Π SAYISININ RASGELELİĞİNİN SINANMASI

Şahin İNANÇ

(Öğr.Gör. **,** Uludağ Üniversitesi/ Keles Meslek Yüksekokulu, Bilgisayar Teknolojileri

sahininanc@uludag.edu.tr)

H. Kemal SEZEN

(Prof. Dr., Uludağ Üniversitesi/ Ekonometri)

ÖZET

Benzetim çalışmalarının bir parçası olarak rasgele sayılar genellikle bilgisayarlar tarafından üretilir. Bu çalışmanın amacını, benzetim çalışmalarında kullanılabilmesi için sözde rasgele üretecinin rasgeleliğinin sınanması ve doğrulanması oluşturmaktadır. İlave olarak bu çalışmada rasgelelik testlerini yapmadan önce π 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 π sayısının ilk 10000 basamağı elde edilmiş ve beş farklı rasgelelik testi ile rasgeleliğinin sınanması yapılmıştır. Dizi π 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.

Anahtar Kelimeler: *Benzetim, Rasglelelik Testleri, Rasgele Sayılar, Π Sayısı, C# Program.*

Jel Kodu: *C4, C40.*

TESTING RANDOMNESS OF GENERATED NUMBERS BASED ON Π NUMBER

ABSTRACT

As a part of their simulation work, random numbers are usually generated by computers. 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, the acquisition of the number of Π before the randomness tests were mentioned. 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 10000 steps of the Π number were obtained and the randomness test was performed with five different randomness tests. The series is made up after the comma next steps of the number of Π. 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.*

Key Words: *Simulation, Randomness Tests, Random Numbers, Π Number, C# Program.*

Jel Codes: *C4, C40.*

1. GİRİŞ

Benzetim tekniği, maliyetsiz olması, politikaların geliştirilen model sayesinde saniyelik bilgisayar zamanıyla karşılaştırılma imkanı 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 farklı kullanım olanakları sunmaktadır.

Bu çalışmanın kapsamını; Π sayısının 10000 basamağının rasgeleliğinin araştırılması oluşturmaktadır. Dizi Π sayısının virgülden sonraki basamaklarından oluşturulmuştur. Bu dizinin rasgeleliğinin araştırılması için C# programlama dilinde bir program geliştirilmiştir.

2. LİTERATÜRE KISA BAKIŞ

Bu konulara ilişkin yapılan literatür taramasına aşağıda yer verilmiştir. Sena vd. (2007) yapmış oldukları çalışmada, Monte Carlo çalışmalarında kullanılması amacıyla altın oran ve pi sayılarının uygun rasgele dizi olarak göreli yararlarını tartışmışlardır. Hee 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. Şenaras vd. (2014), çalışmalarının amacı; lojistik harita ile üretilen dizinin rasgeleliğinin araştırılmasıdır. Bu dizinin benzetim çalışmalarında kullanılabilirliğinin araştırılması temeline dayalı C# programlama dilinde rasgeleliğinin test edilmesi için bir program geliştirilmiştir. Şenaras vd. (2015), yapmış oldukları çalışmada birçok ilginç yönüyle her alanda karşımıza çıkan ünlü altın oran sayısının virgülden sonraki 49999 hanesinin rasgelelik testleri C# programa dilinde geliştirilen program aracılığıyla gerçekleştirilmiştir. Şenaras ve İnanç (2016), yapmış oldukları çalışmada farklı “a” parametreleriyle üretilen çeşitli henon haritalarının rasgeleliğinin sınanmasını amaçlamışlardır. Henon haritasını oluşturmak için C# programlama dilinde geliştirilen bir yazılım kullanmışlardır. Harita oluşturulurken özellikle a katsayısının 1,4 civarındaki değerleri denenmiştir. a katsayısının 1,4’e çok yakın değerleri için sistemin kaotik olduğunu ortaya koymuşlardır.

