



T.C.  
NECMETTİN ERBAKAN  
ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ



**DÖNÜŞTÜRÜLMÜŞ DAĞILIMLARDA  
META-SEZGİSEL YAKLAŞIMLAR İLE  
PARAMETRE TAHMİNİ**

Shuaib Mursal **İBRAHİM**

**YÜKSEK LİSANS TEZİ**

**İstatistik Anabilim Dalı**

**Temmuz-2024  
KONYA  
Her Hakkı Saklıdır**

## TEZ KABUL VE ONAYI

Shuaib Mursal **İBRAHİM** tarafından hazırlanan “**Dönüştürülmüş Dağılımlarda Meta-Sezgisel Yaklaşımlar İle Parametre Tahmini**” adlı tez çalışması 04/07/2024 tarihinde aşağıdaki jüri tarafından oy birliği ile Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü İstatistik Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

### Jüri Üyeleri

#### Başkan

Doç. Dr. Yunus AKDOĞAN

#### Danışman

Doç. Dr. Aydın KARAKOCA

#### Üye

Prof. Dr. Murat ERİŞOĞLU

### İmza

.....

.....

.....

Fen Bilimleri Enstitüsü Yönetim Kurulu’nun .../.../20.. gün ve ..... sayılı kararıyla onaylanmıştır.

Prof. Dr. Şerife Yurdagül KUMCU  
FBE Müdürü

## TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

## DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

İmza

Shuaib Mursal İBRAHİM

Tarih:

## ÖZET

### YÜKSEK LİSANS TEZİ

## DÖNÜŞTÜRÜLMÜŞ DAĞILIMLARDA META-SEZGİSEL YAKLAŞIMLAR İLE PARAMETRE TAHMİNİ

Shuaib Mursal İBRAHİM

Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü  
İstatistik Anabilim Dalı

Danışman: Doç. Dr. Aydın KARAKOCA

2024, 87 Sayfa

Jüri

Doç. Dr. Aydın KARAKOCA  
Prof. Dr. Murat ERİŞOĞLU  
Doç. Dr. Yunus AKDOĞAN

Olasılık dağılımları, biyoloji, ekonomi, mühendislik, tıp ve çevre bilimleri dahil olmak üzere çeşitli alanlardaki rastgele olayların modellenmesinde çok önemli bir rol oynar. Ancak mevcut dağılımlar her zaman gerçek dünya verilerine en iyi uyumu sağlayamayabilir. Bu gibi durumlarda, daha iyi bir uyum elde etmek için bazı dönüşümlerin mevcut dağılımlara uygulanması gerekli hale gelir. Bu dönüşümlerden biri de Shaw Buckley (2009) tarafından önerilen karesel dönüşüm yöntemidir. Bu çalışmada karesel dönüşüm yöntemiyle elde edilen dönüştürülmüş üstel, Weibull ve Fréchet dağılımlarının parametre tahmini için meta-sezgisel optimizasyon algoritmaları incelenmiştir.

Dönüştürülmüş dağılımların parametrelerini tahmin etmek için Genetik Algoritma (GA), Parçacık Sürü Optimizasyonu (PSO), Yapay Arı Kolonisi (ABC) optimizasyonu ve Diferansiyel Evrim (DE) algoritmaları incelenmiştir. Çalışmada meta sezgisel algoritmalarla elde edilen parametre tahminleri En Çok Olabilirlik (EÇO) tahminlerini elde etmede sıklıkla kullanılan iteratif Newton-Raphson (NR) algoritması ile karşılaştırılmıştır.

Bu algoritmaların performansını farklı dönüşüm parametre değerleri ve örneklem büyüklüklerini içeren çeşitli durumlar altında değerlendirmek için kapsamlı bir simülasyon çalışması yapılmıştır. Simülasyon çalışmasında parametre tahmin performansları yan, Mutlak Hata Yüzde Ortalaması (MAPE), Hata Kareler Ortalaması (MSE) ve log-olabilirlik değerleri kullanılarak karşılaştırılmıştır. Dönüştürülmüş dağılımlarının gerçek veri uygulamalarının performansı log olabilirlik, Akaike Bilgi Kriteri (AIC), Bayes Bilgi Kriteri (BIC) ve Kolmogorov-Smirnov (KS) istatistikleri ile değerlendirilmiştir.

Simülasyon ve gerçek veri uygulamaları sonucunda meta sezgisel algoritmaların dönüştürülmüş dağılımlarda parametre tahminleri için oldukça başarılı olduğunu göstermiştir.

**Anahtar Kelimeler:** Dönüştürülmüş Dağılımlar, Genetik Algoritma, Meta-sezgisel Algoritmalar, Parametre Tahmini, Simülasyon, Yapay Arı Kolonisi

## ABSTRACT

## MS THESIS

# METAHEURISTIC APPROACHES TO PARAMETER ESTIMATION IN TRANSMUTED DISTRIBUTIONS

Shuaib Mursal İBRAHİM

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF  
NECMETTİN ERBAKAN UNIVERSITY  
THE DEGREE OF MASTER OF SCIENCE IN STATISTICS

Advisor: Assoc. Prof. Dr. Aydın KARAKOCA

2024, 87 Pages

### Jury

Assoc. Prof. Dr. Aydın KARAKOCA

Prof. Dr. Murat ERİŞOĞLU

Assoc. Doç. Dr. Yunus AKDOĞAN

Probability distributions play a crucial role in modeling random phenomena across various fields, including biology, economics, engineering, medicine, and environmental sciences. However, existing distributions may not always provide an optimal fit to real-world data. In such cases, applying transformations to existing distributions becomes necessary to achieve a better fit. One such transformation is the square transformation method proposed by Shaw Buckley (2009). This study investigates metaheuristic optimization algorithms for parameter estimation of transformed exponential, Weibull, and Fréchet distributions obtained using the square transformation method.

Four metaheuristic algorithms are employed to estimate the parameters of these transmuted distributions: Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC) optimization, and Differential Evolution (DE). Additionally, the Maximum Likelihood Estimation (MLE) method utilizing the Newton-Raphson (NR) algorithm is used for comparison.

A comprehensive simulation study is conducted to evaluate the performance of these algorithms under various conditions involving different transmutation parameter values and sample sizes. The performance is assessed using four key criteria: bias, Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE) and log-likelihood for simulation study. For the real-world dataset applications, Kolmogorov-Smirnov test is used as a goodness-of-fit measure, while Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC) are used for model selection.

The results from both the simulation study and real-world data applications demonstrated that metaheuristic algorithms are highly successful in estimating parameters for transmuted distributions.

**Keywords:** Artificial Bee Colony, Genetic Algorithm, Meta-heuristic Algorithms, Parameter Estimation, Simulation, Transmuted Distributions

## ÖNSÖZ

Tüm bu süreçte desteğini esirgemeyen danışmanım Doç. Dr. Aydın KARAKOCA'ya çok teşekkür ediyorum. Değerli rehberliği, derinlemesine geri bildirimleri ve sürekli teşviki, bu tezi şekillendirmede önemli bir rol oynadı. Akademik gelişimime olan özverisi, deneyimlemek gerçek bir ayrıcalıktı.

Prof. Dr. Murat ERİŞOĞLU'na ve Necmettin Erbakan Üniversitesi İstatistik Bölümü öğretim üyelerine özellikle teşekkür ediyorum. Uzmanlıkları ve bilgilerini paylaşma isteklilikleri paha biçilmez kaynaklardı. Onların tavsiyeleri ve desteklerinin yanı sıra gerekli materyallerin sağlanması da bu araştırmanın geliştirilmesine önemli ölçüde katkıda bulunmuştur.

Bu başarının temelindeki sonsuz sevgi ve bitmeyen destekleri için aileme en derin şükranlarımı sunarım.

Son olarak, Türkiye Burslarına bu yolculuğu mümkün kılan mali yardımları için minnettarım.

Shuaib Mursal İBRAHİM  
KONYA-2024

## İÇİNDEKİLER

ÖZET .....	iv
ABSTRACT.....	v
ÖNSÖZ .....	vi
İÇİNDEKİLER.....	vii
1. GİRİŞ .....	1
2. KAYNAK ARAŞTIRMASI .....	5
3. META-SEZGİSEL ALGORİTMALAR .....	9
3.1 Genetik Algoritma (GA).....	9
3.1.1 Bir Örnek .....	11
3.2 Parçacık Sürü Optimizasyonu (PSO).....	16
3.2.1 Bir Örnek .....	17
3.3 Yapay Arı Kolonisi (ABC).....	22
3.3.1 Bir Örnek .....	25
3.4 Diferansiyel Gelişim (DE).....	30
3.4.1 Bir Örnek .....	32
4. DÖNÜŞTÜRÜLMÜŞ DAĞILIMLAR.....	36
4.1 Dönüştürülmüş Düzgün Dağılım .....	39
4.2 Dönüştürülmüş Üstel Dağılım .....	40
4.3 Dönüştürülmüş Weibull dağılım.....	42
4.4 Dönüştürülmüş Fréchet Dağılım.....	45
5. SİMÜLASYON VE BULGULAR.....	49
5.1 Dönüştürülmüş Üstel Dağılım .....	52
5.2 Dönüştürülmüş Weibull Dağılımı.....	58
5.3 Dönüştürülmüş Fréchet Dağılımı.....	63
6. UYGULAMALAR.....	68
6.1 Dönüştürülmüş Üstel Dağılım .....	69
6.2 Dönüştürülmüş Weibull Dağılımı.....	71
6.3 Dönüştürülmüş Fréchet Dağılımı.....	73
7. SONUÇLAR VE ÖNERİLER.....	75
KAYNAKLAR .....	77
ÖZGEÇMİŞ .....	80

## 1. GİRİŞ

Olasılık dağılımları, biyoloji, ekonomi, mühendislik ve çevre bilimleri gibi çok çeşitli alanlarda olayların ve süreçlerin rastgeleliğini modellemek için kullanılır. Problemin niteliğine ve mevcut verilere uyacak şekilde seçilmiş farklı olasılık dağılımları mevcuttur. Bununla birlikte, mevcut dağılımların eldeki verileri yeterince temsil etmediği durumlarla da karşılaşılabilir. Bu gibi durumlarda mevcut dağılımlar üzerinde farklı dönüştürme teknikleri kullanılarak yeni dağılımlar elde edilebilmektedir. Örneğin Weibull (1951), üstel olarak dağılan rastgele değişkenlerin kuvvetlerini alarak Weibull dağılımını elde etti. Marshall (1997) ise Weibull, Gama ve lognormal dağılımlarına alternatif olarak üstel dağılıma yeni bir parametre ekleyerek geliştirilmiş üstel dağılımı önerdi.

Gupta vd. (1998)  $F(t)$  dağılım fonksiyonuna  $\theta > 0$  olmak üzere  $F(t)^\theta$  formundaki dönüşümle özellikle yaşam zamanı verilerinin modellenmesinde kullanılan dağılımları önerdiler. Bu yaklaşımın hem monoton hem de monoton olmayan başarısızlık oranlarını modelleyebildiğini de gösterdiler.

Eugene vd. (2002) beta rastgele değişkenin lojit dönüşümü ile oluşturulan genel bir dağılım sınıfını tanıttı. Bu sınıfın özel bir hali olan beta normal dağılımının simetrik ağır kuyruklu dağılımların yanı sıra, çarpık ve iki modlu dağılımlar da dahil olmak üzere çeşitli dağılım türlerinin modellenmesinde daha fazla esneklik sunduğunu gösterdiler. Bunu, çarpıklığı ortaya çıkaran ve kuyruk ağırlığı değişimine izin veren iki parametre kullanarak başardılar. Benzer şekilde Cordeiro ve Castro (2011) geliştirilmiş bir Kumaraswamy dağılım ailesi önerdi. Çalışmalarında çarpıklığı ortaya çıkarmak ve dağılımlar için kuyruk ağırlıklarını ayarlamak için iki parametreyi birleştirdiler.

Shaw ve Buckley (2009), finansal verilerin modellenmesinde yetersiz kalan dağılımları daha esnek hale getirmek için karesel dönüştürülmüş dağılım (QT-G)

yöntemini önerdiler. Önerilen yöntem, paralel ve seri sistemlerin yaşam zamanlarının konveks bileşimlerine dayanarak geliştirilmiştir.

QT-G yöntemi ile geliştirilen birçok yeni dağılım araştırmacının ilgisini çekmiştir (Aryal ve Tsokos, 2009, 2011). Bu teknik, belirli bir temel dağılıma karşılık gelen dönüştürülmüş dağılımı bulmak için uygulanmıştır. QT-G dağılım ailesi hakkında daha fazla ayrıntı Bölüm 2 ve 4'te bulunabilir.

İki modlu bozulma oranları gibi daha karmaşık verileri modellemek için araştırmacılar dönüştürülmüş dağılımların esnekliğini artıran kübik rank dönüşüm yöntemini (CRTM) önerdiler. Örneğin Granzotto vd. (2017), bozulma oranı, yaşam fonksiyonları, olasılık yoğunluk ve dağılım fonksiyonları elde ederek kübik rank dönüşümü ile dönüştürülmüş Weibull dağılımını ve kübik rank dönüştürülmüş log-lojistik dağılımını tanıttı. Çelik (2018), kübik rank dönüşümünü kullanarak kübik rank dönüştürülmüş Gumble ve Frechet dağılımlarını tanıttı. Bu dağılımların yoğunluk fonksiyonları gibi bazı istatistiksel yönlerinin yanı sıra bozulma ve yaşam fonksiyonlarını da inceledi. Saraçoğlu ve Tanış (2018), kübik rank dönüşümünü kullanarak kübik rank dönüştürülmüş Kumaraswamy dağılımını tanıttılar. Önerilen dağılımın bozulma oranı, momentler, varyans, çarpıklık ve basıklık katsayıları, moment çıkaran fonksiyon ve sıra istatistikleri gibi çeşitli özelliklerini incelediler. Tanış ve Saraçoğlu (2023) geliştirilmiş Gompertz dağılımının kübik rank dönüşümünü geliştirdi. Önerilen dağılımın bozulma oranı, özel momentler, çarpıklık katsayısı ve basıklık katsayısı gibi bazı karakteristik özelliklerini incelediler.

İstatistiğin temel görevlerinden biri, örnek verilere dayanarak popülasyon parametrelerini tahmin etmektedir. Bunu elde etmek için kullanılan popüler tekniklerden biri En Çok Olabilirlik (EÇO) tahminidir. EÇO verilerin belirli bir olasılık dağılımından geldiği varsayımıyla çalışır. Daha sonra bu dağılımdan çekilen verilerin gözlemlenme

olasılığını en yüksek yapan parametre değerlerini araştırır. EÇÖ tahminlerinin hesaplanması, sürekli ve türevlenebilir bir amaç fonksiyonu gerektirir. Olabilirlik fonksiyonu adı verilen bu fonksiyon, kısmi türevlerinin sıfıra eşitlenmesi ve bilinmeyen parametrelerin çözülmesi yoluyla optimize edilmektedir. Çözümler analitik olarak veya Newton-Raphson (NR) veya Yinelemeli Yeniden Ağırlıklandırılmış En Küçük Kareler gibi algoritmalar kullanılarak yinelemeli olarak bulunabilir. Ancak bu yinelemeli algoritmaların seçilen başlangıç parametre değerlerine duyarlı olabileceğini unutmamak önemlidir. Yanlış başlangıç noktasının seçimi, global optimal çözümü bulmak yerine yerel minimum/maksimumlara takılma riskini barındırır (Abbasi vd., 2006).

Geleneksel tahmin yöntemleri, dönüştürülmüş dağılımların parametrelerini doğru bir şekilde tahmin etmek için uygun olmayabilir. Özellikle en sık kullanılan parametre tahmin yöntemlerinden olan EÇÖ yönteminde parametre tahminleri analitik olarak elde edilemediğinde tekrarlamalı (iteratif) yöntemler kullanılarak parametre tahminleri elde edilebilmektedir.

Meta-sezgisel algoritmalar genel olarak doğada var olan durumlardan esinlenerek geliştirilmiş algoritmalar, örneğin; yapay arı kolonilerinin davranışı veya diferansiyel gelişim gibi doğal fenomenlerden esinlenir. Bu algoritmalar, soruna daha iyi çözümler bulmak için yinelemeli olarak çalışır. Meta-sezgisel algoritmalar, olasılık fonksiyonunun karmaşık olduğu durumlarda bile parametre tahmin problemi için iyi çözümler bulabilir. Bu özelliği meta-sezgisel algoritmaları dönüştürülmüş dağılımlarda parametre tahmini için kullanışlı bir araç haline getirir. Bu çalışmada parametre tahmini için meta-sezgisel algoritmaların dönüştürülmüş dağılımlarda özellikle parametre tahminlerinin analitik olarak elde edilemediği durumlarda çözümlere alternatif olarak kullanımı incelenecektir

İstatistiğin temel problemlerinden biri olan parametre tahmini problemine özellikle dönüştürülmüş dağılımlar üzerinden alternatif bir yaklaşım olarak meta-sezgisel

algoritmalar ile çözüm bulmak tezin temel amacını oluşturmaktadır. Özellikle gerçek dünya verilerinin uygun istatistiksel model kullanılarak modellenmesinde parametre tahmini probleminin çözümünün doğruluğu ve güvenilirliği için meta-sezgisel yaklaşımlar araştırmacılara bir alternatif oluşturacaktır.

Yukarıdaki amaca ulaşmak için, çalışmanın aşağıdaki spesifik hedefleri vardır:

1. Mevcut bazı dönüştürülmüş dağılımlara ve özelliklerine genel bir bakış sunmak.
2. Meta-sezgisel yaklaşımlar günümüzde birçok problemin çözümünde kullanılmasına rağmen dönüştürülmüş dağılımlarda parametre tahmininde kullanımını literatüre kazandırmak.
3. Önerilen meta-sezgisel yaklaşımın performansını geleneksel tahmin yöntemlerine karşı deneysel olarak değerlendirmek.

Meta-sezgisel yaklaşımlar günümüzde birçok problemin çözümünde kullanılmasına rağmen dönüştürülmüş dağılımlarda parametre tahmininde kullanımı ilk kez bu çalışmada tanıtılacaktır. Meta-sezgisel algoritmalar, ilk olarak parametre uzayının dışbükey olmaması ve verilerde aykırı değerlerin varlığı nedeniyle zorlu bir problem olan dönüştürülmüş dağılımlarda parametre tahmininin doğruluğunu ve verimliliğini önemli ölçüde artırma potansiyeline sahiptir. İkincisi, önerilen meta-sezgisel yaklaşım, dönüştürülmüş dağılımlarda parametre tahmininin özel zorluklarına göre uyarlanacaktır. Üçüncüsü, dönüştürülmüş dağılımlarda parametre tahmini için bir meta-sezgisel yaklaşımın geliştirilmesi, finansal modelleme, tıbbi çalışmalar, mühendislik ve üretim gibi çeşitli alanlarda bir dizi önemli uygulama sahası bulacaktır. Bu, daha iyi karar vermeye, geliştirilmiş ürünlere ve hizmetlere yol açabilir.

## 2. KAYNAK ARAŞTIRMASI

Dönüştürülmüş dağılımlar, bir dağılımın birikimli dağılım fonksiyonu üzerinde uygulanan dönüşüm sonucunda oluşturur. Farklı dönüştürme yöntemleri kullanılarak bir çok yeni dağılım elde edilebilmektedir. Shaw ve Buckley (2009) Karesel Dönüştürülmüş Dağılımlar Ailesi (Quadratic Transmuted Family – QT-G) yöntemi ilk defa bir finans matematiği problemini çözmek için önermişlerdir. Dağılıma yeni bir parametre ekleyerek çarpıklık veya basıklık kontrolü imkânı sunan yöntemi dönüştürülmüş dağılımlar elde etmek için kullanmışlardır.

Aryal ve Tsokos (2009), daha esnek dağılımlar oluşturmak için karesel rank dönüşümünü (QRTM) kullanarak uç değer dağılımını temel dağılım olarak kullanmıştır. Gumbel dağılımı üzerinden bir vaka çalışması yürüterek bazı istatistiksel ölçümleri türetmiş ve NR algoritmasını kullanan EÇO yöntemi ile parametreleri tahmin etmişlerdir.

Aryal ve Tsokos (2011), iki parametrelili Weibull dağılımını temel alarak karesel dönüşümü kullanarak dönüştürülmüş Weibull dağılımını önermişlerdir. Momentleri ve kantilleri türetmiş, sıra istatistiklerini incelemişlerdir. Ayrıca bilinmeyen dağılım parametrelerini EÇO yöntemi ile NR algoritması kullanarak elde etmişlerdir.

Ashour ve Eltehiwy (2013), karesel rank dönüşümünü kullanarak Lomax dağılımının bir geliştirilmesini önermişlerdir. Bu geliştirmeye Dönüştürülmüş Lomax Dağılımı adı verilmiştir. Önerilen dağılımın momentleri, kantilleri ve ortalama sapmalar gibi farklı istatistiksel özellikleri türetilmiştir. Dağılımın parametre tahminlerini EÇO yöntemi ile elde etmişlerdir.

Khan ve King (2013), Weibull dağılımının yeni bir geliştirilmesini önermişler ve buna Düzeltilmiş Dönüştürülmüş Weibull Dağılımı adını vermişlerdir. Yeni dağılımı üç parametrelili Weibull Dağılımından karesel rank dönüşümünü kullanarak oluşturmuşlardır. Önerilen dağılımın momentler, kantiller ve sıra istatistikleri gibi bazı

özelliklerini incelemişlerdir. Önerilen dağılımın bilinmeyen parametrelerini tahmin etmek için EÇO yöntemini kullanmışlardır.

Mahmoud ve Mandouh (2013), temel dağılımdan daha esnek olan dönüştürülmüş Fréchet dağılımını oluşturmak için karesel rank dönüşümünü kullanmışlardır. Dağılımın momentler, kantiller, sıra istatistikleri ve diğer matematiksel özelliklerini incelemişlerdir. Dağılım parametrelerini tahmin etmek için EÇO ve Bayes tahmini yöntemlerini kullanmışlardır.

Abdul-Moniem ve Seham (2015), Gompertz dağılımının "Dönüştürülmüş Gompertz Dağılımı" adında yeni bir genelleştirmesini sunmuşlardır. Momentler, TL-momentler ve sıra istatistikleri gibi bazı istatistiksel özelliklerini türetmişlerdir. Bilinmeyen parametreleri tahmin etmek için EÇO tahmini yöntemini kullanmışlardır. Son olarak, yeni modelin uygunluğunu test etmek için gerçek veri seti üzerinde uygulayarak önerilen modelin Gompertz dağılımından daha etkili olduğunu belirtmişlerdir.

Pobočková vd. (2018), dönüştürülmüş Weibull dağılımı adında yeni bir dağılım önermiştir. Momentler ve kantiller gibi bazı özelliklerini türetmiş ve modeli iki veri kümesine uygulamıştır. Sonuçlarını 2 ve 3 parametrelili Weibull dağılımlarıyla karşılaştırmış ve parametre tahmini için EÇO tahminini kullanmıştır. Sonuçlar, dönüştürülmüş Weibull dağılımının temel dağılımlardan daha esnek olduğunu göstermiştir.

Liu vd. (2011), Tayvan Boğazı'ndaki bir rüzgâr çiftliğinde 2006-2008 yılları arasında rüzgâr hızı dağılımını analiz etmek için log-normal, Weibull ve lojistik olasılık dağılımlarını kullandılar. Bu dağılımların parametrelerini Genetik Algoritma (GA) kullanarak tahmin ettiler. Geleneksel yineleme yöntemine göre zaman tasarrufu açısından GA'nın daha verimli olduğunu buldular.

Jiang vd. (2015) Çin'deki iki farklı bölgedeki dört farklı konumdaki rüzgâr kaynaklarının potansiyelini değerlendirmek için dört adımdan oluşan bir sistem geliştirdi. Rüzgâr hızı frekans dağılımını değerlendirmek için log-normal dağılımı da dahil olmak üzere üç olasılık dağılımı kullandılar. Bu dağılımların parametrelerini EÇO, Moment yöntemi (MM) ve Parçacık Sürü Optimizasyonu, Diferansiyel Gelişim ve Karınca Kolonisi Optimizasyonu kullanarak tahmin ettiler. Yapay optimizasyon tekniklerine dayalı parametre tahminlerinin, EÇO ve MM' den iyi olduğunu buldular.

Jiang vd. (2017) düşük rüzgârlı bölgelerde rüzgâr enerjisi potansiyelinin değerlendirmesi için geleneksel sayısal yöntemler ve meta-sezgisel optimizasyon algoritmalarını karşılaştırdı. Weibull, Rayleigh, gama ve log-normal olasılık dağılımlarını değerlendirdiler. Bu dağılımların bilinmeyen parametrelerini tahmin etmek için üç sayısal yöntem: MM, EÇO ve en küçük kareler (EKK) yöntemi kullandılar. Ayrıca, parametre tahminlerini meta-sezgisel algoritmalarından Yarasa Algoritması (BA), Kuku Arama Algoritması (CS) ve Parçacık Sürü Optimizasyonu (PSO) ile karşılaştırdılar. Sonuçlar, meta-sezgisel optimizasyon algoritmalarının sayısal yöntemlerden daha iyi performansa sahip olduğunu gösterdi. En iyi meta-sezgisel algoritmayı belirlemede sonuçlar kesin olmamakla beraber BA ve PSO'nun CS'den daha iyi performansa sahip olduğunu gösterdi.

Yalçınkaya vd. (2018), GA kullanarak çarpık normal dağılımın parametrelerini EÇO ile karşılaştırdılar. GA'nın performansını NR, Nelder Mead ve yinelemeli olarak yeniden ağırlıklandırma algoritması gibi diğer geleneksel arama teknikleriyle karşılaştırdılar. Yan, HKO ve yetersizlik ölçütlerini kullanarak, GA tahminlerinin geleneksel arama tekniklerine kıyasla en verimli olduğunu buldular.

Wang vd. (2018), Çin'deki büyük ölçekli rüzgâr çiftliklerindeki rüzgâr enerjisini değerlendirmek için log-normal, Weibull, Rayleigh, gama ve log-lojistik olasılık

dağılımlarını kullandılar. Bu dağılımların parametrelerini üç meta-sezgisel algoritma Gri Kurt Optimizasyonu (GWO), PSO, CS ve dört geleneksel yöntem kullanarak tahmin ettiler. Sonuçlara göre, seçilen dağılım parametrelerini tahmin etmek için en verimli ve doğru yöntemin gri kurt optimizasyonu algoritması olduğunu belirttiler.

Karakoca ve Pekkör (2019) Weibull dağılımından ilerleyen tip-2 sansürlenmiş örneklerin parametrelerinin EÇO tahminlerini elde etmek için GA tabanlı bir yaklaşım önerdiler. Kapsamlı simülasyonlar gerçekleştirdiler ve GA tabanlı tahminlerini Newton yöntemi kullanılarak elde edilen tahminlerle karşılaştırdılar. Performans kriterlerine göre GA tabanlı yöntemin Newton yönteminden daha iyi performans gösterdiğini belirttiler.

Guedes vd. (2020) rüzgâr hızı olasılık dağılımlarını log-normal dağılımı da dahil olmak üzere 11 dağılım ile modellemek için parametre tahminlerini meta-sezgisel optimizasyon algoritmalarından Göçmen kuş optimizasyonu (MBO), İmparatorluk Rekabetçi Algoritması (ICA), Uyum Araması (HS) ve CS ile elde ettiler. Brezilya'nın iki farklı bölgesindeki rüzgâr hızı verileri için yapılan uygulamada karşılaştırma amacıyla EÇO tahminlerini de hesapladılar. Sonuçlar, parametre tahminlerinin doğruluğu açısından meta-sezgisel optimizasyon algoritmalarının EÇO' ten daha iyi performansa sahip olduğunu gösterdi.

