

**T.C.
MUĞLA SITKI KOÇMAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

İSTATİSTİK ANABİLİM DALI

**KATEGORİK BAĞIMLI DEĞİŞKEN MODELLERİNDE
PARAMETRE TAHMİNİ İÇİN KLASİK
YAKLAŞIMLARA ALTERNATİF BAZI SEZGİSEL
OPTİMİZASYON TEKNİKLERİ**

YÜKSEK LİSANS TEZİ

EMRE DEMİR

HAZİRAN 2014

MUĞLA

T.C.
MUĞLA SITKI KOÇMAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İSTATİSTİK ANABİLİM DALI

KATEGORİK BAĞIMLI DEĞİŞKEN MODELLERİNDE
PARAMETRE TAHMİNİ İÇİN KLASİK
YAKLAŞIMLARA ALTERNATİF BAZI SEZGİSEL
OPTİMİZASYON TEKNİKLERİ

YÜKSEK LİSANS TEZİ

EMRE DEMİR

HAZİRAN 2014

MUĞLA

MUGLA SITKI KOÇMAN ÜNİVERSİTESİ

Fen Bilimleri Enstitüsü

TEZ ONAYI

EMRE DEMİR tarafından hazırlanan “KATEGORİK BAĞIMLI DEĞİŞKEN MODELLERİNDE PARAMETRE TAHMİNİ İÇİN KLASİK YAKLAŞIMLARA ALTERNATİF BAZI SEZGİSEL OPTİMİZASYON TEKNİKLERİ” başlıklı tezinin, 25/06/2014 tarihinde aşağıdaki jüri tarafından İstatistik Anabilim Dalı’nda yüksek lisans derecesi için gerekli şartları sağladığı oybirliği/oyçokluğu ile kabul edilmiştir.

TEZ SINAV JURİSİ

Doç. Dr. Ayten YİĞİTER (**Jüri Başkanı**)

İstatistik Anabilim Dalı,
Hacettepe Üniversitesi, Ankara

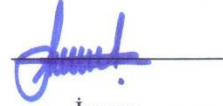
İmza:



Doç. Dr. Özge AKKUŞ (**Danışman**)

İstatistik Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

İmza:



Doç. Dr. Atilla GÖKTAŞ (**Üye**)

İstatistik Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

İmza:



ANA BİLİM DALI BAŞKANLIĞI ONAYI

Doç. Dr. Dursun AYDIN

İstatistik Ana Bilim Dalı Başkanı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

İmza:



Doç. Dr. Özge AKKUŞ

Danışman, İstatistik Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

İmza:



Savunma Tarihi: 25/06/2014

Tez çalışmalarım sırasında elde ettiğim ve sunduğum tüm sonuç, doküman, bilgi ve belgelerin tarafımdan bizzat ve bu tez çalışması kapsamında elde edildiğini; akademik ve bilimsel etik kurallarına uygun olduğunu beyan ederim. Ayrıca, akademik ve bilimsel etik kuralları gereği bu tez çalışması sırasında elde edilmemiş başkalarına ait tüm orijinal bilgi ve sonuçlara atıf yapıldığını da beyan ederim.

Emre Demir

25/06/2014

ÖZET
KATEGORİK BAĞIMLI DEĞİŞKEN MODELLERİNDE PARAMETRE
TAHMİNİ İÇİN KLASİK YAKLAŞIMLARA ALTERNATİF BAZI
SEZGİSEL OPTİMİZASYON TEKNİKLERİ

Emre DEMİR

Yüksek Lisans Tezi

Fen Bilimleri Enstitüsü

İstatistik Anabilim Dalı

Danışman: Doç. Dr. Özge AKKUŞ

Haziran 2014, 105 sayfa

Bağımlı ve açıklayıcı değişken (ler) arasındaki ilişkinin araştırılması için yaygın olarak kullanılan doğrusal regresyon analizi, bağımlı değişkenin iki ya da daha fazla düzey içeren kategorik bir değişken olması durumunda bazı istatistiksel varsayım bozulmalarından dolayı kullanılamamaktadır. Bu durumda doğrusal regresyona alternatif bazı doğrusal olmayan modeller önerilmiştir. Lojistik Regresyon Modelleri bu alternatif modeller arasında en fazla tercih edilen modeller olmuştur. Bu modellerde parametre tahminleri genellikle geleneksel En Çok Olabilirlik, Yeniden Ağırlıklandırılmış İteratif En Küçük Kareler Yöntemi ve Minimum Logit Ki-Kare Yöntemi ile yapılmaktadır. Bu yöntemlerde en iyi parametre tahminlerine ulaşabilmek için gradyant optimizasyon tekniklerinden yararlanılır. Oysaki optimize edilmeye çalışılan fonksiyon her zaman türevlenebilir özellikte olmayabilir. Bu durumda, fonksiyonların sürekli (dolayısıyla türevlenebilir) olmasını gerektirmeyen alternatif optimizasyon tekniklerinin de bilinmesi gerekmektedir. Yapılan çalışmalar, bağımlı değişkenin sürekli olması durumunda Genetik Algoritma Yaklaşımının parametre tahminlerinde başarılı bir biçimde uygulandığını ortaya çıkarmıştır.

Bağımlı değişkenin kategorik olduğu modeller dikkate alındığında, parametrelerin etkin tahminlerine ulaşabilmek için Genetik Algoritma yaklaşımı kullanılarak yapılmış yeterli sayıda çalışmaya rastlanmamıştır. Bu çalışmada, bağımlı değişkenin kategorik olması durumunda parametrelerin tahmin edilmesinde bazı sezgisel optimizasyon yöntemlerinin etkinliği araştırılmıştır. Genetik Algoritmaya ek olarak Nelder-Mead Algoritması da ayrıntıları ile incelenmiştir. Lojistik regresyon analizinde klasik Newton-Raphson tahminleri, Genetik Algoritma ve Nelder-Mead tahminleri ile karşılaştırılmıştır. Son olarak, kategorik bağımlı değişken modellerinde Genetik Algoritma ve Nelder-Mead optimizasyon tekniklerinin kullanılabilirliği gösterilmiş ve tahminlerin Matlab kodları tanıtılmıştır.

Anahtar Kelimeler: Genetik Algoritma, Kategorik Veri, Nelder-Mead, Parametre Tahmini, En Çok Olabilirlik

ABSTRACT

SOME HEURISTIC OPTIMIZATION TECHNIQUES TO THE CLASSICAL APPROACHES FOR THE PARAMETER ESTIMATION IN CATEGORICAL DEPENDENT VARIABLE MODELS

Emre DEMİR

Master of Science (M.Sc.)

Graduate School of Natural and Applied Sciences

Department of Statistics

Supervisor: Assoc. Prof. Dr. Özge AKKUŞ

June 2014, 105 pages

Linear Regression Analysis which is widely used for examining the relationship between the dependent and explanatory variables could not be used because of some assumptions violations when the dependent variable is categorical with two or more levels. In such a case, some nonlinear models as alternatives to the linear regression have been proposed. Logistic Regression Models are most preferred models among all the alternatives. In these models, parameter estimation is generally made by the conventional methods of Maximum Likelihood, Iteratively Reweighted Least Squares Method and Minimum Logit Chi-Square Method. These methods require gradient optimization techniques to reach efficient parameter estimations. However, the objective function to be optimized does not have differentiable structure at all times. In this case, alternative optimization techniques should be known, which do not require the continuousness (so differentiability) of the function. Studies made in this field have brought to light that the Genetic Algorithm Approach to the parameter estimation has been successfully applied when the dependent variable is continuous. Taking into consideration of the categorical structure of the dependent variable, it has been realised that there is no adequate number of research made for obtaining efficient parameter estimation via the Genetic Algorithm approach.

In this study, efficiency of some heuristic optimization techniques in the parameter estimation has been examined in case of the categorical dependent variable models. In addition to the Genetic Algorithm, the use of the Nelder-Mead algorithm has also been analyzed, in detail. The classical Newton-Raphson estimations have been compared with the Genetic Algorithm and Nelder-Mead estimations. Finally, the applicability of the concerned optimization techniques in categorical dependent variable models has been shown and Matlab commands of the estimations have been introduced.

Keywords: Genetic Algorithm, Categorical Data, Nelder-Mead, Parameter Estimation, Maximum Likelihood.

Sevgili Eşim Sevil DEMİR'e

ÖNSÖZ

Bu tezin ana hatlarının belirlenmesinde ve oluşmasında bana her türlü yardımı gösteren ve rehber olan değerli hocam Sayın Doç. Dr. Özge AKKUŞ' a teşekkürü bir borç bilir, şükranlarımı sunarım.

Çalışmamın uygulama bölümü için gerekli olan verileri elde etmemde yardımlarını esirgemeyen Sayın Yrd. Doç. Dr. Engin ŞENEL' e,

Ayrıca çalışmamın tüm aşamalarında beni destekleyen değerli aileme ve eşim Sevil DEMİR' e minnettar olduğumu belirtmek isterim.

İÇİNDEKİLER

ÖNSÖZ.....	vii
İÇİNDEKİLER	viii
ÇİZELGELER DİZİNİ	x
ŞEKİLLER DİZİNİ	xi
SEMBOLLER VE KISALTMALAR DİZİNİ	xii
1. GİRİŞ	1
2. DOĞRUSAL OLMAYAN MODELLERDE PARAMETRE TAHMİNİ.....	6
2.1. Parametrelerde Doğrusallık ve Doğrusal Olmama Durumu	6
2.2. Doğrusal Olmayan Regresyon Analizi	10
2.2.1. Doğrusal Olmayan En Küçük Kareler ve En Çok Olabilirlik Tahmin Edicileri	12
3. LOJİSTİK REGRESYON ANALİZİ	15
3.1. LR ve DR Analizlerinin karşılaştırılması.....	19
3.2. LR Modeli	20
3.3. Odds ve Odds Oranı Kavramları	23
4. DOĞRUSAL OLMAYAN REGRESYON MODELLERİNDE PARAMETRELERİN TAHMİNİNDE KULLANILAN KLASİK OPTİMİZASYON TEKNİKLERİ.....	28
4.1. Newton-Raphson Algoritması	30
4.2. Nelder-Mead Algoritması.....	31
4.2.1. İki Boyutlu Örnek Bir Problem İçin NM metodunun Çıktıları.....	36
4.2.2. Matlab Programı İle Nelder-Mead Algoritmasının Çözümü	39
5. DOĞRUSAL OLMAYAN MODELLERDE PARAMETRE TAHMİNİNE GENETİK ALGORİTMA YAKLAŞIMI.....	43
5.1. Genetik Algoritmanın Uygulama Alanları	45
5.2. Genetik Algoritmaların Geleneksel Arama Metodlarından Farkı	46
5.3. Genetik Algoritmanın Çalışma Yapısı	47
5.3.1. Temsil Mekanizması Kodlama	49
5.3.2. Başlangıç Popülasyonu	52

5.3.3. Genetik Algoritma Operatörleri	53
5.3.3.1 Seçme Operatörleri.....	53
5.3.3.2. Çaprazlama Operatörleri	56
5.3.4. Şema Teoremi	59
5.4. Basit Bir Genetik Algoritma İşleyişi	61
6. UYGULAMA.....	65
6.1. Gerçek Veriler İle Uygulama	65
6.1.1. Uygulamada Kullanılan Veriler	65
6.1.2. LR Analizi ve Newton-Raphson Yöntemi İle Parametre Tahmini	67
6.1.3. Genetik Algoritma İle Parametre Tahmini.....	67
6.2. Rastgele Üretilen Veriler İle Uygulama.....	70
6.2.1. LR Analizi ve NR Yöntemi İle Parametre Tahmini	71
6.2.2. LR Analizi ve NM Yöntemi İle Parametre Tahmini.....	71
6.2.3. Genetik Algoritma İle Parametre Tahmini.....	72
7. SONUÇ VE YORUMLAR	78
KAYNAKLAR	80
EKLER.....	84
Ek A. İki Boyutlu Örnek Bir Problem için Nelder - Mead Matlab Çıktıları	84
Ek B. Olabilirlik Fonksiyonunun Matlab Kodu	86
Ek C. İkinci Uygulamada Kullanılan Rastgele Üretilen Veriler	91
Ek D. Yapay Veriler İçin Olabilirlik 2 Fonksiyonunun Matlab Kodu	93
Ek E. Matlab İle GA Kodları ve Açıklamaları	95
Ek F. Matlab Yazılımı İle Nelder-Mead Kodları ve Açıklamaları.....	101
ÖZGEÇMİŞ.....	104

ÇİZELGELER DİZİNİ

Çizelge 3.1. GDM’de yer alan modeller ve kullanılan bağı fonksiyonları	16
Çizelge 3.2. Kategorik veri analizleri	18
Çizelge 3.3. Örnek veriler	25
Çizelge 3.4. LR’de bazı olasılık değerlerine karşılık gelen odds ve logit değerleri ..	26
Çizelge 4.1. Nelder-Mead algoritması için mantıksal kararlar	36
Çizelge 4.2. Üçgenlerin fonksiyon değerleri	38
Çizelge 5.1. Sayıların kromozomlarla ifade edilmesi	50
Çizelge 5.2. Uygunluk değerlerinin hesaplanması.....	62
Çizelge 5.3. Seçim değerlerinin hesaplanması.....	62
Çizelge 5.4. Çaprazlama değerlerinin hesaplanması.....	63
Çizelge 5.5. Mutasyon değerlerinin hesaplanması.....	63
Çizelge 5.6. Yeni iterasyon değerlerinin hesaplanması	64
Çizelge 6.1. Analizde kullanılan değişkenler	66
Çizelge 6.2. LR modelinde Newton Raphson algoritması ile elde edilen sonuçlar	67
Çizelge 6.3. Parametre tahminlerinin karşılaştırılması	69
Çizelge 6.4. Lojistik regresyon analizi sonuçları	71
Çizelge 6.5. Nelder-Mead algoritması sonuçları	72
Çizelge 6.6. GA için popülasyon boyutuna göre karşılaştırmalar	73
Çizelge 6.7. GA için durdurma kriterlerine göre karşılaştırmalar	73
Çizelge 6.8. GA için Seçim yöntemine göre karşılaştırmalar	74
Çizelge 6.9. Kodlamada kullanılan parametreler	75
Çizelge 6.10. GA’da iterasyon sayıları ve ulaşılan optimum değerler	76
Çizelge 6.11. En iyi parametre tahminlerinin karşılaştırılması	76

ŞEKİLLER DİZİNİ

Şekil 2.1. Doğrusal olmayan EKK'nin gözlem uzayı	8
Şekil 2.2. Doğrusal olmayan modellere örnekler	9
Şekil 3.1. Lojistik fonksiyon grafiği	23
Şekil 3.2. DOM ve LR modeli	27
Şekil 4.1. M orta noktası ve yansıma işlemini kullanarak R noktasının bulunması .	33
Şekil 4.2. Genişleme işlemi ile BGE üçgeninin ve E noktasının elde edilmesi	34
Şekil 4.3. C1 ve C2 daralma noktaları	34
Şekil 4.4. B noktasına doğru daraltma işlemi	35
Şekil 4.5. f fonksiyonunun üç boyutlu grafiği	37
Şekil 4.6. Optimizasyonu (3,2) noktasına yaklaştıran {Tk} üçgenleri dizisi	39
Şekil 4.7. Nelder-Mead algoritması iterasyonları	41
Şekil 5.1. Genetik algoritmanın işleyiş sürecinin akış diyagramı	48
Şekil 5.2. Turnuva seçimi	55
Şekil 5.3. Tek nokta çaprazlama örneği	57
Şekil 5.4. Çift nokta çaprazlama örneği	57
Şekil 5.5. Mutasyon örneği	58
Şekil 6.1. Fitness value (20000 iterasyon)	75

SEMBOLLER VE KISALTMALAR DİZİNİ

X	Bağımsız Değişken
Y	Bağımlı Değişken
n	Gözlem sayısı
β_0	Sabit
β_1	Regresyon Doğrusunun Eğimi
β'	Parametrelerin Vektörü
ε	Rasgele Hata Terimi
ε_i	i. Gözlem İçin Hata Terimi
μ	Ortalama
σ	Standart Sapma
$\pi(x)$	Olasılık Fonksiyonu
Σ	Toplam
∂	Kısmi türev operatörü
P_c	Çaprazlama Olasılığı
P_m	Mutasyon Olasılığı
EÇOTE	En Çok Olabilirlik Tahmin Edicisi
EKK	En Küçük Kareler
GA	Genetik Algoritma
LR	Lojistik Regresyon
NM	Nelder-Mead
NR	Newton-Raphson
GN	Gauss-Newton
HKT	Hata Kareler Toplamı
DR	Doğrusal Regresyon
DOM	Doğrusal Olasılık Modelleri
GDM	Genelleştirilmiş Doğrusal Modeller
OR	Odds Oranı

1.GİRİŞ

Bir bağımlı ve bir veya daha fazla bağımsız değişken arasındaki ilişkinin araştırılması için yaygın olarak kullanılan doğrusal regresyon analizi, bağımlı değişkenin iki ya da ikiden daha fazla düzey içeren kategorik bir değişken olması durumunda, hata terimlerinin normal dağılması gibi bazı model varsayımlarının ihlal edilmesinden dolayı kullanılamamaktadır.

Doğrusal regresyonda bilinmeyen parametreleri tahmin etmek için en sık kullanılan yöntem En Küçük Kareler (EKK) yöntemidir. Bu yöntemde parametreler, bağımlı değişken Y'nin gözlenen değerlerinin, tahmin edilen değerlerinden sapmalarının kareler toplamını minimum yapacak şekilde seçilir. Alternatif olarak hata teriminin dağılımının özel olarak Normal olması durumunda bağımlı değişken Y'nin gözlemlenme olasılığını maksimum yapan En Çok Olabilirlik Tahmin Edicisi (EÇOTE) de yaygın olarak kullanılmaktadır.

Doğrusal olmayan regresyon modellerinde ise parametreler doğrusal olmayan yapıdadır ve EKK ya da EÇOTE ile tahmin edilemezler. Bu nedenle doğrusal olmayan regresyon modelinde etkin parametre tahminlerine ulaşmak için doğrusal olmayan EKK yöntemi ya da hata dağılımı ile ilgili bir bilginin olması durumunda EÇOTE kullanılmaktadır. Yöntemlerin uygulanabilir olması, amaç fonksiyonunun türevlenebilir özellikte olmasına dayalıdır ve iteratif olarak sonuçlara ulaşılır. İteratif olarak optimum değerlere ulaşılan bu süreçte en fazla bilinen algoritmalar, Gauss-Newton (GN), Newton-Raphson (NR), Direkt Arama Yöntemi, Hooke-Jeeves Yöntemi, Nelder-Mead Yöntemi, Gradyant Yöntemi, En Dik İne Yöntemi, Marquardt Yöntemidir.

Bu algoritmalar genellikle özel problemler için uygundur ve süreklilik, türevlerin varlığı, çözüm için bir başlangıç noktasının belirlenmesi gibi kısıtlayıcı varsayımlar

gerektirir. Örneğin, yaygın olarak kullanılan GN algoritmasında başlangıç noktası iyi seçilmez ise çözüm yerel optimumlara takılabilir ve sadece bölgesel optimumlar bulunabilir. Bu algoritmaların seçilecek başlangıç noktasına bağlı olarak çözüme ulaşamama riski de vardır. Bu nedenle klasik optimizasyon tekniklerinin gerektirdiği bu olumsuzlukları ortadan kaldıran ve başlangıç noktasının seçiminden çok fazla etkilenmeyen bazı sezgisel yöntemler önerilmiştir.

Geleneksel optimizasyon yaklaşımından farklı olarak Genetik Algoritma (GA) Yaklaşımı, optimize edilecek fonksiyonun türevlenebilir olması gibi kısıtlayıcı varsayımları gerektirmediği için optimizasyon problemlerine de başarıyla uygulanabilmektedir. Optimize edilecek amaç fonksiyonunun süreksiz olması halinde, süreksizlik noktalarında fonksiyonun türevi alınamayacağından, türev almaya dayalı optimizasyon yöntemleri kullanılamamaktadır. Oysa ki, GA'lar, problemlerin çözümü için türev veya diğer yardımcı bilgilere gereksinim duymadığından, yerel optimum noktalara takılmadan global optimum noktaları bulabilmesi gibi üstün özelliklerinden dolayı bu tip problemlerin çözümünde geleneksel yöntemlere göre önemli bir avantaj sağlamaktadır.

Yapılan çalışmalar incelendiğinde, bağımlı değişkenin sürekli olduğu doğrusal olmayan regresyon modellerinde genellikle doğrusal olmayan EKK ya da EÇOTE yöntemleri ve sürecin ikinci aşamasında klasik optimizasyon tekniklerinin kullanıldığı görülmektedir. Bağımlı değişkenin sürekli olmasına rağmen yukarıda bahsedilen diğer olumsuzluklar dikkate alındığında GA yaklaşımının bu koşullar altındaki başarısını gösteren birçok çalışma mevcuttur. Ancak uygulamada bağımlı değişkenin her zaman sürekli bir değişken olması beklenemez. Özellikle anket çalışmalarında bağımlı değişkenin iki ya da daha fazla düzeyli kategorik bir değişken olduğu durum ile de sıklıkla karşılaşılmaktadır.

Bağımlı değişkenin 0/1 gibi iki düzeyli (binary) ya da ikiden çok düzey içeren kategorik bir değişken olması durumunda oluşturulacak regresyon modelleri parametrelerde doğrusal olmadıklarından dolayı başvurulacak geleneksel en iyi iki yöntem doğrusal olmayan EKK ya da EÇOTE'dir. Bağımlı değişkenin dağılımının Bernoulli dağıldığı bilindiğinden dolayı olabilirlik fonksiyonu yazılabilmekte ve

EÇOTE de kullanılabilir. Bu yöntemlerde model parametreleri analitik olarak elde edilemediğinden, iteratif olarak tahminlere ulaşılabilir. Bilindiği üzere EÇOTE, bağımlı değişkenin gözlemlenme olasılığını maksimum yapmayı amaçlamaktadır. Genel olarak EÇOTE, gözlenen veri kümesini elde etmenin olasılığını maksimum yapan bilinmeyen parametrelerin değerlerini verir. Bu parametrelerin EÇOTE'leri, fonksiyonu maksimum yapan değerleri bulacak şekilde seçilir. Doğrusal olmayan bir amaç fonksiyonu ile çalışıldığından dolayı parametre tahminlerine daha önce de belirtildiği gibi iteratif olarak ulaşılabilir.

Doğrusal regresyonda, karelerin sapmaları toplamının β 'ya göre türevi alınarak elde edilen olasılık eşitlikleri bilinmeyen parametrelerde doğrusaldır ve çözümleri kolaydır. Ancak kategorik bağımlı değişken modellerinde parametrelerine göre doğrusal olmayan bir yapı ile çalışılmakta ve amaç fonksiyonunun birçok yerel maximuma sahip olabilmesinden dolayı bu problemi ortadan kaldıracak alternatif yöntemlerin geliştirilmesine ihtiyaç duyulmuştur. Bu nedenle klasik yöntemlerin sınırlılıklarından dolayı olasılık fonksiyonunu maksimize edecek en iyi parametre kümesini oluşturmak için GA yaklaşımını kullanmanın analitik ve diğer iteratif yöntemlere iyi bir alternatif olabileceği ihtimali üzerinde durulmuş ve çalışma bu doğrultuda devam ettirilmiştir.

Kategorik bağımlı değişken modelleri özellikle sosyal ve sağlık bilimlerinde olmak üzere her alanda araştırmacılar tarafından yaygın olarak kullanılmaktadır. Biyoloji, tıp, ekonometri gibi alanlardaki araştırmalarda kullanılan verilerin özelliklerinden dolayı genellikle analizler lojistik, probit, complementary-log-log gibi doğrusal olmayan kategorik regresyon modelleri ile yapılmaktadır. Ancak bu analizlerin çözümü uzun süren karmaşık matematiksel denklemlere gerek duymaktadır. Sürekli önem kazanan ve karmaşık uygulamalar gerektirmeyen GA ile kategorik verilerin analizinin yapılabilmesi sonucunda, araştırmacılara bilimsel çalışmalarda hem zaman hem de geçerlilik açısından kolaylık sağlanabileceği düşünülmektedir.

Literatürde var olan çalışmaların çoğunda bağımlı değişkenin sürekli olduğu durum için doğrusal olmayan optimizasyon probleminde GA yaklaşımının başarısı tartışılmakta ancak bağımlı değişkenin kategorik olması durumunun çok fazla

incelenmediği görülmektedir. GA ile Lojistik Regresyon (LR) modellerindeki parametre tahminlerinin yapıldığı bir çalışmaya rastlanmaması bu çalışmanın önemini artırmaktadır.

Yukarıda açıklanan tüm olumsuzluklar ve eksiklikler dikkate alındığında tez çalışmamızın temel amacı, kategorik bir bağımlı değişken ile çalışılması durumunda parametre tahmininde kullanılan klasik yöntemlere alternatif yeni yaklaşımlar önermektir. Farklı modellerdeki başarısı kanıtlanan GA ve GA'ya da alternatif olan Nelder-Mead (NM) algoritmalarının teorisini ortaya koymak ve bu iki yaklaşımın kategorik bağımlı değişkenlerdeki etkinliğini karşılaştırmaktır. Bu karşılaştırmanın yapılabilmesi için gerçek ve yapay veriler üzerinden klasik Newton-Raphson (NR) parametre tahminlerine ulaşılmış ve GA ve NM yaklaşımlarının üstünlüğü farklı senaryolar altında araştırılmıştır.

Tez çalışması altı bölümden oluşmaktadır.

Çalışmanın birinci bölümünde, tezin konusu, içeriği ve ulaşılmak istenen hedeften bahsedilmiştir.

İkinci bölümde, doğrusallık ve doğrusal olmama durumları incelenmiş, bazı özel dönüşümlerle doğrusallaştırma işlemleri gösterilmiştir. Doğrusal olmayan regresyonda parametre tahmini detaylı bir şekilde incelenmiş, bu doğrultuda doğrusal olmayan regresyonda EKK tahmini ve normal denklemlerin çözümü verilmiştir. Buradaki denklemlerin çözümü için doğrusal olmayan modellerin çözümünde kullanılan bazı iteratif yöntemlerden bahsedilerek özellikle NR Metodu üzerinde yoğunlaşmıştır.

Üçüncü bölümde, LR analizi hakkında genel bilgiler, logit ve bahis oranı kavramları, modelin parametrelerinin klasik yöntemler ile tahmini, uyum iyiliği ölçüleri konularına yer verilmiştir. Ayrıca LR'nin, doğrusal regresyon ile ilişkisine değinilmiştir.

Dördüncü bölümde, doğrusal olmayan regresyon modellerinde parametre tahminlerine klasik yaklaşımlar ele alınmıştır. İterasyona dayalı yöntemlerden en çok kullanım alanı bulmuş olan klasik NR algoritması ve arama yöntemleri kategorisindeki NM

algoritmaları ile ilgili detaylı bilgiler verilmiştir. Matlab kodları da tanıtılarak yöntemlerin pratikte uygulanabilirliği kolaylaştırılmıştır.

Beşinci bölüm, klasik yaklaşımlara alternatif olan sezgisel optimizasyon tekniklerinden GA ile ilgili genel bilgileri, algoritmanın işleyişi ile ilgili temel bilgileri, başlangıç popülasyonu kavramını, amaç fonksiyonu tanımını, seçim yöntemlerini, kullandığı operatörleri ve işleyiş sürecini ifade eden şema teorisini içermektedir. Ayrıca GA'ların direk arama yöntemlerinden farkı açıklanarak, uygulama alanları ve özellikle parametre tahmininde kullanımına değinilmiştir.

Altıncı bölüm, teorilerin gerçek ve yapay veriler üzerinden uygulamaya döküldüğü bölümdür. Bu amaçla ilk olarak bağımlı değişkenin iki düzeyli olduğu gerçek bir veri kümesi üzerinden parametreler geleneksel NR algoritması, NM algoritması ve alana uygunluğu araştırılan GA algoritmaları ile tahmin edilmiştir. Sonuçlar, iterasyon sayısı, uygulama kolaylığı, başlangıç noktası problemleri gibi birçok kriter dikkate alınarak karşılaştırılmıştır.

Uygulama bölümünün ikinci aşamasında rastgele üretilen veriler kullanılarak her üç algoritmanın başarısı araştırılmıştır. Parametre tahminleri ve olabilirlik fonksiyon değerleri elde edilerek sonuçlar karşılaştırılmıştır.

Çalışmanın son bölümünde ise tahmin yöntemlerinin performansı ile ilgili genel bir değerlendirme yapılmış ve önerilen iki algoritmanın klasik yöntemlere göre başarısı ortaya konmuştur. Ek olarak, bundan sonra yapılması düşünülen çalışmalar ile ilgili bilgi verilmiştir.

2. DOĐRUSAL OLMAYAN MODELLERDE PARAMETRE TAHMİNİ

2.1. Parametrelerde Doğrusallık ve Doğrusal Olmama Durumu

Regresyon analizi, bir deđişkenin bir veya daha fazla deđişken ile arasındaki ilişkinin matematiksel bir fonksiyonla ifade edilmesidir. Matematiksel fonksiyonun yapısına göre doğrusal ve doğrusal olmayan regresyon modelleri olarak ikiye ayrılırlar (Orhunbilge, 2002). Doğrusal olmayan modeller tıp, biyoloji, kimya, ekonomi, tarım ve hayvancılık gibi birçok alanda yaygın olarak kullanılmaktadır (Kutner vd., 2004).

Regresyon analizinde model her zaman doğrusal olmayabilir. Araştırmacılar bazen deđişkenler arasındaki ilişkiyi incelemeyen doğrusal ilişki varmış gibi kabul ederek doğrusal regresyon modeli kurma eğilimindedir. Ancak deđişkenler arasında doğrusal ilişki zayıf ya da yok ise, kurulan doğrusal model ve yapılan testler gerçekçi olmaz. Araştırmacı öncelikle serpilme diyagramını oluşturarak noktaların doğrusal veya eğrisel bir ilişki içerisinde olup olmadığına bakmalıdır. Serpilme diyagramı verilerin bir doğru etrafında dizilip dizilmediğine ilişkin ön bilgi verir. İlişkinin doğrusal olup olmadığı ile ilgili kesin bilgi ise yapılan istatistik testler ile ortaya çıkarılır.

Bazı modellerin matematiksel yapıları eğrisel olmasına rağmen, özel matematiksel dönüşümlerle doğrusallaştırılabilir. Buradaki amaç, doğrusal modellerin tahmin ve testlerinin nispeten daha kolay olmasından kaynaklanmaktadır. Doğrusal hale dönüştürülebilen modeller ile ilgili testler ve analizler doğrusal regresyon analizinde olduğu gibi yapılabilir.

Deđişken ve parametre bakımından doğrusal bir regresyon modeli,

$$Y = \beta_0 + \beta_1 X + \epsilon_i \quad (2.1)$$

biçiminde ifade edilebilir. Parametreler bakımından doğrusal, değişkenler bakımından doğrusal olmayan regresyon modeline örnek,

$$Y = \beta_0 + \beta_1 X^2 + \beta_2 \sqrt{X} + \beta_3 \log X_3 + \varepsilon_i \quad (2.2)$$

olarak gösterilebilir. Sadece parametreler açısından doğrusal olan bir regresyon modeline örnek ise aşağıda verilmiştir:

$$\ln Y = \beta_0 + \beta_1 \ln X + \varepsilon_i \quad (2.3)$$

Parametrelere göre doğrusallık, modeldeki tüm parametrelerin birinci dereceden olmasıdır yani üstel durumda veya başka bir parametre ile çarpım ya da bölüm halinde olan bir parametrenin bulunmamasıdır. Eşitlik (2.1)'deki model, değişken ve parametre bakımından doğrusal iken Eşitlik (2.2)'deki model parametre bakımından doğrusal, değişken bakımından doğrusal değildir. Eşitlik (2.3)'deki model ise sadece parametreler açısından doğrusaldır. Çalışmada doğrusallık ile ifade edilmeye çalışılan durum parametrelerin doğrusal olması durumudur.

Doğrusal olmayan regresyon modelleri genellikle,

$$Y_i = f(X_i, \gamma) + \varepsilon_i, \quad i = 1, 2, \dots, n \quad (2.4)$$

biçiminde ifade edilir.

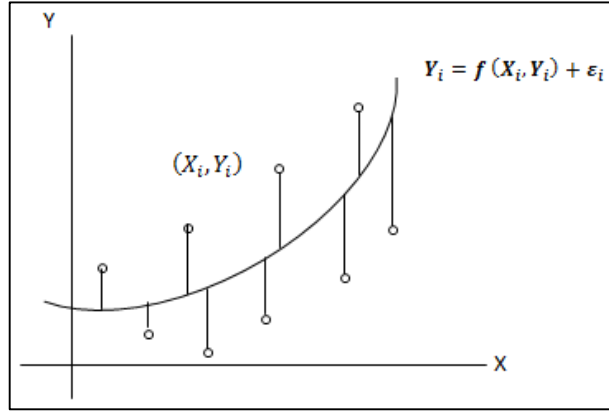
Eşitlik (2.4)'de n , gözlem sayısını, Y , bağımlı değişken vektörünü, X , bağımsız değişkenler vektörünü, γ , doğrusal olmayan parametreleri içeren bilinmeyen parametreler vektörünü, $f(\cdot)$, açıklayıcı değişkenleri bağımlı değişkene doğrusal bir yapı ile bağlayan bilinen bir fonksiyonu, ε_i ise gözlenemeyen, sıfır ortalamalı ve σ^2 varyanslı Normal dağılım göstermesi beklenen, modelin açıklanamayan kısmını ifade eden rasgele değişkenler vektörünü ifade etmektedir.