2.1. Π Sayısı

Pi (π) yunan alfabesinin 16. harfi ve Yunanca “çevre (περίμετρον)” sözcüğünün ilk harfidir. Matematik dünyasında ve fen bilimlerinde önemli bir yere sahip olan π sayısının Şu anki değerini hesaplamak için günümüze kadar pek çok bilim insanı yıllarını adamıştır. Nitekim π sayısının geçmişinin Eski Mısır ve Mezopotamya'ya dayandığı belirtilmektedir (Horzum, 2016: 44).

M.Ö. 2000 yılı civarında; r'nin tüm çemberler için sabit bir değer olduğunun farkına varıldı. Böylece π sayısının serüveni başlamış oldu. Pi sayısının esrarının anlaşılması ile ilgili çalışmaları incelediğimizde karşımıza çıkan ilk bulgu, Mısırlıların ve Babillilerin kullandıkları π değerinin, bugün bilinen sayısal değerine yakın olduğudur (Gültekin ve Asyalı, 2007).

Tablo 1'de π sayısının hesaplanmasındaki hassasiyetin (doğru basamak sayısının) tarihsel gelişiminin bir özeti sunulmuştur.

Tablo 1. Pi' nin Nümerik Tarihi

Kişi	Zaman	Doğru Basamak Sayısı
Archimedes	M.Ö. 240	3
Ptolemy	150	3
Liu Hui	263	5
Tsu Ch'ung Chi	480?	7
Al-Kashi	1429	14
Romanus	1593	15
Van Ceulen	1615	35
Sharp	1699	71
Machin	1706	100
Strassnitzky ve Dase	1844	200
Rutherford	1853	440
Shanks	1874	527
Reitwiesner ve ark. (ENIAC)	1949	2,037
Genuys	1958	10,000
Shanks ve Wrench	1961	100,265
Guilloud ve Bouyer	1973	1.001.250
Miyoshi ve Kanada	1981	2.000.036
Kanada, Yoshino ve Tamura	1982	16.777.206
Gosper	1985	17.526.200
Bailey Jan.	1986	29.360.111
Kanada ve Tamura	Eylül 1986	33.554.414
Kanada ve Tamura	Ekim 1986	67.108.839
Kanada ve ark.	Ocak 1987	134.217.700
Kanada ve Tamura	Ocak 1988	201.326.551
Chudnovskys	Mayıs 1989	480.000.000
Kanada ve Tamura	Temmuz 1989	536.870.898
Kanada ve Tamura	Kasım 1989	1.073.741.799
Chudnovskys	Ağustos. 1991	2.260.000.000
Chudnovskys	Mayıs 1994	4.044.000.000
Kanada ve Takahashi	Ekim 1995	6.442.450.938
Kanada ve Takahashi	Temmuz 1997	51.539.600.000
Kanada ve Takahashi	Eylül 1999	206.158.430.000
Kanada, Ushiro ve Kuroda	Aralık 2002	1.241.100.000.000

(Kaynak: Gültekin ve Asyalı, 2007)

2.2.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 v.d., 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 v.d., 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 v.d., 1996: 299). Bu testler izleyen şekilde ele alınabilir.

2.2.1. 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).

2.2.2. 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. Ki-kare (χ^2) uygunluk testinde izlenen adımlar aşağıdaki gibidir.

Adım : 1

χ^2 uygunluk testinde ilk adım olarak amaca ilişkin hipotezler oluşturulmalıdır. Burada;

H_0 : Örneklem ana kütleyi temsil edebilir.

H_1 : Örneklem ana kütleyi temsil edemez.

hipotezleri yazılabilir.

Adım : 2

"Anlamlılık düzeyi" için %1 ve %5 düzeylerinden biri, kararın etkilenmemesi için öncelikle belirlenir.

Adım : 3

" Red bölgesi " ise, şu şekilde tanımlanabilir;

Red bölgesi : (Hesaplanan test istatistik değeri) $\chi^2_{hes} > \chi^2_t$ (Tablo değeri).

