Turkish Journal of Science & Technology
15(1), 13-18, 2020

A Statistical Randomness Generation Algorithm Based on Nonlinear Behavior of Discrete Time Chaotic Systems

Erkan TANYILDIZI[1], Fatih ÖZKAYNAK[2*]
[1,2] Department of Software Engineering, Technology Faculty, Fırat University, Elazığ, Turkey
[1] etanyildizi@firat.edu.tr, [2]ozkaynak@firat.edu.tr

(Geliş/Received: 10/11/2019; Kabul/Accepted: 19/02/2020)

Abstract: Statistical randomness is a critical requirement for many applications. Generally, it is common to use a generator algorithm for statistical randomness. In this study, a generator algorithm proposed benefiting from chaotic systems. This proposed approach is based on chaotic maps with a simpler mathematical model compared to other chaotic system classes. So the generator has high practical applicability. In addition, optimization algorithms to guarantee statistical properties of generator.

Key words: Randomness, chaos, optimization.

Ayrık Zamanlı Kaotik Sistemlerin Doğrusal Olmayan Davranışını Temel Alan İstatistiksel Rasgelelik Üreteç Algoritması

Öz: İstatistiksel rasgelelik birçok uygulama için kritik bir gereksinimdir. Genellikle istatistiksel rasgelelik için bir üreteç algoritması kullanılması yaygın bir yaklaşımdır. Bu çalışmada kaotik sistemlerden yararlanılarak bir üreteç algoritması önerilmiştir. Önerilen bu yaklaşım diğer kaotik sistem sınıflarına göre daha basit matematiksel model sahip kaotik haritaları temel almaktadır. Bu yüzden üretecin pratik uygulanabilirliğinin yüksektir. Ek olarak optimizasyon algoritmaları üretecin istatistiksel özelliklerini garanti etmektedir.

Anahtar kelimeler: Rasgelelik, kaos, optimizasyon.

1. Introduction

Statistical randomness based on chaos theory is a hot topic [1-3]. A raw literature review shows that over the last decade, the number of studies related to chaos-based random number generators has been over 15,000. At this point, a question comes to mind. Have chaotic systems been used with the most appropriate approach? The aim of this study is to search for answers to this question and to determine the chaotic system parameters which can produce the deterministic random numbers required by various applications with the using of optimization algorithms.

The rest of the study has been organized as follows. In the second section, the general characteristics of chaotic systems are shortly explained. In the third section, randomness and deterministic random number generators are briefly introduced. In the fourth section, the details of the proposed method which show how to select of the chaotic system parameters to be used to provide the best randomness requirements by using optimization algorithms well-known in the literature are presented. The obtained results are discussed and the study is summarized in the last section.

2. Chaotic Systems

The relationship between the causes and consequences of real-world events is very complex. Chaotic systems is precisely based on this complex relationship. Irregular and unpredictable behavior of nonlinear systems is called chaos [1]. A small change in the initial conditions and control parameters of chaotic systems leads to very different outputs. This phenomenon is known as the butterfly effect and indicates that a very small change in a butterfly's flapping can trigger a hurricane. It is stated that the estimation of outputs in chaotic systems is impossible. In other words, a chaotic system is similar to a probabilistic system. But the source of the disorder is not the unpredictable external influences but the real dynamics of the system.

There are various classes of chaotic systems such as discrete-time, continuous-time, time-delayed, spatial and hyper-chaotic. These chaotic system classes are ranked from simple to complex considering the mathematical

* Corresponding author: ozkaynak@firat.edu.tr. ORCID Number of authors: [1] 0000-0003-2973-9389, [2] 0000-0003-1292-8490

models they possess. This study has been based on discrete-time chaotic systems. Because these systems have a simpler structure than others. The most important reason for this preference is to ensure effectiveness by keeping the complexity of the deterministic-random number generator as low as possible. Two different discrete-time chaotic systems have been used in the study. The mathematical models of these chaotic systems are given in Table 1 [1].

