

# Genetik Algoritmalar (GA) İle Nüfus Artış Tahmini

## Estimation of Population By Genetic Algorithm (GA)

Müslüm ÖZTÜRK  
Kilis 7 Aralık Üniversitesi  
[mozturk@kilis.edu.tr](mailto:mozturk@kilis.edu.tr)

Turan PAKSOY  
Selçuk Üniversitesi  
[tpaksoy@yahoo.com](mailto:tpaksoy@yahoo.com)

Melek ÖZTÜRK  
Kilis 7 Aralık Üniversitesi  
[melekozturk@kilis.edu.tr](mailto:melekozturk@kilis.edu.tr)

### Öz

*Bu çalışmada, nüfus artış tahminini yapmak üzere yapay zekanın bir alt disiplini olan genetik algoritmalarından yararlanılarak .NET ortamında C# programlama dili kullanılarak görsel ara yüze sahip bir yazılım geliştirilmiştir. Bu bağlamda TÜİK'ten yıllara göre yapılan sayım bilgileri baz alınmış ve geliştirilen yazılım sayesinde ileriki yıllar için nüfus artış tahmini yapılabilmektedir. Çalışma kapsamında, hız ve performans açısından ikili kodlu genetik algoritma yerine Gerçek Kodlu Genetik Algoritma; seçme yöntemi olarak da Turnuva Seçim Yöntemi kullanılmıştır. Çünkü Genetik Algoritmalar (GA) İkili kodlu algoritmalar; parametrelerin "1" ve "0" larla ifade edilmesi, kromozomların boyutlarını oldukça artırdığından sınırlı hassasiyete sahip olmaktadır. Oysa bunun yerine gerçek rakamlarla kodlama yapabilen, gerçek kodlu GA'yı kullanmak daha avantajlıdır. Gerçek kodlu GA, hem daha hassas hem de PC belleğinde daha az yer kaplamaktadır. Ayrıca mevcut yazılımların kullanım karmaşıklığının ve zorluğunun önüne geçmek için geliştirilen yazılımda, başlangıçta kullanıcı tarafından popülasyon sayısı, iterasyon sayısı, çaprazlama oranı, mutasyon oranı ve TÜİK'ten alınmış geçmiş yıllara ait nüfus sayım verileri girişi kullanıcı tarafından girişi yapıldıktan sonra elde edilen veriler doğrultusunda istenen yıla ait nüfus artış tahmini yapılmaktadır. Bu çalışma kapsamında da TÜİK'ten alınan veriler kullanılarak öncelikle Türkiye ve Konya ili için 2016 yılı nüfus artış tahminleri yapılmış ve bu tahmini artış oranı gerçek verilerle karşılaştırılmıştır. Daha sonra ise Türkiye geneli ve Konya ili için 2020 yılına ait nüfus artış tahmini oranına yer verilmiştir. Ayrıca geliştirilen yazılım elde edilen veriler doğrultusunda kıtalar, ülkeler, şehirler, beldeler ve hatta köyler için nüfus artış tahmini yapabilecek esneklik ve yapıda tasarlanmıştır.*

**Anahtar Kelimeler:** Yapay Zeka, Genetik Algoritma, Gerçek Kodlu Genetik Algoritma, Nüfus Artış Tahmini, C# .NET, Görsel Arayüz.

### Abstract

*In this study, it is aimed to predict population growth by using software with graphical user interface C# programming language which was developed in the .NET environment by making use of genetic algorithms which is a sub-discipline of artificial intelligence. In this context, based on the census conducted by Turkish Statistical Institute was taken as basis and the developed software was able to estimate the population growth for the following years. In the scope of the study, in terms of speed and performance Real Coded Genetic Algorithms was used instead of Binary Genetic Algorithms; Tournament Selection Method was used as the selection method. Because Genetic Algorithms (GA) is Binary Genetic Algorithms; the expression of the parameters with "1" and "0" increases size of the chromosomes considerably and that results limited sensitivity. However, it is more advantageous to use genuine coded GA, which can code with real numbers instead. The real code GA is both more accurate and takes up less space in the PC memory. In addition, in the developed software in order to avoid the complexity and difficulties of use of existing software, population growth forecast is being made initially after user enters the number of populations, the number of iterations, the crossover rate and the population counts of previous years information obtained from Turkish Statistical Institute. Within the context of this study, data collected from Turkish Statistical Institute for 2016 estimates of population growth were made for Turkey and Konya provinces and estimated increase rate was compared with the actual data. The estimated population increase rate for 2020 for Turkey and Konya province is given thereafter. In*

Gönderme ve kabul tarihi: 10.02.2018-05.05.2018

*addition to that, the developed software is designed for flexibility and structure that can predict population growth for continents, countries, cities, borders and even villages with available data.*

**Keywords:** Artificial Intelligence, Genetic Algorithm, Binary Genetic Algorithms, Real Coded Genetic Algorithms, Population Growth Prediction, C#. NET, graphical user interface.

## 1. Giriş

Genetik algoritmalar, doğal seleksiyon ve doğal genetik mekanizmasına dayanan araştırma algoritmalarıdır [1]. Genetik ve doğal seçim ilkesini büyük ölçüde taklit eden genetik algoritmalar, optimizasyon süreçlerinde ve hesaplama yükünün çok fazla olduğu durumlarda başarıyla uygulama potansiyeline sahiptir [2].

Basit bir genetik algoritma üç temel genetik işlemi içerir: seçim, çaprazlama ve mutasyon. Seçimde, nüfustaki çözümler ebeveyn olarak seçilir; Geçişte ebeveynler çocuk doğurmak için melezleştirilir ve mutasyonda ise yavru mutasyon kurallarına göre değiştirilebilir. Genetik algoritmalarda bir algoritmanın yinelemelerine ise nesiller denir [3].

GGA'lar, genetik operatörler yardımıyla evrim mekanizmasını uygulayan araştırma algoritmalarıdır. Olası çözümleri ifade eden kromozomlar çözüm uzayı sınırları içerisinde yer alan onluk düzendeki gerçek sayısal değerlerle ifade edilir. Bu özellik GGA'nın ikili kodlu GA'dan temel farkını da ortaya koymaktadır [4]. Birçok gerçek optimizasyon problemi için genlerin gerçek değerleriyle kullanılması problemlerin çözümünde daha yararlıdır. Çözümlerin ikili düzende tanımlanması, kullanılan bit sayısı ile ilişkili olarak çözüm kesinliğini etkiler. Eğer kromozomların değerleri çok büyük ve ondalıklı yapıya sahipse, bu kromozomları ikili kod ile temsil edilmesi çok karmaşık ve zor olacaktır. Ayrıca, çözümlerin gerçek değerlerle tanımlanmış olması hesaplama etkinliğini de oldukça artırır. Bu nedenle birçok mühendislik probleminin çözümünde GGA'ların kullanılması kullanıcıya büyük kolaylık ve yararlar sağlayabilmektedir [5-6].

