



Effects of the stochastic and deterministic movements in the optimization processes

Ebubekir Seyyarer^{1,*} , Ali Karcı² , Abdullah Ateş² 

¹Department of Computer Programming, Van Yüzüncü Yıl University, Van, 65080, Turkey

²Department of Computer Engineering, İnönü University, Malatya, 44000, Turkey

Highlights:

- Stochastic and deterministic initial populations
- Multivariate linear regression model
- Obtaining and comparing results with gradient based optimization algorithms

Keywords:

- Deterministic initial population
- stochastic initial population
- multivariate linear regression
- optimization algorithms
- iris data set

Article Info:

Research Article
Received: 01.03.2021
Accepted: 29.08.2021

DOI:

10.17341/gazimmfd.887976

Correspondence:

Author: Ebubekir Seyyarer
e-mail: eseyyarer@yyu.edu.tr
phone: +90 432 612 2434

Graphical/Tabular Abstract

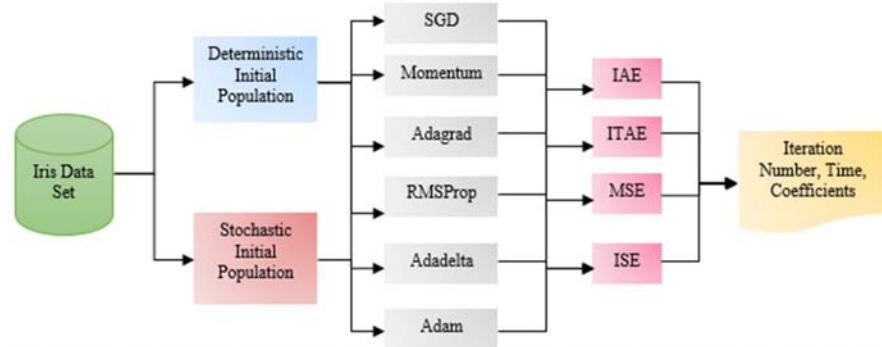


Figure A. All modules of the study

Purpose: The aim of this study is to show the effect of a deterministic initialization method suggested to be used instead of random start method which is still used in optimization algorithms.

Theory and Methods:

This study focuses on the topic of initial population in optimization algorithms. Outputs of stochastic and deterministic initial populations are compared. Proposed deterministic initialization method reaches the global optimum much faster, which can be given as an important advantage. Multivariate linear regression (MLR) model is used as the model and iris dataset is used as the dataset. Six gradient descent based optimization methods (Stochastic Gradient Descent (SGD), Momentum, Adagrad, RMSProp, Adadelta ve Adam) are used to minimize the error rate and four error functions (integral of the absolute value of the error (IAE), integral of the time-weighted absolute error (ITAE), Mean Square of the Error (MSE) and integral of the square error (ISE)) are used as objective functions.

Results:

It is sufficient to run the application with initial population developed using the proposed deterministic method once. Outputs of each application are the same because it is run with fixed values. However, Stochastic initialization methods must be run at least 10 times before the outputs can be presented to literature. It is observed that the outputs of deterministic and stochastic methods are equal. However, in terms of operation time, the deterministic method result in an improvement by 90%.

Conclusion:

According to deterministic and stochastic initialization methods obtained coefficients and iteration numbers are found to be close. However, temporal gain is achieved from the application that is initialized deterministic. According to comparisons, the linear model obtain using the Adadelta optimization algorithm and the MSE objective function perform best.



Stokastik ve deterministik hareketlerin optimizasyon süreçlerindeki etkileri

Ebubekir Seyyarer^{1*}, Ali Karci², Abdullah Ateş²

¹Van Yüzüncü Yıl Üniversitesi, Gevaş Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü, 65080, Van, Türkiye

²İnönü Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 44000, Malatya, Türkiye

Ö N E Ç İ K A N L A R

- Stokastik ve deterministik başlangıç popülasyonları
- Çok değişkenli lineer regresyon modeli
- Türeve dayalı optimizasyon algoritmaları ile sonuçların elde edilmesi ve karşılaştırılması

Makale Bilgileri

Araştırma Makalesi

Geliş: 01.03.2021

Kabul: 29.08.2021

DOI:

10.17341/gazimmfd.887976

Anahtar Kelimeler:

Deterministik başlangıç popülasyonları, stokastik başlangıç popülasyonları, çok değişkenli lineer regresyon, optimizasyon algoritmaları, iris veri seti

ÖZ

Bu çalışmada Çok Değişkenli Lineer Regresyon (Multivariate Linear Regression-MLR) modeli kullanarak iris veri setini temsil eden lineer bir fonksiyon elde edilmektedir. Bu fonksiyondaki katsayıların optimum değerlerini bulmak için Stokastik Dik İniş (Stochastic Gradient Descent-SGD), Momentum, Adagrad, RMSProp, Adadelta ve Adam optimizasyon algoritmaları kullanılmaktadır. MLR'lerde genelde sabit veya rastgele bir değer ile başlatılan bu katsayılar için başlangıç popülasyonuna sahip bir başlatma yöntemi önerilmektedir. Kullanılan MLR modelinde amaç fonksiyonu olarak Mutlak Hata Değeri (Integral Absolute Error-IAE), Zaman Ağırlıklı Mutlak Hata Değeri (Integral Time-Weighted Absolute Error-ITAE), Ortalama Karesel Hata Değeri (Mean Square Error-MSE) ve Karesel Hata Değeri (Integral Square Error-ISE) hata fonksiyonlarından yararlanılmaktadır. Alt ve üst sınırlar arasında önerilen bir deterministik ve klasik stokastik başlatma yöntemleri ile uygulamaların başlangıç popülasyonları oluşturulmaktadır. Stokastik olarak başlatılan uygulama literatürdeki gibi birçok defa çalıştırılıp ortalama değerleri alınmaktadır. Buna karşılık deterministik olarak başlatılan uygulama ise bir kez çalıştırılmaktadır. Deterministik ve stokastik başlatma yöntemlerinin sonuçlarına göre iki uygulamada da elde edilen katsayıların ve adım sayılarının birbirine yakın olduğu gözlemlenmektedir. Buna rağmen deterministik olarak başlatılan uygulamada çok yüksek zamansal kazanım elde edilmektedir. Karşılaştırmalar sonucunda Adadelta optimizasyon algoritması ve MSE amaç fonksiyonuyla elde edilen lineer model en kısa zamanda sonuca ulaşmaktadır.

Effects of the stochastic and deterministic movements in the optimization processes

H I G H L I G H T S

- Stochastic and deterministic initial populations
- Multivariate linear regression model
- Obtaining and comparing results with gradient based optimization algorithms

Article Info

Research Article

Received: 01.03.2021

Accepted: 29.08.2021

DOI:

10.17341/gazimmfd.887976

Keywords:

Deterministic initial population, stochastic initial population, multivariate linear regression, optimization algorithms, iris data set

ABSTRACT

In this study, a linear function representing the iris data set is obtained by making use of the MLR model. SGD, Momentum, Adagrad, RMSProp, Adadelta and Adam optimization algorithms are used to find the optimum values of coefficients of this function. An initialization method with initial population is recommended for these coefficients, which are generally initialized with a fixed or random value in MLRs. IAE, ITAE, MSE and ISE error functions are used as objective functions in the MLR model used. Initial populations of the methods are developed by using a proposed deterministic and classical stochastic initialization methods between upper and lower bounds. The method that are initialized stochastically is run several times as seen in literature and the mean values are calculated. On the other hand, the application that is initialized deterministic is only run once. According to the results of deterministic and stochastic initialization methods, it is observed that the coefficients and iteration numbers obtained in both applications are close to each other. Despite very high temporal gain is achieved from the application that is initialized deterministic. As a result of the comparisons, the linear model obtained with Adadelta and MSE reaches the result in the shortest time.

*Sorumlu Yazar/Yazarlar / Corresponding Author/Authors : *eseyyarer@yyu.edu.tr, ali.karci@inonu.edu.tr, abduallah.ates@inonu.edu.tr /

Tel: +90 432 612 2434

1. GİRİŞ (INTRODUCTION)

Kelime anlamı “eniyeleme” olan optimizasyon, bir problemin en uygun çözümünü bulmak veya bulmaya çalışmak anlamına gelmektedir [1, 2]. Tüm alanlarda optimize edilecek problemler mevcuttur [3, 4]. Örneğin mühendisler en iyi performansı, üreticiler maksimum verimliliği ve yatırımcılar ise aşırı riskten kaçınmak istedikleri için yaptıkları işleri optimize ederler [5]. Benzer şekilde doğanın kendisi de minimum enerji için optimizasyon yapmaktadır [5]. En iyi performans, maksimum verimlilik, aşırı riskten kaçınmak ve minimum enerji gibi örnekler optimizasyonda “amaçlar” olarak adlandırılmaktadır [6]. Üçten fazla amacı olan optimizasyon problemlerine Çok Amaçlı Optimizasyon Problemleri denilmektedir [6]. Optimizasyon problemlerini çözmek için kullanılan yöntemler Kesin (Tam) metotlar (Analitik yöntemler) ve Yaklaşık metotlar olmak üzere iki grupta kategorize edilmektedir. Yaklaşık metotlar da Sezgisel ve Meta-Sezgisel yöntemler olarak ikiye ayrılmaktadır [5, 6].

Tam metotlar olarak da adlandırılan Kesin metotlar; sezgisel olmayan, deterministik ve sistematik optimizasyon algoritmalarının özel bir türüdür [1]. Amaç fonksiyonlarıyla ilgili ön bilgiler yeteriyse global optimumu bulmayı makul miktarda kaynak kullanarak garanti edebilmektedir [2–4]. Bu kaynaklara örnek olarak hesaplama süresi veya fonksiyon değerlendirmeleri verilebilir. Kesin metotların probleme uygulanabilirliği varsa sezgisel yöntemlerden daha güvenilir olmaktadır [7]. Ayrıca bu metotları teoride kara kutu problemlerine uygulamak mümkündür. Çünkü sonlu sürede genel optimumu bulma yeteneği mevcuttur [8]. Ancak masraflı ve oldukça fazla süren bir arama gerekmektedir. Bu nedenle hesaplama süreleri üstel olmaktadır ve bazı uygulamalar dışında tercih edilmemektedirler [9, 10].

Yaklaşık yöntemlerin ilki olan sezgisel algoritmalar; optimizasyon problemleri için hedeflenen çözümleri genellikle “deneme yanılma” yöntemiyle makul bir hesaplama süresi içinde bulmayı amaçlamaktadırlar [1–3]. “Sezgisel” kelimesi “aramak” veya “bulmak” anlamlarına gelmektedir. Sezgisel yöntemler en uygun çözümü bulmayı garanti etmezler [4]. Kısmen uygun olan bir çözüm bulurlar ve genellikle yetersiz kalırlar. Sezgisel yöntemlere yerel arama yöntemleri de denilmektedir [7]. Parametre aramaları sezgisel yöntemlerle bulunmaktadır. Ayrıca zaman kısıtlarıyla yerel bölgelerde bulunan uygun çözümler, günümüzdeki birçok optimizasyon problemi için yeterli kabul edilmektedirler [8–10].

