Anemon Muş Alparslan Üniversitesi Sosyal Bilimler Dergisi 2018 6(1) 133–138

Journal of Social Sciences of Mus Alparslan University

anemon

Derginin ana sayfası: http://dergipark.gov.tr/anemon

Araştırma Makalesi ● Research Article

C# Programlama Dilinde Geliştirilen Program İle Euler Sayısının Rasgeleliğinin Sınanması*

Testing the Randomness of the Euler's Number Using the Program Developed in C # Programming Language

Arzu Eren Şenaras, [a,**] Şahin İnanç [b]

[a] Arş. Gör. Dr., Uludağ Üniversitesi, İktisadi ve İdari Bilimler Fakültesi, Ekonometri Bölümü, 16059, Bursa/Türkiye.
ORCID: 0000-0002-3862-4551

[b] Öğr. Gör., Uludağ Üniversitesi, Keles Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü, 16059, Bursa/Türkiye.
ORCID: 0000-0001-7739-3752

MAKALE BİLGİSİ

Makale Geçmişi:
Başvuru tarihi: 14 Temmuz 2017
Düzeltme tarihi: 09 Aralık 2017
Kabul tarihi: 23 Aralık 2017

Anahtar Kelimeler:
Benzetim
Rasgelelik Testleri
Rasgele Sayılar
Euler Sayısı
C# Program

ÖZ

Bu çalışmanın amacını, benzetim çalışmalarında kullanılabilmesi için sözde rasgele üretecinin rasgeleliğinin sınanması ve doğrulanması oluşturmaktadır. İlave olarak bu çalışmada rasgelelik testlerini yapmadan önce Euler sayısının elde edilişine değinilmiştir. Bunun için hangi hesaplama yöntemi kullanıldığı, hesaplanmanın matematiksel olarak daha basit bileşenlere ayrılması, bilgisayar programı ile nasıl programlanabileceği ve istenilen basamağa kadar basit bir şekilde nasıl hesaplanabileceğine yer verilmiştir. Bu çalışmada Euler sayısının ilk 100000 basamağı elde edilmiş ve beş farklı rasgelelik testi ile rasgeleliğinin sınanması yapılmıştır. Dizi Euler sayısının virgülden sonraki basamaklarından oluşturulmuştur. Bu dizinin rasgeleliğinin araştırılması için C# programlama dilinde bir program geliştirilmiştir. Rasgeleliğin sınanması için Ki-kare testi, Kolmogorov- Smirnov testi, Poker testi, Gap (Aralık) testi, Run (koşu) testi uygulanmıştır.

ARTICLE INFO

Article history:
Received 14 July 2017
Received in revised form 09 December 2017
Accepted 23 December 2017

Keywords:
Simulation
Randomness Tests
Random Numbers
Euler's Number
C# Program

ABSTRACT

The purpose of this work is to test and verify the randomness of the so-called random generator so that it can be used in simulation runs. Additionally, in this study, before making the randomness tests, it was mentioned how to obtain the number of Euler. In this study, it is performed that how this calculation method is used, how to divide the calculation into mathematically simpler components, how to program it with the computer program, and how to calculate it as simple as the desired step. In this study, the first 100000 steps of the Euler number were obtained and the randomness test was performed with five different randomness tests. The series is made up of the steps of the Euler number after comma. A program in the C # programming language was developed to investigate the randomness of this sequence. The chi-square test, the Kolmogorov-Smirnov test, the poker test, the gap test and the run test were applied to test the randomness.

1. Giriş

Benzetim tekniği, maliyetsiz olması, politikaların geliştirilen model sayesinde saniyelik bilgisayar zamanıyla karşılaştırılma imkânı sağlaması, deneylerin tekrarlanabilme olanağı ve sıra dışı koşulların etkilerini güvenlik açısından sorun yaratmadan öngörülmesi gibi sebeplerden dolayı günümüz yöneticilerinin en önemli karar araçlarından birisidir. Günümüzde rassal sayılar şifreleme, güvenlik, kodlama gibi birçok alanda kullanılmaktadır. Rassal sayıların en sık kullanıldığı alanlardan birisi de tartışmasız benzetim tekniğidir. Ancak kullanılan rassal sayıların rasgeleliği dikkat edilmesi gereken önemli bir noktadır. Altın oran, pi, e gibi irrasyonel sayıların rasgeleliğin bilinmesi

* Bu çalışma, 20-21 Mayıs 2017 tarihlerinde İstanbul'da düzenlenen International Congress of Management Economy and Policy kongresinde bildiri olarak sunulmuştur.