Khamees vd. (2021) ABD'nin Teksas eyaletinde bir rüzgâr çiftliğinden beş yıl boyunca toplanan rüzgâr hızı verilerinin dağılımını inceledi. Rüzgâr hızı verilerini temsil etmek için log-normal, Weibull, gama ve ters gama olasılık dağılımlarını kullandılar. Bu dağılımların parametrelerini EÇO ve yeni bir meta-sezgisel optimizasyon algoritması olan Aquilla optimize edici (AO) kullanarak tahmin ettiler. AO'nun, parametre tahmini doğruluğu açısından EÇO'dan daha iyi performans gösterdiğini buldular.

### 3. META-SEZGİSEL ALGORİTMALAR

Meta-sezgisel algoritmalar, sezgisel optimizasyon algoritmaları geliştirmek için bir dizi yönerge veya strateji sağlayan üst düzey, problemden bağımsız optimizasyon teknikleridir (Sörensen ve Glover, 2015). Meta-sezgisel algoritmalar çeşitli şekillerde sınıflandırılabilir. Örneğin Osman (2003) bunları yerel arama, yapıya dayalı arama ve popülasyona dayalı arama olarak sınıflandırmıştır. Gendreau & Potvin (2005) ise meta-sezgisel algoritmaları yörünge temelli ve popülasyon temelli olarak iki şekilde sınıflandırmıştır.

Yörünge temelli algoritmalar tek bir çözümle başlar ve onu her yinelemede mevcut en iyi çözümle değiştirir. Popülasyon tabanlı algoritmalar rastgele oluşturulmuş başlangıç çözümlerinden oluşan bir popülasyonla başlar. Başlangıç popülasyonu her yinelemede aşamalı olarak geliştirilir. Her yinelemede, yeni üretilen en iyi çözümler tüm popülasyonun veya bir kısmının yerini alır.

Tipik popülasyon tabanlı algoritmalar arasında Genetik Algoritma (GA), Diferansiyel Evrim (DE), Parçacık Sürü Optimizasyonu (PSO) ve Yapay Arı Kolonisi optimizasyonu (ABC) başta olmak üzere birçok algoritma yer alır.

#### 3.1 Genetik Algoritma (GA)

Genetik Algoritma (GA), karmaşık sorunlara optimal veya optimale yakın çözümler bulmak için doğal seçim sürecini taklit eden bir tür evrimsel algoritmadır. GA ilk olarak 1975 yılında John Holland tarafından önerildi ve o zamandan beri makine öğrenimi, mühendislik ve finans dahil olmak üzere çok çeşitli uygulamalarda başarıyla kullanıldı. GA'nın çalışma adımları aşağıdaki şekilde özetlenebilir.

İlk adım, problemin potansiyel çözümlerinin nasıl temsil edileceğine karar vermektir. GA'da potansiyel çözümler kromozomlarla ifade edilir. Kromozomlar ikili diziler, gerçek değerli vektörler ve hatta grafikler gibi birçok farklı yolla temsil edilebilir. Bir kromozom temsili seçildikten sonra bir sonraki adım başlangıç popülasyonunu

oluşturmaktır. Böylece her biri problemin potansiyel çözümünü temsil eden bir dizi rastgele kromozom oluşturulur. Amaç fonksiyonu her bir kromozomun uygunluğunu değerlendiren bir fonksiyondur. GA'nın amacı optimizasyonun minimizasyon veya maksimizasyon olmasına bağlı olarak en yüksek uygunluğa sahip kromozomu bulmaktır.

Seçim kısmında elitist strateji kullanılarak mevcut en iyi çözüm korunarak sonraki nesiller oluşturulur. Bu strateji, her bir neslin en iyi çözümlerini koruyarak, bunların sonraki nesillere aktarılmasını sağlar. Temel amacı, popülasyonun kalitesini artırmak ve çözüm uzayındaki en iyi çözümleri koruyarak gelişimi teşvik etmektir. Bu sayede, genetik algoritmaların her bir iterasyonunda, problemi daha iyi çözebilen ve istenen sonuca daha hızlı ulaşabilen bir popülasyon oluşturur. Elitist stratejinin kullanımı, algoritmanın çözüm uzayında daha hızlı ve daha doğru bir şekilde hareket etmesine olanak tanır, böylece optimizasyon süreci daha etkili hale gelir. Bu stratejinin uygulanması, genetik algoritmaların başarısını artırır ve problemlerin daha verimli bir şekilde çözülmesine yardımcı olur.

GA popülasyonu geliştirmek için üç genetik operatör kullanır: seçim, çaprazlama ve mutasyon. Seçim, popülasyondan gelecek neslin ebeveynleri olacak kromozomları seçer. Seçim operatörü için rastgele seçim, turnuva seçimi ve rulet çarkı seçimi gibi farklı seçim yöntemleri kullanılabilir. Uygunluğu daha yüksek olan kromozomların sonraki nesle seçilme olasılığı daha yüksektir.

Çaprazlama, sonraki nesilde çeşitliliği sağlayacak iki yavru kromozomu iki ebeveyn kromozomu kullanarak birleştirir. Bu, GA'nın arama uzayının yeni bölgelerini keşfetmesine olanak tanır.

Mutasyon, bir kromozomdaki bir veya daha fazla genin rastgele değişmesine neden olur. Bu, GA'nın yerel optimuma takılıp kalmasının önlenmesine yardımcı olur.

GA prosedürü aşağıdaki gibidir:

- 1) Algoritmada kullanılacak parametre değerleri (popülasyon büyüklüğü, çaprazlama olasılığı, mutasyon olasılığı, ...) belirlendikten sonra başlangıç popülasyonunu oluşturulur.
- 2) Her kromozomun uygunluğu değerlendirilir.
- 3) Popülasyondaki en iyi çözüm belirlenir.
- 4) Ebeveynler seçilir ve çaprazlama gerçekleştirilir.
- 5) Yavrulara mutasyon operatörü uygulanır.
- 6) Yavruların uygunluk değerleri hesaplanır.
- 7) Ebeveyn popülasyonu yavrularla birleştirilir.
- 8) Popülasyon uygunluk fonksiyonuna göre sıralanır.
- 9) En iyi çözümü kaydedilir.
- 10) Sonlandırma kriteri karşılanana kadar 2-9 arasındaki adımlar tekrarlanır.

### 3.1.1 GA Örneği

GA'yı çalıştırmadan önce öncelikle kullanıcı tanımlı parametreler tanımlanır.

- |  |                                   |
|--|-----------------------------------|
| 1. Popülasyon büyüklüğü                | : 6                               |
| 2. İterasyon sayısı                    | : 200                             |
| 3. Çaprazlama Olasılığı                | : 0,8                             |
| 4. Mutasyon Oranı                      | : 0,1                             |
| 5. Mutasyon Adım Boyutu (sigma)        | : 0,05                            |
| 6. Karar değişkenlerinin sayısı( $D$ ) | : 2                               |
| 7. Alt – Üst sınırlar                  | : $0 < x_i < 10$                  |
| 8. Amaç fonksiyonu                     | : $\text{mak} \sum_{i=1}^D x_i^2$ |

Adım 1: Rastgele çözümler üretilir

$$P = \begin{bmatrix} \text{Gen 1} & \text{Gen 2} \\ 4 & 1 \\ 10 & 9 \\ 7 & 6 \\ 6 & 7 \\ 2 & 0 \\ 2 & 10 \end{bmatrix} \begin{matrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{matrix}$$

Adım 2: Amaç fonksiyonu hesaplanır ( $\text{mak} \sum_{i=1}^D x_i^2$ )

$$f = \begin{bmatrix} 4^2 + 1^2 \\ 10^2 + 9^2 \\ 7^2 + 6^2 \\ 6^2 + 7^2 \\ 2^2 + 0^2 \\ 2^2 + 10^2 \end{bmatrix} = \begin{bmatrix} 17 \\ 181 \\ 85 \\ 85 \\ 4 \\ 104 \end{bmatrix}$$

Adım 3: En iyi çözüm belirlenir. Uygunluk değerlerine ( $f$ ) bakılarak en büyük değer 181 olduğundan, popülasyondaki bu değere karşılık gelen en iyi kromozom  $P_2$  belirlenir.

Adım 4: Çaprazlama işlemi gerçekleştirilir. Bu örnekte popülasyonun %80'i yavru olacak şekilde seçilir. Bu işlem örnek için uygulandığında  $P_3$ ,  $P_4$ ,  $P_2$  ve  $P_5$  bireyleri çaprazlama için rastgele seçilsin. Rastgele seçilen iki bireyden birinci ebeveyn  $P_3$ , [7 6]'ya karşılık gelirken, ikinci ebeveyn  $P_4$ , [6, 7] ile temsil edilmektedir. Düzgün Çaprazlama işlemi Denklem 3.1 ve Denklem 3.2 kullanılarak uygulanır.

$$O_1 = \alpha_1 \times P_1 + (1 - \alpha_1) \times P_2 \quad (3.1)$$

$$O_2 = \alpha_2 \times P_2 + (1 - \alpha_2) \times P_1 \quad (3.2)$$

Burada  $\alpha$ , ebeveynlerin boyutuyla birlikte [0, 1] aralığında rastgele bir değerdir. Rastgele oluşturulan  $\alpha = [0,432 \quad 0,291]$  vektörü kullanılarak çaprazlama işlemi gerçekleştirilir.

Böylece 1. ve 2. yavrular oluşturulur.

$$O_1 = 0,432 \times [7 \quad 6] + (1 - 0,432) \times [6 \quad 7] = [6,432 \quad 6,709]$$

$$O_2 = 0,291 \times [6 \quad 7] + (1 - 0,291) \times [8 \quad 9] = [6,568 \quad 6,291]$$

İkinci çaprazlama için rastgele seçilen iki ebeveyn,  $P_2 = [10 \quad 9]$ ,  $P_5 = [2 \quad 0]$  ve rastgele oluşturulan  $\alpha = [0,2000 \quad 0,514]$  kullanılarak, 3 ve 4 numaralı yavrular oluşturulur.

$$O_3 = 0,200 \times [10 \quad 9] + (1 - 0,200) \times [2 \quad 0] = [3,600 \quad 1,800]$$

$$O_4 = 0,514 \times [2 \quad 0] + (1 - 0,514) \times [10 \quad 9] = [5,888 \quad 4,374]$$

$$\begin{bmatrix} 6,432 & 6,709 \\ 6,568 & 6,291 \\ 3,600 & 1,800 \\ 5,888 & 4,374 \end{bmatrix}$$

Adım 5: Mutasyonu gerçekleştirilir. Mutasyon işlemi Denklem 3.3'e göre gerçekleştirilir.

$$O' = O + \sigma \times rand(1, size(O)) \quad (3.3)$$

Mevcut genin mutasyona uğrayıp uğramayacağını belirlemek için 0 ile 1 arasında rastgele bir vektör ( $\alpha$ ) üretilir. Eğer ilgili gen için  $\alpha$ 'nın elemanı mutasyon oranından küçükse o gen Denklem 3.3'e göre mutasyona uğrar. Örnekte bir önceki adımda oluşturulan 1. yavru için ( $[6,432 \quad 6,709]$ )  $\alpha$ 'nın  $[0,712 \quad 0,043]$  olduğu varsayalım. Mutasyon oranı 0,1 olduğundan sadece ikinci gen mutasyona uğrayacaktır. İkinci gen Denklem 3.3'e göre,

$$6,709 + 0,05 \times 0,1705 = 6,717$$

$$O'_1 = [6,432 \quad 6,717]$$

şeklinde elde edilir. 2. yavru için  $\alpha = [0,912 \quad 0,238]$  olursa her iki değer de mutasyon oranından küçük olmadığından mutasyon meydana gelmez. Bu nedenle mevcut yavru basitçe kopyalanır.

$$O'_2 = [6,568 \quad 6,291]$$

3. yavru için  $\alpha = [0,1708 \quad 0,431]$ . Her iki değer de mutasyon oranından küçük olmadığından mutasyon meydana gelmez. Bu nedenle mevcut yavru basitçe kopyalanır.

$$O'_3 = [3,600 \quad 1,800]$$

4. yavru için  $\alpha = [0,06 \quad 0,118]$  olsun. Bu yavrunun birinci geni mutasyon oranından küçük olduğu için değiştirilecek, ikinci gen ise mutasyon oranından büyük olduğu için kopyalanacaktır.

$$5,888 + 0,05 \times 0,259 = 5,901$$

$$O'_4 = [5,901 \quad 4,374]$$

Mutasyon işlemi tamamladıktan sonra alt ve üst sınırların dışındaki değişkenler sınırlanır.

Ancak tüm değişkenler zaten 0 ile 10 aralığında olduğundan bu durumda sınırlamaya gerek yoktur.

$$O' = \begin{bmatrix} 6,432 & 6,717 \\ 6,568 & 6,291 \\ 3,600 & 1,800 \\ 5,901 & 4,374 \end{bmatrix}$$

Adım 6: yavruların uygunluk değerleri hesaplanır.

$$f_o = \begin{bmatrix} 86.489 \\ 82.715 \\ 16.200 \\ 53.954 \end{bmatrix}$$

Adım 7: Popülasyon ve yavrular birleştirilir.

$$P + O = \begin{bmatrix} 4 & 1 \\ 10 & 9 \\ 7 & 6 \\ 6 & 7 \\ 2 & 0 \\ 2 & 10 \\ 6,432 & 6,717 \\ 6,568 & 6,291 \\ 3,600 & 1,800 \\ 5,901 & 4,374 \end{bmatrix}, \quad f + f_o = \begin{bmatrix} 17 \\ 181 \\ 85 \\ 85 \\ 4 \\ 104 \\ 86,489 \\ 82,715 \\ 16,200 \\ 53,954 \end{bmatrix}$$

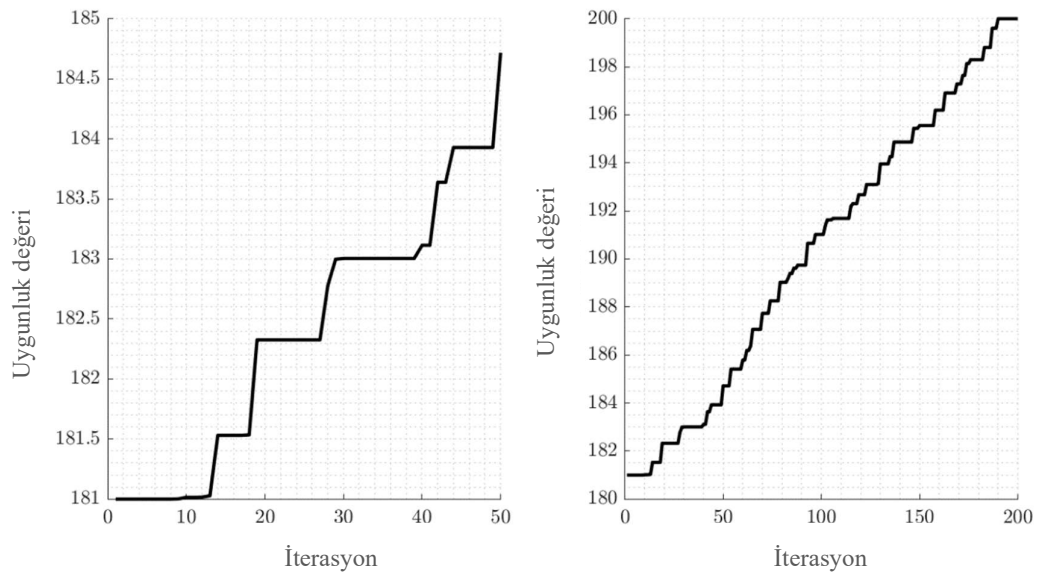
Adım 8: Birleştirilmiş popülasyon uygunluk değerlerine göre sıralanır. En yüksek uygunluk değerine sahip popülasyon büyüklüğü kadar birey saklanır:

$$P = \begin{bmatrix} 10 & 9 \\ 2 & 10 \\ 6,432 & 6,717 \\ 7 & 6 \\ 6 & 7 \\ 6,568 & 6,291 \end{bmatrix}$$

Adım 9: En iyi çözümü belirler. Bu optimizasyon probleminin doğasına bağlıdır. Minimizasyon problemleri için amaç fonksiyonunun en düşük değerine sahip popülasyon saklanır. Tersine, maksimizasyon problemlerinde amaç fonksiyonunun en yüksek değerine sahip birey popülasyonda korunur. Bu problem için amaç fonksiyonu en büyükleme (maksimum) problemi olduğundan amaç fonksiyonunun en yüksek değeri 181'dir ve bu değer [10 9] kromozomuna karşılık gelir. Bulunan sonuç eski uygunluk değerinden büyük olmadığı için güncelleme yapılmayacaktır.

Adım 10:  $N$  maksimum nesil (yineleme) sayısı için 2-8 arasındaki adımlar tekrarlanır.

Şekil 3.1'de GA'nın amaç fonksiyonunun farklı iterasyon sayılarına göre uygunluk değerlerindeki değişimler verilmiştir.



Şekil 3.1. GA uygunluk değerlerinin yineleme sayısına göre değişimi.

### 3.2 Parçacık Sürü Optimizasyonu (PSO)

Parçacık Sürü Optimizasyonu (PSO), Kennedy ve Eberhart tarafından 1995 yılında önerilen, popülasyona dayalı bir meta-sezgisel algoritmadır. Yiyecek aramak için kuş sürülerinin veya balık sürülerinin sosyal davranışlarından ilham alır. PSO, genetik algoritmalar ve diğer evrimsel algoritmalar gibi, bir amaç fonksiyonunu yinelemeli olarak optimize etmeyi amaçlar. Ancak GA' lardan farklı olarak PSO, çaprazlama ve mutasyon gibi genetik operatörleri kullanmaz. Bunun yerine parçacık adı verilen potansiyel çözümler, en uygun çözümü bulmak için problemin arama uzayını tarar.

Sürüdeki her parçacığın bir konumu vardır ve kişisel en iyi konumunu ve onunla ilişkili uygunluk değerini bilir. Ayrıca mevcut genel uygunluk değerinin yanı sıra sürü içindeki global en iyi konumu da bilir.

Sürüdeki her parçacık  $n$  boyutlu bir vektörle temsil edilir. Sürüdeki her parçacık aynı zamanda konumuyla aynı boyutta bir hıza sahiptir. Hız, parçacıkların genel en iyiye doğru hareket etmesini sağlar. Parçacıkların hareketi, sürü parçacıkları arasındaki bilgi alışverişiyle belirlenir, çünkü her parçacık kendi kişisel en iyisini bilir. Bu nedenle, problemin genel optimumuna ulaşana kadar hareketleri iletişim ve öğrenmeye dayalıdır.

Sürüdeki her bir  $i$  parçacığının hızı, her  $t$  (yineleme) anında aşağıdaki denkleme göre güncellenir:

$$v_i(t + 1) = wv_i(t) + c_1r_1(p_i(t) - x_i(t)) + c_2r_2(g(t) - x_i(t)) \quad (3.4)$$

burada  $w$ , genel keşif ve yerel sömürüyü dengelemek için kullanılan eylemsizlik ağırlığıdır.  $c_1$  ve  $c_2$  sırasıyla bilişsel ve sosyal bileşenlere ilişkin ivme katsayılarıdır.  $r_1$  ve  $r_2$ ,  $[0,1]$  arasında düzgün dağılmış  $n$  boyutlu rastgele vektörlerdir.  $p_i(t)$ ,  $i$ . parçacığının  $t$  zamanındaki kişisel en iyi konumudur.  $g(t)$  sürünün mevcut genel en iyi konumudur. Her parçacığın konumu her  $t$  anında aşağıdaki denkleme göre güncellenir:

$$x_i(t + 1) = x_i(t) + v_i(t + 1) \quad (3.5)$$

$x_0$  ve  $v_0$ 'ın başlangıç vektörleri düzgün dağılım kullanılarak üretilebilir. Benzer şekilde, kişisel en iyi konum,  $i$ . parçacığın  $p_0 = x_0$  olan mevcut (başlangıç) konumu tarafından belirlenir.

Tipik bir PSO algoritması şu şekilde çalışır:

- 1) Sürüyü oluşturulur: Bu, her biri problemin potansiyel çözümünü temsil eden bir dizi parçacığın rastgele oluşturulmasını içerir.
- 2) Her parçacığın uygunluğu değerlendirilir: Bir parçacığın uygunluğu, onun soruna çözüm olarak ne kadar iyi olduğunun ölçüsüdür.
- 3) Her parçacığın kişisel en iyi konumu güncellenir: Bir parçacığın kişisel en iyi konumu, o ana kadar bulduğu en iyi çözüm değeridir.
- 4) Genel en iyi konum belirlenir: Genel en iyi konum, sürüdeki herhangi bir parçacığın bulduğu en iyi çözümdür.
- 5) Her parçacığın hızı güncellenir: Bir parçacığın hızı onun çözüm uzayındaki hareketini belirler. Hız, parçacığın kişisel en iyi konumuna, genel en iyi konumuna ve rastgele bir vektöre göre güncellenir.
- 6) Her parçacığın konumu güncellenir: Bir parçacığın konumu, hızına bağlı olarak güncellenir.
- 7) Bir sonlandırma kriteri karşılanana kadar 2-6 arasındaki adımlar tekrarlanır. Sonlandırma kriteri tipik olarak maksimum yineleme sayısı veya uygunluk eşiğidir.

### 3.2.1 PSO Örneği

Öncelikle PSO kullanıcı tanımlı parametreleri tanımlanır.

1. Karar değişkenlerinin sayısı ( $D$ ) : 2
2. Alt – Üst sınırlar :  $0 < x_i < 10$
3. Amaç fonksiyonu :  $\text{mak } \sum_{i=1}^D x_i^2$

4. Sürünün Büyüklüğü : 4  
 5. Eylemsizlik katsayısı ( $w$ ) : 0,8  
 6. Bilişsel katsayısı ( $c_1$ ) : 2  
 7. Sosyal katsayısı ( $c_2$ ) : 2  
 8. İterasyon sayısı ( $N$ ) : 5

Adım 1: Sürüyü ve hız değerleri rastgele başlatılır.

$$P = \begin{bmatrix} 4 & 2 \\ 10 & 2 \\ 7 & 1 \\ 6 & 9 \end{bmatrix}, \quad V = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Adım 2: Amaç fonksiyonuna göre uygunluk değerleri hesaplanır:

$$f = \begin{bmatrix} 4^2 + 2^2 \\ 10^2 + 2^2 \\ 7^2 + 1^2 \\ 6^2 + 9^2 \end{bmatrix} = \begin{bmatrix} 20 \\ 104 \\ 50 \\ 117 \end{bmatrix}$$

Adım 3: Kişisel en iyi ( $P_{best}$ ) belirlenir. İlk adımda kişisel en iyi, her parçacığın mevcut konumuna eşittir.

$$P_{best} = \begin{bmatrix} 4 & 2 \\ 10 & 2 \\ 7 & 1 \\ 6 & 9 \end{bmatrix}$$

Adım 4: Global en iyi güncellenir (Amaç fonksiyonu bir maksimum problemi olduğu için  $G_{best} = P_4 = [6 \ 9]$ ,  $G_{f_{best}} = 117$  alınır).

Adım 5: Her parçacığın hızını ve konumu Denklem (3.4) ve (3.5) 'ye göre güncellenir.

Parçacık 1 için:

$$r_1 = [0,30 \ 0,53], \quad r_2 = [0,43 \ 0,29]$$

$$v_{1,1} = 0,8 \times [0 \ 0] + 2 \times [0,30 \ 0,53] \times ([10 \ 2] - [10 \ 2])$$

$$+ 2 \times [0,43 \ 0,29] \times ([6 \ 9] - [10 \ 2])$$

$$v_{1,1} = [1,73 \ 4,1]$$

$$P_{1,1} = [4 \ 2] + [1,73 \ 4,1] = [5,73 \ 6,1]$$

şeklinde hız ve konum değerleri belirlenir. Çözümlerden herhangi biri sınırların dışındaysa değerler sınırları sağlayacak şekilde değiştirilir. Çözümün tamamı 0 ile 10 aralığında olduğundan sınırlamaya gerek yoktur. Birinci parçacığın 1. iterasyondaki değerinin ( $P_{1,1}$ ) uygunluğu hesaplanır.

$$f_{1,1} = 69,71$$

Kişisel en iyi, uygunluk değerlerine göre güncellenir.  $P_{1,1}$ 'in uygunluk değeri  $P_{0,1}$ 'den büyükse kişisel en iyi  $P_{1,1}$  olacaktır.

$$f_{1,1} = 69,71, \quad f_{0,1} = 20$$

$$f_{1,1} > f_{0,1}$$

$$P_{best,1} = [5,73 \quad 6,1]$$

Ancak  $f_{1,1} < G_{fbest}$  olduğundan genel en iyi güncellenmeyecektir.

$$P = \begin{bmatrix} 5,73 & 6,1 \\ 10 & 2 \\ 7 & 1 \\ 6 & 9 \end{bmatrix}, \quad V = \begin{bmatrix} 1,73 & 4,1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad P_{best} = \begin{bmatrix} 5,73 & 6,1 \\ 10 & 2 \\ 7 & 1 \\ 6 & 9 \end{bmatrix}, \quad f = \begin{bmatrix} 69,74 \\ 104 \\ 50 \\ 117 \end{bmatrix}$$

Parçacık 2 için:

$$r_1 = [0,61 \quad 0,14], \quad r_2 = [0,29 \quad 0,37]$$

$$v_{1,2} = 0,8 \times [0 \quad 0] + 2 \times [0,61 \quad 0,14] \times ([10 \quad 2] - [10 \quad 2])$$

$$+ 2 \times [0,29 \quad 0,37] \times ([6 \quad 9] - [10 \quad 2])$$

$$v_{1,2} = [-2,3 \quad 5,1]$$

$$P_{1,2} = [10 \quad 2] + [-2,3 \quad 5,1] = [7,7 \quad 7,1]$$

şeklinde hesaplamalar yapıldığında Çözümler sınırları sağladığı için çözümleri sınırlamaya gerek yoktur. İkinci parçacığın 1. iterasyondaki uygunluk değeri ( $P_{1,2}$ ) hesaplanır.

$$f_{1,2} = 109,51$$

Kişisel en iyi, uygunluk değerine göre güncellenir.  $P_{1,2}$  'nin uygunluk değeri  $P_{0,2}$ 'den büyük olduğundan kişisel en iyi  $P_{1,2}$  olacaktır.

$$f_{1,2} = 109,571, \quad f_{0,2} = 104$$

$$f_{1,2} > f_{0,2}$$

$$P_{best,2} = [7,7 \quad 7,1]$$

Ancak  $f_{1,2} < G_{f_{best}}$  olduğu için global en iyi güncellenmeyecektir.

$$P = \begin{bmatrix} 5,73 & 6,1 \\ 7,7 & 7,1 \\ 7 & 1 \\ 6 & 9 \end{bmatrix}, \quad V = \begin{bmatrix} 1,73 & 4,1 \\ -2,3 & 5,1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad P_{best} = \begin{bmatrix} 5,73 & 6,1 \\ 7,7 & 7,1 \\ 7 & 1 \\ 6 & 9 \end{bmatrix}, \quad f = \begin{bmatrix} 69,74 \\ 109,54 \\ 50 \\ 117 \end{bmatrix}$$

Parçacık 3 için:

$$r_1 = [0,46 \quad 0,79], \quad r_2 = [0,2 \quad 0,51]$$

$$v_{1,3} = 0,8 \times [0 \quad 0] + 2 \times [0,46 \quad 0,79] \times ([7 \quad 1] - [7 \quad 1])$$

$$+ 2 \times [0,2 \quad 0,51] \times ([6 \quad 9] - [7 \quad 1])$$

$$v_{1,3} = [-0,4 \quad 8,2]$$

$$P_{1,3} = [7 \quad 1] + [-0,4 \quad 8,2] = [6,6 \quad 9,2]$$

hesaplamalar yapıp  $P_{1,3}$ 'ün uygunluk değeri hesaplanır.

