Gerçekleşen	4502	625	7685	5997

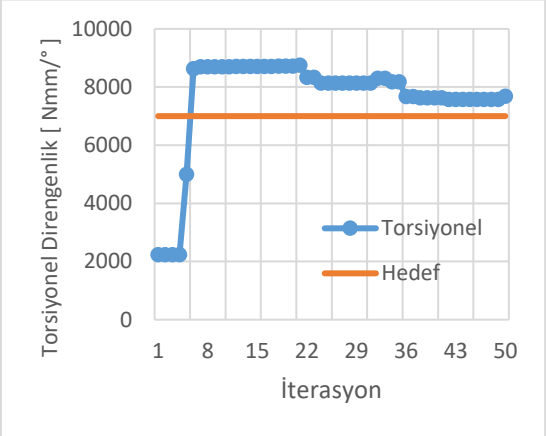
Direngenliklerin, tasarım değişkenlerinden bazılarının ve amaç fonksiyonun optimizasyon süresince olan değişimleri ise Şekil 4.24 - Şekil 4.29 ‘de gösterilmiştir.



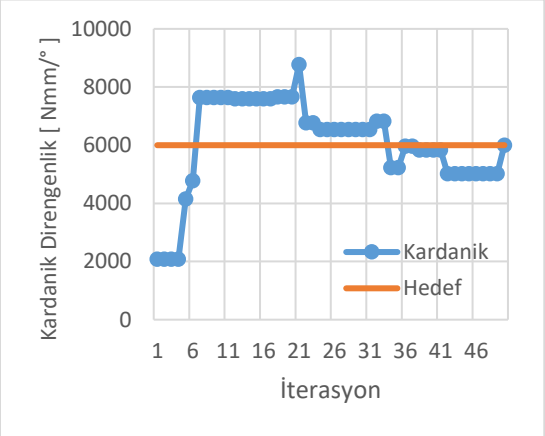
Şekil 4.24 Radyal direngenlik Değişimi



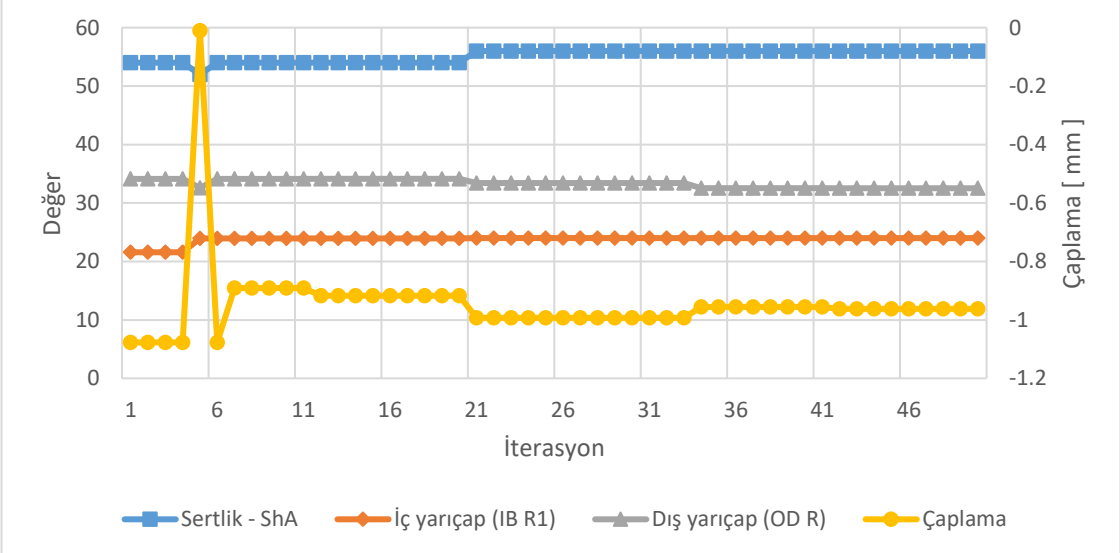
Şekil 4.25 Eksenel direngenlik değişimi



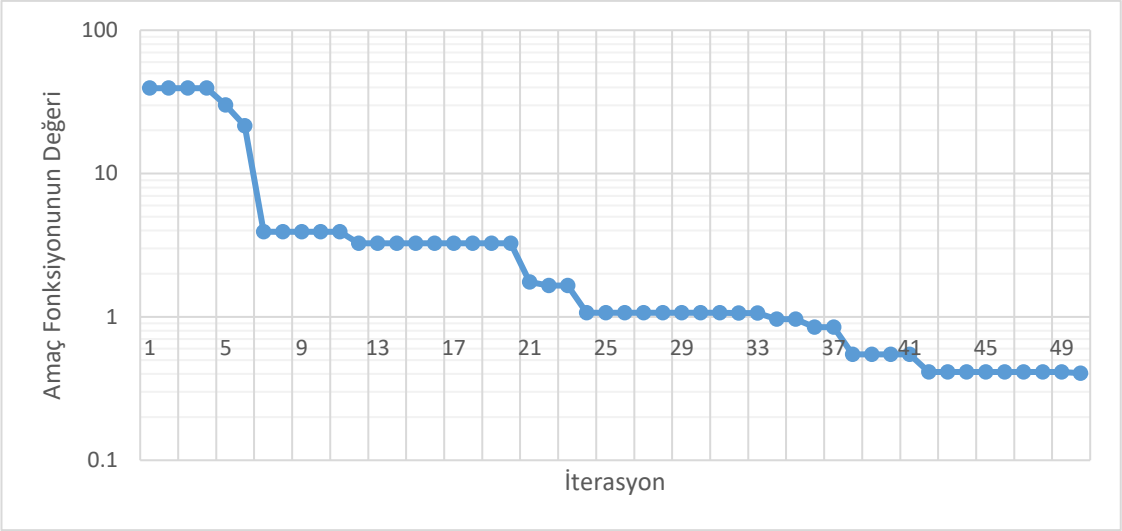
Şekil 4.26 Torsiyonel direngenlik değişimi



Şekil 4.27 Kardanik direngenlik değişimi



Şekil 4.28 Değişkenlerin çözüm sırasındaki değişimi



Şekil 4.29 Amaç fonksiyonunun optimizasyon süresince değişimi

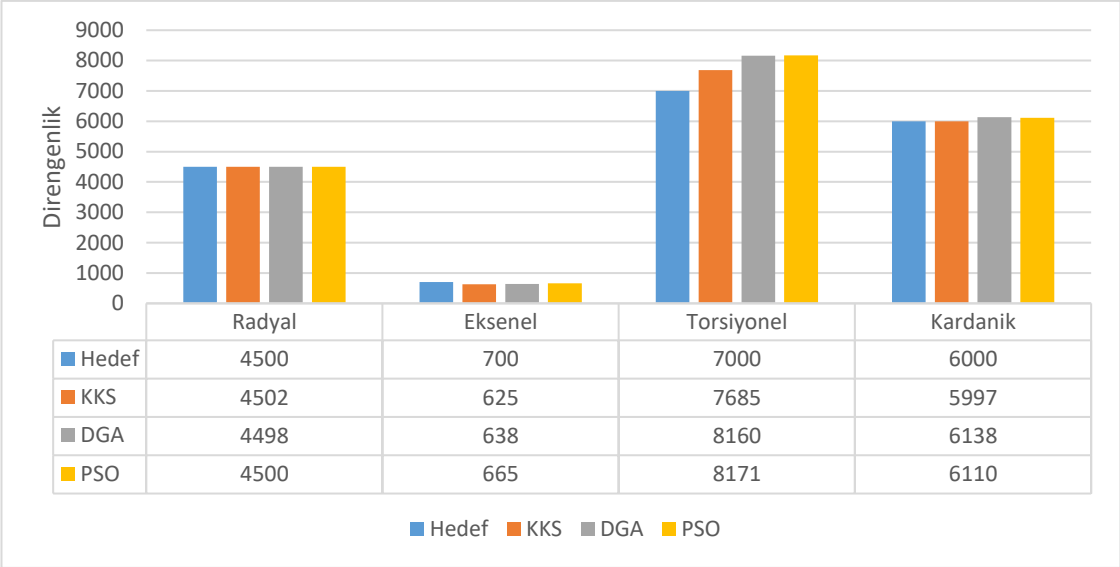
4.4. Kauçuk Burç Optimizasyonunda Farklı Optimizasyon Algoritmalarının Karşılaştırılması

Geliştirilen KKS optimizasyon yöntemini kullanılarak gerçekleştirilen kauçuk burç optimizasyonunun performansını test etmek için Diferansiyel Gelişim Algoritması (DGA) ile Parçacık Sürüsü Optimizasyonu (PSO) yöntemleri seçilmiştir. Bir önceki bölümde açıklanan sonlu elemanlar analizleri ile gerçekleştirilen optimizasyon problemi literatürde yer alan bu iki yöntem kullanılarak çözülmüştür.

DGA, İlk olarak 1995 yılında Storn ve Price tarafından geliştirilmiştir (Storn ve Price 1996). Price, benzetimli tavlama yönteminin yavaş yakınsama problemi ve kontrol parametrelerinin verimli kullanılamaması gibi dezavantajlara sahip olmasından dolayı bu yöntemi geliştirmeye çalışmıştır. İkili kodlama ve doğru/yanlış operatörlerini kullanan benzetimli tavlama yöntemi üzerinde değişiklik yaparak onun yerine aritmetik operatörler ve gerçek kodlama kullanan ve diferansiyel gelişim algoritmasında kilit rol oynayacak olan yeni bir diferansiyel mutasyon operatörü geliştirmiştir. Diğer tüm evrimsel algoritmalar gibi diferansiyel gelişim algoritması da popülasyon tabanlı bir optimizasyon algoritması olup özellikle başlangıç değer probleminin çözümü üzerine geliştirilmiş bir algoritmadır (K. Price, Storn, ve Lampinen 2005), (İdris Karen 2011).