Temel olarak, GA, biyolojik dünyadaki en uygun doğal seçilimi ve hayatta kalmayı taklit eden bir tür popülasyona dayalı stokastik araştırma yaklaşımıdır

[7]. Gri kodlama [8], tamsayı genler [9] gibi birçok mevcut kodlama şemasının yanı sıra, gerçek kodlanmış genetik algoritmaların [10,11], en sezgisel GA'lar içerisinde yer almaktadır. Sayısal çözümün gerçek sayı [12,13] ile doğrudan temsil edilebilmesi ve kodlama ve kod çözme prosedürlerine gerek duyulmamasından dolayı gerçek kodlu genetik algoritmalar; özellikle yüksek boyutlu veya yüksek hassasiyetli problemler için GA'lar içerisinde önemli bir yere sahiptirler. Bu konuda gerçek kodlu genetik algoritmaların çözüm verimliliğini ve kararlılığını kanıtlamak için yapılan teorik çalışmalara [14,15,16] ek olarak, bazı dikkate değer çalışmalar [17], birçok gerçek dünya optimizasyon uygulamasında, özellikle yüksek boyutlu veya yüksek hassasiyetli problemler için, birçok karşıt GA'yı geride bırakmaktadır.

Gerçek kodlu algoritma, olası çözümlerden (kromozomlardan) oluşan rastgele başlangıç popülasyonunun oluşturulmasıyla başlar. Kromozomlar belli sınırlar içerisinde belirlenirler ve daha önceden belirlenen bir amaç fonksiyonundan elde edilen uygunluk değerlerine göre değerlendirilirler. Seçim yöntemi ile en iyi uygunluk değerine sahip kromozomların (ebeveynlerin) seçilmesi sağlanır. Seçim operatörü ile belirlenen iki ebeveyn çaprazlama operatörü uygulanır. İki ebeveynin genetik bilgisinin değiştirilmesiyle çaprazlama operatörü uygulanmış olur ve yeni bir çocuk birey elde edilir. Fakat GGA'da çaprazlama operatörünün uygulanması bu kadar basit değildir. Gerçek değerlerle tanımlanmış kromozomlar arasındaki çaprazlama işlemi çeşitli şekillerde uygulanabilmektedir. Tek noktalı ve çok noktalı çaprazlama işlemlerinin yanında kromozom yapısında oluşturulan değişikliğin belirli bazı kural ve eşitliklere bağlanması da mümkündür. Algoritmanın bir diğer aşaması olan mutasyon işlemi de yine gerçek kodlu yapıya uygun formda gerçekleştirilir. Elde edilen yeni olası çözümlerin kaliteleri istenen değerlere ulaşana ya da istenilen sonlandırma kriteri sağlanana dek bu değerlendirme ve yeni popülasyon oluşturma işlemi devam eder [4].

Bu çalışmada, gerçek kodlu genetik algoritma kullanılarak Türkiye geneli ve Konya ili için 2020 yıllarına ait nüfus artış tahmininin yapılması amaçlanmıştır. Ayrıca Geçmiş yıllara ait Türkiye İstatistik Kurumu (TÜİK) verileri baz alınarak Türkiye ve Konya ili için 2016 yılı tahmini nüfus

sayısı hesaplanmış ve hesaplanan bu veriler 2016 yılı TÜİK verileri ile karşılaştırılmıştır. Böylece tasarlanan yazılımın gelecek yıllar için yapacağı tahmini nüfus hesaplamalarının da tahmini yanılma payı da test edilmiş olacaktır. Bu bağlamda TÜİK'ten yıllara göre yapılan sayım bilgileri baz alınarak geliştirilen yazılım sayesinde ileriki yıllar için nüfus artış tahmininde bulunabilecektir. Yapılan yazılım esnek olup iller bazındaki nüfus sayım bilgileri baz alındığında ilgili il için de nüfus artış tahmininde bulunmayı gerçekleştirebilecek yapıdadır. Bu bağlamda gerçek kodlu genetik algoritması kullanılarak nüfus artış tahmini yapılırken aşağıdaki bilgiler kullanılmıştır.

Nüfus artış hızı  $p$  olmak üzere;

$$p = 100 \left( \left( \frac{N_y}{N_e} \right)^{\frac{1}{t_y - t_e}} - 1 \right) \quad (1)$$

Burada;

$N_y$ : yeni (son) sayım ( $t_y$  yılının sayımı)

$N_e$ : uygun bir eski sayım ( $t_e$  yılının sayımı) Şeklinde hesaplanır. Burada  $p$  üzerinde aşağıda yer alan kısıtlar tanımlanmıştır:

$$1 \leq p \leq 3 \text{ olmalıdır.}$$

$$p < 1 \text{ ise } p=1$$

$p > 3$  ise  $p=3$  alınarak hesap yapılır. Ara değerler ise olduğu gibi alınarak kullanılır.

Gelecekteki nüfus ( $t_n$  yılı nüfusu)  $t_p$  tahminin yapıldığı yıl,  $N_y$  son sayım ve  $p$  nüfus artış hızı esas alınarak,

$$N_n = N_y \left( 1 + \frac{p}{100} \right)^{(35 + t_p - t_y)} \quad (2)$$

şeklinde hesaplanır. Burada,  $t_p \geq t_y$  dir.

## 2. Alan Araştırması

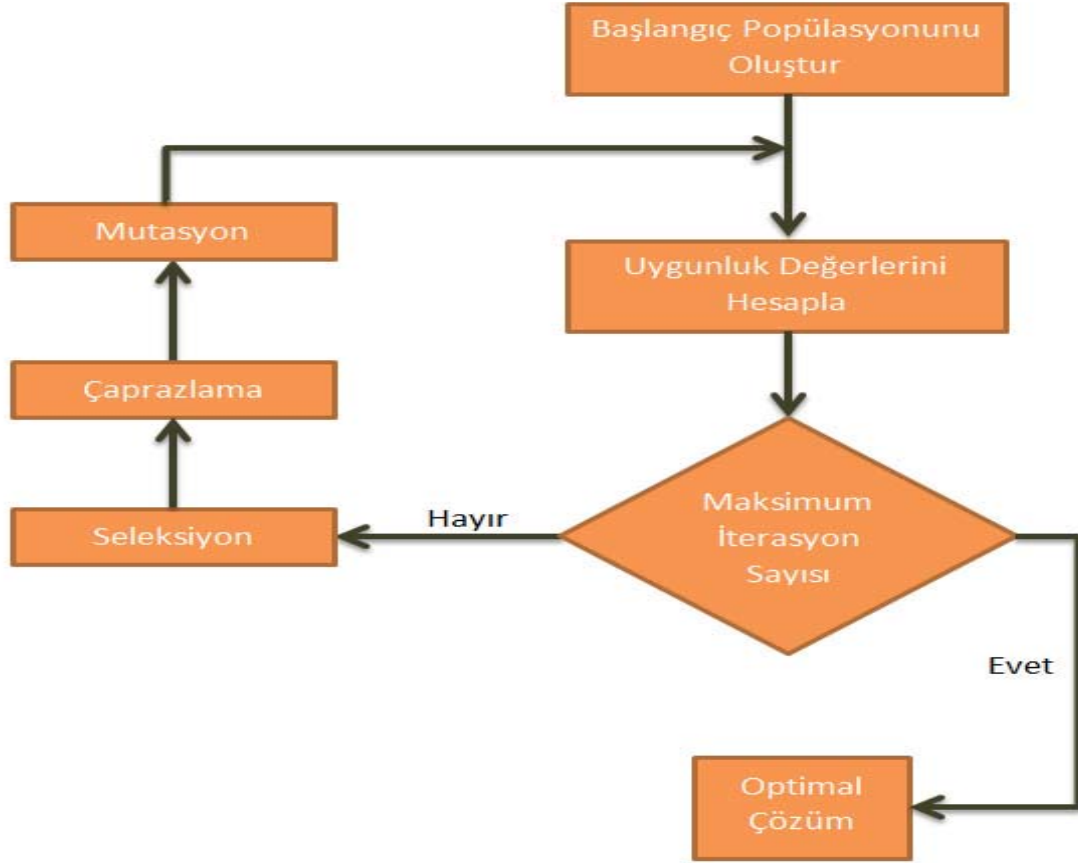
### 2.1. Genetik algoritmalar

Doğada gözlemlenen evrimsel sürece benzer bir şekilde çalışan arama ve eniyileme yöntemidir. Karmaşık çok boyutlu arama uzayında en iyinin hayatta kalması ilkesine göre bütünsel en iyi çözümü arar.