$$f_{1,3} = 128,72$$

Kişisel en iyi, uygunluk değerine göre güncellenir.  $P_{1,3}$ 'ün uygunluk değeri  $P_{0,3}$ 'ten büyük olduğundan kişisel en iyi  $P_{1,3}$  olur.

$$f_{1,3} = 128,72, \quad f_{0,3} = 50$$

$$f_{1,3} > f_{0,3}$$

$$P_{best,3} = [6,6 \quad 9,2]$$

Global en iyi  $f_{1,3} > G_{f_{best}}$  olduğu için güncellenir.

$$G_{best} = [6,6 \quad 9,2], \quad G_{fbest} = 128,72$$

$$P = \begin{bmatrix} 5,73 & 6,1 \\ 7,7 & 7,1 \\ 6,6 & 9,2 \\ 6 & 9 \end{bmatrix}, \quad V = \begin{bmatrix} 1,73 & 4,1 \\ -2,3 & 5,1 \\ -0,4 & 8,2 \\ 0 & 0 \end{bmatrix}, \quad P_{best} = \begin{bmatrix} 5,73 & 6,1 \\ 7,7 & 7,1 \\ 6,6 & 9,2 \\ 6 & 9 \end{bmatrix}, \quad f = \begin{bmatrix} 69,74 \\ 109,54 \\ 128,72 \\ 117 \end{bmatrix}$$

Parçacık 4 için:

$$r_1 = [0,60 \quad 0,05], \quad r_2 = [0,61 \quad 0,18]$$

$$v_{1,4} = 0,8 \times [0 \quad 0] + 2 \times [0,60 \quad 0,05] \times ([6 \quad 9] - [6 \quad 9]) \\ + 2 \times [0,61 \quad 0,18] \times ([6,6 \quad 9,2] - [6 \quad 9])$$

$$v_{1,4} = [0,73 \quad 0,1]$$

$$P_{1,4} = [6 \quad 9] + [0,73 \quad 0,1] = [6,73 \quad 9,1]$$

hesaplamalar yapılır ve  $P_{1,4}$ 'ün uygunluk değeri hesaplanır.

$$f_{1,4} = 127,7$$

Kişisel en iyi, uygunluk değerine göre güncellenir.  $P_{1,4}$ 'ün uygunluk değeri  $P_{0,4}$ 'ten büyük olduğundan kişisel en iyi  $P_{1,4}$  ile güncellenir.

$$f_{1,4} = 127,7, \quad f_{0,4} = 117$$

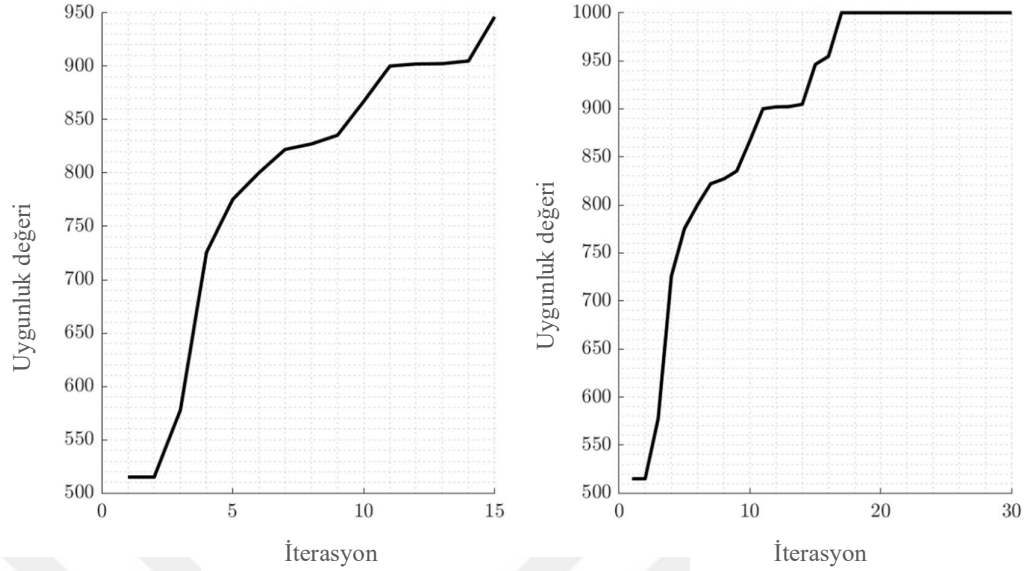
$$f_{1,4} > f_{0,4}$$

$$P_{best,4} = [6,73 \quad 9,1]$$

Global en iyi daha büyük olduğundan ( $f_{1,4} < G_{fbest}$ ) güncellenmeyecektir.

$$P = \begin{bmatrix} 5,73 & 6,1 \\ 7,7 & 7,1 \\ 6,6 & 9,2 \\ 6,73 & 9,1 \end{bmatrix}, \quad V = \begin{bmatrix} 1,73 & 4,1 \\ -2,3 & 5,1 \\ -0,4 & 8,2 \\ 0,73 & 0,1 \end{bmatrix}, \quad P_{best} = \begin{bmatrix} 5,73 & 6,1 \\ 7,7 & 7,1 \\ 6,6 & 9,2 \\ 6,73 & 9,1 \end{bmatrix}, \quad f = \begin{bmatrix} 69,74 \\ 109,54 \\ 128,72 \\ 127,7 \end{bmatrix}$$

Bu işlem maksimum N iterasyon için tekrarlanır. Şekil 3.2'de amaç fonksiyonunun maksimum problemi olduğu durumda 10 değişken ve 20 parçacıkla elde edilen uygunluk değerleri verilmiştir. Soldaki şekil, uygunluk değerinin 15 iterasyon sonraki ilerlemesini gösterirken, sağdaki şekil 30 iterasyon sonraki uygunluk değerlerini gösterir.



Şekil 3.2. PSO aramada amaç fonksiyonunun iterasyon sayısına göre değişimi.

### 3.3 Yapay Arı Kolonisi (ABC)

Yapay Arı Kolonisi (ABC), Karaboğa ve Baştürk tarafından 2007 yılında geliştirilen, doğadan ilham alan popülasyon temelli meta-sezgisel bir algoritmadır. Bal arılarının yiyecek arama davranışlarından esinlenilerek geliştirilmiştir.

ABC algoritmasının üç önemli bileşeni vardır. İlk bileşen besin kaynaklarıdır. Bir arı yiyecek bulmak için kovanın etrafındaki bölgeyi araştırır. Arı besin kaynağını seçer, yiyeceğin kovana yakınlığı, nektar miktarı, besin kaynağından yararlanma kolaylığı gibi birçok özelliği değerlendirir. ABC algoritmasında besin kaynağı, optimizasyon problemine potansiyel bir çözümü temsil eder ve besin kaynağındaki nektar miktarı, uygunluk fonksiyonuna karşılık gelen yiyecek kaynağının kalitesini temsil eder.

ABC algoritmasının ikinci bileşeni çalışan toplayıcıdır. Çalışan bir toplayıcı, daha önce bir besin kaynağını ziyaret etmiş ve onu kullanan bir arıdır. Ziyaret ettiği besin kaynağının yerini ezberler ve kovana uzaklığı, besin kaynağının yönü ve kârlılığı gibi besin kalitesine ilişkin bilgileri taşıyarak kovana geri döner.

Üçüncü bileşen ise işsiz toplayıcıdır. İşsiz bir toplayıcı, şu anda yararlanacak besin kaynağı arayan bir arıdır. İşsiz bir toplayıcı, izleyici veya kâşif arı olabilir. Gözcü bir arı, işçi arıların diğer arıları işe almak için sallanma dansı yaptığı dans alanında işçi arıyı izler. Bu dans, arıların iletişim kurmasının ve bilgi alışverişinde bulunmasının bir yoludur. Gözcü arılar, görevli arıların paylaştığı bilgilere dayanarak gelecek vaat eden bir besin kaynağı seçerler. Kâşif arılar, mevcut kaynaklar tükendiğinde yeni besin kaynakları keşfederler.

ABC algoritmasında besin kaynağı sayısı, görevli veya gözcü arıların sayısına eşittir. Her besin kaynağı bir arı tarafından sömürülür. Algoritmanın ilk adımında,  $D$ 'nin optimizasyon problemindeki parametre sayısını temsil ettiği  $D$  boyutlu uzayda her arı için rastgele yiyecek kaynağı üretilir. Her besin kaynağının amaç fonksiyonu ve uygunluk değeri değerlendirilir ve saklanır. Uygunluk değeri Denklem 3.6 ile hesaplanabilir:

$$\begin{cases} \frac{1}{1+f_i} & f_i \geq 0 \\ \frac{1}{1+|f_i|} & f_i < 0 \end{cases} \quad (3.6)$$

burada  $f_i$ ,  $i$ 'inci besin kaynağının amaç fonksiyonu değeridir. Daha sonra görevli arılar Denklem 3.7'yi kullanarak mevcut besin kaynağı (çözüm) etrafında yeni yiyecek kaynağı (çözüm) keşfederler:

$$x'_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{pj}) \quad (3.7)$$

burada  $\phi_{ij}$   $-1$  ile  $1$  arasında rastgele bir sayıdır,  $x_{ij}$ ,  $j$ . boyutun  $i$ . besin kaynağını temsil eder.  $x_{pj}$ ,  $j$ . boyutun  $p$ . ortak arı besin kaynağıdır ( $p \in (1,2,3, \dots, N)$  ve  $i \neq p$ ).

Yeni çözümün uygunluk değeri hesaplanır. Mevcut çözüm yeni çözümle karşılaştırılır. Açgözlü bir seçim yapılır ve iki çözümden en iyisi saklanır, diğeri atılır. Tüm görevli arılar aramalarını tamamladıktan sonra her besin kaynağının uygunluk değerini gözcü arılarla paylaşırlar. Gözcü arı, besin kaynağının uygunluğunu belirten  $p_i$

olasılık değerine göre bir besin kaynağı seçer. Olasılık Denklem 3.8 kullanılarak hesaplanır:

$$p_i = \left(0.9 \times \frac{f_i}{\max(f)}\right) + 0.1 \quad (3.8)$$

Bir besin kaynağı seçildikten sonra Denklem (3.7) kullanılarak seçilen besin kaynağı etrafında yeni bir çözüm üretilir. Yeni besin kaynağının uygunluk değeri hesaplanır ve mevcut besin kaynağıyla karşılaştırılır. Yeni besin kaynağı mevcut olandan daha iyi ise yeni besin kaynağı depolanır, aksi takdirde mevcut olan korunur.

Besin kaynağının kalitesi düşükse o besin kaynağı terk edilir ve bir kâşif arı tarafından yeni bir besin kaynağı keşfedilir. ABC algoritmasında, bir gıda kaynağının karlılığı önceden tanımlanmış sayıda denemede artırılamıyorsa, o gıda kaynağından vazgeçilir.

ABC algoritmasının temel adımları şu şekilde özetlenebilir:

1. Başlatma: Değişkenleri ve kısıtlamaları dahil olmak üzere çözülecek sorun tanımlanır. Arı sayısı, maksimum döngü sayısı ve terkedilmiş yiyecek kaynaklarının sınırı gibi algoritma parametreleri belirlenir. Besin kaynaklarının başlangıç popülasyonun rastgele oluşturulur (potansiyel çözümler).
2. Görevli Arılar Aşaması: Her görevli arı, mevcut kaynağının yakınında yeni bir besin kaynağı (çözüm) araştırır. Bunu mevcut çözümün rastgele seçilen bir boyutunu değiştirerek yapar. Yeni çözümün uygunluğu (kalite) değerlendirilir. Yeni çözüm mevcut çözümden daha iyi ise mevcut çözümün yerine geçer.
3. Gözcü Arılar Aşaması: Gözcü arılar besin kaynaklarını uygunluklarıyla orantılı olan olasılıklarına göre seçerler. Uygunluğu yüksek besin kaynaklarının seçilme şansı daha yüksektir. Gözcü arılar daha sonra, görevli

arılar benzer şekilde, seçilenlerin yakınındaki yeni besin kaynaklarını keşfederler.

4. Kâşif Arılar Aşaması: Bir besin kaynağı belirli sayıda denemeden sonra iyileştirilmemişse terk edilmiş sayılır. Bu besin kaynağıyla ilişkili görevli arı, kâşif arı olur. Kâşif arı, terk ettiği besin kaynağının yerine rastgele yeni bir besin kaynağı arar.
5. Ezberleme: Şu ana kadar bulunan en iyi besin kaynağı ezberlenir.
6. Sonlandırma: Algoritma, maksimum döngü sayısı veya tatmin edici bir çözümün bulunması gibi bir sonlandırma kriteri karşılanana kadar 2-5 arasındaki adımları tekrarlar.

### 3.3.1 ABC Örneği

Öncelikle ABC kullanıcı tanımlı parametreleri belirlenir.

- |   |                                   |
|---|-----------------------------------|
| 1. Karar değişkenlerinin sayısı ( $D$ )               | : 2                               |
| 2. Alt – üst sınırlar                                 | : $0 < x_i < 10$                  |
| 3. Amaç fonksiyonu                                    | : $\text{mak} \sum_{i=1}^D x_i^2$ |
| 4. Sürünün büyüklüğü                                  | : 6                               |
| 5. Besin kaynağı sayısı, görevli arılar, gözcü arılar | : 3                               |
| 6. İterasyon sayısı                                   | : 10                              |
| 7. Limit  | : 1                               |

Adım 1: Besin kaynaklarını ve iz (trail) vektörü rastgele oluşturulur, Denklem 3.6 kullanılarak amaç fonksiyonu ve uygunluk değeri hesaplanır.

$$P = \begin{bmatrix} 4 & 6 \\ 10 & 2 \\ 7 & 2 \end{bmatrix}, \quad f = \begin{bmatrix} 52 \\ 104 \\ 53 \end{bmatrix}, \quad fit = \begin{bmatrix} 0,0189 \\ 0,0095 \\ 0,0185 \end{bmatrix}, \quad trail = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Adım 2: Çalışan arı aşaması gerçekleştirilir. Çözüm 1 için, değiştirilecek bir değişken rastgele seçilir ( $X_{i,j}$ ). Daha sonra, rastgele bir  $X_{p,j}$  ortağı seçilerek phi ( $\phi$ ) değeri

oluşturulur. Bu örnek için  $i = 1, j = 1, p = 3$  ve  $\phi = 0,2$  olarak kabul edilsin.

Değiştirilen değişken Denklem (3.7)' ye göre hesaplanır

$$X'_{1,1} = 4 + 0,2 \times (4 - 7) = 3,39.$$

$X'_{1,1}$ 'in çözüm alanı içinde olup olmadığı kontrol edilir. Bu durumda  $x'_{1,1}$  çözüm uzayı içerisine düştüğü için herhangi bir sınırlamaya gerek yoktur. Bu nedenle değiştirilmiş  $X'_{11}$  karşılık gelen fonksiyon değeri  $f'_1 = 47,5$  ve uygunluk değeri  $fit'_1 = 0,0206$  ile  $[3,39 \ 6]$ 'dır. Açgözlü bir seçim yaklaşımı kullanılarak  $fit'_1 < fit_1$  ise yeni çözümü korunur.  $fit'_1 > fit_1$  olduğundan yeni çözüm atılır ve iz (trail) limiti 1 artırılır.

$$P = \begin{bmatrix} 4 & 6 \\ 10 & 2 \\ 7 & 2 \end{bmatrix}, \quad f = \begin{bmatrix} 52 \\ 104 \\ 53 \end{bmatrix}, \quad fit = \begin{bmatrix} 0,0189 \\ 0,0095 \\ 0,0185 \end{bmatrix}, \quad trail = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Çözüm 2 için, değiştirilecek değişken rastgele seçilir ( $X_{i,j}$ ). Daha sonra, rastgele bir  $X_{p,j}$  ortağı seçilir ve phi ( $\phi$ ) değeri oluşturulur. Bu örnek için  $i = 2, j = 2, p = 1$  ve  $\phi = 0,94$  olarak kabul edilsin. Değiştirilen değişken Denklem (3.7)' ye göre hesaplanır

$$X'_{2,2} = 2 + 0,94 \times (2 - 6) = -1,76.$$

$X'_{2,2}$  çözüm alanı içinde olmadığı ve alt sınırdan daha küçük değer aldığından  $(-1,76) \notin (0, 10)$  bu çözüm altı sınır değeri ile değiştirilir. Bu nedenle,  $X'_{22} [10 \ 0]$  alınır, ilgili fonksiyon değeri  $f'_2 = 100$  ve uygunluk değeri  $fit'_2 = 0,0099$  olur. Açgözlü bir seçim yaklaşımı ile eğer  $fit'_2 < fit_2$  ise yeni çözüm korunur.  $fit'_2 > fit_2$  olduğundan yeni çözüm atılır ve iz (trail) limiti 1 artırılır.

$$P = \begin{bmatrix} 4 & 6 \\ 10 & 2 \\ 7 & 2 \end{bmatrix}, \quad f = \begin{bmatrix} 52 \\ 104 \\ 53 \end{bmatrix}, \quad fit = \begin{bmatrix} 0,0189 \\ 0,0095 \\ 0,0185 \end{bmatrix}, \quad trail = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

Çözüm 3 için, değiştirilecek değişken rastgele seçilir ( $X_{i,j}$ ). Daha sonra, rastgele bir  $X_{p,j}$  ortağı seçilir ve phi ( $\phi$ ) değeri oluşturulur. Bu örnek için  $i = 3, j = 2, p = 1$  ve  $\phi = -0,64$  olarak kabul edilsin. Değiştirilen değişken Denklem (3.7)' ye göre hesaplanır

$$X'_{3,2} = 2 + -0,64 \times (2 - 6) = 4,55.$$

$X'_{3,2}$ 'nin çözüm alanı içinde olup olmadığı kontrol edilir. Bu durumda çözüm uzayının içine düştüğü için herhangi bir sınırlamaya gerek yoktur. Bu nedenle  $X'_{3,2}$ , [7 4,55] alınıp ilgili fonksiyon değeri  $f'_3 = 69,66$  ve uygunluk değeri  $fit'_3 = 0,0142$  hesaplanır.  $fit'_3 < fit_3$  olduğundan yeni çözüm kabul edilir ve iz (trail) limiti sıfırlanır.

$$P = \begin{bmatrix} 4 & 6 \\ 10 & 2 \\ 7 & 4,55 \end{bmatrix}, \quad f = \begin{bmatrix} 52 \\ 104 \\ 69,66 \end{bmatrix}, \quad fit = \begin{bmatrix} 0,0189 \\ 0,0095 \\ 0,0142 \end{bmatrix}, \quad trail = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Gözcü arı aşamasına geçmeden önce Denklem 3.8'e göre her bir çözüm için ilişkili olasılıklar hesaplanır.

$$prob = \begin{bmatrix} 0,55 \\ 1,00 \\ 0,70 \end{bmatrix}$$

Adım 3: Gözcü arı aşaması gerçekleştirilir.

Çözüm 1 için, gözcü arının mevcut besin kaynağına gidip gitmediğini kontrol etmek için 0 ile 1 arasında bir sayı ( $r = 0,18$ ) üretilir.  $r = 0,18 < 0,55$  olduğundan rastgele bir değişken seçilir ( $X_{i,j}$ ). Daha sonra rastgele bir  $X_{p,j}$  ortağı seçilir ve phi ( $\phi$ ) değeri oluşturulur. Bu örnek için  $i = 1$ ,  $j = 1$ ,  $p = 2$  ve  $\phi = -0,14$  olarak kabul edilsin.

Değiştirilen değişken Denklem (3.7)'ye göre hesaplanır

$$X'_{1,1} = 4 + -0,14 \times (4 - 10) = 4,82.$$

$X'_{1,1}$ 'nin çözüm alanı içinde olup olmadığı kontrol edilir. Bu durumda çözüm uzayının içine düştüğü için herhangi bir sınırlamaya gerek yoktur. Bu nedenle,  $X'_{1,1}$  [4,82 6] alınır ilgili fonksiyon değeri  $f'_1 = 59,2$  ve uygunluk değeri  $fit'_1 = 0,0166$  hesaplanır.  $fit'_1 < fit_1$  ise yeni çözümü korunur.  $fit'_1 < fit_1$  olduğundan yeni çözüm kabul edilir ve iz (trail) limiti sıfırlanır.

$$P = \begin{bmatrix} 4,82 & 6 \\ 10 & 2 \\ 7 & 4,55 \end{bmatrix}, \quad f = \begin{bmatrix} 59,2 \\ 104 \\ 69,66 \end{bmatrix}, \quad fit = \begin{bmatrix} 0,0166 \\ 0,0095 \\ 0,0142 \end{bmatrix}, \quad trail = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Çözüm 2 için, gözcü arının mevcut besin kaynağına gidip gitmediğini kontrol etmek için 0 ile 1 arasında bir sayı ( $r = 0,29$ ) üretilsin  $0,29 < 1$  olduğundan rastgele bir değişken seçilir ( $X_{i,j}$ ). Daha sonra, rastgele bir  $X_{p,j}$  ortağı seçilir ve phi ( $\phi$ ) değeri oluşturulur. Bu örnek için  $i = 2, j = 2, p = 1$  ve  $\phi = -0,42$  olarak kabul edilsin. Değiştirilen değişken Denklem (3.7)'ye göre hesaplanır

$$X'_{2,2} = 2 + -0,42 \times (2 - 6) = 3,66.$$

$X'_{2,2}$ 'nin çözüm alanı içinde olup olmadığı kontrol edilir. Bu durumda çözüm uzayının içine düştüğü için herhangi bir sınırlamaya gerek yoktur. Bu nedenle,  $X'_{2,2}$  [10 3,66] alınır ilgili fonksiyon değeri  $f'_2 = 113,42$  ve uygunluk değeri  $fit'_2 = 0,0087$  hesaplanır.  $fit'_2 < fit_2$  olduğundan yeni çözüm kabul edilir ve iz (trail) limiti sıfırlanır.

$$P = \begin{bmatrix} 4,82 & 6 \\ 10 & 3,66 \\ 7 & 4,55 \end{bmatrix}, \quad f = \begin{bmatrix} 59,2 \\ 113,42 \\ 69,66 \end{bmatrix}, \quad fit = \begin{bmatrix} 0,0166 \\ 0,0087 \\ 0,0142 \end{bmatrix}, \quad trail = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Çözüm 3 için, gözcü arının mevcut besin kaynağına gidip gitmediğini kontrol etmek için 0 ile 1 arasında bir sayı ( $r = 0,37$ ) üretilir.  $0,37 < 0,70$  olduğundan rastgele bir değişken seçilir ( $X_{i,j}$ ). Daha sonra, rastgele bir  $X_{p,j}$  ortağı seçilir ve phi ( $\phi$ ) değeri oluşturulur. Bu örnek için  $i = 3, j = 1, p = 1$  ve  $\phi = 0,03$  olarak kabul edilsin. Değiştirilen değişken Denklem (3.7)'ye göre hesaplanır

$$X'_{3,1} = 4,82 + 0,03 \times (4,82 - 10) = 7,06.$$

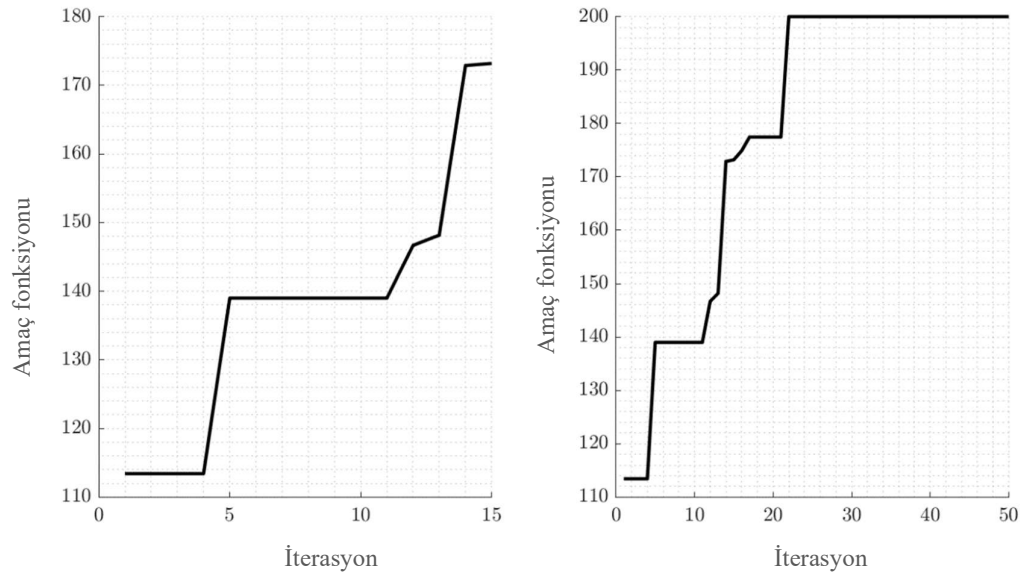
$X'_{3,1}$ 'nin çözüm alanı içinde olup olmadığı kontrol edilir. Bu durumda çözüm uzayının içine düştüğü için herhangi bir sınırlamaya gerek yoktur. Bu nedenle,  $X'_{3,1}$  [7,06 4,55] için ilgili fonksiyon değeri  $f'_3 = 70,53$  ve uygunluk değeri  $fit'_3 = 0,0140$  olur.  $fit'_3 < fit_3$  olduğundan yeni çözüm kabul edilir ve iz (trail) limiti sıfırlanır.

$$P = \begin{bmatrix} 4,82 & 6 \\ 10 & 3,66 \\ 7,06 & 4,55 \end{bmatrix}, \quad f = \begin{bmatrix} 59,2 \\ 113,42 \\ 70,53 \end{bmatrix}, \quad fit = \begin{bmatrix} 0,0166 \\ 0,0087 \\ 0,0140 \end{bmatrix}, \quad trail = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Adım 4: En iyi çözümleri ezberleyin: Bu bir maksimizasyon problemi olduğundan, en yüksek amaç fonksiyonuna sahip çözüm saklanır. En iyi çözüm [10 3,66] ve ilgili amaç fonksiyonu  $f = 113,42$ 'dir.

Adım 5: İzci arı aşaması. İzci Arı aşaması, izlerinin limiti aştığı konumlarda gerçekleştirilir. Bu durumda limit 1 olarak ayarlanır ve tüm denemeler 1'den küçük olduğundan izci arı aşamasına gerek kalmaz.

Bu işlem maksimum N iterasyon için yinelenir. Şekil 3.3 ABC'nin amaç fonksiyonunu nasıl maksimuma çıkardığını göstermektedir. Soldaki şekil, amaç fonksiyon değerinin 15 nesil sonraki ilerlemesini gösterirken, sağdaki şekil 50 nesil sonraki amaç fonksiyon değerini gösterir.



Şekil 3.3. ABC aramada amaç fonksiyonunun iterasyon sayısına göre değişimi.

### 3.4 Diferansiyel Gelişim (DE)

Diferansiyel Gelişim (DE) algoritması, Storn ve Price tarafından 1997 yılında önerilen, popülasyona dayalı bir meta-sezgisel algoritmadır. Algoritma, Darwin'in evrim teorisinden esinlenmiştir. DE, biyolojik evrimlere benzer şekilde nesiller halinde çalışır. Her nesil, her bir bireyin kromozom olarak vektörlerle temsil edildiği bir popülasyondan oluşur. Popülasyondaki her birey, mevcut problemin potansiyel bir çözümüdür. DE' nin üç operatörü vardır ve parametre uzayında en uygun çözümü bulmak için bu üç operatörü uygular. Bu operatörler mutasyon, çaprazlama ve seçimdir.

