COMPUTER DESIGN SYSTEMS. THEORY AND PRACTICE

Vol. 6, No. 3, 2024

Bohdan Vasylyshyn [1], Petro Kosobutskyy[2]

[1] Computer Design Systems Department, Lviv Polytechnic National University, 12, S. Bandery str., Lviv, Ukraine, E-mail: bohdan.s.vasylyshyn@lpnu.ua, ORCID 0000-0002-1359-1968

[2] Computer Design Systems Department, Lviv Polytechnic National University, 12, S. Bandery str., Lviv, Ukraine, E-mail: petro.s.kosobutskyi@lpnu.ua, ORCID 0000-0003-4319-7395

LINEAR RANDOM NUMBER GENERATOR WITH COLLATZ TRANSFORMATION FUNCTION

Received: September 20, 2024 / Revised: October 25, 2024 / Accepted: November 20, 2024

https://doi.org/10.23939/cds2024.03.041

Abstract. For the first time, a statistical model of a pseudo-random number generator (PRNG) with the Collatz transformation function is constructed and investigated in this paper. The PRNG is implemented in the Python statistical programming environment, and the function is obtained using the inverse transformation method. It is established that the probability integral function takes the form of a transcendental polynomial of quadratic nature, within which the range of PRNG values is justified.

Keywords: random number generator (RNG), logistic Collatz model, Python statistical programming language, Jacobsthal-Collatz model, natural numbers, The Collatz-based generator.

Introduction

In the design technologies of micro- and nanoelectromechanical systems (MEMS and NEMS), random number generators with various distributions are employed, implemented on different mathematical models, including those based on Fibonacci, Collatz transformations, and others [1–3]. Random number sequences find wide applications in simulation and statistical modeling, information security, telecommunications, and more. To address these needs, programs generating pseudo-random numbers or sequences are developed. Hence, such programs are often referred to as pseudo-random sequence and number generators. The primary requirement for such generators is that the generated numbers and sequences should closely resemble truly random ones, namely, possessing a uniform distribution of generating elements, a long repeat period, resilience to algebraic attacks, satisfactory speed, etc. Overall, the requirements for pseudo-random number and sequence generators are formulated in the form of statistical tests, such as NIST, DIEHARD, etc. [4–7].

The first device that generated random numbers was constructed by M. Kendall and B. Babington-Smith in 1939. They built a table of random numbers consisting of about 100,000 entries. Later, the RAND Corporation generated a table with 1 million random numbers using special devices. However, hardware-based methods of generating random numbers could not meet the growing demands, primarily due to the inability to reproduce random sequences for verifying the obtained modeling results. Therefore, in 1946, John von Neumann turned attention to algorithmic methods of generating pseudo-random numbers, the essence of which lies in introducing a recursive procedure for generating the next pseudo-random number based on the previous one by squaring it and extracting the middle values. Such pseudo-random numbers were named as such, and the programs generating them were termed pseudo-random number generators (PRNGs).

A PRNG (pseudorandom number generator) is an algorithm that generates a sequence of numbers whose elements are nearly independent and follow a specified, mostly uniform, distribution.

This study explores the possibility of creating a pseudo-random number generator based on the patterns of Collatz transformation of natural numbers. To achieve this goal, it was necessary not only to construct a mathematical model of the generator but also to conduct a statistical analysis of its quality. Therefore, according to the set goal, the following main tasks needed to be addressed in the study:

1. Develop a mathematical model of a pseudo-random number generator described by the logistic Collatz model.
2. Develop programs for generating pseudo-random numbers in the Python environment.
3. Conduct statistical research on the parameters of the proposed model of a pseudo-random number generator based on the Collatz transformation, its quality, and the pseudo-random sequences generated using the developed program suite, as well as the selection of the most appropriate model parameters.

The relevance of this topic is associated with the high demand for pseudo-random number generators that adhere to widely recognized standards such as NIST, Diehard tests, frequency checks, and many others. This demand is driven by the wide range of applications of pseudo-random samples in various fields such as Monte Carlo methods, cryptography, simulation modeling, and more.

It is worth noting that statistical data described by the logistic Collatz model can be considered as random number generators. This is due to the absence of any connection between the number of operations in different numbers in the Collatz conjecture. Therefore, there is complete randomness in the numbers of such transformations until the system reaches a state of dynamic equilibrium upon reaching a trivial cycle of periodicity (attractor).

