РАНДОМІЗАЦІЯ ПСЕВДОВИПАДКОВИХ ЧИСЕЛ НА ВИХОДІ ПРОГРАМНОГО ГЕНЕРАТОРА

Щербина Ю. В.

доцент кафедри інформаційних технологій
Національного університету «Одеська юридична академія»

Постановка проблеми

Результатом розвитку сучасної комп'ютерної техніки є те, що, на відміну від минулого сторіччя, моделювання складних стохастичних систем перетворилось із другорядного етапу проектування на один із основних способів розроблення процесів сучасного виробництва, управління інфраструктурними об'єктами та економічною діяльністю.

Для створення комп'ютерної моделі процесу, який протікає в умовах відсутності абсолютної визначеності, в першу чергу необхідно мати у своєму розпорядженні генератор з абсолютно рівномірним розподіленням випадкових чисел на його виході. Числові послідовності з іншими типами розподілення ймовірностей формуються саме із рівномірного розподілення за принципом, описаним у [1]. Зважаючи на те, що обчислювальні пристрої – це детерміновані автомати, вони здатні формувати на виході лише псевдовипадкові числа (ПВЧ) з обмеженим періодом повторення *T*. Щоб така послідовність виглядала як випадкова, період має бути достатньо великим, тобто він повинен на порядки перевищувати довжину процесу, що підлягає моделюванню. На даний момент ця проблема розробниками програмного забезпечення вже вирішена. Так, наприклад, генератор ПВЧ, побудований з використанням чисел Мерсена (MT), має період повторення 2^{19937} – 1 біт. Що ж стосується рівномірності розподілення чисел на його виході, то проведені експерименти показують, що такий генератор не тільки не може бути використаний для криптографічних перетворень, але й не відповідає вимогам моделювання. Те ж саме сказати і про лінійний конгруентний генератор (LCG) [2].

Ціль дослідження. Зважаючи на те, що двоетапна система моделювання дуже чутлива до рівномірності розподілення чисел на виході обраного генератора, і те, що генератори LCG і MT, включені у більшість середовищ програмування, не відповідають вимогам рівномірності за критерієм Пірсона χ^2 [3], єдиним виходом вбачається пошук такого способу постоброблення потоку псевдовипадкових чисел на виході генератора, який відповідав би відповідність вимогам моделювання.

Визначення способу постоброблення потоку псевдовипадкових чисел

В середині минулого сторіччя фон Нейманом було зроблене обґрунтоване зауваження про неприйнятність фізичних генераторів випадкових чисел для задач моделювання через технічні труднощі можливості повторної реалізації отриманої вибірки. З урахуванням цього положення, були запропоновані такі

алгоритми, як метод середніх квадратів [4] та лінійний конгруентний метод. У той же період Д. Кнутом [2], було показано, що вони теж не забезпечували необхідної рівномірності сформованого потоку. Оскільки зробити ідеальний генератор практично не можливо, ідея постоброблення як для генераторів реальних випадкових чисел, так і для генераторів ПВЧ залишається актуальною і донині.

Для потреб постоброблення використовують спеціальні прості коректори (Ad hoc simple correctors), алгоритми видалення (Extractor algorithms) та хеш-функції (Whitening with hash functions). Загальною вимогою до всіх способів постоброблення є мінімізація обчислювальних ресурсів, для їх реалізації.

Прикладом простого коректора може бути коректор, описаний фон Нейманом, де він пропонує об'єднувати пару бітів, отриманих від незалежних джерел за принципом: якщо біти співпадають (00 або 11), то вони скасовуються, комбінації бітів 01 відповідає 0-й вихідний біт, а комбінації 10 відповідає 1-й вихідний біт. Максимальна ефективність такого алгоритму складає в середньому 4 вхідних бити на 1 вихідний біт. Саме у цій роботі фон Нейманом було зроблено акцент на труднощах генерації випадкових десяткових чисел.

Видалення (або відбілювання) передбачає зменшення кореляції символів на виході джерела ентропії. Воно збільшує однорідність та рівномірність розподілення символів у складі потоку на виході генератора. Для його реалізації використовують такі хеш-алгоритми як MD5, SHA-1, SHA-2, SHA-256 або SHA-512.