Başlangıç popülasyonu oluşturulduktan sonra gerçekleştirilecek DE' nin ilk operatörü mutasyondur. DE' de hedef (ebeveyn) vektör, donör (mutant) vektör ve iz (yavru) vektör olmak üzere üç farklı vektör vardır. Mutasyon hedef vektöre uygulanır. Mevcut bireye mutasyon uygulamak için rastgele seçilen üç farklı birey daha belirlenir. Bu üç bireyin mevcut bireyden farklı olması gerekir. Rastgele seçilen ilk birey, önceden belirlenmiş bir F faktörü ile ağırlıklandırılan diğer iki rastgele seçilen birey arasındaki farka eklenir; burada F genellikle 0 ile 2 arasındadır. Her birey bir mutasyona uğradıktan sonra bir donör vektörü oluşturulur. Mutasyon Denklem 3.9 kullanılarak gerçekleştirilir:

$$V = X_{r_1} + F(X_{r_2} - X_{r_3}) \quad (3.9)$$

$r_1, r_2$  ve  $r_3$ , rastgele seçilen bireylerin (kromozomlar) endeksleridir; burada  $r_1, r_2$  ve  $r_3 \in (1,2,3, \dots, NP)$  ve  $NP$  popülasyon sayısını temsil eder.

DE' nin ikinci operatörü rekombinasyon (çaprazlama) operatörüdür. Popülasyonun çeşitliliğini arttırmak için donör vektörü ile hedef vektörün değiştirilmesi yoluyla çaprazlama gerçekleştirilir. DE algoritmasında kullanılan çaprazlama operatörü üstel veya düzgün çaprazlama olabilir. Düzgün çaprazlamada öncelikle çaprazlama olasılığı önceden belirlenir. Bu durumda her gen (karar değişkeni) için 0 ile 1 arasında rastgele bir sayı üretilir. Benzer şekilde, 1 ile karar değişkenlerinin uzunluğu arasında

rastgele bir sayı üretilir. Çaprazlama olasılığı oluşturulan rastgele sayıyla karşılaştırılır. Rastgele sayının çaprazlama olasılığından küçük veya ona eşit olması veya üretilen sayının mevcut gen veya karar değişkenine eşit olması durumunda bu gen donör vektörden gelir, aksi halde hedef vektörden gelir. Rekombinasyonun matematiksel ifadesi Denklem 3.10 ile verilmiştir.

$$u_j = \begin{cases} v_j & r \leq P_c \text{ veya } j = j_r \\ x_j & r > P_c \text{ ve } j \neq j_r \end{cases} \quad (3.10)$$

$P_c$  parametresi çaprazlama olasılığını belirtir,  $v_j$  donör vektörden türetilen  $j$ 'inci değişkeni temsil eder ve  $x_j$ , hedef vektörden çıkarılan  $j$ 'inci değişkene karşılık gelir. Ek olarak  $j_r$  rastgele oluşturulmuş bir sayıdır. Burada  $j_r \in (1,2,3, \dots, D)$  ve  $D$ , karar değişkenlerinin uzunluğunu belirtir. Her bir kromozoma çaprazlama operatörü uygulandıktan sonra ortaya çıkan vektöre iz vektörü adı verilir.

Üçüncü DE operatörü seçim operatörüdür. Popülasyon büyüklüğünü sabit tutmak ve bir sonraki nesilde hangi vektörün hayatta kalacağını belirlemek amacıyla hedef vektör ile iz vektörü arasında ağgözlü seçim gerçekleştirilir. Bu ağgözlü seçim uygunluk fonksiyonuna göre yapılır ve Denklem 3.11 kullanılarak gerçekleştirilir:

$$x_{i,g+1} = \begin{cases} u_i, & f(u_i) \geq f(x_i) \\ x_i, & f(u_i) < f(x_i) \end{cases} \quad (3.11)$$

Temel DE algoritması şu şekilde çalışır:

1. Başlatma: Arama uzayında rastgele bir aday çözüm başlangıç popülasyonu oluşturulur.
2. Mutasyon: Diferansiyel mutasyon operatörü kullanarak mevcut çözümleri değiştirerek yeni aday çözümler oluşturulur. Bu, bir mutasyon faktörüyle ölçeklenen rastgele seçilmiş iki birey arasındaki farkı içerir.
3. Çaprazlama: Mutasyona uğramış çözümlerden elde edilen bilgiler, bir çaprazlama işlemi aracılığıyla mevcut popülasyonla birleştirilir. Bu işlem,

yeni çözümün hangi bileşenlerinin mutasyona uğramış adaydan miras alınacağını belirler.

4. Seçim: Yeni aday çözümlerin uygunluğunu değerlendirir ve gelecek nesil için popülasyonu oluşturacak bireyler seçilir. Seçim, orijinal ve mutasyona uğramış çözümlerin uygunluğunun karşılaştırılmasına dayanmaktadır.
5. Sonlandırma: Önceden tanımlanmış sayıda nesil için veya yakınsama kriteri karşılanana kadar 2-4 arasındaki adımlar tekrarlanır.

### 3.4.1 DE Örneği

Öncelikle DE kullanıcı tanımlı parametreleri ayarlanır.

- |                                     |                                   |
|-------------------------------------|-----------------------------------|
| 1. Karar değişkenlerinin sayısı (D) | : 2                               |
| 2. Amaç fonksiyonu                  | : $\text{mak} \sum_{i=1}^D x_i^2$ |
| 3. Alt – üst sınırlar               | : $0 < x_i < 10$                  |
| 4. Popülasyon hacmi                 | : 5                               |
| 5. Çaprazlama olasılığı ( $pC$ )    | : 0,8                             |
| 6. Ölçeklendirme faktörü ( $F$ )    | : 0,85                            |
| 7. İterasyon Sayısı                 | : 10                              |

Adım 1: Popülasyon rastgele oluşturulur.

$$P = \begin{bmatrix} 4 & 2 \\ 10 & 1 \\ 7 & 9 \\ 6 & 6 \\ 2 & 7 \end{bmatrix}$$

Adım 2: Uygunluk fonksiyonu hesaplanır:

$$f = \begin{bmatrix} 20 \\ 101 \\ 130 \\ 72 \\ 53 \end{bmatrix}$$

Adım 3: Denklem (3.9)' a göre mutasyon gerçekleştirilir. Çözüm 1 için üç rastgele ortak seçilir.

$$r_1 = 5, r_2 = 2, r_3 = 4$$

$$V_1 = [2 \ 7] + 0,85 \times ([10 \ 1] - [6 \ 6]) = [5,4 \ 2,75]$$

Çözüm 2 için  $r_1 = 3, r_2 = 4, r_3 = 1$  ve

$$V_2 = [7 \ 9] + ([6 \ 6] - [4 \ 2]) = [8,7 \ 12,4]$$

Çözüm 3 için  $r_1 = 2, r_2 = 1, r_3 = 4$  ve

$$V_3 = [10 \ 1] + 0,85 \times ([4 \ 2] - [6 \ 6]) = [8,3 \ -2,4]$$

Çözüm 4 için  $r_1 = 3, r_2 = 1, r_3 = 2$  ve

$$V_4 = [7 \ 9] + 0,85 \times ([4 \ 2] - [10 \ 1]) = [1,9 \ 9,85]$$

Çözüm 5 için  $r_1 = 2, r_2 = 1, r_3 = 4$  ve

$$V_5 = [10 \ 1] + 0,85 \times ([4 \ 2] - [6 \ 6]) = [8,3 \ -2,4]$$

$$V = \begin{bmatrix} 5,4 & 2,75 \\ 8,7 & 12,4 \\ 8,3 & -2,4 \\ 1,9 & 9,85 \\ 8,3 & -2,4 \end{bmatrix}$$

değerleri hesaplanır.

Adım 4: Denklem (3.10)'a göre çaprazlama gerçekleştirilir ve karar değişkeni 2 tane olduğundan  $5 \times 2$  boyutlu  $U(0,1)$  dağılımından  $r$  vektörü ve  $5 \times 1$  boyutlu  $DU(1,2)$  (kesikli düzgün dağılımdan)  $\delta$  vektörü üretilir.

$$r = \begin{bmatrix} 0,52 & 0,43 \\ 0,20 & 0,51 \\ 0,81 & 0,30 \\ 0,26 & 0,66 \\ 0,89 & 0,60 \end{bmatrix}, \quad j = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}, \quad \delta = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 2 \\ 2 \end{bmatrix}$$

Eğer  $r < pC$  veya  $j = \delta$  ise donör vektör ( $V$ ), aksi halde hedef vektör ( $P$ ) elemanlarından

$U$  oluşturulur.

$$U = \begin{bmatrix} 5,4 & 2,75 \\ 8,7 & 12,4 \\ 7 & -2,4 \\ 1,9 & 9,85 \\ 2 & -2,4 \end{bmatrix}$$

Alt-üst sınırların dışında kalan değişkenler değerleri kontrol edilir. Sınır dışındakiler yakın oldukları sınır değerlerine atanır.

$$U = \begin{bmatrix} 5,4 & 2,75 \\ 8,7 & 10 \\ 7 & 0 \\ 1,9 & 9,85 \\ 2 & 0 \end{bmatrix}$$

Adım 5: Uygunluk değerleri hesaplanır.

$$f_u = \begin{bmatrix} 36,72 \\ 175,69 \\ 49 \\ 100,63 \\ 4 \end{bmatrix}$$

Adım 6: Açgözlü seçim gerçekleştirilir.  $f_u > f$  ise, iz (trail) vektöründen ( $U$ ) gelen çözümü, aksi takdirde  $P$  hedef (target) vektöründen çözümü kabul edilir.

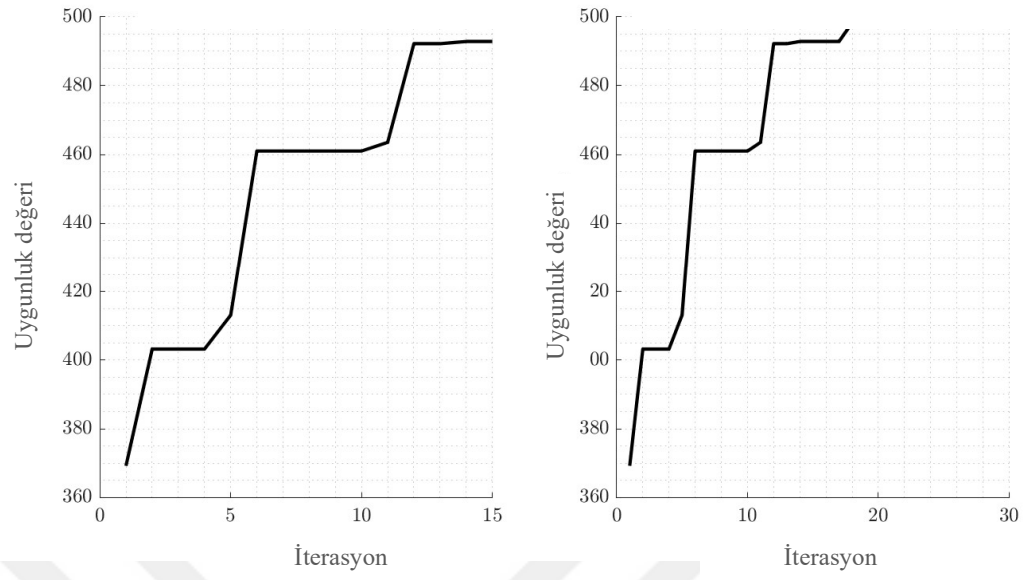
$$f_u = \begin{bmatrix} 36,72 \\ 175,69 \\ 49 \\ 100,63 \\ 4 \end{bmatrix}, \quad f = \begin{bmatrix} 20 \\ 101 \\ 130 \\ 72 \\ 53 \end{bmatrix}$$

$$P = \begin{bmatrix} 5,4 & 2,75 \\ 8,7 & 10 \\ 7 & 9 \\ 1,9 & 9,85 \\ 2 & 7 \end{bmatrix}$$

Çözüm 1, 2 ve 4 iz vektöründen, çözüm 3 ve 5 ise hedef vektörden oluşturulur.

Adım 7: Her yineleme için en iyi çözümü saklanır. Bu yinelemede en iyi çözüm, en yüksek uygunluk değerine sahip olduğundan  $P_2'$  dir.

Bu işlem maksimum  $N$  iterasyon için yinelenir. Şekil 3.4 bir DE algoritmasının amaç fonksiyonunun değerlerini göstermektedir. Soldaki şekil, uygunluk değerinin 15 nesil sonraki ilerlemesini gösterirken, sağdaki şekil 30 nesil sonraki uygunluk değerini gösterir. Şekil 3.4' e dikkat edilirse, amaç fonksiyonunun 5 değişken ve 10 kromozomla maksimum çözümü bulunduğu görülür.



Şekil 3.4. DE aramada amaç fonksiyonunun uygunluk değerlerinin iterasyon sayısına göre değişimi.

#### 4. DÖNÜŞTÜRÜLMÜŞ DAĞILIMLAR

Dönüştürülmüş dağılımları incelemeye önce bazı temel istatistiksel kavramları gözden geçirmek faydalı olacaktır.  $X$  sürekli değişken olmak üzere  $X$ 'in olasılık yoğunluk fonksiyonu (oyf) genellikle  $f(x)$  ile gösterilir.  $X$ 'in oyf  $f(x)$  bir olasılık değeri değil,  $X$ 'in belirli bir değerine karşılık gelen yükseklik değeridir.  $X$ 'in oyf  $f(x)$ 'in belirli bir  $(a, b)$  aralığı boyunca integrali,  $X$ 'in bu aralık içinde olma olasılığını temsil eder.

$$P(a \leq X \leq b) = \int_a^b f(x) dx \quad (4.1)$$

Rastgele değişkenin geçerli bir oyf aşağıdaki koşulları karşılamalıdır:

$$(1) \quad f(x) \geq 0,$$

$$(2) \quad \int_{-\infty}^{\infty} f(x) dx = 1.$$

$F(x)$  ile gösterilen birikimli dağılım fonksiyonu, sürekli bir rastgele değişken  $X$ 'in belirli bir  $x$  değerinden küçük veya ona eşit olma olasılığını tanımlar. Matematiksel olarak

$$F(x) = P(X \leq x) = \int_{-\infty}^x f(x) dx \quad (4.2)$$

şeklinde tanımlanır.

Güvenilirlik (yaşam) fonksiyonu  $R(x)$  bir üretimin belirli bir zaman diliminde en az belirtilen süre arızalanmadan çalışmaya devam etme olasılığını verir ve matematiksel olarak  $F(x)$ 'in tamamlayıcısıdır.

$$R(x) = 1 - F(x) \quad (4.3)$$

Bozulma oranı  $h(x)$ ,  $f(x)$ 'in güvenilirlik fonksiyonu  $R(x)$ 'e oranıdır. En az  $x$  süre çalıştığı göz önüne alındığında, bir nesnenin  $x + \delta$  çalışma oranını verir. Matematiksel olarak:

$$\begin{aligned}
h(x) &= P(x < X < x + \delta | X > x) = \frac{P(X < x + \delta, X > x)}{P(X > x)} \\
&= \frac{P(X < x + \delta) - P(X < x)}{P(X > x)} \\
h(x) &= \frac{F(x + \delta) - F(x)}{1 - F(x)} = \frac{f(x)}{R(x)} \tag{4.4}
\end{aligned}$$

Bir diğ er önemli kavram ise bir rastgele de ğ iş kenin maksimum ve minimum sıra istatistiklerinin dağılımıdır.  $X_1$  ve  $X_2$  iki bağımsız ve aynı şekilde dağılmış rastgele de ğ iş ken olsun.  $Y_1$ ,  $X_1$  ve  $X_2$ 'nin minimumu,  $Y_2$  ise  $X_1$  ve  $X_2$ 'nin maksimumu olsun.  $Y_1$  ve  $Y_2$  iki yeni rastgele de ğ iş kendir ve bu iki rastgele de ğ iş kenin sıra istatistiklerinin ( $Y_1$  ve  $Y_2$ )'nin dağılımını bulalım.  $F(y_1)$   $Y_1$ 'in ve  $F(y_2)$   $Y_2$ 'nin dağılım fonksiyonları olsun.  $F_{1:2}(y_1)$  1. sıra istatistiğ in dağılımı olsun ve  $F_{2:2}(y_2)$   $n$ . sıra istatistiğ in dağılımı olmak üzere,

$$F_{1:2}(y_1) = P(Y_1 < y_1) = 1 - P(Y_1 > y_1)$$

$$F_{1:2}(y_1) = 1 - P(\min(X_1, X_2) > y_1)$$

$$F_{1:2}(y_1) = 1 - [P(X_1 > y_1), P(X_2 > y_1)] = 1 - P(X_1 > y_1) \times P(X_2 > y_1)$$

$$F_{1:2}(y_1) = 1 - (1 - F(y_1)) \times (1 - F(y_1))$$

$$F_{1:2}(y_1) = 1 - [1 - F(y_1)]^2 \tag{4.5}$$

$$F_{2:2}(y_2) = P(Y_2 < y_2) = P(\max(X_1, X_2) < y_2)$$

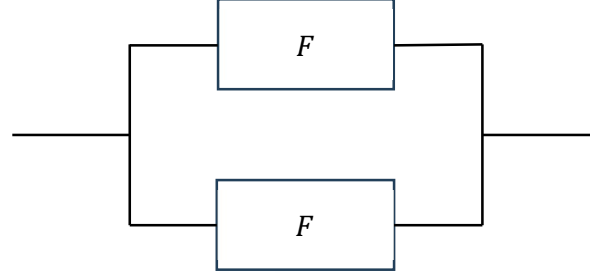
$$F_{2:2}(y_2) = P(X_1 < y_2, X_2 < y_2) = P(X_1 < y_2) \times P(X_2 < y_2)$$

$$F_{2:2} = F(y_2) \times F(y_2) = F(y_2)^2 \tag{4.6}$$

şeklinde elde edilir.

Paralel veya seri konfigürasyonda düzenlenmiş,  $X_1$  ve  $X_2$  olarak gösterilen iki bileşenden oluşan bir sistem düşünelim. Bu bileşenlerin ömürlerinin temel dağılım fonksiyonu  $F(x)$  olsun. Paralel bir sistemde (Şekil 4.1), en az bir bileşen çalışır durumda

kaldığı sürece çalışır. Tersine, bir seri sistemin (Şekil 4.2) bileşenlerinden herhangi biri arızalandığında sistem bozulur.



Şekil 4.1. İki bileşenli paralel sistem



Şekil 4.2. İki bileşenli seri sistem

Paralel bir sistemin bozulma zamanı, bileşenlerden en uzun süre çalışanın ömrü yani maksimum bozulma zamanının dağılımı kullanılarak ifade edilebilir  $\max(X_1, X_2) = F(x)^2$ . Seri bir sistemin bozulma zamanı ise bileşenlerden ilk bozulan parçanın çalışma süresi yani minimum bozulma zamanı dağılımı ile temsil edilir  $\min(X_1, X_2) = 1 - [1 - F(x)]^2 = 2F(x) - F(x)^2$ . Bu durumda, sistemin bozulma zamanı dağılımı  $F(x)^2$  ile  $2F(x) - F(x)^2$  arasında yer alır. Bu ifade  $F(x)^2 < F < 2F(x) - F(x)^2$  şeklinde yazılabilir. Dolayısıyla  $F, F(x)^2$  ve  $2F(x) - F(x)^2$ 'nin konveks birleşimi olarak ifade edilebilir.

$$F_Y(y) = \alpha \times P(X_{1:2} \leq x) + (1 - \alpha) \times P(X_{2:2} \leq x)$$

Burada  $\alpha, [0,1]$  aralığında bulunan bir konveks birleşim parametresidir.

$$\begin{aligned} F_Y(y) &= \alpha \times (1 - [1 - F_X(x)]^2) + (1 - \alpha) \times [F_X(x)]^2 \\ &= \alpha(2F_X(x) - F_X(x)^2) + (1 - \alpha)F_X(x)^2 \\ &= 2\alpha F_X(x) - \alpha F_X(x)^2 + F_X(x)^2 - \alpha F_X(x)^2 \\ &= 2\alpha F_X(x) + F_X(x)^2 - 2\alpha F_X(x)^2 \end{aligned}$$

$$2\alpha F_X(x) - (2\alpha - 1)F_X(x)^2$$

burada  $\lambda = 2\alpha - 1$  alınırsa,

$$F_Y(y) = (1 + \lambda)F_X(x) - \lambda F_X(x)^2 \quad (4.7)$$

olup bu denkleme Shaw ve Buckley tarafından 2009'da önerildiği gibi karesel rank dönüşümü denir. Olasılık yoğunluk fonksiyonu, (4.7) eşitliğinin türevi alınarak

$$\begin{aligned} f_Y(y) &= f(x)(1 + \lambda) - 2\lambda f(x)F(x) \\ f_Y(y) &= f(x)[1 + \lambda - 2\lambda F(x)] \end{aligned} \quad (4.8)$$

şeklinde elde edilir.

#### 4.1 Dönüştürülmüş Düzgün Dağılım

Shaw ve Buckley (2009) çarpık-düzgün dağılım elde etmek için karesel rank dönüşümünü düzgün dağılıma uygulamıştır.  $F_1(x)$  ve  $F_2(x)$  sırasıyla temel ve dönüştürülmüş dağılımların dağılım fonksiyonları olsun. Benzer şekilde  $f_1(x)$  ve  $f_2(x)$  de yine sırasıyla temel ve dönüştürülmüş dağılımların yoğunluk fonksiyonları olsun. Düzgün  $U(0,1)$  dağılımın yoğunluk ve dağılım fonksiyonları sırasıyla

$$\begin{aligned} f_1(x) &= \begin{cases} 1 & 0 \leq x \leq 1 \\ 0 & d. y. \end{cases} \\ F_1(x) &= \begin{cases} 0, & x < 0 \\ x, & 0 \leq x \leq 1 \\ 1, & x > 1 \end{cases} \end{aligned}$$

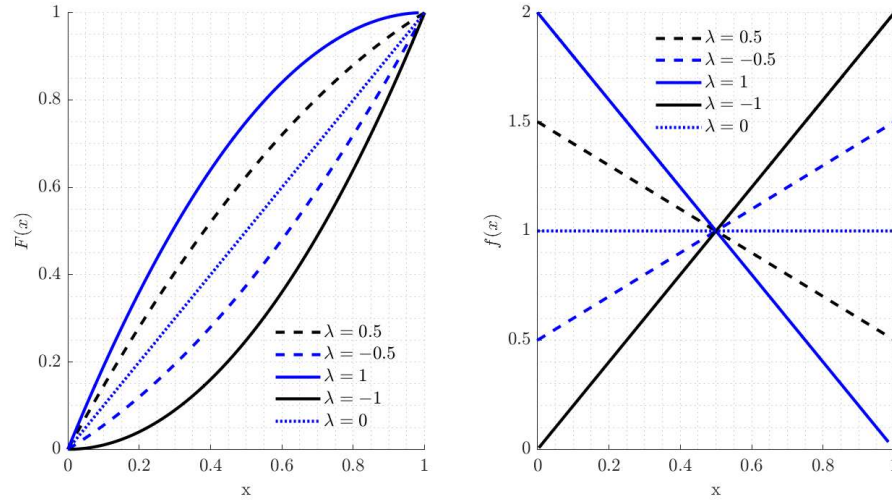
şeklinde dir. Dönüştürülmüş yoğunluk ve dağılım fonksiyonları eşitlik (4.8) ve (4.7)

kullanılarak sırasıyla

$$f(x) = \begin{cases} (1 + \lambda - 2\lambda x), & 0 \leq x \leq 1, \\ 0, & d. y. \end{cases} \quad -1 \leq \lambda \leq 1 \quad (4.9)$$

$$F_2(x) = \begin{cases} 0 & x < 0 \\ x(1 + \lambda) - \lambda x^2 & 0 \leq x \leq 1 \\ 1 & x > 1 \end{cases} \quad (4.10)$$

şeklinde bulunur. Farklı lambda değerleri için dönüştürülmüş düzgün dağılıma ait dağılım ve olasılık yoğunluk fonksiyonu çizimleri Şekil 4.3'te verilmiştir.



Şekil 4.3. Farklı  $\lambda$  değerleri için dönüştürülmüş düzgün dağılımın  $F(x)$  ve  $f(x)$  grafikleri

#### 4.2 Dönüştürülmüş Üstel Dağılım

Owoloko vd. (2015) üstel dağılıma karesel rank dönüşümünü uygulayarak dönüştürülmüş üstel dağılımı önermiştir. Üstel dağılımın olasılık yoğunluk ve dağılım fonksiyonları sırasıyla

$$f_1(x) = \frac{1}{\theta} \exp\left(-\frac{x}{\theta}\right), \quad x \geq 0, \quad \theta > 0$$

$$F_1(x) = 1 - \exp\left(-\frac{x}{\theta}\right)$$

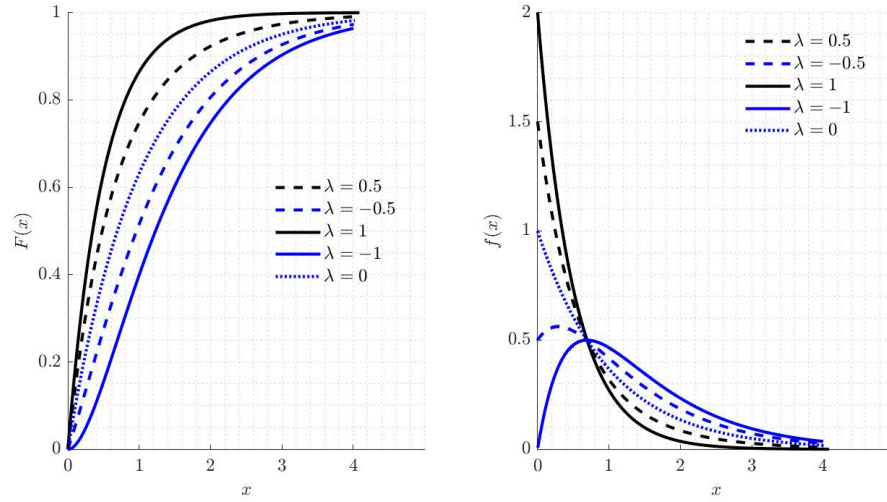
şekilde verilir. Burada  $\theta$  ölçek parametresidir.