PSO, son yıllarda araştırmacılar tarafından en çok tercih edilen sürü tabanlı bir optimizasyon yöntemidir (Nabaei vd. 2018). Parçacık sürü optimizasyonu 1995 yılında Kennedy ve Eberhart tarafından hayvanlar alemindeki sürü ve koloni hareketlerinden esinlenerek ileri sürülmüş bir algoritmadır. Bir sürü veya koloni, arılar, karıncalar ve kuşlar gibi organizmaların etkileşimleri sonucu toplanmış bir yapıdır. Bir sürüdeki her organizma bir parçacık olarak adlandırılmaktadır ve evrimsel algoritmalardaki bireyler burada parçacık olarak popülasyon ise sürü olarak ele alınmaktadır. Parçacık sürü optimizasyonunda bir sürü içindeki parçacığın pozisyonu optimizasyon parametrelerinin bir vektörünü temsil etmektedir. Bir parçacık evrimsel algoritmalardaki bireyin özelliklerini taşımanın yanında iki ek özniteliğe de sahiptir. Bunlar karşılaşılan en iyi pozisyon için belirlenen hız ve hafıza bileşenleridir. Buna ek olarak sürünün de bir hafızası mevcut olup burada her parçacığın en iyi pozisyon bilgileri tutulmaktadır. Benzer olarak her parçacığa ait en iyi hız bileşenleri de sürünün hız belleğinde tutulmaktadır(İdris Karen 2011), (James Kennedy ve Russell Eberhart 1995) .

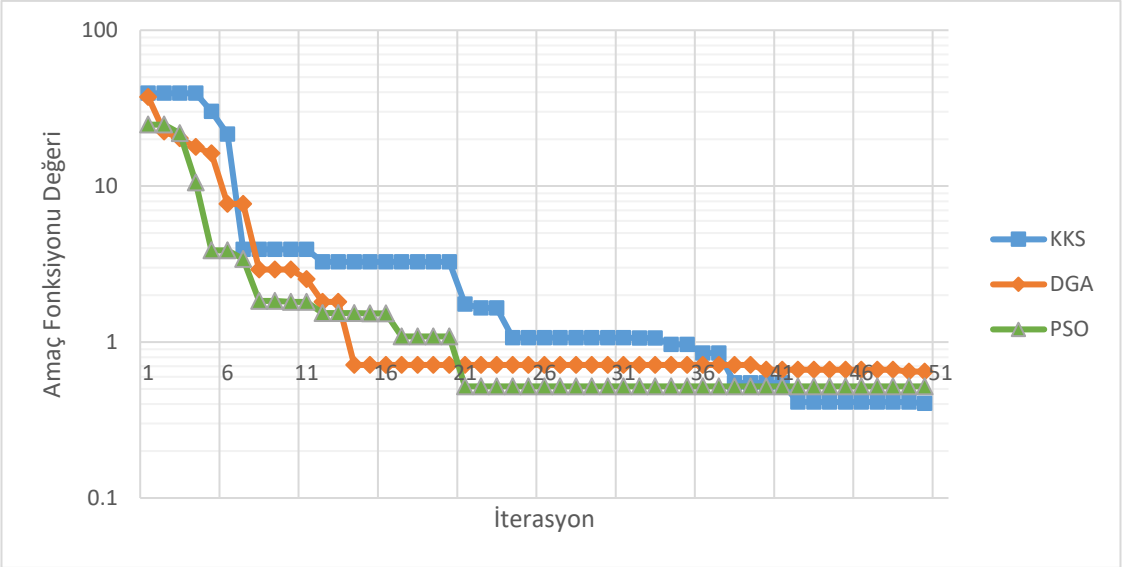
KKS, DGA ve PSO optimizasyon algoritmaları ile elde edilen direngenlik değerleri Şekil 4.30 ‘da gösterilmiştir. Hedef olarak alınan ve kısıtlar içerisinde en büyük ağırlığa sahip olan radyal direngenlik değerlerinin her üç algoritmada da tam hedef değerine ulaştığı görülmüştür. Eksenel direngenlik değerinde hedefe en yakın sonuca PSO yöntemi ile ulaşılmıştır. Torsiyonel ve kardanik direngenlik değerlerinde ise en iyi değer geliştirilen KKS yöntemi ile elde edilmiştir. Geliştirilen KKS yöntemi, eksenel hariç diğer 3 direngenlik değerinde en iyi sonuca ulaşan algoritma olmuştur. Hedeflenen radyal ve kardanik direngenlik değerlerine tam olarak ulaşırken, torsiyonel direngenlikte ise hedefe en yakın sonucu veren algoritmadır.



	Radyal	Eksenel	Torsiyonel	Kardanik
Hedef	4500	700	7000	6000
KKS	4502	625	7685	5997
DGA	4498	638	8160	6138
PSO	4500	665	8171	6110

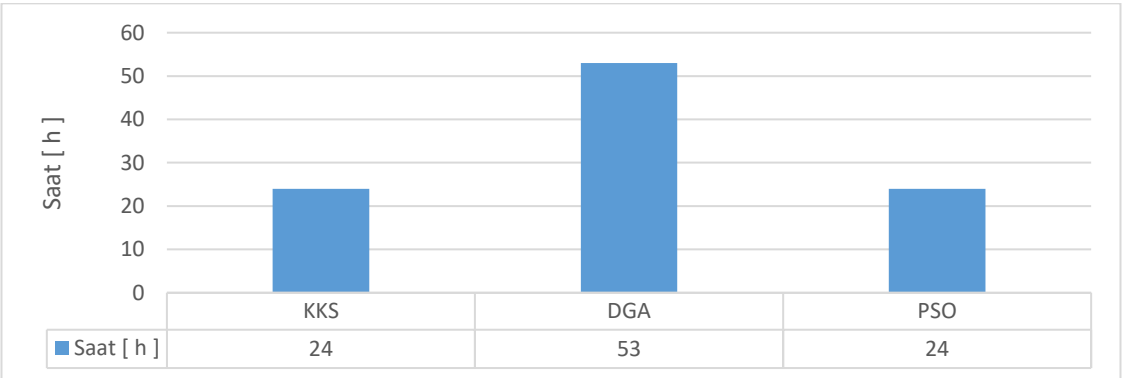
Şekil 4.30 KKS, PSO ve DGA yöntemlerinden elde edilen direngenlik değerleri

Şekil 4.31 ‘de her üç yöntem ile gerçekleştirilen çözümler sırasında, amaç fonksiyonundaki değişim raporlanmıştır. 50 iterasyonluk optimizasyon çözümünün sonundaki amaç fonksiyonları, KKS, DGA ve PSO için sırasıyla 0.42, 0.65 ve 0.52 ‘ye kadar minimize edilmiştir. Her üç algoritmada da başlangıç popülasyonları rastlantısal olarak belirlendiği için başlangıçtaki amaç fonksiyonun diğerine göre daha iyi bir değere sahip olması, algoritmanın bir başarısı olarak ifade edilemez. PSO, çözümün başlarında amaç fonksiyonunu hızlı bir şekilde iyileştirdiği ve 14. adıma kadar en iyi amaç değerine sahip olan yöntem olduğu görülmektedir. Bununla birlikte ilerleyen adımlarda DGA ‘nın öne geçtiği görülmüştür. Ancak DGA ‘sının, burada bulunan yerel bir minimuma takılı kaldığı ve çözümün kalanında neredeyse hiçbir ilerleme gösteremediği görülmektedir. Aksine KKS optimizasyon süresince az da olsa ilerlemeler göstermiştir. Bu da KKS ‘nün lokal minimumlarda sıkışmama ve daha başarılı keşif yeteneğine sahip olmasının bir sonucudur. Bu sayede optimizasyon sonunda DGA ‘nın ve PSO ‘nun önünde daha küçük bir amaç fonksiyonu ile çözümü tamamlayabilmiştir.



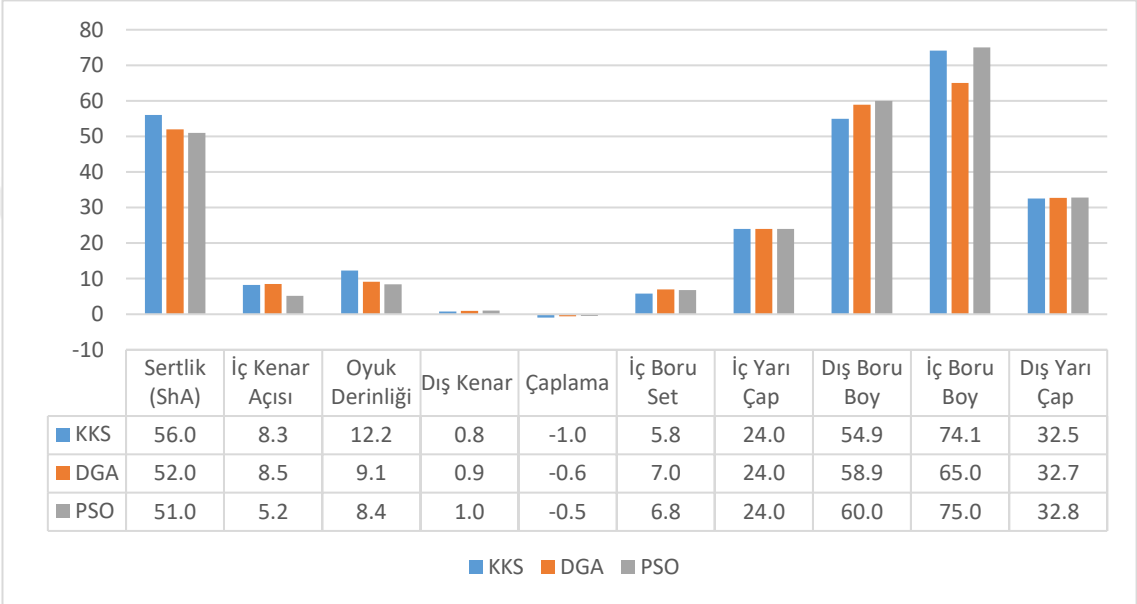
Şekil 4.31 KKS , PSO ve DGA amaç fonksiyonu değişimi

Şekil 4.32 'de ise her üç yönteme ait çözüm süreleri gösterilmiştir. KKS ve PSO yöntemlerinin çözüm sürelerinin eşitken, DGA yöntemine göre 2 kattan fazla daha hızlı oldukları açıkça görülmektedir. DGA 'daki bu yavaşlık, her bir iterasyonda popülasyondaki her bir bireyin amaç fonksiyonunu, yeni oluşturulan aday ile teker teker karşılaştırmak zorunda olmasıdır. Bu zorunluluk aynı anda birden fazla bireye ait çözüm yapmayı engellemektedir. Buna karşılık KKS ve PSO yöntemlerinde ise aynı iterasyonda tüm bireylere ait amaç fonksiyonu paralel olarak hesaplanabilmektedir. Gerçekleştirilen çözümlerde aynı anda 2 çözüm yapıldığı için çözüm süresi farkı 2 kat çıkmıştır. Daha çok işlemcili bilgisayarların kullanılması ve aynı anda koşturulacak analiz sayısının artırılması durumunda çözüm süreleri arasındaki bu fark daha da açılacaktır.



