УДК 004.94:519.21:004.421.2
DOI https://doi.org/10.32782/2663-5941/2025.4.2/42

Щербина Ю.В.
Національний університет «Одеська юридична академія»

Логінова Н.І.
Національний університет «Одеська юридична академія»

Манаков С.Ю.
Національний університет «Одеська юридична академія»

АНАЛІЗ СУЧАСНИХ СПОСОБІВ КОМП'ЮТЕРНОГО МОДЕЛЮВАННЯ СТОХАСТИЧНИХ ПРОЦЕСІВ

Стаття присвячена проблемі підвищення якості стохастичного комп'ютерного моделювання під час проєктування складних технічних систем, що забезпечують роботу транспорту, виробництва, управління інфраструктурними об'єктами та обмін даними в інформаційно-комунікаційних системах. Ефективність такого моделювання визначається наявністю адекватних математичних моделей, що формально описують поведінку реальних систем в стохастичних умовах їх експлуатації, та якістю джерел випадковості, що дозволяють відтворювати такі умови. Однією з головних проблем, що визначає якість стохастичного комп'ютерного моделювання, є вибір програмного способу генерації псевдовипадкових чисел з високим ступенем рівномірності розподілення їх ймовірностей. Ця проблема має вирішуватись у двох напрямах: по-перше, алгоритм формування псевдовипадкового числового потоку має бути невибагливим до таких обчислювальних ресурсів як пам'ять та процесорний час і, по-друге, такий потік не повинен відрізнятись від справжнього випадкового рівномірно розподіленого числового потоку. В роботі надано аналіз відомих програмних способів формування псевдовипадкових чисел, що використовуються для потреб комп'ютерного моделювання. Розглянуто ті з них, що вважаються найбільш економічними, а саме, генератор Фібоначчі з запізненням, генератори, побудовані на основі лінійних регістрів зсуву зі зворотним зв'язком та Xorshift генератор, в основу якого закладені економні, з точки зору комп'ютерної реалізації, логічні операції та операції зсуву. На основі проведеного аналізу наукових публікацій, було зроблено висновок про те, що з урахуванням сучасних вимог до якості комп'ютерного моделювання, найбільш ефективним джерелом випадковості слід використовувати саме Xorshift генератор. Що ж стосується генераторів Фібоначчі з запізненням та генераторів, побудованих на основі регістрів зсуву зі зворотним зв'язком, то їх можна використовувати тільки у якості складових частин більш складних алгоритмів формування псевдовипадкових чисел.

Ключові слова: *комп'ютерне моделювання, метод зворотних функцій, метод відбракування випадкових чисел, генератор псевдовипадкових чисел, статистичний аналіз.*

Постановка проблеми. Створення складних систем, що забезпечують роботу сучасного виробництва, транспортних комунікацій, технічне управління інфраструктурними об'єктами та оброблення і обмін даними в комп'ютерних мережах, зазвичай, з метою попередньої оцінки складності задачі і необхідних ресурсних витрат, вимагають попереднього натурного моделювання. В окремих випадках, такі натурні випробування, через надмірну необхідність матеріальних ресурсів, є неприйнятними і це є проблемою. Сьогодні, така проблема ефективно вирішується за рахунок використання комп'ютерного моделювання, що передбачає попереднє створення математичних моделей фізичних процесів та їх програмної реалізації.

Статистичні випробування, побудованих на основі математичних моделей, віртуальних стохастичних систем, дають змогу отримати попередню оцінку їх поведінки у різних умовах експлуатації. Суть такого моделювання зводиться до формального описання зв'язків, які пов'язують між собою зовнішні і внутрішні фактори, що впливають на поведінку складних систем.

У разі, якщо така поведінка не може напряму бути формально математично пов'язана з факторами, що впливають на процес, який підлягає дослідженню, виконують його симуляцію, тобто комп'ютерні статистичні випробування. Такий спосіб дослідження з використанням чисельних комп'ютерних методів передбачає декомпозицію загального складного процесу на простіші підпроцеси, що можуть бути описані елементарними статистичними видами розподілень або їх достатньо простими комбінаціями [1]. Позитивними сторонами такого моделювання є високий рівень деталізації, відсутність обмежень на параметри моделювання зовнішнього середовища і можливість дослідження динаміки зміни параметрів процесу у просторі та часі.