** Sorumlu yazar/*Corresponding author*.
e-posta: arzueren@uludag.edu.tr

http://dx.doi.org/10.18506/anemon.327324

farklı kullanım olanakları sunmaktadır. Bu konulara ilişkin yapılan literatür taramasına aşağıda yer verilmiştir. Sena, Agarwal, Shaykhian (2007) yapmış oldukları çalışmada, Monte Carlo çalışmalarında kullanılması amacıyla altın oran ve pi sayılarının uygun rasgele dizi olarak göreli yararlarını tartışmışlardır. Kim ve Neggers (2008), altın oran ortalaması ve fibonacci ortalaması arasındaki ilişkiyi araştırmışlardır. Kabirian ve Olafsson (2009) çalışmalarında, altın oran taraması olarak adlandırılan yeni SO (Simulation Optimization) yaklaşımını tartışmışlardır. Benavoli vd. (2009) yapmış oldukları çalışma, Fibonacci dizisinin altın oran ve diğer taraftan Kalman filtresi ile olan ilişkisini ortaya koymaktadır. Bu çalışmanın kapsamını; Euler sayısının 100000 basamağının rasgeleliğinin araştırılması oluşturmaktadır. Dizi Euler sayısının virgülden sonraki basamaklarından oluşturulmuştur. Bu dizinin rasgeleliğinin araştırılması için C# programlama dilinde bir program geliştirilmiştir.

2. Euler Sayısı

Tarihte e sayısına dolaylı olarak ilk değinen İskoç matematikçi John Napier (1560-1617) olmuştur (O'Connor ve Robertson, 2001b). Napier, 1618'de logaritmalar üzerine yayımlanan kitabının ekinde, çeşitli sayıların doğal logaritmalarını verirken $\log_e 10= 2.302585$ eşitliğini vererek sayısını kullanmıştır (Maor, 1994), fakat e' nin kendisiyle ilgilenmemiştir. Napier'den sonra William Oughtred (1574-1660), Henry Briggs (1561-1630), Gregorius Saint Vincent (1584-1667), Huygens (1629-1695) ve Nicolaus Mercator'da (1620-1687) logaritma ve e sayısını kullanmalarına rağmen bizzat e sayısı üzerinden çalışmalar yapmamışlardır. e sayısını gerçek anlamda ilk keşfeden Jacob Bernoulli (1654-1705) olmuştur (O'Connor ve Robertson, 2001b).

Bernoulli, 1683'te birleşik faiz problemini incelerken $\lim_{n\to e}\left(1+\frac{1}{n}\right)^n$ ifadesinin değeri üzerinde çalışmış ve bu esnada e sayısını keşfederek bu sayının yaklaşık değerini hesaplamıştır. e sayısının gösterimi önceleri Leibniz'in (1646-1716) Huygens'e 1690'da yazdığı mektupta *b* olarak gösterilmiştir. Ancak Euler, 1731'de Prusyalı bir matematikçi olan Christian Goldbach'a (1690-1764) yazdığı bir mektupta bu sabit sayıdan " e sayısı" diye bahseden ilk kişi olmuştur (O'Connor ve Robertson, 2001b). Euler 1748'de Introductio in Analysin infinitorum adlı çalışmasında $e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \frac{1}{5!} + \ldots\ldots$ eşitliğini göstererek e sayısına Euler sayısı adını vermiş ve $e = \lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n$ eşitliğini ortaya koymuştur. 18. yüzyıldan itibaren e sayısı ile ilgili pek çok gelişme yaşanmıştır. Bu tarihsel gelişim, bazı kaynaklardan (Avcı vd., 1995; Coolidge, 1950; Maor, 1994; O'Connor ve Robertson, 2001b; Sandifer, 2006; Yenilmez ve Palabıyık, 2008) faydalanarak özetlenmiştir. Pek çok insan e'nin irrasyonel olduğunu ilk ispatlayan kişinin Euler olduğunu kabul ederken, e'nin cebirsel bir sayı olmadığını ispatlayan kişinin Hermite olduğu yönünde kesin bir bilgi olduğunu belirtmektedir. Ancak incelenen çalışmalarda e'nin tarihi ile ilgili bazı çelişkilerde görülmüştür. Örneğin Hermite'in bu ispatının 1873 veya 1874 olduğu konusunda farklı ifadeler bulunmaktadır (Horzum, 2016).

3. Rassallık Testleri

Benzetim çalışmalarında önemli olan sahte rassal sayıların nasıl üretildiğinden çok, bu sayıların gerçekten rassal olup olmadıklarıdır. Sahte rassal sayıların gerçek rassal sayılarda doğal olarak bulunan rassallık özelliğine sahip olup olmadığı, rassallık testleriyle kontrol edilir. Sahte rassal sayılar deterministik olarak üretilse de, benzetimde kullanılabilmeleri için rassal olmaları gerekir. Benzetim diliyle konuşulurken telaffuz edilen rassal sayı, değeri 0 ile 1 arasında bulunan ve üretilme sansı bu aralıktaki diğer bütün sayıların üretilme sansına eşit olan sayıdır. Bu özellik kısaca U(0,1) olarak gösterilir. Rassal sayıların en önemli iki istatistiksel özelliği düzgün dağılıma uygunluk ve bağımsızlık olarak ifade edilebilir. (Banks vd., 2005). Bu nedenle sayıların rassallığını incelemeden önce sayıların tekdüze dağılıma uygun biçimde dağılıp dağılmadıkları araştırılmalıdır. Ardışık olarak üretilen sayılar arasında ilişki bulunmaması gerekir. Özetle, sahte rassal sayıların gerçek rassal sayılarla aynı istatistiksel özellikler taşıması beklenir. Bu amaçla geliştirilmiş testlere tabi tutulmaksızın sahte rassal sayıların kullanılması doğru olmaz.