Doğrusal olmayan regresyon modellerinde, parametre sayısı modeldeki açıklayıcı değişken sayısı ile doğrudan ilişkili değildir. Ayrıca parametre tahminleri de doğrusal bir modeldeki parametreler kadar kolay elde edilememektedir. Parametre tahminlerine

analitik olarak ulaşılamamasından dolayı sayısal yöntemlere ihtiyaç duyulmaktadır.

Buradan çıkan sonuç, yapılan dönüşümler ile doğrusallığın sağlanamaması durumunda doğrusal olmayan modellerde parametre tahminlerinin elde edilmesi için kullanılacak en uygun yöntemin belirlenmesi son derece önem arz etmektedir.

Aşağıda doğrusal olmayan modellerin parametrelerinin tahmininde kullanılan doğrusal olmayan EKK yönteminin gözlem uzayı görülmektedir.



Şekil 2.1. Doğrusal olmayan EKK'nin gözlem uzayı

$f(\underline{x}, \underline{\theta})$, $\underline{\theta} = (\theta_1, \theta_2, \dots, \theta_p)'$ parametre vektörünün bileşenlerine göre türevlenebilir fonksiyon olmak üzere, $\frac{\partial f(\underline{x}, \underline{\theta})}{\partial \theta_i}$ ifadesi en az bir θ_i 'ye ($i = 1, 2, \dots, p$) bağlı ise, $f(\underline{x}, \underline{\theta})$, $\underline{\theta}$ parametre vektörüne göre doğrusal olmayan fonksiyondur denir (Bates ve Watts, 1988).

Bu tanımları örnek ile açıklamak gerekirse, Eşitlik (2.5) ile verilen üstel bir yapıya sahip olan Y fonksiyonunun β_1 'e göre türevi alınırsa,

$$Y = \beta_1 e^{\beta_2 x} \quad (2.5)$$

$$Y' = \frac{\partial f(x, \beta)}{\partial \beta_1} = e^{\beta_2 x} \quad (2.6)$$

ifadelerine ulaşılır. Y'nin β_2 'ye göre türevi alındığında ise,

$$Y' = \frac{\partial f(x, \beta)}{\partial \beta_2} = \beta_1 e^{\beta_2 x} \quad (2.7)$$

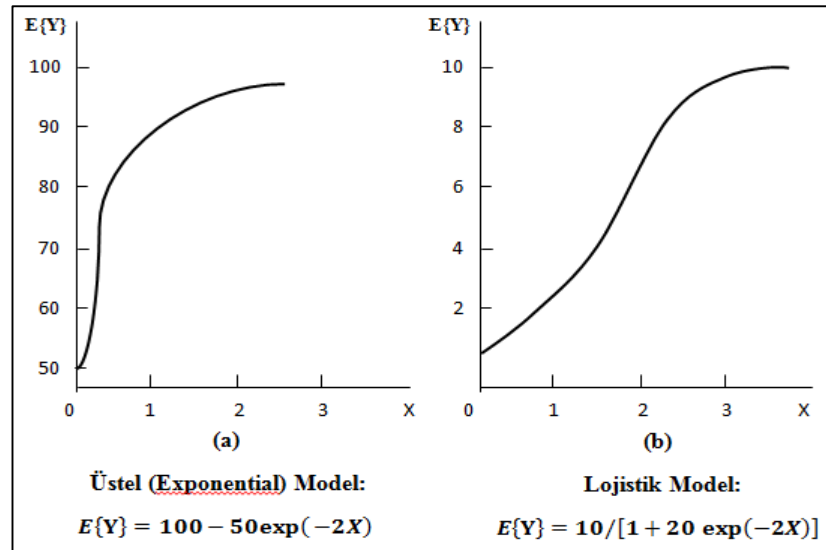
sonucu elde edilir. Fonksiyonların kısmi türevleri bilinmeyen parametreler β_1 ve β_2 'nin bir fonksiyonu olduğundan model doğrusal değildir.

Doğrusal olmayan regresyon modellerini doğrusal regresyon modellerinden ayıran diğer önemli fark ise modeldeki parametrelerin sayısının X bağımsız değişkenlerinin sayısı ile doğru orantılı olmamasıdır. Doğrusal regresyon modellerinde n-1 adet X bağımsız değişkenin varlığında n adet parametrenin modelde bulunması gereklidir. Örneğin parametrelerine göre doğrusal olan ikinci dereceden (Kuadratik) model incelendiğinde,

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 \quad (2.8)$$

fonksiyonunda iki bağımsız değişken yer almakta iken üç adet parametrenin var olduğu görülmektedir. Eşitlik (2.5) ile verilen doğrusal olmayan üstel regresyon modelinde ise bir adet bağımsız değişken olmasına karşın birden fazla parametrenin yer aldığı görülür. Özel büyüme eğrileri ve lojistik modelleri için de benzer durum geçerlidir.

Bazı yaygın olarak kullanılan doğrusal olmayan modellere ilişkin iki örnek aşağıda verilen grafikteki gibidir: (Kutner vd., 2004).



Şekil 2.2. Doğrusal olmayan modellere örnekler

Çalışmamızın ana temasını oluşturan bağımlı değişkenin kategorik olması durumu, dönüşümler yapılarak doğrusallaştırılamayan doğrusal olmayan regresyon modelleri kategorisindedir.

2.2. Doğrusal Olmayan Regresyon Analizi

Özellikle son yıllarda doğrusal olmayan regresyon analizi, pratik uygulamalar için alternatif bir araç haline gelmiştir. Çünkü fizik, kimya, biyoloji ve mühendislik gibi pek çok alanda elde edilen modellerin hemen hemen hepsi doğrusal değildir.

Doğrusal olmayan modeller üzerindeki çalışmalar, 1960'larda Marquardt (1963), Hartley ve Booker (1963)'ın EKK yöntemi ile doğrusal olmayan modellerde parametre tahmini için önerdikleri algoritmalar öncülüğünde gerçekleşmiştir.

Gallant (1975) ile Seber ve Wild (1989) doğrusal olmayan modellerde parametre tahmini ve hipotez testi üzerinde çalışmıştır.

Gallant(1975), Bates ve Watts(1988), Seber ve Wild (1989) çok değişkenli doğrusal olmayan modellerde parametre tahmini ve sonuç çıkarımı üzerindeki çalışmalar yapmıştır (Genç 1997).

Doğrusal olmayan modeller konusunda yapılan çalışmalarda, modellerin verilere uydurulmasının zor ve uğraştırıcı olduğu görülmüştür. Bu modellerde tahmin için kullanılan algoritmalar yakınsama konusunda birçok durumda yetersiz kalmaktadır (Gallant, 1987). Bu modellerin tahmininde kullanılan yaklaşımlardan bir tanesi modelin doğrusallaştırılmasıdır. Bu tür modeller aslen doğrusal olarak anılır (Bates ve Watts, 1988). Bu durumda eğer tüm model varsayımları sağlanıyorsa klasik doğrusal EKK ya da EÇOTE yöntemleri ile bu modellerin parametreleri tahmin edilebilir (Miller, 1990). Ancak varsayımlar sağlanmıyorsa bulunan tahminler yanlı olacaktır. Ayrıca, bu yönde yapılan doğrusallaştırma girişimleri ile elde edilen yapıların uygunluğunda bazı belirsizlikler bulunabilmektedir. Bununla birlikte parametre tahminlerinin de bazen istenmeyen özellikleri olabilmektedir (Gallant, 1987).

Doğrusal olmayan regresyon modellerinde parametre tahminleri doğrusal regresyon modellerine benzer şekilde EKK ya da EÇOTE yöntemleri kullanılarak bulunur. Doğrusal olmayan regresyondaki hata terimleri bağımsız, normal ve sabit varyanslı olma özelliklerini taşıdığı zaman iki yöntemle aranan parametre tahminleri birbirlerine eşit olarak bulunacaktır. Ancak doğrusal olmayan regresyonda bu yöntemlerle parametre tahminlerini bulmak ve modelden sonuç çıkarmak doğrusal regresyon modellerindeki kadar kolay değildir. Bu modellerin analizi genellikle bazı ileri düzeyde bilgisayar algoritmalarına dayalı olarak bulunabilmektedir. Bilgisayar teknolojisinin gelişmesi ile denklemlerin daha kolay çözümüne ulaşılmış ve doğrusal olmayan regresyon modelleri araştırmacılar tarafından daha fazla kullanılabilir hale gelmiştir.

Doğrusal olmayan modellerin parametre tahminlerine iteratif yöntemler ile ulaşılabilir (Draper ve Smith, 1981). Eğer bir model, yeniden-parametrelendirme sonucunda doğrusallaştırılmıyor ise, parametre tahminleri yansızlık, normallik, ve minimum varyans gibi istenen özelliklere sahip olamayacaktır. Bu sebeple de Marquardt yöntemi gibi karmaşık tahmin yöntemlerine gereksinim duyulacaktır (Ratkowsky, 1983).

Doğrusal olmayan model tahminlerinde GN ve NR algoritmaları yaygın olarak kullanılmasına rağmen bu algoritmalarda başlangıç noktasının seçimi oldukça önemlidir. Uygun olmayan başlangıç noktaları uygun olmayan çözümler elde edilmesine neden olabileceği gibi çözüm zamanı bakımından da ekonomik olmayabilir. Bu nedenle başlangıç noktasının seçiminden çok fazla etkilenmeyen bazı sezgisel yöntemler önerilmiştir. Çok değişkenli bir fonksiyonun yerel minimumunu bulduran bir algoritma Nelder ve Mead (1964) tarafından geliştirilmiştir. Daha sonraları Nelder-Mead (NM) algoritmasına dayalı farklı yöntemler önerilmiştir. Bu yöntemlerden biri, Tvrdik ve arkadaşları (2006) tarafından önerilen stokastik bir algoritmadır (Serin, 2010).

Daha önce de bahsedildiği gibi, doğrusal olmayan modellerin parametreleri doğrusal olmayan EKK ve EÇOTE yöntemleri ile tahmin edilebilmektedir. Bu algoritmaların kullanılabilmesi bağımsız değişken(ler)in en az iki kez türevlenebilmesi şartı

gerekmektedir. Ayrıca bu algoritmaların seçilecek başlangıç noktasına göre çözüme ulaşamama riski de vardır (Karakoca, 2009).

2.2.1. Doğrusal olmayan en küçük kareler ve en çok olabilirlik tahmin edicileri

Doğrusal ya da doğrusal olmayan regresyon analizinde model parametrelerine klasik EKK yaklaşımı, aşağıda verilen Hata Kareler Toplamı (HKT)'nin minimizasyonuna dayanır.

$$\text{HKT} = \sum_{i=1}^n [Y_i - \hat{Y}_i]^2 \quad (2.9)$$

Burada, Y_i ($i = 1 \dots, n$) bağımlı değişken değerini, \hat{Y}_i ise tahmin değerini göstermektedir. \hat{Y}_i nin özel olarak aşağıdaki biçimde doğrusal bir model olarak tanımlanması durumunda k tane açıklayıcı değişken içeren doğrusal regresyon modelinin HKT'sine ulaşılır.

$$\text{HKT} = \sum_{i=1}^n [Y_i - (\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \dots + \beta_k X_{ki})]^2 \quad (2.10)$$

Eşitlik (2.10) ile verilen HKT, optimize edilecek fonksiyon olması nedeniyle optimizasyon kuramında amaç fonksiyonu olarak tanımlanır. Amaç fonksiyonunun parametrelere göre ayrı ayrı 1. dereceden türevleri alınıp "0" a eşitlendiğinde ise açıklayıcı değişken sayısı kadar (k tane) "Normal Denklemler" elde edilir. Normal denklemlerin ortak çözümü ile de model parametrelerinin tahminlerine ulaşılır. Doğrusal bir model ile çalışıldığından dolayı normal denklemler eşanlı olarak çözülüp analitik olarak parametre tahminleri elde edilebilir. Aynı tahminlere, hata dağılımı ile ilgili olarak Normal dağılım varsayımı yapılması durumunda da EÇOTE ile ulaşılabilir.

Görüldüğü gibi bağımlı değişkenin sürekli olduğu doğrusal bir model ile çalışıldığında optimizasyon sürecinde herhangi bir iterasyona gereksinim duyulmamaktadır. Tek koşul, amaç fonksiyonunun türevlenebilir olmasıdır. Klasik doğrusal regresyon

analizinde zaten bağımlı değişkenin sürekli olması koşulu olduğundan alternatif optimizasyon yöntemlerine ihtiyaç duyulmamaktadır.

Bağımlı değişkenin sürekli, çalışılan modelin ise doğrusal olmayan bir yapıda olması durumunda ise, türevlenebilirlik özellikleri sağlandığından amaç fonksiyonu olan HKT'nin yine parametrelere göre 1. dereceden kısmi türevleri alınıp normal denklemlere ulaşılır.

Örneğin,

$$Y_i = \gamma_0 \exp(\gamma_1 X_i) + \varepsilon_1 \quad (2.11)$$

doğrusal olmayan üstel regresyon modelindeki parametreler, EKK yöntemi ile aşağıda verilen HKT'nin minimizasyonu ile elde edilebilir:

$$HKT = \sum_{i=1}^n [Y_i - \gamma_0 \exp(\gamma_1 X_i)]^2 \quad (2.12)$$

$\gamma_0, \gamma_1, \dots, \gamma_{k-1}$ parametrelerine göre HKT'nin γ_k ' ya göre birinci dereceden kısmi türevi alınır;

$$\frac{\partial HKT}{\partial \gamma_p} = \sum_{i=1}^n -2[Y_i - f(X_i, \gamma)] \left[\frac{\partial f(X_i, \gamma)}{\partial \gamma_p} \right] \quad p = 1, 2, \dots, k \quad (2.13)$$

ifadesine ulaşılır. Elde edilen kısmi türevler sıfıra eşitlenerek değişken sayısı (k) kadar normal denklem elde edilir.

Normal denklemler bu durumda parametrelerde doğrusal değildir ve analitik olarak çözüme gidilemez. Bu nedenle parametrelere iteratif olarak ulaşılmaya çalışılır. İteratif yöntemlerden GN ya da NR algoritmaları yaygın olarak kullanılmaktadır. Bu yöntemlerde ise başlangıç noktası problemi ile karşılaşılabilir. Birçok doğrusal olmayan regresyon problemlerinde, normal denklemler elde edildikten sonra sayısal yöntemler kullanılarak bu denklemleri iteratif olarak çözmektense, direk sayısal arama prosedürü ile EKK tahmin edicilerini bulmak daha pratiktir (Kutner vd., 2004; Draper

ve Smith, 1981; Ratkowsky, 1983). Çözüm yollarından bir tanesi sezgisel optimizasyon yöntemlerinden GA yaklaşımının kullanılmasıdır. Literatürde bu yöntemin başarısı ile ilgili yapılmış çok sayıda çalışmaya rastlanmaktadır.

Aynı parametrelere modeldeki hata terimi ϵ 'nin, dolayısıyla bağımlı değişkenin Normal dağılım gösterdiği, sabit varyansa sahip bağımsız bir rasgele değişken olduğu varsayımı altında EÇOTE ile de ulaşılabilmektedir. Logaritmik olabilirlik fonksiyonu $\log L(\cdot)$, aşağıda tanımlanmıştır.

$$\log L(Y, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp \left[-\frac{1}{2\sigma^2} \sum_{i=1}^n [Y_i - \gamma_0 \exp(\gamma_1 X_i)]^2 \right] \quad (2.14)$$

EÇOTE'deki amaç ise amaç fonksiyonu $\log L(Y, \sigma^2)$ 'yi maksimum yapan en etkin parametre tahminlerinin bulunması mantığına dayanır. Olabilirlik eşitliklerinin de parametrelerde doğrusal olmaması dikkate alındığında yine sayısal yöntemler ve alternatif sezgisel optimizasyon yöntemleri ile etkin parametre tahminlerine ulaşılmaya çalışılmaktadır (Kutner vd., 2004).

Çalışmamızda, bağımlı değişkenin iki düzeyli kategorik bir değişken olması durumunda doğrusal olmayan regresyon modelleri kapsamındaki Lojistik Regresyon Model parametrelerinin klasik ve sezgisel yaklaşımlar ile tahmini detaylı bir biçimde açıklanacak, Matlab programında geliştirilen kodlara yer verilecek ve yöntemlerin başarısı yapay ve gerçek veriler üzerinden karşılaştırılacaktır.

Devam eden bölümde, kategorik bağımlı değişken modelleri ailesinden Lojistik Regresyon (LR) Modeli, model parametrelerinin geleneksel yöntemler ile tahminleri detaylı bir biçimde incelenmiştir.

3. LOJİSTİK REGRESYON ANALİZİ

LR modeli bağımlı değişkenin kategorik olması durumunda kullanılan bir regresyon modelidir. Modelde açıklayıcı değişken değerlerine bağlı olarak herhangi bir gözlemin bağımlı değişkenin kategorilerine düşmesi olasılıkları hesaplanır. Klasik Doğrusal Regresyon (DR) modelinde ise bağımlı değişken süreklidir ve açıklayıcı değişken değerlerine bağlı olarak koşullu ortalama modellenir. Bağımlı değişkenin kategorik olması durumunda da akla gelen ilk yöntem DR modelinin sürece uygulanmasıdır. Bu modeller Doğrusal Olasılık Modelleri (DOM) olarak adlandırılmakta ancak modelin gerektirdiği birçok varsayım bozulduğundan dolayı uygulamada etkin olarak kullanılamamaktadır. DR varsayımları üzerine kurulu olmayan alternatif model arayışlarına gidilmiş ve Genelleştirilmiş Doğrusal Modeller (GDM) ailesine üye birçok model ortaya atılmıştır.

GDM klasik doğrusal modellerin bir genişlemesidir. DR modelindeki en önemli varsayım olan açıklayıcı ve bağımlı değişkenler arasındaki doğrusallık kısıtını ortadan kaldırmakta ve açıklayıcı değişkenlerin bir fonksiyonunun bağımlı değişken ile doğrusal olması koşulunu getirmektedir. Bunun nedeni ise bağımlı değişkenin kategorik olması durumunda açıklayıcı değişkenler ile doğrusal bir ilişki içerisinde olması beklenemez.

Uçar (2004) tarafından yapılan çalışmada, GDM model varsayımları, varsayım bozulmaları ve önerilen alternatif yöntemler detaylı olarak tanıtılmıştır.

GDM üç bileşenden oluşmaktadır.

Rasgele Bileşen: GDM'nin rasgele bileşeni, bağımlı değişken Y 'yi tanımlar ve Y için bir olasılık dağılımı belirlemektedir (Agresti, 1996).

Sistemantik Bileşen: GDM'nin sistemantik bileşeni, doğrusal bir fonksiyon ile ifade edilen modeldeki açıklayıcı değişkenleri tanımlamaktadır:

$$\eta = \sum_{i=1}^K x_i \beta_i \quad (3.1)$$

Bağ (link): GDM'nin bağ fonksiyonu ise bağımlı değişken ortalamasının, açıklayıcı değişkenlere nasıl bağlanacağını belirler (Agresti, 1996).

GDM, $\mu_i = E(Y_i)$ iken, $g(\mu_i) = \eta_i$ biçimde gösterilmektedir.

Buradaki $g(\cdot)$ bağ fonksiyonu herhangi bir monoton, diferansiyellenebilir fonksiyon olabilmektedir (McCullagh ve Nelder, 1989). Bağ fonksiyonu değiştikçe modellerin isimleri de değişmektedir. Bu nedenle bu modellerin oluşturduğu aile GDM ailesi olarak adlandırılmaktadır.

GDM ailesine dâhil olan modeller ve kullanılan bağ fonksiyonları aşağıdaki çizelgede verilmektedir.

Çizelge 3.1. GDM'de yer alan modeller ve kullanılan bağ fonksiyonları

Rasgele Bileşen	Bağ	Bağ Adı	Sistemantik Bileşen	Model
Normal	$\eta = \mu,$ $E(Y) = \mu$ iken	Özdeş	Sürekli	Doğrusal Regresyon
Binom	$\eta = \ln \mu / (1 - \mu),$ $E(Y) = \mu$ iken	Lojit	Karışık	Lojistik Regresyon
Binom	$\eta = \Phi^{-1}(\mu),$ $E(Y) = \mu$ iken	Probit	Karışık	Probit Regresyon
Poisson	$\eta = \ln(\mu),$ $E(Y) = \mu$ iken	Logaritma	Karışık	Poisson Regresyon
Çok terimli	$\eta = \ln(\mu_i / \mu_j)$ $E(Y) = \mu$ iken	Genelleştirilmiş Lojit	Karışık	Çok Terimli Cevap

Tez çalışmasında, bağımlı değişkenin iki düzeyli olduğu, rasgele bileşenin Binom dağılımı gösterdiği ve bağ fonksiyonunun lojit olarak belirlendiği LR modeli için parametre tahminleri üzerine yoğunlaşmaktadır.

LR analizinin kullanım amacı istatistikte kullanılan diğer model yapılandırma teknikleriyle aynıdır. Yani, en az değişken kullanarak en iyi uyuma sahip olacak şekilde bağımlı değişken ile bağımsız değişkenler kümesi (açıklayıcı değişkenler) arasındaki ilişkiyi tanımladığı kabul edilebilir bir model kurmaktır (Hosmer ve Lemeshow, 2000).

Kullanım kolaylığının yanında rahat yorumlanabilmesiyle ön plana çıkmış olan LR modelleri özellikle sosyal bilimler, biyoloji, tıp, ekonomi, tarım, veterinerlik gibi alanların birçok araştırma ve uygulamalarında yaygın olarak kullanılmaya başlanmıştır. Kategorik verilerin istatistiksel analizi, hemen hemen tüm alanları ilgilendirmektedir. Sağlık alanında yapılan çalışmaların çoğunda incelenen bağımlı değişken iki ya da daha fazla düzeyli kategorik bir değişken yapısındadır. Hastalık şiddetine (az-orta-şiddetli), tümör derecesine (I-II-III) veya bir ilacın tedavide etkili ya da etkisiz olmasına etki eden önemli faktörleri belirlemek amacıyla LR analizine sıklıkla başvurulmaktadır.

LR analizi, bağımsız değişkenlerin etkilerine dayanarak verilerin sınıflandırılmasında kullanılan bir yöntemdir. Bu yöntem, olasılık kurallarına uygun olarak verilerin belirli sınıflara atanmasını sağlar. Matematiksel olarak esnekliği ve kolay yorumlanabilirliği, bu yönteme olan ilgiyi de arttırmaktadır (Tatlıdil, 1996; Özdamar, 2004).

Verilerin iki ya da daha çok gruba ayrılmasında kullanılan yöntemlerden üç tanesi Diskriminanat, Kümeleme ve LR analizidir. Bu üç analiz yöntemin birbirine göre karşılaştırması Çizelge 3.2'deki gibidir (Şahin, 1999).

Veri yapısına uygun olarak LR analizinin uygulanacağına karar verildikten sonra izlenecek adımlar aşağıdaki gibidir:

- 1- Katsayıların EÇOTE ile tahmin edilmesi,
- 2- Katsayıların yorumlanması,
- 3- Katsayılara ait hipotez kontrollerinin yapılması,
- 4- Modelin değerlendirilmesi.

LR modeline dahil edilen bağımsız değişkenlerin tümünün bağımlı değişkeni açıklamak için etkili olması her zaman mümkün olmamaktadır. Bağımlı değişken

üzerinde etkisi olmayan değişkenlerin denklemde tutulması LR denkleminin etkinliğini ve tahmin gücünü düşürmektedir. Bu durumda, bu değişkenlerin ölçümleri ile yapılacak olan gereksiz harcamaları ortadan kaldırmak amacıyla değişken eleme yöntemleri olarak bilinen istatistiksel yöntemler kullanılmaktadır.

Çizelge 3.2. Kategorik veri analizleri

Diskriminant	<ul style="list-style-type: none">-Küme sayısı önceden bilinmekte ve analiz boyunca değişmemektedir.-Gelecekte kullanılabilir fonksiyonlar verir. Bu özelliği ile kümelemeden ayrılır.-Değişkenlerin bazıları kategorik, bazıları sürekli olamaz.-Sık kullanılan, en çok bilinen yöntemdir.-Veriler normal dağılımlı olmalıdır.
Kümeleme	<ul style="list-style-type: none">-Küme sayısı önceden tam bilinmemektedir.-Gelecekte kullanılabilir özelliği yoktur.-Veriler veya uzaklık değerleri normal dağılımlı olmalıdır.-Kovaryans matrisine ilişkin varsayım yoktur.-Çok değişkenli varyans analizi, lojistik regresyon analizi, çok boyutlu ölçekleme gibi çok değişkenli analizlerle yakından ilgilidir.
Lojistik Regresyon	<ul style="list-style-type: none">-Değişkenlerden bazıları sürekli, bazıları kesikli olabilir.-Küme sayısı önceden bilinmektedir.-Varsayım (normallik, ortak varyans) bozulmalarında diskriminant ve çapraz tablolara alternatif oluşturur. Çünkü normallik varsayımı yoktur (y, ikili veya çok sınıflı olduğundan).-Model esnek ve kolay yorumlanabilir.

Değişken seçimi yöntemlerinden yaygın olarak bilinenleri; “ileriye doğru seçim” (forward selection), “geriye doğru eleme” (backward elimination) ve “tüm olası altkümeler regresyon yaklaşımı” (all possible subset) gibi değişik yaklaşımları içeren adımsal regresyon (stepwise), ve en iyi regresyon modeli bulma (best regression) yöntemleridir (Miller 1990; Önder 2001; Özdamar 2004).

LR yönteminin kullanılabilmesi için gereken varsayımlar kısaca aşağıdaki gibi özetlenebilir:

- Gözlemlerin bağımsız olması,
- Az sayıda gözlemlerle çalışılmaması,
- Bağımsız değişkenler arasında çoklu doğrusal bağlantı olmaması,
- Bağımsız değişkenlerle ilgili bağımlı değişken arasında doğrusal olmayan bir ilişki olmasıdır.

3.1. LR ve DR Analizlerinin Karşılaştırılması

Klasik DR modelleri genellikle nicel olarak ifade edilen yani ölçülebilen değişkenler üzerine kurulmuştur. LR analizinde ise bağımlı değişken kategoriktir. Açıklayıcı değişkenlerin veri türü ile ilgili herhangi bir kısıt yoktur.

DR analizinde bağımlı değişken ve bağımsız değişken(ler) nicel olarak belirtilir. Nitelik olarak belirtilemezler. Bağımlı değişken nitelik olarak belirtilirse, bağımsız değişken ya da değişkenlerle arasındaki ilişki LR yöntemiyle aranır.

DR analizinde bağımlı değişken, sadece rakamla ifade edilebilen nicel (kantitatif) değişkenler (gelir, fiyat, sıcaklık) tarafından etkilenmez. Aynı zamanda ırk, cinsiyet, renk, din, uyruk, savaşlar, grevler, siyasi çalkantılar, hükümetin iktisat politikasındaki değişimler gibi nitel (kalitatif) değişkenlerden de etkilenir. Bu durum dikkate alındığında açıklayıcı değişkenler ile ilgili herhangi bir kısıt bu yöntemde de bulunmamaktadır (Frank, 1971).

Ekonometrik ve istatistiksel araştırmalarda kullanılan değişkenler genellikle nicel değişkenlerdir. Ancak araştırmalarda karşılaşılan değişkenler her zaman nicel değişken yapısında olmayabilir. Nicel karakterli olmayan problemlere sayısal olarak değerler verilerek nicel hale dönüştürülebilirler. Örneğin bağımlı veya bağımsız değişken olarak cinsiyet değişkeni ele alındığında yeni bir değişken tanımlanarak erkek için '1', kadın için '0' değeri verilir ve yeni bir değişken oluşturulur. Bu yapay değişkenler modelde bağımsız değişken olarak yer aldığında gölge (kukla) değişken olarak adlandırılmaktadır.

Hangi gruba 0 ya da 1 olarak değer atandığı önemli olmamakla birlikte, bu bilgiye analiz sonucu elde edilen modeldeki katsayılar yorumlanırken ihtiyaç vardır. Örneğin, eğer ilgilenilen değişken cinsiyet ise, herhangi bir gruba 1 (örneğin kadın) ve diğer gruba 0 (erkek) değeri atanabilir. Bu durumda katsayılar, bağımsız değişken ya da değişkenlerin, bir bireyin kadın olması olasılığı üzerindeki etkisini yansıtır. Çünkü "1" olarak kodlanan kategori "kadın"dır (Hair vd., 2006).

LR analizi, temelde regresyon analizi olmakla birlikte doğrusal regresyon analizinden üç önemli farklılığı vardır:

- 1- Doğrusal regresyon analizinde bağımlı değişken sürekli iken LR analizinde kategoriktir,
- 2- Doğrusal regresyon analizinde bağımlı değişkenin değeri, LR'de ise bağımlı değişkenin alabileceği değerlerden birinin gerçekleşme olasılığı tahmin edilir,
- 3- Doğrusal regresyon analizinde bağımsız değişkenlerin çoklu normal dağılım göstermesi koşulu aranırken, LR'nin uygulanabilmesi için bağımsız değişkenlerin dağılımına ilişkin hiçbir koşul gerekmez (Hosmer ve Lemeshow, 2000).

LR modelleri bağımlı değişken kategorisinin yapısına göre üçe ayrı başlık altında incelenebilir.

İkili (Binary) LR modeli, kategorik bağımlı değişkenin iki durumlu (hastalık var-yok, başarı iyi-kötü, ilaç etkili-etkisiz) olduğu durumda kullanılmaktadır.

Çoklu (Multinomial) LR modeli, kategorik bağımlı değişkenin çok kategorili, sınıflanabilir yani nominal ve sıralama içermediği (Medeni durum, Evli- Bekar- Boşanmış) durumlarda kullanılırken;

Sıralı (Ordinal) LR modeli, çok kategorili ve sıralı bir yapı söz konusu ise (hafif-orta-şiddetli) kullanılmaktadır.

Tez çalışmasında iki düzeyli LR yönteminde parametre tahminleri üzerine alternatif optimizasyon teknikleri üzerine çalışıldığı için bundan sonraki bölümlerde sadece iki düzeyli LR ile ilgili tanımlar ve açıklamalar verilecektir.

3.2. LR Modeli

Logit model, bağımlı değişkenin tahmini değerlerini olasılık olarak hesaplayarak olasılık kurallarına uygun sınıflama yapma imkanı veren, tablolaştırılmış ya da ham veri setlerini analiz eden bir istatistiksel yöntemdir (Özdamar, 2004).

İki düzeyli LR modellerinde kategorik bağımlı değişken (Y), bir olgunun gerçekleşmesi (Y=1) ve gerçekleşmemesi (Y=0) gibi iki durumudur.

Bağımsız değişken değerlerine bağlı olarak (x), i. gözlemin bağımlı değişkenin “1” olarak kodlanan düzeyine ait olması olasılığı için matematiksel eşitlik ($\pi(x_i)$) aşağıdaki biçimde tanımlanabilir:

$$\pi(x_i) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p)}} = \frac{e^{\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p}} \quad (3.2)$$

Burada, β_0 , sabit terim; $\beta_1, \beta_2, \dots, \beta_p$, regresyon katsayıları; X_1, X_2, \dots, X_p , bağımsız değişkenler; p, bağımsız değişken sayısı; e = 2.718 doğal logaritma tabanı ve $\pi(x_i)$, i'inci incelenen olayın gerçekleşme olasılığını; $1 - \pi(x_i)$, i'inci incelenen olayın gerçekleşme olasılığını göstermektedir.

$$Z_i = \beta_0 + \beta_1 X_1 + \dots + \beta_p X_p = X_i' \beta \quad (3.3)$$

olarak tanımlanan açıklayıcı değişkenlerin doğrusal kombinasyonları Eşitlik (3.2)'de yerine yazıldığında, $\pi(x_i)$ olasılığı,

$$\pi(x_i) = \frac{1}{1 + e^{-Z_i}} = \frac{e^{Z_i}}{1 + e^{Z_i}} \quad (3.4)$$

biçimine dönüşür. Burada,

$$e^{Z_i} = \frac{\pi(x_i)}{1 - \pi(x_i)} \quad (3.5)$$

olarak ifade edilir. Bu eşitlik bize bir olayın olma olasılığının, olamama olasılığına oranını verir. Bu oran “odds oranı (OR)” olarak adlandırılmaktadır. OR, $\exp(\beta)$ olarak ifade edilir ve Y değişkeninin X_p değişkeninin etkisi ile kaç kat daha fazla ya da yüzde kaç oranında fazla gözlenme olasılığına sahip olduğunu belirtir. Eşitlik (3.5) ile verilen modelin e tabanına göre doğal logaritması alındığında Eşitlik (3.6)'ya ulaşılır.

$$\ln \frac{\pi(x_i)}{1 - \pi(x_i)} = \ln e^{Z_i} = Z_i = \beta_0 + \beta_1 X_1 + \dots + \beta_p X_p = X_i' \beta \quad (3.6)$$

Odds oranının doğal logaritması (L_i) “logit” olarak bilinir. Modeldeki parametreleri tahmin etmek için X 'e ve parametrelerine göre doğrusal olan L_i fonksiyonu,

$$\text{Lojit} = L_i = \ln \left(\frac{\pi(x_i)}{1 - \pi(x_i)} \right) = \beta_0 + \beta_1 X_1 + \dots + \beta_p X_p = X_i' \beta \quad (3.7)$$

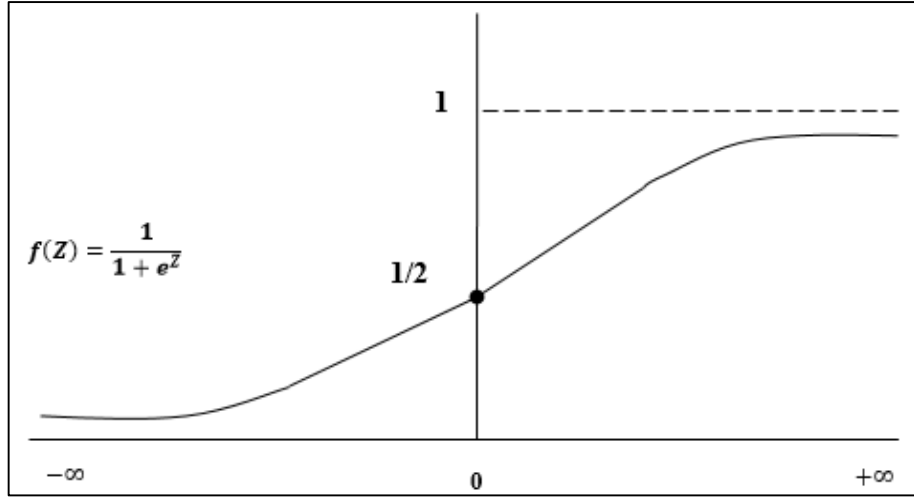
şeklinde yazılır ise bu modele “Lojit Model”,

$$\pi(x_i) = \frac{e^{\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p}} \quad (3.8)$$

biçiminde ifade edildiğinde ise bu modele “Lojistik Regresyon Modeli” adı verilir.