Комп'ютерне відтворення будь-якого випадкового процесу, перш за все, вимагає наявності "джерела випадковості", але комп'ютер, це дискретний детермінований автомат, який за визначенням не здатен генерувати випадкові величини. Тому, рішенням цієї другої проблеми комп'ютерного моделювання є генерація, так званих псевдовипадкових чисел (ПВЧ). Програмні генератори такого типу уявляють собою обчислювальні алгоритми, які створюють, циклічно повторювані з періодом *T*, послідовності, що для стороннього спостерігача виглядають як випадкові. Програмні алгоритми формування таких чисел в науковій літературі називають генераторами псевдовипадкових чисел (Pseudo Random Number Generators – PRNG).

Для якісного комп'ютерного моделювання необхідно, щоб довжина циклу повторення псевдовипадкових чисел була якомога більшою, а розподілення їх ймовірностей у складі вихідної послідовності була рівномірною [2]. Виконати обидві ці умови для якісного моделювання вкрай важко. На даний час існує велика кількість програмних варіантів генераторів ПВЧ, але вони або занадто складні і потребують великих обчислювальних ресурсів, або не забезпечують необхідного рівню рівномірності розподілення символів у складі вихідної послідовності і, через це, проблема побудови хороших економічних програмних генераторів такого типу залишається достатньо актуальною.

Аналіз останніх досліджень і публікацій. Історія створення програмних алгоритмів генерації псевдовипадкових чисел розпочинається практично одночасно з появою обчислювальної техніки. З тої пори цей напрям став добре вивченою областю математики. Першою значущою роботою у цій сфері була стаття Дж. фон Неймана "Різні методи, що пов'язані з отриманням випадкових чисел" [3], де він дав оцінку намаганням тодішнього покоління математиків створювати випадкові числа арифметичними методами і надав математичне обґрунтування, відомим на той час, способам створення комп'ютерних моделей із заданими елементарними законами розподілення ймовірностей. Основним висновком, який він зробив, є твердження про те, що випадковість створених алгоритмічними методами числових послідовностей ПВЧ, може бути оцінена лише методами статистичного аналізу.

Наступною фундаментальною роботою, де були обґрунтовані наукові математичні підходи до оцінки "випадковості" чисел, створених арифметичними методами, була робота Д. Кнута "Мистецтво програмування" [2]. У цій роботі він описав, відомі на той час, прості елементарні алгоритми генерації ПВЧ, обґрунтував якість їх вихідних послідовностей і дав рекомендації щодо використання основних критеріїв тестування.

Досягнення у розвитку обчислювальної техніки та постійне збільшення її обчислювальних ресурсів привели до збільшення популярності комп'ютерного моделювання серед фахівців, що працюють над підвищенням його ефективності. Їх зусилля були зосереджені на створені нових генераторів ПВЧ, які забезпечували б кращі показники рівномірності вихідного числового потоку і, в той же час, залишались не вибагливими до таких обчислювальних ресурсів як пам'ять і процесорний час [4].

Протягом останніх десятиріч з'явились такі методи генерації ПВЧ як "шторм Мерсена" (Mersenne twister) [5, 6, 7] та генератор Xorshift [8]. Обидва ці генератори проходять пакет тестів, визначений стандартом Національного інституту стандартів і технологій Сполучених Штатів Америки (National Institute of Standards and Technology – NIST) [9]. Сьогодні MT-генератор входить до складу бібліотек переважної більшості середовищ програмування.

Що стосується MT-генератора, то його автори ставили за мету забезпечення максимально можливого періоду повторення і це їм вдалось. Вони побудували генератор, період якого T дорівнює $2^{19937} - 1$, але це досягається за рахунок алгоритму, який містить вкрай витратні обчислювальні операції, що виконуються над w-бітними цілими числами над кінцевим двійковим полем [6].

Під час розроблення Xorshift-генератора його автор Джордж Марсалья ставив собі за мету, перш

за все, створити його економічним і швидкодіючим. Причина полягає в тому, що паралельно з розвитком і удосконаленням комп'ютерної техніки суттєво збільшується складність прикладного програмного забезпечення, і, через це, невибагливість будь-якого обчислювального алгоритму до необхідних обчислювальних ресурсів завжди буде залишатись актуальною задачею [10, 11].

