МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерних наук

МУНДЯК Віталій Романович

Математичне та програмне забезпечення генератора псевдовипадкових чисел на основі нечіткої логіки / Mathematical Tools and Software for Pseudorandom Number Generator Based on Fuzzy Logic

спеціальність: 121 - Інженерія програмного забезпечення
освітньо-професійна програма - Інженерія програмного забезпечення

Кваліфікаційна робота

Виконав студент групи ІПЗм-21
В. Р. Мундяк

_____

Науковий керівник:
к.е.н., доцент, Л. І. Гончар

_____

Кваліфікаційну роботу
допущено до захисту:

"___" _____ 20___ р.

Завідувач кафедри
_____ **А. В. Пукас**

ТЕРНОПІЛЬ - 2022

ЗМІСТ

ВСТУП

Актуальність теми. В даний час генератори псевдовипадкових послідовностей та чисел стали невід’ємними елементами вирішення завдань у багатьох прикладних областях, включаючи статистичне та імітаційне моделювання, телекомунікації, інформаційну безпеку та ін. При цьому до таких генераторів пред'являються суворі вимоги до якості формованих послідовностей для того, щоб вони були близькими до істинно випадкових.

З метою отримання псевдовипадкових послідовностей, близьких до істинно випадкових, генератор має задовольняти наступним основним вимогам:

- рівномірність розподілу елементів сформованих послідовностей;
- задоволення вимогам найбільш відомих наборів статистичних тестів: NIST, DIEHARD та ін;
- непередбачуваність;
- великий період формованих послідовностей;
- стійкість до алгебраїчних атак;
- швидкодія.

Мета і задачі дослідження

Метою роботи є підвищення якості генерації псевдовипадкових послідовностей, заснованих на регістрах зсуву з лінійним зворотним зв'язком, за рахунок застосування апарату теорії нечітких множин.

Відповідно до поставленої мети у роботі потрібно вирішити такі основні завдання дослідження:

1) розробка моделі генератора псевдовипадкових чисел, заснованого на нечіткій логіці;

2) розроблення комплексу програм генерації псевдовипадкових чисел;

3) проведення досліджень параметрів запропонованої моделі генератора псевдовипадкових чисел, заснованого на нечіткій логіці, а також

якості сформованих при цьому псевдовипадкових послідовностей з застосуванням розробленого комплексу програм, а також вибору найбільш відповідних параметрів моделі.

Об'єкт дослідження – генератори псевдовипадкових чисел, побудовані на регістрах зсуву з лінійним зворотним зв'язком.

Предмет дослідження – методи та програмні засоби генераторів псевдовипадкових чисел.

Методи дослідження

В роботі використовувалися методи математичного моделювання, алгебраїчної теорії чисел та полів, теорії ймовірностей та математичної статистики, теорії нечітких множин.

Наукова новизна одержаних результатів

Запропоновано модель генератора псевдовипадкових чисел, засновану на нечіткій логіці та підборі найбільш підходящих параметрів моделі, що дозволяють підвищити якість сформованих псевдовипадкових послідовностей.

РОЗДІЛ 1
ГЕНЕРАТОРИ ВИПАДКОВИХ І ПСЕВДОВИПАДКОВИХ ЧИСЕЛ

1.1. Випадкові та псевдовипадкові числа

Необхідність використання випадкових чисел під час проведення наукових досліджень та в технічних додатках з’явилася досить давно. Завдання формування послідовності випадкових чисел досить часто потрібно вирішувати в таких прикладних областях, як імітаційне моделювання, статистичне моделювання, захист інформації, телекомунікації.

Досить давно урни з кулями та гральні кістки широко використовувалися в азартні ігри. Досі вони вважаються найпростішими істинними генераторами випадкових чисел. Пізніше було реалізовано ряд механічних генераторів випадкових чисел, що складаються з барабана, що перемішує кулі з числами, та пристрої, що отримує по черзі кулі з барабана [1-4]. У 1890 році англійський дослідник Френсіс Гальтон описав спосіб використання ігрових кісток для генерації випадкових чисел у наукових цілях [5,6]. Надалі завдяки розвитку обчислювальної техніки з'явилися електронні генератори. Одним із відомих прикладів таких генераторів є генератор Тьюринга, побудований з урахуванням резисторного генератора шуму. Однак даний тип генераторів який завжди давав хороші результати, але дляприкладних завдань були необхідні великі масиви випадкових даних.