Adım : 4

χ^2_{hes} istatistiğinin bölünmesi χ^2_t bölünmesine çok yaklaştığı için, test istatistiği belli bir anlamlılık düzeyine ve k-1 serbestlik derecesine göre mevcut bir "χ^2 Değerleri Tablosundan" bulunan kritik değerler ile karşılaştırılmaktadır. ($\chi^2_{hes} \leftrightarrow \chi^2_t$)

" Test istatistiği " burada ;

$$\chi^2_{hes} = \sum_{i=1}^{k} \frac{(G_i - B_i)^2}{B_i}$$

formülüne göre hesaplanır. Burada G_i; i. sınıftaki gözlemlenen frekansı, *Bi*, i. sınıftaki beklenen frekansı ve k ise sınıf sayısını göstermektedir. Yukarıda red bölgesi, (χ^2_{hes})'nın (χ^2_t)'den büyük olduğu bölge seklinde tanımlanmıştır. Bu tanımlamaya göre $\chi^2_{hes} < \chi^2_t$ olduğunda H_0 hipotezi kabul edilirken, $\chi^2_{hes} \geq \chi^2_t$ olduğunda ise reddedilir.

Bu testte dikkat edilmesi gereken, her sınıfın teorik (beklenen) ve gözlenen frekanslarının 5'ten küçük olmamasıdır. Böyle bir durumla karşılaşıldığında sınıfların birleştirilmesi yoluna gidilir. Ancak örnek sayısı az olduğunda bunu gerçekleştirmek mümkün olmayabilir.

2.2.3. 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 , $0 \leq x \leq 1$ 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;

$$S_N(x) = \frac{R_1, R_2, ..., (toplamsayısı) \leq x}{N}$$

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 ;

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

Bu ifade edilenlerin ışığında, üniform sürekli dağılıma sahip bir fonksiyonun test edilmesine ilişkin olarak test yöntemi şu adımları izlemektedir;

Adım : 1

Söz konusu verilerin küçükten büyüğe doğru sıralanmasıdır. Bu bakımdan, (i) R ’ nin i’ninci en küçük gözlemi belirttiği düşünülerek;

Dolayısıyla ;

$R_{(1)} \leq R_{(2)} \leq R_{(N)}$ seklinde sıralanabilir.

Adım : 2

$$D^{+} = \max_{1 \leq i \leq N} \{ i / N - R_{(i)} \}$$

$$D^{-} = \max_{1 \leq i \leq N} \{ R_{(i)} - (i-1) / N \}$$

Adım : 3

$D = \max(D^{+}, D^{-})$ hesaplanır.

Adım : 4

α anlamlılık düzeyi ve verilen N örneklem büyüklüğü ile D_{α} , kritik değeri tablo yardımıyla tanımlanır.

Adım : 5

Karar aşaması olarak adlandırılır. Eğer örneklem istatistiği D , D_{α} ’dan daha büyük ise, o zaman örneklemin bir üniform dağılımdan geldiği bilgisini veren sıfır hipotezi reddedilir. Eğer $D \leq D_{\alpha}$ ise, sonuç olarak{ $R_{(1)}$,$R_{(2)}$,…………., $R_{(N)}$ } ’in doğru dağılımı ile üniform dağılım arasında hiçbir fark olmadığı anlaşılır(Banks v.d., 1996: 299-300).

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 v.d., 2003).

2.2.4. 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).

2.2.5.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.

2.2.6. 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,

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

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

2.2.7. 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 v.d., 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.