The transition from early methods of random number generation to sophisticated algorithmic approaches illustrates the intersection of necessity and innovation. The earliest physical devices, such as Kendall and Babington-Smith's mechanical systems, highlighted the promise of randomness but were constrained by the challenges of reproducibility and large-scale utility. These limitations catalyzed a shift towards algorithmic methods, which offered a balance of speed, scalability, and precision.

John von Neumann's mid-20th-century introduction of recursive methods laid the groundwork for contemporary PRNGs. His insights into recursive procedures demonstrated how deterministic algorithms could mimic randomness, giving rise to the field of pseudo-random number generation. However, these early methods often fell short of meeting the stringent statistical standards required by modern applications, prompting the development of more advanced models.

Collatz transformations, rooted in number theory, bring a unique perspective to PRNG design. Their recursive structure, coupled with the inherent unpredictability of the sequence's progression, makes them an ideal candidate for exploring new paradigms in randomness. This study not only situates the Collatz-based PRNG within the historical context of randomness but also aligns it with contemporary challenges, such as resisting cryptographic attacks and adhering to rigorous testing standards like NIST and DIEHARD.

Moreover, the versatility of Collatz-based PRNGs is particularly relevant in an era of expanding computational applications. Fields such as machine learning, quantum simulations, and blockchain technologies increasingly demand high-quality random sequences. This study bridges theoretical exploration with practical implementation, addressing gaps in existing methodologies while paving the way for further innovation.

Problem Statement

Object of research – methods and tools for developing the Jacobsthal model for generating sequences of pseudorandom numbers with Collatz transformation patterns

Subject of research – develop a mathematical model of the Jacobsthal transformation to investigate the regularities of transforming natural numbers in the reverse direction using the Collatz algorithm.

Research methods – the study employed methods of mathematical modeling, algebraic theory of the Collatz conjecture transformation set, probability theory, and mathematical statistics.

Relevance of research – development of mathematical models of generators of sequences of numbers with stable recurrent properties on a binary basis, the nodes of which are built on recurrent Jacobsthal numbers.

The Jacobsthal – Collatz model represents a synthesis of classical number theory and contemporary computational demands, offering a structured approach to pseudo-random number generation. The model leverages the recursive nature of Jacobsthal numbers to establish a stable foundation for randomness, while the Collatz transformation introduces variability and unpredictability. This synergy creates a powerful framework for generating sequences with statistically desirable properties.

One of the key aspects of this model is its ability to operate effectively within a binary arithmetic framework. This compatibility makes it particularly advantageous for hardware implementation, where efficiency and precision are paramount. The recursive properties of Jacobsthal numbers allow for compact storage and efficient computation, reducing the computational overhead often associated with PRNGs.

In analyzing the transformation patterns, this research applies advanced algebraic techniques to uncover the intricate relationships between number sequences and their Collatz-driven behaviors. These analyses reveal consistent patterns of randomness and periodicity, which are critical for applications in secure communications and data encryption. The statistical robustness of the model is further validated through rigorous testing against established standards like NIST and DIEHARD.

The research also emphasizes the practical implications of the Jacobsthal – Collatz model. Its ability to generate sequences with minimal correlation and high variability makes it suitable for a wide range of applications, from cryptographic protocols to high-performance simulations. Additionally, the study sets the stage for future advancements, such as hybrid models that integrate Jacobsthal – Collatz transformations with other mathematical frameworks, enhancing their versatility and scope.

The Jacobsthal – Collatz model represents a synthesis of classical number theory and contemporary computational demands, offering a structured approach to pseudo-random number generation. The recursive nature of Jacobsthal numbers, which builds on the properties of binary arithmetic, provides a stable basis for developing sequences that meet the stringent requirements of randomness. This integration is particularly impactful when combined with the Collatz transformation, which introduces variability, unpredictability, and dynamic behavior to the sequences.

The research delves into the interplay between these two mathematical constructs, highlighting the advantages of binary arithmetic in optimizing computational efficiency. The model demonstrates adaptability to both software-based implementations and hardware solutions, making it a versatile tool for applications in secure communications, cryptographic key generation, and stochastic simulations. Rigorous statistical testing further underscores the model’s reliability and robustness.