У 1939 М.Ж. Кендал і Б. Бабінгтон-Сміт представили першу машину, що генерує випадкові числа, і використовували її для побудови таблиці, що містить 100 000 випадкових чисел [7-9]. Через 16 років корпорацією RAND за допомогою спеціальних пристроїв була отримана таблиця, що містить 1 млн. випадкових чисел. Надалі розвиток обчислювальної техніки дозволило

ще більше нарощувати швидкість генерації та обсяг генерованих випадкових чисел [10-12].

Однак апаратні методи генерації випадкових чисел не можуть бути використані для проведення низки обчислювальних експериментів у зв'язку з неможливістю повторного формування послідовності та підтвердження отриманого результату. У зв'язку з цим, увага була звернена на алгоритмічні методи формування псевдовипадкових чисел. У 1946 року, Джон фон Нейман створив метод генерації чисел, “схожих на випадкові”. Суть методу полягала у введенні рекурентної процедури формування наступного псевдовипадкового числа на підставі попереднього шляхом зведення його у квадрат та виділення середніх цифр.

Зрозуміло, що отримана послідовність не є випадковою, але є допустимою для низки додатків. Подібні послідовності, що формуються алгоритмічно, були названі псевдовипадковими послідовностями, а формують їх процедури–генераторами псевдовипадкових чисел.

Генератор псевдовипадкових чисел (pseudorandom number generator, PRNG) - алгоритм, що породжує послідовність чисел, елементи якої майже незалежні один від одного і підпорядковуються заданому розподілу (зазвичай рівномірному).

Псевдовипадкові числа широко використовуються в ряді чисельних методів пошуку наближеного вирішення завдань із застосуванням методу Монте-Карло. Як типовий приклад можна навести знаходження наближеного значення певного інтегралу.

В даний час генератори псевдовипадкових чисел широко використовуються при вирішенні різних практичних завдань у галузях імітаційного моделювання, захисту інформації, телекомунікацій та ін. При цьому при формуванні значну увагу слід приділяти вивченню властивостей послідовностей на основі, яких надалі формуються псевдовипадкові числа, оцінки близькості даних властивостей до властивостей випадкових послідовностей.

1.2. Основні вимоги якості до генераторів псевдовипадкових чисел

Для формування якісних послідовностей генератор псевдовипадкових чисел має генерувати так званий «псевдовипадковий шум». Статистичні властивості повинні бути близькими до властивостей істинно випадкових послідовностей із рівномірним законом розподілу. В цьому випадку алгоритмічні генератори з деяким наближенням можуть замінювати собою апаратно реалізовані. Основними вимогами до розроблюваних генераторів псевдовипадкових послідовностей є: підпорядкування формованої послідовності рівномірному закону розподілу, непередбачуваність, відсутність автокореляції. Крім цього, звертають увагу на обчислювальну складність формування псевдовипадкових послідовностей, яка не повинна бути надто високою.

Одним із перших вимог до формованих псевдовипадкових послідовностей були сформовані Соломоном Голомбом. У літературі дані вимоги відомі як «постулати Голомба». Перелік даних вимог включає [13-19]:

1) кількість елементів "1" у кожному з аналізованих періодів псевдовипадкових послідовностей має відрізнятися від кількості елементів "0" лише на одиницю;

2) у кожному періоді половина серій, що включають однакові біти, повинна мати довжину один, одна чверть повинна мати довжину два, одна восьма повинна мати три довжину і т.д. Більш того, для кожної з цих довжин має бути однакова кількість серій з "1" та "0".

3) формована псевдовипадкова послідовность має бути незалежною, значення попередніх елементів псевдовипадкових послідовностей не повинно давати можливості передбачити її наступний елемент. Автокореляційна функція для псевдовипадкових послідовностей повинна приймати лише два значення – 0 та 1.

Послідовність, яка задовольняє основним постулатам (правилам 1-3) часто називається "ПШ-послідовністю", де ПШ позначає "псевдо-шумова". В даний час для перевірки псевдовипадкових послідовностей часто використовуються суворіші умови.

Для додатків та завдань, в яких потрібні дійсно якісні псевдовипадкові числа, кожну псевдовипадкову послідовность перед її використанням необхідно ретельно перевірити за допомогою набору статистичних тестів. Якісні генератори повинні задовольняти низку вимог:

- мати добрі статистичні властивості (успішно проходити всі відомі статистичні тести випадковості);

- забезпечувати великий період для згенерованих псевдовипадкових послідовностей;

- забезпечувати непередбачуваність елементів псевдовипадкових послідовностей;

- мати високу швидкодію (продуктивність), низьке енергоспоживання під час реалізації на апаратному рівні;

- забезпечувати стійкість до алгебраїчних атак.