Eşitlik (3.8)'de X bağımsız değişkenleri ne olursa olsun fonksiyonun paydası daima pozitif olacağı için $\pi(x_i)$ ' nin alt sınırı 0 olur. Logit dağılım fonksiyonu olan Z_i değişkeni $-\infty$ ile $+\infty$ arasında değer aldıkça $\pi(x_i)$ ' de 0 ile 1 arasında değerler alacak ve $\pi(x_i)$ ile Z_i arasında doğrusal olmayan bir ilişki olacaktır. Böylece olasılık için gerekli olan $0 \leq \pi(x_i) \leq 1$ koşulu sağlanmış olur. Dolayısıyla da DOM'un en büyük problemlerinden biri ortadan kalkmış olacaktır. Ancak, (3.8) eşitliği incelendiğinde $\pi(x_i)$ 'nin hem bağımsız değişken X_i 'ler, hem de regresyon katsayıları β_i 'ler ile olan ilişkisi doğrusal değildir. Bu nedenle tahmin yapabilmek için EKK metodu kullanılamayacaktır. Ancak uygulanan bazı işlemlerle doğrusal biçime dönüştürülmektedir. Logit hem bağımsız değişkene hem de regresyon katsayılarına göre doğrusaldır. Olasılıkların lojit fonksiyonunun kullanılmasının amacı, doğrusal bir model elde edilerek, parametre tahminlerinin yapılmasıdır.

Logit model parametrelere göre doğrusal ve $-\infty$ ile $+\infty$ arasında değerler alırken, π olasılığı da (0 ile 1) arasında değer almaktadır. 0-1 aralığında sınırlandırılmış ilişkiyi tanımlamak için LR yöntemi, (Şekil 3.1) ile verilen lojistik eğri özelliklerini kullanmaktadır (Kleinbaum, 2002).



Şekil 3.1. Lojistik Fonksiyon Grafiği

Yukarıda verilen grafikte görüldüğü gibi, z 'nin tanım kümesi $(-\infty, +\infty)$ olmaktadır. Buna karşılık z 'nin değer kümesi $[0,1]$ 'dir. Lojistik fonksiyon $f(z)$, z 'nin değerinden bağımsız olarak, 0 ile 1 aralığında kalmaktadır. Böyle bir model, daima 0 ile 1 aralığında kalan bir olasılığı tanımlamak için tasarlanmıştır. Lojistik modeller, tahminlerin 0 ile 1 aralığında olması gerektiği durumlarda bu kısıtı sağlamaktadır. Bu durum da lojistik modelin kullanılmasını yaygınlaştırmaktadır (Kleinbaum, 2002).

3.3. Odds ve Odds Oranı Kavramları

$\pi(x)$ incelenen olayın gözlenme olasılığını göstermek üzere, incelenen bir olayın olasılığının kendi dışında kalan diğer olayların olasılığına oranına Odds değeri denir. İncelenen iki farklı olayın Odds değerlerinin birbirine oranına ise, Odds Oranı (Odds Ratio:OR) denir. LR'de OR, $\exp(\beta)$ olarak ifade edilir.

Olasılık oranı, bir olayın meydana gelme olasılığının meydana gelmeme olasılığına oranı olduğuna göre, $\exp(\beta_p)$, Y değişkenininin X_p değişkeninin etkisi ile kaç kat daha fazla ya da % kaç oranında fazla gözlenme olasılığına sahip olduğunu belirtir (Özdamar, 2004).

$$\text{Odds Oranı} = \frac{\pi(x)}{1-\pi(x)} \quad (3.9)$$

Örneğin, $\pi(x) = 0,75$ alınırsa $OR = \frac{0,75}{1-0,75} = 3$ 'tür. Burada bir olayın gerçekleşme olasılığı, gerçekleşmeme olasılığının 3 katıdır.

X = 1	X = 0
$\pi(1) = \frac{e^{\beta_0 + \beta_1}}{1 + e^{\beta_0 + \beta_1}}$	$\pi(0) = \frac{e^{\beta_0}}{1 + e^{\beta_0}}$
$1 - \pi(1)$	$1 - \pi(0)$
$= \frac{1}{1 + e^{\beta_0 + \beta_1}}$	$= \frac{1}{1 + e^{\beta_0}}$

OR, $x=1$ 'in odds değerinin $x=0$ odds değerine bölünmesinden elde edilir. Yukarıda bulunan lojistik olasılık değerleri aşağıdaki eşitlikte yerine yazıldığında iki sonuçlu bağımsız değişkenin lojistik regresyonu için OR, e^{β_1} olarak bulunacaktır.

$$OR = \frac{\pi(1)/[1 - \pi(1)]}{\pi(0)/[1 - \pi(0)]} = \frac{\frac{e^{\beta_0 + \beta_1}}{1 + e^{\beta_0 + \beta_1}} / \frac{1}{1 + e^{\beta_0 + \beta_1}}}{\frac{e^{\beta_0}}{1 + e^{\beta_0}} / \frac{1}{1 + e^{\beta_0}}} = \frac{e^{\beta_0 + \beta_1}}{e^{\beta_0}} = e^{(\beta_0 + \beta_1) - \beta_0} = e^{\beta_1}$$

OR, özellikle epidemiyolojide geniş kullanım alanları bulan bir bağlantı ölçümüdür. Çünkü $x=1$ 'in içinde sonucun olma olasılığının $x=0$ 'in içinde olma olasılığından ne kadar çok ya da az olduğunun tahminini yapar. Örneğin Y bağımlı değişkenimiz karaciğer kanserinin olup olmasını ve x bir kişinin alkol kullanıp kullanmadığını gösterdiğinde $OR=3$, herhangi bir popülasyonda alkol kullananlar arasında karaciğer kanseri olma olasılığının, alkol kullanmayanlar arasında olma olasılığının 3 katı olduğunu ifade eder (Hosmer ve Lemeshow, 2000).

Örneğin bağımlı ve bağımsız değişkenlerin her ikisi de iki kategorili olmak üzere “Hastalığa yakalanma” ile “Yaş” arasındaki ilişkiyi inceleyelim:

Bağımlı değişken: Hastalığa yakalanma, Bağımsız değişken: Yaş olmak üzere aşağıdaki bilgiler üzerinden bir tahmin yapılmak istendiğinde,

Çizelge 3.3. Örnek veriler

Yaş	Hastalık	
	Var	Yok
60 ve üzeri	25	5
60 tan küçük	15	30

Yaşı 60 ve üzeri olanlarda hastalığa yakalanma odds'u: $25/5 = 5$; yaşı 60 altında olanlarda hastalığa yakalanma odds'u $15/30 = 0.5$ 'dir. Bu iki odds'un birbirine oranı OR'u vermektedir. $OR = 5/0.5 = 10$

Yani risk altında olan bireylerin hastalığa yakalanma oranının, risk altında olmayan bireylere göre 10 kat daha fazla olduğu söylenebilir. Eğer bu oran 1'e eşit çıksaydı hastalığı yakalanma oranında yaşın herhangi bir etkisinin olmadığı söylenebilirdi. LR'de, regresyon katsayıları olan β 'lerin yorumlanmasında, OR'lar sıklıkla kullanılmaktadır.

LR'de olasılıklar 0-1 arasındadır, ancak odds değerleri 0-1 aralığı ile sınırlı değildir. Odds değeri 0 ile $+\infty$ arasında değerler almaktadır. Yani alt sınırı 0 üst sınırı yoktur. Olasılıklar OR'ye dönüştürülerek üst sınır 1'den $+\infty$ 'a çıkartılır. Daha sonra odds değerinin doğal logaritması alınarak da alt sınır $-\infty$ 'a indirilir ve $(-\infty, +\infty)$ aralığına dönüştürülür. Bu dönüşüme “logit dönüşümü” denilmektedir. Bu dönüşümün bağımsız değişkenlerin bir fonksiyonun olarak ifade edilmesi ile de Logit Model oluşmaktadır.

DOM, $\pi(x_i)$ değerinin x_i değerinin doğrusal bir fonksiyonu olduğunu varsaymaktadır. Bir başka değişle X değişkeninde meydana gelecek her değişime, Y değişkeni üzerinde

aynı oranda etki yapacağını varsaymaktadır. Bu varsayım da gerçeklikten uzak niteliktedir. Bu sorun ise DOM'un kullanılmasını anlamsız kılmaktadır. DOM, bu aşılabilir sorunu taşıdığından bu model yerine $\pi(x_i)$ 'nin x_i 'nin doğrusal bir fonksiyonu olmadığı ve $\pi(x_i)$ 'nin 0 ile 1 aralığının dışına çıkmayacak şekilde hareket ettiği olasılık modellerini kullanabiliriz. Yani Odds, olasılık kestiriminin 0-1 arasında olmasına ilişkin problemi ortadan kaldırır ancak odds'un sıfırın altında bir değer almamasına ilişkin ikinci bir sorun vardır. Bu sorunun çözümü de logit değerinin hesaplanması ile mümkündür. Logit, odds'un doğal logaritmasıdır.

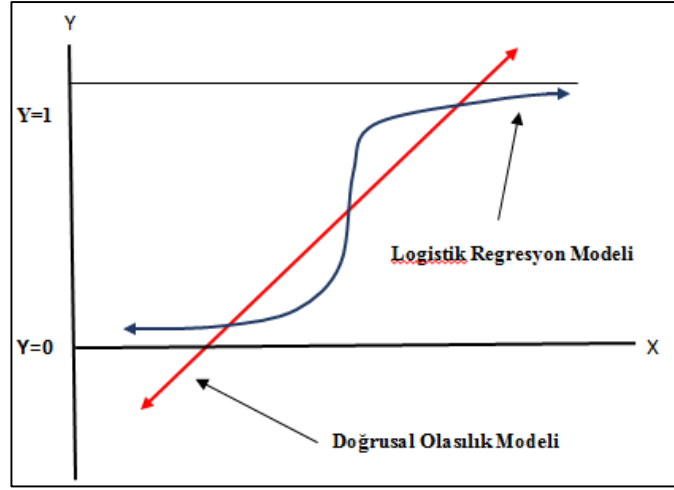
Aşağıdaki çizelgede, bazı olasılıklar ile onlara karşılık gelen odds ve log odds (logit) değerleri verilmektedir.

Çizelge 3.4. LR'de bazı olasılık değerlerine karşılık gelen odds ve logit değerleri

$\pi(x_i)$	$1 - \pi(x_i)$	Odds	Logit
0.0	1.0	0.0	Hesaplanamaz
0.1	0.9	0.111	-2.2
0.2	0.8	0.25	-1.39
0.3	0.7	0.429	-0.847
0.4	0.6	0.667	-0.405
0.5	0.5	1	0
0.6	0.4	1.50	0.405
0.7	0.3	2.33	0.847
0.8	0.2	4	1.39
0.9	0.1	9	2.2
1.0	0.0	Hesaplanamaz	Hesaplanamaz

Çizelge 3.4 incelendiğinde Odds 1'den küçük olduğunda, olasılık 0,5'ten küçük; odds 1'den büyük olduğunda ise olasılık 0,5'ten büyük olmuştur. Ayrıca Odds 1,0'dan küçük olduğu durumlarda logit değeri negatif, büyük olduğu durumlarda ise pozitif olmuştur.

Matematiksel olarak LR olasılık, odds ve odds'un logaritmasına dayanır. LR modeli olasılıklara bir dönüşüm uygulamaktadır. Bu dönüşüm, açıklayıcı değişkenler ile olasılıkların doğrusal bir ilişki içerisinde sonuçlanmasını sağlamaktadır (Menard, 2002).



Şekil 3.2. DOM ve LR Modeli

Görüldüğü gibi, kurulan istatistiksel modellerin başarıya ulaşmasında ilk aşama model parametrelerinin en iyi tahminlerine ulaşmaktır. İstenilen tüm istatistiksel özellikleri sağlayan parametre tahminlerine ulaşılması durumunda modelden elde edilen yorumların geçerliliği de artmaktadır. Bağımlı değişkenin sürekli olduğu durumda kullanılan klasik DR yönteminde parametrelerin tek bir tahmini analitik olarak elde edilebildiğinden dolayı bu modellerde sadece katsayıların anlamlılığı ön plana çıkmaktadır. Ancak LR modelleri gibi doğrusal olmayan modeller ile çalışılması durumunda, parametreler iteratif olarak elde edilebildiğinden dolayı, kullanılacak optimizasyon yöntemi ve sonrasında model parametrelerinin geçerlilik testleri üzerinde durulmaktadır.

Aşağıda, LR model parametrelerinin tahmininde kullanılan EÇOTE ile ilgili matematiksel eşitlikler ve açıklamalara yer verilmiştir. Ayrıca doğrusal olmayan regresyon modellerinin parametre tahmininde kullanılan iterasyona dayalı klasik Newton-Raphson ve alternatif olarak önerilen Nelder-Mead Algoritmaları detayları ile birlikte açıklanmıştır.

4. DOĞRUSAL OLMAYAN REGRESYON MODELLERİNDE PARAMETRELERİN TAHMİNİNDE KULLANILAN KLASİK OPTİMİZASYON TEKNİKLERİ

Doğrusal regresyonda bilinmeyen parametreleri tahmin etmek için en sık kullanılan yöntem EKK yöntemidir. Bu yöntemde parametreler, Y' nin gözlenen değerlerinin, tahmin edilen değerlerinden sapmalarının kareler toplamını minimum yapacak şekilde seçilir. LR modelinde parametre tahmininde literatürde yaygın olarak kullanılan yöntem ise olabilirlik fonksiyonuna dayalı olan EÇOTE'dir. Parametre tahmini için EÇOTE'nin dışında Yeniden Ağırlıklandırılmış İteratif En Küçük Kareler Yöntemi, Minimum Logit Ki-Kare Yöntemi ve bunların dışında özel durumlarda kullanılan bazı tahmin yöntemleri de bulunmaktadır. Ancak, daha önce de belirtildiği gibi yaygın kullanıma sahip olması dikkate alındığında EÇOTE yöntemi ve model parametrelerinin tahminindeki iteratif süreç aşağıda adım adım açıklanmaktadır.

LR'de model parametreleri analitik olarak elde edilemediğinden dolayı, EÇOTE ile iteratif olarak tahminlere ulaşılabilmektedir. Bilindiği üzere EÇOTE, bağımlı değişkenin gözlemlenme olabilirliğini maksimum yapmayı amaçlamaktadır. Genel olarak EÇOTE, gözlenen veri kümesini elde etmenin olasılığını maksimum yapan bilinmeyen parametrelerin değerlerini verir. Bu parametrelerin EÇOTE'leri, fonksiyonu maksimum yapan değerleri bulacak şekilde seçilir. Doğrusal olmayan bir amaç fonksiyonu ile çalışıldığından dolayı parametre tahminlerine daha önce de belirtildiği gibi iteratif olarak ulaşılabilmektedir.

y_i , bir özelliğin olup olmasına göre 0 ve 1 değerini alabilen bağımlı değişkeni ($i=1,2,3,\dots,n$) ve x_i , i 'nci gözlem için bağımsız değişkenin değerini göstermek üzere Eşitlik (3.8)'deki $\pi(x_i)$ ifadesi, verilen x değerleri için y 'nin 1'e eşit olması koşullu

olasılığını vermektedir. LR modeli iki seçenekli bağımlı değişkenin $\pi(x_i)$ olasılığını i. gözlenen değer için Eşitlik (3.10)'daki gibi varsayar. Burada (x_i, y_i) çifti için $y_i = 1$ olduğunda olabirlik fonksiyonuna katkısı $\pi(x_i)$ iken $y_i = 0$ olduğunda olabirlik fonksiyonuna katkısı $1 - \pi(x_i)$ kadardır. Buna göre (x_i, y_i) çiftinin olabirlik fonksiyonuna katkısı Eşitlik (4.1)'deki gibidir.

$$\pi(x_i)^{y_i} [1 - \pi(x_i)]^{1-y_i} \quad (4.1)$$

Gözlemlerin birbirlerinden bağımsız olduğu varsayıldığında, Eşitlik (4.1)'deki terimlerin çarpılmasıyla elde edilen olabirlik fonksiyonu Eşitlik (4.2) ile verilmiştir:

$$l(\beta) = \prod_{i=1}^n \pi(x_i)^{y_i} [1 - \pi(x_i)]^{1-y_i} \quad (4.2)$$

EÇOTE, bu eşitliği maksimum yapan parametreleri tahmin etme temeline dayanır. Kolaylık olması bakımından, bu ifadenin logaritması alındığında çarpım durumu toplam durumuna getirilerek Eşitlik (4.3) ile verilen ifade elde edilir:

$$L(\beta) = \ln[l(\beta)] = \sum_{i=1}^n \{y_i \ln [\pi(x_i)] + (1 - y_i) \ln [1 - \pi(x_i)]\} \quad (4.3)$$

Logaritması alınmış fonksiyon log-olabirlik (log-likelihood) fonksiyonu olarak adlandırılır. Bu eşitliği maksimum yapan β değerlerini bulabilmek için eşitliğin β 'ya göre türevi alınıp sıfıra eşitlenerek Eşitlik (4.4) ile verilen olabirlik eşitlikleri elde edilmektedir (Hosmer ve Lemeshow, 2000):

$$\frac{\partial L(\beta)}{\partial \beta} = \sum_{i=1}^n x_i [y_i - \pi(x_i)] = 0 \quad (4.4)$$

Bu eşitliklerden elde edilen parametre tahminleri, en çok olabirlik tahminidir. LR için Eşitlik (4.4) ile verilen olabirlik eşitlikleri, β 'da doğrusal olmadıklarından dolayı denklemin köklerini oluşturan model parametrelerinin tahmini için aşağıda

ayrıntlarına yer verilen sayısal algoritmalar kullanılmaktadır.

4.1. Newton-Raphson Algoritması

NR algoritması olabirlik eşitliklerinin çözümünde en fazla kullanılan özel iteratif bir yöntemdir (Hosmer ve Lemeshovv, 2000). Olabirlik fonksiyonundaki değişim sonraki aşamalarda ihmal edilebilir duruma gelinceye kadar iteratif süreç devam edilir. Olabirlik fonksiyonunu maksimize edecek en iyi parametre kümesini oluşturmak için yapılan tüm bu işlemler için bilgisayar uygulamalı algoritmalar tasarlanmıştır (Kutner vd., 2004; Menard, 2002)

Yöntem bir çok yerel maksimuma takılabilmekte ve global maksimum ya da minimum noktalara ulaşılmeden iterasyon tamamlanabilmektedir. Bu bir olumsuzluktur.

Eşitlik (4.3)'e tekrar dönelim ve $\pi(x_i)$ yerine Eşitlik (3.8) ile verilen ifadeyi yazalım.

$$L(\beta) = \sum_{i=1}^n \left[y_i \ln \left(\frac{e^{(x_i^T \beta)}}{1 + e^{(x_i^T \beta)}} \right) + (1 - y_i) \ln \left(1 - \frac{e^{(x_i^T \beta)}}{1 + e^{(x_i^T \beta)}} \right) \right] \quad (4.5)$$

NR iterasyon sürecinde, olabirlik fonksiyonunu maksimum yapan β 'nın tahminini elde etmek için başlangıç değerleri belirlenir. Ancak, bu başlangıç değerleri için çeşitli alternatifler öne sürülmüştür (Agresti, 2002). NR yönteminde, EKK tahminlerinin temel alınması önerilmiştir. Newton ve Raphson tarafından önerilen ve $(m+1)$ 'inci iterasyonda $\hat{\beta}$ değerini bulan iterasyon,

$$\hat{\beta}^{(m+1)} = \hat{\beta}^{(m)} + (X^T V_M X)^{-1} X^T (Y - P_m) \quad (4.6)$$

biçiminde ifade edilir.

Burada, $P_m^T = (\pi_{1,m}, \pi_{2,m}, \dots, \pi_{n,m})$ önceki iterasyondan elde edilen yanıt olasılıklar vektörü ve $V_m = \text{diag}[\pi_{i,m} (1 - \pi_{i,m})]$ önceki iterasyonda elde edilen köşegen matristir. Yeniden Ağırlıklandırılmış İteratif En Küçük Kareler Yöntemi (Iteratively

Reweighted Least Square Method) ile $\hat{\beta}^{(m)}$, $\hat{\beta}^{(m+1)}$ 'e yeterince yakın olana kadar iterasyona devam edilir. Yakınsama ölçüsü olarak, $\hat{\beta}^{(m+1)}$ ve $\hat{\beta}^{(m)}$ arasındaki fark için 1×10^{-8} , 1×10^{-7} , 1×10^{-6} ($1e-6$) gibi değerler kullanılmaktadır (Agresti, 2002; Aktaş, 2009; Arıcan, 2010). Başlangıç değerlerinin doğru seçimi iterasyon sayısı üzerinde önemli bir etkiye sahiptir. İyi bir başlangıç değeri ile az sayıda iterasyon ile optimum çözüme ulaşılabilmektedir.

4.2. Nelder- Mead Algoritması

Matematiksel olarak ifade etmek gerekirse optimizasyon kısaca bir fonksiyonun minimize veya maksimize edilmesi olarak tanımlanabilir. Optimizasyon teknikleri içerisinde çok değişkenli, kısıtsız, türev kullanmayan tekniklerden biri olan NM optimizasyon metodu çok değişkenli bir fonksiyonun yerel minimumunu bulmak için John Ashworth NELDER ve Roger MEAD tarafından 1965 yılında geliştirilmiş bir simpleks yöntemidir. N boyutlu bir geometrik şekil olan simpleks, n boyutta (n+1) nokta içerir. Yani iki boyuttaki simpleks, bir üçgenden oluşur. Üç boyuttaki simpleks, dört yüzeyli bir üçgenler prizması şeklini alır. Bu metot ile n değişkenli fonksiyonun minimum olduğu noktanın koordinatlarını ve minimum değerini bulurken, n+1 noktanın başlangıç koordinatlarının verilmesi gerekir. Bu koordinatlar verildikten sonra algoritma yüzey topolojisini arayarak minimuma doğru hareket etmeye başlar. Başlangıç koordinatlarına bağlı olarak hesaplanan değer lokal veya global minimum olabilir.

Bu yöntemde (n + 1) adet nokta, bir düzenli simpleks (basit ve düzenli) olarak bilinen n-boyutlu Öklid uzayında karşılıklı olarak konulur. İki boyutta simpleks bir üçgen olduğundan, ilk üçgenin üç köşe noktası başlangıç vektörleri alınarak işleme başlanır. Bu köşe noktalarda maksimum problem için en büyük, minimum problem için en küçük fonksiyon değerine sahip noktaya “En İyi Köşe Nokta” (Best Vertex: B); sonraki en iyi noktaya “Sonraki En İyi Köşe Nokta” (Good Vertex: G) ve maksimizasyon problemlerinde en küçük, minimizasyon problemlerde en büyük fonksiyon değerine

sahip noktaya da “En Kötü Köşe Nokta” (Worst Vertex: W) adı verilir. Bu işlem her iterasyonda tekrarlanarak iki boyutlu problemlerde üçgenlerin bir dizisi ile optimum çözüme yaklaşılr. B, G ve W ile isimlendirdiğimiz üçgenin bu köşe noktaları birbirlerine eşit olan nokta, aranılan optimum noktadır.

İki boyutta bu yöntem üçgenin köşelerinde fonksiyon değerlerini karşılaştıran bir model arama yöntemidir. İki değişkenli bir $z = f(x, y)$ fonksiyonunun minimizasyon durumunu göz önüne aldığımızda, üçgenin en kötü köşesinde (w=worst vertex) $z = f(x, y)$ fonksiyonunun değeri en büyüktür. Bu nedenle, bu en kötü köşe yeni bir nokta ile yer değiştirir. Böylece yeni bir üçgen elde edilir. Artık arama işlemi bu yeni üçgen içinde devam ettirilir ve böylece bu işlem köşe nokta değerinde fonksiyon değerinin gittikçe küçüldüğü bir üçgenler dizisine dönüşür. İşlem üçgenlerin küçülerek bir noktaya yaklaşması ile son bulur ki bu nokta istenen minimum noktadır. İki değişkenli $z = f(x, y)$ fonksiyonu için geliştirilen bu metot n-değişkenli $f(x) = f(x_1, x_2, \dots, x_n)$ fonksiyonuna genelleştirilebilir.

Bu yöntemde simpleksin hareketi üç temel işlem olan Yansıtma (Reflection: R), Büyüme-Genişleme (Expansion: E) ve Büzülme-Kısalma (Contraction: C) ile sağlanır (Mathews ve Fink, 2004; Tektaş, 2010).

Başlangıç üçgeni

Minimize edilecek olan $f(x, y)$ fonksiyonu için $k=1,2,3$ olmak üzere $V_k = (x_k, y_k)$ üçgenin üç köşe noktası değerleri verilerek oluşturulur. Daha sonra bu üç köşe noktanın her birinde $z_k = f(x_k, y_k)$ olan fonksiyon değerleri bulunduktan sonra $z_1 \leq z_2 \leq z_3$ olacak şekilde sıralanır.

Üçgenin köşe noktaları $B(x_1, y_1)$, $G(x_2, y_2)$, $W(x_3, y_3)$ biçiminde tanımlanır.

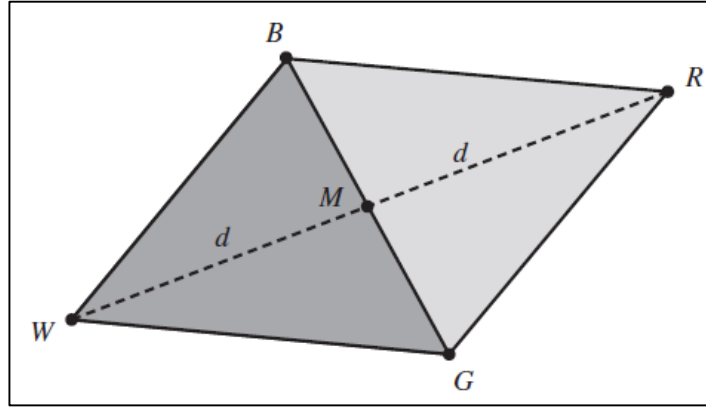
İyi kenarın orta noktası

B ve G noktalarının birleşmesiyle oluşan doğru parçasının orta noktasını bulmak amacıyla kullanılır. Bu nokta, koordinatların ortalamasının alınmasıyla bulunur:

$$M = \frac{B+G}{2} = \left[\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2} \right] \quad (4.7)$$

Yansıma işlemini kullanarak R noktasının bulunması

R noktasını belirleyebilmek için ilk olarak BG kenarının M noktası yukarıda tanımlandığı gibi tespit edilir. İkinci olarak W ile M noktaları birleştirilerek bir doğru elde edilir. Bu doğrunun uzunluğuna d denilir. Son olarak M noktasından d uzunluğunda bir doğru çizilerek bu doğrunun bittiği noktaya R denilir. Böylece yansıma işlemini kullanarak R noktası elde edilmiş olacaktır. Süreç, aşağıda grafiksel olarak gösterilmektedir.



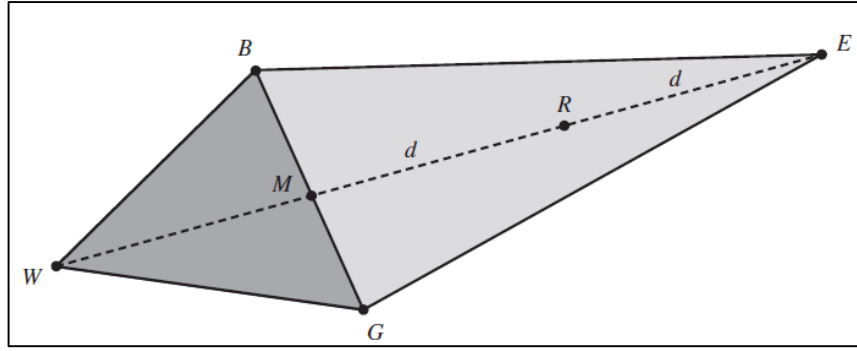
Şekil 4.1. M orta noktası ve yansıma işlemini kullanarak R noktasının bulunması

R noktasının vektörel notasyonu ise Eşitlik (4.8)'de verilmiştir.

$$R = M + (M - W) = 2M - W \quad (4.8)$$

Uzatma işlemini kullanarak E noktasının bulunması

Eğer R'deki fonksiyon değeri W'deki fonksiyon değerinden daha küçükse yani $f(R) \leq f(w)$ ise minimuma doğru bir yönde hareket edilir. R noktası minimum noktaya uzak olabilir. Bu nedenle M noktasından R noktasına olan doğru R noktasından aynı d uzunluğunda genişletildiğinde bitim noktası E elde edilir. Böylece BGW üçgeninden genişleme işlemi ile BGE üçgeni elde edilir.



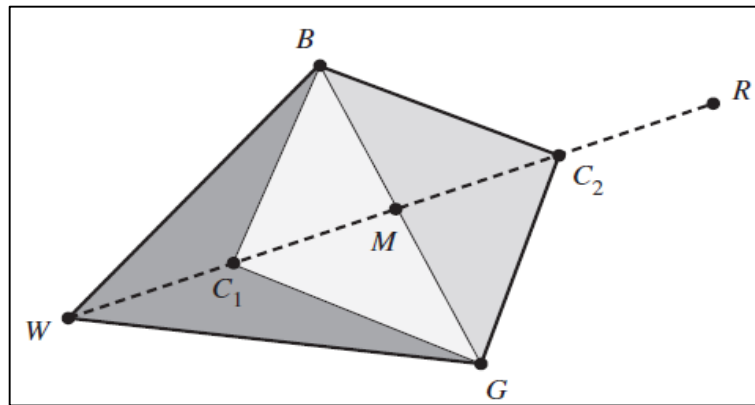
Şekil 4.2. Genişleme işlemi ile BGE üçgeninin ve E noktasının elde edilmesi

Eğer E noktasındaki fonksiyon değeri R noktasındaki fonksiyon değerinden daha küçük ise yani $f(E) < f(R)$ ise, R'den daha iyi bir köşe bulunmuş olur. Genişleme işlemi ile elde edilen E noktasına ait vektör formülasyonu Eşitlik (4.9)'da verilmiştir.

$$E = R + (R - M) = 2R - M \quad (4.9)$$

Daraltma (büzülme) işlemini kullanarak C (C1, C2) noktasının bulunması

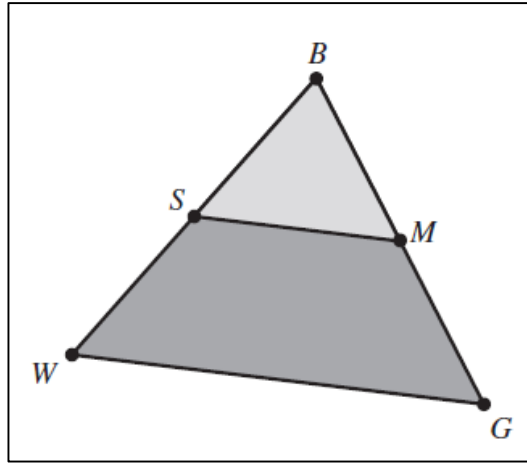
Eğer R noktasındaki fonksiyon değeri W noktasındaki fonksiyon değeri ile aynı ise yani, $f(R) = f(W)$ ise, başka bir nokta test edilmek zorundadır. Fonksiyon M noktasında küçük olabilir fakat M ile W yeniden yazılamaz. Çünkü yeni bir üçgen elde etmek zorunluluğu vardır. WM ve MR doğrularının orta noktalarına sırasıyla C1 ve C2 denildiği zaman yeni üçgenimiz BGC olacaktır. Burada BGC1 veya BGC2 olması iki boyutlu durumunda fark etmeyecektir. Fakat çok boyutlu durumda farklılık gösterir.



Şekil 4.3. C1 ve C2 daralma noktaları

Üçgenin B noktasına doğru daraltılması

Eğer C (C_1 veya C_2) noktasındaki fonksiyon değeri W noktasındaki fonksiyon değerinden daha küçük değilse yani $f(c) \leq f(w)$ ise G ve W noktalarının B noktasına doğru daraltılması gerekir (Şekil 4.4). Bu daraltma işleminin ilk adımında G noktası yerine BG doğrusunun orta noktası olan M noktası ve W noktası yerine BW doğrusunun orta noktası olan S noktası alınır. Sonraki ardışık işlem adımları B noktasına doğru hep orta noktalar alınarak devam ettirilir. İşlem B noktası elde edildiğinde son bulur.