Rassal sayı üreteçlerinin test edilmesine ilişkin olarak testler özelliklerine göre iki farklı kategoriye ayrılmaktadır. Bunlar üniform dağılıma ilişkin testler ve bağımsızlık testleri olarak adlandırılır. Söz konusu bu testlerden Kolmogorov – Smirnov ve Ki –Kare testleri üniform dağılıma ilişkin testler olmakla birlikte Run testi, Otokorelasyon testi, Gap testi ve Poker testi ise bağımsızlık testleri olarak bilinir (Banks vd., 1996: 298).

Bunlardan üniform dağılıma ilişkin olan her iki testte, rassal sayı üreteçleri tarafından üretilen rassal sayıların örnekleme dağılımı ile teorik üniform dağılımların arasındaki uyumun derecesini ölçer. Ayrıca, bu testler örnekleme dağılımı ile teorik dağılım arasında kayda değer bir farklılık olmadığını ifade eden sıfır hipotezi temel alınarak geliştirilmiştir (Banks vd., 1996: 299). Bu testler izleyen şekilde ele alınabilir.

3.1. Serpilme Diyagramları

Serpilme diyagramı (scatter diagram), (x, y) şeklindeki iki değişkene ait sıralı ikililer kartezyen koordinat sistemine yerleştirilerek, iki değişken arasında ne tür bir ilişki olduğunu araştırır (Gürsakal, 2001). Rassal sayı üreteçleri tarafından üretilen diziye istatistik testleri uygulamadan önce dizinin serpilme diyagramının çizilmesi aydınlatıcı olabilir. Serpilme diyagramı bilgisayar ortamında n gecikme olmak üzere, dikey eksene U_k ve yatay eksene U_{k-n} yerleştirilerek çizilir. Elde edilen diyagramda noktaların saçılımında açık bir desen olmaması diğer bir ifadeyle noktaların rassal saçılmaları beklenir. Açık bir desen görülmesi söz konusu diziyi üreten üretecin problemli olduğunu gösterir (Pidd, 2004: 186). Ripley (1977), bu yöntemle en çok ilgilenen kişidir (Sezen, 2009).

3.2. Ki-Kare Uygunluk Testi

Gözlemlenen frekanslar ile teorik (beklenen) frekanslar arasındaki fark Ki-kare testinin temelini oluşturur (Serper, 2010). Ki-kare testi Karl Pearson tarafından geliştirilmiştir.

Ki – kare uygunluk testi, n hacimlik bir örneklemin ana kütleyi iyi temsil edip edemediğini veya hangi bölünmeye sahip bir ana kütleden geldiği unsurlarının incelenmektedir. Gözlemlenen frekanslarla teorik frekanslar arasında az çok bir fark ortaya çıkarsa bu durumda ki – kare uygunluk testi bu farkın rassal sebeplere bağlanıp bağlanamayacağını araştırır.

3.3. Kolmogorov-Smirnov Uygunluk Testi

Bu test 1933'de Rus Matematikçisi A.N.Kolmogorov tarafından önerilmiştir. Kolmogorov tek örnek için uyum iyiliği testini önerdikten sonra 1939'da yeni bir Rus matematikçisi olan N.V. Smirnov iki bağımsız örnek için uyum iyiliği testini önermiştir. Kolmogorov testi ve Smirnov testi benzerlik nedeniyle uygulamada Kolmogorov-Smirnov uyum iyiliği testleri olarak bilinirler. Bu test üniform dağılımın sürekli dağılım fonksiyonu F(x) ile N adet gözlem setinden örneklem olarak alınmış olan ampirik sürekli dağılım fonksiyonu S_N (x)'i karşılaştırmaktadır.

Tanım olarak;

F(x) = x , $0 \le x \le 1$ verilebilir.

Ayrıca, eğer rassal sayı üreticilerinden alınan örneklem seti ; R_1 ,R_2 ,......,R_N olarak belirlenir ise, o zaman ampirik sürekli dağılım fonksiyonu S_N (x) şu şekilde tanımlanabilir;

$$S_N(x) = \frac{R_1, R_2, \ldots\ldots\ldots\ldots, (toplamsayısı) \le x}{N}$$

Bununla birlikte, N değeri büyüdükçe (gözlem sayısı arttıkça), S_N (x) fonksiyonu, F(x) fonksiyonuna daha iyi bir yaklaşım ile sıfır hipotezinin doğruluğunu sağlayacaktır.