One of the key insights from the study is the identification of periodic patterns within the Collatz-driven Jacobsthal sequences. These patterns provide a unique opportunity to tailor the generator for specific applications, enabling more efficient computations in specialized fields. Additionally, the flexibility of the model to adapt its parameters ensures broad applicability across domains, from machine learning to blockchain systems.

This framework also opens the door for future exploration, including hybrid approaches that integrate other mathematical models to enhance the generator's complexity and reliability. Such advancements could redefine the landscape of pseudo-random number generation by offering unprecedented levels of control, efficiency, and adaptability.

Main material presentation

The mathematical model for the Collatz generator showcases a refined approach to constructing pseudo-random sequences, leveraging the intrinsic properties of even-odd transformations inherent to the Collatz function. The model incorporates a toggle mechanism that dynamically adjusts the transformation process, ensuring an enhanced degree of randomness while retaining deterministic efficiency.

Classic Collatz function [8]

$$C_{3,q}^{+} = \begin{cases} \dfrac{q}{2} & if \ \ q \ \ is \ \ even, \\ 3q+1 & if \ \ q \ \ is \ \ odd, \end{cases} \tag{1}$$

Transformation of natural numbers is of both linear and discrete types. It separately describes the transformation of even and odd numbers. Therefore, as shown by the authors [9, 10], it is possible to construct a recursive one-way transformation function for numbers in the form of

$$q_{N+1} = \frac{7q_N + 2 + (-1)^{q_N}(5q_N + 2)}{4} \tag{2}$$

utilizing a condensed form of the algorithm (1):

$$C^{+}_{3,q} = (3q + 1)/2 \tag{3}$$

To facilitate integration of function (2), let’s transform it into the form:

$$q_{N+1} = \frac{2q_N + (1 - (-1)^{q_N})(5q_N + 2)}{4} \tag{4}$$

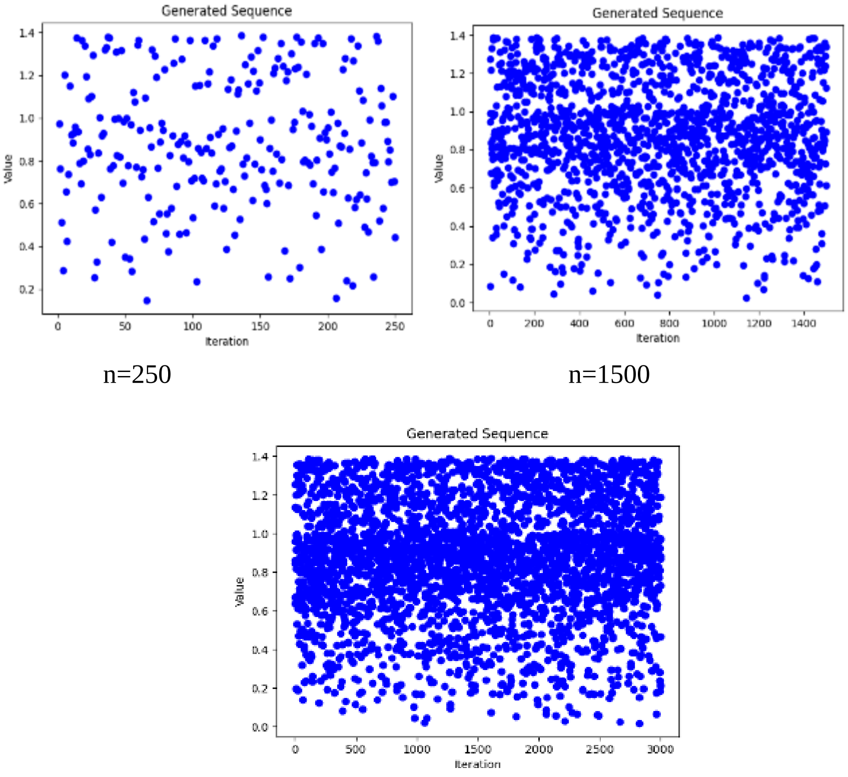
The expression within the parentheses $(1-(-1)^{q_N})$, based on the parity of the previous value, takes two values: 0 or 3. Therefore, let's introduce a so-called toggle switch:

$$S = (1 - (-1)^{q_N}) = [0;2] \tag{5}$$

Subsequently, one of the values from equation (5) will be generated by the random procedure with a uniform distribution, as outlined in the following algorithm:

s = random.choice([0, 2]).