Dönüştürülmüş üstel dağılımın olasılık yoğunluk ve dağılım fonksiyonları sırasıyla

$$f_2(x) = \frac{1}{\theta} \exp\left(-\frac{x}{\theta}\right) \left[1 - \lambda + 2\lambda \exp\left(-\frac{x}{\theta}\right)\right] \quad (4.11)$$

$$F_2(x) = \left[1 - \exp\left(-\frac{x}{\theta}\right)\right] \left[1 + \lambda \exp\left(-\frac{x}{\theta}\right)\right] \quad (4.12)$$

şeklinde yazılabilir. Burada  $x > 0$ ,  $\theta > 0$  ve  $\lambda \leq |1|$ 'dir



Şekil 4.4. Dönüştürülmüş üstel dağılımın farklı  $\lambda$  değerleri için  $F(x)$  ve  $f(x)$  grafikleri

Dönüştürülmüş üstel dağılımın  $r$ 'inci dereceden momenti

$$E(X^r) = \theta^r \Gamma(1 + r) [1 - \lambda + \lambda 2^{-r}] \quad (4.13)$$

olup ortalama ve varyans eşitlik (4.13) kullanılarak

$$E(X) = \theta \left( \frac{2 - \lambda}{2} \right) \quad (4.14)$$

ve

$$Var(X) = \theta^2 \left[ \left( \frac{4 - 3\lambda}{2} \right) - \left( \frac{4 - 4\lambda - \lambda^2}{4} \right) \right] \quad (4.15)$$

şeklinde hesaplanabilir. Güvenilirlik ve bozulma oranı fonksiyonları ise sırasıyla

$$R(t) = \lambda \exp\left(-\frac{2t}{\theta}\right) - (\lambda - 1) \exp\left(-\frac{t}{\theta}\right) \quad (4.16)$$

$$h(t) = \frac{\frac{1}{\theta} (1 - \lambda + 2\lambda \exp(-\frac{t}{\theta}))}{(\lambda \exp(-\frac{t}{\theta}) + 1 - \lambda)} \quad (4.17)$$

şeklinindedir. Dönüştürülmüş üstel dağılımın parametre tahmini için EÇÖ tahmininde kullanılan Log-olabilirlik fonksiyonu ise

$$l = -n \ln \theta - \sum_{i=1}^n \left( \frac{x_i}{\theta} \right) + \ln \left[ 1 - \lambda + 2\lambda \exp \left( -\frac{x_i}{\theta} \right) \right] \quad (4.18)$$

şekildedir. Eşitlik (4.18)'i sırasıyla  $\theta$  ve  $\lambda$ 'ya göre türevi alınarak

$$\frac{\partial l}{\partial \theta} = \frac{1}{\theta} \sum_{i=1}^n \left[ 1 - \left( \frac{x_i}{\theta} \right) \right] + \frac{2\lambda}{\theta} \sum_{i=1}^n \frac{\frac{x_i}{\theta} \exp \left( -\frac{x_i}{\theta} \right)}{\left[ 1 - \lambda + 2\lambda \exp \left( -\frac{x_i}{\theta} \right) \right]} \quad (4.19)$$

$$\frac{\partial l}{\partial \lambda} = \sum_{i=1}^n \frac{2 \exp \left( -\frac{x_i}{\theta} \right) - 1}{\left[ 1 - \lambda + 2\lambda \exp \left( -\frac{x_i}{\theta} \right) \right]} \quad (4.20)$$

elde edilen (4.19) ve (4.20) eşitliklerinden oluşan doğrusal olmayan denklem sisteminin çözümü, bilinmeyen  $\theta$  ve  $\lambda$  parametrelerinin en çok olabilirlik tahminlerini verir. Ters dönüşüm yöntemi kullanılarak, dönüştürülmüş üstel dağılımdan rastgele sayılar  $u \sim U(0,1)$  olmak üzere Eşitlik (4.21) kullanılarak elde edilebilir.

$$x = \theta \left[ -\ln \left\{ 1 - \left( \frac{1 + \lambda - \sqrt{(1 + \lambda)^2 - 4\lambda u}}{2\lambda} \right) \right\} \right] \quad (4.21)$$

### 4.3 Dönüştürülmüş Weibull dağılım

Aryal ve Tsokos (2011), karesel dönüşüm yöntemini kullanarak iki parametrelili Weibull dağılımını genelleştirerek, dönüştürülmüş Weibull dağılımı adı verilen yeni bir olasılık dağılımını ortaya koydu. İki parametrelili Weibull dağılımının birikimli dağılım fonksiyonu ve olasılık yoğunluk fonksiyonu aşağıdaki şekilde verilmektedir:

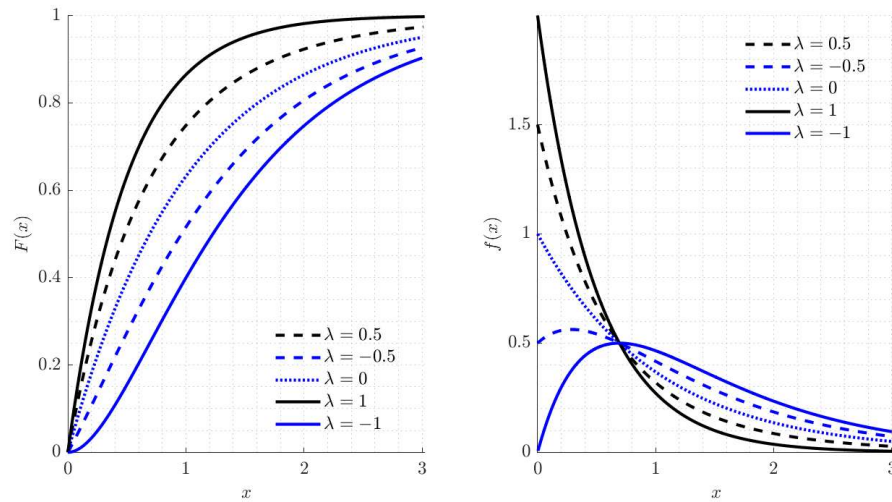
$$f_1(x) = \frac{\eta}{\sigma} \left( \frac{x}{\sigma} \right)^{\eta-1} \exp \left( -\left( \frac{x}{\sigma} \right)^\eta \right), \quad x > 0$$

$$F_1(x) = 1 - \exp\left(-\left(\frac{x}{\sigma}\right)^\eta\right)$$

Aryal ve Tsokos, karesel dönüşüm yöntemini kullanarak dönüştürülmüş Weibull dağılımı için Eşitlik (4.22) ve (4.23)'te verilen dağılım ve olasılık yoğunluk fonksiyonlarını elde etti:

$$F_2(x) = \left[1 - \exp\left(-\left(\frac{x}{\sigma}\right)^\eta\right)\right] \left[1 + \lambda \exp\left(-\left(\frac{x}{\sigma}\right)^\eta\right)\right] \quad (4.22)$$

$$f_2(x) = \frac{\eta}{\sigma} \left(\frac{x}{\sigma}\right)^{\eta-1} \exp\left(-\left(\frac{x}{\sigma}\right)^\eta\right) \left[1 - \lambda + 2\lambda \exp\left(-\left(\frac{x}{\sigma}\right)^\eta\right)\right] \quad (4.23)$$



Şekil 4.5. Dönüştürülmüş Weibull dağılımının farklı  $\lambda$  değerleri için  $F(x)$  ve  $f(x)$  grafikleri

Aryal vd. (2011),  $\eta = 1$  için dağılımın dönüştürülmüş bir üstel dağılım haline geldiğini,  $\lambda = 0$  için Weibull dağılımına indirgenildiğini ve  $\eta = \lambda = 1$  olduğunda ortaya çıkan dağılımın  $\sigma/2$  parametresi ile üstel bir dağılım olduğunu belirtmiştir.

Dönüştürülmüş Weibull dağılımının momentleri Eşitlik (4.24)'te verilmektedir.

$$E(X^k) = \sigma^k \Gamma\left(1 + \frac{k}{\eta}\right) \left(1 - \lambda + \lambda 2^{\frac{k}{\eta}}\right) \quad (4.24)$$

Ortalama ve varyans yukarıdaki Denklem (4.24) kullanılarak elde edilebilir.

$$E(X) = \sigma \Gamma\left(1 + \frac{1}{\eta}\right) \left(1 - \lambda + \lambda 2^{\frac{1}{\eta}}\right) \quad (4.25)$$

$$Var(X) = \sigma^2 \left\{ \Gamma\left(1 + \frac{2}{\eta}\right) \left(1 - \lambda + \lambda 2^{\frac{2}{\eta}}\right) - \Gamma\left(1 + \frac{1}{\eta}\right) \left(1 - \lambda + \lambda 2^{\frac{1}{\eta}}\right)^2 \right\} \quad (4.26)$$

Aryal ve ark. (2011), dönüştürülmüş Weibull dağılımının parametrelerini tahmin etmek için EÇO tahminini kullanmıştır. Bu yöntem doğrusal olmayan modellerde parametre tahmini için kullanılan yaygın bir yöntemdir. Log-olabilirlik fonksiyonunun bilinmeyen parametrelere göre türevlerinin sıfıra eşitlenmesiyle elde edilen denklemler doğrusal olmadığından genelde bilinmeyen parametreleri çözmek için Quasi-Newton algoritması adı verilen yinelemeli bir yöntem kullanılmaktadır.

Dönüştürülmüş Weibull Dağılımının log olabilirlik fonksiyonu,

$$\begin{aligned} l &= n \ln \frac{\eta}{\sigma} - \sum_{i=1}^n \left(\frac{x_i}{\sigma}\right)^{\eta} + \sum_{i=1}^n \left(\frac{x_i}{\sigma}\right)^{\eta-1} + \sum_{i=1}^n \left[1 - \lambda + 2\lambda \exp\left(-\frac{x_i}{\sigma}\right)^{\eta}\right] \\ l &= n \ln \eta - n\eta \ln \sigma + (\eta - 1) \sum_{i=1}^n \ln(x_i) - \sum_{i=1}^n \left(\frac{x_i}{\sigma}\right)^{\eta} \\ &\quad + \sum_{i=1}^n \left[1 - \lambda + 2\lambda \exp\left(-\frac{x_i}{\sigma}\right)^{\eta}\right] \end{aligned} \quad (4.27)$$

olup, bilinmeyen  $\lambda$ ,  $\eta$  ve  $\sigma$  için EÇO tahmin edicileri log-olabilirlik fonksiyonunun maksimize edilmesiyle elde edilir. Denklem (4.27)'nin  $\eta$ ,  $\sigma$  ve  $\lambda$ 'ya göre kısmi türevleri alınırsa normal denklemler Eşitlik (4.28-4.30)'da görüldüğü gibi elde edilir.

$$\frac{\partial l}{\partial \eta} = \frac{n}{\eta} + \sum_{i=1}^n \left[1 - \left(\frac{x_i}{\sigma}\right)^{\eta}\right] \ln\left(\frac{x_i}{\sigma}\right) - 2\lambda \sum_{i=1}^n \frac{\ln\left(\frac{x_i}{\sigma}\right) \left(\frac{x_i}{\sigma}\right)^{\eta} \exp\left(-\left(\frac{x_i}{\sigma}\right)^{\eta}\right)}{\left[1 - \lambda + 2\lambda \exp\left(-\left(\frac{x_i}{\sigma}\right)^{\eta}\right)\right]} = 0 \quad (4.28)$$

$$\frac{\partial l}{\partial \sigma} = -\frac{\eta}{\sigma} \sum_{i=1}^n \left[ 1 - \left( \frac{x_i}{\sigma} \right)^\eta \right] + \frac{2\lambda\eta}{\sigma} \sum_{i=1}^n \frac{\left( \frac{x_i}{\sigma} \right)^\eta \exp\left(-\left(\frac{x_i}{\sigma}\right)^\eta\right)}{\left[ 1 - \lambda + 2\lambda \exp\left(-\left(\frac{x_i}{\sigma}\right)^\eta\right) \right]} = 0 \quad (4.29)$$

$$\frac{\partial l}{\partial \lambda} = \sum_{i=1}^n \frac{2 \exp\left(-\left(\frac{x_i}{\sigma}\right)^\eta\right) - 1}{\left[ 1 - \lambda + 2\lambda \exp\left(-\left(\frac{x_i}{\sigma}\right)^\eta\right) \right]} = 0 \quad (4.30)$$

Weibull dağılımından rastgele sayılar  $u \sim U(0,1)$  olmak üzere Denklem (4.31) kullanılarak ters dönüşüm yöntemi ile elde edilir.

$$x = \sigma \left[ -\ln \left\{ 1 - \left( \frac{1 + \lambda - \sqrt{(1 + \lambda)^2 - 4u\lambda}}{2\lambda} \right) \right\} \right]^{\frac{1}{\eta}} \quad (4.31)$$

Güvenilirlik ve bozulma oranı fonksiyonları sırasıyla Eşitlik (4.32) ve Eşitlik (4.33)'te verilmiştir.

$$R(t) = \exp\left(-\left(\frac{t}{\sigma}\right)^\eta\right) \left[ 1 - \lambda + \lambda \exp\left(-\left(\frac{t}{\sigma}\right)^\eta\right) \right] \quad (4.32)$$

$$h(t) = \frac{\eta}{\sigma} \left(\frac{t}{\sigma}\right)^\eta \left\{ \frac{1 - \lambda + 2\lambda \exp\left(-\left(\frac{t}{\sigma}\right)^\eta\right)}{1 - \lambda + \lambda \exp\left(-\left(\frac{t}{\sigma}\right)^\eta\right)} \right\} \quad (4.33)$$

#### 4.4 Dönüştürülmüş Fréchet Dağılım

Mahmoud ve Mandouh (2013), karesel rank dönüşümünü uygulayarak dönüştürülmüş Fréchet dağılımını tanıttı. Fréchet dağılımı için olasılık yoğunluk ve birikimli dağılım fonksiyonları sırasıyla verilmiştir.

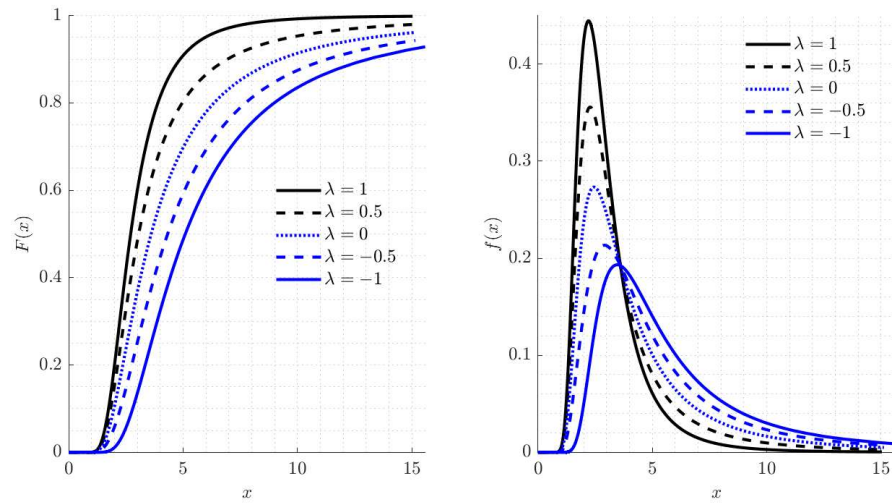
$$f_1(x) = \mu \sigma^\mu x^{-(\mu+1)} \exp\left(-\left(\frac{\sigma}{x}\right)^\mu\right)$$

$$F_1(x) = \exp\left(-\left(\frac{\sigma}{x}\right)^\mu\right)$$

Bu fonksiyonlara Shaw ve Buckley (2009) tarafından önerilen dönüşüm uygulanarak Eşitlik (4.34) ve (4.35)'te verilen Dönüştürülmüş Fréchet Dağılımının olasılık yoğunluk ve dağılım fonksiyonlarını elde etmişlerdir.

$$f_2(x) = \frac{\mu}{\sigma} \left(\frac{\sigma}{x}\right)^{\mu+1} \exp\left(-\left(\frac{\sigma}{x}\right)^\mu\right) \left[1 + \lambda - 2\lambda \exp\left(-\left(\frac{\sigma}{x}\right)^\mu\right)\right] \quad (4.34)$$

$$F_2(x) = \exp\left(-\left(\frac{\sigma}{x}\right)^\mu\right) \left[1 + \lambda - \lambda \exp\left(-\left(\frac{\sigma}{x}\right)^\mu\right)\right] \quad (4.35)$$



Şekil 4.6. Dönüştürülmüş Fréchet dağılımın farklı  $\lambda$  değerleri için  $F(x)$  ve  $f(x)$  grafikleri

Dönüştürülmüş Fréchet Dağılımı için Eşitlik (4.36)'da verilen  $r$ .moment kullanılarak Eşitlik (4.37) ve Eşitlik (4.38)'de  $E(X)$  ve  $Var(X)$  değerleri hesaplanmıştır.

$$E(X^r) = \sigma^r \Gamma\left(1 - \frac{r}{\mu}\right) \left[(1 + \lambda) - \lambda 2^{\frac{r}{\mu}}\right], 1 - \frac{r}{\mu} > 0 \quad (4.36)$$

Ortalama ve varyans yukarıdaki Eşitlik (4.36)'dan hesaplanabilir:

$$E(X) = \sigma \Gamma\left(1 - \frac{1}{\mu}\right) \left[(1 + \lambda) - \lambda 2^{\frac{1}{\mu}}\right] \quad (4.37)$$

$$Var(X) = \sigma^2 \left\{ \Gamma \left( 1 - \frac{2}{\mu} \right) \left[ 1 + \lambda - \lambda 2^{\frac{2}{\mu}} \right] - \Gamma^2 \left( 1 - \frac{1}{\mu} \right) \left[ 1 + \lambda - \lambda 2^{\frac{1}{\mu}} \right] \right\}, \mu > 2 \quad (4.38)$$

Güvenilirlik ve bozulma oranı fonksiyonları ise sırasıyla Eşitlik (4.39) ve (4.40)'ta verilmiştir.

$$R(x) = 1 - \left[ \exp \left( - \left( \frac{\sigma}{x} \right)^\mu \left[ 1 + \lambda - \lambda \exp \left( - \left( \frac{\sigma}{x} \right)^\mu \right) \right] \right) \right] \quad (4.39)$$

$$h(x) = \frac{\frac{\mu}{\sigma} \left( \frac{\sigma}{x} \right)^{\mu+1} \exp \left( - \left( \frac{\sigma}{x} \right)^\mu \right) \left[ 1 + \lambda - 2\lambda \exp \left( - \left( \frac{\sigma}{x} \right)^\mu \right) \right]}{1 - \left[ \exp \left( - \left( \frac{\sigma}{x} \right)^\mu \left[ 1 + \lambda - \lambda \exp \left( - \left( \frac{\sigma}{x} \right)^\mu \right) \right] \right) \right]} \quad (4.40)$$

Dönüştürülmüş Fréchet Dağılımından rastgele sayılar ters dönüşüm yöntemi kullanılarak,  $u \sim U(0,1)$  olmak üzere Eşitlik (4.41) ile üretilebilir.

$$x = \sigma \left[ - \ln \left( \frac{1 + \lambda - \sqrt{(1 + \lambda)^2 - 4\lambda u}}{2\lambda} \right) \right]^{\frac{1}{\mu}} \quad (4.41)$$

Dönüştürülmüş Fréchet dağılımı için log-olabilirlik fonksiyonu,

$$l = n \ln \mu - n \ln \sigma - \sum_{i=1}^n \left( \frac{\sigma}{x_i} \right)^\mu + (\mu + 1) \sum_{i=1}^n \ln \left( \frac{\sigma}{x_i} \right) + \sum_{i=1}^n \ln \left[ 1 + \lambda - 2\lambda \exp \left( - \left( \frac{\sigma}{x_i} \right)^\mu \right) \right] \quad (4.42)$$

olup, bilinmeyen  $\lambda, \mu$  ve  $\sigma$  için EÇÖ tahmini log-olabilirlik fonksiyonunun maksimize edilmesiyle elde edilir. Eşitlik (4.42)'nin  $\mu, \sigma$  ve  $\lambda$ 'ya göre sırasıyla kısmi türevleri alınarak normal denklemler Eşitlik (4.43-4.45)'te verilmiştir.

$$\frac{\partial l}{\partial \mu} = \frac{n}{\mu} - \sum_{i=1}^n \left( \frac{\sigma}{x_i} \right)^\mu \ln \left( \frac{\sigma}{x_i} \right) + \sum_{i=1}^n \ln \left( \frac{\sigma}{x_i} \right) + \sum_{i=1}^n \frac{2\lambda \exp \left( - \left( \frac{\sigma}{x_i} \right)^\mu \right) \left( \frac{\sigma}{x_i} \right)^\mu \ln \left( \frac{\sigma}{x_i} \right)}{\left[ 1 + \lambda - 2\lambda \exp \left( - \left( \frac{\sigma}{x_i} \right)^\mu \right) \right]} \quad (4.43)$$

$$\frac{\partial l}{\partial \sigma} = -\frac{n}{\sigma} - \sum_{i=1}^n \frac{\mu}{\sigma} \left(\frac{\sigma}{x_i}\right)^\mu + (\mu + 1) \frac{n}{\sigma} + \sum_{i=1}^n \frac{2\lambda \frac{\mu}{\sigma} \left(\frac{\sigma}{x_i}\right)^\mu \exp\left(-\left(\frac{\sigma}{x_i}\right)^\mu\right)}{\left[1 + \lambda - 2\lambda \exp\left(-\left(\frac{\sigma}{x_i}\right)^\mu\right)\right]} \quad (4.44)$$

$$\frac{\partial l}{\partial \lambda} = \frac{1 - 2\lambda \exp\left(-\left(\frac{\sigma}{x_i}\right)^\mu\right)}{\left[1 + \lambda - 2\lambda \exp\left(-\left(\frac{\sigma}{x_i}\right)^\mu\right)\right]} \quad (4.45)$$

Normal denklemlerin ortak çözümü analitik olarak elde edilmediğinden iteratif çözüm yöntemleri kullanılarak parametre tahminleri elde edilir.

## 5. SİMÜLASYON VE BULGULAR

Bu bölümde, dönüştürülmüş üç dağılım için farklı parametre tahmin algoritmalarının performansını değerlendirmeyi amaçlayan kapsamlı bir simülasyon çalışması sunulmaktadır: Dönüştürülmüş üstel, Weibull ve Frechet dağılımlarının parametre tahminleri elde edilirken algoritmaların farklı senaryolar altındaki davranışını incelemek için dönüşüm parametresi ( $\lambda$ )  $-1, -0,5, 0,5$  ve  $1$  değerleri özel olarak seçilerek farklı durumlar incelenmiştir. Örnek boyutunun ( $n$ ) parametre tahmin doğruluğu üzerindeki etkisini incelemek için her senaryoda üç farklı örnek boyutu ( $50, 100$  ve  $200$ ) dikkate alınmıştır. Simülasyon çalışmasında incelenen özellik dışında diğer tüm parametreler sabit tutulmuştur. Her algoritmanın performansı log olabilirlik ( $l$ ), Mutlak Hatalar Yüzde Ortalaması (MAPE), yan ve hata kareler ortalaması (MSE) kullanılarak değerlendirilmiştir. En düşük MSE ve yanlılığa sahip algoritma, en iyi performansa sahip algoritma olarak kabul edilmiştir. Tüm simülasyonlar MATLAB R2022b programı kullanılarak gerçekleştirilmiştir. Her simülasyonda tekrar sayısı 2000 olacak şekilde belirlenmiştir.

$$MAPE(\hat{\beta}_i) = \frac{1}{p} \sum_{i=1}^p \frac{|\beta - \hat{\beta}|_i}{|\beta_i|} \quad (5.1)$$

$$MAPE = \frac{1}{r} \sum_{i=1}^r MAPE(\hat{\beta}_i) \quad (5.2)$$

$$yan(\hat{\beta}) = E(\hat{\beta}) - \beta \quad (5.3)$$

$$MSE(\hat{\beta}) = E[(\hat{\beta} - \beta)^2] = Var(\hat{\beta}) + [yan(\hat{\beta})]^2 \quad (5.4)$$

Eşitliklerde  $\beta$  gerçek parametre değerini,  $\hat{\beta}$  parametre tahminini,  $r$  tekrar sayısını ve  $p$  ise parametre sayısını belirtir.

Simülasyonda kullanılan her bir meta-sezgisel algoritma tarafından kullanılan önceden tanımlanmış parametreler Tablo 1'de sunulmaktadır.

**Çizelge 5.1.** Meta-Sezgisel Algoritmalar için Önceden Tanımlanmış Parametreler

<b>Algoritma</b>	<b>Parametreler</b>
GA	Popülasyon sayısı: 100; İterasyon sayısı: 100; Çaprazlama Olasılığı: 0,8; Mutasyon Oranı: 0,02; Mutasyon Adım Boyutu (sigma): 0,1
PSO	Sürünün Büyüklüğü: 100; Eylemsizlik katsayısı ( $w$ ): 0,8; Bilişsel katsayısı ( $c_1$ ): 1.5; Sosyal katsayısı ( $c_2$ ): 1.5; İterasyon sayısı ( $N$ ): 100
DE	Popülasyon sayısı: 100; Çaprazlama olasılığı: 0,8; Ölçeklendirme faktörü ( $F$ ): 0,85; İterasyon Sayısı: 100
ABC	Sürünün büyüklüğü: 100; Besin kaynağı sayısı, görevli arılar, gözcü arılar: 50; İterasyon sayısı: 100; limit: 5

Yapılan simülasyon çalışmasında birden fazla parametre için meta-sezgisel algoritmaların sıklıkla ilk parametreyi doğru bir şekilde tahmin etme eğilimi gösterirken diğer parametreleri için gerçek değerlerinden farklı tahmin ettiği gözlenmiştir. Bu sorunu çözmek ve tahmin doğruluğunu artırmak için her algoritmaya bir başlangıç tahmin adımı uygulanmıştır. İki parametrelili problemler için, her seferde bir parametrenin yinelemeli olarak sabitlenmesiyle tek parametrelili bir tahmin yaklaşımı benimsenmiştir. Üç parametrelili problemler, tek parametrelili ve iki parametrelili alt tahmin problemlerine ayrıştırılır. Örneğin, iki parametrelili bir problem üç çözümün hesaplanmasını gerektirir. Benzer şekilde, üç parametrelili bir problem için 7 farklı şekilde çözümlerin hesaplanmasını içerir. Problemin iki veya üç parametrelili olmasına bağlı olarak üç veya

yedi olası çözüm karşılaştırılır. Daha sonra bulunan çözümlerden log-olabilirlik değeri en yüksek olan çözüm seçilir.

Newton-Raphson optimizasyon algoritması, gerçek değerli bir fonksiyon için kritik noktaları (yerel minimum veya maksimum) bulmak için kullanılan yinelemeli bir tekniktir. Çok değişkenli optimizasyon problemleri için Newton-Raphson yönteminin genel adımları aşağıdaki gibidir:

1. Başlatma: Optimum çözüm için

$$x^{(0)} = [x_1^{(0)}, x_2^{(0)}, \dots, x_p^{(0)}]$$

olarak gösterilen bir başlangıç tahminiyle başlanır. Burada  $p$ , parametre sayısıdır.

2. Gradyan vektörü ve Hessian matrisleri hesaplanır: Geçerli  $x^{(k)}$ ,  $k = 0, 1, 2, \dots$  yinelemesinde  $f(x)$  amaç fonksiyonunun gradyan vektörünü,  $\nabla f(x^{(k)})$  ve Hessian matrisi  $\nabla^2 f(x^{(k)})$  hesaplanır. Burada  $k$  yineleme sayısını temsil etmektedir.

$$\nabla f(x^{(k)}) = \left[ \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial x_3}, \dots, \frac{\partial f}{\partial x_p} \right]^T$$

$$\nabla^2 f(x^{(k)}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_p} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_p} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_p \partial x_1} & \frac{\partial^2 f}{\partial x_p \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_p^2} \end{bmatrix}$$

3. Güncelleme: Güncelleme yönünü  $d^{(k)}$  hesaplanır:

$$d^{(k)} = -[\nabla^2 f(x^{(k)})]^{-1} \nabla f(x^{(k)})$$

4. Yineleme: Geçerli yineleme  $x^{(k)}$  kullanılarak  $x^{(k+1)}$  güncellenir:

$$x^{(k+1)} = x^{(k)} + d^{(k)}$$

5. Gradyan vektörünün büyüklüğü veya fonksiyon değerindeki değişikliğin belirlenen tolerans değerinden küçük olması gibi yakınsama kriterleri karşılanana kadar 2-4 arasındaki adımlar tekrarlanır.

Simülasyon sonuçları, her bir dönüştürülmüş dağılıma ayrılmış ayrı bölümlerde sırasıyla Bölüm 4.1: Dönüştürülmüş Üstel dağılım, Bölüm 4.2: Dönüştürülmüş Weibull dağılımı ve Bölüm 4.3: Dönüştürülmüş Frechet dağılımı şeklinde incelenmiştir.