Kolmogorov – Smirnov uygunluk testi, F(x) ve S_N (x) fonksiyonları arasındaki en büyük kesin sapmanın, rassal değişkenler dizisi üzerinden elde edilmesi ile uyarlanan bir testtir. Bu durum bir istatistik üzerinden uyarlamalı olarak ;

D max $|F(x) - S_N(x)|$ şeklinde ifade edilir.

Bircan, Karagöz ve Kasapoğlu 2003 yılında yapmış oldukları çalışma sonucunda; Ki-Kare uygunluk testi ile Kolmogorov-Simirnov tek örnek testlerinin aralarında önemli bir farklılık olmadığını, küçük örnekler için Ki-Kare uygunluk testi yerine kullanımı daha kolay ve ön şarta bağlı olmayan Kolmogorov-Simirnov testinin kullanılabileceği belirtmişlerdir (Bircan vd., 2003).

3.4. Diziler (Runs) Testi

Diziler testi n birimlik bir veri setinde, değerlerin gözlenme sıralarına göre rasgeleliğini sınamaktadır. n birimlik bir veri setini k gibi bir değere göre (K=Ortanca, K=Ortalama, K=a) ard arda gelişlerindeki kümelenmenin rasgelelik koşullarına uygunluğu diziler ile test edilir. Bir veri setinde değerlerin gözlenme sıralarına göre ard arda gelişlerinde k'dan küçük ya da büyük olmalarına göre oluşturdukları kümelere dizi(run) adı verilir. n birimlik bir veri setinde değerlerin birbirlerine bağımlı olarak sıralanıp sıralanmadıklarını araştırmak için gözlenen küme sayısı (run) r ile beklenen ortalama dizi sayısı arasındaki fark aşağıdaki test modeli ile test edilir (Özdamar, 2004).

$$z = \frac{r-\mu_r}{\sigma_r} = \frac{r - \dfrac{2n_1n_2}{n_1+n_2} + 1}{\sqrt{\dfrac{(2n_1n_2)(2n_1n_2 - n_1 - n_2)}{(n_1+n_2)^2(n_1n_2-1)}}}$$

Tek örneklem run anlamlılık testi genellikle bir örnekte rasgeleliğin test edilmesinde kullanılır. Ancak belirtilmesi gereken önemli bir nokta run testinin rasgeleliğin test edilmesi için gerekli olduğu fakat yeterli olmadığıdır. Ölçümlerin bazı belirlenmiş sıralamalara göre yapıldığı çalışmalarda sıkça sorulan sorulardan bir tanesi ölçümlerin ortalama değerlerinin seride farklı noktalarda farklılık gösterip göstermediğidir (Kalaycı, 2008: 97).

Üretilen sayıların belirli bir ortalamanın aşağısında ve yukarısında ya da altında ve üstünde yer alma durumunun gerçek değerler ile beklenen değerler karşılaştırılarak test edilmesine ilişkindir. Karşılaştırma için Ki – kare istatistiğinden yararlanılmaktadır. Knuth'a göre sadece bağımsızlığı sınayan diziler testi ki-kare testinden daha güçlü bir testtir. Çünkü üreteçlerin çoğu ki-kare testi ve serisel testten geçebilmektedir, ancak diziler testinden geçmede başarısız olmaktadırlar (Law ve Kelton, 2000).

3.5. Otokorelasyon Testi

Sayılar arasındaki korelasyonu test ederek, örneklem korelasyonu ile beklenen korelasyonu karşılaştırır. Otokorelasyon fonksiyonunun grafiği çizildiğinde ilişkileri yansıtan korelasyon katsayıları mutlak değerce sıfıra yakın olması u_t 'lerin ilişkisiz olduğunu gösterir.

3.6. Gap Testi

Bu test belirlenen bir aralıkta sıralı değerlerin oluşmaları arasındaki boşlukları aramaya yöneliktir (Sezen, 2009). (a,b) $\subset (0,1)$ olmak üzere, belli bir üreteç ile (0,1) aralığından üretilen $u_1, u_2, \ldots$.. sayılarına bağlı

$$b_n = \begin{cases} 1, u_n \in (a,b) \\ 0, u_n \notin (a,b) \end{cases}$$

sayıları üretilsin. b_1, b_2,............. dizisi 0,1'lerden oluşan biz dizidir. Dizide 1 rakamlarının aralarında bulunan 0 rakamlarının sayısı rasgeledir. Hatta, bir 1'den sonra ilk 1'in ortaya çıkışına kadarki 0'ların sayısı (X) geometrik dağılıma sahip olacaktır ve p=b-a olmak üzere,