Genetik algoritmaların temel ilkeleri ilk kez Michigan Üniversitesi'nde John Holland tarafından ortaya atılmıştır [19]. Daha sonra Holland, 1975 yılında yaptığı çalışmaları "Adaptation in Natural and Artificial Systems" adlı kitabında bir araya getirmiştir. Bu, doğal seçim sürecinden esinlenen bir metaheuristikdir (sezgi üst). Genetik algoritmalar özellikle, çok popülasyonlu bir rastlantısal algoritma kullanarak; nüfus arasındaki rekabet ve evrim optimizasyonuna elverişlidir [20]. İlk olarak Holland evrim yasalarını genetik algoritmalar içinde eniyileme problemleri için kullanmıştır.

Genetik algoritmalar problemlere tek bir çözüm üretmek yerine farklı çözümlerden oluşan bir çözüm kümesi üretir. Böylelikle, arama uzayında aynı anda birçok nokta değerlendirilmekte ve sonuçta bütünsel çözüme ulaşma olasılığı yükselmektedir. Çözüm kümesindeki çözümler birbirinden tamamen bağımsızdır. Her biri çok boyutlu uzay üzerinde bir vektördür [18]. Genetik algoritmalar problemlerin çözümü için evrimsel süreci bilgisayar ortamında taklit ederler. Diğer eniyileme yöntemlerinde olduğu gibi çözüm için tek bir yapının geliştirilmesi yerine, böyle yapılardan meydana gelen bir küme oluştururlar. Problem için olası pek çok çözümü temsil eden bu küme genetik algoritma terminolojisinde nüfus adını alır. Nüfuslar vektör, kromozom veya birey adı verilen sayı dizilerinden oluşur. Birey içindeki her bir elemana gen adı verilir. Nüfustaki bireyler evrimsel süreç içinde genetik algoritma işlemcileri tarafından belirlenirler.

Bilgisayar dilinde, bu doğal süreci gerçeklemek için diziler ve alt diziler kullanılır. Diziler popülasyonu, bireyleri ve bireylerin genlerini barındırırlar. Burada bireylerden kasıt, çözüm uzayındaki yani çözümü içerisinde barındıran küme içerisindeki tüm elemanlardır. Her bir eleman, bütün elemanları kapsayabilecek uzunluktaki bitler dizisi ile temsil edilir. Bireyler popülasyonu oluşturur ve her bireyin bir uygunluk değeri vardır. Genetik algoritmanın genel akış şeması şekil 1'de verilmiştir.



**Şekil 1.** Genetik Algoritmanın Genel Akış Şeması

Genetik algoritmanın adımlarını ve başlıca tanımlarını özetleyecek olursak:

#### a) Başlangıç Popülasyonu:

GA, çözüm adımlarına belirlenen gösterim şekline uygun kodlanmış bireylerden oluşan bir başlangıç popülasyonu oluşturarak başlarlar.

Kromozom, başlangıçta verilen popülasyon miktarına göre rasgele üretilir [21].

Başlangıç popülasyonundaki her bir kromozom, problemin olası bir çözümünü temsil eder.

#### b) Kodlama:

Kodlama genetik algoritmanın çok önemli bir kısmını oluşturmaktadır. Bir problem çözümüne başlarken sorulması gereken ilk soru olmasının yanı sıra kurulan genetik modelin hızlı ve güvenilir çalışması için bu kodlama doğru yapılmalıdır.

#### c) Seçim:

Amaç; seçilen uygunluk fonksiyonuna ve seçim yöntemine göre elimizdeki popülasyondan yeni bir neslin bireylerini seçmektir. Ebeveynler uygunluk değerlerine göre eşleşmek üzere seçilirler. Uygunluğu yüksek olan bireyin, yeni nesle aktarılma ihtimali daha yüksektir.

### 2.2. Seçim Yöntemleri

Genetik algoritmalarda kullanılan başlıca seçim yöntemleri; rulet çemberi, rank seçimi (doğrusal sıralama seçimi, kesme seçimi), sabit durum seçimi, elitist seçim ve turnuva seçimi olarak sıralanabilir.

### 2.3. Çaprazlama

Amaç, ata kromozomun yerlerini değiştirerek çocuk kromozomlar üretmek ve böylelikle zaten uygunluk değeri yüksek olan ata kromozomlardan daha yüksek uygunluklu çocuk kromozomlar üretmektir. Aşağıda

yer alan Çizelge 1.'de örnek kromozom yapıları gösterilmiştir.

**Çizelge 1.** Örnek bir kromozom yapısı

<b>Kromozom1</b>	<b>11011</b>	<b>00100110110</b>
<b>Kromozom1</b>	<b>11011</b>	<b>11000011110</b>
<b>Yavru1</b>	<b>11011</b>	<b>11000011110</b>
<b>Yavru2</b>	<b>11011</b>	<b>00100110110</b>

## 2.4. Mutasyon

Yeniden ve sürekli yeni nesil üretimi sonucunda, belli bir süre sonra nesildeki kromozomlar birbirlerini tekrar edebilir. Böylece farklı kromozom üretimi durur veya azalır. İşte bu nedenle nesildeki kromozom çeşitliliğini artırmak için kromozomlardan bazıları mutasyona tabi tutulur. Aşağıda yer alan Çizelge 2.'de mutasyona uğramış kromozom örneği gösterilmiştir.

**Çizelge 2.** Mutasyona uğramış kromozom örneği

<b>Asıl Yavru1</b>	<b>1101111000011110</b>
<b>Asıl Yavru2</b>	<b>1101100100110110</b>
<b>Mutasyonlu Yavru1</b>	<b>1100111000011110</b>
<b>Mutasyonlu Yavru2</b>	<b>1101101100110100</b>

## 2.5. Çaprazlama Olasılığı (Oranı)

Çaprazlamanın ne kadar sıklıkla yapılacağını belirtir. Bu problemde çaprazlama olasılığı (oranı) 0.7 alınmıştır. Fakat kullanıcı isterse yazılımda bu değeri değiştirebilir.

## 2.6. Mutasyon Olasılığı (Oranı)

Kromozom parçalarının ne kadar sıklıkla mutasyon geçireceğini belirtir. Bu problemde mutasyon olasılığı (oranı) 0.01 alınmıştır. Fakat kullanıcı isterse yazılımda bu değeri değiştirebilir.