Екстракторами випадковості називають алгоритми, які перетворюють низькоякісний потік вхідних величин у майже рівномірний потік числових символів за допомогою невеликої кількості гарантовано випадкових бітів. Спосіб такого перетворення описано в роботі Люка Тревісена [5]. Він вводить поняття мінімальної ентропії, яка характеризує нерівномірність розподілення величини X у діапазоні $\{0,1\}^n$, де n – це бінарна комбінація на виході джерела. У разі рівномірного розподілення всі комбінації будуть рівно імовірними і ентропія буде максимальною.

Ще один спосіб підсилення випадковості величин вихідного потоку генератора ПВЧ запропоновано в роботі [6]. Він основується на постфільтрації через деякий детермінований процес. Ідея полягає в тому, щоб за рахунок використання еластичних функцій розділити вихідні символи на випадкові і "недостатньо випадкові". По суті, пропонується "просіяти" вихідний потік, щоб видалити із нього «зайві» числа, які «потворять» загальну картину рівномірності розподілення ймовірностей. Такий спосіб достатньо просто реалізувати, оскільки він вимагає невелику кількість обчислювальних ресурсів.

Використання критерію Пірсона χ^2, передбачає розділення діапазону чисел на виході генератора на k – інтервалів з подальшим обчисленням показника рівномірності по правилу $\chi^2 = (n_i - Np_i)^2/Np_i$. Тут N – це розмір вибірки, p_i – ймовірність попадання в i-тий інтервал, яка у разі рівномірного розподілення

дорівнює $1/k$, n_i – реальна кількість чисел, що попала в i-тий інтервал, а Np_i – очікувана кількість чисел, що має попасти в i-тий інтервал.

Математичне очікування величини, що має попасти в сегмент, обмежений умовами $x_{min} \le n_i < x_{min}$ визначається як $m_i = (x_{min} + x_{min})/2$. Тоді кількість чисел, що попали в i-тий сегмент s_i, орієнтовно буде дорівнювати величині $s_i^* = n_i^* m_i$. Тут x_{min} і x_{min} відповідні межі кожного з інтервалів. Звичайно, що кожного разу ця сума буде або менше, або більше s_i^* але суттєвої різниці не буде. Це дає можливість сформулювати наступний алгоритм. Якщо сума чисел s_i^*, що попали в i-тий сегмент гістограми перевищує величину s_i^*, то всі інші числа які в нього попадають пропускаються. Звичайно, що кількість чисел в сегментах залишиться різною, але нерівномірність вибірки стане меншою.

На рисунку 1 наведені результати випробувань вибірки довжиною N = 1024 дійсних десяткових псевдовипадкових чисел на виході MT генератора до і після постоброблення описаним способом. В останньому разі для 5% точності показник χ^2 = 03438, що менше критичного значення χ^2 = 7.2609.

Висновки

Найчастіше для сучасного комп'ютерного моделювання використовуються програмні пакети Boost, Glib, C++, Python, Ruby, R, PHP, MATLAB та їм подібні. Всі вони написані на мові C++ і використовують її генератори ПВЧ. Зважаючи на це, хорошим варіантом є будування моделей стохастичних процесів саме на цій мові, використовуючи програмні середовища Visual Studio або QT5, до складу яких входить велика кількість додаткових бібліотек, що містять різноманітні генератори ПВЧ.