Şekil 4.32 KKS ve DGA optimizasyon problemi çözüm süreleri

Şekil 4.33 ‘de her üç yöntem ile optimize edilen tasarım parametreleri listelenmiştir. Her üç yöntem ile elde edilen tasarım değerlerindeki bu farklılık, gerçekçi endüstriyel problemlerdeki doğrusalsızlıkları ve tasarımın çok sayıda lokal optimumlarının olabileceğini göstermiştir. Bu da kauçuk burç şekil optimizasyonu problemi gibi karmaşık doğrusal olmayan optimizasyon problemlerinde global optimuma ulaşmak için yarı sezgisel algoritmaların tercih edilmesini gerektirir.



	Sertlik (ShA)	İç Kenar Açısı	Oyuk Derinliği	Dış Kenar	Çaplama	İç Boru Set	İç Yarı Çap	Dış Boru Boy	İç Boru Boy	Dış Yarı Çap
KKS	56.0	8.3	12.2	0.8	-1.0	5.8	24.0	54.9	74.1	32.5
DGA	52.0	8.5	9.1	0.9	-0.6	7.0	24.0	58.9	65.0	32.7
PSO	51.0	5.2	8.4	1.0	-0.5	6.8	24.0	60.0	75.0	32.8

Şekil 4.33 KKS ve DGA ile optimize edilen tasarım değişkenleri

5. TARTIŞMA ve SONUÇ

5.1. KKS Katkıları ve Güçlü Yönleri

Wang ve ark. tarafından geliştirilen KKS algoritması incelendiğinde kaotik algoritmanın yem arama adımına eklendiği görülmektedir. Eşitlik 11 'deki atalet ağırlığı w_f katsayısı, Gandomi ve ark. tarafından sunulan orijinal KS algoritmasında sabit bir sayı olarak tanımlanmıştır. Wang ve ark. bu parametreyi sabit almak yerine optimizasyon süresince kaotik olarak değiştirmenin etkili olduğunu göstermişlerdir. Bu sayede bu sabitin kaotik olarak hem optimizasyon süresince hem de her bir kril bireyi için farklı bir değer almasıyla kril bireyinin yeme yönelmesi kaotik bir davranış göstermiştir. Bu davranış ta, lokal optimumlardan kurtularak global optimuma ulaşmayı kolaylaştırdığı belirtilmiştir. Bununla birlikte KS algoritmasında kril bireyinin hareketi, diğer kril bireyleri ve fiziksel yayılmadan da etkilenmektedir. Bu nedenle sadece yem arama adımının kaotik olması algoritma üzerinde kısmi bir etkiye sahip olacaktır. Diğer kril bireylerinin ve fiziksel yayılma aşamalarının da kaotik hale getirilmesi algoritmanın performansını iyileştirecektir.

Diğer kril bireylerinin sebep olduğu hareketi hesaplayan Eşitlik 3' te, doğrultu vektörü α_i , α_i^{lokal} ve α_i^{hedef} parametrelerine bağlı olarak 4 nolu eşitlik ile hesaplanır. Burada α_i^{lokal} komşu bireylerin etkisiyle oluşan doğrultuyu belirlerken, α_i^{hedef} ise en iyi kril bireyinin etkisiyle oluşan doğrultuyu belirler. α_i^{hedef}, 9 nolu eşitlik ile elde edilir. Bu eşitlikteki C^{eniyi} , 10 nolu eşitlik ile elde edilir. Bu eşitlikte rastgele üretilen bir sayı yer almaktadır. Bu rastgele sayı en iyi kril bireyinin etkisini rastlantısal olarak değiştirmektedir.

Bu çalışmada 10 numaralı eşitlikte yer alan rastgele sayı kaotik harita fonksiyonları kullanılarak belirlenmiştir. Bu sayede Wang ve ark. tarafından geliştirilen KKS algoritmasına, diğer kril bireylerinin sebep olduğu hareketin hesabının da kaotik yapılmasıyla katkı sağlanmıştır.

Bununla birlikte, KS algoritmasında kaotik sistemlerin kullanılabileceği bir başka alan ise başlangıç popülasyonunun belirlenmesi adımıdır. Hem Gandomi ve ark. tarafından sunulan orijinal KS yöntemi hem de Wang ve ark. tarafından sunulan KKS yöntemi bu aşamada rastgele belirlenen sayıları kullanmışlardır. Bu çalışmada başlangıç popülasyonun belirlenmesi sırasında da farklı kaotik harita fonksiyonlarının kullanılması test edilmiştir.

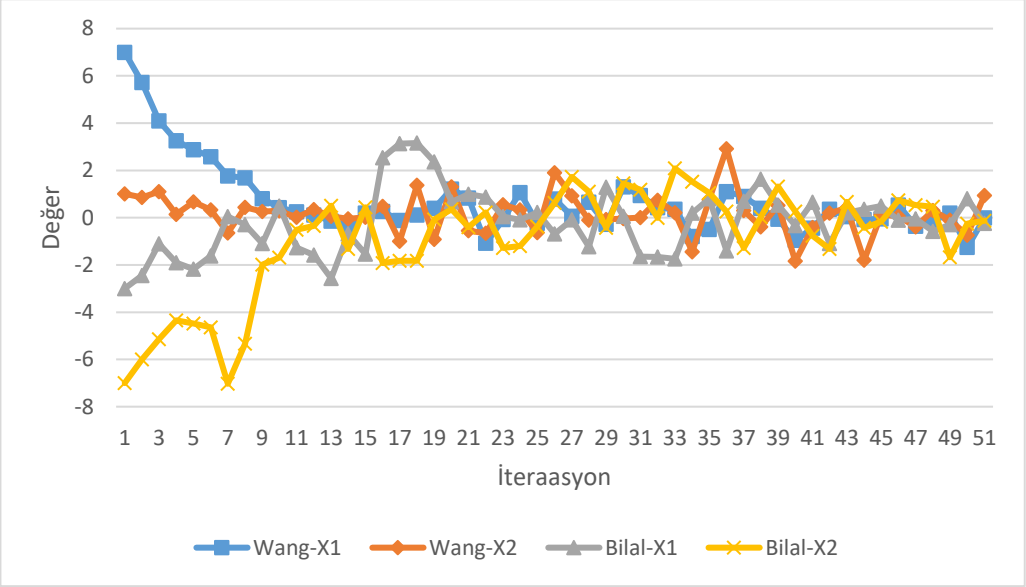
Yapılan çalışmada, en iyi birey etkisinin ve başlangıç popülasyonun kaotik olarak hesaplanmalarının hem tekil olarak hem de birlikte kullanılmaları durumunda optimizasyonun performansı nasıl değiştireceği araştırılmıştır. Oluşturulan ve test edilen algoritmalar aşağıdaki şekilde adlandırılmaktadır.

Algoritma 1: Mevcut KS algoritması
Algoritma 2: Mevcut Wang KKS algoritması
Algoritma 3: Wang + Sadece başlangıç popülasyonunun pozisyonlarının kaotik olduğu KKS (Önerilen – 1)
Algoritma 4: Wang + Sadece C^{eniyi} hesabının kaotik olduğu KKS (Önerilen – 2)
Algoritma 5: Wang + Her iki adımın da kaotik olduğu KKS (Önerilen – 3)
Algoritma 6: Başlangıç popülasyonunun ve C^{eniyi} hesabının kaotik olduğu KKS (Önerilen – 4)
Algoritma 7: PSO Algoritması
Algoritma 8: GA
Algoritma 9: DGA

Çizelge 4.1 - Çizelge 4.38 ve Şekil 4.1 - Şekil 4.15 ‘de sunulan sonuçlara göre Algoritma 6 ‘da yer alan kaotik fonksiyon kombinasyonunun diğer algoritmalardan daha etkili olduğunu söylemek mümkündür. Wang ve ark. tarafından geliştirilen KKS (Algoritma 2) ’de kullanılan kaotik sayı çözümdeki maksimum iterasyon sayısı kadardır. Her bir iterasyonda her bir birey için aynı atalet ağırlığı değeri atanır. Bununla birlikte geliştirilen yeni KKS ‘de (Algoritma 6) her iterasyonda ve her bir birey için sürekli değişmektedir. Bu algoritmada kullanılan kaotik sayı miktarı, amaç fonksiyonundaki parametre sayısı ile her bir popülasyondaki kril sayısı ve toplam iterasyon sayısının çarpımı kadardır. Bu

çalışmada kullanılan değerlere göre Algoritma 2'de 50 adet kaotik sayı kullanılırken Algoritma 6 'da bu sayı 5000 'e çıkmıştır. Kullanılan kaotik sayıdaki bu artış algoritmadaki kaotik davranışı da artırmaktadır.

Algoritma 6 ile önerilen algoritmanın etkisini daha iyi ortaya koyabilmek için, rastgele olarak seçilen bir krilin optimizasyon süresince X1 ve X2 değişkenlerinin aldıkları değerler Şekil 5.1 ve Şekil 5.2 'de karşılaştırılmıştır. Burada amaç fonksiyonu olarak Ackley fonksiyonu kullanılmıştır. Şekil 5.1 'de değişkenlerin değerlerinin optimizasyon süresince nasıl değiştikleri gösterilmiştir. Karşılaştırılan her iki algoritma da değişkenlerinin başlangıç değerleri benzer seviyelerdedir.