Diğer bir yaklaşık yöntem olan meta sezgisel algoritmalar, üst düzey sezgisel algoritmalar olarak bilinmektedir. “meta” kelimesi “ötesi” veya “üst düzey” anlamlarına gelmektedir. Bu nedenle meta-sezgisel, sezgisel yöntemlerdeki deneme-yanılma süreçlerinin yanı sıra uygun çözümlerin üst düzey teknikler kullanarak bulunması anlamına gelmektedir [1, 2]. Başka bir ifadeyle, parametre aramalarında büyük alanın taranması için düşük seviyeli yöntemlerin birleştirilmesiyle

elde edilen üst düzey teknikler olarak kabul edilmektedirler [3, 4]. “meta-sezgisel” kelimesi Fidan Gelişim Algoritması (FGA), Simüle Tavlama, Yapay Sınır Ağları, Genetik Algoritmalar, Tabu Araması, Karınca Kolonisi Optimizasyonu ve Parçacık Sürü Optimizasyonu gibi tüm modern üst düzey algoritmalarını ifade etmektedir [7–9]. Bu yöntemler, klasik sezgisel yöntemlere rehberlik etmektedir [10–12].

Literatürde çok sayıda optimizasyon yöntemi bulunmaktadır [13–15]. Optimizasyon algoritmalarında popülasyon başlatma teknikleri araştırılması gereken bir konu haline gelmiştir [16, 17]. Problem çözümünde iyi başlatma yöntemlerinin optimum çözümlere daha hızlı yakınsamasına sebep olduğu görülmekte, sonuçları önemli ölçüde iyileştirebileceğini göstermektedir [18, 19].

Musrrat Ali vd., Farksal Gelişim (Differential Evolution-DE) algoritmasının başlangıç popülasyonunu iki ayrı yöntemle (Doğrusal Olmayan Simpleks DE (Nonlinear Simplex DE-NSDE) ve Eğrisel Yaklaşım DE (Quadratic Interpolation DE-QIDE)) başlatarak temel DE ile yola devam etmektedirler. NSDE ve QIDE’nin temel DE’ye göre işlemcinin üzerindeki iyileştirme oranları sırasıyla %18 ve %14’tür [20]. Kazimipour vd., 2014 tarihine kadar popülasyon başlatma teknikleri ile ilgili yapılan çalışmalar hakkında detaylı bir inceleme yapmışlardır. İncelenen çalışmalar temel alınarak başlatma teknikleri; rastgelelik, kompozisyonellik ve genellilik olmak üzere üç ana kategoriye ayrılmıştır ve her bir ana kategori ise alt kategorilere ayrılmıştır [21].

Kazemzadeh azad, çelik kafes boyutlandırma problemi için üç metasezgisel (Adaptif Boyutlu Arama, Değiştirilmiş Büyük Patlama - Büyük Çöküş ve Üssel Büyük Patlama - Büyük Çöküş) yöntem kullanmıştır. Uygulanabilir en hafif tasarımı elde etmek için optimizasyon yöntemlerinin ilk popülasyonunu seçme işlemine odaklanmıştır. Optimizasyonların hesaplama sürelerinde iyileşme gözlemlenmiştir [22].

Zhang vd., Esnek Atölye Çizelgeleme Problemi için geçici üretim zamanını en aza indirmek istemişlerdir. Genel Seçim, Yerel Seçim ve Rastgele Seçim isimli üçlü bir karışım ile başlangıç popülasyonu oluşturulmaktadır. Etkin bir başlangıç popülasyonunun başarı oranını pozitif yönde arttırdığı gözlemlenmiştir [23]. Tometzki vd., iki aşamalı stokastik karışık tamsayı problemleri çözmede kullanılan evrimsel algoritmalara üç tane başlatma yaklaşımı sunmaktadırlar. Birincisi Klasik Rastgele Başlatma, ikincisi Tamsayı Gevşetmesi, üçüncüsü ise İkinci-Aşama Tamsayı Gevşetmesi’dir. Önerilen yöntemlerin başarılı olduğu gözlemlenmiştir [24].

Dong vd., çember tespit için yeni bir Kaotik Hibrit Algoritma (Chaotic Hybrid Algorithm-CHA) önermektedirler. Bu algoritma Genetik Algoritma ve Parçacık Sürüsü Optimizasyonu Algoritmasının birleştirilmesinden meydana

gelmektedir. CHA'nın ilk popülasyonu için Muhalefet Temelli Öğrenme (Opposition Based Learning-OBL) önerilmektedir. Yapılan çalışmaların sonuçlarına bakıldığında OBL'nin oldukça genel olduğu, algoritmaları hızlandırdığı ve çeşitli optimizasyon algoritmalarında kullanılabilceği görülmektedir [25].

Makine Öğrenmesi alanında kullanılan optimizasyon yöntemleri işlem esnasında oluşan hatayı minimum seviye indirmektedir [13]. Dik İniş tabanlı olan bu yöntemler işlemleri adım adım yapmaktadırlar. Adım miktarına Öğrenme Katsayısı denilmektedir ve sonuca ulaşırken etkisi çok büyüktür. Uygun bir değer seçilmesi gerekmektedir. Öğrenme Katsayısı'nın küçük seçilmesi halinde çözüme ulaşılması çok fazla zaman almaktadır, büyük seçilmesinde ise minimum noktanın bulunamaması ihtimaline sebep olmaktadır. Bu durum bir vadinin en yüksek yerinden başlayıp en alt noktasına ulaşmak gibi düşünülebilir [15]. Adım miktarı küçük seçilirse en alt noktaya ulaşmak için çok fazla zaman harcanır. Adım miktarı büyük seçilirse tepeden tepeye geçilebilir ve en alt nokta bulunmayabilir. Dik İniş tabanlı en popüler optimizasyon yöntemleri SGD, Momentum, Adagrad, RMSProp, Adadelta ve Adam algoritmalarıdır [17, 26].

Vani vd., Hint Pines veri seti üzerinde Konvolüsyonel Sinir Ağı (Convolutional Neural Network-CNN) yöntemini kullanarak sınıflandırma yapmışlardır. CNN'deki hatayı düşürmek için SGD, RMSProp, Adam, Adamax, Adagrad, Adadelta ve Nadam gibi optimizasyon yöntemleri kullanmaktadırlar. %99,58 başarı oranı ile Adamax yöntemi en iyi sonucu vermektedir [27].

Rajakumaran vd., Servis Hizmet Reddi saldırılarının tespiti için MLR kullanmışlardır ve MSE hata fonksiyonundan yararlanmışlardır. %3,3 olan hata oranını SGD optimizasyon yöntemi kullanarak %0,3'e düşürmüşlerdir [28].

Dogo vd., üç ayrı veri sete (Cats and Dogs, Fashion Mnist, Natural Images) CNN uygulamışlardır. Hata oranını azaltmak için yedi ayrı Dik İniş tabanlı optimizasyon yöntemi (SGD, vSGD, SGDm, SGDm+n, RMSProp, Adam, AdaGrad, AdaDelta, Adamax, Nadam) kullanmışlardır. En iyi sonucu veren Nadam en kötü sonucu veren ise adadelta olmuştur [29].

Yazan vd., parametre değerlerini güncellemek için beş ayrı Dik İniş tabanlı optimizasyon yöntemi (Momentum, AdaGrad, RMSProp, AdaDelta, Adam) kullanmaktadırlar. Bu yöntemler Griewank fonksiyonunda test edilmiştir. Adadelta en başarılı yöntem olarak gözlemlenmiştir [30]. Makine Öğrenme yöntemleri olan Regresyon ve Sınıflandırma günümüzde oldukça popülerdirler [15]. Regresyon, sayısal (sürekli) verilerin tahmini için kullanılmaktadır. Sınıflandırma ise kategorik (ayrık) yani sayısal olmayan verilerin tahmininde kullanılmaktadır [16, 17]. Regresyon bir veya birden fazla bağımsız değişken üzerinden bağımlı bir değişkenin davranışını modellemektir. Örneğin, bir kişinin cinsiyeti, eğitim durumu ve yaşı gibi

değişkenler üzerinden ortalama gelirini tahmin etmek için Regresyon yöntemine başvurulmaktadır [18, 31]. Performans ölçme yöntemleri olarak IAE, ITAE, MSE ve ISE gibi yöntemler kullanılmaktadır. Regresyon Ağacı ve Linear Regresyon gibi popüler yöntemleri mevcuttur [17, 18]. IAE, ITAE, MSE ve ISE gibi performans ölçme yöntemlerinin amacı; birçok sonucu tek bir sonuçta birleştirmektir. Uygulamanın sonucu olan tahmin hatalarının farklı özelliklerini ortaya çıkarmaktadırlar [32].

Parnianifard vd., Oransal-İntegral-Türevsel (Proportional-Integral-Derivative-PID) kontrolörü ayarlama için kullanılan parametrelerin optimum değerlerini bulmak için yeni bir yöntem önermişlerdir. IAE, ITAE, Zaman Ağırlıklı Hata Karelerinin Toplamı (Integral Time Square Error-ITSE) ve ISE amaç fonksiyonları kullanılmıştır. ITAE ve ITSE'nin sonuçları ile IAE ve ISE'nin sonuçlarının aynı olduğu gözlemlenmiştir [33].

Ansari vd., PID kontrolörünün parametrelerini bulmak için Karınca Kolonisi Optimizasyonu algoritmasını kullanmışlardır. Yapılan çalışmada beş farklı amaç fonksiyonunun (IAE, ITAE, ITSE, MSE ve ISE) sonuçları analiz edilmiştir. ITAE'nin başarı oranının yüksek olduğu gözlemlenmiştir [34]. Amador-Angulo vd., bulanık kontrolörlerin üye fonksiyonlarındaki parametrelerin optimum değerlerini bulmak için hibrit bir yaklaşım önermektedirler. Uygulamada amaç fonksiyonu olarak ITAE, IAE, ISE, ITSE, Ortalama Hata Kare Kökü (Root Mean Square Error-RMSE) ve MSE hata fonksiyonları kullanılmıştır. MSE'nin performansının yüksek olduğu gözlemlenmiştir [35].

MLR, Birden fazla bağımsız değişken ve bağımlı bir değişken kullanarak bu değişkenleri en iyi şekilde ifade edebilen bir doğru uydurmaktır. Yani X giriş değerleri ile Y çıkış değerleri arasındaki ilişkiyi en uygun şekilde ifade edebilecek doğrusal bir fonksiyon elde etmektir [36-38]. MLR için genel fonksiyon Eş. 1'de gösterilmektedir.

$$Y=b+\theta X+c \quad (1)$$

Burada b sabit değeri, θ tahmin edilecek katsayıları ve c ise hata payını temsil etmektedir. X ve Y değerleri dışarıdan alınmaktadır.

Kakchapati vd., Tayland'da halen büyük bir sağlık sorunu olan tüberkülozun hastalık, ölüm ve sakatlık oranlarındaki etkisini modellemek ve tahmin etmek için MLR yöntemi kullanmışlardır. Çalışmanın sonunda elde edilen faydalı bulgular sayesinde çeşitli sağlık programları geliştirilmiştir [36].

Gustafsson vd., İsveç'te 24056 otopsi vakası değerlerine MLR uygulamışlardır. Vücut ağırlığı, vücut boyu, cinsiyeti ve yaş gibi parametreler ile bir model oluşturmak hedeflenmiştir. Hedeflenen model oluşturulmuş fakat başarı oranı düşük çıkmıştır. "Belirsiz parametreler ile MLR, düşük başarı oranı vermektedir" sonucuna varmaktadırlar [37].