$$f(x) = P(X = x) = p(1-p)^x, x = 0,1,2,\ldots..$$

dır. Böylece, b_1, b_2,.... dizisinde 1'ler arasında bulunan sıfırların sayıları gözlenerek geometrik dağılıma uyum testi yapılabilir (Öztürk ve Özbek,2004). Bu test belirlenen bir aralıkta sıralı değerlerin oluşmaları arasındaki boşlukları aramaya yöneliktir (Sezen, 2009). (a,b) $\subset (0,1)$ olmak üzere, belli bir üreteç ile (0,1) aralığından üretilen $u_1, u_2, \ldots$.. sayılarına bağlı

$$b_n = \begin{cases} 1, u_n \in (a,b) \\ 0, u_n \notin (a,b) \end{cases}$$

sayıları üretilsin. b_1, b_2,............ dizisi 0,1'lerden oluşan biz dizidir. Dizide 1 rakamlarının aralarında bulunan 0 rakamlarının sayısı rasgeledir. Hatta bir 1'den sonra ilk 1'in ortaya çıkışına kadarki 0'ların sayısı (X) geometrik dağılıma sahip olacaktır ve p=b-a olmak üzere,

$$f(x) = P(X = x) = p(1-p)^x, x = 0,1,2,.....$$

dır. Böylece, b_1, b_2,.... dizisinde 1'ler arasında bulunan sıfırların sayıları gözlenerek geometrik dağılıma uyum testi yapılabilir (Öztürk ve Özbek, 2004).

3.7. Poker Testi

Bir poker elindeki gibi sayıların gruplandırıldığı düşünülür. Daha sonra ise, Ki – kare testi kullanılarak elden elde edilen ile beklenen durum karşılaştırılır. Bununla birlikte, üniformluğun test edilmesinde hipotezler izleyen şekilde oluşmaktadır;

$H_0 : R_i = U[0,1]$

$H_1 : R_i \neq U[0,1]$

Burada boş (sıfır) hipotez olan H_0 hipotezi, [0,1] aralığı üzerinde sayıların üniform olarak dağıldığını ifade eder. Ayrıca bağımsızlık için test yapmada ilgili hipotezler şu şekildedir;

H_0: R_i = Bağımsız (olarak birbirini etkilemeden dağılış gösterir.)

H_1: $R_i \neq$ Bağımsız (olarak birbirini etkilemeden dağılış gösterir.)

Burada ise H_0 hipotezi, sayıların bağımsız olduklarını ifade etmektedir. Bunlara ilave olarak, karar alıcı burada ifade edilen her bir test için α değerini belirlemektedir. Bu α değeri sık sık 0.01 ya da 0.05 anlamlılık düzeylerinde belirlenmektedir (Banks vd., 1996: 298).

Poker testi, üretilen rasgele sayıların rasgele olup olmadığını incelemek için kullanılır. Uygulamada önce 5 ondalıklı sayılar üretilir ya da üretilen rasgele sayılar 5'erli gruplar haline sokulur. Basamaklar belirli bir poker eli gibi gruplanır. Olasılıklar her bir elin beklenen vuku bulması ile vuku bulma sayısının çarpımı olarak hesaplanabilir. Bu ise iki düzeltme faktörünün ve kombinatuar formül kullanılması ile başarılır.

İlk faktör elin görünüş(şekil) olasılığını temsil eder. İlk çift halinde ½ olan üçüncü faktöre ihtiyaç vardır. Çünkü AABBC ve BBAAC aynı sonucu verir, fakat iki farklı sonuç olarak kabul edilir.

4. Uygulama

Euler sayısının virgülden sonraki 100000 tane rakamının rasgelelik testleri ile sınanması için öncelikle 100000 basamak duyarlılıkta euler sayısının hesaplanması gerekir. Euler sayısını hesaplamak için birçok formül vardır. Bu çalışmada aşağıdaki formül kullanılmıştır.

$$e = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \cdots$$

Bu formülden de görüleceği gibi faktöriyel işlemlerinden 10-15 admı kadar sonrasında ve devamında çok büyük sayılar çıkacaktır. Euler sayısını hassas bir şekilde (100000 basamak) hesaplayabilmek için bu sayıların da bulunması gerekir. Bunun için klasik yöntemleri kullanarak hesaplama yapmak pek olası görünmemektedir. Bu çalışmada bu hesaplamaları yapmak için bilgisayar programı geliştirilmiştir. Program yazılmadan önce Euler sayısını hesaplamak için kullanılan formül biraz basitleştirilmiştir. Formüldeki faktöriyel işlemleri çarpım işlemlerine, çarpma işlemleri toplamaya ve bölme işlemleri de fark almaya dönüştürülüp hesaplanmıştır. Böylece programda sadece toplama ve çıkarma işlemlerine yönelik çok büyük diziler için yeni bir algoritma tanımlanarak sonuç bulunmuştur. Toplama ve çıkarma işlemleri için iki ayrı fonksiyon tanımlanmıştır. Toplama için kullanılan fonksiyon girilen iki sayının toplamını, çıkarma işlemi için kullanılan fonksiyon da girilen iki sayının farkını bulup geriye gönderir. Bu fonksiyonlar girdi olarak çok büyük sayı dizilerini (binlerce basamak) alıp çıktı olarak da çok büyük sayı dizileri üretir. Konuyu biraz daha anlaşılır kılmak için toplama ve çıkarma işlemlerinin nasıl yapıldığını ayrıntılı gösterirsek;