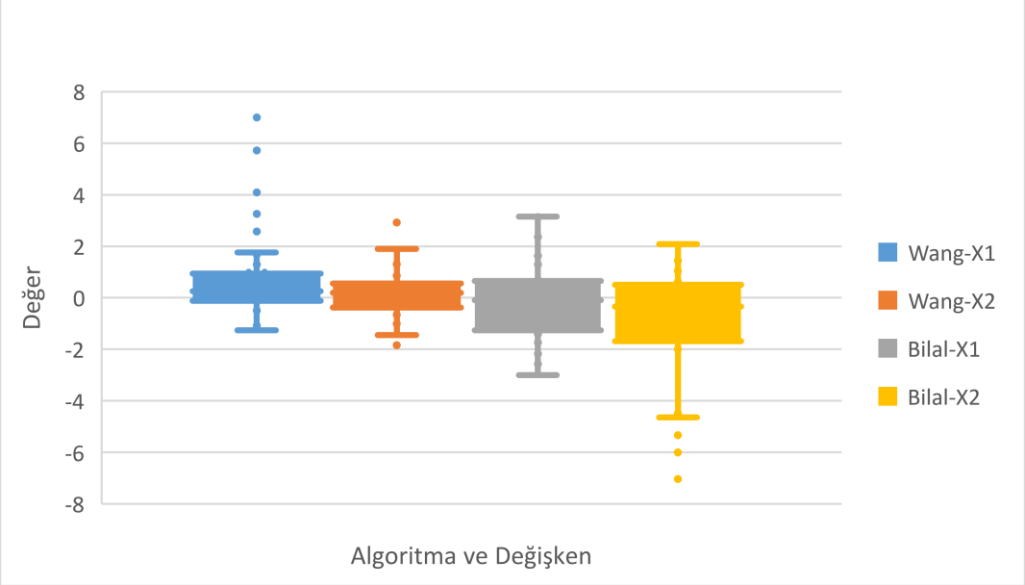
Şekil 5.1 X1 ve X2 değişkenlerinin İterasyon boyunca aldıkları değerler

Bununla birlikte bu grafiğe bakarak değerlerdeki değişimin büyüklükleri konusunda net bir çıkarımda bulunmak mümkün değildir. Değerlerin istatistiksel olarak dağılımını gözlemleyebilmek için Şekil 5.2 - Şekil 5.4'te sunulan kutu ve çubuk dağılım grafikleri oluşturulmuştur. Şekil 5.2'de X1 ve X2 değişkenlerinin çözüm noktası olan 0 değerinin etrafındaki salınımları kutu grafikte gösterilmiştir. Wang algoritmasında X1 ve X2 değişkenlerinin aldıkları değerlerin, çözümün etrafında, Algoritma 6 'ya göre daha dar bir aralıkta dağıldığı net bir şekilde görülmektedir. Algoritma 6 'nın saçınıklığı Şekil 5.3 ve Şekil 5.4 'te sunulan çubuk dağılım grafiklerinde de açıkça görülmektedir. Algoritma

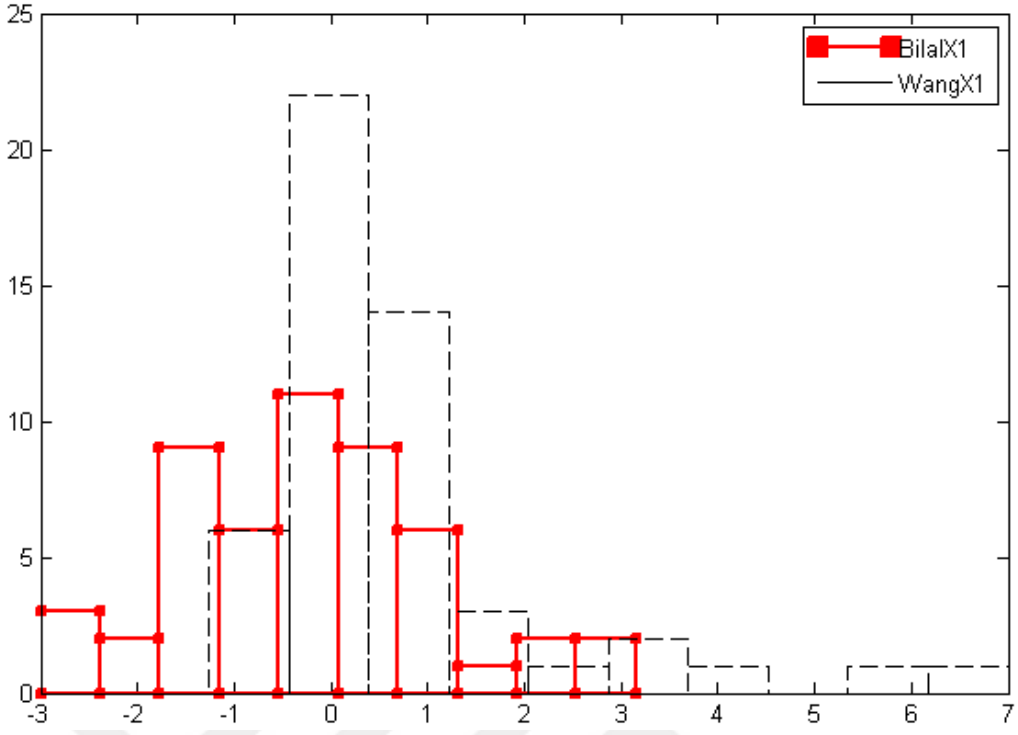
6 ’nın bu davranışı en iyi noktayı bulma yönünde sürekli arayışta olduğu ve daha çok sayıdaki değeri tarayabildiğini göstermektedir.

Her iki algoritmadan elde edilen değerlerin hesaplanan standart saplamaları da Şekil 5.5’te gösterilmiştir. Algoritma 6 ile önerilen yöntemin çözüm sırasında taradığı değerlerin daha büyük standart sapmaya sahip olmaları daha geniş bir uzayı taradığının bir başka göstergesidir.

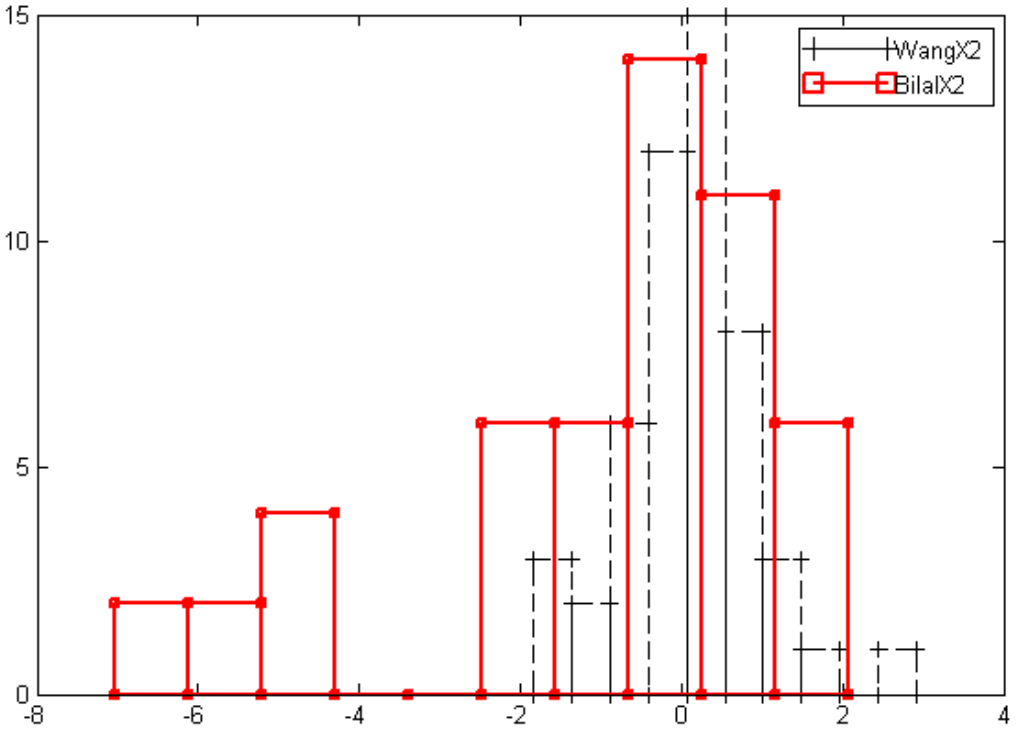
Algoritma 6 ‘nın bu saçınıklığı barındırdığı Chebyshev harita fonksiyonunun Çizelge 3.1 ‘de görüldüğü üzere, göreceli olarak yüksek denilebilecek bir Lyapunov üsteline (0.6932) sahip olmasının bir sonucudur. Bununla birlikte Şekil 3.34 ‘da görülebileceği üzere bu harita fonksiyonun olasılık dağılım fonksiyonu limit değerlere izin verirken ara değerlerde de eşit oranda bir dağılıma sebep olmaktadır.