### 5.1 Dönüştürülmüş Üstel Dağılım

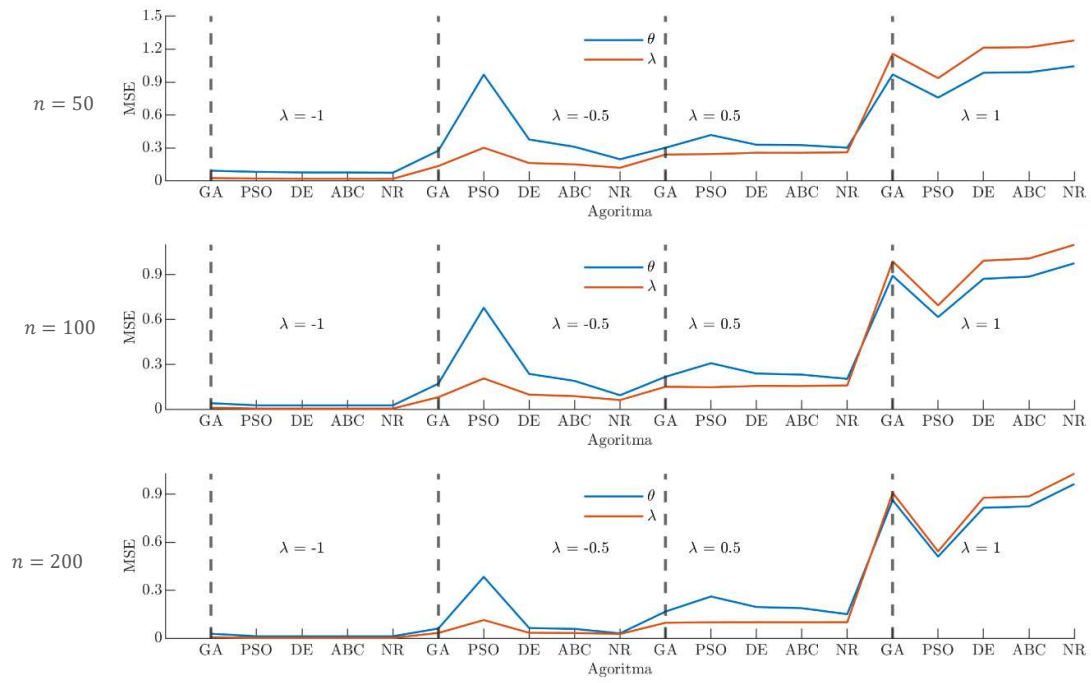
Bu bölümde, dönüştürülmüş üstel dağılım için çeşitli örneklem boyutları ( $n = 50, 100$  ve  $200$ ) ve lambda ( $\lambda$ ) değerleri ( $-1, -0,5, 0,5$  ve  $1$ ) için elde edilen simülasyon sonuçlarının ayrıntılı bir analizi sunulmaktadır. Analiz, dönüştürülmüş üstel dağılımın teta ( $\theta$ ) ve lambda ( $\lambda$ ) parametrelerinin tahmininde kullanılan parametre tahmin algoritmalarının (ABC, GA, PSO, DE ve NR) performansına odaklanmaktadır. Performans değerlendirmesi log-olabilirlik, yan, MAPE ve MSE ölçütlerine dayanmaktadır.

Tüm örneklem hacimleri ve  $\lambda$  değerleri için meta-sezgisel algoritmaların MSE, MAPE ve log-olabilirlik ölçütlerine göre daha iyi performans sergilediği görülmektedir. Çizelge 5.2, 5.3 ve 5.4 incelediğinde artan MSE ve yan değerleri tüm örnek boyutlarında  $\lambda$  değeri  $-1$ 'den  $1$ 'e çıktıkça algoritmaların performansının genel olarak kötüleştiğini göstermektedir.  $\lambda$  negatif olduğunda, algoritmalar pozitif  $\lambda$  değerlerine sahip senaryolarla karşılaştırıldığında parametrelerin tahmin edilmesinde üstün performans gösterir. Bu sonuç, algoritmaların negatif  $\lambda$  değerlerine karşı doğal bir önyargıya sahip olduğunu göstermektedir.

Çizelge 5.2-5.4 daha yakından incelendiğinde, farklı  $\lambda$  değerleri için algoritmaya özgü güçlü ve zayıf yönler ortaya çıkmaktadır. Negatif lambda ( $\lambda = -1$  ve  $-0,5$ ) değerleri ele alındığında, PSO hariç tüm algoritmalar  $n = 50$ ,  $n = 100$  ve  $n = 200$  için benzer performans göstermektedir. Bununla birlikte, daha büyük örnek boyutları için (Çizelge 5.3 ve 5.4) NR, her iki negatif  $\lambda$  değerinde de  $\lambda$  tahmini için en düşük MSE ile hafif bir avantaj sergilemektedir. İlginçtir ki,  $\lambda$  negatif olduğunda GA ve PSO diğer algoritmalarla karşılaştırıldığında sürekli olarak daha yüksek MSE ve yana sahiptir.

Pozitif Lambda ( $\lambda = 0,5$  ve  $1$ ) değerleri incelendiğinde GA ve PSO dışındaki tüm algoritmaların performansında önemli bir düşüş gözlenmektedir. GA ve PSO, pozitif  $\lambda$  değerlerine göre göreceli bir sağlamlık sergileyerek tüm örnek boyutlarında hem  $\theta$  hem de  $\lambda$  için tutarlı bir şekilde en düşük MSE'ye sahiptir (Çizelge 5.2-5.4). Bu durum, dönüştürülmüş üstel dağılımda pozitif  $\lambda$  değerlerine sahip senaryolar için GA ve PSO'nun daha uygun bir seçim olabileceğini göstermektedir.

Örneklem büyüklüğü ile parametre tahmin doğruluğu arasında pozitif bir ilişki göze çarpmaktadır. Örneklem boyutu 50'den 200'e çıktıkça (Çizelge 5.2-5.4) algoritmalar giderek daha düşük MSE ve sapma değerleri üretme eğilimindedir. Bu da gelişmiş parametre tahminlerine işaret eder. Bu sonuç, daha büyük örneklem boyutlarının daha güvenilir parametre tahminlerine yol açtığı yerleşik istatistiksel ilkelerle uyumludur.



**Şekil 5.1.** Dönüştürülmüş üstel dağılım farklı  $\lambda$  değerleri ve  $n$  için MSE değerleri.

Çizelge 5.2. Dönüştürülmüş üstel dağılım parametre tahmin değerleri ( $n = 50$ ,  $\theta = 2$ )

$\lambda$	Algoritma	Ortalama		Varyans		Yan		MSE		MAPE	$l$
		$\hat{\theta}$	$\hat{\lambda}$	$\hat{\theta}$	$\hat{\lambda}$	$\hat{\theta}$	$\hat{\lambda}$	$\hat{\theta}$	$\hat{\lambda}$		
-1	GA	2,1274	-0,9230	0,0763	0,0192	0,1274	0,0770	0,0925	0,0252	0,0924	-99,6085
	PSO	2,0556	-0,9481	0,0787	0,0179	0,0556	0,0519	0,0818	0,0206	0,0741	-99,4684
	DE	2,0543	-0,9487	0,0742	0,0165	0,0543	0,0513	0,0772	0,0191	0,0735	-99,4683
	ABC	2,0544	-0,9486	0,0742	0,0165	0,0544	0,0514	0,0771	0,0191	0,0735	-99,4683
	NR	2,0538	-0,9485	0,0712	0,0159	0,0538	0,0515	0,0741	0,0186	0,0734	-99,4725
-0,5	GA	2,1045	-0,4661	0,2636	0,1338	0,1045	0,0339	0,2746	0,1349	0,3460	-94,0663
	PSO	2,2810	-0,3981	0,8887	0,2914	0,2810	0,1019	0,9677	0,3018	0,4951	-94,0416
	DE	2,1053	-0,4751	0,3656	0,1617	0,1053	0,0249	0,3767	0,1623	0,3761	-94,0303
	ABC	2,0892	-0,4811	0,3021	0,1498	0,0892	0,0189	0,3100	0,1502	0,3657	-94,0303
	NR	2,0550	-0,4938	0,1938	0,1192	0,0550	0,0062	0,1969	0,1192	0,3416	-94,0475
0,5	GA	1,7795	0,2327	0,2531	0,1686	-0,2205	-0,2673	0,3017	0,2401	0,4958	-69,2204
	PSO	1,9820	0,3694	0,4173	0,2267	-0,0180	-0,1306	0,4177	0,2437	0,5126	-69,2235
	DE	1,8025	0,2429	0,2903	0,1898	-0,1975	-0,2571	0,3293	0,2559	0,5092	-69,2133
	ABC	1,7911	0,2348	0,2819	0,1851	-0,2089	-0,2652	0,3255	0,2554	0,5083	-69,2135
	NR	1,7328	0,1926	0,2306	0,1655	-0,2672	-0,3074	0,3020	0,2600	0,5091	-69,2170
1	GA	1,0656	0,0191	0,0968	0,1951	-0,9344	-0,9809	0,9699	1,1573	0,7252	-48,9278
	PSO	1,2884	0,2458	0,2525	0,3670	-0,7116	-0,7542	0,7588	0,9359	0,5677	-48,9372
	DE	1,0700	0,0083	0,1199	0,2305	-0,9300	-0,9917	0,9849	1,2140	0,7305	-48,9020
	ABC	1,0622	0,0021	0,1099	0,2221	-0,9378	-0,9979	0,9894	1,2178	0,7349	-48,9028
	NR	1,0124	-0,0524	0,0696	0,1716	-0,9876	-1,0524	1,0450	1,2793	0,7733	-48,9049

Çizelge 5.3 Dönüştürülmüş üstel dağılım parametre tahmin değerleri ( $n = 100$ ,  $\theta = 2$ )

$\lambda$	Algoritma	Ortalama		Varyans		Yan		MSE		MAPE	$l$
		$\hat{\theta}$	$\hat{\lambda}$	$\hat{\theta}$	$\hat{\lambda}$	$\hat{\theta}$	$\hat{\lambda}$	$\hat{\theta}$	$\hat{\lambda}$		
-1	GA	2,0991	-0,9397	0,0312	0,0066	0,0991	0,0603	0,0410	0,0103	0,0695	-199,2219
	PSO	2,0199	-0,9672	0,0262	0,0042	0,0199	0,0328	0,0266	0,0053	0,0483	-198,9346
	DE	2,0199	-0,9672	0,0262	0,0042	0,0199	0,0328	0,0266	0,0053	0,0483	-198,9346
	ABC	2,0199	-0,9672	0,0262	0,0042	0,0199	0,0328	0,0266	0,0053	0,0483	-198,9346
	NR	2,0198	-0,9672	0,0262	0,0042	0,0198	0,0328	0,0266	0,0053	0,0483	-198,9371
-0,5	GA	2,0802	-0,4606	0,1647	0,0794	0,0802	0,0394	0,1712	0,0810	0,2505	-189,1716
	PSO	2,2143	-0,4099	0,6327	0,1984	0,2143	0,0901	0,6786	0,2065	0,3619	-189,1547
	DE	2,0789	-0,4698	0,2304	0,0971	0,0789	0,0302	0,2366	0,0980	0,2688	-189,1398
	ABC	2,0662	-0,4748	0,1854	0,0874	0,0662	0,0252	0,1898	0,0880	0,2604	-189,1399
	NR	2,0366	-0,4865	0,0928	0,0618	0,0366	0,0135	0,0942	0,0620	0,2401	-189,1573
0,5	GA	1,8855	0,3507	0,2043	0,1282	-0,1145	-0,1493	0,2174	0,1505	0,3998	-138,8548
	PSO	2,0949	0,4953	0,2991	0,1474	0,0949	-0,0047	0,3081	0,1474	0,4130	-138,8663
	DE	1,9243	0,3753	0,2334	0,1406	-0,0757	-0,1247	0,2391	0,1561	0,4080	-138,8507
	ABC	1,9067	0,3632	0,2231	0,1370	-0,0933	-0,1368	0,2318	0,1557	0,4067	-138,8508
	NR	1,8149	0,2972	0,1692	0,1187	-0,1851	-0,2028	0,2035	0,1599	0,4048	-138,8608
1	GA	1,1069	0,0888	0,0948	0,1579	-0,8931	-0,9112	0,8924	0,9882	0,6798	-99,3146
	PSO	1,3944	0,3790	0,2498	0,3093	-0,6056	-0,6210	0,6165	0,6949	0,4749	-99,3327
	DE	1,1381	0,1110	0,1292	0,2030	-0,8619	-0,8890	0,8722	0,9933	0,6625	-99,2863
	ABC	1,1218	0,0956	0,1149	0,1894	-0,8782	-0,9044	0,8861	1,0073	0,6732	-99,2875
	NR	1,0411	0,0091	0,0564	0,1180	-0,9589	-0,9909	0,9759	1,0999	0,7353	-99,2951

Çizelge 5.4. Dönüştürülmüş üstel dağılım parametre tahmin değerleri ( $n = 200$ ,  $\theta = 2$ )

$\lambda$	Algoritma	Ortalama		Varyans		Yan		MSE		MAPE	$l$
		$\hat{\theta}$	$\hat{\lambda}$	$\hat{\theta}$	$\hat{\lambda}$	$\hat{\theta}$	$\hat{\lambda}$	$\hat{\theta}$	$\hat{\lambda}$		
-1	GA	2,1010	-0,9466	0,0182	0,0043	0,1010	0,0534	0,0284	0,0071	0,0594	-400,1823
	PSO	2,0203	-0,9760	0,0122	0,0018	0,0203	0,0240	0,0126	0,0024	0,0341	-399,6162
	DE	2,0203	-0,9760	0,0122	0,0018	0,0203	0,0240	0,0126	0,0024	0,0341	-399,6162
	ABC	2,0203	-0,9760	0,0122	0,0018	0,0203	0,0240	0,0126	0,0024	0,0341	-399,6162
	NR	2,0206	-0,9760	0,0123	0,0018	0,0206	0,0240	0,0127	0,0024	0,0342	-399,6181
-0,5	GA	2,0348	-0,4774	0,0595	0,0325	0,0348	0,0226	0,0607	0,0330	0,1649	-378,8764
	PSO	2,1205	-0,4444	0,3689	0,1106	0,1205	0,0556	0,3834	0,1137	0,2395	-378,8813
	DE	2,0188	-0,4919	0,0636	0,0341	0,0188	0,0081	0,0639	0,0342	0,1676	-378,8320
	ABC	2,0177	-0,4922	0,0583	0,0332	0,0177	0,0078	0,0586	0,0332	0,1668	-378,8322
	NR	2,0109	-0,4932	0,0306	0,0265	0,0109	0,0068	0,0307	0,0266	0,1626	-378,8635
0,5	GA	1,9571	0,4211	0,1650	0,0906	-0,0429	-0,0789	0,1668	0,0968	0,3347	-278,8727
	PSO	2,1593	0,5570	0,2354	0,0963	0,1593	0,0570	0,2608	0,0995	0,3561	-278,8812
	DE	2,0145	0,4589	0,1950	0,0983	0,0145	-0,0411	0,1952	0,1000	0,3439	-278,8675
	ABC	1,9994	0,4488	0,1879	0,0972	-0,0006	-0,0512	0,1879	0,0999	0,3425	-278,8678
	NR	1,8889	0,3718	0,1382	0,0837	-0,1111	-0,1282	0,1505	0,1001	0,3339	-278,8885
1	GA	1,1246	0,1217	0,0951	0,1369	-0,8754	-0,8783	0,8615	0,9083	0,6588	-199,4578
	PSO	1,4800	0,4809	0,2395	0,2731	-0,5200	-0,5191	0,5100	0,5425	0,3998	-199,4952
	DE	1,1816	0,1714	0,1449	0,1904	-0,8184	-0,8286	0,8146	0,8770	0,6216	-199,4308
	ABC	1,1722	0,1622	0,1383	0,1838	-0,8278	-0,8378	0,8235	0,8856	0,6282	-199,4324
	NR	1,0381	0,0241	0,0379	0,0754	-0,9619	-0,9759	0,9632	1,0277	0,7284	-199,4547

## 5.2 Dönüştürülmüş Weibull Dağılımı

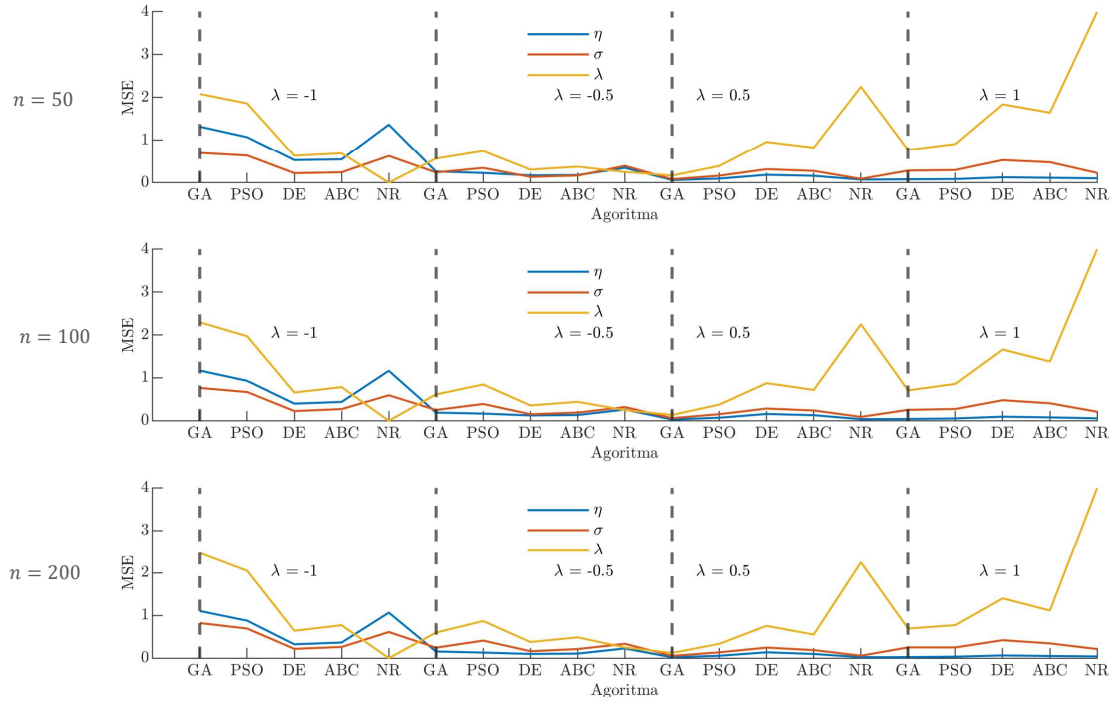
Bu bölümde dönüştürülmüş Weibull dağılımı için elde edilen simülasyon sonuçları sunulmaktadır. Farklı örneklem boyutlarında ( $n = 50, 100$  ve  $200$ ) ve  $\lambda$  değerleri ( $-1, -0,5, 0,5$  ve  $1$ ) için parametrelerin ( $\eta, \sigma$  ve  $\lambda$ ) tahmin edilmesinde kullanılan çeşitli algoritmaların (ABC, GA, PSO, DE ve NR) performansı MSE, MAPE ve log-olabilirlik değerleri açısından incelenerek Çizelge 5.5, 5.6 ve 5.7'de verilmiştir.

Tüm örneklem hacimleri ve  $\lambda$  değerleri için meta-sezgisel algoritmaların MSE, MAPE ve log-olabilirlik ölçütlerine göre daha iyi performans sergilediği görülmektedir.

$\lambda$ 'nın etkisinin incelenmesi, algoritmaların güçlü ve zayıf yönlerini ortaya çıkarır. Tüm örnek büyüklükleri için geçerli olmak üzere, negatif  $\lambda$  değerleri için ( $\lambda = -1, -0,5$ ), DE algoritması, MSE, MAPE ve log-olabilirlik değerlerine göre üç parametrenin tamamını ( $\eta, \sigma, \lambda$ ) tahmin etme konusunda diğerlerinden daha iyi performans gösterir (bkz. Çizelge 5.5-5.7). Bu durum DE'nin dönüştürülmüş Weibull dağılımıyla ilişkili negatif  $\lambda$  değerlerini ele almadaki sağlamlığını göstermektedir.

Tüm örnek büyüklükleri için geçerli olmak üzere, pozitif  $\lambda$  değerleri için ( $\lambda = 0,5, 1$ ) genel olarak GA tüm örnek boyutlarında en iyi performans gösteren tahmin edici olarak ortaya çıkmaktadır (bkz. Çizelge 5.5-5.7). GA'nın üstün performansı, arama alanı içindeki keşif kapasitesine atfedilebilir, bu da  $\lambda$ 'nın kendisi de dahil olmak üzere tüm parametreler için daha doğru tahminlere yol açar.

En uygun algoritmanın seçimi hem spesifik  $\lambda$  değerine hem de istenilen parametreye bağlıdır. Negatif  $\lambda$  değerlerinin tahmini için DE, pozitif  $\lambda$  değerleri için tüm parametrelerde GA üstün performans göstermektedir. Bununla birlikte, genel performans ve sağlamlığın çok önemli olduğu ve gerçek  $\lambda$  değerinin bilinmediği durumlarda DE veya GA gibi daha çok yönlü bir algoritma, farklı  $\lambda$  koşullarındaki göreceli tutarlılıkları nedeniyle daha güvenli bir seçim olabilir.



**Şekil 5.2.** Dönüştürülmüş Weibull dağılımı farklı  $\lambda$  değerleri ve  $n$  için MSE değerleri.

Çizelge 5.5. Dönüştürülmüş Weibull dağılımı parametre tahmin değerleri ( $n = 50$ ,  $\eta = 2$ ,  $\sigma = 2$ )

$\lambda$	Algoritma	Ortalama			Varyans			Yan			MSE			MAPE	$l$
		$\hat{\eta}$	$\hat{\sigma}$	$\hat{\lambda}$	$\hat{\eta}$	$\hat{\sigma}$	$\hat{\lambda}$	$\hat{\eta}$	$\hat{\sigma}$	$\hat{\lambda}$	$\hat{\eta}$	$\hat{\sigma}$	$\hat{\lambda}$		
-1	GA	3,0870	2,7963	0,4023	0,1342	0,0618	0,1128	1,0870	0,7963	1,4023	1,3157	0,6960	2,0791	0,7815	-61,2863
	PSO	2,9104	2,6977	0,2123	0,2419	0,1490	0,3902	0,9104	0,6977	1,2123	1,0707	0,6358	1,8597	0,6766	-61,2282
	DE	2,4704	2,2846	-0,5106	0,3071	0,1410	0,3909	0,4704	0,2846	0,4894	0,5283	0,2220	0,6304	0,3072	-61,0124
	ABC	2,4817	2,3002	-0,4841	0,3125	0,1512	0,4216	0,4817	0,3002	0,5159	0,5446	0,2414	0,6878	0,3202	-61,0145
	NR	3,1036	2,7626	-0,9995	0,1464	0,0422	0,0000	1,1036	0,7626	0,0005	1,3643	0,6237	0,0000	0,3118	-86,0067
-0,5	GA	2,3965	2,3892	0,1394	0,1022	0,0861	0,1560	0,3965	0,3892	0,6394	0,2593	0,2376	0,5648	0,5676	-65,0415
	PSO	2,3145	2,4091	0,1353	0,1276	0,1764	0,3362	0,3145	0,4091	0,6353	0,2265	0,3438	0,7398	0,6165	-65,0182
	DE	2,1411	2,1234	-0,2967	0,1487	0,1180	0,2629	0,1411	0,1234	0,2033	0,1686	0,1332	0,3042	0,3968	-64,9238
	ABC	2,1686	2,1686	-0,2288	0,1472	0,1366	0,2989	0,1686	0,1686	0,2712	0,1756	0,1650	0,3725	0,4304	-64,9302
	NR	2,4769	2,5259	-0,9990	0,1166	0,1151	0,0001	0,4769	0,5259	-0,4990	0,3441	0,3917	0,2490	0,5037	-91,2790
0,5	GA	2,0090	1,8858	0,3013	0,0591	0,0657	0,1253	0,0090	-0,1142	-0,1987	0,0592	0,0787	0,1648	0,2877	-57,0981
	PSO	1,9277	1,8863	0,2618	0,0876	0,1483	0,3328	-0,0723	-0,1137	-0,2382	0,0928	0,1612	0,3896	0,4135	-57,0688
	DE	1,7468	1,5754	-0,2674	0,1184	0,1332	0,3708	-0,2532	-0,4246	-0,7674	0,1825	0,3135	0,9597	0,6917	-56,9445
	ABC	1,7907	1,6467	-0,1455	0,1149	0,1491	0,4076	-0,2093	-0,3533	-0,6455	0,1587	0,2739	0,8242	0,6238	-56,9576
	NR	2,0321	1,9121	-0,9990	0,0690	0,0795	0,0001	0,0321	-0,0879	-1,4990	0,0700	0,0872	2,2470	1,0614	-82,4734
1	GA	2,1115	1,5059	0,1939	0,0640	0,0371	0,1229	0,1115	-0,4941	-0,8061	0,0764	0,2813	0,7728	0,3873	-45,8592
	PSO	2,0363	1,5484	0,2341	0,0817	0,0917	0,3239	0,0363	-0,4516	-0,7659	0,0830	0,2956	0,9105	0,3717	-45,8314
	DE	1,8828	1,3196	-0,2451	0,1103	0,0651	0,2876	-0,1172	-0,6804	-1,2451	0,1240	0,5280	1,8378	0,5761	-45,7365
	ABC	1,9141	1,3698	-0,1420	0,1053	0,0795	0,3410	-0,0859	-0,6302	-1,1420	0,1127	0,4766	1,6450	0,5314	-45,7485
	NR	2,1564	1,5749	-0,9986	0,0748	0,0459	0,0002	0,1564	-0,4251	-1,9986	0,0992	0,2266	3,9947	0,7815	-71,6154

Çizelge 5.6 Dönüştürülmüş Weibull dağılımı parametre tahmin değerleri ( $n = 100$ ,  $\eta = 2$ ,  $\sigma = 2$ )

$\lambda$	Algoritma	Ortalama			Varyans			Yan			MSE			MAPE	$l$
		$\hat{\eta}$	$\hat{\sigma}$	$\hat{\lambda}$	$\hat{\eta}$	$\hat{\sigma}$	$\hat{\lambda}$	$\hat{\eta}$	$\hat{\sigma}$	$\hat{\lambda}$	$\hat{\eta}$	$\hat{\sigma}$	$\hat{\lambda}$		
-1	GA	3,0521	2,8506	0,4858	0,0627	0,0412	0,0854	1,0521	0,8506	1,4858	1,1695	0,7648	2,2930	0,8125	-123,9321
	PSO	2,8547	2,7186	0,2423	0,2002	0,1539	0,4247	0,8547	0,7186	1,2423	0,9307	0,6703	1,9681	0,6801	-123,8162
	DE	2,3898	2,2756	-0,5271	0,2480	0,1508	0,4350	0,3898	0,2756	0,4729	0,3999	0,2268	0,6586	0,2834	-123,4726
	ABC	2,4228	2,3140	-0,4620	0,2606	0,1735	0,4958	0,4228	0,3140	0,5380	0,4394	0,2721	0,7853	0,3165	-123,4793
	NR	3,0469	2,7596	-0,9992	0,0691	0,0182	0,0001	1,0469	0,7596	0,0008	1,1651	0,5953	0,0001	0,3014	-172,4875
-0,5	GA	2,3711	2,4167	0,1775	0,0530	0,0787	0,1556	0,3711	0,4167	0,6775	0,1907	0,2523	0,6146	0,5875	-131,0836
	PSO	2,2966	2,4647	0,2112	0,0793	0,1758	0,3398	0,2966	0,4647	0,7112	0,1672	0,3918	0,8456	0,6554	-131,0535
	DE	2,1434	2,1717	-0,2226	0,1039	0,1228	0,2809	0,1434	0,1717	0,2774	0,1244	0,1523	0,3578	0,4078	-130,9328
	ABC	2,1779	2,2264	-0,1384	0,1025	0,1396	0,3107	0,1779	0,2264	0,3616	0,1341	0,1908	0,4415	0,4508	-130,9458
	NR	2,4510	2,5144	-0,9986	0,0588	0,0553	0,0002	0,4510	0,5144	-0,4986	0,2621	0,3199	0,2488	0,4952	-182,5387
0,5	GA	1,9805	1,9186	0,3483	0,0291	0,0542	0,1130	-0,0195	-0,0814	-0,1517	0,0294	0,0609	0,1360	0,2535	-115,6765
	PSO	1,8949	1,9228	0,3050	0,0599	0,1478	0,3390	-0,1051	-0,0772	-0,1950	0,0710	0,1538	0,3770	0,3945	-115,6335
	DE	1,7384	1,6230	-0,1885	0,0904	0,1431	0,4029	-0,2616	-0,3770	-0,6885	0,1588	0,2853	0,8770	0,6423	-115,4717
	ABC	1,7898	1,7122	-0,0378	0,0863	0,1576	0,4296	-0,2102	-0,2878	-0,5378	0,1305	0,2404	0,7188	0,5615	-115,4980
	NR	1,9905	1,9264	-0,9991	0,0374	0,0853	0,0001	-0,0095	-0,0736	-1,4991	0,0375	0,0908	2,2474	1,0488	-166,5562
1	GA	2,0828	1,5285	0,2343	0,0346	0,0327	0,1218	0,0828	-0,4715	-0,7657	0,0415	0,2550	0,7081	0,3607	-93,0850
	PSO	2,0076	1,5675	0,2649	0,0542	0,0874	0,3190	0,0076	-0,4325	-0,7351	0,0543	0,2745	0,8593	0,3491	-93,0516
	DE	1,8817	1,3572	-0,1646	0,0814	0,0663	0,3028	-0,1183	-0,6428	-1,1646	0,0954	0,4795	1,6592	0,5360	-92,9364
	ABC	1,9294	1,4239	-0,0243	0,0741	0,0763	0,3315	-0,0706	-0,5761	-1,0243	0,0791	0,4082	1,3807	0,4745	-92,9604
	NR	2,1281	1,5729	-0,9990	0,0404	0,0284	0,0002	0,1281	-0,4271	-1,9990	0,0568	0,2107	3,9961	0,7706	-144,1732