Şekil 4.4. B noktasına doğru daraltma işlemi

Her bir adım için mantıksal kararlar

Yukarıda açıklanan her bir adımda yeni bir köşe bulunur ve bu W ile değiştirilir. Bunu bir algoritma olarak şu şekilde açıklayabiliriz.

$f(R) < f(G)$ ise Safha – 1; $f(R) \leq f(G)$ ise Safha - 2 düzenlenir.

Safha – 1: Yansıma veya Genişleme

$f(B) < f(R)$ ise W yerine R alınır.

$f(B) \leq f(R)$ ise E ve $f(E)$ hesaplanır.

$f(E) < f(B)$ ise W yerine E alınır.

$f(E) \leq f(R)$ ise W yerine R alınır.

Safha – 2: Büzülme veya Daralma

$f(R) < f(W)$ ise W yerine R alınır. $C = \frac{W+M}{2}$ ve $f(C)$ değeri hesaplanır.

$f(C) < f(W)$ ise W yerine C alınır.

$f(C) \leq f(W)$ ise S yerine $f(S)$ hesaplanır. W yerine S; G yerine ise M alınır (Mathews ve Fink, 2004; Tektaş, 2010).

Algoritmanın kodlanmış biçimi aşağıda verilmektedir.

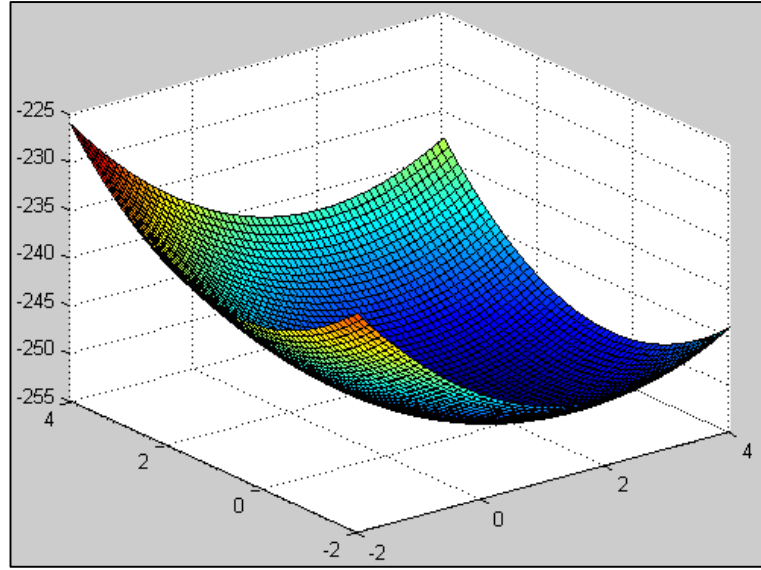
Çizelge 4.1. Nelder-Mead algoritması için mantıksal kararlar (SQL)

IF $f(R) < f(G)$, THEN Perform Case (i) {either reflect or extend} ELSE Perform Case (ii) {either contract or shrink}	
BEGIN {Case (i).} IF $f(B) < f(R)$ THEN replace W with R ELSE Compute E and $f(E)$ IF $f(E) < f(B)$ THEN replace W with E ELSE replace W with R ENDIF ENDIF END {Case (i).}	BEGIN {Case (ii).} IF $f(R) < f(W)$ THEN replace W with R Compute $C = (W + M)/2$ or $C = (M + R)/2$ and $f(C)$ IF $f(C) < f(W)$ THEN replace W with C ELSE Compute S and $f(S)$ replace W with S replace G with M ENDIF END {Case (ii).}

4.2.1. İki boyutlu örnek bir problem için NM metodunun çıktıları

NM algoritması kullanılarak aşağıda üç boyutlu grafiği verilen iki değişkenli $f(x, y)$ fonksiyonun minimum değerinin bulunması aşamalarını inceleyelim.

$$f(x, y) = x^2 - 4x + y^2 - y - xy \quad (4.10)$$



Şekil 4.5. f fonksiyonunun üç boyutlu grafiği

Bu problemde f fonksiyonunu minimum yapan x,y koordinatları ve fonksiyonun minimum değeri aranmaktadır. NM simplex metodunda iterasyona başlanılabilmesi için, değişken sayısının bir fazlası kadar noktada değişkenlerin başlangıç değerinin belirlenmesi gerekmektedir. Bu örnekte fonksiyon iki değişkenli olduğundan dolayı üç nokta için x ve y başlangıç değerleri verilmelidir. Bu noktalar aşağıdaki V_1, V_2, V_3 köşe noktası değerleri olsun.

$$V_1 = (0, 0) \quad V_2 = (1.2, 0) \quad V_3 = (0, 0.8)$$

Bu noktalar için $f(x,y)$ fonksiyonu aşağıdaki değerleri alacaktır.

$$f(0, 0) = 0.0 \quad f(1.2, 0) = -3.36 \quad f(0, 0.8) = -0.16$$

B,G,W noktalarını belirlemek için fonksiyon değerleri karşılaştırılmalıdır.

$$Z_1 = B = (1.2, 0) \quad Z_2 = G = (0, 0.8) \quad Z_3 = W = (0, 0)$$

$W = (0,0)$ değeri yer değiştirilecektir. M ve R noktaları;

$$M = \frac{B+G}{2} = (0.6, 0.4)$$

$$R = 2M - W = (1.2, 0.8)$$

olarak bulunur.

R fonksiyonunun değeri $f(R) = f(1.2, 0.8) = -4.48$, $f(G)$ değerinden daha küçük olduğundan dolayı 1. durum geçerlidir. Bu durumda sağ yöne doğru hareket edildiğinde daha minimum değerlere ulaşılabileceği düşünülmektedir.

E noktasının tepe değeri şu şekilde oluşturulmalıdır:

$$E = 2R - M = 2*(1.2, 0.8) - (0.6, 0.4) = (1.8, 1.2)$$

$f(E) = f(1.8, 1.2) = -5.88$ fonksiyonunun değeri $f(B)$ değerinden düşük değerlidir ve yeni üçgen aşağıdaki tepe noktası değerlerini alır.

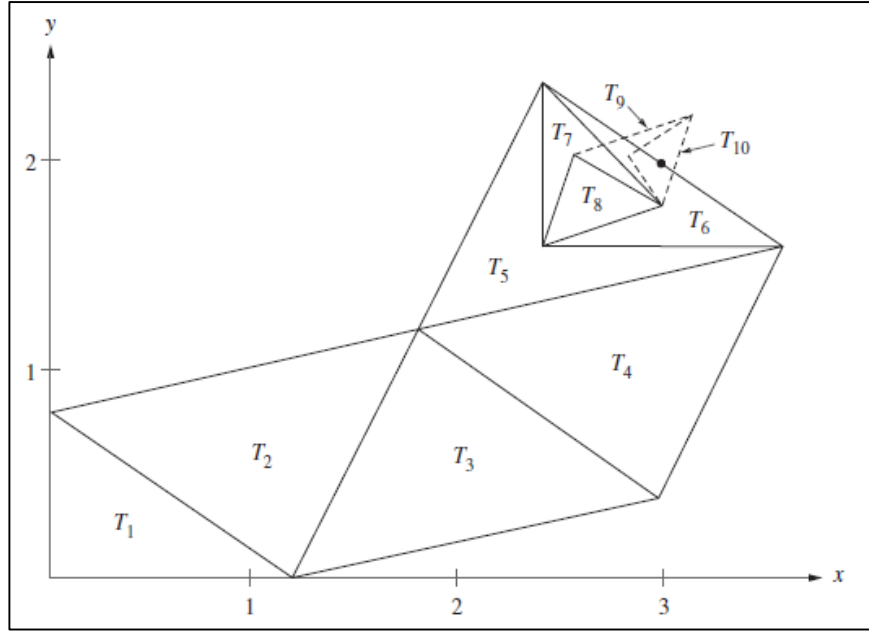
$$V_1 = (1.8, 1.2) \quad V_2 = (1.2, 0.0) \quad V_3 = (0.0, 0.8)$$

Bu süreç devam eder ve çözüm noktası (3,2) noktasına yaklaşan bir üçgenler dizisi oluşturur. Çizelge (2.2)'de iterasyonun birçok adımı için üçgenin tepe noktalarındaki fonksiyon değeri verilmiştir. En iyi B tepe noktası ve $f(B)$ değerleri sırasıyla $B=(2.99996456, 1.99983839)$ ve $f(B) = -6.99999998$ değerlerine ulaşıncaya kadar 33 adım boyunca algoritmanın bilgisayar uygulaması devam eder. Bu değerler $f(3, 2) = -7$ değerlerine yaklaşık değerlerdir.

Bu aşama ve iterasyonların birçoğu aşağıdaki çizelge ve şekilde gösterilmiştir (Mathews ve Fink, 2004).

Çizelge 4.2. Üçgenlerin fonksiyon değerleri

k	B	G	W
1	$f(1.2, 0.0) = -3.36$	$f(0.0, 0.8) = -0.16$	$f(0.0, 0.0) = 0.00$
2	$f(1.8, 1.2) = -5.88$	$f(1.2, 0.0) = -3.36$	$f(0.0, 0.8) = -0.16$
3	$f(1.8, 1.2) = -5.88$	$f(3.0, 0.4) = -4.44$	$f(1.2, 0.0) = -3.36$
4	$f(3.6, 1.6) = -6.24$	$f(1.8, 1.2) = -5.88$	$f(3.0, 0.4) = -4.44$
5	$f(3.6, 1.6) = -6.24$	$f(2.4, 2.4) = -6.24$	$f(1.8, 1.2) = -5.88$
6	$f(2.4, 1.6) = -6.72$	$f(3.6, 1.6) = -6.24$	$f(2.4, 2.4) = -6.24$
7	$f(3.0, 1.8) = -6.96$	$f(2.4, 1.6) = -6.72$	$f(2.4, 2.4) = -6.24$
8	$f(3.0, 1.8) = -6.96$	$f(2.55, 2.05) = -6.7725$	$f(2.4, 1.6) = -6.72$
9	$f(3.0, 1.8) = -6.96$	$f(3.15, 2.25) = -6.9525$	$f(2.55, 2.05) = -6.7725$
10	$f(3.0, 1.8) = -6.96$	$f(2.8125, 2.0375) = -6.95640625$	$f(3.15, 2.25) = -6.9525$



Şekil 4.6. Optimizasyonu (3,2) noktasına yaklaştıran $\{T_k\}$ üçgenleri dizisi

Yukarıda bir örnek üzerinden açıklanan NM algoritmasının Matlab kodları ve sonuçları aşağıda verilmektedir.

4.2.2. Matlab programı ile Nelder-Mead algoritmasının çözümü

Matlab paket programı kullanılarak NM ile bir optimizasyon probleminin çözümünün iki yolu vardır.

Birincisi kod yazmayı gerektirmeyen, “Optimization Toolbox” işlemlerini kullanmaktır.

Bu yöntemde sırasıyla,

“Matlab/Start/Toolboxes/Optimization/Optimization tool/”

menüsü seçilerek açılan ekranda “Solver” bölümünden “fminsearch” seçilir.

Daha sonra buradaki “Objective function” bölümüne aşağıda tanımlanan fonksiyon ve “Start Point” bölümüne başlangıç köşe noktaları girilir. “start” seçildiğinde, sağ tarafta girilen özelliklere göre ‘fonksiyon’ minimize edilecektir. Burada açılan ekranın sağ tarafında bulunan ‘options’ kısmı önemlidir. Burada NM algoritmasında kullanılan

tüm operatörler seçenek olarak bulunmaktadır (stopping criteria, plot functions, output functions vb.). Optimize edilecek fonksiyonun özelliklerine göre seçenekler değiştirilerek daha kısa sürede çözüme yakınsama sağlanabilmekte veya çözüm için daha iyi sonuçlar bulunabilmektedir.

İkinci yol ise Matlab’de “command window” bölümüne gerekli kodları yazarak sonuca ulaşmaktır. “Fminsearch” kısıtsız çok boyutlu doğrusal olmayan minimizasyonu NM Algoritması ile çözmektedir.

NM algoritması kullanılarak aşağıda Matlab kodları verilen iki değişkenli $f(x, y)$ fonksiyonun minimum değerinin bulunmasını inceleyelim.

$$f(x, y) = x_1^2 - 4x_1 + x_2^2 - x_2 - x_1x_2$$

Eşitlik (4.10) ile verilen fonksiyon, Matlab Paket programında ‘fonksiyon’ ismiyle m dosyası olarak aşağıdaki gibi tanımlanmıştır.

```
function [a] = fonksiyon(x)
a = x(1)^2 - 4*x(1) + x(2)^2 - x(2) - x(1)*x(2);
end
```

(4.11)

Daha sonra kod yazımı için kullanılan “command window” ekranına aşağıdaki optimum sonucu veren kodlar ve köşe nokta değerleri girilmiştir.

```
[X,Y,Exitflag, Output]=fminsearch(@fonksiyon, [0:0,1.2:0,0:0.8])
```

Bu komut verildiğinde Matlab paket programı aşağıdaki çıktıyı verecektir. Çıktının tamamı Ek.A’da verilmiştir.

Iteration	Func-count	min f(x)	Procedure
0	1	0	
1	3	-0.000999937	initial simplex
2	5	-0.00187486	expand
⋮	⋮	⋮	⋮
68	133	-7	contract inside
69	134	-7	reflect

EXITFLAG = 1

OUTPUT =

iterations: 69

funcCount: 134

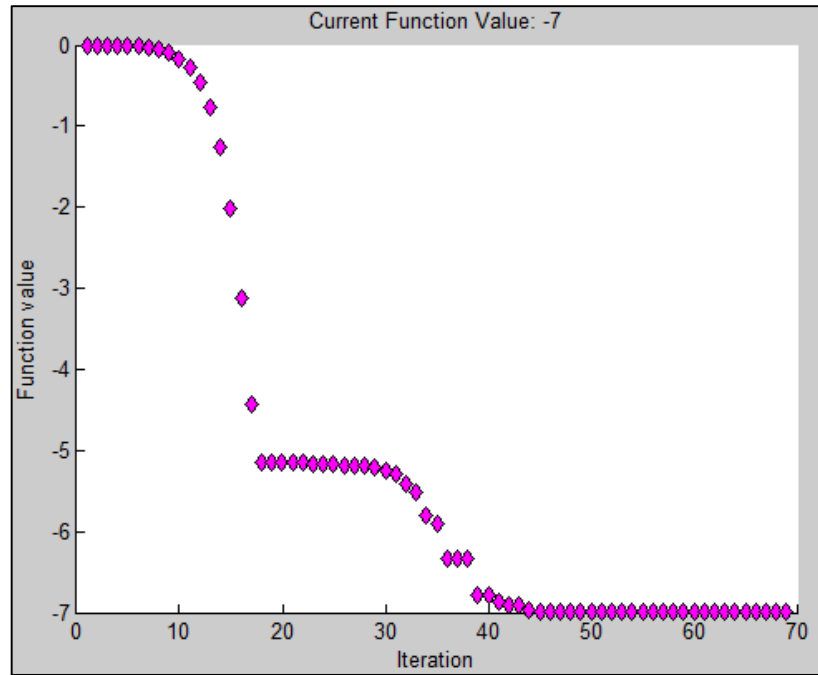
algorithm: 'Nelder-Mead simplex direct search'

message: [1x196 char] Optimization terminated: the current x satisfies the termination criteria using OPTIONS. TolX of 1.000000e-004 and F(X) satisfies the convergence criteria using OPTIONS. TolFun of 1.000000e-004

X = [3.0000,2.0000]

Y = - 7.0000

69 iterasyon sonucu fonksiyonunun negatifini minimum yapan (fonksiyonu maksimum yapan) x_1 ve x_2 değerleri sırasıyla 3 ve 2 olarak bulunmuştur. Fonksiyonun minimum değeri ise -7 olarak bulunmuştur. Fonksiyonun 69 iterasyon sonucunda minimum değere yaklaşırken aldığı değerler Şekil (4.7)'de gösterilmiştir.



Şekil 4.7. Nelder-Mead Algoritması İterasyonları

Burada herhangi bir fonksiyonu maksimum ya da minimum yapan deęişken deęerleri, optimizasyon problemlerinde “Arama Yöntemleri” olarak adlandırılan yöntemlere benzer bir seyir izleyen Nelder-Mead Algoritması ile belirlenmiştir. Bizim asıl ilgilendiğimiz nokta ise LR analizinde Olabilirlik Fonksiyonunu maksimum yapan, olabilirlik eşitliklerinin köklerini $(\hat{\beta}_0, \hat{\beta}_1 \dots \hat{\beta}_k)$ en iyi optimizasyon yöntemi kullanarak bulmaktır.

İteratif yöntemlerin kısıtlayıcı varsayımları dikkate alındığında olabilirlik fonksiyonunu maksimize edecek en iyi parametre kümesini oluşturmak için her ne kadar NM algoritması NR algoritmasına iyi bir alternatif gibi görünse de analitik yöntemler ile karşılaştırıldığında daha hızlı yakınsama hızına sahip, fonksiyonların diferansiyellenebilir olmasını gerektirmeyen bazı sezgisel optimizasyon yöntemlerinin de var olduğu bilinmektedir.

Genetik Algoritma (GA) yaklaşımı son yıllarda oldukça yaygın bir şekilde kullanılan sezgisel yöntemlerden birisidir ve özellikle doğrusal olmayan fonksiyonların optimizasyonunda sıkça kullanılmaktadır.

5. DOĞRUSAL OLMAYAN MODELLERDE PARAMETRE TAHMİNİNE GENETİK ALGORİTMA YAKLAŞIMI

Genetik algoritmaların temeli Charles Darwin'in görüşlerine dayandırılan doğayı taklit etme ve bunun sonucu olarak güçlü olan bireylerin yaşaması prensibine dayanır (Michalewicz, 1996). GA'lar, canlılardaki doğal seçim ilkelerini ve genetik süreçleri taklit eden optimizasyon ve arama algoritmalarıdır.

GA kavramı literatüre 1960 yılında Michigan Üniversitesinden Prof. John Holland tarafından kazandırılmış ve 1970'lerde kendisi ve öğrencileri tarafından geliştirilmeye başlanmıştır. Holland'ın (1975) "Adaptation in Natural and Artificial Systems" adlı eserinde iki temel başlık olarak doğal sistemlerin uyum sürecini anlayıp özetlemek ve doğal sistemlerin önemli mekanizmalarını içeren yapay sistemlerin yazılım olarak tasarlanması amaçlanmıştır (Goldberg, 1989).

John Holland, Charles Darwin'in "en iyinin hayatta kalması" prensibinden yola çıkarak, 1970'li yılların başında GA'lar ile ilgili çalışmalarına başlamıştır. Bu çalışmalar günümüzde evrim stratejileri, evrimsel programlama, yapay zeka, sınıflama sistemleri ve genetik programlama gibi çalışma alanlarının açılmasına öncülük etmiştir (Reeves ve Rowe, 2002).

GA, Holland'ın çalışmasıyla literatüre kazandırılmış, sonrasında öğrencisi Goldberg'in de katkılarıyla bir yapay zeka optimizasyon tekniği olarak geliştirilmiş ve gerçek hayata ilişkin problemlerin çözümünde kullanılmıştır. Goldberg'in optimizasyon problemlerinde kullanımı ile ilgili "Genetik Algorithms in Search, Optimization and Machine Learning" isimli eseri GA'ların uygulama alanlarının gelişmesini sağlamıştır (Michalewicz, 1996).

GA, popülasyon (erişkinlerin ya da yetişkinlerin bir kitlesi) denilen ve kromozomlar ile gösterilen çözümlerin bir kümesi ile başlatılır. Bir popülasyondan çözümler alınır. Bu çözümler daha sonra yeni bir popülasyon oluşturmak için kullanılır. Bu işlem yeni popülasyonun eski popülasyondan daha iyi olacağı varsayımından hareketle yapılır. Yeni çözümleri (nesilleri) oluşturmak için seçilen çözümler uygunluk ya da uyumluluk değerlerine göre seçilir (Goldberg, 1989).

GA' da evrim süreci potansiyel çözüm uzayı boyunca kromozom popülasyonlarının GA'nın operatörlerinin işletilmesiyle gerçekleştirilir. GA'nın kullandığı operatörler Kopyalama, Çaprazlama ve Mutasyon operatörleridir. Bu operatörler yardımıyla GA, ilgilenilen problemin çözümü için çözüm uzayının taranması ve en iyi çözümün bulunmasını amaçlar (Goldberg, 1989).

Popülasyon için hesaplanan uygunluk değerlerine göre kromozomlar iyi veya kötü olarak değerlendirilebilir. İyi bireylerin yaşamlarını devam ettirme olasılıkları ve kötü olan bireylerin ölme olasılıkları artırılarak iyi bireylerin çoğalmaları ve kötü bireylerin yok olmaları beklenir. Evrim sürecinde GA'nın evrim operatörleri yardımıyla kromozomlar farklı bir popülasyon oluşturur ve bir iterasyon tamamlanır. Belirli iterasyon sayısına ulaşıncaya veya durdurma koşulu sağlandığında son popülasyondaki en iyi uygunluk değerine sahip kromozomun kodladığı çözüm, problemin çözümü olarak belirlenir (Mitchell, 1999).

Popülasyondaki her bir bireye kromozom denilmektedir. Kromozomlar dizin sembolü olarak kullanılırlar. Kromozomlar kural olmamakla birlikte kullanım aşamasında ikili sayı düzeninde bit dizini olarak ifade edilirler. Kromozomların birbirini takip eden iterasyonlarla evrim geçirmelerine jenerasyon (nesil) adı verilir. Her bir jenerasyon süresince kromozomlar uygunluk koşulu uyarınca evrim geçirirler. Yeni jenerasyon yaratmak için, mevcut jenerasyonun iki kromozomu çaprazlama (gen takası) operatörü (crossover operator) ya da mutasyon operatörü (mutation operator) kullanılarak elde edilir. Elde edilen yeni bireye döl (offspring) denir. Yeni jenerasyonda uygunluk kriterine (fitness value) uyan ebeveynler ve dölleri, jenerasyon içerisinde yer alırken, uygunluk koşuluna uymayan ebeveynler ve dölleri jenerasyondan yok edilirler. Bu işlem doğal seleksiyon (seçim) işlemi adını alır. Böylelikle en iyilerin yaşama şansının

bulunduđu jenerasyonların popölasyon büyüklüğü korunmuş olur. Bu işlem uyarınca en uygun kromozomların seçilme şansı maksimize edilmiş olur. Birkaç jenerasyon sonra, algoritma en iyi kromozomların bulunduğu optimum çözüme yakınsar (Akyol, 2006).

GA'ların bir problemin çözümünde kullanılabilmesi için aşağıdaki beş bileşenin sağlanması gerekir. (Goldberg, 1989)

- Problem için potansiyel çözümlerin genetik gösterimi (çözümlerin kromozomlarla ifade edilmesi),
- Potansiyel çözümlerin (başlangıç popölasyonunun) oluşturulması için bir yol,
- Çözümlerin uygunluğunu değerlendiren değerlendirme fonksiyonunu,
- Genetik operatörlerin (Kopyalama, Çaprazlama ve Mutasyon) belirlenmesi,
- GA'nın kullanacağı parametre değerlerinin belirlenmesi (popölasyon büyüklüğü ve genetik operatörlerin değerleri)

5.1. Genetik Algoritmanın Uygulama Alanları

GA'lar uygulamadaki kolaylıkları nedeniyle bilgisayar bilimi, işletme, mühendislik, eğitim, matematik, tıp ve ziraat gibi birçok alanda yaygınlıkla kullanılabilmektedir. GA'ların başlıca uygulama alanları optimizasyon, otomatik programlama ve bilgi sistemleri, mekanik öğrenme, ekonomik ve sosyal sistem modelleri, başlıkları altında toplanabilir (Emel ve Taşkın, 2002).

GA'nın uygulama alanlarından bazılarını, haberleşme şebekleri tasarımı, elektronik devre dizaynı, gaz boruları şebeke optimizasyonu, görüntü ve ses tanıma, veri tabanı sorgulama optimizasyonu, bilgisayar çipleri tasarımı, ders programı hazırlanması ve ağların çizelgelenmesi, uçak tasarımı, fiziksel sistemlerin kontrolü, gezgin satıcı problemlerinin çözümü, iş atölye çizelgeleme problemi, araç rotalama problemleri, ulaşım problemleri ve optimal kontrol problemleri örnek verilebilir.

GA arařtırmalarının önemli bir bölümü fonksiyon optimizasyonu ile ilgilidir. Optimize edilecek amaç fonksiyonunun süreksiz olması halinde, süreksizlik noktalarında fonksiyonun türevi alınamayacağından, türev almaya dayalı optimizasyon yöntemleri kullanılamamaktadır. Oysa ki, GA, problemlerin çözümü için türev veya diđer yardımcı bilgilere gereksinim duymadığından özellikle bu tip problemlerin çözümünde geleneksel yöntemlere göre önemli bir üstünlük sağlamaktadır (Karr ve Freeman, 1999).

GA, başta üretim olmak üzere finans ve pazarlama gibi işletmelerin fonksiyonel alanlardaki birçok farklı iş probleminin çözümü için kullanılmaktadır. GA'ların özellikle, kaynak tahsisi, iş atölyesi çizelgelemesi, makine parça gruplaması ve bilgisayar ađ tasarımı gibi çeşitli alanlarda uygulamaları mevcuttur. İşletmelerdeki en yaygın kullanım alanları arasında finans, pazarlama ve üretim yer almaktadır (Emel ve Taşkın, 2002).

5.2. Genetik Algoritmaların Geleneksel Arama Metotlarından Farkı

Geleneksel optimizasyon yaklaşımından farklı olarak GA amaç fonksiyonunun gradyantlarını hesaplamaya gereksinim duymaz ve bir yerel optimuma sınırlanmaz (Goldberg, 1989). GA, optimize edilecek fonksiyonun doğrusal olması gibi analitik bir forma sahip olması ve türevlenebilmesi gibi şartları gerektirmediği için optimizasyon problemlerine de başarıyla uygulanabilmektedir.

Diđer optimizasyon yöntemlerinden farklı olarak GA'lar, arama işlemini tek bir aday çözüm ile gerçekleştirmek yerine birden çok aday çözümün oluşturduğu bir topluluk ile gerçekleştirir. Böylece çözüm uzayının birden çok başlangıç noktasıyla paralel olarak taranması sağlanmış olur. Bu özellik, GA'ların yerel optimum değerlere takılmadan, global optimum değeri bulabilmesinde en büyük etkidir. İkinci bir fark olarak GA'lar, aday çözümlerin kendisi ile değil, aday çözümlerin belirli bir alfabeğe göre kodlanmış biçimleriyle çalışmaktadır. Bu kodlama klasik GA'larda 0 ve 1'lerden oluşan bir alfabe (BIT – Binary digit) ile oluşturulmaktadır. Her ne kadar GA'larda

genel olarak ikili kodlama yaygın olsa da semboller ve gerçel sayılarla kodlama da geniş uygulama alanına sahiptir (Mitchell, 1999).

Goldberg'e göre GA'ların geleneksel arama metotlarından farkı aşağıdaki şekilde verilebilir.

- Genetik algoritmalar parametrelerin kendisi ile değil parametreler kümesinin kodlanması ile çalışır.
- Genetik algoritmalar arama aracı olarak tek bir noktayı değil, noktalar kümesini kullanır. Bu sebepten yerel en iyiye takılma riskleri düşüktür.
- Genetik algoritmalar arama sırasında probleme ilişkin türevlenebilme koşulu veya diğer ek bilgilere ihtiyaç duymaz, amaç fonksiyonunun bilinmesi genetik algoritmalar için yeterlidir.
- Genetik algoritmalar *belirli* geçiş kurallarını değil *stokastik* geçiş kuralları ile çalışır (Goldberg, 1989).

5.3. Genetik Algoritmanın Çalışma Yapısı

Basit bir GA beş temel elemandan oluşmakta ve bunların her biri algoritmanın performansını önemli derecede etkilemektedir. Bu parametreler: çözümlerin temsil şekli, başlangıç popülasyonu oluşturma yöntemi, uygunluk veya kalite değerlendirme fonksiyonu, kullanılan genetik operatörler ve kontrol parametreleridir (Karaboğa, 2011)

GA'da bir problemin optimum çözümü bazı temel işlemler ile gerçekleştirilir. Bunlar; amaç fonksiyonunun belirlenmesi, kodlama, başlangıç popülasyonunun oluşturulması, üreme, çaprazlama ve mutasyon işlemleridir.

Daha önce bahsedildiği gibi önce başlangıç popülasyonu oluşturulmakta ve sonra doğal seçme işlemi ile birlikte genetik operatörler çaprazlama ve mutasyon gelecek jenerasyondaki çözümleri üretmek amacıyla kullanılmaktadır. Kalite veya uygunluk değerlendirme işlemi tekrar üreme olayında uygulanan seçme işlemini

gerçekleştirebilmek için her bir bireye uygulanmaktadır. Birbirini takip eden jenerasyonların geliştirilmesi ve değerlendirilmesi çevrimi, optimal bir çözüm bulununcaya kadar devam etmektedir. (Karaboğa, 2011)

GA ile problemlerin çözümünde aşağıda belirtilen adımlar izlenmektedir (Yeniay, 1999).

Adım 1: GA sürecinin işletiminden önce problemin doğasına uygun olarak kodlama yapılır.

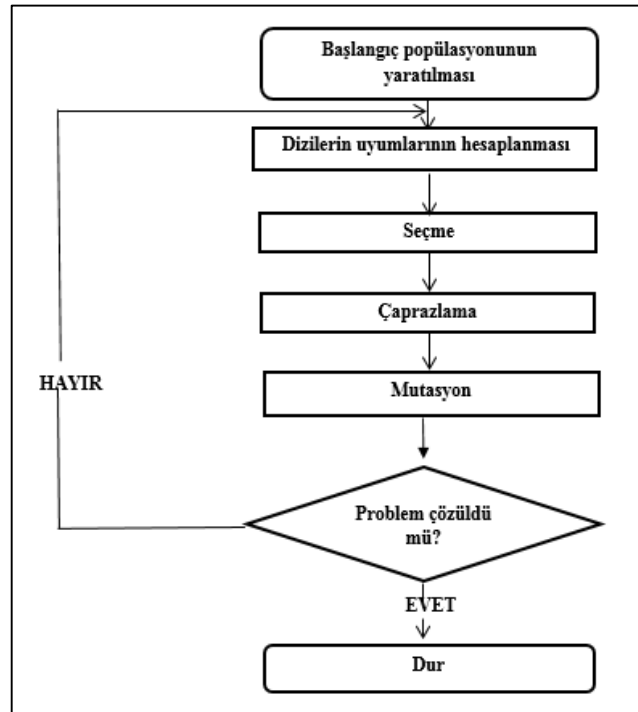
Adım 2: Genellikle rasgele olarak başlangıç popülasyonu yaratılır.

Adım 3: Başlangıç popülasyonunda yer alan her bir dizinin uyum değeri hesaplanır.

Adım 4: Popülasyonu değiştirmek ve yeni jenerasyonlar yaratmak için ise seçme, çaprazlama ve mutasyon operatörleri kullanılır.

Adım 5: Yeni popülasyona uyum değeri atanarak en iyi çözüm değeri bulunana kadar GA operasyonları devam ettirilir.

Problemlerin GA ile çözümü için yukarıda verilen adımlar, akış şeması biçiminde şu şekilde ifade edilebilir (Akyol, 2006):



Şekil 5.1. Genetik Algoritmanın İşleyiş Sürecinin Akış Diyagramı

5.3.1. Temsil mekanizması kodlama

GA'larda karmaşık yapıların temsilinde ilk ve en çok kullanılan mekanizma (0,1) alfabesini kullanan ikili bit dizisidir. Bu tekniği kullanan GA'lar ikili kodlanmış GA'lar olarak da adlandırılır. Bununla birlikte ikili sistemde uniform (tek biçimli) kodlama ve Gray kodlama gibi temsiller mümkündür (Goldberg, 1989)

GA'da ilk olarak kodlama biçimine karar verilmelidir. Çözümlerin kodlanması, probleme özgü bilgilerin GA'nın kullanacağı şekle çevrilmesidir. Tüm problemler için geçerli olabilecek genel bir kodlama tekniği yoktur. Kodlama biçimi problemlere özgü olmakla birlikte genellikle ikili kodlama, permütasyon kodlama ve gerçek değerli kodlama kullanılmaktadır. İkili kodlama 1 ve 0 değerlerinden oluşmaktadır. Problemin olası çözümleri 0 ve 1 sayıları ile kodlanır. Bu yöntemde değişkenler, değer aralığına göre belirlenen sayıda genden oluşan ikili düzende temsil edilmektedir. Fonksiyon değeri hesaplanırken ikili değerler, ondalık değere çevrilerek elde edilen değişkenin gerçek değeri, fonksiyonda yerine konulur. Permütasyon kodlama, sıralamanın önemli olduğu ve tekrarın mümkün olmadığı, araç rotalama, en kısa yol, gezgin satıcı vb. problemleri gibi genellikle birleşik optimizasyon problemlerin çözümünde kullanılmaktadır. Gerçek değerli kodlamada ise kromozomlar parametrelerin kodlanması yerine doğrudan kendi değerleriyle temsil edildikleri kodlama biçimi olarak karşımıza çıkmaktadır. Gerçek değer kodlamasında her dizi, bir değerler kümesinden oluşmaktadır. Değerler probleme göre sayı ya da karakter olabilir.