Table 1. Properties of some discrete time chaotic systems

Chaotic Map	Mathematical Model	Initial Condition	Control Parameters
Logistic Map	$x_{n+1} = a * x_n * (1 - x_n)$	$x_n \in [0,1]$	$a \in [3.5, 4]$
Circle Map	$x_{n+1} = x_n + a - \frac{b}{2\pi}\sin(2\pi x_n)$ mod1	$x_n \in [0,1]$	a $\in [0, 1]$, $b \in [0, 4\pi]$

3. Random Number Generators

Randomness is a fundamental characteristic required in many applications such as statistics, game theory, simulation, numerical analysis, entertainment and cryptology. Real world Random Number Generators (RNG) are divided into two basic classes. These classes are Deterministic RNG (also known as pseudo random number generators) and True RNG [3, 4].

Figure 1 shows the general design architecture of the Deterministic RNG (DRNG). In this general design architecture, $r_1, r_2, \dots, r_n \in R$ shows random numbers, and $s_n \in S$, shows the internal state. Here the S and R finite sets are called the state space and output space of the DRNG, respectively. The $\psi: S \to R$ output function generates the r_n random number from the current s_n internal state. The s_n status is then updated to $s_{n+1} := \varphi(s_n)$ using the φ state transition function. The initial internal state value s_1 is updated using the seed value s_0 in $s_1 := \varphi(s_0)$ format or by using more complex designs. It is clear from s_0 seed value that all $s_1, s_2, \dots, s_n$ inner states and $r_1, r_2, \dots, r_n$ random numbers can be estimated. Therefore, the seed value must be selected randomly in safety-priority applications and the state transition function and the output function must be sufficiently complex. The disadvantage of DRNG is that the output values can be fully determined by the seed value and that future random numbers are only dependent on the current internal state. Therefore, the internal state must be protected even if the device is not activated. The advantages of DRNG are that they are cheaper since they do not need any dedicated hardware [3, 4].

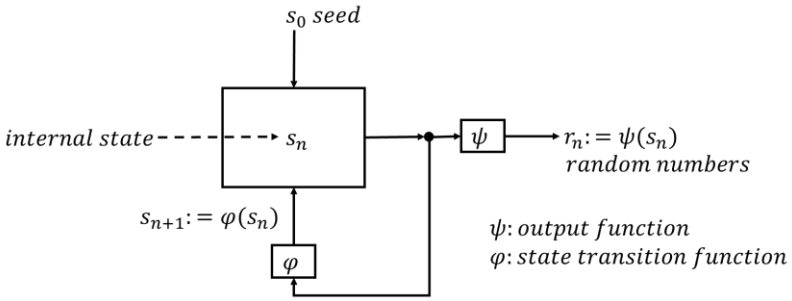

Figure 1. General design architecture of DRNG

One of the important sources for design and analysis of DRNG is Knuth's classic work called "The Art of Programming" [5]. Since the publication of this study, many RNG algorithms have been proposed and many researchers have continued to work on this hot topic. Ripley addressed the problems faced by personal computer users in order to produce random numbers in his study published in 1983 [6]. Ripley was developed effective methods of generating exponential, normal and Poisson distribution arrays. In 1990, L'Ecuyer solved the problem of generating uniform random variables for a user with a moderate computer use knowledge [7]. In a 1990 study by James, pseudo-random number generators for Monte Carlo calculations were addressed [8]. In another study conducted in 1990, Lagarias identified pseudo-random number generators based on number theory, one way functions and secret-key encryption systems [9]. Lagarias also summarizes the results of these generators on cryptanalysis. A comprehensive evaluation study on random number generators was performed by Ritter [10].

When the current literature is examined, it is seen that another important topic used in RNG design is chaos based randomness [11-18]. The close similarities between the two disciplines led the researchers to focus their attention on this subject. Because the complex relationship between the causes and consequences of real world events is one of the basic requirements expected from a RNG and is based on this complex relationship in chaotic systems.