Постановка завдання. Мета статті полягає у проведені аналізу існуючих алгоритмів формування рівномірно розподілених псевдовипадкових числових послідовностей, придатних для комп'ютерного моделювання стохастичних процесів з урахуванням ефективності використання ними обчислювальних ресурсів.

Виклад основного матеріалу. Генерація випадкових величин для потреб комп'ютерного моделювання передбачає забезпечення їх відповідності певному розподіленню ймовірностей, що описується функцією щільності розподілення ймовірностей $f(x)$ (Probability Density Function – PDF). Це може бути рівномірне розподілення, нормальне, пуассонівське або інше, в залежності від умов експерименту. Загальні процедури, що використовуються у моделюванні, описані роботі [12]. Повне математичне обґрунтування всіх стандартних статистичних розподілень та алгоритми їх симуляції можна знайти в роботі [13].

Формування числових розподілень, що можуть бути описані аналітичними виразами, вимагають наявність початкового вхідного рівномірно розподіленого на відрізку [0, 1] числового потоку $x_1, x_2, \ldots, x_n$. Для його отримання зазвичай використовується генератор ПВЧ. Найпростішим алгоритмом такого типу є найбільш популярний, запропонований Лемером [14], лінійний конгруентний генератор (ЛКГ). Він може бути описаний так

$$x_{n+1} = (ax_n + c) \bmod m,$$

де m – це модуль (ціле число, таке, що $m \geq 2$), a – множник (ціле число, таке, що $0 < a < m$), c – додаток (ціле число, таке, що $0 \leq c < m$).

Довжина періоду такого генератора T дорівнює величині m, лише тоді, коли числа c і m взаємно прості, різниця $b = a - 1$ ділиться на просте p, яке є дільником m і, нарешті, якщо b кратне 4, у разі, коли m кратне 4. У всіх інших випадках період T менший за m. Саме невеликий період повторення часто обмежує застосування цього генератор ПВЧ у моделюванні і, тому, там де вимоги до точності відтворення випадкового процесу високі, доводиться користуватись іншими алгоритмами.

У разі, коли хороший генератор псевдовипадкових рівномірно розподілених чисел обрано, для генерації процесу із заданим розподіленням $f(x)$ можна скористатись або методом відбракування чисел, що не задовольняють заданому типу розподілення або методом зворотних функцій.

Метод відбракування чисел дозволяє отримати реалізацію необхідного випадкового процесу $f(x)$ із чисел деякого іншого випадкового процесу $g(x)$, відкидаючи ті з них, що не вписуються у процес, який підлягає моделюванню. Він використовується коли пряма генерація випадкових чисел цільового розподілення не можлива.

У відповідності до цього методу, спочатку обирають генератор додаткового розподілення зі щільністю $g(x)$, вихідні значення якого, більше або дорівнюють розподіленню процесу зі щільністю $f(x)$.

Потім генерують послідовність псевдовипадкових чисел $x_1, x_2, \ldots, x_n$ додаткового розподілення, наприклад, рівномірно розподіленого на інтервалі $[a, b]$.

Далі, для кожного сформованого числа x_i з розподіленням $g(x)$, обчислюється значення функції $f(x_i)$. Якщо виявляється, що значення функції $g(x_i)$ більше ніж значення функції $f(x_i)$, воно відкидається.

Описана послідовність дій повторюється до тих пір поки не буде набрано достатню кількість необхідного статистичного матеріалу з розподіленням $f(x_i)$.

Такий спосіб моделювання дозволяє скористатись генератором з довільним розподіленням, не звертаючи уваги на відповідність сформованих ним чисел будь якому розподіленню $g(x_i)$. Єдиною вимогою залишається великий період повторення T. Його негативною стороною є велика витратність – у разі, коли основне й додаткове розподілення не близькі, кількість відбракованих величин занадто велика.

Метод моделювання випадкового процесу, оснований на використанні зворотних функцій, передбачає наявність аналітично визначеної функції розподілення ймовірностей $F(x)$ (Cumulative Distribution Function – CDF), що описує такий процес. Доказано, що коли безперервна випадкова величина x з довільним розподіленням має відповідну функцію розподілу ймовірностей $F(x)$, існує відповідна до неї, рівномірно розподілена на інтервалі [0, 1] випадкова безперервна величина $\alpha = F^{-1}(\alpha)$ [15].