Toplama işlemi:

Sayı 1:
7395201942391871984164981636153174315981724012374128731924640564351

Sayı 2:
1982371946378426410301470287034723984627618351531724128746567585395

Programda yukarıdaki gibi çok büyük sayı dizilerini toplarken öncelikle her iki sayının basamak sayısı eşitlenir. Bunun için öndeki eksik hanelere sıfır eklenir. Sonra en sağdan başlanarak her iki diziden birer rakam alınıp toplanır. Bulunan sayını 10'a göre modu bulunup sonraki toplama işlemine devredilmek üzere hafızaya kaydedilir. İlk rakam için bulunan toplam sonucu ise yeni bir sayı tanımlanıp (Sayı 3 gibi) bu sayının sağına eklenir. İkinci işlemde bir önceki işlemden mod alınarak devreden sayı yeni işleme eklenir ve kalan işlemler aynen devam ettirilir. Bütün işlemler (döngü) bittiğinde Sayı 3 değişkeni iki sayının toplamını tutuyor olacaktır. Bu algoritma çarpma ve faktöriyel yerine de kullanılabilmektedir. Örneğin 4! = 1.2.3.4 = 1.2.3 + 1.2.3 + 1.2.3 + 1.2.3 = 1.2 + 1.2 + 1.2 + 1.2 + 1.2 + 1.2 + 1.2 + 1.2 + 1.2 + 1.2 + 1.2 + 1.2 = on iki tane ikinin toplamına dönüşmüş hali toplam fonksiyonu ile hesaplanıp bulunur. Benzer şekil çıkarma işlemi de toplamada olduğu gibi kullanılır. Bu işlem de bölme yerine kullanılır.

Çıkarma işlemi:

Sayı 1: 5

Sayı 2: 2 olsun

Sayı 1'den Sayı 2 çıkartılır ve Sayı 1'in değeri 3 olur. Bu çıkarma işlemi Sayı 1, Sayı 2'den küçük olana kadar her adımda devam ettirilir. Her çıkarma işleminde bölüm sonucunu saklamak için bir sayaç artırılır. İşlem bittiğinde sayaç bölümü tutacaktır. Sayı 1 Sayı 2'den küçük hale geldiğinde ise Sayı 1'in sonuna sıfır eklenerek işleme devam edilir. Sayı 1 Sayı 2'den küçük olduğunda bölüm sonucunu tutmak için kullanılan değişkene virgül eklenerek işlemlere devam edilir. Sayı 1 sıfırlandığında veya algoritmada belirtilen iterasyon limitine ulaşıldığında işlem durdurulur.

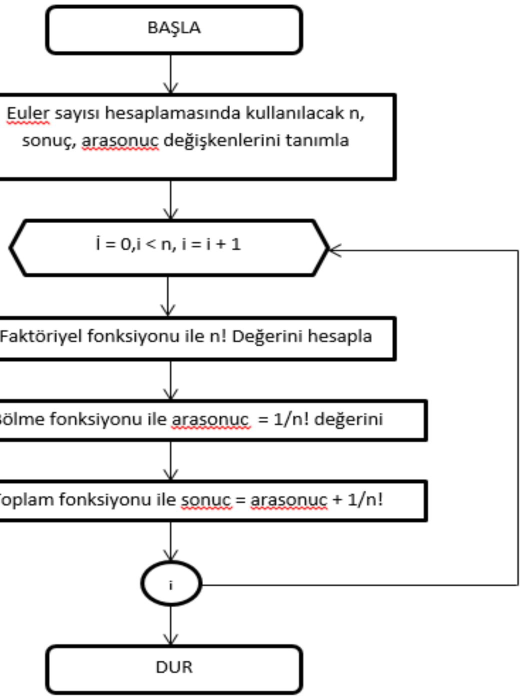
Çıkarma işlemi için kullanılan fonksiyon da toplam için kullanılan fonksiyona benzer mantıktadır. Her iki fonksiyon da mantık olarak ilkokul matematiğinde sayıları toplamak ve çıkarmak için kalemle yapılan işlemlere benzemektedir.

Bu işlemler için bilgisayarda zaten hâlihazırda olan çarpma ve benzeri işlemleri kullanmamamızın sebebi bilgisayarın bu tür işlemler için desteklediği sayı büyüklüğü ancak 30 basamak civarında olmasındandır. Binlerce basamaklı sayılar için çarpma, bölme gibi işlemler yapılamamaktadır. Bu sebeple Euler sayısını hesaplamak için bu çalışmada bu algoritmalar geliştirilip programlanmıştır.