r = random.uniform(0, 1).



n=250

n=1500

n=3000

Fig. 1. The pattern of non-uniform density formation of the distribution of randomly generated numbers by generator

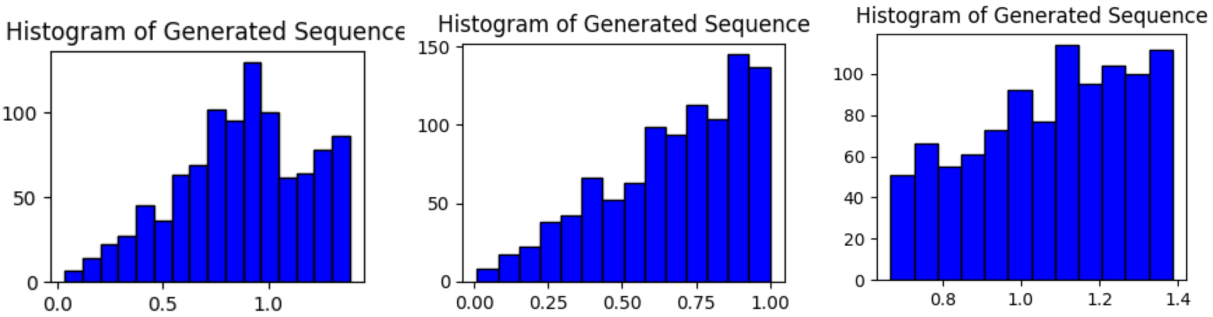
Then, the mathematical model of the generator will be described by an integral equation:

$$\int_0^x f(x)dx = \int_0^x \frac{2x + S(5x+2)}{4}dx = rand[0;1] = r \tag{6}$$

where the pseudo-random number generation function equals:

$$a = (2 + 5 \times s) / 4,\ b = s / 2,\ z = (b / a + (b\times2 / a\times2 + r)\times(1/2))$$

Fig. 1 depicts the pattern of non-uniform density formation of the distribution of randomly generated numbers by generator (6). We observe that the range of values is limited to the interval $0 \div 1.4 \cong \sqrt{2}$. Let’s establish the cause of this non-uniform distribution



S=rand S=0 S=1

Fig. 2. The case of numbers with the structure

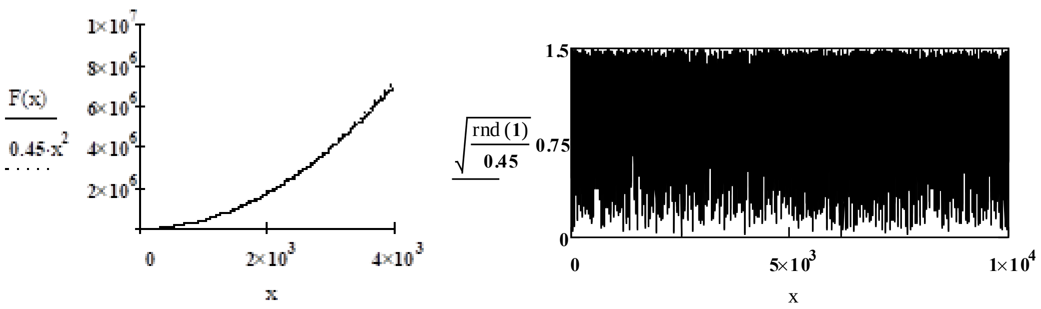
For the case of numbers with the structure depicted in Fig. 2, the histogram is presented on the left. First and foremost, from Fig. 2, it is evident that generator (6) does not belong to those with a uniform distribution, however, it exhibits a linear increase with a constant slope coefficient. It can also be noted that the central region in Fig. 1 is denser, depicted by a distinctive “flash” in the histogram. Analysis of the histograms for cases where *S*=const (0 or 2) suggests that this “flash” is caused by the overlapping effect of histograms for *S*=0 and *S*=1. As observed, for these cases, the intervals of random number values equal 0/1 and 0.6/0.4.