Моделювання з використанням такого методу передбачає, перш за все, існування функції розподілення ймовірностей, що описує процес, який

має моделюватись. Після цього необхідно вибрати генератор псевдовипадкових чисел, що забезпечує високий рівень рівномірності розподілення чисел $P(\alpha)$ у вихідному потоці $\alpha_1, \alpha_2, \ldots, \alpha_n$ на інтервалі [0, 1]. Наприкінці, через зворотну функцію $P^{-1}(\alpha)$ визначаються величини $x_1, x_2, \ldots, x_n$ числового потоку з необхідним розподіленням $F(x)$.

Метод зворотних функцій забезпечує високу точність і простоту програмної реалізації, але вимагає наявності аналітичного описання процесу, що підлягає моделюванню і, головне, високоякісного, з точки зору рівномірності розподілення символів у вихідній послідовності. Саме він найчастіше використовується у більшості випадків.

Таким чином, однією з проблем комп'ютерного моделювання є вибір генератора ПВЧ, що має найбільшу ентропію [16], тобто ймовірність появи нульового символу у складі вихідної послідовності $P(x_i = 0)$ повинна максимально наближатись до ймовірності одиничного символу $P(x_i = 0)$. Розбіжність між цими величинами e називають зміщенням вихідного розподілення чисел і визначають як

$$e = \frac{1}{2}\left(P\left(x_i = 1\right) - P\left(x_i = 0\right)\right),$$

де величини $P\left(x_i = 1\right) = \frac{1}{2} + e$ і $P\left(x_i = 0\right) = \frac{1}{2} - e$, а x_i – це символи, що належать, послідовності $x_1, x_2, \ldots, x_i$. Саме цей показник використовується для оцінки придатності якості обраного для потреб моделювання генератора ПВЧ. Для високоякісного генератора ця величина має наближатись до величини 0.5, а величина e повинна бути мінімально можливою.

Історично, пошук придатних для комп'ютерного моделювання алгоритмів генерації ПВЧ мав на меті забезпечення простоти обчислень і високого ступеню рівномірності розподілення сформованих чисел та великої довжини періоду повторення T. Оскільки лінійних конгруентний метод не забезпечував виконання цих вимог, було запропоновано й інші типи генераторів. Найбільш відомим з них були генератори Фібоначчі з запізненням та лінійні генератори, побудовані на основі регістрів зсуву зі зворотним зв'язком (Linear Feedback Shift Register – LFSR).

Принцип роботи генераторів Фібоначчі з запізненням у різних його модифікаціях добре описані Д. Кнутом в роботі [2] і, пізніше, їх детальну оцінку було надано в роботі [17]. Принцип дії такого генератора описується виразом

$$X_n = \left(X_{n-24} + X_{n-55}\right) \bmod m, \quad n \geq 55,$$

де m – парне число, а $X_0, \ldots, X_{54}$ довільні цілі числа.

Тут вихідне число утворюється з двох попередніх чисел, затриманих на тактів. Числа 24 і 55 – це числа Фібоначчі, їх значення знайдено простим підбором. В роботі [2] наведено й інші пари таких чисел, що забезпечують прийнятний рівень рівномірності їх розподілення, що свого часу було підтверджено відповідними статистичними випробуваннями. Інші пари підібраних чисел Фібоначчі достатньо великі і це приводить до суттєвого збільшення необхідного об'єму пам'яті. Довжина періоду послідовності на виході генератора Фібоначчі складає $2^{q-1}(2^{55} - 1)$ де q – це розрядність мікропроцесора. В минулому сторіччі такі генератори ПВЧ вважались одними з найкращих, але сьогодні їх використовують лише у якості складових частин більш складних алгоритмів.

Генератори псевдовипадкових чисел, побудовані на основі регістрів зсуву зі зворотним зв'язком широко використовувались раніше в теорії кодування для формування так званих m послідовностей для потреб зв'язку. Саме через це їх математична теорія добре розроблена і описана у багатьох наукових джерела. Їх популярність визначалась тим, що вони дозволяли просту апаратну реалізацію.