Tablo 1. Euler Sayısını Hesaplayan Algoritma

1.	Başla
2.	Değişkenleri tanımla: Sonuc = 0, i, arasonuc
3.	Adım: Euler sayısını hesaplamada kullanılacak n değerini gir
4.	0’dan n’e kadar çalışacak bir döngü aç (i=0,i<=n,i=i+1)
5.	Döngüde i değeri için i faktöriyeli hesapla. Faktöriyel hesaplarken i faktöriyeli toplam şekline dönüştürüp toplam fonksiyonunu kullanarak faktöriyel sonucunu bul
6.	arasonuc = 1/i! Değerlerini bul. arasonuc değerlerini hesaplarken bölme işlemi yerine fark işlemi fonksiyonunu kullanarak sonucu hesapla
7.	Sonuc değişkenine arasonucu ekle
8.	Döngü sona erene kadar Adım 5’e git
9.	Sonuc değişkeni Euler sayısını içeriyor olacak. Sonuc değişkenini yazdır
10.	Dur
Yardımcı Fonsiyonlar	
Toplama Fonsiyonu Algoritması:	
1.	Başla
2.	Toplanacak iki sayıyı gir Sayi1, Sayi2, sonuç, akılda = 0
3.	N = sayıların rakam sayısını bul
4.	Sayıların rakam sayıları aynı değil ise rakam sayısı az olanın önüne sıfırlar ekle
5.	N elemanlı bir döngü aç (i=0,i<=N,i=i+1)
6.	Sayıların i indeks numaralı rakamlarını alıp topla ve üzerine akılda değişkenini ilave et
7.	Toplamın 10’a göre modunu al
8.	Kalanı akılda değişkenine yaz
9.	Toplamı sonuç değişkenine sondan ekle
10.	Döngü bitmedi ise Adım 6’ya git
11.	Sonuç değişkeni iki sayı dizisinin toplamını içeriyor olacaktır. Sonuç değişkenini fonksiyon değerine aktar
12.	Dur
Çıkarma Fonksiyonu Algoritması:	
1.	Başla
2.	Toplanacak iki sayıyı gir Sayi1, Sayi2, sonuç, akılda = 0
3.	N = sayıların rakam sayısını bul
4.	Sayıların rakam sayıları aynı değil ise rakam sayısı az olanın önüne sıfırlar ekle
5.	N elemanlı bir döngü aç (i=0,i<=N,i=i+1)
6.	Sayıların i indeks numaralı rakamlarının farkını al ve üzerine akılda değişkenini ekle
7.	Sonuç negatif çıkar ise ilk sayıya 10 ilave edip akılda değişkenini bir eksilt
8.	Fark sonucu ilk sayıya 10 ilave edilmiş hali ile tekrar hesapla
9.	Bulunan sayıyı sonuç değişkenine sondan ekle
10.	Döngü bitmedi ise Adım 6’ya git
11.	Sonuç değişkeni iki sayı dizisinin toplamını içeriyor olacaktır. Sonuç değişkenini fonksiyon değerine aktar
12.	Dur

Şekil 2. Euler Sayısının Hesaplanmasına İlişkin Akış Çizgesi



Ki-Kare Testi :

Ki kareTablo : 9998,5

H_1: 8,1574646652704

H_0 reddedilemiyor. Dizide yer alan sayıların rassal olduğu söylenebilir.

RUN Testi :

Z (0.01) : 10366,0919780366

H_1: -1,53392038790529

H0 reddedilemiyor. Dizide yer alan sayıların rassal olduğu söylenebilir.

Poker Testi :

X (0.01): 10366,0919780366

H_1: 1,81353585465336

H_0 reddedilemiyor. Dizide yer alan sayıların rassal olduğu söylenebilir.

Kolmogorov & Smirnov Testi :

D (0.01) : 0,0163016302445408

H_1 : 0,00504050405040513

H_0 reddedilemiyor. Dizide yer alan sayıların rassal olduğu söylenebilir.

Gap Testi:

0 sayısı : 0,00269026902690499

1 sayısı : 0,00119011901190351

2 sayısı : 0,000310031003098024

3 sayısı : 0,000710071007098453

4 sayısı : 0,00189018901890417

5 sayısı : 0,000890089008903153

6 sayısı : 0,00781078107810551

7 sayısı : 0,000710071007098398

8 sayısı : 0,000690069006902966

9 sayısı : 0,00319031903190548

Bütün rakamlar için ayrı ayrı değerler D (0.01) = 0,0163008150611301 değerinin altında olduğundan H_0 reddedilemiyor. Dizide yer alan sayıların rassal olduğu söylenebilir.