Nasri vd., Uzun Vadeli Evrim (Long Term Evolution-LTE) geniş kablosuz ağlarında servis kalitesini etkileyen parametrelerin tahmini için MLR kullanmışlardır. Modelde ortalama kullanıcı verimi, ortalama gecikme, ağ trafiği ve ortalama kanal kalitesini içeren bir öğrenme veri seti kullanmaktadırlar. Kullanıcı veriminin ağ trafiği ve ortalama kanal kalitesi ile arttığı, gecikme ile düştüğü gözlemlenmektedir [38]. Çalışmalardaki önerilen yöntemlerin başarı oranlarının yüksek olması beklenmektedir. Bu yöntemlerin başarı oranları literatürdeki diğer yöntemlerin başarı oranlarıyla kıyaslanmalıdır. Kıyaslanmaların sağlıklı olabilmesi için de yöntemlerin aynı veri setleri üzerinde çalışmış olmaları gerekmektedir.

Patel vd., iris veri setini kullanarak dört farklı algoritmanın (Karar Ağaçları, Çok Katmanlı Algılayıcılar, Naive Bayes ve Çok Sınıflı Sınıflayıcılar) sonuçlarını (TP-Oran, Fp-Oran, Kesinlik, Duyarlılık ve ROC Eğrisi Parametreleri) kıyaslamışlardır. WEKA programı ile uygulama gerçekleştirilmiştir. Çok Katmanlı Algılayıcılar yönteminin tüm sonuç değerlerinde diğer yöntemlerden daha başarılı olduğu gözlemlenmiştir [39]. Shekhawat vd., Birlikte Kural Çıkarımı ve Sınıflandırma Kural Çıkarımı yöntemlerini birleştirerek Sinir Ağları Birlikte Kural Çıkarım Sistemi adında hibrit bir algoritma önermişlerdir. Bu algoritmayı kullanarak iris veri seti üzerinden başarı oranları analiz edilmiştir. 0,2 Öğrenme katsayısı, 0,7 momentum sabiti ve 2000 iterasyon sayısı ile başarı oranı %96,667 ile en yüksek seviyeye çıkmıştır [40]. Tiwari vd., iris veri seti üzerinde k-ortalamlar ve k-medoid kümeleme algoritmalarını kullanarak zaman ve mekân karmaşıklık değerlerini kıyaslanmışlardır. İki algoritmanın analizi sonucunda k-ortalamlar yönteminin zaman olarak k-medoid yöntemine göre az maliyetli olduğu görülmüştür. Buna rağmen k-ortalamlar yönteminin k-medoid yöntemine göre gürlüğe daha hassas olduğu gözlemlenmiştir [41].

Demir vd., iris veri setine FGA uygulayarak kümeleme işlemi yapmışlardır. Bu veri setine daha önce uygulanan algoritmalarda alınan başarı oranı %33 iken FGA'daki başarı oranı %79,33 olarak gözlemlenmiştir. Bu başarının sebebi başlangıç popülasyonunun rastgele yerine düzenli olarak üretilmesidir [42]. Bu çalışmada, optimizasyon algoritmalarındaki başlangıç popülasyonu konusuna odaklanılmaktadır. Stokastik ve deterministik yöntemlerle oluşturulan başlangıç popülasyonlarının sonuçları karşılaştırılmaktadır. Önerilen deterministik başlatma yönteminde uygulamanın sonuçlarına çok kısa sürede ulaşılmaktadır ve bu durum çok büyük bir zaman kazancı sağlamaktadır. Model olarak MLR kullanılmaktadır ve bu model bir başlangıç popülasyonu ile başlatılmaktadır. Başlangıç popülasyonu MLR modeli çalışmanın diğer yeniliği olarak kabul edilmektedir. MLR modelleri genelde tek bir değer ile başlatılmaktadır. Veri seti olarak da iris veri seti kullanılmaktadır. Dört öznitelige sahip bu veri setinin iki özniteliği ile çalışmaya devam edilmektedir. Bunun sebebiyse, "az öznitelikle de başarı sağlanabilir mi?" sorusuna cevap aramaktır. Kullanılan MLR modelinde hata oranının minimuma indirilmesi için altı tane Dik İniş tabanlı optimizasyon yöntemi (SGD, Momentum, Adagrad, RMSProp, Adadelta ve Adam) kullanılmaktadır. Amaç

fonksiyonu olarak da dört hata fonksiyonundan (IAE, ITAE, MSE, ISE) yararlanılmaktadır.

2. OPTİMİZASYON ALGORİTMALARI (OPTIMIZATION ALGORITHMS)

Makine Öğrenme yöntemlerinde her iterasyon sonucunda bir hata miktarı elde edilmektedir. Bu hata miktarını düşürmek için yaygın olarak kullanılan altı optimizasyon yöntemi mevcuttur. Bunlar; SGD, Momentum, Adagrad, RMSProp, Adadelta ve Adam yöntemleridir.

2.1. SGD

Dik İniş olarak da adlandırılan SGD yöntemi, güncelleme yaparken tüm türevler yerine rastgele seçilen bir kısım türev kullanılmaktadır [30, 43]. Eş. 2'de gösterildiği gibi mevcut türev ($\partial L / \partial w_t$) ile öğrenme katsayısını (a) çarparak mevcut ağırlığı (w_t) güncellemektedir [44-46].

$$w_{t+1} = w_t - a \frac{\partial L}{\partial w_t} \quad (2)$$

2.2. Momentum

Optimum nokta aranırken SGD yöntemi çok fazla salınım yapmaktadır. Bu salınımları azaltmak ve dolayısıyla hedefe gitme hızını arttırmak için Momentum yöntemi önerilmektedir [30, 43]. Eş. 3 ve Eş. 4'te gösterildiği gibi bu yöntemde mevcut türevler yerine momentumlu türev kullanılmaktadır [44-46].

$$w_{t+1} = w_t - a V_t \quad (3)$$

$$V_t = \beta V_{t-1} + (1-\beta) \frac{\partial L}{\partial w_t} \quad (4)$$

Burada V_t 'nin başlangıç değeri 0'dır. β , 0 ile 1 arasında olup genelde kullanılan değeri ise 0,9'dur ve geçmiş türevlerin işleme ne kadar katılacağını ayarlamak için kullanılmaktadır [44-46].

2.3. Adagrad

SGD ve Momentum yöntemlerinde sabit bir öğrenme katsayısı kullanılmaktadır. Bu problemi ortadan kaldırmak için Adagrad yöntemi önerilmektedir [30, 43]. Adagrad yönteminin amacı her adımda farklı bir öğrenme katsayısı ile işlem yapmaktır. Eş. 5 ve Eş. 6'da gösterildiği gibi Eş. 2'de kullanılan öğrenme katsayısını geçmiş türevlerin karelerinin kümülatif toplamının kareköküne bölmektedir [44-46].

$$w_{t+1} = w_t - \frac{a}{\sqrt{S_t + \epsilon}} \cdot \frac{\partial L}{\partial w_t} \quad (5)$$

$$S_t = S_{t-1} + \left[\frac{\partial L}{\partial w_t} \right]^2 \quad (6)$$

Burada S başlangıçta 0, ϵ ise işlemi sıfıra bölme hatasından kurtarmak için genelde çok küçük bir sayı (10^{-7}) olarak alınmaktadır [44-46].

2.4. RMSProp

Adagrad yönteminde olduğu gibi sabit öğrenme katsayısı problemine odaklanmaktadır [30, 43]. Aralarındaki farksa, Eş. 7 ve Eş. 8'de gösterildiği gibi Adagrad yöntemindeki türevlerin karelerini almak yerine momentumlu türevlerin karelerini almaktadır [44–46].

$$w_{t+1} = w_t - \frac{a}{\sqrt{S_t + \epsilon}} \cdot \frac{\partial L}{\partial w_t} \quad (7)$$

$$S_t = \beta S_{t-1} + (1-\beta) \left[\frac{\partial L}{\partial w_t} \right]^2 \quad (8)$$

Burada S başlangıçta 0, a=0,001, $\beta=0,9$ ve ϵ ise 10^{-6} olarak tercih edilmektedir [44–46].

2.5. Adadelta

Adadelta ve RMSProp yöntemleri farklı kişiler tarafından aynı yıl içerisinde önerilmektedir. Adagrad'ın iki farklı versiyonu olan bu yöntemler sabit öğrenme katsayısı problemini çözmeye odaklanmaktadır [30, 43]. Adagrad ve RMSProp yöntemlerinden farklı olarak Adadelta yönteminde öğrenme katsayısı seçme zorunluluğu yoktur. Eş. 9,-Eş. 12'de gösterildiği gibi öğrenme katsayısı yerine geçerli ağırlıklar ile güncellenen ağırlıklar arasındaki farkı ifade eden delta değerlerinin karelerinin momentumlu toplamları kullanılmaktadır [44–46].

$$w_{t+1} = w_t - \frac{\sqrt{D_{t-1}}}{\sqrt{S_t + \epsilon}} \cdot \frac{\partial L}{\partial w_t} \quad (9)$$

$$D_t = \beta D_{t-1} + (1-\beta) [\Delta w_t]^2 \quad (10)$$

$$S_t = \beta S_{t-1} + (1-\beta) \left[\frac{\partial L}{\partial w_t} \right]^2 \quad (11)$$

$$\Delta w_t = w_t - w_{t-1} \quad (12)$$

Burada S ve D başlangıçta 0, $\beta=0,95$ ve ϵ ise 10^{-6} olarak tercih edilmektedir [44–46].

2.6. Adam

Momentum ve RMSProp yöntemlerinin avantajlı yönlerinin birleştirilmesi ile elde edilmiştir [30, 43]. Eş. 13-Eş. 17'de gösterildiği gibi Momentum yöntemindeki V ile RMSProp yöntemindeki S'i kullanmaktadır [44–46].

$$w_{t+1} = w_t - \frac{a}{\sqrt{S_t + \epsilon}} \cdot \hat{V}_t \quad (13)$$

$$\hat{V}_t = \frac{V_t}{1-\beta_1} \quad (14)$$

$$\hat{S}_t = \frac{S_t}{1-\beta_2} \quad (15)$$

$$V_t = \beta_1 V_{t-1} + (1-\beta_1) \frac{\partial L}{\partial w_t} \quad (16)$$

$$S_t = \beta_2 S_{t-1} + (1-\beta_2) \left[\frac{\partial L}{\partial w_t} \right]^2 \quad (17)$$

Burada S ve V başlangıç olarak 0, a=0,001, $\beta_1=0,9$, $\beta_2=0,999$ ve ϵ ise 10^{-8} olarak tercih edilmektedir [44–46].

3. AMAÇ FONKSİYONLARI (OBJECT FUNCTIONS)

Makine öğrenme alanında Kayıp Fonksiyonu, Hata Fonksiyonu, Maliyet Fonksiyonu ve Amaç Fonksiyonu kavramlarıyla sıkça karşılaşılmaktadır. Yapılan çalışmalarda genellikle aynı oldukları ve birbirleri yerine kavramsal olarak kullanılmalarına rağmen aralarında küçükte olsa fark bulunmaktadır. Kayıp Fonksiyonu ve Hata Fonksiyonu aynı kavramlardır. Genellikle tek veri örneği üzerinde tanımlanan fonksiyonlardır. Hatanın karesini almak örnek olarak verilebilir [34]. Maliyet Fonksiyonu, tüm veri seti üzerindeki toplam hata olarak tanımlanmaktadır. Hatanın karelerinin ortalaması örnek olarak verilebilir. Amaç Fonksiyonu ise eğitim esnasında optimize edilecek herhangi bir fonksiyon için kullanılan genel bir terimdir. Diğer bir deyişle maksimize veya minimize etmek istediğimiz fonksiyon Amaç Fonksiyonu olarak adlandırılmaktadır. Eğer fonksiyonu minimize etmek istiyorsak bu fonksiyon aynı zamanda Maliyet Fonksiyonu olarak adlandırılmaktadır [47–49].