Çizelge 5.7. Dönüştürülmüş Weibull dağılımı parametre tahmin değerleri ( $n = 200$ ,  $\eta = 2$ ,  $\sigma = 2$ )

$\lambda$	Algoritma	Ortalama			Varyans			Yan			MSE			MAPE	$l$
		$\hat{\eta}$	$\hat{\sigma}$	$\hat{\lambda}$	$\hat{\eta}$	$\hat{\sigma}$	$\hat{\lambda}$	$\hat{\eta}$	$\hat{\sigma}$	$\hat{\lambda}$	$\hat{\eta}$	$\hat{\sigma}$	$\hat{\lambda}$		
-1	GA	3,0328	2,8890	0,5483	0,0364	0,0301	0,0677	1,0328	0,8890	1,5483	1,1032	0,8205	2,4650	0,8364	-249,2838
	PSO	2,8334	2,7343	0,2707	0,1825	0,1530	0,4340	0,8334	0,7343	1,2707	0,8771	0,6922	2,0487	0,6880	-249,0948
	DE	2,3177	2,2482	-0,5688	0,2222	0,1548	0,4556	0,3177	0,2482	0,4312	0,3232	0,2164	0,6415	0,2503	-248,4943
	ABC	2,3564	2,2898	-0,4984	0,2375	0,1767	0,5198	0,3564	0,2898	0,5016	0,3646	0,2606	0,7714	0,2866	-248,5131
	NR	3,0110	2,7665	-0,9992	0,0432	0,0234	0,0001	1,0110	0,7665	0,0008	1,0653	0,6109	0,0001	0,2968	-346,8718
-0,5	GA	2,3506	2,4132	0,1674	0,0320	0,0731	0,1527	0,3506	0,4132	0,6674	0,1550	0,2438	0,5980	0,5752	-263,1975
	PSO	2,2656	2,4722	0,2079	0,0582	0,1872	0,3681	0,2656	0,4722	0,7079	0,1287	0,4101	0,8692	0,6500	-263,1292
	DE	2,1457	2,1976	-0,1860	0,0759	0,1215	0,2780	0,1457	0,1976	0,3140	0,0972	0,1606	0,3766	0,3974	-262,9666
	ABC	2,1877	2,2696	-0,0782	0,0696	0,1384	0,3084	0,1877	0,2696	0,4218	0,1048	0,2111	0,4863	0,4501	-262,9947
	NR	2,4333	2,5214	-0,9992	0,0379	0,0627	0,0001	0,4333	0,5214	-0,4992	0,2256	0,3346	0,2493	0,4941	-366,4583
0,5	GA	1,9688	1,9403	0,3817	0,0170	0,0477	0,1074	-0,0312	-0,0597	-0,1183	0,0180	0,0513	0,1214	0,2312	-232,5199
	PSO	1,8921	1,9525	0,3523	0,0424	0,1317	0,3128	-0,1079	-0,0475	-0,1477	0,0540	0,1340	0,3346	0,3618	-232,4795
	DE	1,7622	1,6831	-0,0832	0,0787	0,1448	0,4160	-0,2378	-0,3169	-0,5832	0,1353	0,2452	0,7561	0,5728	-232,2883
	ABC	1,8277	1,7989	0,1104	0,0667	0,1468	0,3993	-0,1723	-0,2011	-0,3896	0,0964	0,1872	0,5511	0,4664	-232,3463
	NR	1,9836	1,9152	-0,9986	0,0210	0,0483	0,0002	-0,0164	-0,0848	-1,4986	0,0213	0,0555	2,2460	1,0382	-333,2511
1	GA	2,0661	1,5320	0,2410	0,0186	0,0315	0,1166	0,0661	-0,4680	-0,7590	0,0229	0,2505	0,6927	0,3516	-187,1631
	PSO	1,9988	1,5954	0,3162	0,0331	0,0860	0,3060	-0,0012	-0,4046	-0,6838	0,0331	0,2498	0,7736	0,3201	-187,1206
	DE	1,9156	1,3998	-0,0650	0,0550	0,0594	0,2666	-0,0844	-0,6002	-1,0650	0,0622	0,4196	1,4008	0,4869	-186,9889
	ABC	1,9580	1,4817	0,0989	0,0470	0,0755	0,3044	-0,0420	-0,5183	-0,9011	0,0487	0,3441	1,1163	0,4154	-187,0238
	NR	2,1050	1,5811	-0,9983	0,0278	0,0359	0,0002	0,1050	-0,4189	-1,9983	0,0388	0,2114	3,9934	0,7648	-289,8929

### 5.3 Dönüştürülmüş Fréchet Dağılımı

Bu bölümde, farklı  $\lambda$  (dönüşüm parametresi) değerleri ve örnek boyutları ( $n = 50, 100, 200$ ) genelinde çeşitli parametre tahmin algoritmalarının (meta-sezgisel ve NR) performansına odaklanarak dönüştürülmüş Fréchet dağılımı için simülasyon sonuçları incelemiştir. Ortalama, varyans, yan, MSE, MAPE ve log-olabilirlik dahil olmak üzere ayrıntılı sonuçlar Çizelge 5.8-5.10'da sunulmaktadır.

Log-olabilirlik değerine göre  $\lambda = -1$  dışında tüm örneklem hacimlerinde meta-sezgisel algoritmalar NR'den daha iyi performans göstermiştir.  $\lambda = -1$  alınması durumunda ise PSO, DE ve ABC,  $n = 50$  ve  $n = 100$  örneklem hacimlerinde NR'dan daha iyi performans gösterirken,  $n = 200$  için DE ve ABC, NR'dan daha iyi performans sergilemiştir.

Pozitif  $\lambda$  değerleri için MAPE ölçütüne tüm örneklem hacimlerinde meta-sezgisel algoritmalar NR'dan daha iyi performans göstermiştir.  $\lambda = -1$  için MAPE ölçütüne göre  $n = 50$  için GA,  $n = 100$  ve  $n = 200$  için GA ve PSO, NR'dan daha iyi performans göstermiştir.

$\lambda$  parametre tahminleri, pozitif  $\lambda$  değerleri için MSE ölçütüne göre incelediğinde tüm örneklem hacimlerinde meta-sezgisel algoritmaların NR'dan daha iyi performans sergilediğini göstermektedir.  $\lambda = -1$  için MSE ölçütüne göre  $n = 50$  olması durumunda GA,  $n = 100$  ve  $n = 200$  için ise GA ve PSO NR'dan daha iyi performans göstermiştir.

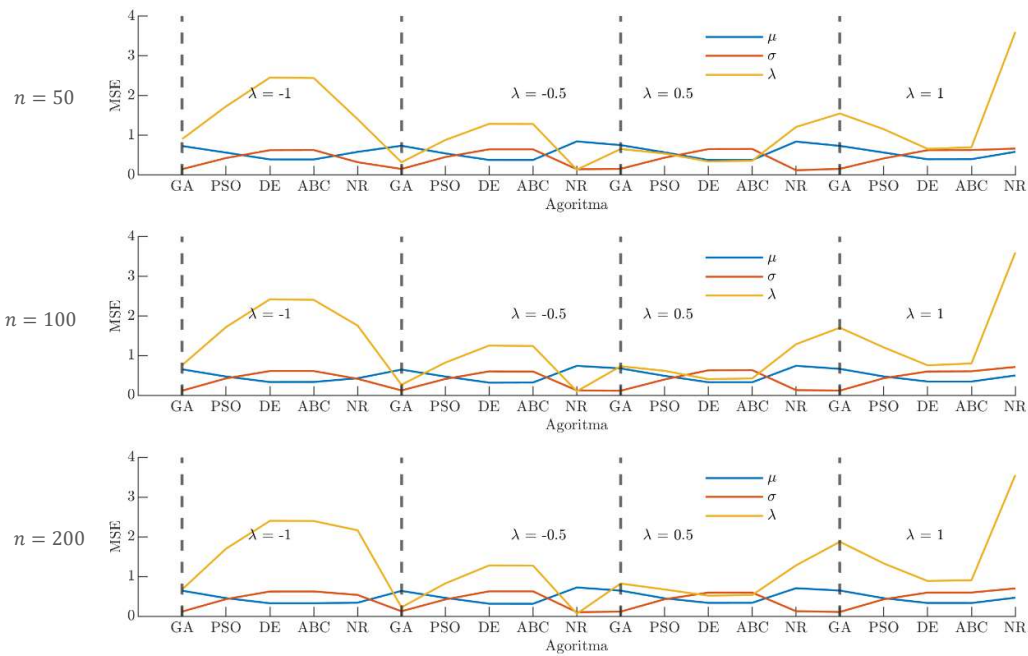
Meta-sezgisel algoritmaların pozitif  $\lambda$  değerlerini olduğundan daha düşük, negatif  $\lambda$  değerlerini ise olduğundan daha yüksek tahmin ettiği görülmektedir. NR ise  $\lambda = -1$  dışında,  $\lambda$  değerlerini daha düşük tahmin etme eğilimiyle göze çarpmaktadır.

Örnek boyutunun algoritmaların performansı üzerindeki etkisi, dönüşüm parametresinin ( $\lambda$ ) gerçek değerine bağlıdır.  $\lambda$  negatif olduğunda, tüm algoritmalar için örneklem boyutu arttıkça tahmin hatasının azalma eğilimi vardır. Bu gözlem, daha büyük

veri kümelerinin genellikle daha doğru tahminlere yol açtığı istatistiksel ilkelerle uyumludur. Tersine, pozitif  $\lambda$  değerleri için mantığa aykırı bir eğilim ortaya çıkar.

Örneklem büyüklüğü arttıkça tahmin hatası da artar.

En uygun algoritmanın seçimi hem gerçek  $\lambda$  değerine hem de istenen parametre tahminlerine bağlıdır.  $\lambda$  negatif olduğunda GA veya NR (spesifik değere bağlı olarak) tercih edilebilir. Pozitif lambda için DE en sağlam seçim gibi görünmektedir.



**Şekil 5.3.** Dönüştürülmüş Fréchet dağılımı farklı  $\lambda$  değerleri ve  $n$  için MSE değerleri.

Çizelge 5.8. Dönüştürülmüş Fréchet dağılımı parametre tahmin değerleri ( $n = 50$ ,  $\mu = 2$ ,  $\sigma = 3$ )

$\lambda$	Algoritma	Ortalama			Varyans			Yan			MSE			MAPE	$l$
		$\hat{\mu}$	$\hat{\sigma}$	$\hat{\lambda}$	$\hat{\mu}$	$\hat{\sigma}$	$\hat{\lambda}$	$\hat{\mu}$	$\hat{\sigma}$	$\hat{\lambda}$	$\hat{\mu}$	$\hat{\sigma}$	$\hat{\lambda}$		
-1	GA	2,7660	3,0161	-0,1669	0,1431	0,1482	0,2128	0,7660	0,0161	0,8331	0,7298	0,1484	0,9067	0,4392	-94,3078
	PSO	2,5507	3,2889	0,1439	0,2611	0,3440	0,4206	0,5507	0,2889	1,1439	0,5643	0,4274	1,7292	0,5408	-94,2307
	DE	2,3629	3,5377	0,4446	0,2607	0,3361	0,3679	0,3629	0,5377	1,4446	0,3924	0,6253	2,4549	0,6330	-94,1377
	ABC	2,3628	3,5291	0,4321	0,2602	0,3491	0,3912	0,3628	0,5291	1,4321	0,3918	0,6291	2,4422	0,6293	-94,1380
	NR	2,6174	3,1512	-0,0058	0,2013	0,2991	0,4178	0,6174	0,1512	0,9942	0,5825	0,3219	1,4062	0,4883	-94,2725
-0,5	GA	2,7629	3,0178	-0,1757	0,1544	0,1513	0,2154	0,7629	0,0178	0,3243	0,7364	0,1516	0,3206	0,4314	-94,4954
	PSO	2,5259	3,3207	0,1718	0,2670	0,3497	0,4301	0,5259	0,3207	0,6718	0,5436	0,4526	0,8814	0,6566	-94,3983
	DE	2,3462	3,5635	0,4655	0,2591	0,3299	0,3559	0,3462	0,5635	0,9655	0,3790	0,6474	1,2881	0,8173	-94,3091
	ABC	2,3474	3,5579	0,4584	0,2577	0,3356	0,3667	0,3474	0,5579	0,9584	0,3784	0,6468	1,2852	0,8160	-94,3094
	NR	2,8590	2,7778	-0,5094	0,1084	0,0946	0,1332	0,8590	-0,2222	-0,0094	0,8463	0,1440	0,1333	0,3673	-94,5347
0,5	GA	2,7766	3,0202	-0,1688	0,1494	0,1553	0,2085	0,7766	0,0202	-0,6688	0,7525	0,1557	0,6557	0,6549	-94,1755
	PSO	2,5520	3,3016	0,1539	0,2665	0,3455	0,4216	0,5520	0,3016	-0,3461	0,5712	0,4365	0,5415	0,5983	-94,0897
	DE	2,3463	3,5744	0,4857	0,2570	0,3231	0,3400	0,3463	0,5744	-0,0143	0,3770	0,6530	0,3402	0,5052	-93,9904
	ABC	2,3470	3,5675	0,4756	0,2569	0,3335	0,3581	0,3470	0,5675	-0,0244	0,3773	0,6555	0,3587	0,5117	-93,9907
	NR	2,8620	2,7483	-0,5338	0,0988	0,0564	0,1414	0,8620	-0,2517	-1,0338	0,8418	0,1197	1,2102	0,8678	-94,2232
1	GA	2,7554	3,0353	-0,1546	0,1627	0,1545	0,2156	0,7554	0,0353	-1,1546	0,7333	0,1557	1,5486	0,5455	-94,6502
	PSO	2,5503	3,2909	0,1381	0,2627	0,3336	0,4105	0,5503	0,2909	-0,8619	0,5656	0,4182	1,1533	0,4459	-94,5798
	DE	2,3624	3,5474	0,4456	0,2651	0,3256	0,3511	0,3624	0,5474	-0,5544	0,3964	0,6252	0,6584	0,3368	-94,4835
	ABC	2,3639	3,5370	0,4314	0,2655	0,3398	0,3747	0,3639	0,5370	-0,5686	0,3980	0,6281	0,6980	0,3423	-94,4838
	NR	2,4119	2,2928	-0,8865	0,4145	0,1644	0,0435	0,4119	-0,7072	-1,8865	0,5841	0,6645	3,6022	0,8212	-98,7531

Çizelge 5.9. Dönüştürülmüş Fréchet dağılımı parametre tahmin değerleri ( $n = 100$ ,  $\mu = 2$ ,  $\sigma = 3$ )

$\lambda$	Algoritma	Ortalama			Varyans			Yan			MSE			MAPE	$l$
		$\hat{\mu}$	$\hat{\sigma}$	$\hat{\lambda}$	$\hat{\mu}$	$\hat{\sigma}$	$\hat{\lambda}$	$\hat{\mu}$	$\hat{\sigma}$	$\hat{\lambda}$	$\hat{\mu}$	$\hat{\sigma}$	$\hat{\lambda}$		
-1	GA	2,7524	2,9638	-0,2423	0,0901	0,1178	0,1897	0,7524	-0,0362	0,7577	0,6561	0,1191	0,7639	0,4083	-190,1784
	PSO	2,5024	3,2813	0,1199	0,2223	0,3493	0,4665	0,5024	0,2813	1,1199	0,4747	0,4284	1,7206	0,5260	-190,0396
	DE	2,3276	3,5286	0,4171	0,2287	0,3345	0,4136	0,3276	0,5286	1,4171	0,3360	0,6139	2,4219	0,6181	-189,8804
	ABC	2,3301	3,5205	0,4063	0,2294	0,3424	0,4281	0,3301	0,5205	1,4063	0,3384	0,6133	2,4058	0,6150	-189,8811
	NR	2,4878	3,2840	0,1400	0,1937	0,3365	0,4589	0,4878	0,2840	1,1400	0,4317	0,4171	1,7585	0,5266	-190,0563
-0,5	GA	2,7447	2,9720	-0,2266	0,0953	0,1244	0,1936	0,7447	-0,0280	0,2734	0,6499	0,1252	0,2683	0,3963	-190,3050
	PSO	2,5093	3,2652	0,1088	0,2155	0,3436	0,4555	0,5093	0,2652	0,6088	0,4749	0,4139	0,8261	0,6210	-190,1800
	DE	2,3222	3,5216	0,4178	0,2174	0,3328	0,4130	0,3222	0,5216	0,9178	0,3212	0,6048	1,2554	0,7892	-190,0304
	ABC	2,3276	3,5087	0,4006	0,2182	0,3436	0,4321	0,3276	0,5087	0,9006	0,3255	0,6023	1,2432	0,7845	-190,0313
	NR	2,8317	2,7610	-0,5195	0,0526	0,0667	0,1039	0,8317	-0,2390	-0,0195	0,7444	0,1239	0,1043	0,3415	-190,3304
0,5	GA	2,7651	2,9632	-0,2390	0,0935	0,1158	0,1875	0,7651	-0,0368	-0,7390	0,6789	0,1172	0,7336	0,6849	-189,6654
	PSO	2,5269	3,2559	0,0989	0,2151	0,3353	0,4581	0,5269	0,2559	-0,4011	0,4927	0,4007	0,6190	0,6256	-189,5184
	DE	2,3187	3,5464	0,4438	0,2300	0,3339	0,4052	0,3187	0,5464	-0,0562	0,3316	0,6325	0,4084	0,5316	-189,3689
	ABC	2,3191	3,5391	0,4333	0,2300	0,3452	0,4241	0,3191	0,5391	-0,0667	0,3318	0,6359	0,4286	0,5390	-189,3696
	NR	2,8328	2,7097	-0,5829	0,0519	0,0472	0,1173	0,8328	-0,2903	-1,0829	0,7455	0,1315	1,2899	0,8974	-189,7031
1	GA	2,7563	2,9640	-0,2297	0,0956	0,1192	0,1927	0,7563	-0,0360	-1,2297	0,6676	0,1205	1,7048	0,5666	-189,6271
	PSO	2,5013	3,2839	0,1369	0,2255	0,3520	0,4639	0,5013	0,2839	-0,8631	0,4768	0,4327	1,2088	0,4408	-189,4945
	DE	2,3396	3,5129	0,4118	0,2341	0,3398	0,4119	0,3396	0,5129	-0,5882	0,3494	0,6029	0,7579	0,3431	-189,3509
	ABC	2,3400	3,5017	0,3958	0,2335	0,3562	0,4394	0,3400	0,5017	-0,6042	0,3491	0,6079	0,8044	0,3492	-189,3516
	NR	2,3499	2,2523	-0,8864	0,3804	0,1571	0,0434	0,3499	-0,7477	-1,8864	0,5028	0,7162	3,6019	0,8216	-198,4337

Çizelge 5.10. Dönüştürülmüş Fréchet dağılımı parametre tahmin değerleri ( $n = 200$ ,  $\mu = 2$ ,  $\sigma = 3$ )

$\lambda$	Algoritma	Ortalama			Varyans			Yan			MSE			MAPE	$l$
		$\hat{\mu}$	$\hat{\sigma}$	$\hat{\lambda}$	$\hat{\mu}$	$\hat{\sigma}$	$\hat{\lambda}$	$\hat{\mu}$	$\hat{\sigma}$	$\hat{\lambda}$	$\hat{\mu}$	$\hat{\sigma}$	$\hat{\lambda}$		
-1	GA	2,7551	2,9225	-0,2956	0,0727	0,1124	0,1816	0,7551	-0,0775	0,7044	0,6429	0,1184	0,6778	0,3909	-380,7001
	PSO	2,4912	3,2614	0,0950	0,2150	0,3626	0,5027	0,4912	0,2614	1,0950	0,4562	0,4309	1,7016	0,5168	-380,4726
	DE	2,3138	3,5056	0,3838	0,2277	0,3680	0,4873	0,3138	0,5056	1,3838	0,3261	0,6236	2,4022	0,6070	-380,2239
	ABC	2,3134	3,5021	0,3786	0,2278	0,3732	0,4973	0,3134	0,5021	1,3786	0,3260	0,6254	2,3979	0,6056	-380,2255
	NR	2,3716	3,4225	0,3003	0,2035	0,3579	0,4752	0,3716	0,4225	1,3003	0,3416	0,5365	2,1659	0,5761	-380,4532
-0,5	GA	2,7510	2,9193	-0,3012	0,0744	0,1185	0,1856	0,7510	-0,0807	0,1988	0,6383	0,1250	0,2252	0,3603	-381,1947
	PSO	2,4991	3,2411	0,0682	0,2136	0,3647	0,5031	0,4991	0,2411	0,5682	0,4627	0,4229	0,8260	0,6045	-380,9985
	DE	2,3029	3,5105	0,3889	0,2235	0,3672	0,4897	0,3029	0,5105	0,8889	0,3152	0,6278	1,2799	0,7832	-380,7426
	ABC	2,3037	3,5063	0,3836	0,2224	0,3713	0,4963	0,3037	0,5063	0,8836	0,3146	0,6277	1,2771	0,7824	-380,7441
	NR	2,8371	2,7432	-0,5391	0,0242	0,0371	0,0635	0,8371	-0,2568	-0,0391	0,7250	0,1030	0,0650	0,3059	-381,2222
0,5	GA	2,7592	2,9124	-0,3059	0,0689	0,1086	0,1741	0,7592	-0,0876	-0,8059	0,6453	0,1163	0,8236	0,7287	-380,9970
	PSO	2,4919	3,2508	0,0852	0,2097	0,3647	0,5024	0,4919	0,2508	-0,4148	0,4517	0,4275	0,6745	0,6527	-380,8046
	DE	2,3351	3,4703	0,3455	0,2242	0,3744	0,4950	0,3351	0,4703	-0,1545	0,3365	0,5956	0,5188	0,5840	-380,5727
	ABC	2,3389	3,4614	0,3335	0,2246	0,3823	0,5066	0,3389	0,4614	-0,1665	0,3394	0,5952	0,5343	0,5900	-380,5745
	NR	2,8252	2,6960	-0,5968	0,0250	0,0334	0,0794	0,8252	-0,3040	-1,0968	0,7059	0,1258	1,2823	0,9039	-381,0190
1	GA	2,7630	2,9094	-0,3091	0,0649	0,0997	0,1627	0,7630	-0,0906	-1,3091	0,6470	0,1079	1,8764	0,5923	-381,0677
	PSO	2,4902	3,2562	0,0902	0,2144	0,3618	0,5020	0,4902	0,2562	-0,9098	0,4548	0,4274	1,3298	0,4548	-380,8362
	DE	2,3328	3,4798	0,3583	0,2228	0,3658	0,4764	0,3328	0,4798	-0,6417	0,3335	0,5960	0,8882	0,3590	-380,6176
	ABC	2,3328	3,4754	0,3521	0,2223	0,3718	0,4864	0,3328	0,4754	-0,6479	0,3331	0,5977	0,9062	0,3614	-380,6193
	NR	2,3424	2,2633	-0,8731	0,3496	0,1582	0,0494	0,3424	-0,7367	-1,8731	0,4669	0,7008	3,5580	0,8138	-397,9558

## 6. UYGULAMALAR

Bu bölümde gerçek veri setleri üzerinde meta-sezgisel ve iteratif algoritmaların parametre tahmini üzerindeki etkileri incelenmiştir. Önceki bölümde tanılan dönüştürülmüş dağılımlar ile modellenebilecek veri setleri ele alınarak farklı algoritmaların aynı veri seti üzerinde performansları karşılaştırılmıştır.

Parametre tahmin algoritmalarının performansı Akaike Bilgi Kriteri (AIC), Bayes Bilgi Kriteri (BIC) ve Kolmogorov-Smirnov (KS) testi kullanılarak karşılaştırılmıştır. KS, verilerin ampirik dağılım fonksiyonu  $\hat{F}(x)$  ile değerlendirilen dönüştürülmüş dağılımın teorik dağılım fonksiyonu  $F(x)$  arasındaki uyumun iyiliğini değerlendiren parametrik olmayan bir testtir. KS istatistiği bu iki kümülatif dağılım arasındaki maksimum mutlak farkı ölçer. Sıfır hipotezi ( $H_0$ ), verilerin belirli bir dönüştürülmüş dağılımdan geldiğini varsayar.  $\alpha = 0,05$  anlamlılık düzeyinde sıfır hipotezinin reddedilmesi, veriler ile teorik dağılım arasında istatistiksel olarak anlamlı bir fark olduğunu gösterir. KS test istatistiği Denklem (6.1)'de sunulmaktadır.

$$KS = \max |\hat{F}(x) - F(x)| \quad (6.1)$$

AIC ve BIC kriterleri parametre sayısı ( $p$ ) ve log-olabilirlik değerlerine dayalı ceza fonksiyonlarıdır. AIC ve BIC sırasıyla Denklem (6.2) ve (6.3)'te sunulmaktadır; burada  $\log(L)$ , modelin olabilirlik fonksiyonunun logaritmasını temsil eder,  $n$  örneklem boyutunu ve  $p$  modeldeki parametre sayısını gösterir. En düşük AIC ve BIC değerlerine sahip algoritmanın diğerlerinden iyi olduğu kabul edilir.

$$AIC = -2 \times \log(L) + 2 \times p \quad (6.2)$$

$$BIC = -2 \times \log(L) + p \times \log(n) \quad (6.3)$$

### 6.1 Dönüştürülmüş Üstel Dağılım

Bu analize ilişkin veriler, Owoloko vd. (2015) tarafından yapılan, dönüştürülmüş üstel dağılımın uygulanabilirliğinin, %90 stres seviyesinde sabit basınç altında Kevlar 373/epoksinin yorulma ömrünü modellemek için araştırıldığı çalışmadan elde edilmiştir. Bu çalışmada veriler, bu koşullar altında kırıkların oluşması için geçen süreyi temsil eder. Bu analiz, dönüştürülmüş üstel dağılımı daha fazla araştırmak için aynı veri kümesinden yararlanır, ancak odak noktası parametre tahmin tekniklerine kaydırılır. Burada bu amaçla meta-sezgisel optimizasyon algoritmalarından yararlanılmaktadır. Veri seti Çizelge 6.1'de sunulmaktadır. Parametre tahminleri ve dağılıma uygunluk testine ilişkin sonuçlar Çizelge 6.2'de verilmiştir. Ek olarak, Şekil 6.1'de Ampirik (gözlenen) ve teorik (beklenen) dağılımları görsel olarak karşılaştırılmıştır.