Застосування кожного з існуючих наборів тестів не гарантує того факту, що досліджуваний генератор є якісним. У зв'язку з цим, при тестуванні слід покладатися на такі положення:

1. При тестуванні генератора псевдовипадкових послідовностей необхідно використовувати як можна більше відомих статистичних критеріїв, відшукуючи в псевдовипадкових послідовностях всі можливі закономірності. Зокрема, слід застосовувати такі добре зарекомендовані набори тестів, як DIEHARD, NIST, графічні тести та ін.

2. Слід керуватися принципом, згідно з яким генератор є якісним, якщо жоден із використовуваних статистичних тестів не забракує його.

3. Слід ранжувати досліджувані генератори псевдовипадкових послідовностей за комплексним критерієм, що враховує результати проходження даного генератора різних статистичних тестів.

1.3. Класифікація методів генерації псевдовипадкових чисел

Проблема отримання випадкових (псевдовипадкових) чисел на ЕОМ може бути вирішена у різний спосіб. Питання про вибір способу генерування випадкових чисел є першорядним, оскільки від його успішного вирішення багато в чому залежить успіх вирішення всього завдання.

Традиційно для вирішення цієї проблеми використовують два різних підходи. Перший підхід полягає в тому, що випадкові числа формуються на основі сформованих псевдовипадкових послідовностей, які використовують якийсь реальний фізичний процес, що володіє певними властивостями, та перетворений на форму, придатну для використання апаратнопрограмними частинами моделі, зрештою - ЕОМ. Другий підхід орієнтований застосуванням спеціальних математичних перетворень, частіше всього заданих у рекурентній формі, що дозволяють отримати детерміновані числові послідовності з характеристиками, близькими до справді випадкових послідовностей.

У літературі присутні різні класифікації генераторів псевдовипадкових послідовностей, які представлено на рисунку 1.1. При цьому чітко виділяються детерміновані генератори та недетерміновані генератори. Також можна навести класифікацію у вигляді, поданому на рисунку 1.2.