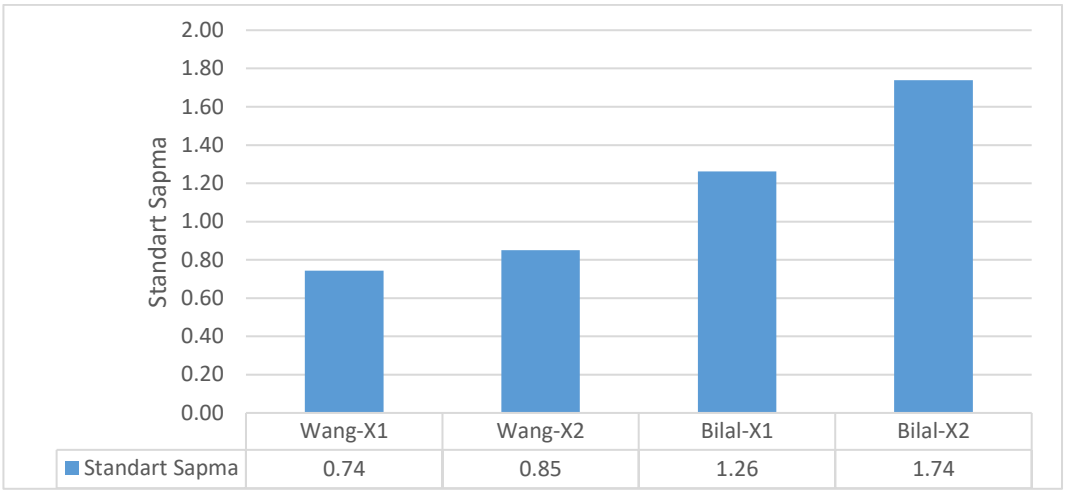
Şekil 5.2 Değişkenlerin aldığı değerlere ait kutu grafiği



Şekil 5.3 X1 değerlerinin çubuk dağılım grafiği



Şekil 5.4 X2 değerlerinin çubuk dağılım grafiği



Şekil 5.5 X1 ve X2 değerlerinin Standart Sapmaları

Bu çalışmada geliştirilen KKS yöntemi kıyaslama fonksiyonları ile test edilmiş ve literatürdeki diğer sürü tabanlı algoritmalara göre üstünlüğü kanıtlanmıştır. Ardından gerçekçi bir endüstriyel problem üzerindeki uygulamasıyla hem endüstriyel problemlerde kullanılabilirliğini ispatlanmış hem de diğer optimizasyonlara göre üstünlüğünü bir kez daha kanıtlamıştır.

5.2. Ek Çalışmalar

Gerçekleştirilen çalışmada önerilen algoritmalardaki her bir kaotik tanımlamada hep aynı harita fonksiyonu kullanılmıştır. Şöyle ki, başlangıç popülasyonundaki kaotik dizinin oluşturulmasında kullanılan harita fonksiyonu ne ise aynı algoritmadaki C^{eniyi} adımında kullanılan harita fonksiyonu da aynıdır. Her bir kaotik tanımdaki harita fonksiyonun bir diğerinden farklı olduğu durum bu çalışma kapsamı dışında kalmıştır. Bu nedenle bu esnekliğin KKS yönteminin performansına nasıl bir etkisi olacağı araştırmacılar için iyi bir araştırma konusu olabilir.

Bununla birlikte yapılan çalışma sırasında gerek kıyaslama fonksiyonlarındaki gerekse kauçuk burç optimizasyonundaki başarısıyla PSO ‘nun oldukça rekabetçi bir optimizasyon algoritması olduğu görülmüştür. Bu nedenle bu çalışmada olduğu gibi literatürdeki kaotik PSO hakkında yapılan çalışmalar incelenerek ardından kaotik davranışını artıracak yeni öneriler geliştirilebilir.

KAYNAKLAR

Abdel-Basset, Mohamed, Gai-Ge Wang, Arun Kumar Sangaiah, ve Ehab Rushdy. **2017**. "Krill herd algorithm based on cuckoo search for solving engineering optimization problems". *Multimedia Tools and Applications* 78 (4): 3861–84. https://doi.org/10.1007/s11042-017-4803-x.

Aggestam, Emil, ve Jens C.O. Nielsen. **2019**. "Multi-objective optimisation of transition zones between slab track and ballasted track using a genetic algorithm". *Journal of Sound and Vibration* 446 (Nisan): 91–112. https://doi.org/10.1016/j.jsv.2019.01.027.

Agrawal, Prasun Kumar, Manjaree Pandit, ve Hari Mohan Dubey. **2016**. "Improved Krill Herd Algorithm with Neighborhood Distance Concept for Optimization". *International Journal of Intelligent Systems and Applications* 8 (11): 34–50. https://doi.org/10.5815/ijisa.2016.11.05.

Ahmed, Khaled, Ahmed Ibrahem Hafez, ve Aboul Ella Hassanien. **2015**. "A discrete Krill herd optimization algorithm for community detection". *2015 11th International Computer Engineering Conference (ICENCO)*, 297–302. IEEE. https://doi.org/10.1109/ICENCO.2015.7416365.

Akdeniz, Çağrı, Erkut Yalçın, ve Halil Bilal. **2016**. "Bir binek araç sonlu elemanlar modelinin burulma modu optimizasyonu". *OTEKON 2016*. Bursa.

Alam, Mahamad Nabab. **2016**. "Particle Swarm Optimization: Algorithm and its Codes in MATLAB". https://doi.org/10.13140/RG.2.1.4985.3206.

Alikhani, Amirhossein, Amir Abolfazl Suratgar, Kayvan Nouri, Mina Nouredanesh, ve Sarah Salimi. **2013**. "Optimal PID tuning based on Krill Herd optimization algorithm". *The 3rd International Conference on Control, Instrumentation, and Automation*, 11–15. IEEE. https://doi.org/10.1109/ICCIAutom.2013.6912801.

Altair Inc. 2018. *Practical Aspects of Optimizasyon*. Altair Inc.

Altidis, P, ve B Warner. **2005**. "Analyzing Hyperelastic Materials/Some Practical Considerations". https://pdfslide.net/documents/ansys-users-grouphyperelastic-materials.html.

Amaran, Satyajith, Nikolaos V. Sahinidis, Bikram Sharda, ve Scott J. Bury. **2016**. "Simulation optimization: a review of algorithms and applications". *Annals of Operations Research* 240 (1): 351–80. https://doi.org/10.1007/s10479-015-2019-x.

Arora, Jasbir Singh. **2017**. *Introduction to Design Optimization*. *Introduction to*

Optimum Design. Third Edit. Elsevier. https://doi.org/10.1016/b978-0-12-800806-5.00001-9.

Arora, Vaibhav, Pulkit Sood, ve Kumar Utkarsh Keshari. **2016**. "A comparison of HPSOWM, krill herd and Spider Monkey optimization algorithms". *2015 2nd International Conference on Recent Advances in Engineering and Computational Sciences, RAECS 2015*, 1–5. IEEE. https://doi.org/10.1109/RAECS.2015.7453377.

Ashrafzadeh, Afshin, Mohammad Ali Ghorbani, Seyed Mostafa Biazar, ve Zaher Mundher Yaseen. **2019**. "Evaporation process modelling over northern Iran: application of an integrative data-intelligence model with the krill herd optimization algorithm". *Hydrological Sciences Journal* 64 (15): 1843–56. https://doi.org/10.1080/02626667.2019.1676428.

Aulig, Nikola, Emily Nutwell, Stefan Menzel, ve Duane Detwiler. **2016**. "Preference based Topology Optimization of Body-in-white Structures for Crash and Static Loads". *4 th International LS-DYNA Users Conference*.

Baby Resma, K.P., ve Madhu S. Nair. **2018**. "Multilevel thresholding for image segmentation using Krill Herd Optimization algorithm". *Journal of King Saud University - Computer and Information Sciences*, Nisan. https://doi.org/10.1016/j.jksuci.2018.04.007.

Bahrami, Helena, Karim Faez, ve Marjan Abdechiri. **2010**. "Imperialist competitive algorithm using chaos theory for optimization: (CICA)". *UKSim2010 - UKSim 12th International Conference on Computer Modelling and Simulation*, 98–103. https://doi.org/10.1109/UKSIM.2010.26.

Baykasoglu, Adil. **2012**. "Design optimization with chaos embedded great deluge algorithm". *Applied Soft Computing Journal* 12 (3): 1055–67. https://doi.org/10.1016/j.asoc.2011.11.018.

Bendsoe, Martin Philip, ve Noboru Kikuchi. **1988**. "GENERATING OPTIMAL TOPOLOGIES IN STRUCTURAL DESIGN USING A HOMOGENIZATION METHOD". *COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING*. C. 71.

Bentouati, Bachir, Saliha Chettih, ve Ragab A El-Sehiemy. **2017**. "A Chaotic Krill Herd Algorithm for Optimal Solution of the Economic Dispatch Problem". *International Journal of Engineering Research in Africa* 31: 2017–20.

https://doi.org/10.4028/www.scientific.net/JERA.31.155.

Bhise, Vivek D. 2017. *Automotive Product Development*. *Automotive Product Development*. CRC Press. https://doi.org/10.1201/9781315119502.

Bhowmik, Mahua, ve P. Malathi. **2019**. "Spectrum Sensing in Cognitive Radio Using Actor–Critic Neural Network with Krill Herd-Whale Optimization Algorithm". *Wireless Personal Communications* 105 (1): 335–54. https://doi.org/10.1007/s11277-018-6115-5.

Bilal, Halil. **2018**. "Suspansiyon Parçalarında Optimizasyon Uygulamaları". *Altair Users Meeting Türkiye*. Bursa.

Bilal, Halil, Ugurcan Serbest, ve Tamer Aydıner. **2018**. "Using Genetic Algorithm to Optimize Hand Brake Bracket Design". *Altair Technology Conference*, 25. Paris: Altair Inc.

"Bushing (isolator) - Wikipedia". y.y. Erişim 11 Ağustos 2020. https://en.wikipedia.org/wiki/Bushing_(isolator).

Cengiz, Ekrem, Hasan Ayyildiz, ve Fazıl Kirkbir. **2006**. "Yeni Ürün Geliştirme Sürecinde Aşama-Eşiği Yöntemiyle Süreç Performans Değerlemesi". *Atatürk Üniversitesi Sosyal Bilimler Enstitüsü Dergisi* 07 (01): 435–52.

Chaturvedi, Shivam, Pallavi Pragya, ve H. K. Verma. **2015**. "Comparative analysis of particle swarm optimization, genetic algorithm and krill herd algorithm". *2015 International Conference on Computer, Communication and Control (IC4)*, 1–7. IEEE. https://doi.org/10.1109/IC4.2015.7375552.

Cheng, Chun Tian, Wen Chuan Wang, Dong Mei Xu, ve K. W. Chau. **2008**. "Optimizing hydropower reservoir operation using hybrid genetic algorithm and chaos". *Water Resources Management* 22 (7): 895–909. https://doi.org/10.1007/s11269-007-9200-1.

Ciornei, I., ve E. Kyriakides. **2012**. "Hybrid Ant Colony-Genetic Algorithm (GAAPI) for Global Continuous Optimization". *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 42 (1): 234–45. https://doi.org/10.1109/TSMCB.2011.2164245.