## 2.7. Popülasyon Büyüklüğü (Sayısı)

Toplumdaki birey sayısını belirtir. Bu problemde popülasyon büyüklüğü (oranı) başlangıç değeri olarak 100 değeri atanmıştır. Fakat kullanıcı isterse yazılımda bu değeri değiştirebilir.

## 2.8. Uygunluk Fonksiyonu

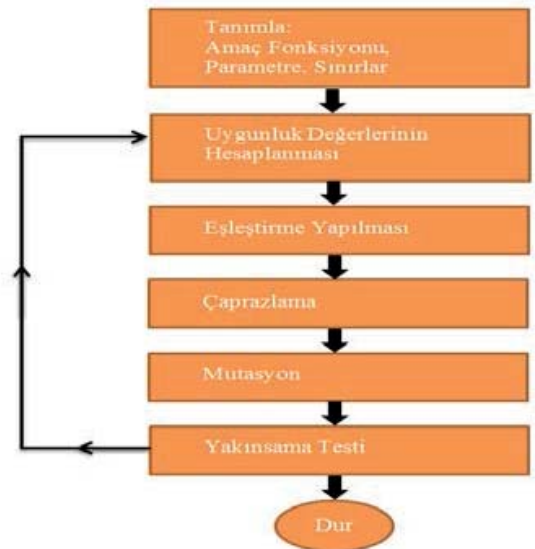
Bu fonksiyon ile kromozom içerisindeki kodlanmış ya da kodlanmamış bilgiler çözümlenerek sayısal bir değer elde (çıkı) edilir.

## 3. Problemin Çözülmesi ve Geliştirilen Yazılımın İncelenmesi

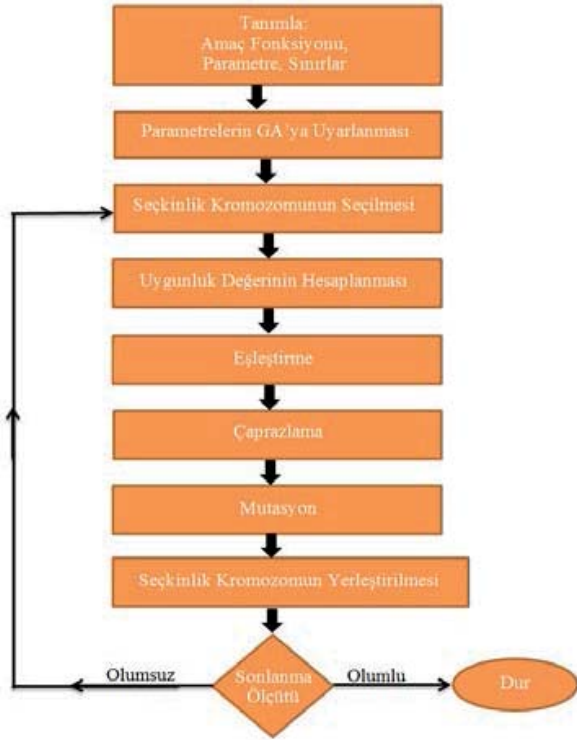
Bu çalışmada genetik algoritma kodlamalarından Gerçek Kodlu Genetik Algoritma kullanılmıştır.

Problemlerin çözümünde rakamların hassasiyeti önemli bir yer tutmaktadır. İkili kodlu GA'da, parametrelerin "1" ve "0" larla ifade edilmesi, kromozomların boyutlarını oldukça artırdığından sınırlı hassasiyete sahiptir. Bunun yerine gerçek rakamlarla kodlama yapabilen, gerçek kodlu GA'yı kullanmak avantajlıdır. Gerçek kodlu GA, hem daha hassas hem de PC belleğinde daha az yer kaplamaktadır. [17]'a göre, değişik test fonksiyonları kullanarak ikili kodlu GA ve gerçek kodlu GA'nın temel farklılıklarını irdelemişler. Gerçek kodlu GA'ların ikili kodlu GA'ya göre hızlı çalıştığını ve global optimumu daha kısa sürede bulduğunu göstermişlerdir.

Gerçek kodlu GA bazı literatürlerde "Sürekli Parametrelili GA" olarak da bilinmektedir. Bu çalışmada "Gerçek Kodlu GA" tanımlaması kullanılmıştır. Gerçek kodlu GA, ikili kodlu GA'ya çok benzemektedir. Aralarındaki en önemli farklılık parametrelerin "1" ve "0" lar yerine gerçek rakamlarla ifade edilmesidir. Gerçek kodlu GA'nın akış diyagramı şekil 2.'de verilmiştir. Şekil 3.'te ise geliştirilen yazılımın algoritması verilmiştir.



**Şekil 2.** Gerçek Kodlu GA'nın Akış Diyagramı



Program başlangıçta kullanıcıdan dışarıdan girdiği popülasyon sayısı, iterasyon sayısı, çaprazlama oranı ve mutasyon oranını almaktadır. Daha sonra problem durumuna göre (Örneğin Türkiye için nüfus artış tahmini yapılacaksa Türkiye'ye ait yıllara göre nüfus sayım bilgileri, aynı işlem bir il için yapılacak ise o ile ait yıllara göre nüfus sayım bilgileri) kullanıcı veri seç butonu vasıtasıyla ilgili excel (\*.xls/xlsx) formatındaki bilgileri seçerek veri yükle butonu ile verileri yazılıma yüklemektedir. Son olarak kullanıcı yüklediği verilere göre "giriş verisi gir" bölümünden eldeki verilere göre nüfus sayımlarına ait bir başlangıç tarihi, bir bitiş tarihi ve bir de nüfus artış sayısını tahmin etmek istediği bir tahmini yıl girmektedir. Bundan sonra "veriyi çek" butonu ile kullanıcının girdiği başlangıç ve son yıla göre daha önce sisteme yüklemiş olduğu veriden bu yıllara ait bilgileri sisteme aktarmaktadır. Bununla ilgili programın ana formu şekil 4.'te görüldüğü gibidir.

Şekil 3. Program Algoritması

Şekil 4. Sisteme Verilen Yüklenmesi ve Sistemden Verilerin Çekilmesi

Daha sonra bilgiler başarılı bir şekilde sisteme yüklendikten sonra kullanıcı, GA'nın popülasyonunu “**Başlangıç Popülasyonunu Oluştur**” butonuna basarak oluşturur. Başlangıç popülasyonu oluşturulurken referans alınan tarihler arasındaki nüfus sayım bilgilerine göre başlangıç tarihi nüfus değerleri ile son tarih nüfus değerleri bu iki tarih aralığındaki nüfus sayım değerlerine göre rastgele üretilirler. Yazılımda kısıt olarak verilen  $p$  değeri denklem (5) kullanılarak hesaplanır.