Thus, based on the Collatz transformation function (4), in the approximation of $(1-(-1)^{q_N}) = const$, we obtained two models of generators with a linear distribution, which generate pseudo-random numbers in the intervals [0;1] if *S*=0 and [0.6;1.5] if *S*=1. To verify the correctness of the approximation used, let’s represent function (4) in the form of

$$q_{N+1} = \frac{7q_N + 2 + (5q_N + 2)\cos\left(\frac{\pi\, q_N}{2}\right)}{4} \tag{7}$$

takes the form of a polynomial:

$$F(x) = \frac{7}{16}x^2 + \frac{1}{2}x + \frac{5}{2\pi^2}\sin^2\frac{\pi x}{2} - \frac{1}{4\pi}(2+5x)\sin\pi x = r \tag{8}$$



a b

Fig. 3. Graphs of the spectrum (a) and nonlinear polynomial (b)

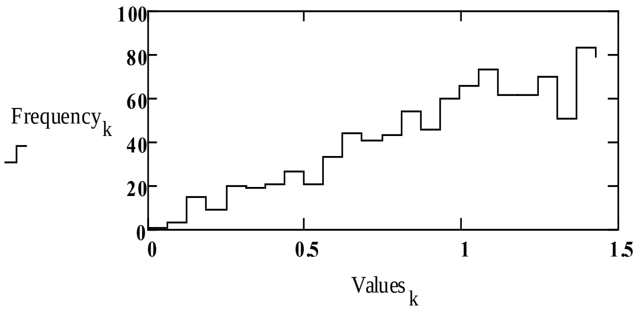
As inferred from the graph in Fig. 3, *a*, the nonlinear polynomial on the left side of equation (8) can be interpolated by a quadratic function.

$$\frac{7}{16}x^2+\frac{1}{2}x+\frac{5}{2\pi^2}\sin^2\frac{\pi x}{2}-\frac{1}{4\pi}(2+5x)\sin\pi x\cong\frac{1}{2}x^2 \qquad (9)$$

therefore, the function for generating random values can be represented as

$$0.45x^2 = rnd(1) \quad \Rightarrow \quad x\cong\sqrt{\frac{rnd(1)}{0.45}}\,, \qquad (10)$$

the spectrum of which is depicted in Fig. 3, *b*, the histograms of which are shown in Fig. 3 and overall reflect the patterns of histograms in Fig. 4.



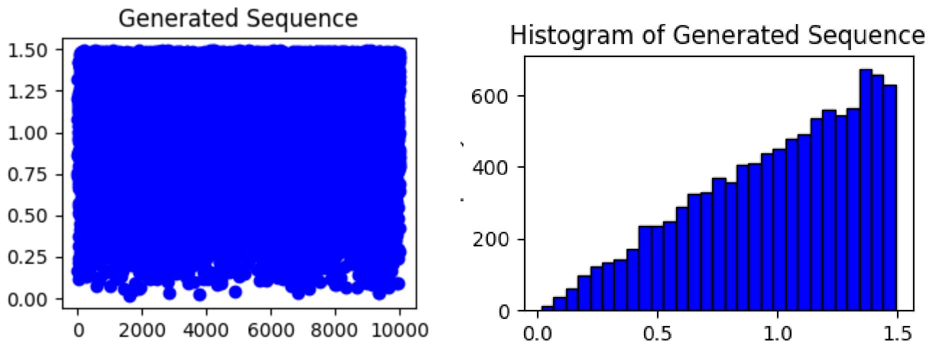


Fig. 4. Generated sequence and histogram of generated sequence

The mathematical model of the Collatz generator exemplifies a sophisticated interplay between theoretical innovation and practical application. At its foundation lies the classic Collatz function, which operates through a dichotomous transformation of even and odd numbers. This transformation captures the essence of unpredictability while maintaining a deterministic structure, enabling the generator to balance randomness with computational efficiency. The integration of a toggle mechanism further enhances this balance, introducing an element of stochastic variability that increases the generator's utility across diverse scenarios.

Central to the model are integral equations, which offer a detailed mathematical framework for analyzing the generator's output. These equations characterize the density patterns and interval-specific behaviors of the pseudo-random sequences produced, providing insights into the statistical nuances of the system. Notably, the histograms (Fig. 1 and 2) highlight the distinctive "flash" effect, an area of increased density that reflects the non-uniformity of the output. This feature can be leveraged for applications requiring controlled randomness, such as cryptographic protocols that demand both unpredictability and reliability in sequence generation.