GA'nın bir problemin çözümünde kullanılabilmesi için öncelikle problemin çözümünün kromozomlarla ifade edilebilmesi gerekir. GA'nın temelinde popülasyonu oluşturan ve problemin muhtemel çözümünü içeren ve genellikle ikili sistem ile ifade edilen kromozomlar bulunmaktadır. Kromozomlar ise popülasyonu oluşturan bireyleri temsil ederler ve gen dizilerinden oluşurlar. Kromozom gösterimi parametrelerin değerlerinin kodlanmasından oluşur. Kromozomlar ikili sisteme çevrilerek çaprazlama ve mutasyon işlemlerine uygun hale getirilmektedir. Örneğin 32 sayısı ikili sayı sisteminde 100000 olarak yazılabilir. 100000 bir kromozomu gösterirken, kromozomu oluşturan her 1 ve 0 ifadesi de bir geni göstermektedir. Çizelge 5.1.'de bazı sayıların

kromozomlarla ifade edilişi gösterilmiştir.

Örneğin ikili dizi 1101010 ifadesinin onluk taban karşılığı:

$$1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 106 \text{ dır.}$$

Çizelge 5.1. Sayıların kromozomlarla ifade edilmesi

Sıra	Kromozom	Aday Çözüm
1	1000000	64
2	1101010	106
3	1111111	127

İkili kodlama

Arama uzayı $\Phi_g = \{0,1\}^L$ olarak ifade edilmek üzere L uzunluğundaki ikili kodlanmış vektör $x^g = (x_1^g x_2^g \dots x_L^g) \in \{0,1\}^L$ ile gösterilir. (Goldberg, 1989)

Bit sayısının fazla olması hassaslığın artmasına neden olurken aynı zamanda hesaplama süresinin de uzamasına sebep olmaktadır. Bu nedenle ortalama bir bit sayısının belirlenmesi için Eşitlik (5.1) kullanılabilir.

Değişim aralıkları alt_i ve $üst_i$ olarak sırasıyla i. parametrenin alt ve üst sınırlarını ifade etmek üzere (h) ondalık değeri ile i. parametreyi kodlamak için gerekli gen sayısı,

$$2^{L-1} \leq (üst_i - alt_i) \times 10^h \leq 2^L \quad (5.1)$$

koşulunu sağlayan en küçük L değeridir. Belirlenen uzunlukta ikili sistemde kodlanan parametrenin onluk sistemde belirlenen değerine göre karşılığının bulunması, dizinin deşifre edilmesidir. L uzunluğundaki ikili sistemdeki dizi, aşağıdaki şekilde deşifre edilerek gerçel sayılara dönüştürülür.

1. Parametre_{ikili} = ($g_0 g_1 g_2 \dots g_{L-2} g_{L-1}$) dizisinin onluk karşılığı,

$$\text{Parametre}_{\text{onluk}} = \sum_{i=0}^{L-1} g_i \times 2^i \text{ ile hesaplanır.} \quad (5.2)$$

2. Parametrenin deęişim aralıęındaki gerek deęeri,

$$\text{Parametre}_{\text{gerek}} = \text{alt}_i + \text{Parametre}_{\text{onluk}} \times \frac{\text{üst}_i - \text{alt}_i}{2^L - 1} \quad (5.3)$$

eşitlięi ile belirlenir (Rothlauf, 2006).

Örnek: $X_{\text{ikili}} = 10011101101011$ şeklinde kodlanan $L=14$ uzunluęundaki dizinin $[1,12]$ aralıęındaki karşıılığı,

$$X_{\text{onluk}} = 1 \times 20 + 0 \times 21 + 0 \times 22 + 1 \times 23 + 1 \times 24 + 1 \times 25 + 0 \times 26 + 1 \times 27 + 1 \times 28 + 0 \times 29 + 1 \times 210 + 0 \times 211 + 1 \times 212 + 1 \times 213 = 13753$$

ve

$$X_{\text{gerek}} = -1 + 13753 \times \frac{11}{2^{14} - 1} = 8.23$$

olarak bulunur (Karakoca, 2009).

Kodlanmış tüm parametrelerin birbiri ardına sıralanmasıyla kromozomlar elde edilir. Kromozom kodlarının özölmesiyle parametrelerin orijinal deęerleri elde edilebilir. Kod özümünde, genotipin (genotype) fenotip (fenotype) dönüşüm süreci işlemektedir. Genotip, biyolojideki tanımına paralel olarak GA'da bireyin genetik yapısını temsil eden bit dizinini yani kromozomu ifade eder. Fenotip ise, GA'lardaki özümü ifade eden erişkin bireylerdir. ok boyutlu problemlerin özümünde problem yapısına baęlı olarak ikili düzende yapılan kodlama, her zaman yeterli olmayabilir. Bu durumun en başta gelen nedeni, özüm uzayında yeterli duyarlılıęın sağlanabilmesi için ok sayıda bit'e ihtiyaç duyulmasıdır ki, bu da özüm uzayının ok büyümesi anlamına gelir. Dolayısıyla, özüm sürecinin uzamasıyla karşılaşılr. Bir dięer sebep de orijinal deęerleri birbirine yakın olmasına raęmen ikili koda evrildięinde birbirine uzak düşen deęerlerdir.

Örneęin; 30 ve 32 orijinal deęerleri birbirine yakın iki deęer olmasına karşın ikili kodlama sisteminde 30, 11110 deęerine karşılık gelirken, 32, 100000 deęerine karşılık gelmektedir. GA yapısı içerisinde 30 deęerinin 32 deęerine ilerlemesi için 0 ve

1'lerden oluşan yapıda pek çok bit'in yerinin deęismesi gerekecektir. Bu sorunu ortadan kaldıran yöntem ise, Gray kodlama yöntemidir. Gray kodu ikili sistemde kodlanmış olan diziyi alır, komşuluk özelliğine sahip yeni bir dizi oluşturmak için bu diziyi karıştırır. Böylece normalde birkaç iterasyon daha fazla süren yapı, Gray kodlarıyla daha hızlı bir biçimde işletilebilir. Şunu özellikle belirtmek gerekir ki, ikili kodlamanın bu olumsuz tarafı, bu kodlama sistematığının halen çok yaygın bir şekilde tercih edilmesinin önüne geçememiştir (Akyol, 2006).

5.3.2. Başlangıç popülasyonu

GA, çözüm döngüsüne uygun kodlanmış bireylerden oluşan bir başlangıç popülasyonu oluşturarak başlar. Başlangıç popülasyonundaki her bir kromozom, problemin olası bir çözümünü temsil eder. Popülasyon devamlı daha iyi çözümler oluşturmaya çalıştığı için, zaman içinde deęişir.

Geleneksel optimizasyon yöntemlerinde çözümün aranmasına bir noktadan başlanır ve sonuçlara göre başka noktalarla devam edilir. Ancak başlangıç noktasının seçimine göre çözümün bulunamaması veya çözüm süresinin uzaması gibi problemlerle karşılaşılabilir. GA'da ise arama işlemi tek nokta yerine potansiyel çözümlerin bir kümesi üzerinde gerçekleştirilir ve en iyi çözüme ulaşmıcaya kadar çözümler deęerlendirilir.

Başlangıç popülasyonun oluşturulması popülasyon büyüklüğünün belirlenmesi ile başlar. GA başlangıç popülasyonundan hareketle algoritma adımlarını kullanarak optimum çözüme her iterasyonda yakınsayan yeni popülasyonların oluşturulmasını sağlar. Her problem tipi için en iyi çözümün başlangıç popülasyonu sayısı farklıdır. Az sayıda belirlenen popülasyon arama uzayının etkin biçimde aranmamasına ve algoritmada kullanılacak bir çok deęerin işlenmeden atlanmasına neden olur. Çok sayıda belirlenen popülasyon ise işlem yükünü arttırıp algoritmanın yavaşlamasına neden olabilir. Bazı özel durumlar hariç literatürde başlangıç popülasyonunun rasgelelik ilkesine göre oluşturulması benimsenmiştir. Bazı çalışmalarda ise ilk popülasyonun daha önceden elde edilen bir bilgiye veya sezgiye baęlı olarak üretildięi görülmektedir.

GA sürecinde başlangıçta belirlenen popülasyon büyüklüğünün yüksek seçilmesi durumunda çözüm uzayı iyi örnekleneceğinden arama etkinliği sağlanabilecektir. Fakat hızlı bir şekilde yüksek çözüm kalitesine erişmede aynı performansı gösteremez. Popülasyon büyüklüğünün yeterince büyük seçilemediği durumlarda ise optimal çözüm uzayının saklanması problemiyle karşılaşılabilir. Yapılan çalışmalarda önerilen popülasyon büyüklüğü 50-100 arasındadır. Ancak yine yapılan çalışmalara göre, 30 diziye sahip popülasyonlar ile yüksek verimde sonuç elde edilebildiği de gözlemlenmiştir (Reeves, 2002).

5.3.3. Genetik algoritma operatörleri

GA'nın işleyiş süreci esnasında yeni popülasyonların oluşması ve çözüm uzayına yaklaşılması GA'nın üç temel parametresinden biri olan seçme, çaprazlama ve mutasyon operatörleri yardımıyla gerçekleşmektedir.

5.3.3.1. Seçme operatörleri

Popülasyondaki her bir bireye bir uygunluk değeri atanır. Bu değer bir bireyin çözüm kümesi içinde nasıl yer aldığını gösterir. Bir popülasyondaki dizilerden bir kısmı bir sonraki nesile aktarılırken bir kısmı da yok olur. Hangi dizilerin bir sonraki nesile aktarılacağı, kurulan seçim mekanizmaları ile sağlanır. Seçim işlemi, eski popülasyondaki bir kromozomu, uygunluk değerine bağlı olarak stokastik yöntemlerle yeni oluşturulacak bir popülasyon içine seçme işlemidir. Bu işlem, ilerleyen jenerasyonlarda daha yüksek uygunluk değerlerine sahip bireylerin oluşmasına neden olur. Bu nedenle bu işleme en uygunun hayatta kaldığı test (survival of the fittest test) adı verilir. Seçim yöntemleri, genellikle en iyi olan yaşar mantığına göre oluşturulur. Amaç, yeni jenerasyonda daha yüksek uygunluğa sahip bireylerin sayısını arttırmaktır. En çok kullanılan seçme operatörleri rulet tekerleği seçimi, turnuva seçimi ve sıralama seçimidir.

Rulet tekerleđi seçim yöntemi

Yöntem, genel olarak rulet oyununa benzetilir. Bu yöntemde bireyin beklenen seçim olasılıđını, bireyin uygunluk deđerinin jenerasyonun toplam uygunluk deđerine bölümü belirler. Her birey rulet tekerleđinde uygunluk deđeri kadar yer kaplar ve jenerasyondaki bireylerin toplam uygunluk deđerini rulet tekerleđinin büyüklüğünü verir. Yani her dizi uyum deđeriyle orantılı bir olasılık deđeri ile seçilmektedir.

Bu yöntemde seçim işlemi yapılırken çember, popülasyon büyüklüğü kadar çevrilir. Her defasında [0, 1] aralıđından rasgele sayı üretilerek, bu sayının dahil olduđu ilk aralıktaki birey sonraki nesil için aday olarak seçilir (Goldberg, 1989).

Bu yöntem iki aşamada gerçekleşir:

-Rulet tekerleđi popülasyon büyüklüğü kadar çevrilir.

-Her çevrim işleminde rulet işaretinin gösterdiđi birey bir sonraki jenerasyonu oluşturmak için seçilir. Daha iyi uyum deđerine sahip bireyler tekerlekte daha fazla yer aldıklarından seçim şansları daha yüksek olur. Basit ifadeyle seçim olasılıđı, kromozomun deđerinin popülasyondaki bütün kromozomların deđerlerinin toplamına oranıdır.

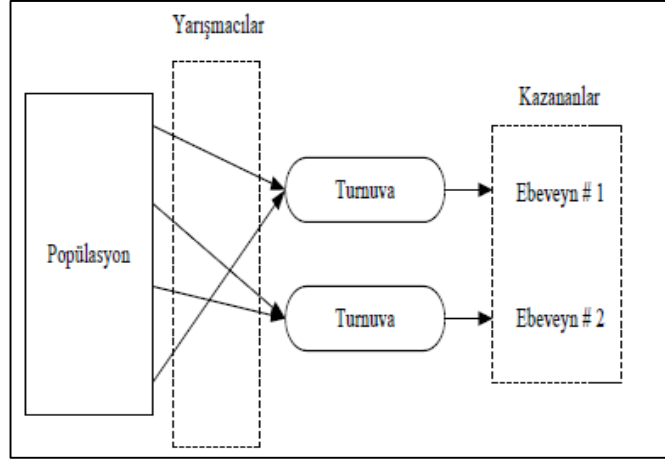
$$P_i = \frac{f_i}{\sum_{i=1}^n f_i} \quad (5.4)$$

Eşitlik (5.4)'de n popülasyon büyüklüğünü, P(i), i. kromozomun seçim olasılıđını, i, popülasyonun kaç bireyden oluştuđu, f(i) ise i. kromozomun uygunluk deđerini ifade etmektedir.

Turnuva seçim yöntemi

Bu seçim yönteminde, bireyler rastgele olarak gruplanır ve grupta bireyler seçim işlemi için rekabete girer. Grup içinde, en yüksek uygunluk deđerine sahip olan birey, bir sonraki jenerasyonda popülasyonu oluşturmak için seçilmiş olur. Yani yapılan

turnuvada hangi birey daha iyiye o kazanır. Bu işlem, toplam popülasyon sayısı oluşturuluncaya kadar devam eder. Bu seçim yönteminde grup büyüklüğü önemlidir ve seçim yönteminin performansını önemli ölçüde etkiler. Bazı uygulamalarda grup büyüklüğü 2 olarak seçilirken bazı uygulamalarda çok daha büyük gruplar oluşturulur.



Şekil 5.2. Turnuva Seçimi (Öztürk ve Çelik, 2012)

Sıralı seçim yöntemi

Bu yöntemde bireyler uygunluk değerlerine göre sıralanır. Bireyin seçim şansı, uygunluk değerinden ziyade oluşturulan liste içindeki yerine bağlıdır. Bu stratejide,

$S(\theta_i, t)$: t 'inci popülasyondaki θ_i bireyin uygunluk değeri,

$n_i(t)$: $S(\theta_i, t)$ ' ler büyükten küçüğe doğru sıralandığında $S(\theta_i, t)$ ' nin sıra sayısı,

N : Popülasyondaki birey sayısı olmak üzere,

θ_i 'nin sonraki nesile ebeveyn olarak seçilme olasılığı,

$$r(\theta_i, t) = \frac{2n_i(t)}{N(N + 1)} \quad (5.5)$$

biçiminde belirlenir (Altunkaynak ve Esin, 2004).

Seçme işleminin tamamlanmasının ardından yeni popülasyonların oluşturulabilmesi için seçilmiş kromozomlar çaprazlanır ve yeni bireylerin oluşması sağlanır.

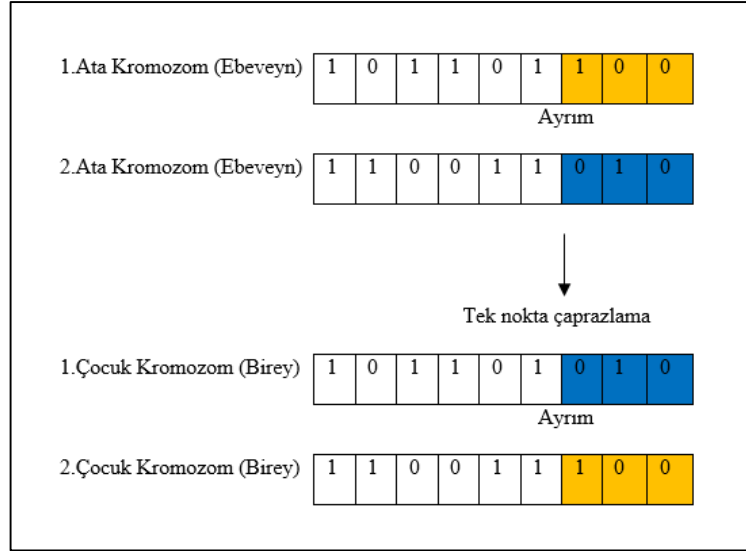
5.3.3.2. *Çaprazlama Operatörleri*

Çaprazlama, GA'da en önemli operatördür ve yeni çözümlerinin üretiminden sorumludur. Çoğunlukla rastgele olarak seçilen iki bireyin kromozomları çaprazlanarak yer değiştirmesiyle gerçekleştirilir. Bu işlemde, bireylerin kromozomunu oluşturan dizilerin farklı kısımları yer değiştirerek yeni bireyin üretimi sağlanır. Bu şekilde yeni oluşturulan birey, popülasyonda daha az uygunluk değerine sahip bireylerin yerine konularak çözüm için zayıf bireyler yerine güçlü bireylerin getirilmesi sağlanır.

Çaprazlama operatörü, seçim işleminden sonra uygulanır ve basit ifadeyle yeni kromozomlar elde etmek için eşit uzunluktaki iki kromozomdan seçilen belirli bir noktanın yer değiştirmesidir. GA'da genel olarak, çaprazlama nokta sayısı tektir ve şema teoremi bu noktadan çaprazlama operatörüne göre oluşturulmuştur. Ancak çaprazlama, iki veya daha fazla nokta seçimi ile de yapılabilir. Tek noktalı çaprazlama çok yaygın olarak kullanılan bir çaprazlama tekniğidir. Bu teknik, iki kromozomda rastgele seçilen bir ayırma noktası ile iki kromozomun ilgili kısımlarının yerini değiştirir. Çaprazlama işlemi seçilen iki dizinin ortasından rastgele alınan bir noktadan karşılıklı olarak çaprazlama yapılarak gerçekleştirilir. Uygun bir çaprazlama noktası seçildiğinde bireylerden uyum değeri daha yüksek diziler elde edilebilmektedir. Fakat uygun çaprazlama noktası genellikle bilinmediğinden rastgele bir nokta seçilmektedir.

Tek nokta çaprazlama

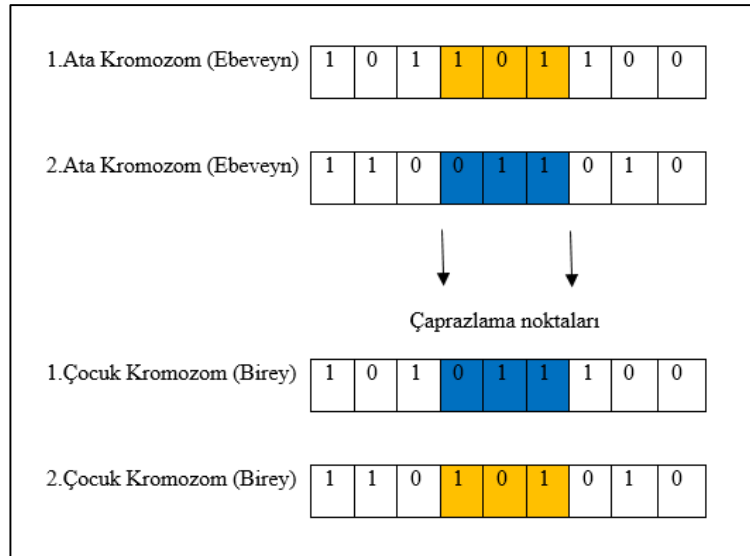
Tek nokta çaprazlamada kromozom yapısı üzerinde rastgele bir ayırma noktası belirlenir. Bu ayırma noktası öncesindeki gen yapısı aynen korunurken, ayırma noktası sonrasındaki gen yapısı karşılıklı olarak çaprazlanarak değiştirilir. Aşağıdaki örnekte ayırma noktası sonrasındaki (1,0,0) ve (0,1,0) çaprazlanarak yer değiştirmiştir.



Şekil 5.3. Tek Nokta Çaprazlama Örneği

İki nokta çaprazlama

Bazı uygulamalarda iki ayırım noktası kullanılmaktadır. Ayırım noktaları tek nokta çaprazlamada olduğu gibi rastgele yapılmaktadır. Aşağıdaki örnekte iki ayırım noktası belirlenerek noktalar arasında kalan (1,0,1) ve (0,1,1) genleri yer değiştirmiştir.



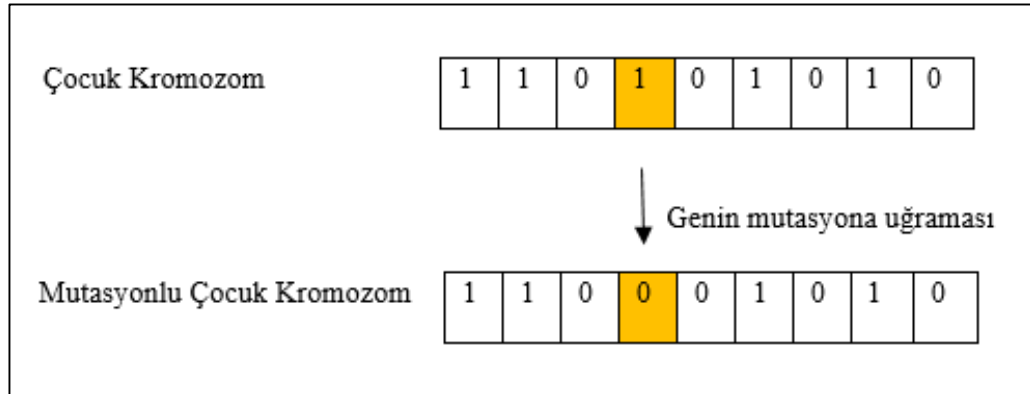
Şekil 5.4. Çift Nokta Çaprazlama Örneği

Mutasyon

Mutasyon, bireyin kromozomunu oluşturan bir dizi içindeki bir veya daha fazla Gen'in mutasyon olasılığına bağlı olarak rastgele değiştirilmesidir. Mutasyon yeni oluşan çözümlerin birbirini kopyalamasını önleyerek sonuca daha hızlı ulaşılmasını sağlayan operatördür. Mutasyon operatörü çok küçük olasılıkta kullanılan ve çözümün yerel optimal noktalara takılmasını önleyen önemli bir operatördür. Bu işlem ile yeni kromozomlar üretilerek popülasyonun optimal noktalara yakalanma olasılığı engellenir. Buna rağmen mutasyon sonucu bazen iyi bireylerin yok olmasına sebep olunabilir. Bunu engellemek için GA içerisinde en iyi bireylerin bir sonraki nesile aktarılmasını sağlayacak bir yapı oluşturulmuştur. İkili kodlamada örneğin 1' den 0'a bir değişimdir.

Çaprazlama işleminin bireylerin ne kadarına uygulanması gerektiği çaprazlama yüzdesi (Pc) ile, mutasyon işleminin ne sıklıkla gerçekleştirileceği ise mutasyon yüzdesi/olasılığı ile belirlenir. Mutasyon genellikle çaprazlamaya göre daha az olasılıkta kullanılır. Bunun nedeni çaprazlama sonucu elde edilen uyum değeri yüksek dizileri kaybetmemektir. Çaprazlama olasılığı, genellikle 0.25 ile 1 arasında, mutasyon olasılığı 0.01 ile 0.001 arasında seçilir.

Şekil 5.5.'de görüldüğü üzere, bir genin mutasyona uğraması sonucu, genin değeri 1'den 0'a doğru değişmiştir.



Şekil 5.5. Mutasyon Örneği

5.3.4. Şema teoremi

GA'nın teorik temelleri J.Holland tarafından tanımlanan Şema (Schema) teoremi ile verilmiştir. Holland, geliştirmiş olduğu GA'nın teorik analizini göstermek amacıyla şema kavramını kullanmıştır. Şema teoremine göre her bir kromozom belirli bir şemanın örneğidir.

$V = \{0,1,*\}$ alfabeti ile ifade edilmektedir.

Her kromozom 0 ve 1'lerin bir vektörü olarak ele alınarak ikili kodlamada kullanılan $\{0,1\}$ elamanlarına $\{*\}$ sembolü eklenir. * sembolü, bulunduğu konumdaki değer $\{0\}$ veya $\{1\}$ olmasının önemsiz olduğunu gösterir. Örneğin,

$V = (1\ 1\ *\ 0\ 1\ 0)$ bir şemayı göstermek üzere A_1 ve A_2 , V şemasının kromozomlarıdır.

$A_1 = (1\ 1\ 1\ 0\ 1\ 0)$

$A_2 = (1\ 1\ 0\ 0\ 1\ 0)$

Şema içerisindeki * sembolünün sayısı t olmak üzere en fazla 2^t tane kromozom belirtilebilir. Bununla beraber k uzunluğundaki her bir kromozom 2^k tane şema tarafından belirtilebilir.

Şema yapısı $(*1*011)$ olmak üzere, bu şemaya uygun aşağıdaki ikili diziler yazılabilir.

010011, 011011, 110011, 111011

Burada $t=2$ olmak üzere $2^2=4$ adet kromozom oluşur.

Bir H şemasının derecesi, pozisyonları sabit olan bit'lerin sayısıdır ve $O(H)$ ile gösterilir. $(*1*011)$ şema yapısının derecesi, 4'tür. Derece, bir şemanın belirginliğini gösterir. Farklı derecelere sahip H şemaları şöyle verilebilir:

$H_1 = (**000*111)$

$H_2 = (**11*1****)$

$H_3 = (0101*1*1**)$

Buna göre, $O(H_1)=6$, $O(H_2)=3$, $O(H_3)=6$ derecelerinde olacaktır. Burada en belirgin nitelikteki şema ise H_3 şemasıdır. Bir şemanın derecesi mutasyona rağmen şemanın yaşama olasılığını hesaplarken faydalı olmaktadır (Michalewicz, 1996).

Bir şemanın tanımlanan uzunluğu, şemanın tanımlanmış ilk ve son bit'lerinin pozisyonlarının farkı alınarak hesaplanır ve $\delta(H)$ ile gösterilir. Örneğin H_1 şemasının ilk olarak 4. ve son olarak 10. bitleri tanımlandığından $\delta(H_1) = 10 - 4 = 6$ olarak hesaplanır. Benzer şekilde $\delta(H_2) = 3$ ve $\delta(H_3) = 7$ 'dir.

Diğer önemli bir parametre uygunluk oranıdır. Bu, şemanın ortalama uygunluk değerinin popülasyonun ortalama uygunluk değerine oranı olarak tanımlanır. (Reeves, 2002; Goldberg, 1989)

Uygunluk değeriyle orantılı bir üreme işlemi kullanıldığında S şemasının $t+1$ anındaki beklenen sayısı,

$$E(S, t + 1) = f(S, t)N(S, t) \quad (5.6)$$

ifadesi ile tanımlanabilir. Burada $f(S, t)$, S şemasının uygunluk oranı, $N(S, t)$ ise t anındaki S 'nin eleman sayısıdır.

Eğer, $l(S)$ uzunluklu S şemasına P_c çaprazlama oranıyla t anında çaprazlama işlemi uygulanırsa; bu durumda $t+1$ anında S 'nin popülasyonda temsil edilme olasılığı, yani çaprazlama işlemi tarafından tahrip edilmeme ihtimali aşağıdaki biçimde tanımlanabilir.

$$P[S, t + 1] \geq 1 - \frac{P_c l(S)}{n - 1} (1 - P[S, t]) \quad (5.7)$$

Burada n , kromozom uzunluğunu ve $(1 - P[S, t])$ ikinci ebeveynin farklı bir şemanın elemanı olma ihtimalini temsil etmektedir. Bu ifade alt sınırı tanımlamaktadır. Çünkü S , çaprazlama işlemi aracılığıyla farklı şemalardan eleman kazanabilir.

Eğer, $k(S)$ dereceli S şemasına P_m mutasyon oranıyla t anında mutasyon işlemi uygulanırsa; $t+1$ anında popülasyonda S 'nin temsil edilme ihtimali,

$$P[S, t + 1] \geq 1 - P_m k(S) \quad (5.8)$$

ile tanımlanır.

Bu sonuçları birleştirmek suretiyle aşağıdaki teorem elde edilir. Yukarıda tanımlanan üreme planı kullanıldığında, çaprazlama ve mutasyon oranları sırasıyla P_c ve P_m , S şemasının derecesi $k(S)$, uzunluğu $l(S)$ ve uygunluk oranı $f(S, t)$ olmak üzere $t+1$ anında S şemasının popülasyondaki temsilcilerinin beklenen sayısı için,

$$E[S, t + 1] \geq \left\{ 1 - \frac{P_c l(S)}{n - 1} (1 - P[S, t]) - P_m k(S) \right\} f(S, t) N(S, t) \quad (5.9)$$

yazılabilir. Böylece, kısa ve düşük dereceli şemalar eğer uygunluk oranları 1'den fazla ise eleman sayılarını artıracak, daha uzun ve daha yüksek dereceden şemalar ise daha fazla uğraşmak zorunda kalacaktır. Bir GA için ideal durumda, kısa ve düşük dereceli şemaların iyi ve daha iyi çözümlerini oluşturmak için birleşmeleri gerekmektedir. Bu fikrin çalışacağı kabulü yapı-bloğu (building- block) hipotezi olarak adlandırılır (Goldberg, 1989; Karaboğa, 2011).

5.4. Basit Bir Genetik Algoritma İşleyişi

[0-31] tamsayı aralığında tanımlanan $f(x) = x^2$ fonksiyonunun maksimum değerinin GA ile bulunmasını gösteren adımlar aşağıda sıralanmıştır.

1.Adım Temsil Mekanizması Kodlama (İkili Kodlama):

0 degeri, uzunluğu 5 bit ikili tamsayı olan “00000” ile, 31 degeri ise “11111” ile kodlanır.

2.Adım Başlangıç Popülasyonunun Oluşturulması:

Rassal olarak popülasyondaki bireylerin sayısı $n = 4$ (4 boyutlu popülasyon), değerleri ise yine rassal olarak aşağıdaki gibi belirlenmiştir.

01100, 11001, 00101, 10011

3.Adım Uygunluk Değerlerinin Hesaplanması:

Uygunluk değeri bireylerin amaç fonksiyonundaki değerleridir. Uygunluk Değerleri için oluşturulan çizelge aşağıdaki gibidir.

Çizelge 5.2. Uygunluk değerlerinin hesaplanması

Birey No	Rastgele Seçilen Başlangıç Popülasyonundaki kromozomlar	X değeri	Uygunluk Değeri $f(x)=x^2$	Olasılık Yüzdesi %
1	01100	12	144	12.47
2	11001	25	625	54.11
3	00101	5	25	02.16
4	10011	19	361	31.26

4.Adım Genetik Operatörler (seçim-çaprazlama-mutasyon) Seçimi:

Seçim: Seçim işlemi için bu örnekte rulet tekerleği kullanılmıştır. Bu değerler için bir rulet çarkı oluşturulur. Rulet çarkında bireyler toplam içindeki yüzdeleri ile orantılı olarak bölünür ve çark popülasyon sayısı kadar çevrilir. Belli bir noktada gelen değerler yeni popülasyonları oluşturur. Rulet çarkında gelen değerler ve yeni popülasyon için oluşturulan çizelge aşağıdaki gibidir.

Çizelge 5.3. Seçim değerlerinin hesaplanması

Birey No	Rastgele Seçilen Başlangıç Popülasyonundaki kromozomlar	X değeri	Uygunluk Değeri $f(x)=x^2$	Olasılık Yüzdesi % (seçim değeri)	Beklenen Geliş Sayısı	Rulette Geliş Sayısı
				$\frac{f_i}{\sum f}$	$\frac{f_i}{\bar{f}}$	
1	01100	12	144	12.47	0.4987	1
2	11001	25	625	54.11	2.1645	2
3	00101	5	25	02.16	0.0866	0
4	10011	19	361	31.26	1.2502	1
Toplam			1155	100	4.0	4
Ortalama			288.75	25	1.0	1
Maksimum			625	54.11	2.1645	2

Burada çark dört kere çevrilmiş ve iki kere “25” değerinin geldiği gözlenmiştir. Ortalama uygunluk degeri 288.75 ve gelen en büyük deger 625’tir.

Çaprazlama: Çaprazlama sonucu elde edilen sonuçlar Çizelge 5.4’te özetlenmiştir. İlk döngüde çaprazlama sonrası toplam uygunluk degeri 1155’ten 1763’e, ortalama uygunluk degeri 288.75’ten 440.75’e ve maksimum deger 625’ten 729’a çıkmıştır.

Çizelge 5.4. Çaprazlama değerlerinin hesaplanması

Birey No	Çiftleşme sonrası oluşan popülasyondaki kromozomlar	Eşler	Çaprazlama Noktası	Yeni popülasyondaki kromozomlar	X deęeri	Uygunluk Deęeri $f(x)=x^2$
1	0110-0	2	4	01101	13	169
2	1100-1	1	4	11000	24	576
3	001-01	4	2	11011	27	729
4	100-11	3	2	10001	17	289
Toplam						1763
Ortalama						440.75
Maksimum						729

Mutasyon: Mutasyon sonucu oluşturulan yeni popülasyon ve yeni iterasyon deęerlerini gösteren çizelgeler ařaęıdaki gibidir.

Çizelge 5.5. Mutasyon deęerlerinin hesaplanması

Birey No	Çiftleşme sonrası oluşan popülasyondaki kromozomlar	Mutasyon Şekli	Mutasyon Sonrası yeni popülasyondaki kromozomlar	X deęeri	Uygunluk Deęeri $f(x)=x^2$
1	01100	0 → 1	11101	29	841
2	11001		11000	24	576
3	00101		11011	27	729
4	10011	0 → 1	10101	20	400
Toplam					2546
Ortalama					636.5
Maksimum					841

Çizelge 5.6. Yeni iterasyon değerlerinin hesaplanması

Rastgele Seçilen Başlangıç Popülasyonundaki kromozomlar	X değeri	Mutasyon Sonrası yeni popülasyonundaki kromozomlar	X değeri	Yeni İterasyondaki kromozomlar	X değeri
01100	12	11101	29	11101	29
11001	25	11000	24	11000	24
00101	5	11011	27	11011	27
10011	19	10101	20	11001	25

Bu örneğin çözümünde tek bir iterasyon yapılmış ve başlangıç popülasyonundan seçim, çaprazlama ve mutasyon işlemlerinden sonra bir sonraki nesil oluşturulmuştur. Bu adımlara en iyi sonuç bulununcaya kadar yani durdurma kriteri sağlanıncaya kadar devam edilir. Durdurma kriteri iterasyon sayısı olabileceği gibi hedeflenen uygunluk değeri de olabilmektedir.