4. Proposed Method

In the literature, a study was carried out to determine the optimal starting conditions and control parameters for the Logistic map using the brute force approach [19]. The pseudo-code of the algorithm is given in Figure 2. In the study proposed by Ozkaynak, it was very difficult to determine the most suitable values in infinite space and only an analysis study has been carried out using the 3 values after the comma. The complexity of the algorithm is $O(n^3)$, as can be easily seen from the pseudo-code. In other words, the problem is computationally difficult. Therefore, optimization algorithms are needed to find an approximate solution.

```
BitSequence[1000000]
for (a in 0, 1, 0.001)
    for (Xn in 3.5, 4, 0.001)
        for (i in 1, 1000000, 1)
            Xn+1=a*Xn*(1-Xn)
            if(Xn+1>0.5)
                BitSequence[i]=0
            else
                BitSequence(i)=1
            Xn=Xn+1
        end
    end
end
```

Figure 2. The pseudo-code of the brute force algorithm in Ref [19]

The optimization process tries to obtain the best solution among all solutions under the given conditions when a problem is solved. The variables that affect the performance of optimization and have values under our control are called decision variables. The objective function is created by analytically demonstrating the effects of decision variables on the objective. In most cases, only certain values of decision variables should be used. These limitations on the values of decision variables are called constraints. In other words, the goal of optimization is to find the best combination of objective functions. providing all constraints given among all possible combinations of decision variables.

The unique aspect of the study is to realize RNG designs by determining the initial conditions and control parameters of chaotic systems using optimization algorithm. Objective function of optimization algorithm is randomness requirements. As a result of the literature review, it was determined that there was no study aiming at reaching this goal before. Although there are studies in the literature that use optimization algorithms for chaos control or chaotic parameter estimation, these studies are very different from the purpose of the proposed method. In the study, the initial conditions and control parameters of two different chaotic systems have been selected with seven well-known meta-heuristic optimization algorithms. The optimization algorithms [20. 21]:

- Optimization algorithms based on Biological: Differential Evolution (DE), Particle Swarm Optimization (PSO), Symbiosis Organisms Search (SOS) Algorithm,
- Optimization algorithms based on Physics and Chemistry: Gravitational Search Algorithm (GSA), Harmony Search Algorithm (HS),
- Optimization algorithms based on Mathematics: Golden Sine Algorithm II (GoldSA-II).

The parameters used in the comparison algorithms are given as follow:

- PSO: Inertia Weight Damping Ratio = 0.99, Inertia Weight = 1, Global Learning Coefficient = 2.0, Personal Learning Coefficient = 1.5.
- GSA: Rnorm = 2, Rpower = 1, Elitist Check = 1.

- HS: Harmony Memory Consideration Rate= 0.9, Fret Width Damp Ratio= 0.995, Fret Width = 0.02*(Upper Bound - Lower Bound), Pitch Adjustment Rate=0.1.
- DE: Upper Bound of Scaling Factor=0.8, Lower Bound of Scaling Factor =0.2, Crossover Probability=0.2.
- SOS: Benefit factor (BF): random number either 1 or 2.
- ACO: Sample Size:40, Intensification Factor (Selection Pressure)= 0.5, Deviation-Distance Ratio=1.
- GoldSA-II : Gold section constant=[-pi*rand, pi*rand], Golden ratio (τ)= 0.618033

Analysis has been realized for 1,000,000 bits have been produced by using two discrete chaotic map outputs. After the generation of random number sequences, properties of statistical randomness have been checked using two different tests. First statistical randomness test is known as the monobit (frequency) test [22]. Definition of this test is that "Monobit test measures whether the number of 0s and 1s produced by the generator are approximately the same as would be expected for a truly random sequence." Monobit test results are given in Table 2. Chaotic maps in Table 1 shows as F1 and F2 symbols, respectively.