Такі генератори уявляють собою послідовність бітів, кількість яких n, визначається довжиною регістра, де вони зберігаються.

Як показано на рисунку 1, кожного разу, після зсуву всіх бітів регістра праворуч на одну позицію, крайній зліва біт є функцією тих бітів, що об'єднуються функцію зворотного зв'язку, яка, у свою чергу, визначається ненульовими коефіцієнтами утворюючого характеристичного поліному $p(x)$ [18].

Для того, щоб LFSR генератор мав максимальний період, його характеристичний поліном повинен бути примітивним по модулю 2, тобто він не повинен розкладатись на простіші поліноми. Його максимальний період не перевищує величини $2^n - 1$.

Сьогодні такі генератори успішно реалізують програмними методами, але вони, так само, як і генератори Фібоначчі з запізненням, використовуються лише у якості складових частин інших алгоритмів формування ПВЧ.

Враховуючи той факт, що сьогодні темпи зростання складності програмного забезпечення суттєво перевищують темпи збільшення ресурсів

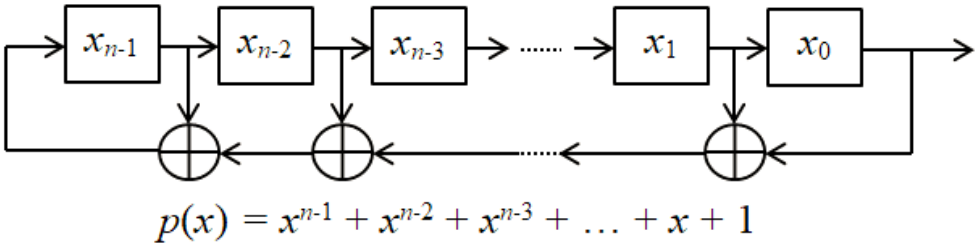

Рис. 1. LFSR регістр у конфігурації Фібоначчі

обчислювальної техніки, зусилля математиків, що займаються комп'ютерним моделюванням, зосереджені на пошуку економічних алгоритмів формування ПВЧ. Хорошим прикладом рішення такої задачі є запропонований Джорджем Марсальєю генератор, відомий під назвою Xorshift [6]. Цей генератор уявляє собою комбінацію декількох LFSR генераторів, що побудовані на основі характеристичних поліномів невисокого ступеню. Головним досягненням автора є те, що у складі запропонованого алгоритму використовуються лише такі економічні обчислювальні операції як зсув та складання по модулю два.

Алгоритм Xorshift заснований на використанні множини ненульових двійкових векторів 1×32 із множини Z, а лінійне перетворення f над множиною Z, описується невиродженою бінарною матрицею T розміром 32×32.

В роботі [19] було показано, що вихідна послідовність чисел такого генератора буде виглядати як $yT, yT^2, yT^3, \ldots$ лише тоді, коли порядок T дорівнює $2^{32} - 1$ у групі двійкових невироджених матриць розміром 32×32, а період вихідної послідовності дорівнює $2^{32} - 1$. Також, було показано, що максимально можливий період забезпечується тоді, коли алгоритм забезпечує такі типи зсувів як

$$y\hat{} = y \ll 13; y\hat{} = y \ll 17; y\hat{} = y < 5.$$

У разі виконання описаних умов, Xorshift алгоритм може бути реалізований мовою С так, як це показано на рисунку 2.

Тут x, y, z і w – це 32-бітні початкові числа, що визначають внутрішній стан генератора та вигляд послідовності, яка формується 32-бітними вихідними числами w. На кожному поточному кроці алгоритму нове вихідне число w утворюється як комбінація розрядів чисел x та поточного числа w. Спочатку число x перетворюється у число t шляхом виконання логічних операцій та операції зсуву, що підвищує "випадковість" його молодших байтів, а потім відбувається послідовний зсув чисел $y \rightarrow x$, $z \rightarrow y$, та $w \rightarrow z$. Наприкінці отримується нове значення числа w, як результату

```
x = 123456789
y = 362436069
z = 521288629
w = 88675123
t = x ^ ((x << 11) & 0xFFFFFFFF);
x = y;
y = z;
z = w;
w = (w ^ (w >> 19)) ^ (t ^ (t >> 8));
```

Рис. 2. Програмна реалізація алгоритму Xorshift

виконання кількох логічних операцій та операцій зсуву над числами t і поточного значення w.