6. UYGULAMA

Çalışmanın uygulama bölümünün ilk kısmının amacı gerçek bir veri kümesi üzerinden iki düzeyli bir bağımlı değişken modelinde parametre tahminlerinin hem klasik NR hem de GA yöntemleri ile tahmin edilerek farklı iterasyon sayılarında sonuçların özdeş olduğunu ve dolayısıyla literatürde var olan klasik yaklaşımların yerine GA'nın kullanılmasının daha da avantajlı olabileceğini ispatlamaktır.

Çalışmanın uygulama bölümünün ikinci kısmında ise NR, NM Algoritması ve GA parametre tahminlerinin karşılaştırılarak, GA'nın parametre tahmininde yakınsama hızı ve optimum sonuç bakımından hangi koşullarda en iyi sonuca ulaştığını bulmak amaçlanmıştır. Bu amaçla rastgele üretilen veriler ile parametre tahmininde en uygun başlangıç popülasyonu, seçim, çaprazlama, mutasyon operatörleri ve çaprazlama, mutasyon olasılıkları belirlenmiştir.

6.1. Gerçek Veriler İle Uygulama

Çalışmanın uygulama bölümünün amacı gerçek bir veri kümesi üzerinden iki düzeyli bir bağımlı değişken modelinde parametre tahminlerinin hem klasik NR hem de GA yöntemleri ile tahmin edilerek farklı iterasyon sayılarında sonuçların özdeş olduğunu ve dolayısıyla literatürde var olan klasik yaklaşımların yerine GA'nın kullanılmasının daha da avantajlı olabileceğini ispatlamaktır.

6.1.1. Uygulamada kullanılan veriler

Bu çalışmada kullanılan veri kümesi, Hitit Üniversitesi Çorum Eğitim ve Araştırma Hastanesinin Cildiye bölümünden alınmıştır. Saç dökülmesi sorunu ile Cildiye

bölümüne 2013 yılında başvuran hastalardan Alopesi (Saçkıran) tanısı konulan 52 hasta deney grubu olarak, alopesi hastalığı bulunmayan 43 hasta ise kontrol grubu olarak belirlenmiştir. Araştırma toplam 95 hastanın verilerinden oluşmaktadır. Çalışmaya katılan 95 hastanın %55'i deney grubunu %45'i kontrol grubunu oluşturmaktadır.

Çalışmada incelenen iki düzeyli bağımlı değişken, Alopesi Tanısı'dır. Tanı konulan hastalar "1", konulmayan hastalar ise "0" olarak kodlanmıştır. Alopesi tanısı konulmasını etkileyen değişkenler ve değişkenler ile ilgili yapılan kısaltmalar Çizelge 6.1'de verilmiştir.

Çizelge 6.1. Analizde kullanılan değişkenler

Değişkenler		Özellikler
Y	Alopesi (Saçkıran) Hastalığı	Nitel (Kategorik) 1 = Hastalık var 0 = Hastalık yok
X ₁	Yaş	Nicel
X ₂	Cinsiyet	Nitel (Kategorik) 1 = Erkek 0 = Kadın
X ₃	ALT (karaciğer hücrelerinde bulunan enzimdir)	Nicel
X ₄	AST (karaciğer hücrelerinde bulunan enzimdir)	Nicel
X ₅	Glukoz	Nicel
X ₆	HDL (High density lipoprotein) kolesterol (iyi huylu)	Nicel
X ₇	LDL (Low-density lipoprotein) kolesterol (kötü huylu) (Karaciğerde üretilen ve kolestrolü kan yoluyla taşıyan moleküler proteinler)	Nicel
X ₈	Kolesterol (dolaşım sisteminde bulunan bir lipiddir (yağlı bir madde)	Nicel
X ₉	Serbest t3 (tiroit bezi hormonları)	Nicel
X ₁₀	Serbest t4 (tiroit bezi hormonları)	Nicel
X ₁₁	TSH (tiroid uyarıcı hormon)	Nicel
X ₁₂	Trigliserit (Tg) (yağların ana bileşeni)	Nicel

“Cinsiyet” açıklayıcı değişkeni kategorik bir yapıya sahip olduğu için Erkek “1” ve Kadın “0” olarak kodlanmıştır. Erkek ve kadın hastaların oranı sırasıyla %60 (n=57) ve % 40 (n=38)'dir.

6.1.2. LR analizi ve Newton-Raphson yöntemi ile parametre tahmini

Daha önce de vurgulandığı gibi çalışmanın uygulama kısmının asıl amacı kurulan LR modeli sonucu alopesi hastalığına etki edebileceği düşünülen tüm aday açıklayıcı değişkenler için NR algoritması kullanılarak tahmin edilmiş olan parametreleri, GA ile bulmaya çalışmak ve dolayısıyla GA'nın bağımlı değişkenin kategorik olduğu modellerde kullanılabilirliğini ölçmektir. Bunun için öncelikle hastalığa etki edebileceği düşünülen önemli risk faktörleri belirlenmiştir. Hastaların laboratuvar test sonuçları ve kategorik bir değişken olan cinsiyet, bağımsız değişkenler olarak alınmıştır. Parametreler öncelikle NR yöntemi ile tahmin edilmiştir. Parametre tahminleri ve diğer LR analiz sonuçları Çizelge 6.2'de verilmiştir.

Çizelge 6.2. LR modelinde Newton Raphson algoritması ile elde edilen sonuçlar

	β	Standart Hata	Wald Testi	Serbestlik Derecesi	Sig.	Exp (β)
Constant	2.985	3.165	0.889	1	0.346	19.782
Yaş	-0.025	0.024	1.029	1	0.310	0.976
Cinsiyet	1.618	0.619	6.840	1	0.009	5.044
ALT	-0.051	0.038	1.761	1	0.185	0.951
AST	0.135	0.059	5.237	1	0.022	1.145
Glukoz	-0.004	0.010	0.162	1	0.687	0.996
HDL	-0.012	0.045	0.070	1	0.791	0.988
Kolesterol	0.011	0.038	0.083	1	0.774	1.011
LDL	-0.032	0.037	0.750	1	0.387	0.968
Serbest t3	-1.611	0.777	4.306	1	0.038	0.200
Serbest t4	2.357	1.056	4.977	1	0.026	10.557
TSH	-0.039	0.240	0.026	1	0.871	0.962
Trigliserit	0.006	0.008	0.423	1	0.515	1.006

6.1.3. Genetik algoritma ile parametre tahmini

GA ile parametre tahmin değerleri Matlab paket programı kullanılarak elde edilmiştir. GA ile amaç fonksiyonunu optimize etmenin iki yolu vardır. Birincisi kod yazmayı gerektirmeyen, "Optimization Toolbox" kullanmaktır. İkinci yol ise Matlab'de "command window" bölümüne gerekli kodları yazarak sonuca ulaşmaktır. Parametre tahmininde kullandığımız aşağıda tanımlanan olabilirlik fonksiyonunu en çok yapan

β parametreleri, optimizasyon sonuç değerleri, kodlar ve işlevleri özet olarak aşağıda verilmiştir.

Uygulamada kullanılan hastalara ait bağımsız değişken verileri Eşitlik (4.5)'te yerine yazılarak aşağıda verilen log-olabilirlik fonksiyonuna ulaşılmıştır.

$$f(\beta_0, \beta_1, \beta_2, \dots, \beta_{12}) = 52*x(1)+1481*x(2)+37*x(3)+1167*x(4)+1189*x(5)+4899*x(6)+2255.4*x(7)+8430*x(8)+4826.07*x(9)+178.45*x(10)+55.28*x(11)+103.67*x(12)+6695*x(13)-\log(1+\exp(x(1)+18*x(2)+x(3)+20*x(4)+17*x(5)+100*x(6)+39.1*x(7)+199*x(8)+76.7*x(9)+4.57*x(10)+1.31*x(11)+2.5*x(12)+416*x(13)))-\log(1+\exp(x(1)+23*x(2)+14*x(4)+30*x(5)+92*x(6)+39.2*x(7)+143*x(8)+93.19*x(9)+2.97*x(10)+0.79*x(11)+0.94*x(12)+53*x(13)))-\dots-\log(1+\exp(x(1)+25*x(2)+10.6*x(4)+16.5*x(5)+88*x(6)+41.3*x(7)+140*x(8)+89.09*x(9)+3.19*x(10)+0.62*x(11)+2.52*x(12)+136.7*x(13)))-\log(1+\exp(x(1)+25*x(2)+13*x(4)+17*x(5)+81*x(6)+69.7*x(7)+202*x(8)+118.7*x(9)+3.01*x(10)+0.86*x(11)+0.19*x(12)+68*x(13)))$$

Ancak program maksimizasyon üzerine değil minimizasyon üzerine kurulu olduğundan dolayı $f(\beta_0, \beta_1, \beta_2, \dots, \beta_{12})$ fonksiyonu (-1) ile çarpılarak bu fonksiyonu minimize eden $\beta_0, \beta_1, \beta_2, \dots, \beta_{12}$ parametrelerinin tahminleri bulunacaktır.

Matlab Paket programında en basit haliyle GA optimizasyon kodu,

$$[x,fval] = ga(@olabilirlik,13,options)$$

şeklinindedir. Burada eşitliğin sağ tarafı girdileri sol tarafı ise çıktıları oluşturmaktadır. Girdiler, fonksiyonun tanımlanması, değişken sayısının girilmesi, çözüm araması hangi operatörler ile yapılacaksa bunların tanımlanmasından oluşmaktadır. Ayrıca özel olarak başlangıç popülasyonu, seçim tekniği, mutasyon ve çaprazlama oranları, iterasyonların tüm durdurma kriterleri bu bölümdeki 'options' kodu ile yapılmaktadır. Çıktı fonksiyonları ise çözüm sonrasında yazdırılması gereken sonuçları belirtmektedir. 'x' fonksiyonu optimize edilen bağımsız değişkenlerin sonucunu, 'fval' fonksiyonun optimum değerini yazdırmaktadır. Buraya yazılacak ek kodlar ile daha fazla çıktı değerinin yazdırılması sağlanabilmektedir.

Çalışmada maksimize edilecek log olabilirlik fonksiyonu ile ilgili gerekli GA kodları ve tanımları Ek E’de detaylı olarak verilmiştir.

Yukarıda açıklanan tüm bilgiler sonucu GA ve NR Algoritması ile elde edilen $\beta_0, \beta_1, \beta_2, \dots, \beta_{12}$ parametreleri için tahmin değerleri ve log-olabilirlik fonksiyonunun negatifinin minimum değerleri Çizelge 6.3’de verilmiştir.

Çizelge 6.3 incelendiğinde, optimum değerler arasındaki fark yaklaşık olarak 0.0498 bulunmuştur. GA ile parametre tahmininde tablodaki 52.833533819034635 optimizasyon sonuç değerini bulabilmek için; population size ‘50’; selection function ‘roulette’; reproduction crossover function ‘0.95’; mutation function ‘gaussian’; mutation scale ‘0.15’; mutation shrink ‘0.88’; crossover function ‘intermediate’; crossover ratio ‘1.0’; durdurma kriterleri ise generations ‘50000’; stall generations ‘50000’; function tolerance ‘1e-6’; nonlinear constraint tolerance ‘1e-6’ olarak seçilmiştir. Diğer operatör ve oranlar ise varsayılan olarak işlem görmüştür.

Çizelge 6.3. Parametre tahminlerinin karşılaştırılması

Parameter Estimations	Logistic Regression Analysis	
	Newton-Raphson	Genetic Algorithm
$\hat{\beta}_0$ (constant)	2.985	1.984
$\hat{\beta}_1$	-0.025	-0.022
$\hat{\beta}_2$	1.618	1.548
$\hat{\beta}_3$	-0.051	-0.049
$\hat{\beta}_5$	-0.004	-0.004
$\hat{\beta}_6$	-0.012	-0.01
$\hat{\beta}_7$	0.011	0.012
$\hat{\beta}_8$	-0.032	-0.032
$\hat{\beta}_9$	-1.611	-1.424
$\hat{\beta}_{10}$	2.357	2.283
$\hat{\beta}_{11}$	-0.039	-0.024
$\hat{\beta}_{12}$	0.006	0.005
Log-olabilirlik fonksiyonunun negatifinin minimize değeri	52.8833	52.833533819034635

Çalışmanın başında da belirttiğimiz gibi amaç, bağımlı değişkenin kategorik olduğu modellerde parametre tahmini için kısıtlayıcı varsayımlar gerektiren klasik NR algoritmasına alternatif bir algoritma önerebilmektir. Bu amaçla GA, daha esnek

varsayımlar gerektirmesi ve yukarıda bahsedilen birçok avantajından dolayı iyi bir optimizasyon yöntemi olarak düşünülmüş ve klasik NR algoritması ile optimize edilen log-olabilirlik fonksiyonunun GA ile optimize edilmesi durumunda nasıl bir sonuç vereceği araştırılmak istenmiştir.

Her iki yöntemle de fonksiyonun negatifinin minimum değeri birbirine çok yakındır. Ancak optimizasyon problemlerinde çok küçük bir fark bile çok önemli olabilmekte; bu durum dikkate alındığında ise GA'nın bu örnek için NR algoritmasına göre daha iyi sonuç verdiği gözlenmiştir. β parametrelerinin tahminleri incelendiğinde ise yine birbirine çok yakın değerler bulunmuştur. En fazla farklılığın β_0 (sabit) parametre tahmininde olduğu görülmektedir.

Bu sonuçlar ile araştırmalarda kullanılacak verilerin özelliklerine göre bazı analizlerde GA ile farklı operatör ve oranlar kullanılarak NR algoritmasının bulunduğu parametre tahmin değerlerinden daha iyilerinin bulunabileceği ortaya çıkmıştır. Böylece çalışmamızın amacına ulaştığı anlaşılmakta ve NR algoritmasına çok yakın hatta daha iyi sonuç veren GA'nın da kategorik bağımlı değişkenlerin parametre tahminlerinde kullanılmasının çok iyi bir alternatif olduğu söylenebilmektedir.

6.2. Rastgele Üretilen Veriler İle Uygulama

Bu çalışmanın uygulama bölümünde kullanılan veri kümesi, bir bağımlı değişken ve dört bağımsız değişkenden oluşmaktadır. Matlab Paket Programı ile rastgele üretilen iki düzeyli kategorik yapıdaki bağımlı değişken Y, Bernoulli dağılımından, bağımsız değişkenlerden üç tanesi (X_1, X_2, X_3) yine Bernoulli dağılımından, X_4 ise Normal dağılımdan gelmektedir. Toplamda her bir değişkenden 50 veri üretilmiştir. LR analizinde genellikle bir özelliğin varlığı "1", yokluğu ise "0" olarak kodlanmaktadır. Rasgele üretilen değişken değerleri EK C'de verilmiştir.

Uygulamada kullanılan açıklayıcı değişken değerleri kullanılarak EK D'de verilen log-olabilirlik fonksiyonu elde edilir. Fonksiyonu maksimum yapan değerlerin elde

edilmeye çalışıldığı bu optimizasyon sürecinde kullanılan Matlab programı minimizasyon üzerine kuruludur. Bu nedenle fonksiyon (-1) ile çarpılarak bu fonksiyonu minimum yapan (tersini maksimum yapan) $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$ parametrelerinin tahminleri bulunacaktır.

6.2.1. LR analizi ve NR yöntemi ile parametre tahmini

NR yöntemini kullanılarak elde edilen parametre tahminleri ve diğer LR analiz sonuçları aşağıdaki verilmiştir. Çizelge incelendiğinde sadece X_2 değişkeninin

anlamlılık sınırına yakın olduğu görülmüştür. Ancak burada hangi bağımsız değişkenlerin bağımlı değişkenleri açıkladığı ile ilgilenilmediği, sadece parametre tahminlerinin değerleri ile ilgilenildiğinden dolayı bu model ile çalışmaya devam edilmiştir.

Çizelge 6.4. Lojistik regresyon analizi sonuçları

	$\hat{\beta}$	S.E	Wald	df	Sig.	Exp ($\hat{\beta}$)
Constant	0.744	0.817	0.829	1	0.363	2.103
X_1	0.333	0.620	0.288	1	0.591	1.395
X_2	-1.175	0.607	3.745	1	0.053	0.309
X_3	-0.060	0.616	0.010	1	0.922	0.941
X_4	-0.006	0.022	0.090	1	0.764	0.994

6.2.2. LR analizi ve NM yöntemi ile parametre tahmini

Çalışmanın bu bölümünde Matlab paket programında NM algoritmasını kullanarak çözüm sağlayan “fminsearch” kodları yardımıyla optimizasyon aranmıştır. Matlab paket programında kod yazımı için “command window” ekranına aşağıdaki parametre tahminlerini ve optimum sonucu veren kodlar girilmiştir.

`[X,Y, EXITFLAG, OUTPUT]=fminsearch(@olabilirlik, [-0:0,-0:0,-0:0,-0:0,-0:0])`

Komut işletildiğinde aşağıdaki çizelgenin üçüncü sütun sonuçları ve aşağıdaki çıktılar elde edilmiştir.

X = 0.6466 0.3135 -1.1803 -0.0365 -0.0025

Y = 32.4829

EXITFLAG = 1

OUTPUT =

iterations: 453

funcCount: 714

algorithm: 'Nelder-Mead simplex direct search'

message: [1x196 char]

Çizelge 6.5. Nelder-Mead algoritması sonuçları

Köşe Noktaları	[-3:3,-3:3,-3:3,-3:3]	[-1:1,-1:1,-1:1,-1:1]	[-0:0,-0:0,-0:0,-0:0]
	$\hat{\beta}$	$\hat{\beta}$	$\hat{\beta}$
Constant	0.647	0.647	0.647
X ₁	0.313	0.313	0.313
X ₂	-1.18	-1.18	-1.18
X ₃	-0.037	-0.037	-0.037
X ₄	-0.003	-0.003	-0.003
Fonksiyonun minimum değeri	32.48285466470697	32.48285466470732	32.48285466876738
İterasyon sayısı	5146	1269	453

Çizelge 6.5 incelendiğinde NM algoritması ile üç ayrı başlangıç köşe değerleri ile farklı iterasyonlar sonucunda optimum değer 32.48285466470697 ve parametre tahminleri sırasıyla 0.647, 0.313, -1.18, -0.037, -0.003 olarak bulunmuştur. Aynı sonuçlara Ek F’de detayları verilen Matlab programının “Optimization Tool” seçeneği ile de ulaşılmıştır.

6.2.3. Genetik algoritma ile parametre tahmini

Çalışmamızın bu bölümünde çalışmanın temelini oluşturan GA’nın parametre tahmininde yakınsama hızı ve optimum sonuç bakımından bu uygulama için hangi koşullarda en iyi sonuca ulaştığı bulunmuştur. Bu amaçla rastgele üretilen veriler ile

GA için parametre tahmininde en uygun başlangıç popülasyonu, seçim, çaprazlama, mutasyon operatörleri ve çaprazlama ve mutasyon olasılıkları belirlenmiştir. Aşağıda verilen çizelgelerde GA algoritması ile elde edilen LR parametre tahminleri farklı senaryolar altında, farklı kriterlere göre karşılaştırılmıştır.

Çizelge 6.6. GA için popülasyon boyutuna göre karşılaştırmalar

PopulationSize	20	40	60	80	100
Selection Function	Roulette	Roulette	Roulette	Roulette	Roulette
Mutation function	Gaussian	Gaussian	Gaussian	Gaussian	Gaussian
Mutation ratio	0.2	0.2	0.2	0.2	0.2
Mutation shrink	1.0	1.0	1.0	1.0	1.0
CrossoverFcn	İntermediate	İntermediate	İntermediate	İntermediate	İntermediate
Crossover ratio	1.0	1.0	1.0	1.0	1.0
Generations	5000	5000	5000	5000	5000
StallGenLimit	5000	5000	5000	5000	5000
TolFun	1e-6	1e-6	1e-6	1e-6	1e-6
TolCon	1e-6	1e-6	1e-6	1e-6	1e-6
Others	Default	Default	Default	Default	Default
$\hat{\beta}_0$ (Sabit)	0.647	0.647	0.647	0.647	0.647
$\hat{\beta}_1$	0.313	0.313	0.313	0.313	0.313
$\hat{\beta}_2$	-1.18	-1.18	-1.18	-1.18	-1.18
$\hat{\beta}_3$	-0.036	-0.037	-0.037	-0.037	-0.037
$\hat{\beta}_4$	-0.003	-0.003	-0.003	-0.003	-0.003
Fonksiyonun negatifinin	32.48285466	32.48285466	32.48285466	32.48285466	32.48285466
Minimize değeri	522573	470753	4724	470948	470892

Çizelge 6.7. GA için durdurma kriterlerine göre karşılaştırmalar

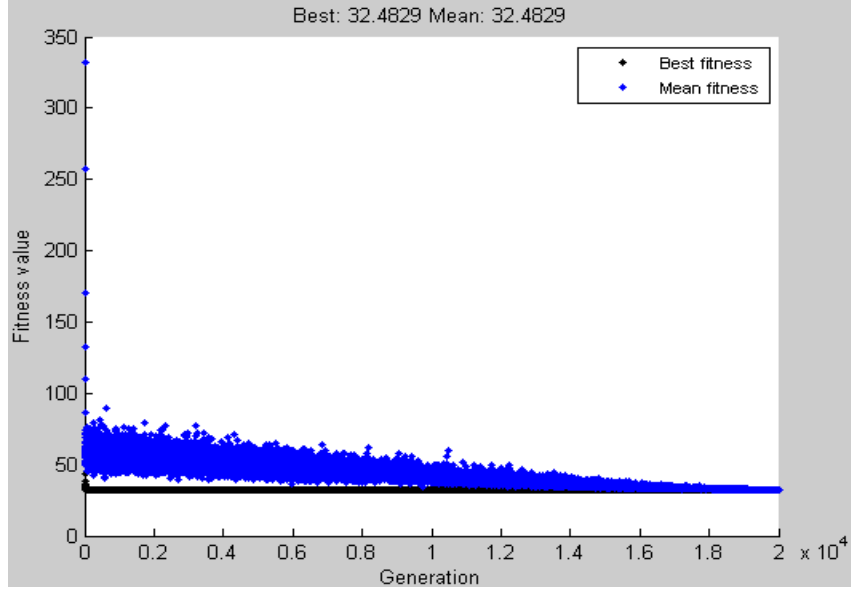
PopulationSize	60	60	60	60	60
Selection Function	Roulette	Roulette	Roulette	Roulette	Roulette
Mutation function	Gaussian	Gaussian	Gaussian	Gaussian	Gaussian
Mutation ratio	0.2	0.2	0.2	0.2	0.2
Mutation shrink	1.0	1.0	1.0	1.0	1.0
CrossoverFcn	İntermediate	İntermediate	İntermediate	İntermediate	İntermediate
Crossover ratio	1.0	1.0	1.0	1.0	1.0
Generations	10000	20000	30000	20000	20000
StallGenLimit	10000	20000	30000	20000	20000
TolFun	1e-6	1e-6	1e-6	1e-10	1e-15
TolCon	1e-6	1e-6	1e-6	1e-10	1e-15
Others	Default	Default	Default	Default	Default
$\hat{\beta}_0$ (Sabit)	0.647	0.647	0.647	0.647	0.647
$\hat{\beta}_1$	0.313	0.313	0.313	0.313	0.313
$\hat{\beta}_2$	-1.18	-1.18	-1.18	-1.18	-1.18
$\hat{\beta}_3$	-0.037	-0.037	-0.037	-0.037	-0.037
$\hat{\beta}_4$	-0.003	-0.003	-0.003	-0.003	-0.003
Fonksiyonun negatifinin	32.48285466	32.48285466	32.48285466	32.48285466	32.48285466
Minimize değeri	470744	4706964	470703	470710	470703

Çizelge 6.8. GA için Seçim yöntemine göre karşılaştırmalar

PopulationSize	60	60	60	60
Selection Function	Stochastic Uniform	Remainder	Uniform	Tournament (size 4)
Mutation function	Gaussian	Gaussian	Gaussian	Gaussian
Mutation ratio	0.2	0.2	0.2	0.2
Mutation shrink	1.0	1.0	1.0	1.0
CrossoverFcn	İntermediate	İntermediate	İntermediate	İntermediate
Crossover ratio	1.0	1.0	1.0	1.0
Generations	20000	20000	20000	20000
StallGenLimit	20000	20000	20000	20000
TolFun	1e-6	1e-6	1e-6	1e-6
TolCon	1e-6	1e-6	1e-6	1e-6
Others	Default	Default	Default	Default
$\hat{\beta}_0$ (Sabit)	0.647	0.647	0.647	0.646
$\hat{\beta}_1$	0.313	0.313	0.313	0.314
$\hat{\beta}_2$	-1.18	-1.18	-1.18	-1.18
$\hat{\beta}_3$	-0.037	-0.037	-0.037	-0.036
$\hat{\beta}_4$	-0.003	-0.003	-0.003	-0.003
Fonksiyonun negatifinin Minimize değeri	32.4828546647070	32.48285466	32.48285478	32.4828550128559
	0	470714	452693	9

Çizelgeler incelendiğinde, GA ile çözüm yapılırken birden çok başlangıç popülasyonu, çaprazlama ve mutasyon oranları ve farklı seçim yöntemleri denendiği anlaşılabilmektedir. Optimum sonuca ulaşmak için tek tek alternatifler denenmek suretiyle GA ile en az 100 ayrı ihtimal üzerinde durulmuştur.

Mutasyon fonksiyonu için dört (Use constraint dependent default, Gaussian, Uniform, Adaptive feasible); Mutasyon oranı için on; crossover fonksiyonu için altı (scattered, sinle point, two point, intermediate, heuristic, arithmetic), crossover oranı için on; seçim tercihi için beş ve diğer değişkenler için farklı ihtimaller denenmiştir. Optimum sonucun bulunduğu sonuç seçenekleri Çizelge 6.9 ile ve çözüme ulaşılırken çizilen grafik ise Şekil 6.1’de verilmiştir.



Şekil 6.1. Fitness Value (20000 iteration)

Çizelge 6.9. Kodlamada kullanılan parametreler

GA parametresi	Değer / Metod
PopulationType	'doubleVector'
PopulationSize	60
Selection Function	Roulette
Mutation function	Gaussian
Mutation ratio	0.2
Mutation shrink	1.0
CrossoverFcn	intermediate
Crossover ratio	1.0
Migration Direction	'forward'
Others (InitialPopulation, InitialScores et al.)	Use default
Stopping criteria	
Generations	20000
StallGenLimit	20000
TolFun	1e-6
TolCon	1e-6
Output	
PlotFcns	@gplotbestf
Display	'iter','final','diagnose'
Others	
	Use default

Şekil 6.1 incelendiğinde, 20000 iterasyon sonucunda uygunluk fonksiyonunun hangi değerlerden başladığı, hangi sonuçlara ulaştığı ve arama noktalarının serpilmeleri daha açık görülmektedir. Uygunluk fonksiyonun değerleri ilk iterasyonlarda 50 ve 100

civarında olduğu daha sonra ise genetik operatör işlemleri sonucunda fonksiyonun optimum değeri olan 30-35 aralığına yaklaştığı görülmektedir. Bu yaklaşmanın 10000 iterasyon sonrasında olduğu şekilden ve aşağıdaki çizelgeden anlaşılmaktadır.

Çizelge 6.10. GA'da iterasyon sayıları ve ulaşılan optimum değerler

Generation	f-count	Best f(x)	Mean f(x)	Stall Generations
19981	1198920	32.48	32.48	1927
19982	1198980	32.48	32.48	1928
19983	1199040	32.48	32.48	1929
19984	1199100	32.48	32.48	1930
19985	1199160	32.48	32.48	1931
19986	1199220	32.48	32.48	1932
19987	1199280	32.48	32.48	1933
19988	1199340	32.48	32.48	1934
19989	1199400	32.48	32.48	1935
19990	1199460	32.48	32.48	1936
19991	1199520	32.48	32.48	1937
19992	1199580	32.48	32.48	1938
19993	1199640	32.48	32.48	1939
19994	1199700	32.48	32.48	1940
19995	1199760	32.48	32.48	1941
19996	1199820	32.48	32.48	1942
19997	1199880	32.48	32.48	1943
19998	1199940	32.48	32.48	1944
19999	1200000	32.48	32.48	1945
20000	1200060	32.48	32.48	1946

Üretilen yapay bir veri kümesi için oluşturulan LR modelinin parametre tahmininde kullanılan üç alternatif optimizasyon yönteminden elde edilen sonuçlar aşağıdaki çizelgede karşılaştırmalı olarak verilmiştir.

Çizelge 6.11. En iyi parametre tahminlerinin karşılaştırılması

Parametreler	Lojistik Regresyon		
	Newton-Raphson	Nelder-Mead	Genetik Algoritma
$\hat{\beta}_0$ (Sabit)	0.744	0.647	0.647
$\hat{\beta}_1$	0.333	0.313	0.313
$\hat{\beta}_2$	-1.175	-1.18	-1.18
$\hat{\beta}_3$	-0.060	-0.037	-0.037
$\hat{\beta}_4$	-0.006	-0.003	-0.003
Fonksiyonun negatifinin Minimize değeri	32.4960	32.48285466470697	32.482854664706964

Çizelge 6.11 incelendiğinde, parametre tahminlerinin işaretlerinde ve büyüklüklerinde önemli farklılıklar gözlenmemiştir. Her üç algoritmadan elde edilen sonuçlara göre, olabilirlik fonksiyonunun negatifinin minimum değerlerine göre GA yaklaşımının en başarılı yöntem olduğu; ikinci sırayı NM yaklaşımının ve son sırayı ise NR yaklaşımının aldığı görülmüştür.

Bu sonuç bize kategorik bağımlı değişken modellemesinde parametre tahmini için klasik optimizasyon tekniklerinin yanısıra sezgisel optimizasyon tekniklerinin de kullanılabilir olduğunu ve daha başarılı sonuçlara bile ulaşılabilceğini göstermiştir.

Tek bir veri kümesi üzerinden elde edilen bu sonucun farklı senaryolar altında yapılacak bir simülasyon çalışması ile desteklenmesi durumunda, GA gibi sezgisel optimizasyon yöntemlerinin üstün özellikleri de dikkate alındığında, kategorik bağımlı değişkenlerde etkin ve en iyi parametre tahminlerine ulaşılabilmesi için optimizasyon sürecine önemli bir katkı sağlayacaktır.

7. SONUÇ VE YORUMLAR

Bu tez çalışmasında, literatürde birçok alanda yaygın olarak kullanılan GA, iteratif kök bulma algoritmalarından NR ve NM Algoritmalarının etkinliği bağımlı değişkenin kategorik yapıda olduğu LR modeli altında incelenmiştir.

LR modelinde parametre tahminlerine ulaşılabilme için kullanılan yöntem En Çok Olabilirlik yaklaşımı ile olabilirlik fonksiyonunun maksimizasyonuna dayanır. Optimizasyon sürecinin başarılı olabilmesi için olabilirlik fonksiyonunun diferansiyellenebilir olması koşulu vardır. Ayrıca, diferansiyellenebilir bir olabilirlik fonksiyonu kullanılarak elde edilen olabilirlik eşitlikleri parametrelerde doğrusal olmadıklarından dolayı iteratif kök bulma yöntemlerinin kullanılması gerekir. Klasik NR algoritması en iyi bilinen ve yaygın olarak kullanımı kabul görmüş bir kök bulma yöntemidir. Çalışmada ilk olarak NR algoritmasının teorisi ortaya konulmuştur. Alternatif optimizasyon yöntemlerinden arama tabanlı NM algoritması tanıtılmıştır. Son olarak, sezgisel optimizasyon yöntemleri kategorisinde olan GA yaklaşımı detaylı bir biçimde incelenmiştir.

NR ve NM gibi benzeri iteratif yöntemlerde çözümün aranmasına bir noktadan başlanır ve sonuçlara göre başka noktalarla devam edilir. Bu yöntemlerde parametre tahmininin başarılı olması için grafik inceleme gibi ön bir çalışma ile başlangıç noktası seçilmesi gerekebilmektedir. Ancak başlangıç noktasının hatalı seçimine göre çözümün bulunamaması veya çözüm süresinin uzaması gibi problemlerle karşılaşılabilir. Ayrıca bu algoritmalar genellikle belirli bir problem için uygundur ve optimize edilecek fonksiyonun sürekli ve türevlenebilir olması, ayrıca çözüm için bir başlangıç noktasının belirlenmesi gibi kısıtlayıcı varsayımlar gerektirmektedir.

GA'da ise arama işlemi tek nokta yerine potansiyel çözümlerin bir kümesi üzerinde gerçekleştirilir ve en iyi çözüme ulaşıncaya kadar çözümler değerlendirilir. GA'ların,

problemlerin çözümü için türev veya diğer yardımcı bilgilere gereksinim duymaması ve yerel optimum noktalara takılmadan global optimum noktalarını bulabilmesi gibi bazı avantajlarından dolayı kategorik bağımlı değişken modellemesinde parametre tahmini için iteratif yöntemlere alternatif olabileceği düşünülmüş ve çalışma bu doğrultuda devam ettirilmiştir.