3. UYGULAMA

Pi sayısının hesaplanmasında $4\sum_{k=0}^{\infty}\frac{(-1)^k}{2k+1} = 4\left(1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\frac{1}{9}-\frac{1}{11}\cdots\right)$ formülü kullanılmıştır. Bu formüle göre Pi sayısı hesaplanırken sadece toplama işlemi (fark işlemi de toplamanın bir türevi) kullanmak yeterli olmaktadır. Parantez önündeki 4 ile çarpa işlemi ise parantez içindeki işlemin üst üste dört kere toplanması ile elde edilebilmektedir. Sonuç olarak sadece toplama işlemi ile bulunabilmektedir.

Pi sayısı hesaplanırken sayı basamakları çok büyük olduğundan toplama işlemi bildiğimiz toplama şeklinde değil de özel bir fonksiyon aracılığı ile yapılmaktadır. Bu fonksiyon için algoritma aşağıda yer almaktadır.

Pi Sayısının Hesaplanması İçin Algoritma:

1. Adım: Başla
2. Sonuç, arasonuç, n, i değişkenlerini tanımla
3. Adım: n elemanlı bir döngü aç. Döngünün sayacı i = 1 den başlayıp ikişer ikişer sayacak şekilde ayarlanmalı. (Döngü 1 den başlayarak tek sayıları sayacak şekilde ayarlanmış olacak)
4. Adım: arasonuç değişkenine bölme fonksiyonu aracılığı ile 1/i hesaplanıp toplama fonksiyonu aracılığı ile üzerine ilave et
5. Adım: 4 defa çalışacak bir döngü aç
6. Adım: sonuç değişkenine arasonuç değişkenini ilave et (Döngü sona erdiğinde sonuç değişkeni 4 tane arasonuç değerinin toplamını tutuyor olacak)
7. Dur

Toplam Fonksiyonu İçin Örnek Algoritma:

Çok büyük sayı dizilerinin toplamını bilgisayar aracılığı ile hesaplanırken normal integer (tamsayı) tipinde değişken kullanarak hesaplamada bazı güçlükler vardır. Integer tipindeki değişkenlerin basamak sayısı sınırlıdır (En fazla 30 basamak civarı vs. olabilmektedir). Bu sebepten dolayı çok büyük iki sayının (100.000 basamaklı iki sayı gibi) toplamını bulmak için öncelikle sayıları string (kelime katarı) şeklinde tanımlamak gerekir. Toplama işleminin yapmak için de bir seferde bütün sayıyı toplamak yerine sayı dizisini tek tek basamaklara ayrıştırıp sondan başlayarak her bir rakamı ayrı ayrı toplayıp başa doğru giderek yapılmaktadır.

Algoritma kısaca aşağıdaki şekildedir.

Adım: Başla

Adım: x, y, toplam, rakamtoplam, devreden, kalan değişkenlerini tanımla (x ve y değişkenleri toplanacak olan sayıları tutacaktır)

Adım: x ve y sayılarının küçük olanının rakam sayısı kadar işlem yapacak bir döngü oluştur.

Adım: Döngünün her adımında x ve y sayılarının sonundan başlayarak birer rakamını toplayıp sonuç üzerine devreden değişkenini ilave et ve rakamtoplam değişkenine aktar.

Adım: rakamtoplam 9'dan büyük çıkarsa devreden değerini 1 yap ve rakamtoplam değerini 10'a göre modunu alıp kalan sonucu rakamtoplama aktar. Rakamtoplam 9'dan büyük çıkmazsa devreden değerini 0 yap.