Özetle, Kayıp Fonksiyonu ve Hata Fonksiyonu Maliyet Fonksiyonu'nun bir parçası, Maliyet Fonksiyonu ise Amaç Fonksiyonu'nun bir türü olarak tanımlanabilir. Çalışmada kullanılan amaç fonksiyonları Eş. 18-Eş. 21'de gösterilmektedir.

$$IAE = \int_0^t |e(t)| dt \quad (18)$$

$$ITAE = \int_0^t t|e(t)| dt \quad (19)$$

$$MSE = \frac{1}{n} \int_0^t (e(t))^2 dt \quad (20)$$

$$ISE = \int_0^t (e(t))^2 dt \quad (21)$$

4. PROBLEMİN FORMÜLASYONU (PROBLEM FORMULATION)

Tablo 1'de stokastik ve deterministik popülasyon başlatma yöntemlerinin detayları sunulmaktadır. MLR modellerinde katsayılar için başlangıçta tek nokta seçilmektedir. Bu noktalar genelde sıfır, bir veya rastgele seçilen bir değer olmaktadır. Yapılan çalışmada katsayıların başlangıcı için bir popülasyon kullanılmaktadır. Bu da çalışmanın yeniliklerinden biri olarak tanımlanmaktadır.

MLR modeli uygulanan iris veri seti 1936'da R.A. Fisher tarafından oluşturulmuştur ve 1988'de Michael Marshall tarafından kullanıma sunulmuştur. Bu veri seti örüntü tanıma ile ilgili belki de en iyi bilinen veri setidir. Üç sınıftan (iris setosa, iris virginica ve iris versicolor) oluşmaktadır ve her sınıfın dört özneliği (sepal_length, sepal_width, petal_length, petal_width) bulunmaktadır. Her bir sınıftan 50'şer olmak üzere toplamda 150 örnek bulunmaktadır İris veri seti için Eş. 22 fonksiyonu temel alınmaktadır. İris veri

setinin sadece iki özneteliğini kullanarak başarı elde etmek amaçlanmaktadır. [50, 51].

$$Y = \theta_1 X_{\text{petal_width}} + \theta_2 X_{\text{petal_length}} \quad (22)$$

Y, $X_{\text{petal_width}}$ ve $X_{\text{petal_length}}$ değerleri iris veri setinden okunarak θ_1 ve θ_2 katsayıları tahmin edilmeye çalışılmaktadır. θ_1 ve θ_2 katsayıları genellikle başlangıç olarak 0 veya 1 alınmaktadır ya da iki değer arasında rastgele üretilmektedirler. İşlemler sonucunda tahmin edilen katsayılar ile X giriş değerleri fonksiyonda yerine konularak bir Y' (Y üssü) elde edilmektedir. Y gerçek değer ile Y' tahmin edilen değer arasındaki fark hata miktarını oluşturmaktadır.

5. DENEYSEL ÇALIŞMA (EXPERIMENTAL STUDY)

Yapılan uygulamanın tüm modülleri Şekil 1'de gösterilmektedir. İki farklı başlangıç popülasyonu, altı farklı optimizasyon algoritması ve dört farklı amaç fonksiyonu ile sonuçlar üretilmektedir. Yani 2x6x4 tane sonuç karşılaştırılmaktadır ve sonuçlar analiz edilmektedir. Çok değişkenli regresyon problemleri çözümünde birçok giriş ve buna karşılık gelen çıkış değerleri kullanılarak bir fonksiyonun bulunması istenmektedir. Bu sebeple regresyon işlemi sonucunda bulunan en uygun değerler Eş. 22'deki θ katsayılarının yerine konularak tüm giriş ve çıkış değerlerini temsil eden fonksiyon bulunmaktadır. Bu katsayılar yerine

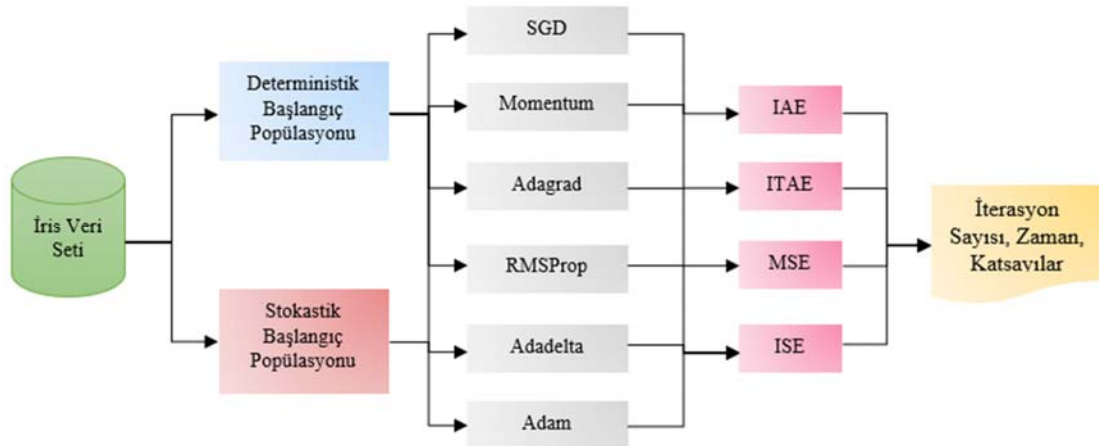
uygun değerler bulmak için bir başlangıç değeri atanarak (genelde 0 ya da 1 verilir) süreç başlatılmaktadır. Bu başlangıç değeri ile Dik İniş optimizasyon algoritmaları kullanılarak adım adım optimum θ katsayısı elde edilmektedir. Yani bir tane θ değeri ile başlanmaktadır. Bu çalışmada stokastik ve deterministik yöntemlerle oluşturulan birçok başlangıç katsayısıyla başlanmaktadır. Çünkü çözüme en kısa sürede gitmek hedeflenmektedir. Uygulamanın Tablo 2'de gösterilen sözde kodundaki "pre" ile başlayan fonksiyonların (pre_SGD, pre_Momentum, pre_Adagrad, pre_RMSProp, pre_Adadelta, pre_Adam) amacı başlangıç çözüm adayları arasındaki en uygun olan çözümü bulmaktır. Bütün çözüm adayları tek tek optimizasyon ve amaç fonksiyonlarından birer kez işlendikten sonra en düşük sonucu veren çözüm adayı ile yola devam edilmektedir. Elde edilen uygun çözüm adayı ile regresyon adımları hata toleransı 0,001 olana kadar devam ettirilmektedir. Regresyon işleminin sonucunda tüm veri setini temsil eden fonksiyonun katsayıları bulunmaktadır. Şekil 2'de uygulamanın akış diyagramı verilmektedir.

6. ARAŞTIRMA BULGULARI (RESEARCH FINDINGS)

Başlangıç popülasyon sayıları farklı başlatılan (n=10, n=50 ve n=100) uygulamalardan elde edilen çıktılar Tablo 3-Tablo 5'te listelenmektedir. Tablo 3'te başlangıç popülasyonu olarak 10 tane çözüm adayı ile başlatılan uygulamanın çıktıları karşılaştırılmaktadır. Burada farklı başlangıç

Tablo 1. Stokastik ve deterministik başlatma yöntemlerinin algoritmaları
(Algorithms of stochastic and deterministic initialization methods)

Stokastik başlatma yöntemi	Deterministik başlatma yöntemi
def StochasticInitPopulation(LB,UB,X): for i in range(X.shape[1]): interval =(UB[i]-LB[i]) for j in range(X.shape[0]-2): X[j+1,i]=LB[i]+random(0,1)* interval return X	def DeterministicInitPopulation (LB,UB,N,X): for i in range(X.shape[1]): interval =(UB[i]-LB[i])/(N-1) temp=0 for j in range(X.shape[0]-2): temp=temp+ interval X[j+1,i] = LB[i]+temp return X
LB: Alt sınır, UP: Üst sınır, N: Popülasyon boyutu, X: Popülasyon matrisi	



Şekil 1. Çalışmanın tüm modülleri (All modules of the study)

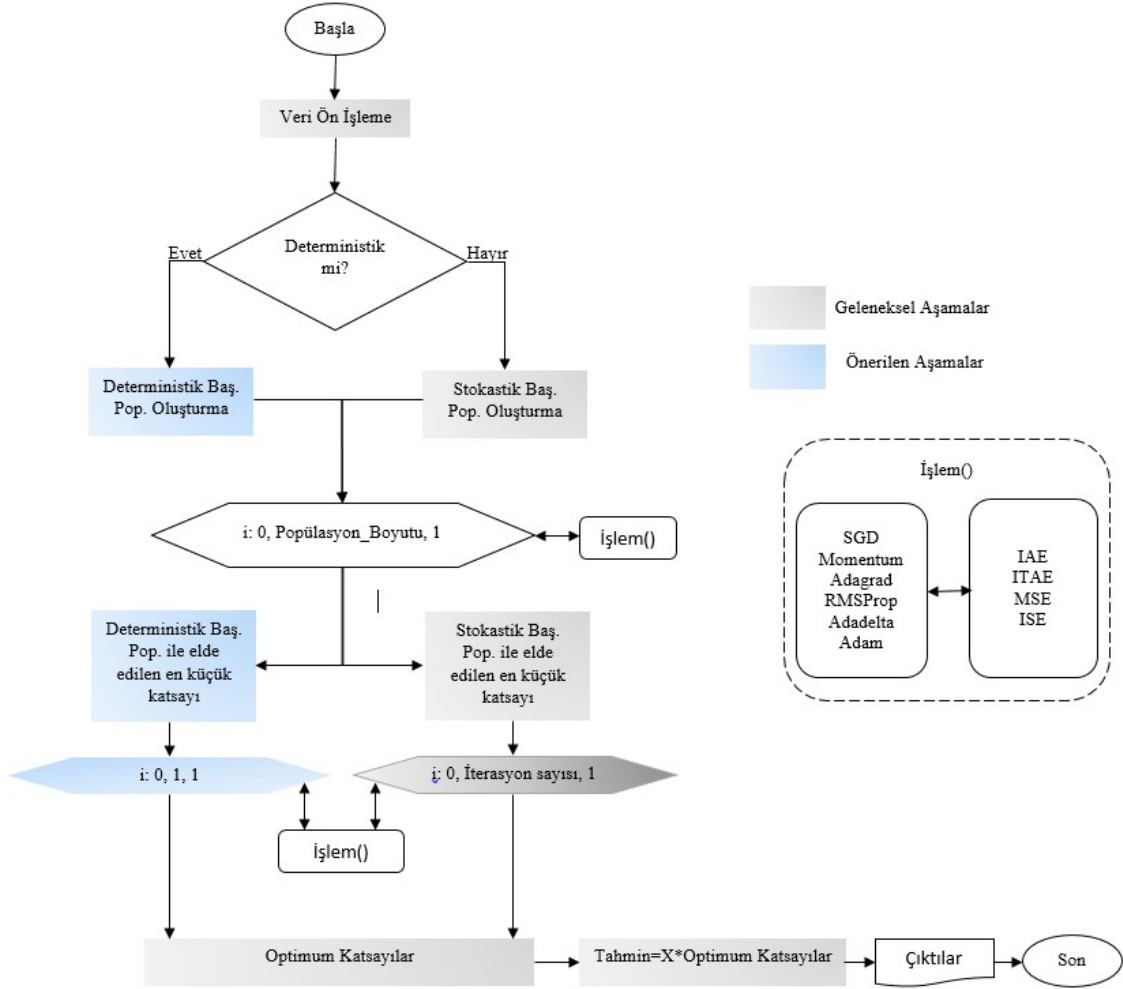
Tablo 2. Uygulamanın sözde kodu (Pseudo codes of the application)