5. Sonuç

Çalışmanın uygulama bölümünde irrasyonel bir sayı olarak kabul edilen Euler sayısının ondalık basamaklarının rasgeleliğinin araştırılması amacıyla C# programlama dilinde geliştirilen program ile uygulanan rassallık testlerine ilişkin sonuçlara yer verilmiştir. Euler sayısı hesaplandıktan sonra geliştirilen rassallık ile ilgili bilgisayar programları kullanılmıştır. Euler sayı dizisi virgülden sonraki 100000 basamak için bu testlerden geçmiştir.

Söz konusu basamaklara Ki-kare testi, Kolmogorov-Smirnov (K-S) testi, poker testi, aralık testi ve koşu testleri uygulanmıştır. Rasgelelik testlerinin sonuçları incelendiğinde testlerin tamamında dizinin düzgün dağılıma uygunluğunu veya dizi elemanlarının birbirinden bağımsızlığını ifade eden sıfır hipotezinin reddedilemediği görülmüştür. Bu sonuç söz konusu dizi elemanlarının rasgele sayılar olduğunu ima etmektedir. Bu testler hiçbir zaman kesin bir uygunluğu ispat etmemektedir, söz konusu testleri geçen üreteçler kontrol edilmemiş bir özelliği barındırıyor olabilir. Dolayısıyla uygulanan testlerde testlerin sadece red sonucunun anlamlı olduğu belirtilmektedir. Ancak tam rasgeleliğin sağlanmasının neredeyse imkânsız olması nedeniyle, rassal sayılarla çalıştırılan benzetim modelinin sistemin gerçek davranışını yansıtabilmesi için kullanılan sayıların birkaç istatistik testi geçmesi yeterli görülmektedir. Yapılan altı adet istatistiksel testi geçen dizinin benzetim çalışmaları başta olmak üzere rassal sayılara ihtiyaç duyulan alanlarda kullanılabileceği söylenebilir.

Kaynakça

Banks, J., Carson, J.S., & Nelson, B. L., (1996). *Discrete-Event System Simulation*. New Jersey: Prentice Hall Inch.

Banks, J., Carson, J.S., Nelson, B. L., & Nicol, D.M. (2005). *Discrete-Event System Simulation*. New Jersey: Prentice Hall Inch.

Benavoli, A., Chisci, L., & Farina, A. (2009). Fibonacci sequence, golden section, Kalman filter and optimal control. *Signal Processing, 89*(8), 1483-1488.

Bircan, H., Karagöz, Y., & Kasapoğlu, Y. (2003). Ki-Kare ve Kolmogorov Smirnov Uygunluk Testlerinin Simülasyon İle Elde Edilen Veriler Üzerinde Karşılaştırılması. *Cumhuriyet Üniversitesi İktisadi ve İdari Bilimler Dergisi*, 4(1), 69-80.

Gürsakal, N. (2001). *Bilgisayar Uygulamalı İstatistik I*. Bursa: Alfa Basım Yayım Dağıtım.

Kim, H. S., & Neggers, J. (2008). Fibonacci mean and golden section mean. *Computers & Mathematics with Applications, 56*(1), 228-232.

Horzum, T. (2016). İrrasyonel Sayıların Öğretimi İçin Görsel Model Önerisi: e ve π sayıları. *Bilecik Şeyh Edebali Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 1, 42-57.

Kabirian, A., & Olafsson, S. (2009). Simulation optimization with hybrid golden region search. In: *Simulation Conference (WSC), Proceedings of the 2009 Winter* (pp. 551-562). IEEE.

Kalaycı, Ş. (2008). *SPSS Uygulamalı Çok Değişkenli İstatistik Teknikleri*. Ankara: Asil Yayın Dağıtım.

Law, A. M., & Kelton, W. D. (2000). *Simulation Modeling And Analysis*. Boston: Mcgraw Hill.

O'Connor, J. J., & Robertson, E. F. (2001). *The Number e*. (Erişim: 21.04.2017), http://www-history.mcs.standrews .ac.uk/ HistTopics/e.html

Özdamar, K. (2004). *Paket Programlar İle İstatistiksel Veri Analizi*. Eskişehir: Kaan Kitabevi.

Öztürk, F., & Özbek, L. (2004). *Matematiksel Modelleme ve Benzetim*. Ankara: Gazi Kitabevi.

Pidd, M. (2004). *Computer Simulation In Management Science*. Chichester: John Wiley & Sons.

Sen, S. K., Agarwal, R. P., & Shaykhian, G. A. (2008). Golden Ratio Versus Pi As Random Sequence Sources For Monte Carlo Integration. *Mathematical and Computer Modelling, 48*(1-2), 161-178.

Yenilmez, K., & Palabıyık, U. (2008). e sayısı ve kayıp tarihi. *XXI. Ulusal Matematik Sempozyumu*, Koç Üniversitesi, İstanbul, H. 1-10.