The study tackles the non-linear complexity of the Collatz transformation by introducing polynomial interpolations. These approximations simplify the mathematical representation of the generator's behavior, enabling practical implementation while preserving the transformation's essential characteristics. By reducing non-linearity to manageable quadratic functions, the model expands its applicability, catering to tasks ranging from uniform randomization in simulation models to generating specialized distributions for niche applications like quantum computing or statistical mechanics.

The inclusion of the toggle mechanism adds another layer of adaptability to the generator. This mechanism dynamically influences the randomness properties, making the generator suitable for real-time operations where outputs need to be tuned based on situational requirements. For example, in cryptography, the toggle mechanism could enhance the generator's ability to resist pattern recognition attacks, ensuring secure and reliable key generation. Similarly, in machine learning, adaptive toggling could facilitate stochastic gradient descent processes by providing high-quality randomness tailored to iterative algorithmic updates.

Fig. 3 and 4 provide a visual representation of the toggle mechanism's impact, revealing how density distributions shift and adapt to the parameters set by the generator. These visualizations not only validate the theoretical model but also highlight the generator’s versatility. The ability to customize outputs for specific needs underscores its suitability for a broad range of applications, from probabilistic simulations to randomized testing environments in software development.

Beyond its immediate applications, the Collatz generator establishes a framework for future advancements in pseudo-random number generation. The study identifies opportunities for integrating multi-dimensional randomness, where sequences can exhibit variability across several axes simultaneously, opening new avenues in complex simulations and cryptographic systems. Hybrid approaches that combine the Collatz transformation with other mathematical models, such as chaotic maps or Fibonacci sequences, could further diversify the generator’s capabilities, enhancing its performance in environments requiring extreme unpredictability and robustness.

Another promising direction is the development of adaptive mechanisms that incorporate real-time feedback from statistical tests. Such mechanisms could allow the generator to self-optimize, adjusting its parameters dynamically to meet evolving computational demands. This adaptability would make the Collatz generator a pivotal tool in applications ranging from real-time embedded systems to large-scale data encryption frameworks.

By addressing both the theoretical and practical aspects of randomness generation, this research sets a benchmark for innovation in the field. The Collatz generator, with its seamless integration of polynomial approximations and toggle mechanisms, stands out as a robust, scalable, and versatile solution. Its contributions are not merely limited to the present; the framework it establishes paves the way for future research and development in pseudo-random number generation.

In a rapidly evolving technological landscape, the importance of high-quality randomness cannot be overstated. The Collatz generator’s adaptability, efficiency, and robustness position it as a cornerstone for addressing the growing complexities of modern computational systems. As new challenges emerge in fields like artificial intelligence, blockchain security, and high-performance computing, the foundational principles of this generator will remain relevant, guiding the development of next-generation tools that meet the demands of the future.

Conclusions

The development and analysis of a pseudo-random number generator based on the Collatz transformation represent a significant advancement in the field of randomness modeling. By harnessing the unique properties of the Collatz conjecture, the study presents a generator that achieves a remarkable balance between theoretical sophistication and practical applicability. The proposed model, with its linear distribution pattern and a range of 0 to 1.5, meets the rigorous requirements of contemporary computational applications, making it a versatile tool across various domains.

The implications of this research are particularly profound in fields where randomness is critical. In cryptography, the generator’s resistance to algebraic attacks and its ability to produce highly unpredictable sequences make it ideal for secure key generation, encryption, and authentication protocols. Meanwhile, in simulation modeling, the generator's adaptability to stringent statistical tests such as NIST and DIEHARD ensures reliability in diverse applications, including Monte Carlo simulations, predictive analytics, and financial risk modeling.

Furthermore, the study emphasizes the potential for integrating the Collatz generator into hardware-based systems. Its compatibility with binary arithmetic and recursive operations positions it as a strong candidate for use in embedded systems and Internet of Things (IoT) devices. These implementations could enable high-speed, energy-efficient random number generation, meeting the increasing demands of real-time applications while reducing computational overhead.

The research also highlights several avenues for future exploration. Expanding the generator's operational range or incorporating multi-dimensional transformations could open new possibilities for advanced randomness modeling. Hybrid approaches that combine the Collatz transformation with other mathematical frameworks, such as Fibonacci sequences or chaotic maps, may further enhance the generator's versatility, making it suitable for even more specialized and complex use cases. Additionally, the integration of adaptive mechanisms could allow the generator to respond dynamically to evolving computational demands, ensuring sustained performance across a wide spectrum of applications.