Table 2. Monobit test results for 1,000,000 random bit

	PSO	**ACO**	**DE**	**GSA**	**HS**	**SOS**	**GoldSA-II**
F1	258	**68**	238	1.484	592	416	212
F2	**88**	132	624	14.854	9.220	15.604	670

Another statistical randomness test is the chi-square test. Definition of this test is that "A chi-square statistic compares these substring proportions to the ideal 1/2. The statistic is referred to a chi-squared distribution with the degrees of freedom equal to the number of substrings. The chi-squared distribution is used to compare the goodness-of-fit of the observed frequencies of a sample measure to the corresponding expected" [22, 23]. In this test, the produced 1,000,000 bit has been divided into 4-bits lengths to produce 250,000 decimal numbers ranging from 0 to 15. In this case, according to the chi-square test, the expected frequency values are 15,625. Observed values for seven different optimization algorithms are given in Table 3.

Table 3. Observed values for produce 250,000 decimal numbers ranging from 0 to 15

		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
PSO	**F1**	15576	15653	15521	15667	15613	15591	15674	15594	15724	15612	15584	15720	15631	15603	15599	15638
	F2	15710	15486	15621	15893	15540	15480	15900	15442	15414	16006	15362	15419	15856	15468	15610	15793
ACO	**F1**	15616	15566	15623	15747	15594	15674	15685	15605	15659	15584	15552	15637	15623	15667	15605	15563
	F2	15613	15352	15710	15790	15758	15469	15787	15442	15504	15768	15719	15639	15553	15788	15469	15639
DE	**F1**	15706	15656	15555	15657	15592	15671	15568	15619	15566	15637	15595	15693	15538	15682	15624	15641
	F2	15720	15540	15698	15679	15838	15466	15793	15537	15482	15727	15415	15523	15673	15643	15549	15717
GSA	**F1**	14983	15631	15952	15416	15862	15661	15659	15664	15720	15477	15617	15589	15685	15689	15859	15536
	F2	14647	17172	16044	15831	17128	14118	16436	15185	17396	15205	15216	15376	15148	16189	14961	13948
HS	**F1**	15615	15553	15607	15681	15524	15648	15730	15712	15541	15615	15678	15642	15615	15703	15578	15558
	F2	16483	16125	16536	15322	16163	15417	13897	16233	16105	15603	14937	14923	15587	14756	16453	15460
SOS	**F1**	15671	15606	15597	15592	15576	15628	15622	15618	15785	15665	15563	15643	15607	15624	15593	15610
	F2	17135	16118	16168	15578	16398	14913	15510	15085	16209	15254	14847	15484	15593	15312	15312	15084
GoldSA-II	**F1**	15671	15660	15624	15577	15554	15653	15623	15684	15643	15582	15561	15610	15565	15687	15664	15642
	F2	15707	15627	15699	15567	15649	15659	15850	15263	15391	15704	15639	15746	15632	15829	15469	15569

In the case study 1, random numbers have been generated between 0 and 15. Therefore, degree of freedom (DF) of chi-square test is 16. All confidence values are given in Table 4 for the degree of freedom 16.

Table 4. Confidence values for degree of freedom 16

	Confidence Values										
DF	**0.995**	**0.975**	**0.20**	**0.10**	**0.05**	**0.025**	**0.02**	**0.01**	**0.005**	**0.002**	**0.001**
16	5.142	6.908	20.465	23.542	26.296	28.845	29.633	32.000	34.267	37.146	39.252

In order to be able to say that the random numbers produced are statistically random, the calculated chi-square value should be smaller than the values in the Table 4. The calculated chi- square values for the seven different optimization algorithms are given in Table 5. The best, worst, average and standard deviation values are given in

the Table 5. The most appropriate chi square values are shown in bold font for the best and mean values, in the Table 5. Among the optimization algorithms, GoldSA-II algorithm has been found to have better results than other algorithms. However, it should not be overlooked that other algorithms meet the randomness requirements for many confidence values.