Cooper, Robert G. 1990. "Stage-gate systems: A new tool for managing new products". *Business Horizons* 33 (3): 44–54. https://doi.org/10.1016/0007-6813(90)90040-i.

Cooper, Robert G. 1994. "Third-Generation New Product Processes". *Journal of Product Innovation Management* 11 (1): 3–14. https://doi.org/10.1111/1540-

5885.1110003.

Dal, Hüsnü, Kemal Açıkgöz, ve Yashar Badienia. **2021**. "On the Performance of Isotropic Hyperelastic Constitutive Models for Rubber-Like Materials: A State of the Art Review". *Applied Mechanics Reviews* 73 (2). https://doi.org/10.1115/1.4050978.

Das, Rashmi Ranjan, Vinodh Kumar Elumalai, Raaja Ganapathy Subramanian, ve Kadiyam Venkata Ashok Kumar. **2018**. "Adaptive predator–prey optimization for tuning of infinite horizon LQR applied to vehicle suspension system". *Applied Soft Computing Journal* 72 (Kasım): 518–26. https://doi.org/10.1016/j.asoc.2018.06.044.

Duddeck, Fabian. **2012**. "A new Topology Optimization Approach for Crashworthiness of Passenger Vehicles Based on Physically Defined Equivalent Static Loads". ICrash Int. Crashworthiness Conf. https://www.researchgate.net/publication/285594449.

Fang, Jianguang, Yunkai Gao, Guangyong Sun, Chengmin Xu, ve Qing Li. **2015**. "Multiobjective robust design optimization of fatigue life for a truck cab". *Reliability Engineering and System Safety* 135: 1–8. https://doi.org/10.1016/j.ress.2014.10.007.

Fathy, Ahmed, ve Almoataz Y. Abdelaziz. **2018**. "Single and multi-objective operation management of micro-grid using krill herd optimization and ant lion optimizer algorithms". *International Journal of Energy and Environmental Engineering* 9 (3): 257–71. https://doi.org/10.1007/s40095-018-0266-8.

Fattahi, Edris, Mahdi Bidar, ve Hamidreza Rashidy Kanan. **2016**. "Fuzzy Krill Herd (FKH): An improved optimization algorithm". *Intelligent Data Analysis* 20 (1): 153–65. https://doi.org/10.3233/IDA-150798.

Feldman, David B. 2012. *Chaos and Fractals*. 1. baskı. Oxford,: Oxford University Press. https://doi.org/DOI:10.1093/acprof:oso/9780199566433.001.0001.

Feng, Junhong, Jie Zhang, Xiaoshu Zhu, ve Wenwu Lian. **2017**. "A novel chaos optimization algorithm". *Multimedia Tools and Applications* 76 (16): 17405–36. https://doi.org/10.1007/s11042-016-3907-z.

Fister, Iztok, Matjaž Perc, Salahuddin M. Kamal, ve Iztok Fister. **2015**. "A review of chaos-based firefly algorithms: Perspectives and research challenges". *Applied Mathematics and Computation*. Elsevier Inc. https://doi.org/10.1016/j.amc.2014.12.006.

Gai, Wendong, Chengzhi Qu, Jie Liu, ve Jing Zhang. **2018**. "A novel hybrid meta-heuristic algorithm for optimization problems". *Systems Science and Control Engineering* 6 (3): 64–73. https://doi.org/10.1080/21642583.2018.1531359.

Gandomi, A. H., Xin She Yang, S. Talatahari, ve A. H. Alavi. **2013**. "Firefly algorithm with chaos". *Communications in Nonlinear Science and Numerical Simulation* 18 (1): 89–98. https://doi.org/10.1016/j.cnsns.2012.06.009.

Gandomi, Amir H., ve Amir H. Alavi. **2015**. "An Introduction Of Krill Herd Algorithm For Engineering Optimization". *Journal Of Civil Engineering And Management* 22 (3): 302–10. https://doi.org/10.3846/13923730.2014.897986.

Gandomi, Amir Hossein, ve Amir Hossein Alavi. **2012**. "Krill herd: A new bio-inspired optimization algorithm". *Communications in Nonlinear Science and Numerical Simulation* 17 (12): 4831–45. https://doi.org/10.1016/j.cnsns.2012.05.010.

Gao, Shangce, Catherine Vairappan, Yan Wang, Qiping Cao, ve Zheng Tang. **2014**. "Gravitational search algorithm combined with chaos for unconstrained numerical optimization". *Applied Mathematics and Computation* 231 (Mart): 48–62. https://doi.org/10.1016/j.amc.2013.12.175.

Gharavian, Leila, Mahdi Yaghoobi, ve Peiman Keshavarzian. **2013**. "Combination of krill herd algorithm with chaos theory in global optimization problems". *2013 3rd Joint Conference of AI & Robotics and 5th RoboCup Iran Open International Symposium*, 1–6. IEEE. https://doi.org/10.1109/RIOS.2013.6595310.

Goelke, Matthias. **2017**. "Introduction into Design of Experiments DOE with HyperStudy".

Gölcük, İlker, Adil Baykasoğlu, F Selen Madenoğlu, Dokuz Eylül Üniversitesi, Müh Fak, Endüstri Müh, ve İzmir ; Böl. **2014**. "KRİL SÜRÜSÜ ALGORİTMASI İLE ATÖLYE ÇİZELGELEME (JOB SHOP SCHEDULING WITH KRILL HERD ALGORITHM)". *DEÜ MÜHENDİSLİK FAKÜLTESİ MÜHENDİSLİK BİLİMLERİ DERGİSİ*.

Gong, Wenyin, Zhihua Cai, ve Charles X. Ling. **2011**. "DE/BBO: A hybrid differential evolution with biogeography-based optimization for global numerical optimization". *Soft Computing* 15 (4): 645–65. https://doi.org/10.1007/s00500-010-0591-1.

Goossens, Joshua R., William Mars, Guy Smith, Paul Heil, Scott Braddock, ve Jeanette Pilarski. **2017**. "Durability Analysis of 3-Axis Input to Elastomeric Front Lower Control Arm Vertical Ride Bushing". *SAE Technical Papers* 2017-June (June): 1–5. https://doi.org/10.4271/2017-01-1857.

Guo, Lihong, Gai-Ge Wang, Amir H. Gandomi, Amir H. Alavi, ve Hong Duan. **2014**.

"A new improved krill herd algorithm for global numerical optimization". *Neurocomputing* 138 (Ağustos): 392–402. https://doi.org/10.1016/j.neucom.2014.01.023.

Guo, Pengfei, Xuezhi Wang, ve Yingshi Han. **2011**. "A hybrid genetic algorithm for structural optimization with discrete variables". *Proceedings - 2011 International Conference on Internet Computing and Information Services, ICICIS 2011*, 223–26. https://doi.org/10.1109/ICICIS.2011.64.

Güven, Caner. **2014**. "Kauçuk Burcun Hiperelastik Modellenmesi Ve Şekil Optimizasyonu". Uludağ Üniversitesi.

Güven, Caner, Merve Yavuz Erkek, ve Necmettin Kaya. **2014**. "Kauçuk Burçlarin Şekil Optimizasyonu". *7. Otomotiv Teknolojileri Kongresi*, 1–6. Bursa.

Hafez, Ahmed Ibrahem, Aboul Ella Hassanien, Hossam M. Zawbaa, ve E. Emary. **2015**. "Hybrid Monkey Algorithm with Krill Herd Algorithm optimization for feature selection". *2015 11th International Computer Engineering Conference (ICENCO)*, 273–77. IEEE. https://doi.org/10.1109/ICENCO.2015.7416361.

Heidari-Bateni, Ghobad, ve Clare D Mcgillem. **1994**. "A Chaotic Direct-Sequence Spread-Spectrum Communication System". *IEEE TRANSACTIONS ON COMMUNICATIONS*. C. 42.

Heißing, Bernd, ve Metin Ersoy. **2015**. *Chassis Handbook*. 1st baskı. C. 3. Berlin: Springer. http://repositorio.unan.edu.ni/2986/1/5624.pdf.

Heris, Mostapha Kalami. **2015**. "Implementation of Differential Evolution (DE) in MATLAB". MATLAB Central File Exchange. https://www.mathworks.com/matlabcentral/fileexchange/52897-differential-evolution-de?focused=5322463&tab=function.

Hilborn, Robert C. 1994. *Chaos and nonlinear dynamics : an introduction for scientists and engineers*. Oxford University Press.

Hofmann, Eileen E, A.G.Edward Haskell, John M Klinck, ve Cathy M Lascara. **2004**. "Lagrangian modelling studies of Antarctic krill (Euphausia superba) swarm formation". *ICES Journal of Marine Science* 61 (4): 617–31. https://doi.org/10.1016/j.icesjms.2004.03.028.

Höhn, Reinhard, ve Stephan Hoeppner. **2008**. *Das V-Modell Xt -Grundlagen, Methodik Und Anwendungen*. *Springer Berlin Heidelberg*. Berlin: Springer-Verlag

Berlin and Heidelberg GmbH & Co. K. https://www.bokus.com/bok/9783540302490/das-v-modell-xt/.

Hosseini, Seyyed Mohammad, Mohammad Arjomandi Rad, Abolfazl Khalkhali, ve Mohammad Javad Saranjam. **2019**. "Optimal design of the S-rail family for an automotive platform with novel modifications on the product-family optimization process". *Thin-Walled Structures* 138 (Mayıs): 143–54. https://doi.org/10.1016/j.tws.2019.01.046.

İdris Karen. **2011**. "Taşit Elemanlarinin Optimum Tasarimi İçin Bilgisayar Destekli Analiz Ve Simülasyon Tabanli Bütünleşik Bir Algoritma Geliştirilmesi". Uludağ Üniversitesi. http://hdl.handle.net/11452/10452.

James Kennedy, ve Russell Eberhart. **1995**. "Particle Swarm Optimization". *In: Proceedings of the IEEE international joint conference on neural networks* 4 (6): 1942–48. https://doi.org/10.1109/ICNN.1995.488968.