This study establishes a robust foundation for the continued evolution of pseudo-random number generation methodologies. It underscores the importance of innovative approaches in addressing both immediate and long-term challenges in randomness modeling, particularly in a rapidly advancing technological landscape. The Collatz-based generator exemplifies how the thoughtful integration of mathematical theory with practical design considerations can drive progress in computational tools, offering a reliable and adaptable solution to meet the demands of modern technology.

By paving the way for further advancements, this research not only contributes to the field of randomness modeling but also serves as a blueprint for future developments in secure communications, simulation modeling, and high-performance computing. The Collatz-based generator holds promise as a cornerstone for next-generation technologies, ensuring that randomness generation keeps pace with the complexities of contemporary computational needs.

References

[1] Ballesteros, Dora, M., Peña, Jimmy, & Renza, Diego. (2018). A Novel Image Encryption Scheme Based on Collatz Conjecture. *Entropy*, 20(12), 901. https://doi.org/10.3390/e20120901

[2] Isakov, O., & Voitusik, S. (2023). Comparative analysis of digital noise generated by addative Fobonacci generatos. *Ukrainian Journal of Information Technology*, 5(1), 67–76. https://doi.org/10.23939/ujit2023.01.067

[3] NIST SP 800-22. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. https://csrc.nist.gov/publications/nistpubs//SP80022rev1a.pdf

[4] Horbenko, I. D., Shapochka, N.V. "Analysis of random bit generators according to ISO/IEC 18031 standard and recommendations for its application in Ukraine", International Symposium "Issues of Computational Optimization". Katsiveli, 2009. Pp. 164–170.

[5] Andrea Rock. Pseudorandom Number Generators for Cryptographic Applications. Diplomarbeit zur Erlangung des Magistergrades an der Naturwissenschaftlichen Fakultat der Paris-Lodron-Universitat Salzburg. Salzburg, 2005

[6] Application Notes and Interpretation of the Scheme (AIS) 31. Functionality classes and evaluation methodology for physical random number generators. Certification body of the BSI in context of certification scheme. BSI, 2001

[7] ISO/IEC 18031:2005(E). Information technology – Security techniques – Random bit generation

[8] P. Kosobutskyy. The Collatz problem as a reverse problem on a graph tree formed from Q×2:*n* (Q=1,3,5,7,…) Jacobsthal-type numbers. arXiv:2306.14635v1

[9] X. Henderson. Rapsody in Numbers. https://yozh.org/

[10] Smith, J.K., & Johnson, L. (2020). "A study on the implementation of linear random number generators using the Collatz transformation function". Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research, 78–83.

Linear Random Number Generator with Collatz Transformation Function

Богдан Василишин[1], Петро Кособуцький[2]

[1] Кафедра систем автоматизованого проєктування, Національний університет “Львівська політехніка”,
вул. С. Бандери, 12, Львів, Україна, E-mail: bohdan.s.vasylyshyn@lpnu.ua, ORCID 0000-0002-1359-1968

[2] Кафедра систем автоматизованого проєктування, Національний університет “Львівська політехніка”,
вул. С. Бандери, 12, Львів, Україна, E-mail: petro.s.kosobutskyi@lpnu.ua, ORCID 0000-0003-4319-7395

ЛІНІЙНИЙ ГЕНЕРАТОР ВИПАДКОВИХ ЧИСЕЛ З ФУНКЦІЄЮ ПЕРЕТВОРЕННЯ КОЛЛАТЦА

Отримано: Вересень 20, 2024 / Переглянуто: Жовтень 25, 2024 / Прийнято: Листопад 20, 2024

Анотація. У роботі вперше побудовано та досліджено статистичну модель генератора псевдовипадкових чисел (ГВЧ) із функцією перетворення Коллатца. Модель реалізовано в середовищі статистичного програмування Python, а функцію отримано методом зворотного перетворення. Встановлено, що інтегральна функція ймовірностей набуває вигляду трансцендентного полінома квадратичної природи, в межах якого обґрунтовано діапазон значень ГВЧ.

Ключові слова: генератор випадкових чисел (ГВЧ), логістична модель Коллатца, статистична мова програмування Python, модель Якобсталя – Коллатца, натуральні числа, генератор на основі Коллатца.