Uygulama bölümünde bağımlı değişkenin iki düzeyli kategorik bir değişken olması durumunda birincisi gerçek tıbbi verilerinden, ikincisi ise rastgele üretilen verilerden oluşan iki farklı uygulama yapılmıştır. LR model parametreleri NR, NM ve GA yaklaşımları ile tahmin edilmiş ve olabirlik fonksiyonunu en büyük yapan algoritmanın her iki durumda da GA olduğu sonucuna ulaşılmıştır. Bu sonuç, bağımlı değişkenin kategorik bir değişken olması durumunda klasik yaklaşımlara alternatif olarak GA yaklaşımının başarılı bir biçimde kullanılabileceğini, dolayısıyla çalışmanın amacına ulaştığını göstermiştir.

Bundan sonra yapılması gerekli olan ilk çalışma, tek bir veri kümesi üzerinden elde edilen bulguların genelleştirilebilmesi için farklı senaryolar altında bir simülasyon çalışması tasarlanması sürecidir.

KAYNAKLAR

- Agresti, A. (2002) *Categorical Data Analysis*, 2th edition, John Wiley&Sons Inc., Hoboken, New Jersey, 710s.
- Aktaş, C. (2009) Lojistik Regresyon Analizi: Öğrencilerin Sigara İçme Alışkanlığı Üzerine Bir Uygulama, *Erciyes Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 1 (26); 107-121.
- Arıcan, E. (2010) *Nitel Yanıt Değişkene Sahip Regresyon Modellerinde Tahmin Yöntemleri*, Yüksek Lisans Tezi, Çukurova Üniversitesi, Adana, 105s.
- Akyol, A.P. (2006) *Doğrusal Olmayan Ekonometrik Modellerin Genetik Algoritma Yaklaşımı ile Parametre Tahmini*, Yüksek Lisans Tezi, Gazi Üniversitesi, Ankara, 105s.
- Altunkaynak, B. ve Esin, A. (2004) Doğrusal Olmayan Regresyonda Parametre Tahmini için Genetik Algoritma Yöntemi, *Gazi Üniversitesi Fen Bilimleri Dergisi*, 17 (2): 43-51.
- Bates, D.M. ve Watts, D.G. (1988) *Nonlinear Regression Analysis and its Applications*, Wiley, New York, 365s
- Draper, N. R. ve Smith, H. (1981) *Applied Regression Analysis*, 2th edition, John Wiley & Sons Inc., New York, 709s.
- Emel, G.G. ve Taskın, Ç. (2002) Genetik Algoritmalar ve Uygulama Alanları, *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 21 (1); 129-152.
- Frank, R. C. (1971) *Statistics and Econometrics*, Holt, Rinehart and Winston Inc., Austin, Texas, 400s.
- Gallant, A. R. (1975) Testing a Subset of the Parameters of a Nonlinear Regression Model, *JASA*, 70: 927-932.

- Gallant, A.R. (1987) *Nonlinear Statistical Models*, John Wiley&Sons, Usa, 610s.
- Genç, A. (1997) *Çok Değişkenli Lineer Olmayan Regresyon Modelleri Parametre Tahmini ve Hipotez Testleri*, Doktora Tezi, Ankara Üniversitesi, Ankara.
- Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Readin, MA, 507s.
- Hair, J. F, Black, W. C , Babin, B.J., Anderson, R. E. ve Tatham, R. L. (2006) *Multivariate data analysis*, 6th edition, Upper Saddle River, NJ: Prentice-Hall, 816s.
- Hartley, H.O. ve Booker, A. (1963) Nonlinear Least Squares Estimation, *Annals of Math. Stat.*, 36: 638-650.
- Holland, J.H. (1975) *Adaptation in Natural and Artificial Systems*, University of Michigan Press, USA, 183s.
- Hosmer, D.W. ve Lemeshow, S. (2000) *Applied Logistic Regression*, 2th edition, John Wiley & Sons Inc., New York, 392s.
- Karaboğa, D. (2011) *Yapay Zekâ Optimizasyon Algoritmaları*, 2. Baskı, Nobel yayıncılık, İstanbul, 244s.
- Karakoca, A. (2009) *Çok Değişkenli Lineer Olmayan Modellerde Genetik Algoritma*, Doktora Tezi, Selçuk Üniversitesi, Konya, 86s.
- Karr C. L. ve Freeman M. L. (1999) *Industrial Applications of Genetic Algorithms*, CRC Press, USA, 350s.
- Kleinbaum, D.G. ve Klein, M. (2002) *Logistic Regression A Self-Learning Text*, Second Edition, Springer-Verlag New York Inc, USA, 513s.
- Kutner, M.H., Nachtsheim, C.J., Neter, J. ve Li, W. (2004) *Applied Linear Statistical Models*, 5th edition, McGraw-Hill/Irwin, New York, USA, 1396s.

- Marquart, D.W. (1963) An Algorithm for Least Squares Estimation of Nonlinear Parameters, *Journal of The Society For Industrial and Applied Mathematics*, 11: 431-441.
- Mathews, J. H. ve Fink, K.K. (2004) *Numerical Methods Using Matlab*, 4th Edition, Prentice-Hall Inc., Upper Saddle River, New Jersey, USA, 696s.
- McCullagh, P. ve Nelder, J.A. (1989) *Generalized Linear Models*, 2th edition, Chapman and Hall, London, 532s.
- Menard, S. (2002) *Applied Logistic Regression Analysis*, 2th Edition, Sage Publications, USA, 128s.
- Michalewicz, Z. (1996) *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd Edition, Springer, USA, 387s.
- Miller, A. J. (1990) *Subset Selection in Regression*, Chapman and Hall, London, 229s.
- Mitchell, M. (1999) *An Introduction to Genetic Algorithms*, 5rd Edition, The Mit Press, Cambridge London, 143s.
- Nelder, J.A. ve Mead, R. (1964) A simplex method for function minimization, *Comput. J.*, 7: 308-313.
- Orhunbilge, N. (2002) *Uygulamalı Regresyon ve Korelasyon Analizi*, İstanbul, İ.Ü. İşletme Fakültesi Yayınları, İstanbul, 261s.
- Önder, H. (2001) *Lojistik regresyonlarda değişken seçimi*, Yüksek Lisan Tezi, Çukurova Üniversitesi, Adana, 91s.
- Özdamar, K. (2004) *Paket Programlar ile İstatistiksel Veri Analizi 1*, Kaan Kitabevi, Eskişehir, 649s.
- Öztürk, N. ve Çelik, E. (2012) Polinom olmayan denklemlerin genetik algoritma tabanlı çözümü, *Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 28(4): 322-328

- Ratkowsky, D.A. (1983) *Nonlinear Regression Modeling: A Unified Practical Approach*, Marcel Dekker, New York, 288s.
- Reeves, C.R. ve Rowe, J.E. (2002) *Genetic Algorithms Principles and Perspectives A Guide to GA Theory*, Kluwer Academic Pres, USA, 332s.
- Rothlauf, F. (2006) *Representations for Genetic and Evolutionary Algorithms*, 2th edition, Springer, Berlin, 342s.
- Seber, G.A.F. ve Wild, C.J. (1989) *Nonlinear Regression*, John Wiley&Sons, USA, 768s.
- Serin, T. (2010) *Doğrusal Olmayan Regresyon Modellerinde Parametre Tahmin Yöntemleri, Öneriler Ve Karşılaştırmaları*, Doktora Tezi, Gazi Üniversitesi, Ankara, 138s.
- Şahin, M. (1999) *Lojistik Regresyon Ve Biyolojik Alanlarda Kullanımı*, Yüksek Lisans Tezi, Kahramanmaraş Sütçü İmam Üniversitesi, Kahramanmaraş, 88s.
- Tatlıdil, H. (1996) *Uygulamalı Çok Değişkenli İstatistiksel Analiz*, Akademi Matbaası, Ankara, 424s.
- Tektaş, M. (2010) Optimizasyon Tekniklerinin Tasnifi Ve İki Teknik Üzerinde İnceleme, *e-Journal of New World Sciences Academy*, 5 (4): 103-115.
- Tvrđik, J., Krivy, I. ve Misik, L. (2006) Adaptive population-based search: Application to estimation of nonlinear regression parameters, *Computational Statistics and Data Analysis* 52: 713-724.
- Uçar, Ö. (2004) *Nitel Verilerin Analizinde Lojit ve Probit Modeller*, Yüksek Lisans Tezi, Hacettepe Üniversitesi, Ankara, 151s.
- Yeniay, M.Ö. (1999) *Taguchi Deney Tasarımı Problemlerine Genetik Algoritma Yaklaşımı*, Doktora Tezi, Hacettepe Üniversitesi, Ankara.

Ek A. İki Boyutlu Örnek Bir Problem için Nelder - Mead Matlab Çıktıları

Iteration	Func-count	min f(x)	Procedure
0	1	0	initial simplex
1	3	-0.000999937	expand
2	5	-0.00187486	expand
3	7	-0.00381168	expand
4	9	-0.00652963	expand
5	11	-0.0117583	expand
6	13	-0.0198034	expand
7	15	-0.0342564	expand
8	17	-0.0575215	expand
9	19	-0.0978636	expand
10	21	-0.164105	expand
11	23	-0.276405	expand
12	25	-0.461152	expand
13	27	-0.766292	expand
14	29	-1.25629	expand
15	31	-2.01725	expand
16	33	-3.11589	expand
17	35	-4.43005	expand
18	37	-5.15723	expand
19	38	-5.15723	reflect
20	40	-5.15723	contract inside
21	42	-5.15723	contract inside
22	44	-5.15723	contract inside
23	46	-5.16197	contract inside
24	48	-5.16651	reflect
25	50	-5.16836	contract inside
26	52	-5.17972	expand
27	54	-5.17972	contract inside
28	56	-5.1931	expand
29	58	-5.21162	expand
30	60	-5.25056	expand
31	62	-5.29848	expand
32	64	-5.40835	expand
33	66	-5.51527	expand
34	68	-5.7946	expand

Ek A. (devam)

Iteration	Func-count	min f(x)	Procedure
35	70	-5.90036	expand
36	72	-6.33289	expand
37	73	-6.33289	reflect
38	74	-6.33289	reflect
39	76	-6.78149	expand
40	77	-6.78149	reflect
41	79	-6.85979	reflect
42	81	-6.90845	reflect
43	83	-6.90993	reflect
44	85	-6.9769	contract inside
45	87	-6.99615	contract inside
46	89	-6.99615	contract outside
47	91	-6.99615	contract inside
48	93	-6.99838	contract outside
49	95	-6.99974	contract inside
50	97	-6.99974	contract inside
51	99	-6.99974	contract inside
52	101	-6.99983	contract inside
53	103	-6.9999	contract inside
54	105	-6.99998	contract inside
55	107	-6.99998	contract inside
56	109	-6.99999	reflect
57	111	-6.99999	contract inside
58	113	-7	contract inside
59	115	-7	contract inside
60	117	-7	contract inside
61	119	-7	contract outside
62	121	-7	contract inside
63	123	-7	contract inside
64	125	-7	contract outside
65	127	-7	contract inside
66	129	-7	contract inside
67	131	-7	contract inside
68	133	-7	contract inside
69	134	-7	reflect

Ek B. Olabilirlik Fonksiyonunun Matlab Kodu

function [a] = olabilirlik(x)

a = -

```
(52*x(1)+1481*x(2)+37*x(3)+1167*x(4)+1189*x(5)+4899*x(6)+2255.4*x(7)+8430
*x(8)+4826.07*x(9)+178.45*x(10)+55.28*x(11)+103.67*x(12)+6695*x(13))-
log(1+exp(x(1)+18*x(2)+x(3)+20*x(4)+17*x(5)+100*x(6)+39.1*x(7)+199*x(8)+76
.7*x(9)+4.57*x(10)+1.31*x(11)+2.5*x(12)+416*x(13)))-
log(1+exp(x(1)+23*x(2)+14*x(4)+30*x(5)+92*x(6)+39.2*x(7)+143*x(8)+93.19*x(
9)+2.97*x(10)+0.79*x(11)+0.94*x(12)+53*x(13)))-
log(1+exp(x(1)+22*x(2)+17*x(4)+18*x(5)+87*x(6)+58.2*x(7)+179*x(8)+111.2*x(
9)+3.01*x(10)+1.32*x(11)+0.82*x(12)+48*x(13)))-
log(1+exp(x(1)+49*x(2)+x(3)+36*x(4)+30*x(5)+100*x(6)+40.5*x(7)+186*x(8)+10
8.3*x(9)+3.63*x(10)+1.47*x(11)+0.51*x(12)+186*x(13)))-
log(1+exp(x(1)+15*x(2)+x(3)+21*x(4)+19*x(5)+84*x(6)+51.6*x(7)+131*x(8)+51.
19*x(9)+5.23*x(10)+1.88*x(11)+1.67*x(12)+141*x(13)))-
log(1+exp(x(1)+48*x(2)+x(3)+23*x(4)+21*x(5)+96*x(6)+61.1*x(7)+209*x(8)+130
.5*x(9)+3.06*x(10)+1.06*x(11)+2.35*x(12)+87*x(13)))-
log(1+exp(x(1)+49*x(2)+x(3)+19*x(4)+14*x(5)+104*x(6)+79.3*x(7)+215*x(8)+11
3.5*x(9)+3.16*x(10)+1.36*x(11)+2.17*x(12)+111*x(13)))-
log(1+exp(x(1)+16*x(2)+17*x(4)+18*x(5)+79*x(6)+40.3*x(7)+150*x(8)+81.5*x(9)
+3.56*x(10)+1.3*x(11)+2.06*x(12)+141*x(13)))-
log(1+exp(x(1)+35*x(2)+x(3)+32*x(4)+25*x(5)+89*x(6)+29.5*x(7)+189*x(8)+111
.9*x(9)+3.29*x(10)+0.82*x(11)+1.28*x(12)+238*x(13)))-
log(1+exp(x(1)+37*x(2)+x(3)+23*x(4)+23*x(5)+79*x(6)+33.2*x(7)+177*x(8)+114
.6*x(9)+2.87*x(10)+0.74*x(11)+1.23*x(12)+112*x(13)))-
log(1+exp(x(1)+19*x(2)+x(3)+17*x(4)+18*x(5)+99*x(6)+34.9*x(7)+117*x(8)+51.
09*x(9)+4.12*x(10)+1.44*x(11)+4.11*x(12)+155*x(13)))-
log(1+exp(x(1)+10*x(2)+x(3)+20*x(4)+31*x(5)+95*x(6)+52*x(7)+141*x(8)+81.8*
x(9)+3.55*x(10)+0.9*x(11)+1.91*x(12)+36*x(13)))-
log(1+exp(x(1)+15*x(2)+x(3)+21*x(4)+35*x(5)+91*x(6)+38*x(7)+128*x(8)+56*x(
9)+3.54*x(10)+0.65*x(11)+2*x(12)+170*x(13)))-
log(1+exp(x(1)+38*x(2)+x(3)+37*x(4)+23*x(5)+102*x(6)+30.4*x(7)+193*x(8)+14
3.6*x(9)+2.99*x(10)+x(11)+1.38*x(12)+95*x(13)))-
log(1+exp(x(1)+17*x(2)+x(3)+16*x(4)+24*x(5)+97*x(6)+47.1*x(7)+120*x(8)+68.
1*x(9)+4.06*x(10)+1.02*x(11)+1.12*x(12)+24*x(13)))-
log(1+exp(x(1)+39*x(2)+x(3)+36*x(4)+24*x(5)+103*x(6)+31.5*x(7)+118*x(8)+75
.69*x(9)+4.06*x(10)+1.66*x(11)+2.09*x(12)+54*x(13)))-
log(1+exp(x(1)+20*x(2)+x(3)+8*x(4)+14*x(5)+106*x(6)+40.3*x(7)+210*x(8)+143
.3*x(9)+2.94*x(10)+0.81*x(11)+3.81*x(12)+132*x(13)))-
log(1+exp(x(1)+21*x(2)+20*x(4)+24*x(5)+102*x(6)+51.2*x(7)+193*x(8)+124*x(9)
)+3.06*x(10)+0.76*x(11)+2.79*x(12)+130*x(13)))-
log(1+exp(x(1)+9*x(2)+x(3)+11*x(4)+18*x(5)+82*x(6)+49.4*x(7)+128*x(8)+69.3
*x(9)+2.67*x(10)+0.7*x(11)+1.78*x(12)+18*x(13)))-
log(1+exp(x(1)+59*x(2)+x(3)+27*x(4)+29*x(5)+93*x(6)+35.5*x(7)+172*x(8)+85.
```

Ek B. (devam)

$$\begin{aligned} &9*x(9)+3.19*x(10)+0.62*x(11)+3.21*x(12)+253*x(13)))- \\ &\log(1+\exp(x(1)+21*x(2)+16*x(4)+16*x(5)+79*x(6)+39.8*x(7)+138*x(8)+71.3*x(9) \\ &+3.05*x(10)+0.94*x(11)+0.97*x(12)+109*x(13)))- \\ &\log(1+\exp(x(1)+24*x(2)+x(3)+13*x(4)+18*x(5)+127*x(6)+47.5*x(7)+163*x(8)+10 \\ &1.9*x(9)+3.67*x(10)+1.33*x(11)+1.67*x(12)+68*x(13)))- \\ &\log(1+\exp(x(1)+7*x(2)+x(3)+16*x(4)+38*x(5)+78*x(6)+66.9*x(7)+137*x(8)+64.0 \\ &9*x(9)+3.54*x(10)+1.02*x(11)+2.56*x(12)+30*x(13)))- \\ &\log(1+\exp(x(1)+31*x(2)+19*x(4)+19*x(5)+93*x(6)+57.2*x(7)+169*x(8)+102.6*x(\\ &9)+3.23*x(10)+1.28*x(11)+1.4*x(12)+46*x(13)))- \\ &\log(1+\exp(x(1)+7*x(2)+19*x(4)+28*x(5)+96*x(6)+42*x(7)+158*x(8)+64.8*x(9)+3 \\ &.87*x(10)+1.53*x(11)+5.09*x(12)+256*x(13)))- \\ &\log(1+\exp(x(1)+49*x(2)+x(3)+12*x(4)+17*x(5)+102*x(6)+33.6*x(7)+187*x(8)+12 \\ &2.8*x(9)+3.32*x(10)+1.07*x(11)+0.33*x(12)+153*x(13)))- \\ &\log(1+\exp(x(1)+26*x(2)+x(3)+18*x(4)+20*x(5)+86*x(6)+50.08*x(7)+176*x(8)+11 \\ &2*x(9)+3.24*x(10)+1.29*x(11)+0.78*x(12)+66*x(13)))- \\ &\log(1+\exp(x(1)+17*x(2)+x(3)+14*x(4)+16*x(5)+72*x(6)+42.5*x(7)+169*x(8)+106 \\ &.1*x(9)+3.19*x(10)+0.7*x(11)+2.23*x(12)+102*x(13)))- \\ &\log(1+\exp(x(1)+6*x(2)+17*x(4)+32*x(5)+84*x(6)+47.4*x(7)+154*x(8)+93.19*x(9) \\ &+3.96*x(10)+0.97*x(11)+3.39*x(12)+67*x(13)))- \\ &\log(1+\exp(x(1)+35*x(2)+x(3)+51*x(4)+26*x(5)+96*x(6)+24.8*x(7)+159*x(8)+73*x \\ &x(9)+3.39*x(10)+0.74*x(11)+2.21*x(12)+306*x(13)))- \\ &\log(1+\exp(x(1)+32*x(2)+x(3)+21*x(4)+23*x(5)+105*x(6)+41.7*x(7)+281*x(8)+18 \\ &5.89*x(9)+3.87*x(10)+1.4*x(11)+1.26*x(12)+267*x(13)))- \\ &\log(1+\exp(x(1)+13*x(2)+13*x(4)+19*x(5)+87*x(6)+41.3*x(7)+130*x(8)+73.9*x(9) \\ &+3.63*x(10)+1.23*x(11)+3.19*x(12)+74*x(13)))- \\ &\log(1+\exp(x(1)+27*x(2)+11*x(4)+15*x(5)+91*x(6)+55.2*x(7)+182*x(8)+123*x(9) \\ &+2.96*x(10)+1.12*x(11)+1.71*x(12)+19*x(13)))- \\ &\log(1+\exp(x(1)+11*x(2)+x(3)+27*x(4)+30*x(5)+87*x(6)+59.8*x(7)+156*x(8)+82*x \\ &x(9)+4.2*x(10)+1.15*x(11)+2.84*x(12)+71*x(13)))- \\ &\log(1+\exp(x(1)+33*x(2)+15*x(4)+15*x(5)+84*x(6)+34.7*x(7)+141*x(8)+90.9*x(9) \\ &+2.71*x(10)+1.01*x(11)+1.45*x(12)+77*x(13)))- \\ &\log(1+\exp(x(1)+29*x(2)+x(3)+20*x(4)+22*x(5)+94*x(6)+35.1*x(7)+140*x(8)+82. \\ &3*x(9)+2.71*x(10)+1.07*x(11)+2.67*x(12)+113*x(13)))- \\ &\log(1+\exp(x(1)+34*x(2)+14*x(4)+13*x(5)+80*x(6)+55.5*x(7)+158*x(8)+92.9*x(9) \\ &+2.22*x(10)+0.66*x(11)+0.01*x(12)+48*x(13)))- \\ &\log(1+\exp(x(1)+56*x(2)+x(3)+47*x(4)+37*x(5)+125*x(6)+31*x(7)+186*x(8)+116. \\ &6*x(9)+3.21*x(10)+0.7*x(11)+0.52*x(12)+192*x(13)))- \\ &\log(1+\exp(x(1)+34*x(2)+x(3)+36*x(4)+22*x(5)+114*x(6)+26.6*x(7)+194*x(8)+11 \\ &2.2*x(9)+3.49*x(10)+0.72*x(11)+2.41*x(12)+276*x(13)))- \\ &\log(1+\exp(x(1)+18*x(2)+x(3)+15*x(4)+20*x(5)+98*x(6)+57*x(7)+130*x(8)+64.59 \\ &*x(9)+3.4*x(10)+0.93*x(11)+2.82*x(12)+42*x(13)))- \\ &\log(1+\exp(x(1)+40*x(2)+46*x(4)+28*x(5)+72*x(6)+28.3*x(7)+124*x(8)+74.9*x(9) \\ &+2.7*x(10)+0.79*x(11)+1.13*x(12)+104*x(13)))- \\ &\log(1+\exp(x(1)+32*x(2)+x(3)+19*x(4)+18*x(5)+79*x(6)+47.1*x(7)+164*x(8)+86. \\ &1*x(9)+3.5*x(10)+0.95*x(11)+2.7*x(12)+154*x(13)))- \end{aligned}$$

Ek B. (devam)

$\log(1+\exp(x(1)+38*x(2)+x(3)+33*x(4)+24*x(5)+126*x(6)+27.3*x(7)+186*x(8)+82.3*x(9)+2.92*x(10)+0.62*x(11)+6.5*x(12)+382*x(13)))-$
 $\log(1+\exp(x(1)+16*x(2)+x(3)+14*x(4)+23*x(5)+105*x(6)+50.1*x(7)+132*x(8)+75.69*x(9)+3.77*x(10)+0.82*x(11)+1.93*x(12)+31*x(13)))-$
 $\log(1+\exp(x(1)+50*x(2)+22*x(4)+21*x(5)+95*x(6)+33.1*x(7)+172*x(8)+115.5*x(9)+4.11*x(10)+0.91*x(11)+0.66*x(12)+117*x(13)))-$
 $\log(1+\exp(x(1)+33*x(2)+x(3)+36*x(4)+30*x(5)+83*x(6)+29.9*x(7)+160*x(8)+81.9*x(9)+3.81*x(10)+1.16*x(11)+1.21*x(12)+241*x(13)))-$
 $\log(1+\exp(x(1)+21*x(2)+x(3)+19*x(4)+22*x(5)+128*x(6)+50.6*x(7)+146*x(8)+68.19*x(9)+3.31*x(10)+1.4*x(11)+3.01*x(12)+136*x(13)))-$
 $\log(1+\exp(x(1)+22*x(2)+x(3)+14*x(4)+14*x(5)+84*x(6)+31*x(7)+135*x(8)+85*x(9)+4.14*x(10)+1.22*x(11)+1.84*x(12)+95*x(13)))-$
 $\log(1+\exp(x(1)+46*x(2)+x(3)+16*x(4)+19*x(5)+88*x(6)+49.2*x(7)+181*x(8)+109.6*x(9)+3.31*x(10)+1.02*x(11)+0.66*x(12)+111*x(13)))-$
 $\log(1+\exp(x(1)+20*x(2)+x(3)+15*x(4)+16*x(5)+75*x(6)+31.2*x(7)+132*x(8)+83.19*x(9)+3.92*x(10)+1.26*x(11)+1.53*x(12)+88*x(13)))-$
 $\log(1+\exp(x(1)+49*x(2)+64*x(4)+53*x(5)+111*x(6)+50*x(7)+152*x(8)+68.59*x(9)+3.95*x(10)+1.41*x(11)+1.52*x(12)+167*x(13)))-$
 $\log(1+\exp(x(1)+45*x(2)+x(3)+20*x(4)+20*x(5)+95*x(6)+54.9*x(7)+140*x(8)+67.69*x(9)+2.62*x(10)+1.25*x(11)+1.74*x(12)+87*x(13)))-$
 $\log(1+\exp(x(1)+20*x(2)+x(3)+18*x(4)+15*x(5)+83*x(6)+61*x(7)+120*x(8)+39.8*x(9)+3.7*x(10)+1.48*x(11)+3.57*x(12)+96*x(13)))-$
 $\log(1+\exp(x(1)+22*x(2)+27*x(4)+17*x(5)+87*x(6)+44.5*x(7)+127*x(8)+73.5*x(9)+4.13*x(10)+1.28*x(11)+1.62*x(12)+45*x(13)))-$
 $\log(1+\exp(x(1)+22*x(2)+x(3)+17*x(4)+23*x(5)+92*x(6)+36.3*x(7)+174*x(8)+105.9*x(9)+3.16*x(10)+0.75*x(11)+0.82*x(12)+159*x(13)))-$
 $\log(1+\exp(x(1)+14*x(2)+15*x(4)+19*x(5)+85*x(6)+44.1*x(7)+179*x(8)+120.5*x(9)+3.62*x(10)+1.36*x(11)+1.66*x(12)+72*x(13)))-$
 $\log(1+\exp(x(1)+38*x(2)+x(3)+38*x(4)+31*x(5)+76*x(6)+33.8*x(7)+218*x(8)+151.6*x(9)+3.32*x(10)+1.22*x(11)+2.65*x(12)+163*x(13)))-$
 $\log(1+\exp(x(1)+68*x(2)+x(3)+16*x(4)+29*x(5)+91*x(6)+45.7*x(7)+190*x(8)+134.1*x(9)+3.09*x(10)+0.8*x(11)+0.21*x(12)+51*x(13)))-$
 $\log(1+\exp(x(1)+49*x(2)+33*x(4)+28*x(5)+106*x(6)+39.9*x(7)+151*x(8)+183.7*x(9)+3.1*x(10)+0.89*x(11)+0.36*x(12)+208*x(13)))-$
 $\log(1+\exp(x(1)+41*x(2)+x(3)+29*x(4)+23*x(5)+104*x(6)+37.2*x(7)+230*x(8)+136.89*x(9)+3.49*x(10)+0.61*x(11)+0.71*x(12)+256*x(13)))-$
 $\log(1+\exp(x(1)+27*x(2)+13*x(4)+17*x(5)+81*x(6)+64.7*x(7)+139*x(8)+61.1*x(9)+3.07*x(10)+0.87*x(11)+1.44*x(12)+66*x(13)))-$
 $\log(1+\exp(x(1)+19*x(2)+13*x(4)+18*x(5)+89*x(6)+28.9*x(7)+143*x(8)+97.09*x(9)+3.48*x(10)+0.93*x(11)+1.78*x(12)+85*x(13)))-$
 $\log(1+\exp(x(1)+40*x(2)+20*x(4)+20*x(5)+87*x(6)+56.6*x(7)+197*x(8)+128.6*x(9)+2.37*x(10)+0.66*x(11)+2.88*x(12)+59*x(13)))-$
 $\log(1+\exp(x(1)+18*x(2)+16*x(4)+18*x(5)+158*x(6)+61.7*x(7)+154*x(8)+75.09*x(9)+3.59*x(10)+1.13*x(11)+4.07*x(12)+86*x(13)))-$
 $\log(1+\exp(x(1)+33*x(2)+x(3)+31*x(4)+28*x(5)+87*x(6)+29.1*x(7)+172*x(8)+121.5*x(9)+3.57*x(10)+0.98*x(11)+0.46*x(12)+107*x(13)))-$

Ek B. (devam)

$\log(1+\exp(x(1)+16*x(2)+12*x(4)+18*x(5)+85*x(6)+39.5*x(7)+156*x(8)+99.69*x(9)+3.58*x(10)+1.44*x(11)+3.25*x(12)+84*x(13)))-$
 $\log(1+\exp(x(1)+79*x(2)+x(3)+13*x(4)+10*x(5)+90*x(6)+39*x(7)+137*x(8)+71.4*x(9)+3.12*x(10)+0.6*x(11)+0.79*x(12)+46*x(13)))-$
 $\log(1+\exp(x(1)+21*x(2)+15*x(4)+18*x(5)+88*x(6)+48.2*x(7)+127*x(8)+70.3*x(9)+3.69*x(10)+0.73*x(11)+2.14*x(12)+29*x(13)))-$
 $\log(1+\exp(x(1)+48*x(2)+12.5*x(4)+11.6*x(5)+87*x(6)+49.9*x(7)+221*x(8)+132.8*x(9)+2.51*x(10)+1.72*x(11)+0.13*x(12)+145*x(13)))-$
 $\log(1+\exp(x(1)+19*x(2)+x(3)+25*x(4)+21*x(5)+81*x(6)+38.7*x(7)+177*x(8)+123.2*x(9)+3.52*x(10)+0.85*x(11)+1.26*x(12)+69*x(13)))-$
 $\log(1+\exp(x(1)+22*x(2)+44*x(4)+29*x(5)+85*x(6)+29.5*x(7)+137*x(8)+64.9*x(9)+3.17*x(10)+0.69*x(11)+2.15*x(12)+213*x(13)))-$
 $\log(1+\exp(x(1)+32*x(2)+x(3)+23*x(4)+19*x(5)+84*x(6)+31.1*x(7)+160*x(8)+80.3*x(9)+4.28*x(10)+1.1*x(11)+0.11*x(12)+243*x(13)))-$
 $\log(1+\exp(x(1)+32*x(2)+18*x(4)+24*x(5)+91*x(6)+45.8*x(7)+175*x(8)+111.8*x(9)+3.71*x(10)+0.98*x(11)+0.9*x(12)+87*x(13)))-$
 $\log(1+\exp(x(1)+31*x(2)+x(3)+17*x(4)+18*x(5)+94*x(6)+45.1*x(7)+147*x(8)+75.5*x(9)+3.52*x(10)+1.4*x(11)+2.32*x(12)+132*x(13)))-$
 $\log(1+\exp(x(1)+26*x(2)+x(3)+18*x(4)+17*x(5)+282*x(6)+48.9*x(7)+177*x(8)+112.9*x(9)+3.75*x(10)+1.28*x(11)+1.96*x(12)+76*x(13)))-$
 $\log(1+\exp(x(1)+20*x(2)+x(3)+13*x(4)+22*x(5)+83*x(6)+52*x(7)+151*x(8)+101.2*x(9)+3.5*x(10)+0.71*x(11)+1.44*x(12)+54*x(13)))-$
 $\log(1+\exp(x(1)+37*x(2)+x(3)+45*x(4)+18*x(5)+80*x(6)+43*x(7)+203*x(8)+121.8*x(9)+2.89*x(10)+0.82*x(11)+2.04*x(12)+125*x(13)))-$
 $\log(1+\exp(x(1)+28*x(2)+x(3)+36*x(4)+20*x(5)+91*x(6)+28.2*x(7)+175*x(8)+115.8*x(9)+3.73*x(10)+0.96*x(11)+1.88*x(12)+155*x(13)))-$
 $\log(1+\exp(x(1)+18*x(2)+19*x(4)+22*x(5)+72*x(6)+41.7*x(7)+143*x(8)+85.3*x(9)+2.92*x(10)+0.93*x(11)+2.54*x(12)+80*x(13)))-$
 $\log(1+\exp(x(1)+34*x(2)+15*x(4)+18*x(5)+109*x(6)+44.1*x(7)+172*x(8)+111.7*x(9)+2.88*x(10)+1.17*x(11)+0.68*x(12)+81*x(13)))-$
 $\log(1+\exp(x(1)+32*x(2)+19*x(4)+21*x(5)+73*x(6)+51.9*x(7)+190*x(8)+107.7*x(9)+3.43*x(10)+0.77*x(11)+1.94*x(12)+152*x(13)))-$
 $\log(1+\exp(x(1)+52*x(2)+x(3)+35*x(4)+26*x(5)+142*x(6)+38*x(7)+234*x(8)+158.2*x(9)+2.79*x(10)+0.84*x(11)+1.43*x(12)+189*x(13)))-$
 $\log(1+\exp(x(1)+15*x(2)+x(3)+15*x(4)+20*x(5)+93*x(6)+42.6*x(7)+159*x(8)+106.6*x(9)+3.63*x(10)+0.82*x(11)+2.57*x(12)+49*x(13)))-$
 $\log(1+\exp(x(1)+21*x(2)+x(3)+15*x(4)+20*x(5)+90*x(6)+36.8*x(7)+126*x(8)+81.6*x(9)+3.9*x(10)+0.82*x(11)+2.27*x(12)+38*x(13)))-$
 $\log(1+\exp(x(1)+66*x(2)+x(3)+34*x(4)+26*x(5)+93*x(6)+36.2*x(7)+179*x(8)+117*x(9)+3.14*x(10)+0.98*x(11)+3.07*x(12)+129*x(13)))-$
 $\log(1+\exp(x(1)+40*x(2)+x(3)+34*x(4)+28*x(5)+84*x(6)+36*x(7)+191*x(8)+130*x(9)+3.23*x(10)+0.79*x(11)+0.28*x(12)+125*x(13)))-$
 $\log(1+\exp(x(1)+22*x(2)+x(3)+22*x(4)+19*x(5)+92*x(6)+53.6*x(7)+225*x(8)+150.6*x(9)+3.84*x(10)+0.56*x(11)+1.36*x(12)+104*x(13)))-$
 $\log(1+\exp(x(1)+18*x(2)+18*x(4)+19*x(5)+90*x(6)+32.8*x(7)+123*x(8)+78.59*x(9)+3.16*x(10)+0.94*x(11)+1.03*x(12)+58*x(13)))-$