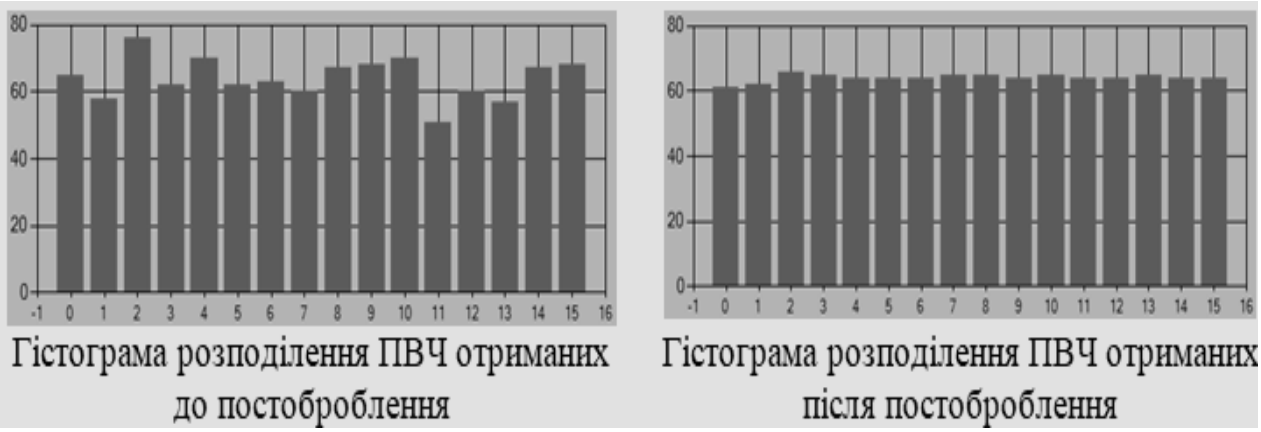

Рис. 1. Гістограма розподілення ПВЧ отриманих за допомогою функції uniform_real_distribution <> mersenne(0, 1)

Щоб забезпечити задану якість моделювання, необхідно попередньо виконати перевірку якості потоку на виході обраного генератора на предмет рівномірності розподілення сформованих чисел і, якщо є необхідність, провести додаткову його рандомізацію.

Список використаних джерел:

1. Mark A. Pinsky, Samuel Karlin. An Introduction to Stochastic Modeling, Fourth Edition., Academic Press, 2010. URL: https://faculty.ksu.edu.sa/sites/default/files/an_introd_to_stoch_modeling_4th_ed.pdf.
2. D. E. Knuth, The Art of Computer Programming, Volume 2: Seminumerical Algorithms, 3rd. ed., Boston, Mass, USA : Addison-Wesley, Longman Publishing, Addison-Wesley, Reading, Mass, 1998. URL: https://doc.lagout.org/science/0_Computer%20Science/2_Algorithms/The%20Art%20of%20Computer%20Programming%20%28vol.%202_%20Seminumerical%20Algorithms%29%20%283rd%20ed.%29%20%5BKnuth%201997-11-14%5D.pdf.
3. J. H. Ahrens, U. Dieter, A. Grube, Pseudo-random numbers. Computing 6 (1970) 121-138). URL: https://doi.org/10.1007/BF02241740.
4. J. von Neumann. Various techniques for use in connection with random digits. Applied Math Series, Notes by G. E. Forsythe, in National Bureau of Standards, Vol. 12, 36 – 38, 1951. URL: https://mcnp.lanl.gov/pdf_files/nbs_vonneumann.pdf.
5. Trevisan L. Extractors and Pseudorandom Generators 1999. Journal of the ACM URL: http://theory.stanford.edu/~trevisan/pubs/extractor-full.pdf.

ПРОГРАМНІ ЗАСОБИ РОЗРОБКИ ЧАТ-БОТІВ

Ярош Є. В.

студентка 1-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

На сьогодні люди все частіше звертаються до чат-ботів. Нині за умов війни люди масово почали використовувати чат-боти для інформаційної боротьби з ворогом, отримання корисної інформації про роботу волонтерів чи про прихистки для людей у різних регіонах. Чат-боти допомагають підбирати музику та книжки, купувати різноманітні товари та квитки онлайн. Крім того, вони використовуються для імітації спілкування з живою людиною, як персональні асистенти, юридичні консультанти та навіть психотерапевти. Експерти прогнозують чат-ботам грандіозний розвиток в найближчі роки та порівнюють цей бум з розвитком вебу та соціальних мереж свого часу [1, 2]. Саме тому розробка чат-ботів є доволі актуальною.

Чат-боти можуть мати різне призначення і різний функціонал, і саме від цього залежить вибір платформи, в якій вони працюватимуть, і вибір засобів розробки. За призначенням розрізняють чат-боти для розмов на широкий спектр тем та чат-боти, орієнтовані на діалог тільки за певною тематикою. За функціоналом чат-боти розділяють на 3 види: інформаційно-комунікаційні, «питання – відповідь» та функціональні [3].

Інформаційно-комунікаційні чат-боти призначені виключно для інформування людей, щоб поділитися інформацією про спеціальні пропозиції та знижки,