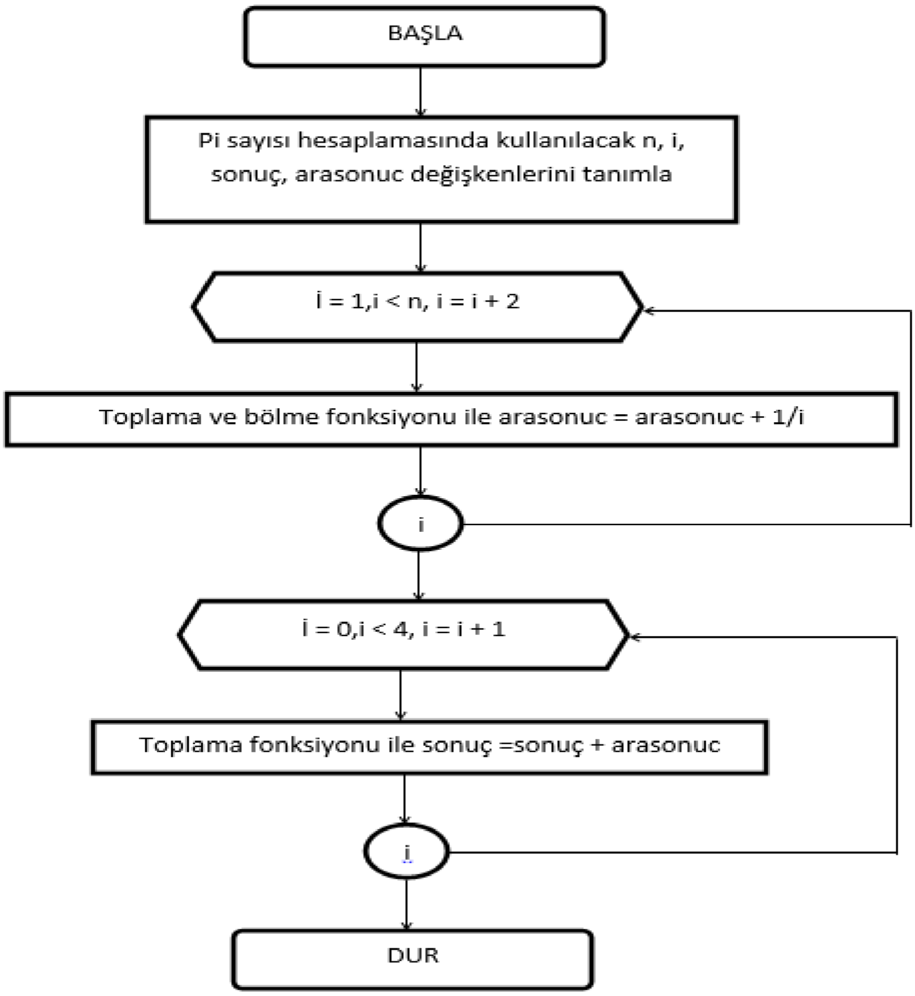
Adım: toplam = rakamtoplam + toplam (string oalarak toplama işlemi yap)

Adım: Dur

Algoritma bittiğinde toplam değişkeni x ve y'nin toplam sonucunu tutuyor olacaktır. Bölme algoritması da toplam algoritmasına benzer şekilde hesaplanabilmektedir.

Program akış çizgeleri sistem akış çizgelerine göre daha ayrıntılıdır ve he bir adımın ayrıntılı analizini içermektedir. mantık akış çizgesi olarak da adlandırılır (Sezen vd., 2016: 53). Şekil 1'de pi sayısının hesaplanması için akış şeması gösterilmektedir.

Şekil 1. Pi Sayısının Hesaplanması İçin Akış Şeması



Ki-Kare Testi :

KikareTablo : 9999,5

H1 : 9,41536192814045

H0 reddedilemiyor. Sayılar rassal

RUN Testi :

Z (0.01) : 10367,1101940181

H1 : -0,173941035429934

H0 reddedilemiyor. Sayılar rassal

Poker Testi :

X (0.01) :10367,1101940181

H1 : 1,70343521825397

H0 reddedilemiyor. Sayılar rassal

Kolmogorov & Smirnov Testi :

D (0.01) : 0,0163008150611301

H1 : 0,00689999999999991

H0 reddedilemiyor. Sayılar rassal

Gap Testi :

GAP tresti sonucuna göre h0 reddedilemiyor sayılar rassal

0 rakamı için : 0,0966999999999977

1 rakamı için : 0,102499999999998

2 rakamı için : 0,101999999999998

3 rakamı için : 0,0973999999999977

4 rakamı için : 0,101099999999998

5 rakamı için : 0,104499999999998

6 rakamı için : 0,101999999999998

7 rakamı için : 0,0967999999999977

8 rakamı için : 0,0946999999999977

9 rakamı için : 0,101299999999998

GAP D(0,01) : 0,0163

Bütün rakamlar için değerler D(0,01) den büyük olduğundan sayılar rassaldır.