$$Taban = \frac{Rastgele \text{ Üretilen Son Nüfus Sayım Değeri}}{Rastgele \text{ Üretilen İlk Nüfus Sayım Değeri}} \quad (3)$$

$$\bar{U}s = 1 / (Son \text{ Nüfus Sayım Yılı} - İlk \text{ Nüfus Sayım Yılı}) \quad (4)$$

$$p = 100 * ((Taban^{\bar{U}s}) - 1) \quad (5)$$

olarak hesaplanır. Daha sonra yazılımda kısıt olarak verilen değer aralıklarına göre hesaplanmış  $p$  değerleri üzerinde ilgili dönüştürme işlemleri yapılarak  $g1$ ,  $g2$  ve  $g3$  değerleri hesaplanır. Daha sonra tahmini nüfus değeri denklem (8) kullanılarak hesaplanır.

$$Taban = 1 + (p/100) \quad (6)$$

$$\bar{U}s = 35 + Tahmini \text{ Yıl} - Son \text{ Yıl Nüfus Değeri} \quad (7)$$

$$Tahmini \text{ Nüfus} = Son \text{ Yıl Nüfus Değeri} * (Taban^{\bar{U}s}) \quad (8)$$

Bilindiği gibi kromozomlar yeniden oluşturulurken bir uygunluk fonksiyonuna göre seçilir ve yeni nesile aktarılırlar. Bu çalışmada hesaplanan tahmini nüfus değerlerini belirlenen kısıtlar içerisinde tutabilmek için aşağıda “**Penaltı**” olarak tanımlanan **uygunluk fonksiyonu** kullanılmıştır. Amaç kısıtları bozan bir durum oluştuğunda, amaç fonksiyonu değerlerini sınırlar içinde tutmaktır. Amaç (uygunluk) fonksiyonu aşağıda denklem (9) olarak ifade edilmiştir.

$$Penaltı = 100 * (g1^2 + g2^2 + g3^2) \quad (9)$$

Başlangıç popülasyonunun oluşturulmasıyla ilgili programın ana formu şekil 5.’te görüldüğü gibidir.

**Seçenekler - Fonksiyon**

Popülasyon Sayısı: 100  
İterasyon Sayısı: 200  
Çaprazlama Oranı: 0,7  
Mutasyon Oranı: 0,01

Veri Seç C:\Nüfus Bilgile Veri Yükle

Yıl Nüfus  
1927 13648

Giriş Verisi Gir  
Başlangıç Yılı: 1927  
Son Yılı: 1990  
Tahmin Yılı: 2020  
Veriyi Çek  
Başlangıç Nüfusu: 13648  
Son Yıl Nüfusu: 56473

Başlangıç Popülasyonunu Oluştur

Genetik Algoritma Başlat

**Kısıtlar:**  
 $1 \leq p \leq 3$  olmalıdır.  $p < 1$  ise  $p=1$   $p > 3$  ise  $p=3$

**Popülasyonu**

	BY	SY	BN	SN	P	G1	G2	G3	Penaltı	TY	TN
1	1948	1976	38510	39197	0,0631...	1	1	1	300	2020	6,7158...
2	1927	1934	17612	18810	0,9445...	1	1	1	300	2020	3,7153...
3	1953	1987	46440	48117	0,1043...	1	1	1	300	2020	1,2403...
4	1938	1946	45627	49017	0,8998...	1	1	1	300	2020	1,0791...

**Eşleştirme-Seçme-Doğal Seçim (Tumuva Seçimi)**

	kromozom1	kromozom2	Penaltı1	Penaltı2	Seçilen	TN
*						

**Çaprazlama Sonuçları**

	İterasyon Sayısı	1. kromozom No	2. Kromozom No	1. Kromozom Değer	2. Kromozom Değer	1. kromozom Yeni Değer	2. kromozom Yeni Değer	Çaprazlama Oranı Sayısı
*								

**Şekil 5.** Başlangıç Popülasyonunun Oluşturulması

Kullanıcı tarafından Şekil 5.'te görüldüğü şekilde başlangıç popülasyonunu, giriş verilerine göre rastgele seçtikten sonra kullanıcı “**Genetik Algoritma Başlat**” butonuna basarak belirlenen iterasyon sayısı kadar programı çalıştırır ve nüfus

artış tahmini hesaplar. Türkiye geneli nüfus sayımı örnekleri baz alınarak yapılan tahmin sonuçlarıyla ilgili yazılımın ana form görüntüleri şekil 6-7-8-9’da görüldüğü gibidir.

**Seçenekler - Fonksiyon**

Popülasyon Sayısı: 100    Veri Seç    C:\Users\Admin\    Veri Yükle

İterasyon Sayısı: 200    Yıl    Nüfus

Çaprazlama Oranı: 0.7    1927    13648

Mutasyon Oranı: 0.01

**Giriş Verisi Gir**

Başlangıç Yılı: 1927

Son Yılı: 2000

Tahmin Yılı: 2016

Başlangıç Popülasyonunu Oluştur    Genetik Algoritma Başlat

**Sonuçlar**

Max Tahmini Nüfus: 77417,1531631796 (30. kromozom)

Başlangıç Yılı: 1948    Başlangıç Nüfusu: 47554,600730814

Son Sayım Yılı: 1979    Son Sayım Nüfusu: 55067

Tahmini Yıl: 2016    Max Tahmini Nüfus: 77417,153163179

**Popülasyonu**

200. iterasyon

	BY	SY	BN	SN	P	G1	G2	G3	Penaltı	TY	TN
1	1948	1979	47589,...	45270	1,8364...	1	1	1	300	2016	40309,...
2	1948	1979	48262,...	33075	1,8364...	1	1	1	300	2016	13751,...
3	1948	1979	52342,...	54083	1,8364...	1	1	1	300	2016	58351,...
4	1948	1979	47954,...	45270	1,8364...	1	1	1	300	2016	39601,...

**Eşleştirme-Seçme-Doğal Seçim (Tumuva Seçimi)**

	kromozom 1	kromozom 2	Penaltı 1	Penaltı 2	Seçilen	TN
1	99	88	300	300	99	40309,01067206...
2	27	87	300	300	27	13751,17223263...
3	54	23	300	300	54	58351,75677618...
4	23	26	300	300	23	39601,34969102...
5	10	36	300	300	10	41435,21634012...