Таким чином, завдяки використанню в алгоритмі Xorshift таких економічних операцій, як складання по модулю два та зсуву, фактично виконується нерегулярне перемішування бітів 32-бітних чисел і, як показує проведений статистичний аналіз [11, 20], він може бути хорошим кандидатом на використання у якості джерела випадковості для комп'ютерного моделювання.

Висновки. В проведеному дослідженні розглянуто алгоритми, які можуть бути використані як джерела випадковості при комп'ютерному моделюванні стохастичних процесів. Було показано, що у тих випадках, коли процес, який підлягає дослідженню, може бути описаний за допомогою аналітичного виразу, зазвичай, використовують метод зворотних функцій, що вимагає використання генератора псевдовипадкових чисел з жорсткими вимогами до рівномірності розподілення вихідних чисел. Другою суттєвою вимогою є ефективність такого генератора, яка визначається швидкодією його роботи та невибагливістю до необхідної пам'яті і процесорного часу.

З проведеного аналізу було зроблено висновки про те, що перші покоління генераторів ПВЧ, таких як генератори Фібоначчі з запізненням та LSFR генератори, не забезпечують достатньо великого періоду повторення T, і через це, у більшості випадків вони можуть бути використані

лише як складові частини більш складних алгоритмів, а це, у свою чергу, робить їх мало ефективними. Теж саме стосується лінійного конгруентного генератора. Таким чином, найбільш придатним для досліджень випадкових процесів шляхом стохастичного комп'ютерного моделювання є, розроблений Джоржем Марсальєю, Xorshift генератор та його модифікації. Наприкінці, слід також зауважити, що Xorshift генератор зовсім не є безпроблемним. Бажано було б знайти спосіб збільшення довжини його періоду і це має бути одним з напрямів подальших досліджень.

Список літератури:

1. Law, A.M. and W. Kelton, 2000. Simulation Modeling and Analysis. 3rd Edn., McGraw-Hill, New York, USA. URL: https://surl.li/ftjgxz
2. Knuth, D. E. The Art of Computer Programming. Volume 2. Seminumerical Algorithms. 3rd edition / D. E. Knuth. – Boston, Mass, USA : Addison Wesley, Longman Publishing, 1997. – 762 p. ISBN 0-201-89683-4. URL: https://surl.lu/txwhys
3. Kamalika Bhattacharjee, Krishnendu Maity, Sukanta Das. A Search for Good Pseudo-random Number Generators : Survey and Empirical Studies. DOI:10.48550/arXiv.1811.04035. November 2018. URL: https://arxiv.org/pdf/1811.04035
4. John von Neumann. Various techniques used in connection with random digits. Collected Works, 1963, 5: 768-770. URL : https://surl.li/ancwgf
5. M. Matsumoto, T. Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Transactions on Modeling and Computer Simulation. 8 (1): 3–30. 1998. DOI:10.1145/272991.272995. URL: https://surl.li/wirfwn
6. Saito, M. An Application of Finite Field: Design and Implementation of 128-bit Instruction-Based Fast Pseudorandom Number Generator. Dept. of Math. Graduate School of Science, February 9th, 2007. URL: http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/SFMT/M062821.pdf
7. Yurii Shcherbina, Nadiia Kazakova, Oleksii Fraze-Frazenko. Using the Xorshift generator to simulate stochastic processes. Processing, transmission and security of information – 5 December 2022. URL: https://ceur-ws.org/Vol-3200/paper39.pdf
8. George Marsaglia. Xorshift RNGs. Journal of Statistical Software 08(i14) January 2003. DOI:10.18637/jss.v008.i14. URL: https://www.researchgate.net/publication/5142825_Xorshift_RNGs
9. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. SP 800-22 Rev. 1a, April 2010. URL: https://surl.li/rixqnc
10. Richard P. Brent. Note on Marsaglia's Xorshift Random Number Generators. Journal of Statistical Software. August 2004, Volume 11, Issue 5. DOI:10.18637/jss.v008.i14. URL: https://www.jstatsoft.org/article/view/v011i05
11. Yurii Shcherbyna, Nadiia Kazakova, Oleksii Fraze-Frazenko, Oleksandr Laptiev, Andrii Sobchuk. SELECTION OF RANDOMNESS SOURCE FOR COMPUTER SIMULATION. DOI: https://doi.org/10.18372/2310-5461.59.17944. URL: https://jrnl.nau.edu.ua/index.php/SBT/article/view/17944
12. G. Jay Kerns. Introduction to Probability and Statistics Using R. First Edition. GNU Free Documentation License. ISBN: 978-0-557-24979-4. July 28, 2010. – 412 p. URL : https://ipsur.r-forge.r-project.org/book/download/IPSUR.pdf.
13 Robert V. Hogg, Joseph W. McKean, Allen T. Craig. Introduction to Mathematical Statistics. Seven Edition. ISBN 978-0-321-79543-4. URL : https://surl.lu/advtyx
14. Lehmer, D.H. (1951) Mathematical Methods in Large-Scale Computing Units. The Annals of' the Computation Laboratory of Harvard University, 26, 141-146.
15. Ross S. A First Course in Probability. 8th Edition. 2010. ISBN-13: 978-0136033134 URL: https://github.com/datalater/probability/blob/master/A_First_Course_in_Probability_8th_Edition.pdf
16. Siew-Hwee Kwok, Yen-Ling Ee, Guanhan Chew, Kanghong Zheng, Khoongming Khoo, Chik-How Tan. A Comparison of Post-Processing Techniques for Biased Random Number Generators. WISTP 2011: Information Security Theory and Practice. Security and Privacy of Mobile Devices in Wireless Communication. pp 175–190. URL: https://link.springer.com/content/pdf/10.1007/978-3-540-74619-5_9.pdf
17. Ethan Bogle, Owen Brass, Owen Shen. Fibonacci random generator and fourier analysis. ID: 237526214. 2021. URL : https://surl.li/tctfxb
18. Golomb, S.W. Shift Register Sequences. Aegean Park Press, Laguna Hill, 1982. 324 p. URL : https://surl.li/oeoizz
19. François Panneton, Pierre L'Ecuyer. On the xorshift random number generators. Transactions on Modeling and Computer Simulation 15(4): 346-361. DOI:10.1145/1113316.1113319. URL : https://surl.lu/qxkxbx
20. Richard P. Brent. Note on Marsaglia's Xorshift Random Number Generators. Journal of Statistical Software 11(i04). July 2004. DOI:10.18637/jss.v011.i05. URL : https://surl.li/daujao

Shcherbyna Yu.V., Loginova N.I., Manakov S.Yu. ANALYSIS OF MODERN METHODS OF COMPUTER MODELING OF STOCHASTIC PROCESSES

The article is devoted to the problem of improving the quality of stochastic computer modeling during the design of complex technical systems that ensure the operation of transport, production, management of infrastructure facilities and data exchange in information and communication systems. The effectiveness of such modeling is determined by the presence of adequate mathematical models that formally describe the behavior of real systems in stochastic conditions of their operation, and the quality of sources of randomness that allow reproducing such conditions. One of the main problems that determines the quality of stochastic computer modeling is the choice of a software method for generating pseudo-random numbers with a high degree of uniformity of their probability distribution. This problem should be solved in two directions: first, the algorithm for forming a pseudo-random number stream should be unpretentious to such computational resources as memory and processor time and, secondly, such a stream should not differ from a real random uniformly distributed number stream. The paper provides an analysis of known software methods for generating pseudo-random numbers used for computer modeling. Those considered to be the most economical are considered, namely, the Fibonacci generator with a delay, generators based on linear shift registers with feedback, and the Xorshift generator, which is based on economical, from the point of view of computer implementation, logical operations and shift operations. Based on the analysis of scientific publications, it was concluded that, taking into account modern requirements for the quality of computer modeling, the Xorshift generator should be used as the most effective source of randomness. As for the Fibonacci generators with a delay and generators based on shift registers with feedback, they can be used only as components of more complex algorithms for generating pseudo-random numbers.

Key words: *computer modeling, inverse function method, random number rejection method, pseudorandom number generator, statistical analysis.*

Дата надходження статті: 25.07.2025
Дата прийняття статті: 31.07.2025
Опубліковано: 27.10.2025