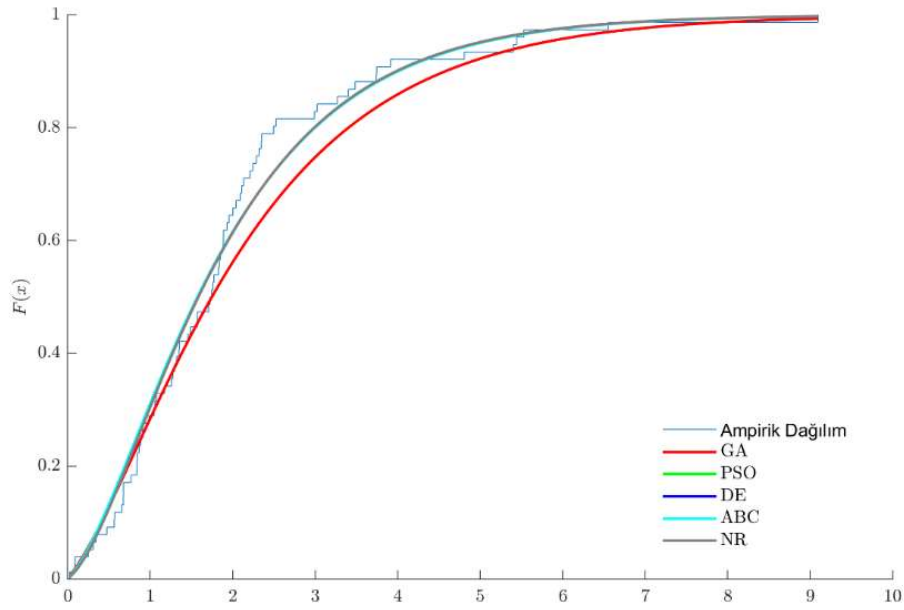
**Çizelge 6.1.** %90 stres seviyesinde Kevlar 373/epoksinin kırılma dayanım değerleri

0,0251	0,0886	0,0891	0,2501	0,3113	0,3451	0,4763	0,5650	0,5671
0,6566	0,6748	0,6751	0,6753	0,7696	0,8375	0,8391	0,8425	0,8645
0,8851	0,9113	0,9120	0,9836	1,0483	1,0596	1,0773	1,1733	1,2570
1,2766	1,2985	1,3211	1,3503	1,3551	1,4595	1,4880	1,5728	1,5733
1,7083	1,7263	1,7460	1,7630	1,7746	1,8275	1,8375	1,8503	1,8808
1,8878	1,8881	1,9316	1,9558	2,0048	2,0408	2,0903	2,1093	2,1330
2,2100	2,2460	2,2878	2,3203	2,3470	2,3513	2,4951	2,5260	2,9911
3,0256	3,2678	3,4045	3,4846	3,7433	3,7455	3,9143	4,8073	5,4005
5,4435	5,5295	6,5541	9,0960					

KS testi sonuçlarına göre, verilerin dönüştürülmüş üstel dağılımla modellenebileceği (Çizelge 6.2'da p-değerleri > 0,05) görülmektedir. En iyi modeli belirlemek için AIC ve BIC seçim kriterlerine göre GA dışındaki tüm algoritmaların benzer sonuçlar verdiği görülmektedir. PSO ve DE algoritmalarıyla elde edilen parametre tahminlerinin marjinal olarak daha düşük hatalara ulaştığı görülmektedir.

**Çizelge 6.2.** Kırılma dayanım dönüştürülmüş Üstel dağılımın parametre tahminleri

Algoritma	$\hat{\theta}$	$\hat{\lambda}$	$l$	AIC	BIC	KS	p- değeri
GA	1,6287	-0,7025	-122,7382	249,4765	249,2381	0,1522	0,0529
PSO	1,3763	-0,8487	-121,5166	247,0331	246,7947	0,0965	0,4504
DE	1,3763	-0,8487	-121,5166	247,0331	246,7947	0,0965	0,4504
ABC	1,3975	-0,8006	-121,5699	247,1397	246,9014	0,0965	0,4500
NR	1,3763	-0,8487	-121,5166	247,0332	246,7948	0,0965	0,4504

**Şekil 6.1.** Kırılma dayanım verileri için ampirik ve teorik dağılım fonksiyonları

## 6.2 Dönüştürülmüş Weibull Dağılımı

Bu analiz, orijinal olarak Nichols ve Padgett (2006) tarafından incelenen, karbon fiberlerin kopma gerilimine (GPa cinsinden) ilişkin 100 gözlem içeren bir veri setini kullanır. Aryal ve Tsokos (2011) dönüştürülmüş Weibull dağılımının bu verilere uygulanabilirliğini araştırmıştır. Veri seti Çizelge 6.3'de sunulmaktadır. Çizelge 6.4, ilgili uygunluk testi istatistikleriyle birlikte her algoritma tarafından elde edilen teta ve lambda için tahmini değerleri sunar. Ek olarak, Şekil 6.2 ampirik (gözlenen) ve teorik (beklenen) dağılımları görsel olarak karşılaştırır.

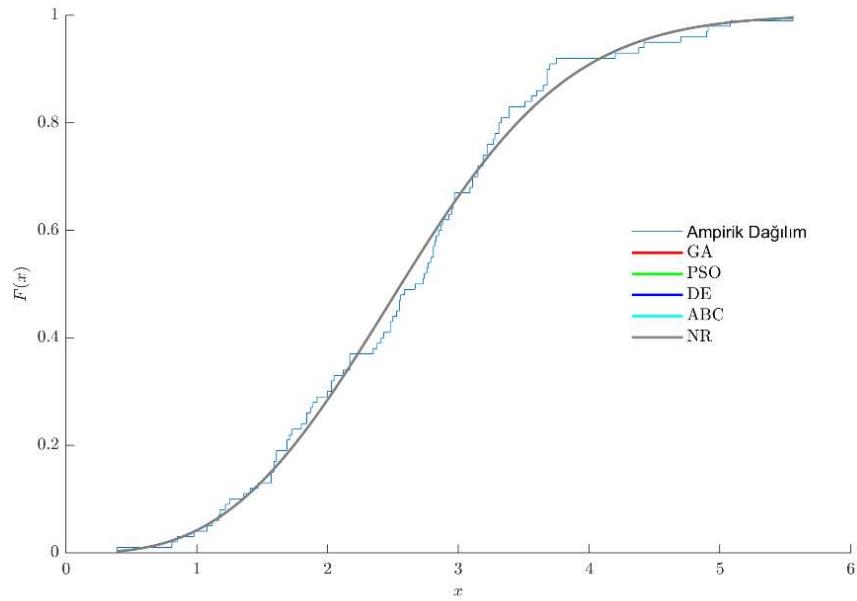
**Çizelge 6.3.** Nichols ve Padgett (2006) tarafından incelenen, karbon fiberler stres dayanım veri seti

3,7000	2,7400	2,7300	2,5000	3,6000	3,1100	3,2700	2,8700	1,4700	3,1100
4,4200	2,4100	3,1900	3,2200	1,6900	3,2800	3,0900	1,8700	3,1500	4,9000
3,7500	2,4300	2,9500	2,9700	3,3900	2,9600	2,5300	2,6700	2,9300	3,2200
3,3900	2,8100	4,2000	3,3300	2,5500	3,3100	3,3100	2,8500	2,5600	3,5600
3,1500	2,3500	2,5500	2,5900	2,3800	2,8100	2,7700	2,1700	2,8300	1,9200
1,4100	3,6800	2,9700	1,3600	0,9800	2,7600	4,9100	3,6800	1,8400	1,5900
3,1900	1,5700	0,8100	5,5600	1,7300	1,5900	2,0000	1,2200	1,1200	1,7100
2,1700	1,1700	5,0800	2,4800	1,1800	3,5100	2,1700	1,6900	1,2500	4,3800
1,8400	0,3900	3,6800	2,4800	0,8500	1,6100	2,7900	4,7000	2,0300	1,8000
1,5700	1,0800	2,0300	1,6100	2,1200	1,8900	2,8800	2,8200	2,0500	3,6500

KS testi, verilerin dönüştürülmüş üstel dağılımla yeterince tanımlanabileceğini (Çizelge 6.4'te  $p$ -değerleri  $> 0,05$ ) göstermektedir. En iyi modeli belirlemek için AIC ve BIC seçim kriterleri incelediğinde tüm algoritmaların benzer sonuçlar verdiği görülmektedir.

**Çizelge 6.4.** Karbon fiber stres dayanım veri etini için dönüştürülmüş Weibull dağılımının parametre tahminleri

Algoritma	$\hat{\eta}$	$\hat{\sigma}$	$\hat{\lambda}$	$l$	AIC	BIC	KS	p-değeri
GA	2,9935	3,4126	0,6789	-141,1349	288,2698	288,2698	0,0642	0,7797
PSO	2,9935	3,4126	0,6789	-141,1349	288,2698	288,2698	0,0642	0,7797
DE	2,9934	3,4126	0,6789	-141,1349	288,2698	288,2698	0,0642	0,7796
ABC	2,9936	3,4119	0,6781	-141,1349	288,2698	288,2698	0,0642	0,7795
NR	2,9934	3,4125	0,6788	-141,1399	288,2798	288,2798	0,0642	0,7795



**Şekil 6.2.** Karbon fiber stres dayanım veri seti için ampirik dağılım ve teorik dağılım fonksiyonları.

### 6.3 Dönüştürülmüş Fréchet Dağılımı

Mahmoud ve Mandouh (2013) tarafından yapılan çalışmada, karbon fiberlerin kopma gerilimine (GPa cinsinden) ilişkin 100 gözlem içeren sansürlü bir veri seti elde edilmiştir. Bu veriler için elde edilen KS testi sonuçlarına göre (Çizelge 6.6) verilerin dönüştürülmüş Fréchet dağılımıyla modellenebileceği görülmektedir ( $p$ -değerleri  $> 0,05$ ) Bu analizde, dönüştürülmüş Fréchet dağılımında parametre tahmini için meta-sezgisel optimizasyon algoritmaları kullanılarak incelenmiştir. Veri seti

Çizelge 6.5'te sunulmaktadır. Çizelge 6.6'da ilgili uygunluk test istatistikleriyle birlikte her algoritma tarafından elde edilen  $\mu$ ,  $\sigma$  ve  $\lambda$  için tahmin değerleri verilmiştir. Ek olarak, Şekil 6.3'te ampirik (gözlenen) ve teorik (beklenen) dağılım görselleri verilmiştir.

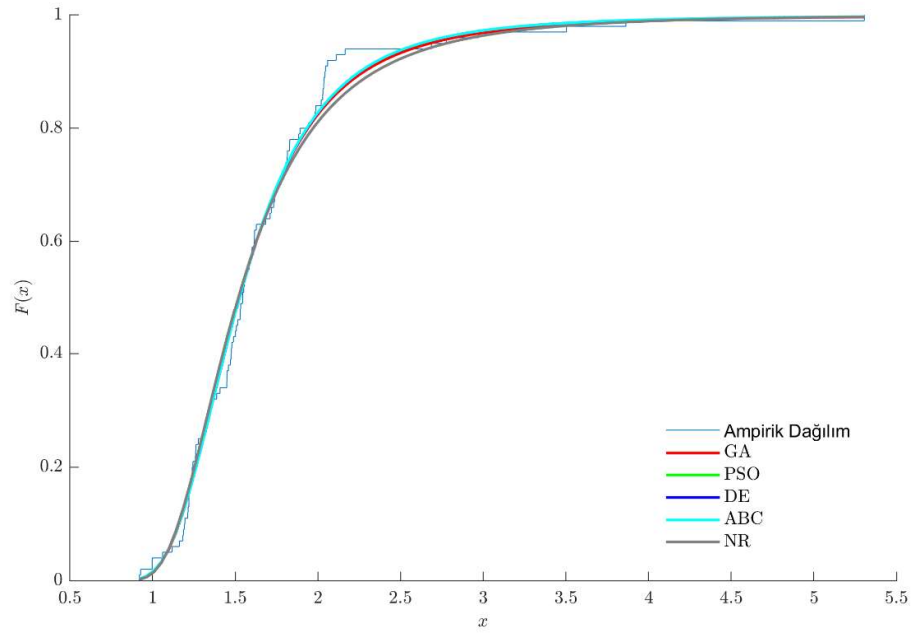
**Çizelge 6.5.** Mahmoud (2013) tarafından yapılan karbon fiberler stres dayanım veri seti

0,9200	0,9280	0,9970	0,9971	1,0610	1,1170	1,1620	1,1830	1,1870	1,1920
1,1960	1,2130	1,2150	1,2199	1,2200	1,2240	1,2250	1,2280	1,2370	1,2400
1,2440	1,2590	1,2610	1,2630	1,2760	1,3100	1,3210	1,3290	1,3310	1,3370
1,3510	1,3590	1,3880	1,4080	1,4490	1,4497	1,4500	1,4590	1,4710	1,4750
1,4770	1,4800	1,4890	1,5010	1,5070	1,5150	1,5300	1,5304	1,5330	1,5440
1,5443	1,5520	1,5560	1,5620	1,5660	1,5850	1,5860	1,5990	1,6020	1,6140
1,6160	1,6170	1,6280	1,6840	1,7110	1,7180	1,7330	1,7380	1,7430	1,7590
1,7770	1,7940	1,7990	1,8060	1,8140	1,8160	1,8280	1,8300	1,8840	1,8920
1,9440	1,9720	1,9840	1,9870	2,0200	2,0304	2,0290	2,0350	2,0370	2,0430
2,0460	2,0590	2,1110	2,1650	2,6860	2,7780	2,9720	3,5040	3,8630	5,3060

En iyi modeli belirlemek için AIC ve BIC seçim kriterleri kullanılmıştır. NR dışındaki tüm algoritmaların çok benzer sonuçlar verdiği görülmektedir. AIC ve BIC kriterlerine göre PSO ve DE marjinal olarak daha küçük değerlere sahiptir.

**Çizelge 6.6.** Karbon fiber stres dayanım veri setini için dönüştürülmüş Frechet dağılımının parametre tahminleri

Algoritma	$\hat{\mu}$	$\hat{\sigma}$	$\hat{\lambda}$	$l$	AIC	BIC	KS	p- değeri
GA	3,4828	1,5675	0,7612	-52,5141	111,0281	111,0281	0,0780	0,5506
PSO	3,3636	1,5913	0,8517	-52,4122	110,8243	110,8243	0,0754	0,5930
DE	3,3631	1,5914	0,8521	-52,4122	110,8243	110,8243	0,0754	0,5938
ABC	3,3688	1,5900	0,8492	-52,4123	110,8247	110,8247	0,0760	0,5830
NR	4,2500	1,4210	0,1200	-53,7380	113,4760	113,476	0,0886	0,3901



**Şekil 6.4.** Karbon fiber stres dayanım veri seti için Ampirik dağılım ve Teorik dağılım.

## 7. SONUÇLAR VE ÖNERİLER

Bu çalışmada, dört meta-sezgisel algoritmanın (GA, PSO, ABC ve DE) performansı, dönüştürülmüş üstel, Weibull ve Fréchet dağılımlarının parametre tahmini üzerinde uygulanarak En çok olabilirlik tahmini yaklaşımını kullanan NR yöntemi ile karşılaştırıldı. Farklı dönüştürülmüş dağılım parametresi ( $\lambda$ ) değerleri ve örnek boyutları kullanılarak kapsamlı simülasyon çalışması ile sonuçlar elde edildi. Yan, MSE, MAPE ve log-olabilirlik değerleri simülasyon çalışmasında performans değerlendirme kriteri olarak kullanılmıştır.

Simülasyon sonuçları genel olarak incelendiğinde dönüştürülmüş dağılım parametrelerinin tahmini için en uygun algoritmanın seçiminin spesifik  $\lambda$  değerine ve ilgilenilen parametreye bağlılığı üzerinde duruldu. Negatif  $\lambda$  değerleri ve  $\theta$  tahmini ile dönüştürülmüş üstel dağılım için NR'nin uygun olduğu gözlemlendi. Pozitif lambda değerleri için GA ve PSO, tüm örnek boyutlarında hem  $\theta$  hem de  $\lambda$  tahmininde üstün performans gösterdi. Genel performansın bir öncelik olduğu ve  $\lambda$ 'nın bilinmediği durumlarda ise GA veya PSO gibi daha sağlam bir algoritma, farklı  $\lambda$  koşullarındaki göreceli tutarlılıkları nedeniyle daha güvenli bir seçenek olarak ortaya çıkmaktadır.

Dönüştürülmüş Weibull dağılımı için negatif  $\lambda$  değerleri için DE tercih edilirken, pozitif  $\lambda$  değerleri için GA ve PSO tüm parametrelerde üstün performans göstermiştir. Bununla birlikte, genel performans ve sağlamlığın çok önemli olduğu ve gerçek  $\lambda$  değerinin bilinmediği durumlarda DE, GA veya PSO, farklı  $\lambda$  koşullarındaki göreceli tutarlılıkları nedeniyle daha güvenli bir seçim olabilir.

Dönüştürülmüş Fréchet dağılımı için NR, orta derecede negatif  $\lambda$  değerleri için meta-sezgisel algoritmalarından daha iyi performans göstermektedir. Ancak aşırı negatif  $\lambda$  değerleri ( $\lambda = -1$ ) ve pozitif  $\lambda$  değerleri için meta-sezgisel algoritmalar NR'den üstün

performans sergilemiştir.  $\lambda$  negatif olduğunda GA veya NR (spesifik değere bağlı olarak) tercih edilebilir. Pozitif  $\lambda$  için DE en sağlam seçim olarak öne çıkmaktadır.

Ayrıca, bu dönüştürülmüş dağılımların her birine gerçek bir veri seti uygulandı ve parametre tahminleri beş algoritmanın tümü (GA, PSO, ABC, DE ve NR) ile tahmin edildi. Sonuçlar meta-sezgisel algoritmaların genel olarak NR algoritmasından daha iyi performans gösterdiğini göstermiştir. AIC ve BIC bilgi kriterlerine dayanarak dönüştürülmüş üstel dağılım için PSO veya DE önerilirken; dönüştürülmüş Weibull dağılımı için meta-sezgisel algoritmaların hepsi benzer sonuçlar vermiştir ve dönüştürülmüş Fréchet dağıtımı için DE diğerlerinden öne çıkmıştır.

Bu kapsamlı simülasyon çalışmasından elde edilen sonuçlara dayanarak, aşağıdaki gelecekteki çalışmalar önerilebilir:

1. Meta-sezgisel algoritmaların ve NR yönteminin performansını farklı parametre tahmini problemleri üzerinde test etmek için diğer dönüştürülmüş olasılık dağılımlarını kullanmak.
2. Meta-sezgisel algoritmaların hibrit yaklaşımlarını geliştirmek ve performanslarını karşılaştırmak.
3. Parametre tahmini için Bayes yaklaşımı kullanmak ve bu yaklaşımın meta-sezgisel algoritmalarla nasıl karşılaştırıldığını araştırmak.

**KAYNAKLAR**

- Abbasi, B., Eshragh Jahromi, A. H., Arkat, J., & Hosseinkouchack, M. (2006). Estimating the parameters of Weibull distribution using simulated annealing algorithm. *Applied Mathematics and Computation*, 183(1), 85-93. <https://doi.org/10.1016/j.amc.2006.05.063>
- Abdul-Moniem, I. B., & Seham, M. (2015). Transmuted Gompertz Distribution. *Çinde Transmuted Gompertz Distribution. Computational and Applied Mathematics (C. 1, Sayı 3)*. <http://www.aascit.org/journal/cam>
- Aryal, G. R., & Tsokos, C. P. (2009). On the transmuted extreme value distribution with application. *Nonlinear Analysis, Theory, Methods and Applications*, 71(12). <https://doi.org/10.1016/j.na.2009.01.168>
- Aryal, G. R., & Tsokos, C. P. (2011). Transmuted Weibull Distribution: A Generalization of the Weibull Probability Distribution. *European Journal of Pure And Applied Mathematics*, 4(2), 89-102. [www.ejpam.com](http://www.ejpam.com)
- Ashour, S. K., & Eltehiwy, M. A. (2013). Transmuted Lomax Distribution. *American Journal of Applied Mathematics and Statistics*, 1(6), 121-127. <https://doi.org/10.12691/ajams-1-6-3>
- Celik, N. (2018). Some Cubic Rank Transmuted Distributions. *Journal of Applied Mathematics, Statistics and Informatics*, 14(2), 27-43. <https://doi.org/10.2478/jamsi-2018-0011>
- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. <https://doi.org/10.1080/00949650903530745>
- Eugene, N., Lee, C., & Famoye, F. (2002). Beta-normal distribution and its applications. *Communications in Statistics - Theory and Methods*, 31(4), 497-512. <https://doi.org/10.1081/STA-120003130>
- Gendreau, M., & Potvin, J.-Y. (2005). Metaheuristics in Combinatorial Optimization. *Çinde Annals of Operations Research (C. 140)*.
- Granzotto, D. C. T., Louzada, F., & Balakrishnan, N. (2017). Cubic rank transmuted distributions: inferential issues and applications. *Journal of Statistical Computation and Simulation*, 87(14), 2760-2778. <https://doi.org/10.1080/00949655.2017.1344239>
- Guedes, K. S., de Andrade, C. F., Rocha, P. A. C., Manguiera, R. dos S., & de Moura, E. P. (2020). Performance analysis of metaheuristic optimization algorithms in estimating the parameters of several wind speed distributions. *Applied Energy*, 268. <https://doi.org/10.1016/j.apenergy.2020.114952>
- Gupta, R. C., Gupta, P. L., & Gupta, R. D. (1998). Modeling failure time data by Lehman alternatives. *Communications in Statistics - Theory and Methods*, 27(4), 887-904. <https://doi.org/10.1080/03610929808832134>
- Jiang, H., Wang, J., Dong, Y., & Lu, H. (2015). Comprehensive assessment of wind resources and the low-carbon economy: An empirical study in the Alxa and Xilin Gol Leagues of inner Mongolia, China. *Çinde Renewable and Sustainable Energy Reviews (C. 50, ss. 1304-1319)*. Elsevier Ltd. <https://doi.org/10.1016/j.rser.2015.05.082>
- Jiang, H., Wang, J., Wu, J., & Geng, W. (2017). Comparison of numerical methods and metaheuristic optimization algorithms for estimating parameters for wind energy potential assessment in low wind regions. *Çinde Renewable and Sustainable Energy Reviews (C. 69, ss. 1199-1217)*. Elsevier Ltd. <https://doi.org/10.1016/j.rser.2016.11.241>

- John, H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan press Ann Arbor MI.
- Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3), 459-471. <https://doi.org/10.1007/s10898-007-9149-x>
- Karakoca, A., & Pekgör, A. (2019). Maximum Likelihood Estimation of the Parameters of Progressively Type-2 Censored Samples From Weibull Distribution Using Genetic Algorithm. *Academic Platform Journal of Engineering and Science*, 1-1. <https://doi.org/10.21541/apjes.452564>
- Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, 4, 1942-1948 c.4. <https://doi.org/10.1109/ICNN.1995.488968>
- Khamees, A. K., Abdelaziz, A. Y., Eskaros, M. R., & Attia, M. A. (2021). Investigation of Different Probability Distribution Functions for Wind Speed Modelling Using Classical and Novel Metaheuristic Methods. *22nd International Middle East Power Systems Conference, MEPCON 2021 - Proceedings*, 26-31. <https://doi.org/10.1109/MEPCON50283.2021.9686280>
- Liu, F. J., Chen, P. H., Kuo, S. S., Su, D. C., Chang, T. P., Yu, Y. H., & Lin, T. C. (2011). Wind characterization analysis incorporating genetic algorithm: A case study in Taiwan Strait. *Energy*, 36(5), 2611-2619. <https://doi.org/10.1016/j.energy.2011.02.001>
- Mahmoud, M. R., & Mandouh, R. M. (2013). On the Transmuted Fréchet Distribution. *Içinde Journal of Applied Sciences Research (C. 9, Sayı 10)*.
- Marshall, A. W. (1997). A new method for adding a parameter to a family of distributions with application to the exponential and Weibull families. *Içinde Biometrika (C. 84, Sayı 3)*. <https://academic.oup.com/biomet/article/84/3/641/217183>
- Nichols, M. D., & Padgett, W. J. (2006). A bootstrap control chart for weibull percentiles. *Quality and Reliability Engineering International*, 22(2), 141-151. <https://doi.org/10.1002/qre.691>
- Osman, I. H. (2003). Focused issue on applied meta-heuristics. *Computers & Industrial Engineering*, 44, 205-207. [www.elsevier.com/locate/dsw](http://www.elsevier.com/locate/dsw)
- Owoloko, E. A., Oguntunde, P. E., & Adejumo, A. O. (2015). Performance rating of the transmuted exponential distribution: an analytical approach. *SpringerPlus*, 4(1), 1-15. <https://doi.org/10.1186/s40064-015-1590-6>
- Pobočíková, I., Sedláčková, Z., & Michalková, M. (2018). Transmuted Weibull distribution and its applications. *MATEC Web of Conferences*, 157. <https://doi.org/10.1051/mateconf/201815708007>
- Saraçoğlu, B., & Tanış, C. (2018). A new statistical distribution: Cubic rank transmuted Kumaraswamy distribution and its properties. *Journal of the National Science Foundation of Sri Lanka*, 46(4), 505-518. <https://doi.org/10.4038/JNSFSR.V46I4.8626>
- Shaw, W. T., & Buckley, I. R. C. (2009). The alchemy of probability distributions: beyond Gram-Charlier expansions, and a skew-kurtotic-normal distribution from a rank transmutation map. <http://arxiv.org/abs/0901.0434>
- Shuaib Khan, M., & King, R. (2013). Transmuted Modified Weibull Distribution: A Generalization of the Modified Weibull Probability Distribution. *European Journal of Pure And Applied Mathematics*, 6(1), 66-88. [www.ejpam.com](http://www.ejpam.com)

- Sörensen, K., & Glover, F. (2015). Metaheuristics. İçinde Encyclopedia of Operations Research and Management Science (3. bs). Springer. <http://metaheuristics.eu>
- Storn, R., & Price, K. (1997). Differential Evolution – A Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces. *Journal of global optimization*, 11, 341-359.
- Taniş, C., & Saraçoğlu, B. (2023). Cubic rank transmuted generalized Gompertz distribution: properties and applications. *Journal of Applied Statistics*, 50(1), 195-213. <https://doi.org/10.1080/02664763.2022.2025585>
- Wang, J., Huang, X., Li, Q., & Ma, X. (2018). Comparison of seven methods for determining the optimal statistical distribution parameters: A case study of wind energy assessment in the large-scale wind farms of China. *Energy*, 164, 432-448. <https://doi.org/10.1016/j.energy.2018.08.201>
- Weibull, W. (1951). A Statistical Distribution Function of Wide Applicability. *Journal of Applied Mechanics*, 293-297.
- Yalçınkaya, A., Şenoğlu, B., & Yolcu, U. (2018). Maximum likelihood estimation for the parameters of skew normal distribution using genetic algorithm. *Swarm and Evolutionary Computation*, 38, 127-138. <https://doi.org/10.1016/j.swevo.2017.07.007>