**Çaprazlama Sonuçları**

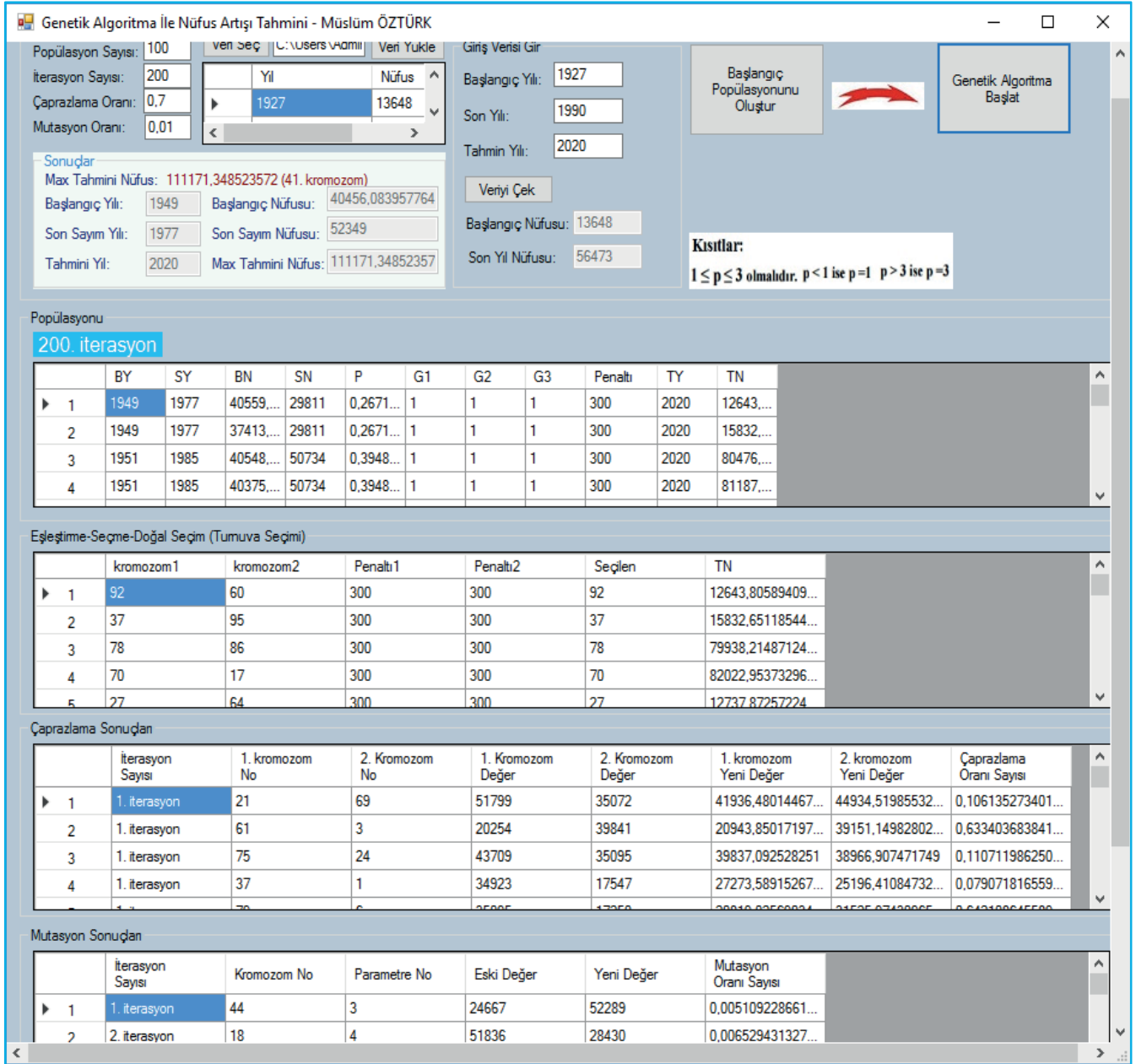
	İterasyon Sayısı	1. kromozom No	2. Kromozom No	1. Kromozom Değer	2. Kromozom Değer	1. kromozom Yeni Değer	2. kromozom Yeni Değer	Çaprazlama Oranı Sayısı
1	1. iterasyon	98	46	21592	26623	24916,45522771...	23298,54477228...	0,542675591326...
2	1. iterasyon	95	88	22986	19803	20959,524323683	21829,475676317	0,100961753214...
3	1. iterasyon	87	81	26623	53108	35935,52420565...	43795,47579434...	0,149508814862...
4	1. iterasyon	91	23	18964	24563	23867,05540767...	19659,94459232...	0,623608870722...

**Mutasyon Sonuçları**

	İterasyon Sayısı	Kromozom No	Parametre No	Eski Değer	Yeni Değer	Mutasyon Oranı Sayısı
1	1. iterasyon	27	3	31605,02588878	49520	0,008054970767