Girdiler: N: Popülasyon Boyutu, LB: Alt Sınır, UB: Üst Sınır
 ϵ : hata, $X_{\text{petal_width}}$, $X_{\text{petal_length}}$, Y
Çıktılar: iteration number, θ_1 , θ_2 , Time

1. For i in range (amaç fonksiyon sayısı= 4 (IAE,ITAE,MSE,ISE)):
2. For j in range (çalışma sayısı):
// Deterministik başlangıç için çalışma sayısı 1 ve stokastik başlangıç için 10'dur.
3. LB ve UB arasında N-boyutlu stokastik veya deterministik bir başlangıç popülasyonu oluştur;
start_time;
4. For k in range (N):
5. $pre_SGD(i,k)$ //Minimum theta'nın (θ_k) bulunması
6. While $\epsilon < 0.001$
7. Optimum_ $\theta_{1,2}$, iteration_number, time=SGD(i, min_k)
8. For k in range (N):
9. $pre_Momentum(i,k)$
10. While $\epsilon < 0.001$
11. Optimum_ $\theta_{1,2}$, iteration_number, time=Momentum(i, min_k)
12. For k in range (N):
13. $pre_Adagrad(i,k)$
14. While $\epsilon < 0.001$
15. Optimum_ $\theta_{1,2}$, iteration_number, time=Adagrad(i, min_k)
16. For k in range (N):
17. $pre_RMSProp(i,k)$
18. While $\epsilon < 0.001$
19. Optimum_ $\theta_{1,2}$, iteration_number, time=RMSProp(i, min_k)
20. For k in range (N):
21. $pre_Adadelta(i,k)$
22. While $\epsilon < 0.001$
23. Optimum_ $\theta_{1,2}$, iteration_number, time=Adadelta(i, min_k)
24. For k in range (N):
25. $pre_Adam(i,k)$
26. While $\epsilon < 0.001$
27. Optimum_ $\theta_{1,2}$, iteration_number, time=Adam(i, min_k)
28. End For
29. End For

popülasyon tipleri, optimizasyon algoritmaları ve amaç fonksiyonları kullanarak yapılan uygulamaların çıktıları (θ_1 , θ_2 , iterasyon sayıları (is)) listelenmektedir. Ayrıca tablonun en alt satırında her bir amaç fonksiyonu için tüm optimizasyon yöntemlerinde geçen işlem zamanları verilmektedir. Deterministik başlatma yöntemi ile başlatılan uygulamalarda bütün amaç fonksiyonlarının işlem zamanları stokastik başlatma yöntemine göre çok düşük olduğu görülmektedir. Optimizasyon yöntemlerinden Adadelta bütün amaç fonksiyonlarında en düşük iterasyon sayısı ile sonuçlara ulaşmaktadır. ISE hariç diğer amaç fonksiyonlarında Adagrad optimizasyon yöntemi de Adadelta ile birbirine yakın iterasyon sayıları ile sonuçlanmaktadır. Her iki başlatma yönteminde de Adam optimizasyon algoritması en yüksek iterasyon sayılarına sahiptir. Amaç fonksiyonlarından MSE bütün optimizasyon algoritmalarında en düşük iterasyon sayıları ile sonuçlara

ulaşmaktadır. Tablo 4'te 50 tane çözüm adayına sahip başlangıç popülasyonu ile başlatılan uygulamanın çıktıları listelenmektedir. Tablo 3'te olduğu gibi deterministik yöntemle başlatılan uygulamaların işlem zamanları oldukça düşüktür. Adadelta en başarılı optimizasyon yöntemi ve MSE en başarılı amaç fonksiyonu olarak gözlemlenmektedir. Burada Adagrad'ın başarısı biraz düşmektedir. Tablo 5'te ise 100 tane çözüm adayı ile başlatılan uygulamanın sonuçları listelenmektedir. Uygulamaların çıktılarının birbirine yakın olduğu gözlemlenmektedir. Burada dikkat edilmesi gereken nokta, başlangıç popülasyon sayısı 100 olduğunda deterministik yöntemde işlem zamanlarının düşmesidir. Çünkü popülasyon sayısı yükseldikçe deterministik başlatma yönteminde optimum çözüme daha yakın bir çözüm adayı ile uygulama başlatılmaktadır. Bu durum sayesinde düşük işlem zamanları ile sonuçlara ulaşmaktadır. Sonuçlar



Şekil 2. Akış diyagramı (Flowchart)

incelendiğinde deterministik başlatma yönteminin stokastik başlatma yönteminden zaman açısından oldukça avantajlı olduğu görülmektedir. Çünkü deterministik başlatma yönteminde uygulamayı bir kez çalıştırmak yeterli olmakta ve sadece bir kez çalışan uygulama ile başarılı sonuçlar elde edilebilmektedir. Stokastik başlatma yönteminde ise başarılı sonuçların elde edilebilmesi için uygulamanın birden fazla çalıştırılarak ortalama sonuçlarının elde edilmesi gerekmektedir. Her iki yöntemde elde edilen çıktılar neredeyse aynı olmasına rağmen toplam zaman açısından büyük fark vardır. Deterministik olarak başlatılan popülasyonda zamansal olarak çok yüksek bir kazanım elde edilmektedir. Örneğin deterministik yöntemle başlatılan popülasyonda ($n=50$ için) IAE amaç fonksiyonu için tüm optimizasyon algoritmalarına uygulandığında geçen zaman 0,22 dk olmaktadır. Buna karşılık aynı amaç fonksiyonu ve optimizasyon algoritmaları ile stokastik başlangıç popülasyonunda 10 çalışma için geçen zaman miktarı 3,46 dk'dır. Şekil 3'te deterministik yöntemle başlatılan ($n=50$ için) ve IAE amaç fonksiyonunu kullanan uygulamanın bütün optimizasyon algoritmalarındaki hata ve iterasyon sayıları grafiksel düzlemde gösterilmektedir. Şekil 4'te ise

aynı amaç fonksiyonu için uygulama stokastik olarak başlatılmaktadır. Burada tüm optimizasyon algoritmaları için uygulamalar 10 defa çalıştırılarak sonuçlar grafiksel olarak sunulmaktadır. Şekil 3 ve Şekil 4'teki aynı optimizasyon yöntemleri karşılaştırıldığında, birbirine yakın iterasyon sayısı ile hataların minimuma ulaştıkları görülmektedir. Her iki başlatma yönteminde de Adagrad ve Adadelata optimizasyon yöntemleri çok küçük iterasyon sayıları ile sonuca ulaşmaktadırlar. Şekil 3 ve 4'te olduğu gibi Şekil 5 ve Şekil 6'da ITAE, Şekil 7 ve Şekil 8'de MSE ve Şekil 9 ve Şekil 10'da ISE amaç fonksiyonu için yapılan uygulamaların hata ve iterasyon sayıları grafiksel olarak gösterilmektedir.

Deterministik ve stokastik başlatma yöntemlerinin farklı amaç fonksiyonlarına göre tüm optimizasyon algoritmalarının sonuçları Şekil 3-Şekil 10'da gösterilmektedir. Deterministik başlatma yöntemi 1 kez, stokastik başlatma yöntemi ise 10 kez çalıştırılmaktadır ve çıktılar aynı grafiksel düzlemde basılmaktadır. Şekillerde de görüldüğü gibi birbirine yakın iterasyon sayıları ile hatalar minimuma indirgenmektedir.

Tablo 3. N=10 için sonuç tablosu (Output table for N=10)

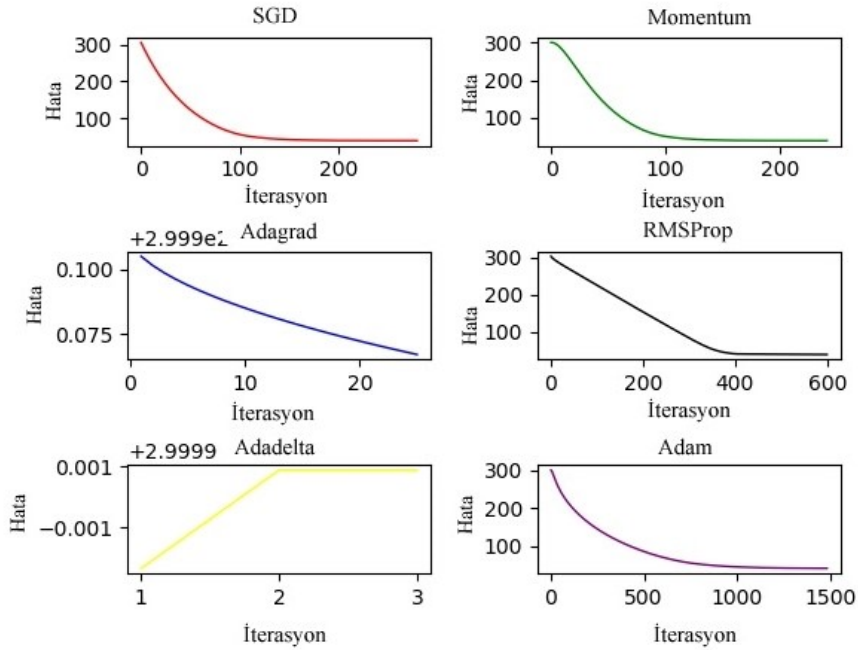
		Deterministik Başlangıç Popülasyonu				Stokastik Başlangıç Popülasyonu (10 çalışmanın ortalaması)			
		IAE	ITAE	MSE	ISE	IAE	ITAE	MSE	ISE
SGD	θ_1	0,462206021	0,462482394	0,429720276	0,461478247	0,462381282	0,462355595	0,430072430	0,461489498
	θ_2	0,151751240	0,151799593	0,141875107	0,151602718	0,151233912	0,152108781	0,141371530	0,151580749
	is	276,0	283,0	131,0	261,0	281,5	287,2	132,2	265,4
Momentum	θ_1	0,462647785	0,462847320	0,434343990	0,461820740	0,462747813	0,462691429	0,434378339	0,461742565
	θ_2	0,151992994	0,152027862	0,143372704	0,151819252	0,151706800	0,152538066	0,142816361	0,151982390
	is	236,0	241,0	120,0	220,0	241,5	246,2	120,7	223,9
Adagrad	θ_1	0,000012170	0,000008087	0,000008087	0,000261290	0,000014126	0,000006023	0,000006773	0,000236369
	θ_2	0,000012202	0,000008096	0,000008096	0,000261602	0,000013198	0,000006106	0,000005350	0,000235142
	is	5,0	2,0	2,0	543,0	9,8	4	2,0	515,8
RMSProp	θ_1	0,441199282	0,386927602	0,364136244	0,422366390	0,444137610	0,387054184	0,364467578	0,428026563
	θ_2	0,219725604	0,377304580	0,363057733	0,274060105	0,209968498	0,376959237	0,362444858	0,257580192
	is	604,0	434,0	390,0	546,0	702,9	433,8	388,6	568,2
Adadelta	θ_1	0,000000157	0,000000157	0,000000155	0,000000157	0,000000157	0,000000157	0,000000155	0,000000157
	θ_2	0,000000155	0,000000155	0,000000153	0,000000155	0,000000155	0,000000155	0,000000153	0,000000155
	is	3,0	3,0	2,0	3,0	3,0	3,0	2,0	3,0
Adam	θ_1	0,407606851	0,407150277	0,320765004	0,407283434	0,408615015	0,407842102	0,321339112	0,407522804
	θ_2	0,298951156	0,298726792	0,238159428	0,298793060	0,296119508	0,297083759	0,237582688	0,297613593
	is	1642,0	1621,0	637,0	1627,0	1745,1	1573,4	638,3	1761,8
zaman (dk)		0,22	0,21	0,10	0,25	3,33	3,26	1,34	4,03