Table 5. The chi- square test results for 1.000.000 random bit

		PSO	ACO	DE	GSA	HS	SOS	GoldSA-II
F1	**Best**	2.7180	2.5686	2.5702	46.4393	3.9171	2.5446	**1.9651**
	Mean	3.9958	2.8057	3.5108	1.7950e+03	4.4168	3.7470	**2.6258**
	Worst	4.7456	3.1441	4.8901	6.4223e+03	4.7971	4.5102	**3.0024**
	Std	0.9422	0.2278	0.9242	2.6923e+03	0.3338	0.7433	0.4048
F2	**Best**	41.1837	19.9224	**14.9336**	1.0426e+03	528.8125	372.6726	22.0122
	Mean	251.3002	241.4392	205.3796	1.6048e+03	872.4748	482.8232	**139.4131**
	Worst	551.7382	507.1160	354.7210	2.4460e+03	1.5448e+03	664.1293	**334.8357**
	Std	188.6843	227.7812	150.2168	756.5389	419.9311	108.5574	122.3634

In the above analysis, all optimizers have been run independently 5 times in equal population and iterations and the results have been recorded. The number of populations for chaotic system functions is 10 and the maximum number of function evaluations is 1.000. Table 6 shows the determined initial conditions and control parameters using seven optimization algorithm. Numerical deterioration is an important problem for chaotic systems [24, 25]. In order not to be affected by this problem, the highest precision provided by the computer where the analyzes have been performed has been used. Therefore, in order to show the numerical values in the best way, Table 6 is given in two parts.

Table 6. Most Suitable System Parameters for for 1,000,000 random bit

	Logistic Map (F1)	
	Xn	**a**
PSO	0.583587285455663	4
ACO	0.747882339784262	4
DE	0.312955856922984	4
GSA	0.438200825697660	3.99647848943457
HS	0.603238510606358	4
SOS	0.269877194226175	4
GoldSA-II	0.805314780517318	4

	Circle Map (F4)		
	Xn	**a**	**b**
PSO	0.765111672245320	1	11.502628728405833
ACO	0.717488331494669	0.999999826893032	11.502276826049323
DE	0.251024684144349	1	11.5022593721011
GSA	0.589147901804019	0.707607013522381	10.313736784726066
HS	0.492349236734390	0.978890082799524	11.718949046051932
SOS	0.181617966853290	0.732578163414611	10.1026142685088
GoldSA-II	0.100001102537958	1	11.5022370468944

5. Conclusion and Discussion

The purpose of this study is to determine the optimal initial conditions and control parameters of the chaotic systems to be used in the chaos based DRNG designs. Optimization algorithms have been used to achieve this purpose. The optimal initial conditions and control parameters for discrete-time chaotic systems have been obtained. All these results draw attention to the problems of doing various studies on the theoretical similarities between chaos and randomness. Therefore, in chaos-based RNG designs, the initial conditions and control parameters of the chaotic system should be selected in the most appropriate way. RNG design architecture and application specific randomness requirements must be taken into account when making this selection. In this study, a method has been proposed to make these choices in the most appropriate way. The parameters used in the analysis are given. Researchers who wish to work in this area can adapt the proposed method according to their

requirements. Therefore, after successful implementation of the proposed method, the outcomes are expected to contribute significantly to the literature of chaos-based randomness.

References

[1] Sprott J. Elegant Chaos Algebraically Simple Chaotic Flows. World Scientific, 2010.

[2] Özkaynak F. Brief review on application of nonlinear dynamics in image encryption. Nonlinear Dynam 2018; 92(2): 305-313. https://doi.org/10.1007/s11071-018-4056-x

[3] Stipčević M, Koç ÇK. True Random Number Generators. In: Koç ÇK (eds) Open Problems in Mathematics and Computational Science. Cham: Springer, 2014.

[4] Schindler W. Random Number Generators for Cryptographic Applications. Koç ÇK (ed.): Cryptographic Engineering. Signals and Communication Theory, Berlin: Springer, 2009.

[5] Knuth D. The Art of Computer Programming, Vol. 2, Seminumerical Algorithms. 2nd ed. Reading, Massachusetts: Addison-Wesley,1981.