Jensi, R., ve G. Wiselin Jiji. **2016**. "An improved krill herd algorithm with global exploration capability for solving numerical function optimization problems and its application to data clustering". *Applied Soft Computing* 46 (Eylül): 230–45. https://doi.org/10.1016/j.asoc.2016.04.026.

Jiang, Jie, Guofu Ding, Jian Zhang, Yisheng Zou, ve Shengfeng Qin. **2018**. "A Systematic Optimization Design Method for Complex Mechatronic Products Design and Development". *Mathematical Problems in Engineering* 2018 (Şubat): 1–14. https://doi.org/10.1155/2018/3159637.

Jiang, Peng, Yang Feng, Feng Wu, Shanen Yu, ve Huan Xu. **2016**. "Dynamic Layered Dual-Cluster Heads Routing Algorithm Based on Krill Herd Optimization in UWSNs". *Sensors* 16 (9): 1379. https://doi.org/10.3390/s16091379.

Johanson, Mathias. **2012**. "Information and Communication Support for Automotive Testing and Validation". *New Trends and Developments in Automotive System Engineering*. InTech. https://doi.org/10.5772/12964.

Kaveh, A., R. Sheikholeslami, S. Talatahari, ve M. Keshvari-Ilkhichi. **2014**. "Chaotic swarming of particles: A new method for size optimization of truss structures". *Advances in Engineering Software* 67: 136–47. https://doi.org/10.1016/j.advengsoft.2013.09.006.

Kaveh, Ali. **2014**. *Advances in Metaheuristic Algorithms for Optimal Design of Structures*. *Springer*. London: Springer Cham Heidelberg New York Dordrecht London.

https://doi.org/10.1007/978-3-319-05549-7.

Kaveh, Ali, Ilchi Ghazaan Majid, ve Ilchi Majid Ghazaan. **2018**. *Meta-heuristic Algorithms for Optimal Design of Real-Size Structures. Modeling and Optimization in Science and Technologies 7*. Switzerland: Springer. https://doi.org/10.1007/978-3-319-78780-0.

Kaya, Necmettin. **2014**. "Shape optimization of rubber bushing using differential evolution algorithm". *Scientific World Journal* 2014. https://doi.org/10.1155/2014/379196.

Kesavaraja, D., ve A. Shenbagavalli. **2018**. "QoE enhancement in cloud virtual machine allocation using Eagle strategy of hybrid krill herd optimization". *Journal of Parallel and Distributed Computing* 118 (Ağustos): 267–79. https://doi.org/10.1016/j.jpdc.2017.08.015.

Kohli, Mehak, ve Sankalap Arora. **2018**. "Chaotic grey wolf optimization algorithm for constrained optimization problems". *Journal of Computational Design and Engineering* 5 (4): 458–72. https://doi.org/10.1016/j.jcde.2017.02.005.

Kong, Xiangyu, Sanyang Liu, ve Zhen Wang. **2013**. "A new hybrid artificial bee colony algorithm for global optimization". *International Journal of Computer Science* 10: 287–301. www.IJCSI.org.

Kumar, B. Sravan, M. Suryakalavathi, ve G.V. Nagesh Kumar. **2015**. "Optimization of real power generation plants for power loss minimization and voltage profile improvement using Krill herd algorithm". *2015 Conference on Power, Control, Communication and Computational Technologies for Sustainable Growth (PCCCTSG)*, 117–21. IEEE. https://doi.org/10.1109/PCCCTSG.2015.7503935.

Leung, Yiu Wing, Yuping Wang, Y. W. Leung, ve Y. Wang. **2001**. "An orthogonal genetic algorithm with quantization for global numerical optimization". *IEEE Transactions on Evolutionary Computation* 5 (1): 41–53. https://doi.org/10.1109/4235.910464.

Li, Liangliang, Yongquan Zhou, ve Jian Xie. **2014**. "A Free Search Krill Herd Algorithm for Functions Optimization". *Mathematical Problems in Engineering* 2014: 1–21. https://doi.org/10.1155/2014/936374.

Li, Qiang, Xiaoli Yu, ve Jian Wu. **2018**. "An Improved Genetic Algorithm to Optimize Spatial Locations for Double-Wishbone Type Suspension System with Time Delay".

Mathematical Problems in Engineering 2018 (Şubat): 1–8. https://doi.org/10.1155/2018/6583908.

Li, Shutao, Mingkui Tan, Ivor W. Tsang, ve James Tin Yau Kwok. **2011**. "A hybrid PSO-BFGS strategy for global optimization of multimodal functions". *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 41 (4): 1003–14. https://doi.org/10.1109/TSMCB.2010.2103055.

Li, Tien Yien, ve James A Yorke. **1975**. "Period Three Implies Chaos". *Source: The American Mathematical Monthly*. C. 82.

Li, Yangyang, Licheng Jiao, Peidao Li, ve Bo Wu. **2014**. "A hybrid memetic algorithm for global optimization". *Neurocomputing* 134 (Haziran): 132–39. https://doi.org/10.1016/j.neucom.2012.12.068.

Li Zhaokun, ve Zhang Xianmin. **2018**. *Topology optimization of compliant mechanisms*. Springer. https://doi.org/10.1049/cp:20060808.

Liu, Bo, Ling Wang, Yi Hui Jin, Fang Tang, ve De Xian Huang. **2005**. "Improved particle swarm optimization combined with chaos". *Chaos, Solitons and Fractals* 25 (5): 1261–71. https://doi.org/10.1016/j.chaos.2004.11.095.

Liu, Shengsong, ve Zhijian Hou. **2002**. "Weighted Gradient Direction Based Chaos Optimization Algorithm for Nonlinear Programming Problem". *Proceedings of the 4th world congress on intelligent control and automation*.

Long, Wen, Ximing Liang, Yafei Huang, ve Yixiong Chen. **2014**. "An effective hybrid cuckoo search algorithm for constrained global optimization". *Neural Computing and Applications* 25 (3–4): 911–26. https://doi.org/10.1007/s00521-014-1577-1.

May, Rober M. 1976. "Simple mathematical models with very complicated dynamics". *Nature* 162: 459–67.

Mehmet Ali ÖZCAN. **2016**. "Kauçuk Malzemelerde Hasar Analizi". İstanbul Teknik Üniversitesi.

Mingjun, Ji, ve Tang Huanwen. **2004**. "Application of chaos in simulated annealing". *Chaos, Solitons and Fractals* 21 (4): 933–41. https://doi.org/10.1016/j.chaos.2003.12.032.

Mirjalili, Seyedali. **2021**. "The Genetic Algorithm (GA) : Selection + Crossover + Mutation + Elitism". MATLAB Central File Exchange. 17 Ağustos 2021. https://www.mathworks.com/matlabcentral/fileexchange/67435-the-genetic-algorithm-

ga-selection-crossover-mutation-elitism?s_tid=prof_contriblnk.

Nabaei, Armin, Melika Hamian, Mohammad Reza Parsaei, Reza Safdari, Taha Samad-Soltani, Houman Zarrabi, ve A. Ghassemi. **2018**. "Topologies and performance of intelligent algorithms: a comprehensive review". *Artificial Intelligence Review*. Springer Netherlands. https://doi.org/10.1007/s10462-016-9517-3.

Peitgen, Heinz-Otto, H. (Hartmut) Jürgens, ve Dietmar Saupe. **2004**. *Chaos and fractals : new frontiers of science*. Springer.

Pfeffer, Raphael, Gustav N. Basedow, Nina R. Thiesen, Markus Spadinger, Albert Albers, ve Eric Sax. **2019**. "Automated driving-challenges for the automotive industry in product development with focus on process models and organizational structure". *SysCon 2019 - 13th Annual IEEE International Systems Conference, Proceedings*. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/SYSCON.2019.8836779.

Price, Holly J. 1989. "Swimming behavior of krill in response to algal patches: A mesocosm study". *Limnology and Oceanography* 34 (4): 649–59. https://doi.org/10.4319/lo.1989.34.4.0649.

Price, Kenneth, Rainer M Storn, ve Jouni A Lampinen. **2005**. *Differential Evolution. A Practical Approach to Global Optimization*. *Natural Computing Series*. https://www.springer.com/gp/book/9783540209508.

Rani, R. Ranjani, ve D. Ramyachitra. **2017**. "Krill Herd Optimization algorithm for cancer feature selection and random forest technique for classification". *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, 109–13. IEEE. https://doi.org/10.1109/ICSESS.2017.8342875.

Rezaee Jordehi, A. 2014. "A chaotic-based big bang–big crunch algorithm for solving global optimisation problems". *Neural Computing and Applications* 25 (6): 1329–35. https://doi.org/10.1007/s00521-014-1613-1.

Secui, Dinu Calin. **2016**. "A modified Symbiotic Organisms Search algorithm for large scale economic dispatch problem with valve-point effects". *Energy* 113 (Ekim): 366–84. https://doi.org/10.1016/j.energy.2016.07.056.

Sheikholeslami, Razi, ve Ali Kaveh. **2013**. "A survey of chaos embedded metaheuristic algorithms". *INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING Int. J. Optim. Civil Eng*. C. 3.

https://www.researchgate.net/publication/258641935.

Shelokar, P S, Patrick Siarry, V K Jayaraman, ve B D Kulkarni. **2007**. "Particle swarm and ant colony algorithms hybridized for improved continuous optimization". *Applied Mathematics and Computation* 188: 129–42. https://doi.org/10.1016/j.amc.2006.09.098.

Shi, Yang, Hongcheng Liu, Liang Gao, ve Guohui Zhang. **2010**. "Cellular particle swarm optimization". *INFORMATION SCIENCES*. https://doi.org/10.1016/j.ins.2010.05.025.

Simulia, Dassault Systemes. **2014**. "Abaqus Analysis User's Guide". *Abaqus Documentation*. Dassault Systemes - Simulia.

Storn, Rainer M, ve Kenneth Price. **1996**. "Minimizing the real functions of the ICEC'96 contest by differential evolution". *Proceedings of the IEEE Conference on Evolutionary Computation*, 842–44. IEEE. https://doi.org/10.1109/icec.1996.542711.