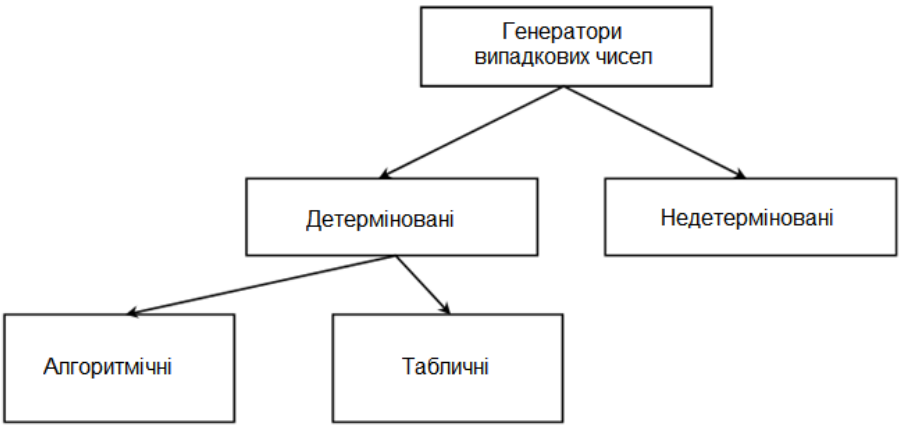

Рис. 1.1. Класифікація генераторів псевдовипадкових чисел

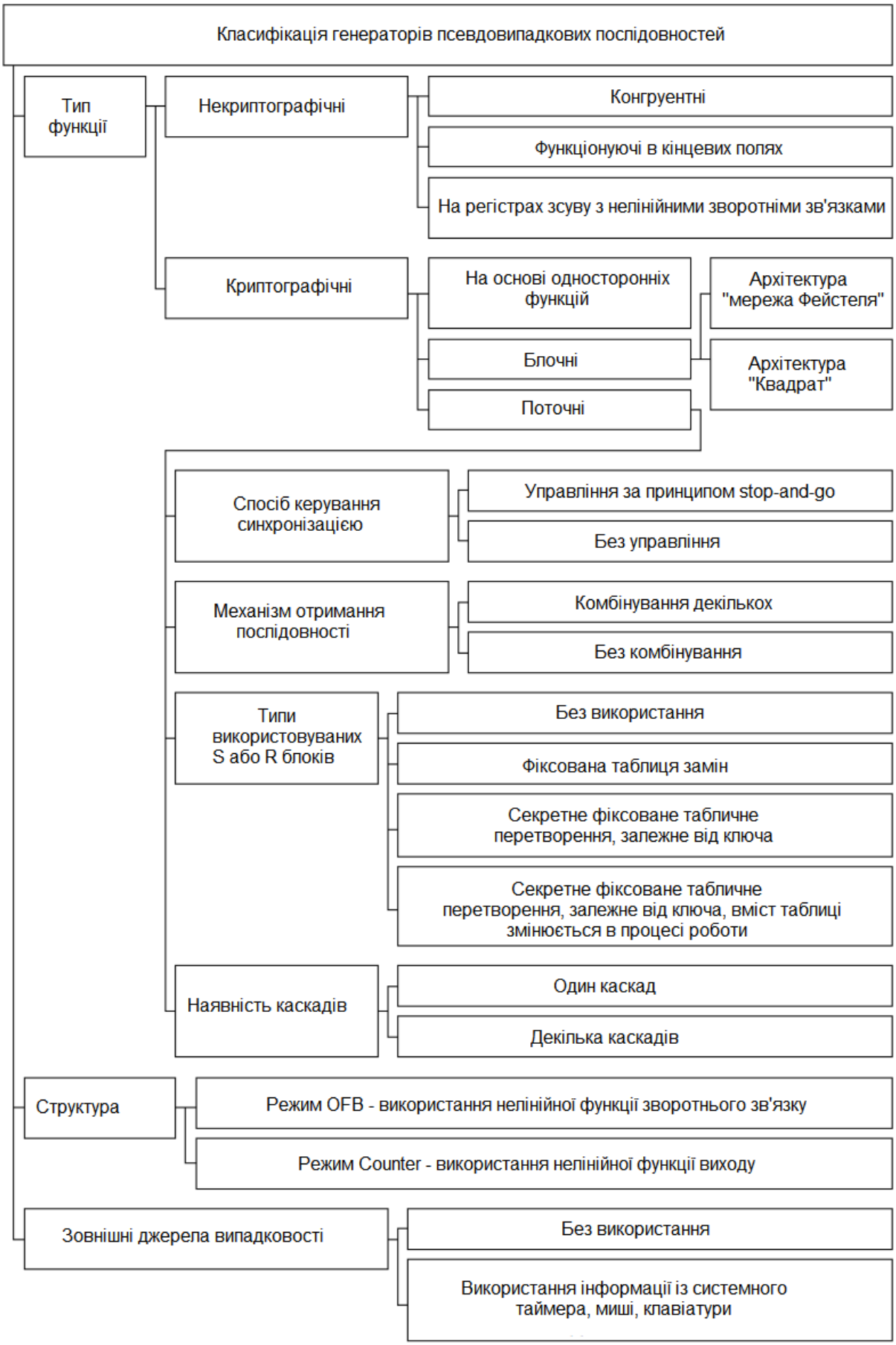

Рис. 1.2. Класифікація генераторів псевдовипадкових послідовностей

Недетерміновані генератори формують результат на основі непередбачуваного фізичного джерела, не керованого людиною. Фізичні властивості деяких природних об'єктів та різні фізичні шуми, наприклад дробовий шум у резисторі чи космічне випромінювання може бути джерелами ентропії для генератора. У наслідок законів квантової фізики, деякі природні явища (наприклад, радіоактивний розпад атомів) абсолютно випадкові і не можуть бути в принципі передбачені.

− Дробовий шум − це шум в електричних ланцюгах, викликаний дискретністю носіїв електричного заряду. Також цим терміном називають шум в оптичних приладах, спричинений дискретністю переносника світла. У даному випадку, як джерело шуму використовують фотоелектронний помножувач або електровакуумні фотоелементи.

− Радіоактивний розпад використовується як джерела шуму, оскільки для нього характерна випадковість кожного окремого акту розпаду. В результаті на приймач (наприклад, лічильник Гейгера) у різні проміжки часу попадає різна кількість частинок.

− Тепловий шум у резисторі або транзисторі, з якого після посилення виходить генератор випадкових напруг. Наприклад, генератор чисел в комп'ютер (Ferranti Mark 1) був заснований на цьому явищі. На рисунку 1.3 наведено приклад побудови генератора випадкових чисел за допомогою теплового пристрою, що шумить (транзистора).