Tablo 4. N=50 için sonuç tablosu (Output table for N=50)

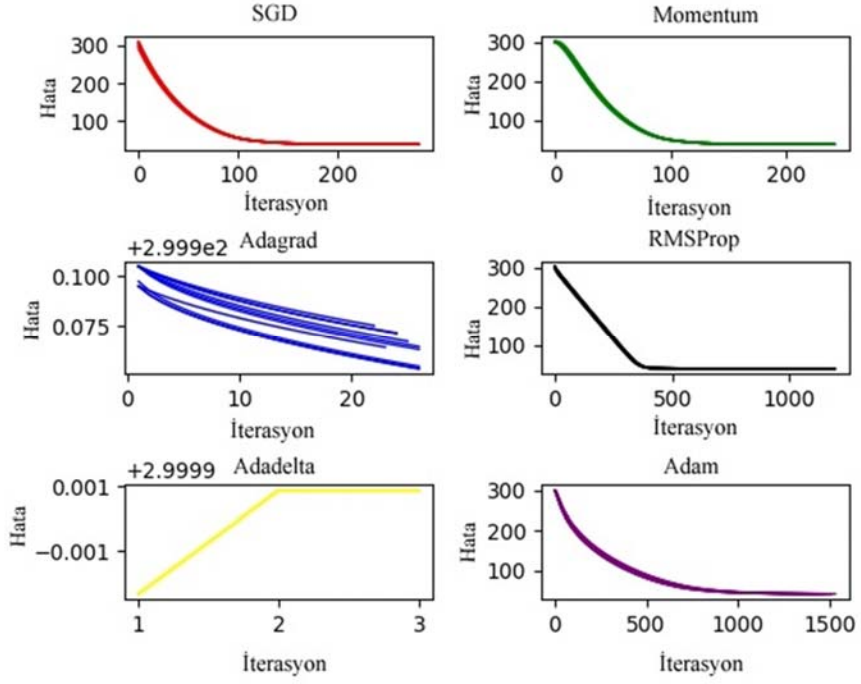
		Deterministik Başlangıç Popülasyonu				Stokastik Başlangıç Popülasyonu (10 çalışmanın ortalaması)			
		IAE	ITAE	MSE	ISE	IAE	ITAE	MSE	ISE
SGD	θ_1	0,462316617	0,462596794	0,430027271	0,461634564	0,462166809	0,462539705	0,430026262	0,461472806
	θ_2	0,151323076	0,151372885	0,141501841	0,151183136	0,151845014	0,151578681	0,141615886	0,151640395
	is	281,0	288,0	137,0	267,0	280,6	286,7	135,9	265,9
Momentum	θ_1	0,462722101	0,462920630	0,434564948	0,461899259	0,462555382	0,462885458	0,434285390	0,461718007
	θ_2	0,151800387	0,151834988	0,143226981	0,151627825	0,152278555	0,151975090	0,143199394	0,152039099
	is	242,0	247,0	126,0	226,0	240,9	245,7	124,5	224,4
Adagrad	θ_1	0,000044225	0,000020110	-0,000001963	0,000293352	0,000051989	0,000027048	0,000004008	0,000304061
	θ_2	0,000043469	0,000019497	-0,000002175	0,000292395	0,000050263	0,000027439	0,000003989	0,000303940
	is	25,0	10,0	2,0	563,0	24,7	9,3	2,0	563,4
RMSProp	θ_1	0,441073806	0,387111122	0,364506347	0,422133727	0,442185742	0,386935784	0,364083276	0,431173302
	θ_2	0,220061749	0,376901858	0,363053012	0,274420379	0,217581782	0,377345055	0,362816055	0,248402339
	is	599,0	430,0	386,0	541,0	680,5	425,9	384,1	564,3
Adadelta	θ_1	0,000000157	0,000000157	0,000000155	0,000000157	0,000000157	0,000000157	0,000000155	0,000000157
	θ_2	0,000000155	0,000000155	0,000000153	0,000000155	0,000000155	0,000000155	0,000000153	0,000000155
	is	3,0	3,0	2,0	3,0	3,0	3,0	2,0	3,0
Adam	θ_1	0,407894868	0,407805304	0,325889342	0,407714671	0,408127869	0,407484993	0,292825095	0,407367498
	θ_2	0,298271254	0,298227971	0,241028162	0,298183834	0,297746476	0,299124553	0,217270928	0,299401247
	is	1483,0	1479,0	573,0	1475,0	1464,3	1455,3	517,6	1442,6
zaman (dk)		0,22	0,24	0,11	0,26	3,46	4,14	1,49	4,29

Tablo 5. N=100 için sonuç tablosu (Output table for N=100)

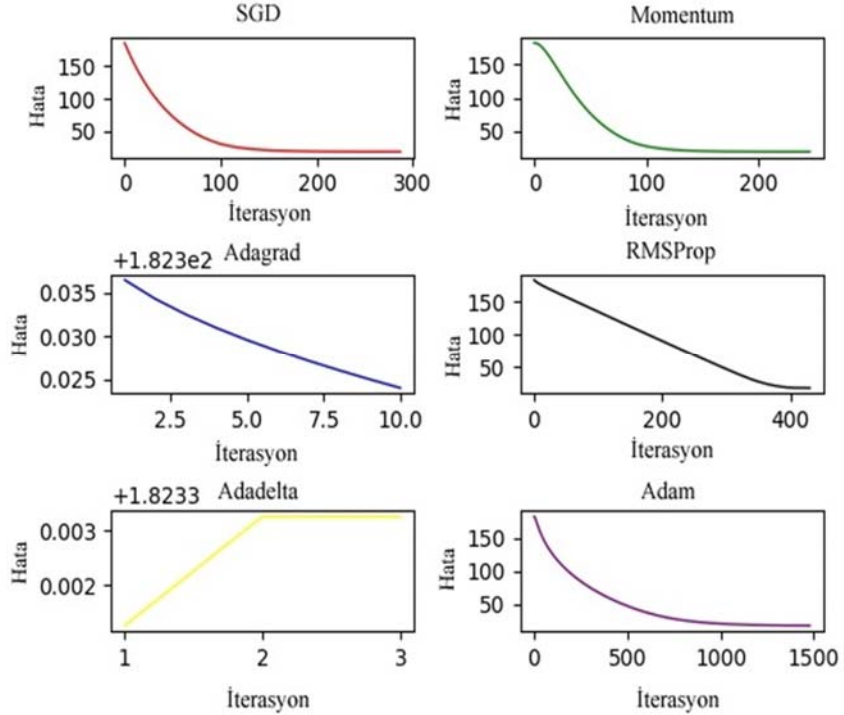
		Deterministik Başlangıç Popülasyonu				Stokastik Başlangıç Popülasyonu (10 çalışmanın ortalaması)			
		IAE	ITAE	MSE	ISE	IAE	ITAE	MSE	ISE
SGD	θ_1	0,462326480	0,462603596	0,429744339	0,461596702	0,462224557	0,462592186	0,429791924	0,461612233
	θ_2	0,151373683	0,151422456	0,141461935	0,151224122	0,151652649	0,151430483	0,141648386	0,151250612
	is	281,0	288,0	136,0	266,0	281,0	286,6	135,4	266,0
Momentum	θ_1	0,462697771	0,462935547	0,434253248	0,461866762	0,462605653	0,462936735	0,390669853	0,461853187
	θ_2	0,151818175	0,151859536	0,143150097	0,151642941	0,152108424	0,151818531	0,129004060	0,151644927
	is	241,0	247,0	125,0	225,0	241,3	245,5	111,9	224,4
Adagrad	θ_1	0,000049849	0,000025842	-0,000000466	0,000298824	0,000047398	0,000029484	0,000006522	0,000303425
	θ_2	0,000049376	0,000025423	-0,000000657	0,000298273	0,000047633	0,000029657	0,000006374	0,000301381
	is	26,0	11,0	2,0	564,0	25,2	9,8	2,0	563,2
RMSProp	θ_1	0,418115359	0,387027311	0,364324209	0,421922377	0,444911827	0,386974293	0,364169619	0,424571950
	θ_2	0,294751343	0,377077030	0,363016574	0,275596067	0,209119297	0,377255082	0,362866161	0,267435560
	is	517,0	428,0	384,0	538,0	738,8	425,2	381,3	543,7
Adadelta	θ_1	0,000000157	0,000000157	0,000000155	0,000000157	0,000000157	0,000000157	0,000000155	0,000000157
	θ_2	0,000000155	0,000000155	0,000000153	0,000000155	0,000000155	0,000000155	0,000000153	0,000000155
	is	3,0	3,0	2,0	3,0	3,0	3,0	2,0	3,0
Adam	θ_1	0,407798214	0,407798214	0,326705178	0,407661844	0,407562067	0,407619641	0,326733561	0,407748405
	θ_2	0,298528493	0,298528493	0,241927180	0,298462402	0,299075225	0,298884701	0,242118885	0,298240120
	is	1454,0	1454,0	561,0	1448,0	1469,9	1444,3	558,7	1446,9
zaman (dk)	0,19	0,2	0,1	0,24	3,32	3,44	1,46	3,56	



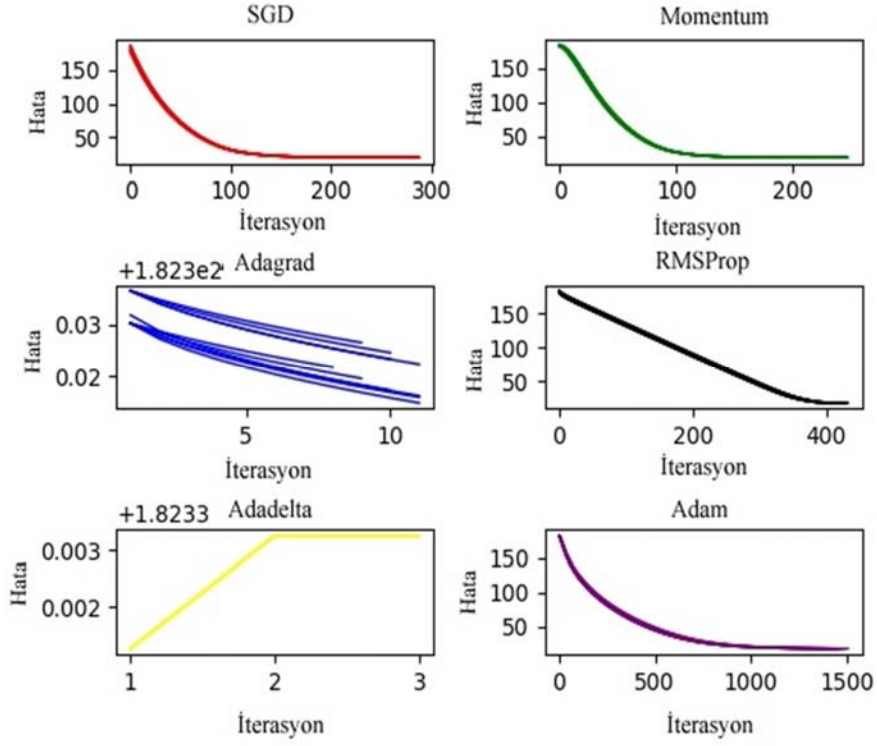
Şekil 3. Deterministik başlangıç popülasyonu ve IAE amaç fonksiyonu ile çıktılar (Outputs with deterministic initial population and IAE objective function)