Ek B. (devam)

```
log(1+exp(x(1)+26*x(2)+12*x(4)+19*x(5)+89*x(6)+52.6*x(7)+165*x(8)+102.4*x(
9)+3.49*x(10)+1.53*x(11)+0.35*x(12)+50*x(13)))-
log(1+exp(x(1)+25*x(2)+13*x(4)+18*x(5)+79*x(6)+61.4*x(7)+181*x(8)+103*x(9)
+2.76*x(10)+0.67*x(11)+6.01*x(12)+83*x(13)))-
log(1+exp(x(1)+24*x(2)+12*x(4)+13*x(5)+80*x(6)+48.8*x(7)+146*x(8)+84.4*x(9)
+3.45*x(10)+1.13*x(11)+1.49*x(12)+64*x(13)))-
log(1+exp(x(1)+18*x(2)+14*x(4)+17*x(5)+85*x(6)+52.1*x(7)+149*x(8)+83.9*x(9)
+3.72*x(10)+1.18*x(11)+2.46*x(12)+65*x(13)))-
log(1+exp(x(1)+23*x(2)+13*x(4)+22*x(5)+79*x(6)+53.5*x(7)+128*x(8)+71.3*x(9)
+3.46*x(10)+1.18*x(11)+1.7*x(12)+16*x(13)))-
log(1+exp(x(1)+25*x(2)+10.6*x(4)+16.5*x(5)+88*x(6)+41.3*x(7)+140*x(8)+89.09
*x(9)+3.19*x(10)+0.62*x(11)+2.52*x(12)+136.7*x(13)))-
log(1+exp(x(1)+25*x(2)+13*x(4)+17*x(5)+81*x(6)+69.7*x(7)+202*x(8)+118.7*x(
9)+3.01*x(10)+0.86*x(11)+0.19*x(12)+68*x(13))));
end
```

Ek C. İkinci Uygulamada Kullanılan Rastgele Üretilen Veriler

Y	X ₁	X ₂	X ₃	X ₄
1	0	1	0	6
1	0	1	1	22
1	1	1	1	20
1	0	1	0	32
1	0	0	1	24
1	1	1	1	16
1	1	0	0	47
1	1	0	0	4
1	0	1	1	6
1	1	0	0	25
1	0	0	0	36
1	0	0	1	32
1	1	0	1	20
1	0	0	1	43
1	0	0	0	18
1	0	1	0	25
1	1	1	0	37
1	1	1	1	17
1	0	0	1	22
1	0	0	0	38
1	1	0	1	36
1	0	1	0	45
1	1	1	1	1
1	0	0	1	5
1	1	0	0	43
0	0	1	0	33
0	0	1	0	29
0	0	1	0	47
0	1	1	0	15
0	0	0	0	14
0	0	1	0	13
0	0	1	1	29
0	1	1	0	33
0	0	1	1	26
0	0	1	0	42

Ek C. (devam)

Y	X ₁	X ₂	X ₃	X ₄
0	0	0	0	12
0	0	1	0	29
0	1	1	1	43
0	1	1	1	40
0	0	0	1	41
0	1	1	1	43
0	0	1	0	1
0	0	1	1	13
0	0	1	1	16
0	0	0	1	7
0	1	1	0	36
0	1	0	1	14
0	0	1	0	15
0	1	0	1	1
0	1	0	1	37

Ek D. Yapay Veriler İçin Olabilirlik 2 Fonksiyonunun Matlab Kodu

```
function [a] = olabilirlik2(x)
a = -(25*x(1)+11*x(2)+11*x(3)+13*x(4)+620*x(5)-
log(1+exp(x(1)+0*x(2)+1*x(3)+0*x(4)+6*x(5)))-
log(1+exp(x(1)+0*x(2)+1*x(3)+1*x(4)+22*x(5)))-
log(1+exp(x(1)+1*x(2)+1*x(3)+1*x(4)+20*x(5)))-
log(1+exp(x(1)+0*x(2)+1*x(3)+0*x(4)+32*x(5)))-
log(1+exp(x(1)+0*x(2)+0*x(3)+1*x(4)+24*x(5)))-
log(1+exp(x(1)+1*x(2)+1*x(3)+1*x(4)+16*x(5)))-
log(1+exp(x(1)+1*x(2)+0*x(3)+0*x(4)+47*x(5)))-
log(1+exp(x(1)+1*x(2)+0*x(3)+0*x(4)+4*x(5)))-
log(1+exp(x(1)+0*x(2)+1*x(3)+1*x(4)+6*x(5)))-
log(1+exp(x(1)+1*x(2)+0*x(3)+0*x(4)+25*x(5)))-
log(1+exp(x(1)+0*x(2)+0*x(3)+0*x(4)+36*x(5)))-
log(1+exp(x(1)+0*x(2)+0*x(3)+1*x(4)+32*x(5)))-
log(1+exp(x(1)+1*x(2)+0*x(3)+1*x(4)+20*x(5)))-
log(1+exp(x(1)+0*x(2)+0*x(3)+1*x(4)+43*x(5)))-
log(1+exp(x(1)+0*x(2)+0*x(3)+0*x(4)+18*x(5)))-
log(1+exp(x(1)+0*x(2)+1*x(3)+0*x(4)+25*x(5)))-
log(1+exp(x(1)+1*x(2)+1*x(3)+0*x(4)+37*x(5)))-
log(1+exp(x(1)+1*x(2)+1*x(3)+1*x(4)+17*x(5)))-
log(1+exp(x(1)+0*x(2)+0*x(3)+1*x(4)+22*x(5)))-
log(1+exp(x(1)+0*x(2)+0*x(3)+0*x(4)+38*x(5)))-
log(1+exp(x(1)+1*x(2)+0*x(3)+1*x(4)+36*x(5)))-
log(1+exp(x(1)+0*x(2)+1*x(3)+0*x(4)+45*x(5)))-
log(1+exp(x(1)+1*x(2)+1*x(3)+1*x(4)+1*x(5)))-
log(1+exp(x(1)+0*x(2)+0*x(3)+1*x(4)+5*x(5)))-
log(1+exp(x(1)+1*x(2)+0*x(3)+0*x(4)+43*x(5)))-
log(1+exp(x(1)+0*x(2)+1*x(3)+0*x(4)+33*x(5)))-
log(1+exp(x(1)+0*x(2)+1*x(3)+0*x(4)+29*x(5)))-
log(1+exp(x(1)+0*x(2)+1*x(3)+0*x(4)+47*x(5)))-
log(1+exp(x(1)+1*x(2)+1*x(3)+0*x(4)+15*x(5)))-
log(1+exp(x(1)+0*x(2)+0*x(3)+0*x(4)+14*x(5)))-
log(1+exp(x(1)+0*x(2)+1*x(3)+0*x(4)+13*x(5)))-
log(1+exp(x(1)+0*x(2)+1*x(3)+1*x(4)+29*x(5)))-
log(1+exp(x(1)+1*x(2)+1*x(3)+0*x(4)+33*x(5)))-
log(1+exp(x(1)+0*x(2)+1*x(3)+1*x(4)+26*x(5)))-
log(1+exp(x(1)+0*x(2)+1*x(3)+0*x(4)+42*x(5)))-
log(1+exp(x(1)+0*x(2)+0*x(3)+0*x(4)+12*x(5)))-
log(1+exp(x(1)+0*x(2)+1*x(3)+0*x(4)+29*x(5)))-
log(1+exp(x(1)+1*x(2)+1*x(3)+1*x(4)+43*x(5)))-
log(1+exp(x(1)+1*x(2)+1*x(3)+1*x(4)+40*x(5)))-
log(1+exp(x(1)+0*x(2)+0*x(3)+1*x(4)+41*x(5)))-
log(1+exp(x(1)+1*x(2)+1*x(3)+1*x(4)+43*x(5)))-
```

Ek D. (devam)

```
log(1+exp(x(1)+0*x(2)+1*x(3)+0*x(4)+1*x(5)))-  
log(1+exp(x(1)+0*x(2)+1*x(3)+1*x(4)+13*x(5)))-  
log(1+exp(x(1)+0*x(2)+1*x(3)+1*x(4)+16*x(5)))-  
log(1+exp(x(1)+0*x(2)+0*x(3)+1*x(4)+7*x(5)))-  
log(1+exp(x(1)+1*x(2)+1*x(3)+0*x(4)+36*x(5)))-  
log(1+exp(x(1)+1*x(2)+0*x(3)+1*x(4)+14*x(5)))-  
log(1+exp(x(1)+0*x(2)+1*x(3)+0*x(4)+15*x(5)))-  
log(1+exp(x(1)+1*x(2)+0*x(3)+1*x(4)+1*x(5)))-  
log(1+exp(x(1)+1*x(2)+0*x(3)+1*x(4)+37*x(5)));  
end
```

Ek E. Matlab Yazılımı İle GA Kodları ve Açıklamaları

GA, en genel şekli ile aşağıdaki biçimde kodlanmaktadır.

```
[x,fval,exitflag,output,population,scores]=GA(FITNESSFCN,NVARS,A,b,Aeq,beq,lb,ub,NONLCON,options);
```

Burada eşitliğin sol tarafı çıktı değişkenlerini oluşturmaktadır ve açıklamaları aşağıda verilmiştir.

```
([x,fval,exitflag,output,population,scores]);
```

X :Fonksiyonu en küçük yapan değişken değerlerini yazdırır.

Fval :Amaç fonksiyonun x değişkeni için bulunan optimum değerini yazdırır.

Exitflag :Algoritmanın sonlandırma nedenini gösteren tamsayı değerini yazdırır. Bu tamsayı değeri doğrusal kısıt olmadığı durumlarda -5,-4,-2,-1, 0, 1, 5 değerlerini almaktadır. Örneğin çözüm sonucu ekranda '5' değeri yazdırılmışsa belirlenen zaman sınırının aşıldığı için iterasyonların durdurulduğu anlaşılmaktadır. Aşağıda tüm tamsayı değerlerinin kısa açıklaması verilmiştir.

(1) Optimum sonuç ve değişken değerleri bulundu.

(5) Belirlenen fonksiyon hassaslığı kısıtlama ihlali aşıldı.

(0) Nesillerin sayısı aşıldı.

(-1) Optimizasyon, çıktı veya çizim fonksiyonu tarafından durduruldu.

(-2) Uygun nokta bulunamadı.

(-4) Durma süresi sınırı aşıldı.

(-5) Zaman sınırı aşıldı.

Output :Her jenerasyonda algoritmanın performansı hakkında aşağıdaki bilgileri içeren bir çıktı verir.

Rngstate :Rasgele sayı üreticinin durumu

Generations :Toplam nesil sayısını

Funccount :Toplam fonksiyon sayısını

Maxconstraint :Maximum kısıtlama ihlali

Message :Sonlandırma mesajı

Population :Çözüm için oluşturulan popülasyonların çıktısını verir.

Scores :Optimizasyon sonuç değerini yazdırır.

Eşitliğin sağ tarafındaki girdi değişkenlerinin açıklamaları ise aşağıdadır.

```
GA(FITNESSFCN,NVARS,A,b,Aeq,beq,lb,ub,NONLCON,options);
```

@fitnessfcn : m dosyası olarak kaydedilen amaç fonksiyonunu çağırır.

Ek E. (devam)

Nvars	:Amaç fonksiyonundaki bağımsız değişken sayısı belirtilir.
A	:Eşitsizlik kısıtlamaları matrisi
B	:Eşitsizlik kısıtlamaları vektörü
Aeq	:Eşitlik kısıtlamaları matrisi
Beq	:Eşitlik kısıtlamaları matrisi
Lb	:Alt sınır
Ub	:Üst sınır
Nonlcon	:Doğrusal olmayan kısıt fonksiyonu
Options	:‘gaoptimset’ ile oluşturulan seçeneklerin yapısını belirler. Kodunun detaylı kullanım şekli aşağıdaki gibidir.

options = gaoptimset('param1',value1,'param2',value2,...).

‘gaoptimset’ ten sonra iki tırnak içerisinde yazılan tüm parametre değişkenleri için virgülden sonra yazılan değere göre çözümün aranmasını sağlar. GA’daki en önemli kodlardan biridir. Optimum çözüm için belirleyeceğimiz mutasyon, seçim, çaprazlama gibi operatörlerin seçimleri, ‘options’ kodu ile yapılmaktadır.

GA’da Kullanılan Önemli Kodlar ve Tanımları

Option	Description	Values
CreationFcn	Handle to the function that creates the initial populations	@gacreationlinearfeasible @gacreationuniform
CrossoverFcn	Handle to the function that the algorithm uses to create crossover children	@crossoverheuristic @crossoverintermediate @crossoversinglepoint @crossovertwopoint @crossoverarithmetic @crossoverscattered
CrossoverFraction	The fraction of the population at the next generation, not including elite children, that is created by the crossover function	positive scalar {0.8}
Display	Level of display	'off' 'iter' 'diagnose' 'final'
FitnessLimit	Scalar. If the fitness function attains the value of FitnessLimit, the algorithm halts	scalar {-Inf}
Generations	Positive integer specifying the maximum number of iterations before the algorithm halts	positive integer {100}
InitialPopulation	Initial population used to seed the genetic algorithm; can be partial	Matrix {[]}
InitialScores	Initial scores used to determine fitness; can be partial	column vector {[]}

Ek E. (devam)

MigrationDirection	Direction of migration	'both' 'forward'
MutationFcn	Handle to the function that produces mutation children	@mutationuniform @mutationadaptfeasible @mutationgaussian
PlotFcns	Array of handles to functions that plot data computed by the algorithm	@gaplotbestf, @gaplotbestindiv, @gaplotexpectation, @gaplotstopping
PopulationSize	Size of the population	positive integer {20}
PopulationType	String describing the data type of the population	'bitstring' 'custom' 'doubleVector'
SelectionFcn	Handle to the function that selects parents of crossover and mutation children	@selectionremainder @selectionuniform @selectionroulette @selectiontournament @selectionstochunif
StallGenLimit	Positive integer. The algorithm stops if there is no improvement in the objective function for StallGenLimit consecutive generations	positive integer {50}
TolCon	Positive scalar. Tolcon is used to determine the feasibility with respect to nonlinear constraints	positive scalar {1e-6}
TolFun	Positive scalar. The algorithm runs until the cumulative change in the fitness function value over StallGenLimit is less than TolFun	positive scalar {1e-6}

Şimdi çalışmamızda kullandığımız tüm kodların açıklamalarını yapalım. ‘olabilirlik’ ismi ile verilen fonksiyon çalışmanın uygulama bölümünde maksimize edilecek log-olabilirlik fonksiyonudur ve amaç fonksiyonu olarak alınmaktadır. Bu fonksiyon kullanılarak öncelikle Matlab Paket programında ‘olabilirlik’ ismiyle m dosyası olarak EK.B’deki gibi tanımlanmıştır.

Amaç fonksiyonu tanımlandıktan sonra kod yazımı için kullanılan “command window” ekranına aşağıdaki parametre tahminlerini ve optimum sonucu veren kodlar girilmiştir.

Ek E. (devam)

Kodları iki bölüm gibi düşünebiliriz.

1.bölüm

```
options=gaoptimset('StallGenLimit',50000,'PopulationSize',50,  
'SelectionFcn',{ @selectionroulette}, 'CrossoverFraction',0.95,  
'MutationFcn',{ @mutationgaussian,0.15,0.88}, 'CrossoverFcn',{ @crossoverinte  
rmediate, 1.0}, 'PlotFcns', @gaplotbestf, 'Generations',50000, 'TolFun',1e-  
6, 'TolCon',1e-6, 'Display','iter', 'Display','final', 'Display','diagnose');
```

Bu bölümde,

**'PopulationSize'dan sonra virgüllerin arasında 50 yazılması, başlangıç popülasyonunun büyüklüğünün 50 olacağını belirtir. Matlab'de popülasyon büyüklüğü için varsayılan değer ise 20'dir.

**'SelectionFcn' ile Seçim operatörü 'roulette' seçilmiştir.

**'CrossoverFraction' ile üreme çaprazlama oranı 0.95 seçilmiştir.

**'MutationFcn' ile mutasyon operatörü 'gaussian' seçilmiştir. Mutasyon oranı 0.15 ve küçültme oranı 0.88 seçilmiştir.

**'CrossoverFcn' ile çaprazlama fonksiyonu 'intermediate' seçilmiştir. Çaprazlama oranı ise 1.0 olarak devamında yazılmıştır.

**'PlotFcns' sonrasında yazılan @gaplotbestf kodu ile çözümün grafiği çizdirilmiştir.

Son olarak kısıtlama kodları yazılmıştır.

**'StallGenLimit' ve 'Generations' kodları ile ardışık nesiller için gelişme olmadığında ve 50000 iterasyon aşıldığında iterasyonların durdurulacağı belirtilmiştir.

**'TolFun' ve 'TolCon' kodları ile yakınsama ölçüsü olarak iki iterasyon değeri arasındaki fark için 1×10^{-6} (1e-6) kullanılmıştır.

**'Display'den sonra yazılan 'iter', 'final', 'diagnose' kodları, optimizasyonun sırasıyla tüm iterasyonlarını, sonuç değerini ve çözüme ulaşırken bulunan tüm aşamalardaki başlangıç popülasyonunu, amaç fonksiyonun değerlerini yazdırmayı sağlar.

Ek E. (devam)

2.bölüm

```
[x,fval,exitFlag,output,population,scores] = ga(@olabilirlik,13,[],[],[],[],[],[],[],[],[],options)
```

Bu bölümde sol tarafta yazılan kodlar, sırasıyla fonksiyonu en küçük yapan değişken değeri, fonksiyonun optimum değerini ve popülasyonların yazdırılmasını sağlamaktadır. Kodlamada bulunmayan diğer operatörler ve oranları, varsayan olarak işlem görmektedir.

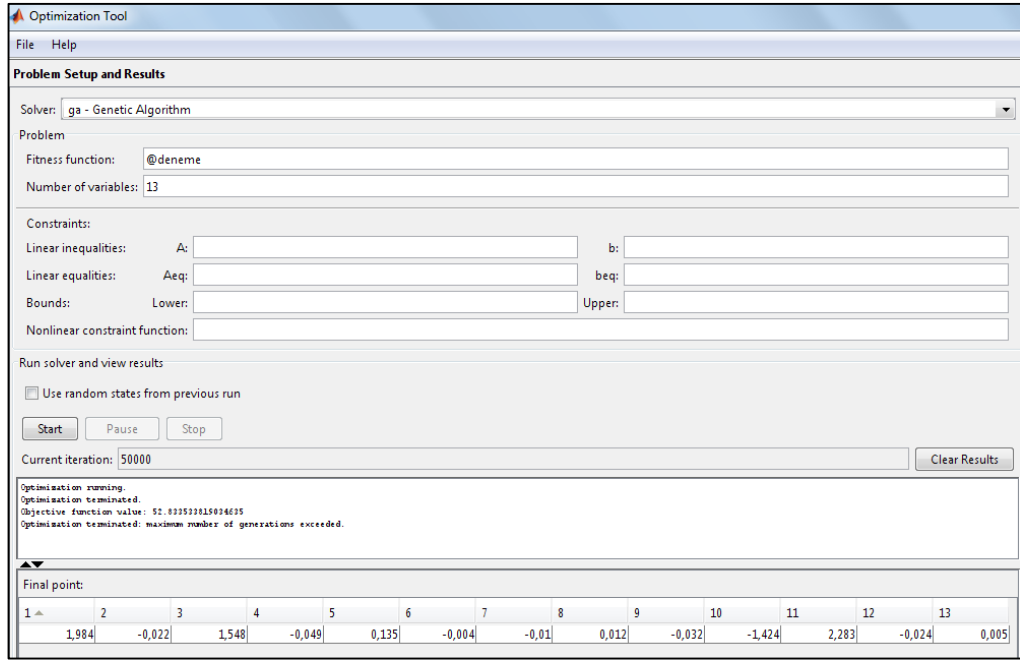
Çalışmamızda GA ile çözüm yapılırken birden çok başlangıç popülasyonu, çaprazlama ve mutasyon oranları ve farklı seçim yöntemleri denenmiştir. Optimum sonucun bulunduğu yukarıda verilen kodlarda kullanılan parametrelerin özellikleri aşağıda detaylı olarak verilmiştir.

Kodlamada Kullanılan Parametreler

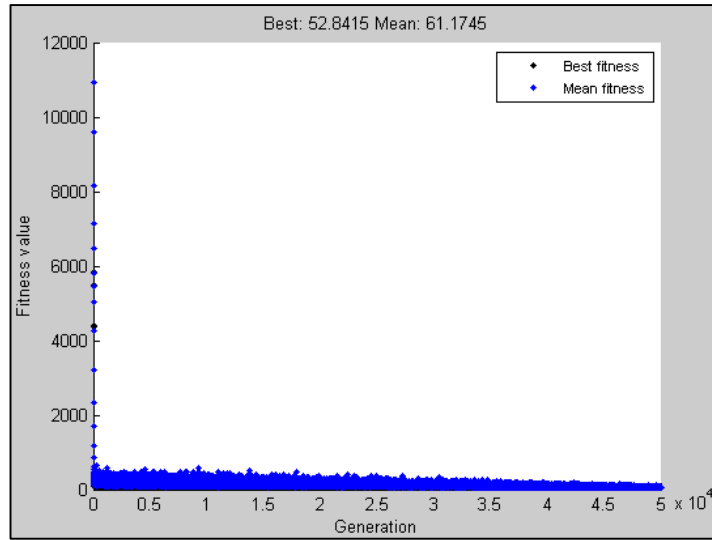
GA parametresi	Değer / Metod
PopulationType	'doubleVector'
PopulationSize	50
Selection Function	Roulette
Reproduction	0.95
Crossover fraction	
Mutation function	Gaussian
Mutation ratio	0.15
Mutation shrink	0.88
CrossoverFcn	Crossoverintermediate
Crossover ratio	1.0
Migration Direction	'forward'
Others (InitialPopulation, InitialScores et al.)	Use default
Stopping criteria	
Generations	50000
StallGenLimit	50000
TolFun	1e-6
TolCon	1e-6
Output	
PlotFcns	@gplotbestf
Display	'iter','final','diagnose'
Others	Use default

Optimum sonucun ve parametrelerin bulunduğu matlab sonuç penceresinin ekran görüntüsü ve çözüme ulaşırken çizilen grafik aşağıda verilmiştir.

Ek E. (devam)



Matlab ile GA Optimizasyon sonuç ekranı



Matlab ile GA iterasyonları (50000 iterasyon)

Şekil incelendiğinde, 50000 iterasyon sonucunda amaç fonksiyonunun hangi değerlerden başladığı, hangi sonuçlara ulaştığı ve arama noktalarının serpilmeleri daha açık görülmektedir. Amaç fonksiyonun değerleri ilk iterasyonlarda 100 civarında olduğu daha sonra ise genetik operatör işlemleri sonucunda fonksiyonun optimizasyon değeri olan 50-55 aralığına yaklaştığı gözlenmektedir.

Ek F. Matlab İle Nelder-Mead Kodları ve Açıklamaları

Çıktı değişkenleri [x,fval,exitflag,output]

X : Fonksiyonu en küçük yapan nihai bağımsız değişken değerlerini yazdırır.

Fval : Uygunluk fonksiyonun x değişkeni için bulunan optimum değerini yazdırır.

Exitflag : Algoritmanın sonlandırma nedenini gösteren tamsayı değerini yazdırır. Bu tamsayı değeri -1, 0, 1, değerlerini almaktadır. Örneğin çözüm sonucu ekranda '0' değeri yazdırılmışsa belirlenen hassaslık yada yineleme sayısı sınırının aşıldığı için iterasyonların durdurulduğu anlaşılmaktadır. Aşağıda tüm tamsayı değerlerinin kısa açıklaması verilmiştir.

(1) Optimum sonuç ve değişken değerleri bulundu. X gibi bir çözüme yaklaşıldı.

(0) Fonksiyon değerlendirme yada maksimum yineleme sayısı aşıldı.

(-1) Algoritma çıkış fonksiyonu tarafından sonlandırıldı

Output : Algoritmanın performansı hakkında aşağıdaki bilgileri içeren bir çıktı verir.

output.algorithm : Kullanılan Algoritmayı yazdırır.

output.funcCount : Fonksiyon değerlendirme sayısını yazdırır.

output.iterations: İterasyon sayısını yazdırır.

output.message : Çıkış mesajını yazdırır.

Girdi değişkenleri fminsearch(fun,x0,options);

Fun : Minimize edilen m fonksiyonunu çağırır.

x0 : Optimizasyon için başlangıç değerlerini girilmesini sağlar

options : Varsayılan optimizasyon parametreleri yerine belirlenen yapı ile optimizasyonun yapılmasını sağlar. Optimizasyon seçeneklerinin değiştirilmesini sağlar. Bu seçenekler Display, TolX, TolFun, MaxFunEvals, MaxIter, FunValCheck, PlotFncs, ve OutputFcn' den oluşmaktadır.

Display : 'off' çıktının yazdırılmayacağını 'iter' her tekrarda iterasyonların yazdırılacağını, 'final' sadece nihai sonucun yazdırılacağını, 'notify' sadece fonksiyon yakınsamadığı zaman çıktıları yazdıracağını belirtir.

FunValCheck : Amaç fonksiyonu değerlerinin geçerli olup olmadığını kontrol eder.

MaxFunEvals : İzin verilen maksimum fonksiyon değerlendirme sayısı

MaxIter : İzin verilen maksimumu iterasyon sayısı

OutputFcn : Her iterasyonda tanımlı fonksiyonu yazdırır.

Ek F. (devam)

- PlotFcns : çözümün grafiğini çizdirir.
TolFun : Fonksiyon için sonlandırma toleransı
TolX : X için Sonlandırma toleransı

Yapay Veri Optimization Tool Seçenekleri

The screenshot shows the Optimization Tool interface with the following settings:

- Solver:** ga - Genetic Algorithm
- Problem:** Fitness function: @emre, Number of variables: 5
- Constraints:** Linear inequalities, Linear equalities, Bounds, Nonlinear constraint function
- Options:** Stall generations: Specify: 20000, Stall time limit: Use default: Inf, Function tolerance: Use default: 1e-6, Nonlinear constraint tolerance: Use default: 1e-6
- Plot functions:** Plot interval: 1, Best fitness, Best individual, Distance, Expectation, Genealogy, Range, Score diversity, Scores, Selection, Stopping, Max constraint, Custom function
- Output function:** History to new window Interval: 1, Custom function
- Display to command window:** Level of display: off

Run solver and view results:

Use random states from previous run

Start Pause Stop

Current iteration: 20000 Clear Results

Optimization running.
Optimization terminated.
Objective function value: 32.482854664707024
Optimization terminated: maximum number of generations exceeded.

Final point:

1	2	3	4	5
0,647	0,313	-1,18	-0,037	-0,003

The screenshot shows the Optimization Tool interface with the following settings:

- Solver:** fminsearch - Unconstrained nonlinear minimization
- Problem:** Objective function: @emre, Start point: [-3,3;-3,3;-3,3;-3,3]
- Options:** Stopping criteria: Max iterations: Use default: 200*numberOfVariables, Max function evaluations: Use default: 200*numberOfVariables, X tolerance: Use default: 1e-4, Function tolerance: Use default: 1e-4
- Function value check:** Error if user-supplied function returns NaN or complex
- Plot functions:** Current point, Function count, Function value
- Output function:** Custom function
- Display to command window:** Level of display: off

Run solver and view results:

Start Pause Stop

Current iteration: 5146 Clear Results

Optimization running.
Optimization terminated.
Objective function value: 32.48285466470697
Optimization terminated:
the current x satisfies the termination criteria using OPTIONS.TolX of 1.000000e-004
and F(X) satisfies the convergence criteria using OPTIONS.TolFun of 1.000000e-004

Final point:

1	2	3	4	5
0,647	0,313	-1,18	-0,037	-0,003

Ek F. (devam)

Optimization Tool

File Help

Problem Setup and Results

Solver: **fminsearch - Unconstrained nonlinear minimization**

Problem

Objective function: **@emre**

Start point: **[-1,1,-1,1,-1,1,-1,1]**

Run solver and view results

Current iteration: **1269**

Optimization running.
 Optimization terminated.
 Objective function value: 32.48285466470732
 Optimization terminated:
 the current x satisfies the termination criteria using OPTIONS.TolX of 1.000000e-004
 and F(X) satisfies the convergence criteria using OPTIONS.TolFun of 1.000000e-004

Final point:

1	2	3	4	5
0,647	0,313	-1,18	-0,037	-0,003

Options

Stopping criteria

Max iterations: Use default: 200*numberOfVariables
 Specify:

Max function evaluations: Use default: 200*numberOfVariables
 Specify:

X tolerance: Use default: 1e-4
 Specify:

Function tolerance: Use default: 1e-4
 Specify:

Function value check

Error if user-supplied function returns NaN or complex

Plot functions

Current point Function count Function value

Custom function:

Output function

Custom function:

Display to command window

Level of display: **off**

Optimization Tool

File Help

Problem Setup and Results

Solver: **fminsearch - Unconstrained nonlinear minimization**

Problem

Objective function: **@emre**

Start point: **[-0,0,-0,0,-0,0,-0,0]**

Run solver and view results

Current iteration: **453**

Optimization running.
 Optimization terminated.
 Objective function value: 32.48285466876738
 Optimization terminated:
 the current x satisfies the termination criteria using OPTIONS.TolX of 1.000000e-004
 and F(X) satisfies the convergence criteria using OPTIONS.TolFun of 1.000000e-004

Final point:

1	2	3	4	5
0,647	0,313	-1,18	-0,037	-0,003

Options

Stopping criteria

Max iterations: Use default: 200*numberOfVariables
 Specify:

Max function evaluations: Use default: 200*numberOfVariables
 Specify:

X tolerance: Use default: 1e-4
 Specify:

Function tolerance: Use default: 1e-4
 Specify:

Function value check

Error if user-supplied function returns NaN or complex

Plot functions

Current point Function count Function value

Custom function:

Output function

Custom function:

Display to command window

Level of display: **off**

ÖZGEÇMİŞ

Kişisel Bilgiler

Ad Soyad : Emre DEMİR
Uyruk : T.C.
Doğum Yeri ve Tarihi: 18/06/1982
Medeni Hali : Evli
Telefon : 0 505 200 6502
E-posta : emredemir@hitit.edu.tr

Eğitim

Alınan Derece	Aldığı Kurum/Üniversite	Mezuniyet Yılı
Lise	Anadolu Öğretmen Lisesi	2000
Lisans	Karadeniz Teknik Üniversitesi	2005
Yüksek Lisans Derecesi	Karadeniz Teknik Üniversitesi	2005
Yüksek Lisans	Muğla Sıtkı Koçman Üniversitesi	2014

İş Tecrübesi

Yıl	Yer	Pozisyon/görev
2005-2009	Çorum Özel Nokta Dershanesi	Öğretmen
2009-2010	Amasya Uğur Dershaneleri	Öğretmen
2010-2012	Muğla Üniversitesi	İdari
2012-	Hitit Üniversitesi	Öğretim Görevlisi

Yabancı Dil(ler)

Dil (İngilizce, vs)	Başlangıç	Orta	İleri
Yazma		X	
Konuşma		X	
Anlama		X	
Okuma		X	

Bilimsel Faaliyetler

Uluslararası bilimsel toplantılarda sunulan bildiriler

1- DEMİR E. and DEMİR S., (2013) *Effectiveness Measurement of High Schools in Çorum by Fuzzy Data Envelopment Analysis*, 8. Uluslararası İstatistik Kongresi, 27-30 Ekim, Kemer-Antalya, s137-138.

2- DEMİR E. and AKKUŞ Ö., (2014) *Genetic Algorithm Approach to the Parameter Estimation in the Analysis of Categorical Dependent Variable Models*, 9th International Statistics Day Symposium, 10-14 May, Side-Antalya.

3- DEMİR E. and AKKUŞ Ö., (2014) *Bağımlı Değişkenin Kategorik Olduğu Modellerde Newton-Raphson Parametre Tahminlerinin Nelder-Mead Ve Genetik Algoritma Tahminleri İle Karşılaştırılması*, 15. Uluslararası Ekonometri, Yöneylem Araştırması ve İstatistik Sempozyumu, 22-25 Mayıs, SDÜ-Isparta.

Yayınlar

1- DEMİR E. ve DURAKOĞLU M. (2013) Çorum İlindeki Liselerin 2012-2013 Eğitim Öğretim Sürecindeki Etkinliğinin Veri Zarflama Analizi İle Ölçülmesi, *Hitit Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, Yıl 6, Sayı 1, s.19-42.

2- ŞENEL E., SABANCILAR E., MANSUROĞLU C. and DEMİR E. (2014) A preliminary study of the contribution of telemicroscopy to the diagnosis and management of skin tumours in teledermatology, *Journal of Telemedicine and Telecare*, 20(4), s.178–183.