Strumberger, Ivana, Nebojsa Bacanin, Milan Tuba, Ivana Stromberger, Nebojsa Bacanin, ve Milan Tuba. **2017**. "Hybridized krill herd algorithm for large-scale optimization problems". *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 000473–78. IEEE. https://doi.org/10.1109/SAMI.2017.7880356.

Sun, Shuangcheng, Hong Qi, Fangzhou Zhao, Liming Ruan, ve Bingxi Li. **2016**. "Inverse geometry design of two-dimensional complex radiative enclosures using krill herd optimization algorithm". *Applied Thermal Engineering* 98 (Nisan): 1104–15. https://doi.org/10.1016/j.applthermaleng.2016.01.017.

Talatahari, S., A. Kaveh, ve R. Sheikholeslami. **2011**. "An Efficient Charged System Search Using Chaos for Global Optimization Problems". *International Journal of Optimization in Civil Engineering* 1 (2): 305–25.

Talatahari, S., A. Kaveh, ve R. Sheikholeslami. **2012**. "Engineering design optimization using chaotic enhanced charged system search algorithms". *Acta Mechanica* 223 (10): 2269–85. https://doi.org/10.1007/s00707-012-0704-2.

Talatahari, Siamak, Ali Kaveh, ve Razi Sheikholeslami. **2012**. "Chaotic imperialist competitive algorithm for optimum design of truss structures". *Structural and Multidisciplinary Optimization* 46 (3): 355–67. https://doi.org/10.1007/s00158-011-0754-4.

Tang, Hesheng, Songtao Xue, ve Cunxin Fan. **2008**. "Differential evolution strategy for structural system identification". *Computers and Structures* 86 (21–22): 2004–12. https://doi.org/10.1016/j.compstruc.2008.05.001.

Tatsumi, Keiji, Yoshinori Obita, ve Tetsuzo Tanino. **2009**. "Chaos generator exploiting a gradient model with sinusoidal perturbations for global optimization". *Chaos, Solitons and Fractals* 42 (3): 1705–23. https://doi.org/10.1016/j.chaos.2009.03.088.

Tavazoei, Mohammad Saleh, ve Mohammad Haeri. **2007**. "Comparison of Different One-dimensional Maps As Chaotic Search Pattern in Chaos Optimization Algorithms". *Appl. Math. Comput.* 187 (2): 1076–85. https://doi.org/10.1016/j.amc.2006.09.087.

Tian, Ying, ve Pingyu Jiang. **2007**. "Optimization of tool motion trajectories for pocket milling using a chaos ant colony algorithm". *Proceedings of 2007 10th IEEE International Conference on Computer Aided Design and Computer Graphics, CAD/Graphics 2007*, 389–94. https://doi.org/10.1109/CADCG.2007.4407914.

Tien, Jia-Ping, ve Tzuu-Hseng S Li. **2013**. "Hybrid Taguchi-Chaos Of Artificial Bee Colony Algorithm For Global Numerical Optimization". *International Journal of Innovative Computing, Information and Control ICIC* 9 (6): 2665–88.

Ting, T. O., Xin She Yang, Shi Cheng, ve Kaizhu Huang. **2015**. "Hybrid metaheuristic algorithms: Past, present, and future". *Recent Advances in Swarm Intelligence and Evolutionary Computation* 585: 71–83. https://doi.org/10.1007/978-3-319-13826-8_4.

Tsai, Jinn Tsong, Tung Kuan Liu, ve Jyh Horng Chou. **2004**. "Hybrid Taguchi-genetic algorithm for global numerical optimization". *IEEE Transactions on Evolutionary Computation* 8 (4): 365–77. https://doi.org/10.1109/TEVC.2004.826895.

Vafashoar, R., M. R. Meybodi, ve A. H. Momeni Azandaryani. **2012**. "CLA-DE: A hybrid model based on cellular learning automata for numerical optimization". *Applied Intelligence* 36 (3): 735–48. https://doi.org/10.1007/s10489-011-0292-1.

Vincylloyd, F., ve B. Anand. **2015**. "A Double Herd Krill Based Algorithm for Location Area Optimization in Mobile Wireless Cellular Network". *The Scientific World Journal* 2015: 1–9. https://doi.org/10.1155/2015/475806.

Volz, Karlheinz Holger. **2011**. "Physikalisch begründete Ersatzmodelle für die Crashoptimierung von Karosseriestrukturen in frühen Projektphasen". Technische Universität München, Germany.

Wang, Gai-Ge, Amir H. Gandomi, Amir H. Alavi, ve Suash Deb. **2016**. "A Multi-Stage Krill Herd Algorithm for Global Numerical Optimization". *International Journal on Artificial Intelligence Tools* 25 (02): 1550030. https://doi.org/10.1142/S021821301550030X.

Wang, Gai-Ge, Lihong Guo, Amir Hossein Gandomi, Amir Hossein Alavi, ve Hong Duan. **2013**. "Simulated Annealing-Based Krill Herd Algorithm for Global Optimization". *Abstract and Applied Analysis* 2013: 1–11. https://doi.org/10.1155/2013/213853.

Wang, Gai-Ge, Amir Hossein Gandomi, ve Amir Hossein Alavi. **2013**. "A chaotic particle-swarm krill herd algorithm for global numerical optimization". *Kybernetes* 42 (6): 962–78. https://doi.org/10.1108/K-11-2012-0108.

Wang, Gai Ge, Lihong Guo, Amir H. Gandomi, Guo Sheng Hao, ve Heqi Wang. **2014**. "Chaotic Krill Herd algorithm". *Information Sciences* 274 (Ağustos): 17–34. https://doi.org/10.1016/j.ins.2014.02.123.

Wang, Gaige, Lihong Guo, Heqi Wang, Hong Duan, Luo Liu, ve Jiang Li. **2014**. "Incorporating mutation scheme into krill herd algorithm for global numerical optimization". *Neural Computing and Applications* 24 (3–4): 853–71. https://doi.org/10.1007/s00521-012-1304-8.

Wang, Jin. **2012**. "A hybrid particle swarm optimization for numerical optimization". *International Journal of Advancements in Computing Technology* 4 (20): 190–96. https://doi.org/10.4156/ijact.vol4.issue20.23.

Wang, Li, Pengfei Jia, Tailai Huang, Shukai Duan, Jia Yan, ve Lidan Wang. **2016**. "A Novel Optimization Technique to Improve Gas Recognition by Electronic Noses Based on the Enhanced Krill Herd Algorithm". *Sensors* 16 (8): 1275. https://doi.org/10.3390/s16081275.

Weber, Julian. **2013**. *Automotive Development Processes. Integrated Computer-Aided Design in Automotive Development*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-11940-8_1.

Wu, Bin, ve Shu Hai Fan. **2011**. "Improved artificial bee colony algorithm with chaos". *Communications in Computer and Information Science*, 158 CCIS:51–56. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-22694-6_8.

Wu, Chi, Yunkai Gao, Jianguang Fang, Erik Lund, ve Qing Li. **2017**. "Discrete

topology optimization of ply orientation for a carbon fiber reinforced plastic (CFRP) laminate vehicle door". *Materials and Design* 128 (Ağustos): 9–19. https://doi.org/10.1016/j.matdes.2017.04.089.

Xiang, Wanli, Shoufeng Ma, ve Meiqing An. **2014**. "HABCDE: A hybrid evolutionary algorithm based on artificial bee colony algorithm and differential evolution". *Applied Mathematics and Computation* 238 (Temmuz): 370–86. https://doi.org/10.1016/j.amc.2014.03.055.

Yan, Jingfeng, Chaofeng Guo, ve Wenyin Gong. **2011**. "Hybrid differential evolution with convex mutation". *Journal of Software* 6 (11 SPEC. ISSUE): 2321–28. https://doi.org/10.4304/jsw.6.11.2321-2328.

Yang, Dixiong, Zhenjun Liu, ve Jilei Zhou. **2014**. "Chaos optimization algorithms based on chaotic maps with different probability distribution and search speed for global optimization". *Communications in Nonlinear Science and Numerical Simulation* 19 (4): 1229–46. https://doi.org/10.1016/j.cnsns.2013.08.017.

Yıldız, Ali Rıza. **2017**. "Taşıt Elemanlarının Yapısal Optimizasyon Teknikleri ile Optimum Tasarımı Optimum Design of Vehicle Components Using Structural Optimization Techniques". *Journal of Polytechnic* 20 (2): 319–23. https://doi.org/10.2339/2017.20.2.

Yuan, Xiaofang, Jingyi Zhao, Yimin Yang, ve Yaonan Wang. **2014**. "Hybrid parallel chaos optimization algorithm with harmony search algorithm". *Applied Soft Computing Journal* 17 (Nisan): 12–22. https://doi.org/10.1016/j.asoc.2013.12.016.

Yuan, Xiaohui, Bo Cao, Bo Yang, ve Yanbin Yuan. **2008**. "Hydrothermal scheduling using chaotic hybrid differential evolution". *Energy Conversion and Management* 49 (12): 3627–33. https://doi.org/10.1016/j.enconman.2008.07.008.

Zhang, Jie, Yajuan Yang, ve Quanju Zhang. **2009**. "The particle swarm optimization algorithm based on dynamic chaotic perturbations and its application to K-means". *CIS 2009 - 2009 International Conference on Computational Intelligence and Security*, 1:282–86. https://doi.org/10.1109/CIS.2009.111.

Zhang, Weihong, Jihong Zhu, ve Tong Gao. **2016**. *Topology optimization in engineering structure design*.

Zhang, Zhiyong, Yibo Zhang, Caixia Huang, ve Xin Liu. **2018**. "Low-noise structure optimization of a heavy commercial vehicle cab based on approximation model". *Journal*

of Low Frequency Noise Vibration and Active Control 37 (4): 987–1002. https://doi.org/10.1177/1461348418798403.

Zhihui, Li, Cao Qian, Zhao Yonghua, Tao Pengfei, ve Zhuo Rui. **2019**. "Krill Herd Algorithm for Signal Optimization of Cooperative Control With Traffic Supply and Demand". *IEEE Access* 7: 10776–86. https://doi.org/10.1109/ACCESS.2019.2891791.