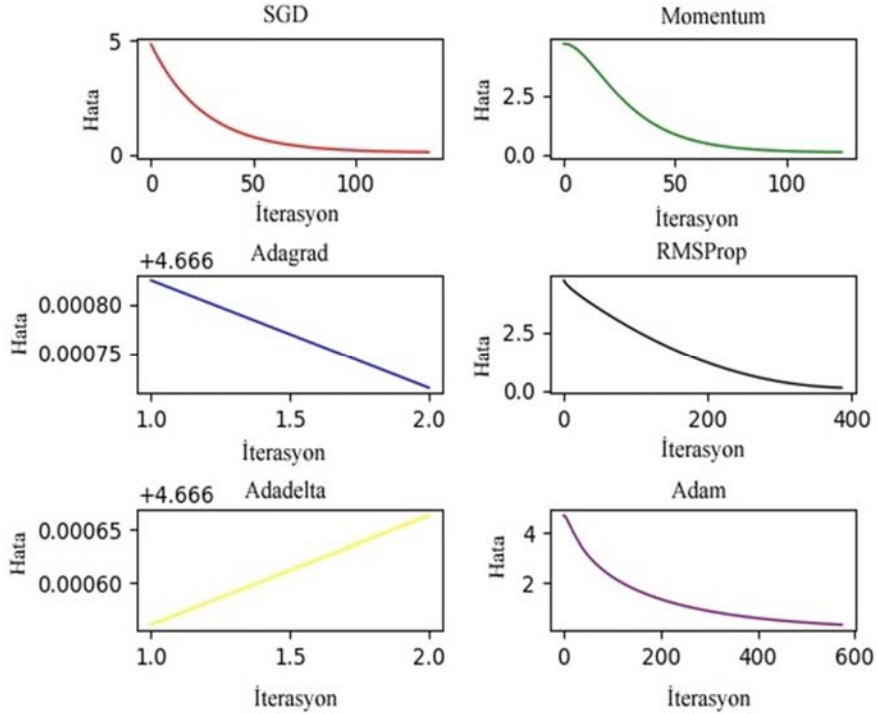
Şekil 4. Stokastik başlangıç popülasyonu ve IAE amaç fonksiyonu ile çıktılar
(Outputs with stochastic initial population and IAE objective function)



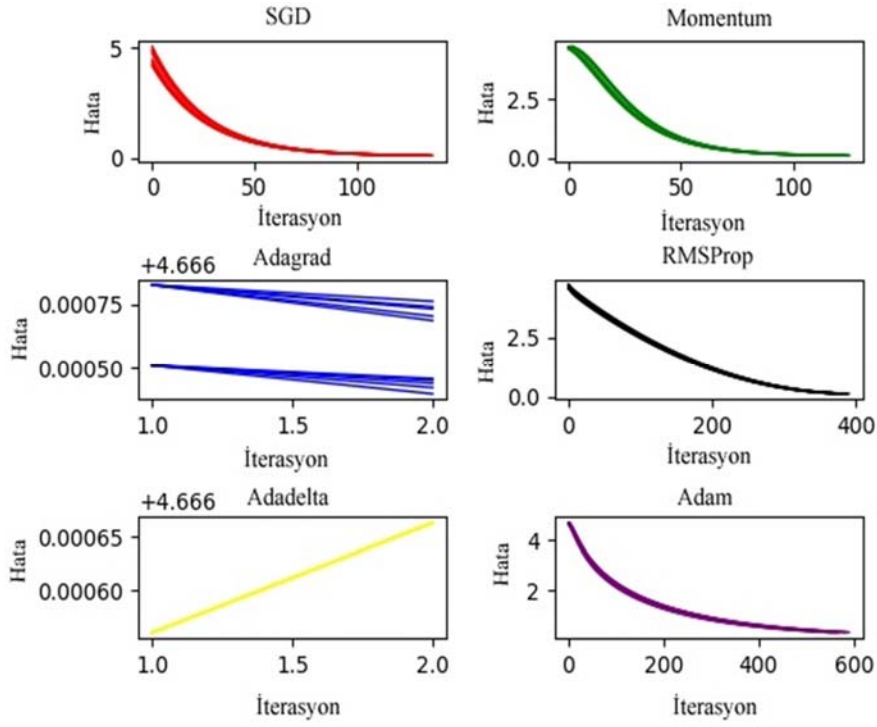
Şekil 5. Deterministik başlangıç popülasyonu ve ITAE amaç fonksiyonu ile çıktılar
(Outputs with deterministic initial population and ITAE objective function)



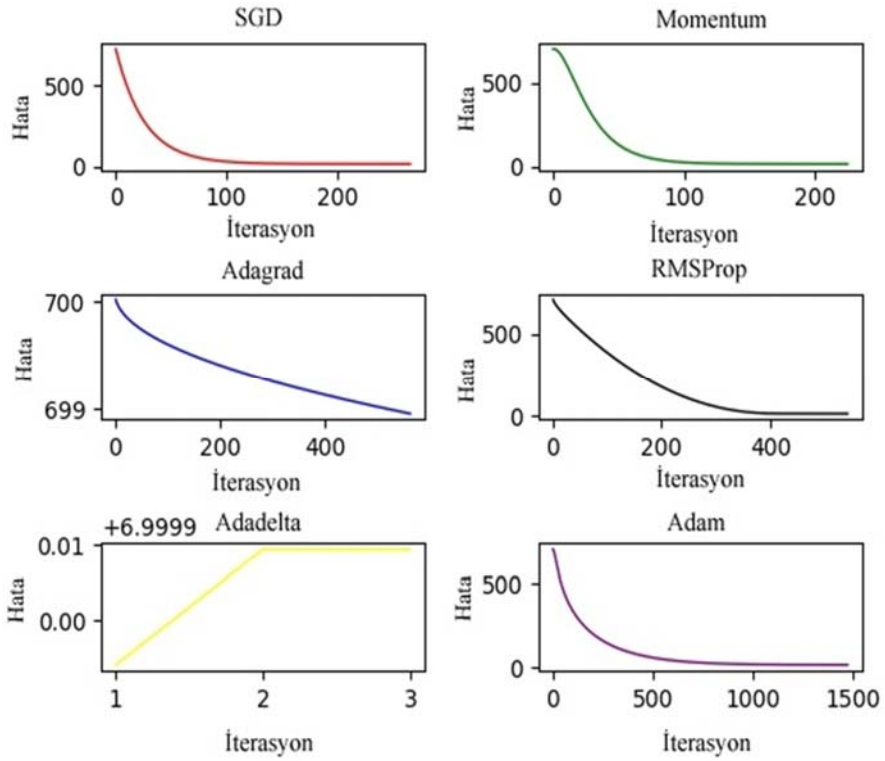
Şekil 6. Stokastik başlangıç popülasyonu ve ITAE amaç fonksiyonu ile çıktılar (Outputs with stochastic initial population and ITAE objective function)



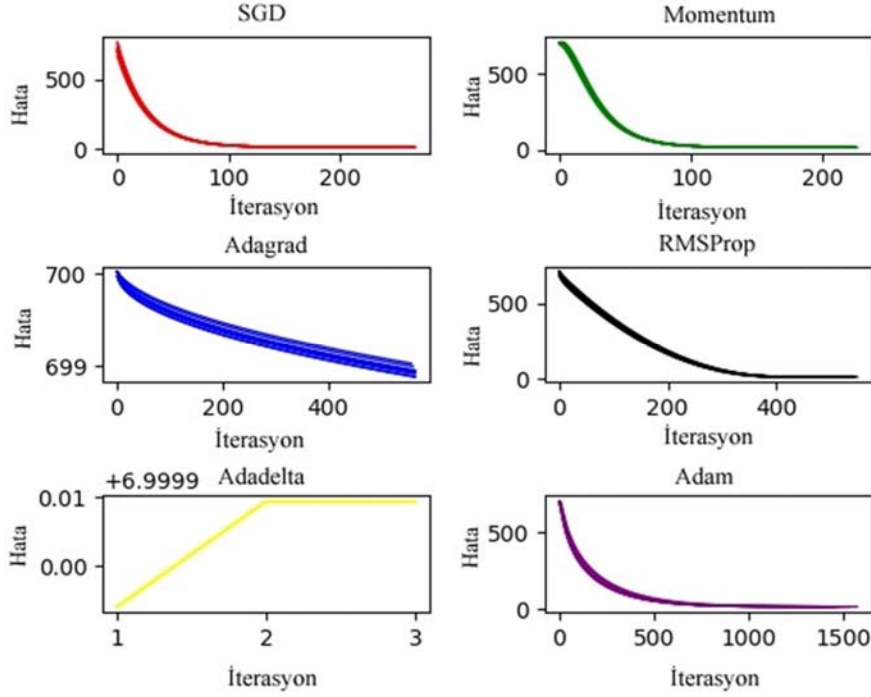
Şekil 7. Deterministik başlangıç popülasyonu ve MSE amaç fonksiyonu ile çıktılar (Outputs with deterministic initial population and MSE objective function)



Şekil 8. Stokastik başlangıç popülasyonu ve MSE amaç fonksiyonu ile çıktılar
(Outputs with stochastic initial population and MSE objective function)



Şekil 9. Deterministik başlangıç popülasyonu ve ISE amaç fonksiyonu ile çıktılar
(Outputs with deterministic initial population and ISE objective function)



Şekil 10. Stokastik başlangıç popülasyonu ve ISE amaç fonksiyonu ile çıktılar
(Outputs with stochastic initial population and ISE objective function)

7. SONUÇLAR (CONCLUSIONS)

Günümüzde Makine Öğrenme yöntemlerinde başlangıç popülasyonu oluşturulurken halen en çok tercih edilen yöntem geleneksel rastgele başlatma yöntemidir. Bu yöntemle başlatılan bir optimizasyon algoritmasından elde edilen ilk çıktılar literatüre direkt verilemez. Çünkü rastgele başlatma yöntemiyle oluşturulan başlangıç popülasyonları ve bu başlangıç popülasyonlarıyla başlatılan uygulamaların sonunda elde edilen çıktılar sürekli farklı olmaktadır. Bu sebeple uygulamaların birden fazla çalıştırılması ve elde edilen çıktıların ortalamasının literatüre sunulması daha kabul edilebilirdir. Her çalışmanın da bir zamansal maliyeti mevcuttur. Bu zamansal maliyeti düşürmek için literatüre deterministik bir başlatma yöntemi önerilmektedir. Önerilen yöntemle oluşturulan başlangıç popülasyonuna sahip uygulamanın 1 kez çalıştırılması yeterli olmaktadır. Çünkü uygulama sabit değerler ile başladığı için her çalışmanın çıktıları eşit olmaktadır. Bunun için deterministik yöntemle başlatılan uygulamanın 10, 50 veya 100 kez çalıştırılması gerekmemektedir. Deterministik yöntemle elde edilen katsayılar ve iterasyon sayıları ile stokastik yöntemle elde edilen ortalama katsayılar ve iterasyon sayılarının neredeyse eşit olduğu gözlemlenmektedir. Gözlemlenen bu eşitliğe rağmen deterministik yöntemin yaklaşık %90 civarında bir zamansal iyileştirme sağladığı görülmektedir. Bu sebeplerle önerilen yöntemin avantajlı olduğu sonucuna varılmaktadır. Her iki başlatma yönteminde de optimizasyon yöntemlerinden Adadelta, amaç fonksiyonlarından da MSE'nin çok kısa sürede sonuçlara ulaştıkları görülmektedir.