Şekil 6. 2016 Yılı İçin Türkiye Geneli Nüfus Tahmini

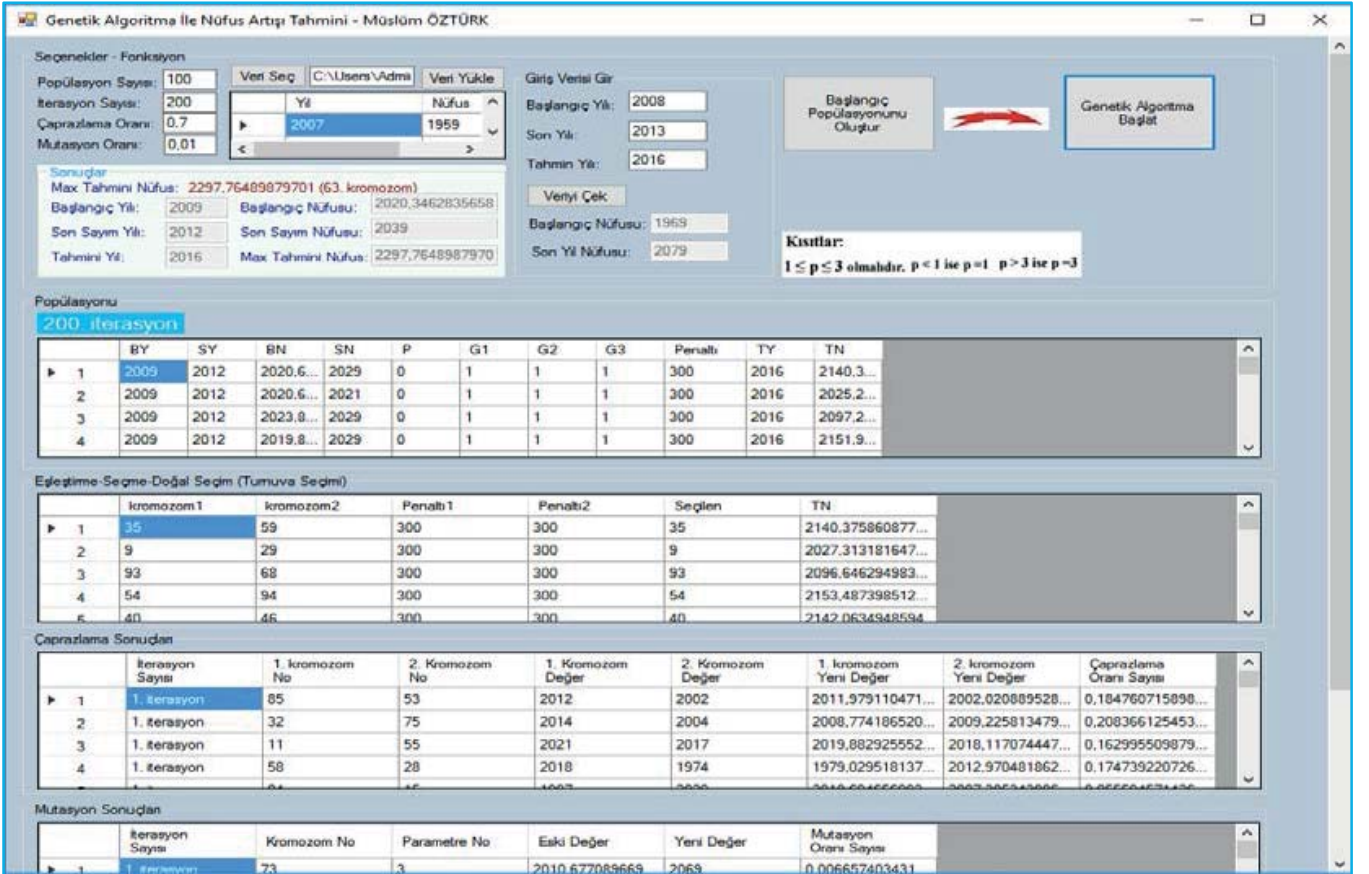




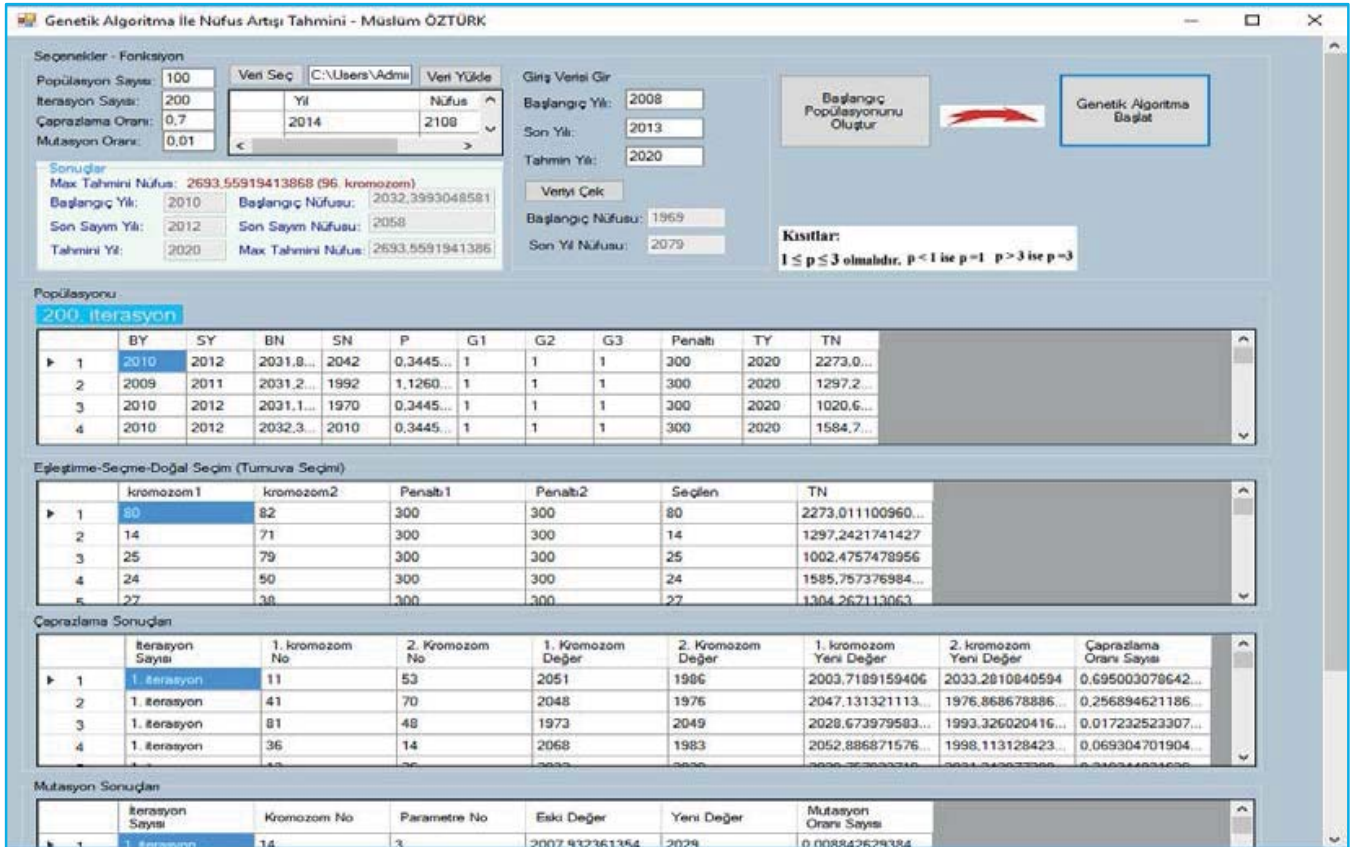
**Şekil 7. 2020 Yılı İçin Türkiye Geneli Nüfus Tahmini**

Kullanıcı program çalışırken ve çalıştıktan sonra son iterasyonun Turnuva Seçimi sonuçlarını, bütün iterasyonların çaprazlama sonuçlarını, mutasyon sonuçlarını ve seçkinlik kromozomu sonuçlarını görebilmektedir. Şekil 6.'da görüldüğü gibi TÜİK resmi sayım verilerine göre Türkiye geneli nüfus sayımı bilgileri baz alındığında (Sayım başlangıç yılı:1927, son sayım yılı:2000 ve tahmini yıl:2016) Türkiye'nin 2016 yılı için hesaplanan maksimum nüfus sayısı yaklaşık olarak 77 milyon 417 bin olarak hesaplanmıştır. En seçkin kromozom ise 30. Kromozom olarak seçildiği görülmüştür. TÜİK resmi verilerine göre ise Türkiye'nin geçtiğimiz

2016 yılı nüfus sayım bilgisi ise 79 milyon 814 bin olarak açıklanmıştır. Şekil 7.'de ise TÜİK resmi sayım verilerine göre Türkiye geneli nüfus sayımı bilgileri baz alındığında (Sayım başlangıç yılı:1927, son sayım yılı:1990 ve tahmini yıl:2020) Türkiye'nin 2020 yılı için hesaplanan maksimum nüfus sayısı yaklaşık olarak 111 milyon 171 bin olarak hesaplanmıştır. En seçkin kromozom ise 41. kromozom olarak seçildiği görülmüştür. Konya iline ait 2016-2020 yılı tahmini nüfus sayım bilgisi aşağıda şekil 8-9.'de yer almaktadır.



Şekil 8. 2016 Yılı İçin Konya Geneli Nüfus Tahmini



Şekil 9. 2020 Yılı İçin Konya Geneli Nüfus Tahmini

Şekil 8.'de görüldüğü gibi TÜİK resmi sayım verilerine göre Konya geneli nüfus sayımı bilgileri baz alındığında (Sayım başlangıç yılı:2008, son sayım yılı:2013 ve tahmini yıl:2016) Konya ilinin 2016 yılı için hesaplanan maksimum nüfus sayısı yaklaşık olarak 2 milyon 297 bin olarak hesaplanmıştır. En seçkin kromozom ise 63. Kromozom olarak seçildiği görülmüştür. TÜİK resmi verilerine göre ise Konya ilinin geçtiğimiz 2016 yılı nüfus sayım bilgisi ise 2 milyon 161 bin olarak açıklanmıştır.

Şekil 9.'da ise TÜİK resmi sayım verilerine göre Konya geneli nüfus sayımı bilgileri baz alındığında (Sayım başlangıç yılı:2008, son sayım yılı:2013 ve tahmini yıl:2020) Konya ilinin 2020 yılı için hesaplanan maksimum nüfus sayısı yaklaşık olarak 2 milyon 693 bin olarak hesaplanmıştır. En seçkin kromozom ise 96. Kromozom olarak seçildiği görülmüştür.