4. SONUÇ

Çalışmanın uygulama bölümünde irrasyonel bir sayı olarak kabul edilen Pi sayısının ondalık basamaklarının rasgeleliğinin araştırılması amacıyla C# programlama dilinde geliştirilen program ile uygulanan rassallık testlerine ilişkin sonuçlara yer verilmiştir. Pi sayısı hesaplandıktan sonra geliştirilen rassallık ile ilgili bilgisayar programları kullanılmıştır. Pi sayı dizisi virgülden sonraki 10000 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 imkansız 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., And Nelson B. L., (1996), *Discrete-Event System Simulation*, Prentice Hall Inch, Usa.

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

Benavoli A., L.Chisci , A.Farina (2009), "Fibonaccisequence, Goldensection, Kalman Fitler And Optimal Control", *Signal Processing*,Vol.89,1483-1488.

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,* C. 4, S. 1, 69-80.

Gültekin Ahmet Turan, Asyalı Musa Hakan (2007), "Pi Sayısının Monte-Carlo Methodu Ve Gregory/Leıbnız Formülüyle Hesaplanması", *Journal Of Yaşar University*, pp.1-8.

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

Hee S.K., Neggers J. (2008), "Fibonacci Mean And Golden Section Mean", *Computers & Mathematics with Applications*, 228.

Horzum Tuğba, (2016), "İrrasyonel Sayıların Öğretimi İçin Görsel Model Önerisi: e ve π sayıları", *Bilecik Şeyh Edebali Üniversitesi Sosyal Bilimler Enstitüsü Dergisi,* Sayı / Volume 1 – s / p. 42-57

Kabirian A., Olafsson S. (2009), "Simulation Optimization With Hybrid Golden Region Search", *Winter Simulation Conference*.

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

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

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

Öztürk Fikri, Özbek Levent (2004), *Matematiksel Modelleme Ve Benzetim*, Gazi Kitabevi, Ankara.

Pidd, Michael (2004), *Computer Simulation In Management Science*, John Wiley & Sons, Ltd, Usa.

Sena S.K., Agarwal Ravi P., Shaykhian Gholam Ali (2007), "Golden Ratio Versus Pi As Random Sequence Sources For Monte Carlo Integration", *Mathematical and Computer Modelling*,vol 48,.161-178.

Sezen H.K., Sert Eteman F., Şenaras Eren A., Arıkan Kargı V.S., (2016), *Yöneylem Araştırmasına Giriş,* Dora Yayınevi, Bursa.

Şenaras Eren Arzu, İnanç Şahin, Sezen Hayrettin Kemal, (2015), "C# Programlama Dilinde Geliştirilen Program İle Rasgeleliğin Sınanması", *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, Cilt/Vol. XXXIV, Sayı/No. 1, 2015, pp. 27-46.

Şenaras Eren Arzu, İnanç Şahin, (2016), "Testing Randomness Of Generated Numbers Based On The Henon Map", *International Peer-Reviewed Journal Of Humanities And Academic Scinence*, Sayı. 17, pp. 17-34.

Şenaras Eren Arzu, İnanç Şahin, Sezen Hayrettin Kemal (2014), "Analyzing Randomness Of Numbers Generated By Logistic Map", *Uluslararası Hakemli Ekonomi Yönetimi Araştırmaları Dergisi*, Sayı:1, Cilt:1, pp. 23-37.