Çalışmanın literatüre diğer bir katkısı da MLR modellerindeki başlatma şekline getirilen yeniliktir. MLR modellerinde parametrelerin optimum değerlere ulaşması için başlangıçta ya sabit ya da rastgele bir değere başlatılması gelenekseldir. Seçilen değer optimum değere uzak olması da zamansal bir maliyet getirmektedir. Bu maliyeti düşürmek için MLR modellerine popülasyon şeklinde bir başlangıç yöntemi önerilmektedir. Bir değer yerine birden fazla değer seçeneği ile MLR modelleri daha avantajlı hale gelmektedir. Başlangıç popülasyonundaki her değer ile uygulama bir kez çalıştırılmakta ve çıktılar sıralanmaktadır. Sıralamadaki en küçük değer uygulamanın başlangıç değeri olmaktadır. Bu başlangıç değeri ile MLR modeli daha kısa sürede uygun çözüme ulaşmaktadır.

KAYNAKLAR (REFERENCES)

1. Eröz E., Tanyildizi E., Güncel Metasezgisel Optimizasyon Algoritmalarının Performans Karşılaştırılması, 2018 International Conference on Artificial Intelligence and Data Processing, Malatya-Türkiye, 1-16, 28-30 Eylül, 2018.
2. Zaki S., El-Nasser A., Zaied H., Meta-heuristics Algorithms: A survey, International Journal of Engineering Trends and Technology, 67 (5), 67-74, 2019.
3. Ponce-Ortega J.M., Hernández-Pérez L.G., Optimization of Process Flowsheets through Metaheuristic Techniques, Springer, UK, 2019.
4. Dokeroglu T., Sevinc E., Kucukyilmaz T., Cosar A., A survey on new generation metaheuristic algorithms, Computer & Industrial Engineering, 137, 106040-106069, 2019.

5. Nocedal J., Stephen W., Numerical Optimization, Springer Science & Business Media, USA, 2006.
6. Chong E. K. P., Zak S. H., An introduction to optimization, Wiley, USA, 2013.
7. Türkay M., Optimizasyon Modelleri ve Çözüm Metodları, New Frontiers in Total Quality and Strategic Management, 309-328, 2006.
8. Stork J., Eiben A. E., Bartz-Beielstein T., A new taxonomy of continuous global optimization algorithms, Natural Computing, *arXiv*, 1-24, 2020.
9. Collet P., Rennard J.-P., Stochastic Optimization Algorithms, Intelligent Information Technologies, 1121-1137, 2008.
10. Fausto F., Reyna-Orta A., Cuevas E., Andrade Á. G., Perez-Cisneros M., From ants to whales: metaheuristics for all tastes, Artificial Intelligence Review, 53, 753–810, 2020.
11. Villarrubia G., De Paz J. F., Chamoso P., De la Prieta F., Artificial neural networks used in optimization problems, Neurocomputing, 272, 1-16, 2018.
12. Garip Z., Çimen M. E., Boz A. F., Comparative performance analysis on parameter extraction of solar cell models using meta-heuristic algorithms, Journal of the Faculty of Engineering and Architecture of Gazi University, 36 (2), 1133-1144, 2021.
13. Kubat M., An Introduction to Machine Learning, Springer International Publishing AG, USA, 2017.
14. Shalev-Shwartz S., Ben-David S., Understanding machine learning: From theory to algorithms, Cambridge University Press, UK, 2014.
15. Deisenroth M. P., Faisal A. A., Ong C. S., Mathematics for Machine Learning, Cambridge University Press, UK, 2020.
16. Alpaydin E., Introduction to Machine Learning, Mit Press, USA, 2020.
17. Akanksha R., ML Classification vs Regression. Geeksforgeeks. <https://www.geeksforgeeks.org/ml-classification-vs-regression/>. Yayın tarihi Aralık 2, 2019. Erişim tarihi Eylül 15, 2020.
18. Garbade D. M. J., Regression Versus Classification Machine Learning: What's the Difference?. Medium. <https://medium.com/quick-code/regression-versus-classification-machine-learning-whats-the-difference-345c56dd15f7>. Yayın tarihi Ağustos 11, 2018. Erişim tarihi Ekim 8, 2020.
19. Huang X., Guan Z., Yang L., An effective hybrid algorithm for multi-objective flexible job-shop scheduling problem, Advances Mechanical Engineering, 10 (9), 1-14, 2018.
20. Ali M., Pant M., Abraham A., Unconventional initialization methods for differential evolution, Applied Mathematic and Computation, 2019 (9), 4474-4494, 2013.
21. Kazimipour B., Li X., Qin A. K., A review of population initialization techniques for evolutionary algorithms, 2014 IEEE Congress on Evolutionary Computation, Beijing-China, 2585–2592, 2014.
22. Kazemzadeh Azad S., Seeding the initial population with feasible solutions in metaheuristic optimization of steel trusses, Engineering Optimization, 50 (1), 89-105, 2018.
23. Zhang G., Gao L., Shi Y., An effective genetic algorithm for the flexible job-shop scheduling problem, Expert Systems with Applications, 38 (4), 3563-3573, 2011.
24. Tometzki T., Engell S., Systematic initialization techniques for hybrid evolutionary algorithms for solving two-stage stochastic mixed-integer programs, IEEE Transactions on Evolutionary Computation, 15 (2), 196-214, 2010.
25. Dong N., Wu C. H., Ip W. H., Chen Z. Q., Chan C. Y., Yung K. L., An opposition-based chaotic GA/PSO hybrid algorithm and its application in circle detection, Computer & Mathematics with Applications, 64 (6), 1886-1902, 2012.
26. Zhang Y., Saxe A. M., Advani M. S., Lee A. A., Energy-entropy competition and the effectiveness of stochastic gradient descent in machine learning, Molecular Physics, 116 (21-22), 3214-3223, 2018.
27. Vani S., Rao T. V. M., An experimental approach towards the performance assessment of various optimizers on convolutional neural network, International Conference on Trends in Electronics and Informatics, Tirunelveli-India, 331-336, 2019.
28. Rajakumaran G., Venkataraman N., Mukkamala R. R., Denial of Service Attack Prediction Using Gradient Descent Algorithm, SN Computer Science, 1 (1), 1-8, 2020.
29. Dogo E. M., Afolabi O. J., Nwulu N. I., Twala B., Aigbavboa C. O., A Comparative Analysis of Gradient Descent-Based Optimization Algorithms on Convolutional Neural Networks, International Conference on Computational Techniques, Electronics and Mechanical Systems, Belgaum-India, 92-99, 2018.
30. Yazan E., Talu M. F., Comparison of the stochastic gradient descent based optimization techniques, International Artificial Intelligence and Data Processing Symposium, Malatya-Türkiye, 1-5, 2017.
31. Dönmez İ., Aslan Z., Document Sentiment Classification Using Hybrid Wavelet Methodologies, Journal of the Faculty of Engineering and Architecture of Gazi University, 36 (2), 701-714, 2021.
32. Pal S., Gauri S. K., Assessing effectiveness of the various performance metrics for multi-response optimization using multiple regression, Computer Industrial Engineering, 59(4), 976-985, 2010.
33. Parnianifard A., Azfanizam A. S., Ariffin M. K. A., Ismail M. I. S., Design and Analysis of Computer Experiments Using Polynomial Regression and Latin Hypercube Sampling in Optimal Design of PID Controller, Journal of Applied Research on Industrial Engineering, 5 (2), 156-168, 2018.
34. Ansari A. Q., Katiyar S., Application of ant colony algorithm for calculation and analysis of performance indices for adaptive control system, Innovative Applications of Computational Intelligence on Power, Energy and Controls with Their Impact on Humanity, Ghaziabad-India, 466-471, 2014.

35. Amador-Angulo L., Mendoza O., Castro J. R., Rodríguez-Díaz A., Melin P., Castillo O., Fuzzy sets in dynamic adaptation of parameters of a bee colony optimization for controlling the trajectory of an autonomous mobile robot, *Sensors (Switzerland)*, 16 (9), 1458, 2016.
36. Kakchapati S., Choonpradub C., Forecasting Tuberculosis (Mortality) in Thailand Using Multivariate Linear Regression, *Journal of Health Research*, 26 (1), 51-54, 2012.
37. Gustafsson T., Eriksson A., Wingren C. J., Multivariate linear regression modelling of lung weight in 24,056 Swedish medico-legal autopsy cases, *Journal of Forensic and Legal Medicine*, 46, 20-22, 2017.
38. Nasri M., Hamdi M., LTE QoS Parameters Prediction Using Multivariate Linear Regression Algorithm, 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops, Paris-France, 145-150, 2019.
39. Patel K., Vala J., Pandya J., Comparison of various classification algorithms on iris datasets using WEKA, *International Journal of Advance Engineering Research Development*, 1 (1), 2014.
40. Shekhawat M., Dhande S. S., Building an Iris Plant Data Classifier Using Neural Network Associative Classification, *International Journal of Advancement in Technology*, 2 (4), 491-506, 2011.
41. Tiwari M., Singh R., Comparative Investigation of K-Means and K-Medoid Algorithm on Iris Data, *International Journal of Engineering Research and Development*, 4 (8), 69-72, 2012.
42. Demir M., Karcı A., Veri kümelemede fidan gelişim algoritmasının kullanılması, *Elektrik, Elektronik, Bilgisayar, Biyomedikal Mühendisliği Ulusal Kongresi ve Fuarı, Eskişehir-Türkiye*, 14-18, 2007.
43. Karim B.R.. 10 Gradient Descent Optimisation Algorithms + Cheat Sheet. Kdnuggets. <https://www.kdnuggets.com/2019/06/gradient-descent-algorithms-cheat-sheet.html>. Yayın tarihi Haziran, 2019. Erişim tarihi Aralık 22, 2020.
44. Zeiler M. D., Adadelta: An Adaptive Learning Rate Method, arXiv preprint arXiv:1212.5701, 2012.
45. Ruder S., An overview of gradient descent optimization algorithms, arXiv preprint arXiv:1609.04747, 1-14, 2016.
46. Lydia A. A., Francis F. S., Adagrad: An Optimizer for Stochastic Gradient Descent, *International Journal of Information And Computing Science*, 6 (5), 599-568, 2019.
47. Krishnan K., Karpagam G., Comparison of PID Controller Tuning Techniques for a FOPDT System, *International Journal of Current Engineering and Technology*, 4 (4), 2667-2670, 2014.
48. Kishnani M., Pareek S., Gupta R., Optimal tuning of DC motor via simulated annealing, *International Conference on Advances in Engineering and Technology Research*, Unnao-India, 1-5, 2014.
49. Allafi N. I., Zayed A. S., Daw M. S., Ahmed W. M., Investigating the effect of GA based PID controller performance indices with application to stirred-Tank heater (jacket model), 17th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering, Sousse-Tunisia, 26-33 2016.
50. Fisher R.A., The use of multiple measurements in taxonomic problems, *Annals of Eugenics*, 7 (2), 179-188, 1936.
51. Ravi J.. Machine learning-Iris classification. Medium. <https://medium.com/@jebaseelanravi96/machine-learning-iris-classification-33aa18a4a983>. Yayın tarihi Ağustos 1, 2018. Erişim tarihi Ekim 11, 2020.