[6] Ripley B. Computer Generation of Random Variables: A Tutorial. Int Stat Rev 1983; 51: 301-319.

[7] L'Ecuyer P. Random Numbers for Simulation. Commun ACM 1990; 33(1): 85-97.

[8] James F. A review of pseudorandom number generators. Compu Phys Commun 1990. 60: 329-344.

[9] Lagarias JC. Pseudorandom Number Generators in Cryptography and Number Theory. Proc Symp Appl Math 1990; 42: 115–143.

[10] Ritter T. The Efficient Generation of Cryptographic Confusion Sequences. Cryptologia 1991; 15(2): 81-139.

[11] Aydın Y, Özkaynak F. A Provable Secure Image Encryption Schema Based on Fractional Order Chaotic Systems. The 23rd International Conference ELECTRONICS 2019; 17-19 June 2019; Palanga, Lithuania.

[12] Tapiero CS, Vallois P. Randomness and fractional stable distributions. Physica A: Statistical Mechanics and its Applications 2018; 511: 54-60

[13] Bürhan Y, Artuğer F, Özkaynak F. A Novel Hybrid Image Encryption Algorithm Based on Data Compression and Chaotic Key Planning Algorithms. IEEE 7th International Symposium on Digital Forensic and Security; June 10-12 2019; Barcelos, Portugal.

[14] Dastgheib MA & Farhang M. A digital pseudo-random number generator based on sawtooth chaotic map with a guaranteed enhanced period. Nonlinear Dynam 2017; 89: 2957-2966 https://doi.org/10.1007/s11071-017-3638-3

[15] Murillo-Escobar MA, Cruz-Hernández C, Cardoza-Avendaño L, et al. A novel pseudorandom number generator based on pseudorandomly enhanced logistic map. Nonlinear Dynam 2017 87: 407-425. https://doi.org/10.1007/s11071-016-3051-3

[16] Lv X, Liao X & Yang B. A novel pseudo-random number generator from coupled map lattice with time-varying delay. Nonlinear Dynam 2018; 94: 325-341. https://doi.org/10.1007/s11071-018-4361-4

[17] Özkaynak F. Cryptographically secure random number generator with chaotic additional input. Nonlinear Dynam 2014; 78: 2015-2020. https://doi.org/10.1007/s11071-014-1591-y

[18] Sahari ML & Boukemara I. A pseudo-random numbers generator based on a novel 3D chaotic map with an application to color image encryption. Nonlinear Dynam 2018; 94: 723-744. https://doi.org/10.1007/s11071-018-4390-z

[19] Özkaynak F. The Effects on Performance of Using Chaotic Systems in Entropy Source of Deterministic Random Number Generators. 11th CHAOS 2018 International Conference; 5 - 8 June 2018; Sapienza University of Rome, Italy. pp.415-420

[20] Özer AB. CIDE: Chaotically Initialized Differential Evolution. Expert Syst Appl 2010; 37(6): 4632-4641

[21] Tanyıldızı E, Demir G. Golden Sine Algorithm: A Novel Math-Inspired Algorithm. Advances in Electrical and Comput Eng 2017; 17(2): 71-78.

[22] Rukhin A, Soto J, Nechvatal J, Smid M, Barker E, Leigh S, Levenson M, Vangel M, Banks D, Heckert A, Dray J, Vo S. A statistical test suite for random and pseudorandom number generators for cryptographic applications. NIST Special Publication 800–22rev1a, 2010.

[23] Masoom MA, Umbach D and Saleh AKMDE. Estimating Life Functions of Chi Distribution Using Selected Order Statistics. IIE Transactions 1992; 24(5): 88-98.

[24] Özkaynak F. A novel method to improve the performance of chaos based evolutionary algorithms. Optik 2015; 126(24): 5434-5438. https://doi.org/10.1016/j.ijleo.2015.09.098

[25] Persohn KJ, Povinelli RJ. Analyzing logistic map pseudorandom number gen- erators for periodicity induced by finite precision floating-point representation. Chaos Solit Fract 2012; 45: 238–245.