## 5. Sonuç ve Öneriler

Bu çalışmada, nüfus artışı tahmini yapan mevcut yazılımların kullanım karmaşıklığının ve zorluğunun önüne geçmek için genetik algoritmalarla gerçek kodlu genetik algoritma kullanılarak bir yazılım geliştirilmiştir. Bu yazılım .NET platformunun en güçlü dillerinden olan C# programlama dili kullanılarak kodlanmıştır. Gerçekleştirilen yazılımda TÜİK verilerinden yararlanılarak Türkiye geneli ve konya ili için nüfus tahmininde bulunulmuş ve yazılımın başarılı sonuçlar verdiği görülmüştür. Türkiye geneli için 2016 yılı TÜİK nüfus sayım bilgisi 79 milyon 814 bin iken geliştirilen yazılım tarafından hesaplanan tahmini nüfus sayısı ise 77 milyon 417 bin olarak hesaplanmıştır. 2020 yılı Türkiye geneli için maksimum tahmini nüfus sayısı ise 111 milyon 171 bin olarak tahmini hesaplama yapılmıştır. Konya ili için 2016 yılı TÜİK nüfus sayım bilgisi 2 milyon 161 bin iken geliştirilen yazılım tarafından hesaplanan tahmini nüfus sayısı ise 2 milyon 297 bin olarak hesaplanmıştır. 2020 yılı Konya ili için maksimum tahmini nüfus sayısı ise 2 milyon 693 bin olarak tahmini hesaplama yapılmıştır. Baz alınan ülke veya iller için iç ve dış göçler, doğal afetler, jeopolitik konum veya ülkeler arasında meydana gelebilecek savaşlar gibi çeşitli etmenler hesaplanmış tahmini nüfus sonuçlarını etkileyen başlıca unsurlar arasındadır. [22] tarafından yapılan gerçek kodlu genetik algoritma kullanılarak sistem kimliklendirme adlı çalışmada model

yapılarındaki farklı sayıdaki parametrelerin gerçek kodlu genetik algoritma, ikili kodlu genetik algoritma ve tabu araştırma algoritmaları ile karşılaştırmalı biçimde sunulmuş ve gerçek kodlu genetik algoritmaların sistem kimliklendirme ve model uydurma çalışmalarında oldukça tatmin edici bir performans sergilediği ortaya konulmuştur. Ayrıca bu çalışma, nüfus tahmininde genetik algoritmaların kullanılması açısından ilk ve tek çalışma olması açısından önemlidir. Bu çalışma kapsamında hesaplanan tahmini sonuçlar sadece TÜİK tarafından yapılan geçmiş yıllara ait sayım bilgileri dikkate alınarak hesaplanmıştır. Nüfus tahmini hesaplanması yapılacak olan ülke veya illerin, doğum-ölüm oranları, aldığı göçler, sınırlarındaki değişiklikler...vb nüfus artış oranını etkileyen faktörlerin de gerçekleştirilen yazılıma entegre edilmesiyle birlikte gelecek yıllar için hesaplanması yapılmak istenen tahmini nüfus sayısı daha az yanılma payı ile hesaplanabilir. Gelecekte yapılacak çalışmalar için ele alınan problem, differansiyel gelişim algoritması gibi çeşitli genetik algoritmalar kullanılarak nüfus tahmin değerleri elde edilerek sonuçlar karşılaştırmalı olarak verilebilir.

## Kaynakça

- [1] Bolat, B., Erol, K. O. ve İmrak, C. E. (2004). "Genetic Algorithms in Engineering Applications and the Function of Operators". Journal of Engineering and Natural Sciences, ss 264-271.
- [2] Satman, M. H. (2007). Durağan Zaman Serilerinde Uygun Arma Modelinin Genetik Algoritmalar İle Bulunması ve İMKB Verileri Üzerine Bir Uygulama. İktisat Fakültesi Mecmuası, 57(1).
- [3] Akbari, M., Rashidi, H. And Alizadeh, S. H. (2017). "An Enhanced Genetic Algorithm With New Operators For Task Scheduling In Heterogeneous Computing Systems". 61, pp 35-46.
- [4] Bağış, A., & Özçelik, Y. Gerçek Kodlu Genetik Algoritma Kullanılarak Sistem Kimliklendirme.
- [5] Chang, W. D. (2007). Nonlinear system identification and control using a real-coded genetic algorithm. Applied Mathematical Modelling, 31(3), 541-550.

- [6] Khalik, M. A., Sherif, M., Saraya, S., & Areed, F. (2007). Parameter identification problem: Real-coded GA approach. *Applied mathematics and Computation*, 187(2), 1495-1501.
- [7] Chuang, Y. C., Chen, C. T., & Hwang, C. (2015). A real-coded genetic algorithm with a direction-based crossover operator. *Information Sciences*, 305, 320-348.
- [8] Press, W. H. (2007). *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press
- [9] MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge university press.
- [10] Goldberg, D. (1989). *Genetic algorithms in optimization, search and machine learning*. Reading: Addison-Wesley.
- [11] Goldberg, D. E. (1991). Real-coded genetic algorithms, virtual alphabets, and blocking. *Complex systems*, 5(2), 139-167.
- [12] Kuo, H. C., & Lin, C. H. (2013). A directed genetic algorithm for global optimization. *Applied Mathematics and Computation*, 219(14), 7348-7364.
- [13] Nakanishi, H., Kinjo, H., Oshiro, N., & Yamamoto, T. (2007). Searching performance of a real-coded genetic algorithm using biased probability distribution functions and mutation. *Artificial Life and Robotics*, 11(1), 37-41.
- [14] Higuchi, T., Tsutsui, S., & Yamamura, M. (2000, September). Theoretical analysis of simplex crossover for real-coded genetic algorithms. In *International Conference on Parallel Problem Solving from Nature* (pp. 365-374). Springer, Berlin, Heidelberg.
- [15] Kita, H., Ono, I., & Kobayashi, S. (1999). Theoretical analysis of the unimodal normal distribution crossover for real-coded genetic algorithms. *Transactions of the Society of Instrument and Control Engineers*, 35(11), 1333-1339
- [16] Qi, X., & Palmieri, F. (1994). Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space. Part I: Basic properties of selection and mutation. *IEEE Transactions on Neural Networks*, 5(1), 102-119.
- [17] Çunkaş, M., & Akkaya, R. (2002). Compare with Binary and Real Coded Genetic Algorithms. *Selcuk University The Journal of Engineering*, 7(2), 11-17.
- [18] Eshelman, L. J., & Schaffer, J. D. (1993). Real-coded genetic algorithms and interval-schemata. In *Foundations of genetic algorithms* (Vol. 2, pp. 187-202). Elsevier.
- [19] Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.
- [20] Zhang, K., Du, H. and Feldman, M. V. (2017). "Maximizing influence in a social network: Improved Results Using a Genetic Algorithm", 478, ss 20–30.
- [21] Cunkaş, M. (2006). *Genetik Algoritmalar ve Uygulamaları*. Selçuk Üniversitesi, Teknik Eğitim Fakültesi, Konya, ss 1-78.
- [22] Bağış, A., & Özçelik, Y. Gerçek Kodlu Genetik Algoritma Kullanılarak Sistem Kimliklendirme.