− Атмосферний шум, виміряний радіоприймачем. Сюди також можна віднести і прийом частинок, що прилітають на землю з космосу, які реєструються приймачем, а їх кількість у різні проміжки часу є випадковою величиною.

− Різниця у швидкості ходу годинника - явище, що полягає в тому, що хід різних годинників не буде абсолютно збігатися. Існують і доступніші джерела випадковості в персональних комп'ютерах. Автори генератора випадкових чисел часто використовують такі джерела ентропії, як шум звукової карти, лічильник тактів процесора та інші.

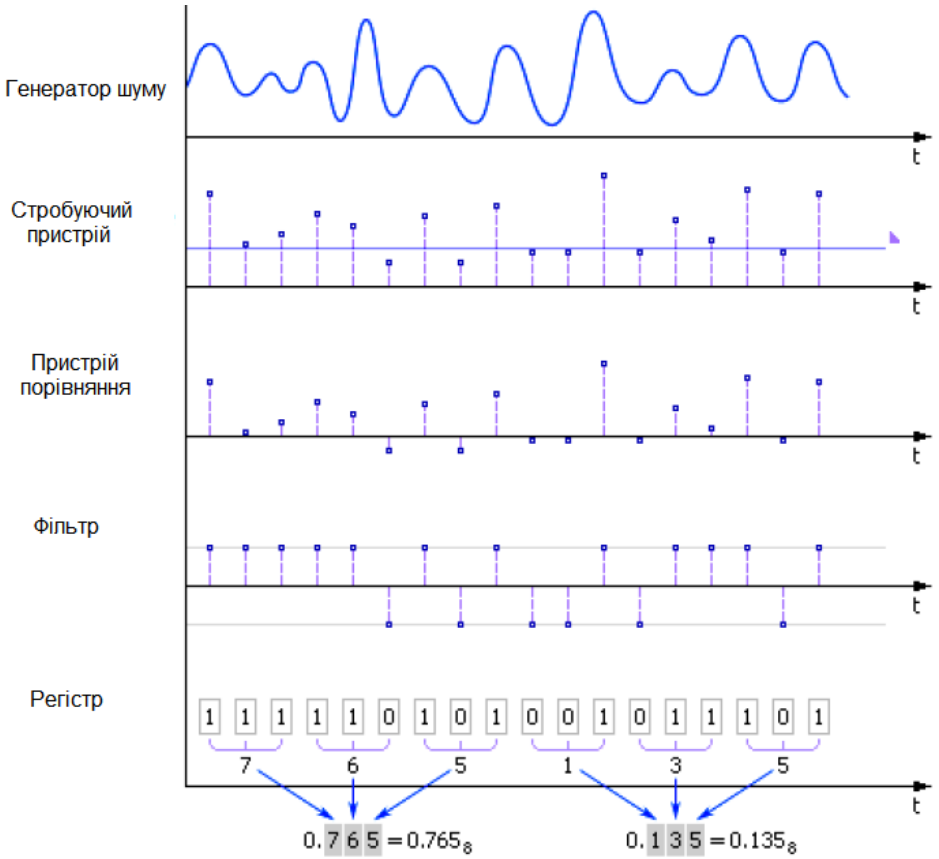

Рис. 1.3. Приклад побудови генератора випадкових чисел за допомогою пристрою теплового шуму

Основні недоліки генераторів випадкових послідовностей можна софрмувати наступним чином:

– складність реалізації та відносно повільна швидкість роботи;

– схильність до впливу дестабілізуючих факторів на точність та стаціонарність роботи;

– обмежена точність формування основних статистичних характеристик;

– неможливість повторного формування послідовності для підтвердження одержаного результату.

Вище перелічені недоліки обмежують використанням недетермінованих генераторів.

Детерміновані генератори (часто називаюьб генераторами псевдовипадкових чисел) включають детермінований алгоритм, що генерує послідовність чисел виходячи зі свого початкового значення.

Тривала історія розвитку методів імітаційного моделювання, зв'язку, захисту інформації поставила їх до ряду найважливіших блоків та алгоритмів сучасних ЕОМ. Існує два основних типи детермінованих генераторів: табличні та алгоритмічні.

Табличні генератори використовують спеціальним чином складені таблиці, містять некорельовані цифри. У таблиці 1.1 наведено частину такої таблиці. Обходячи таблицю зліва направо зверху донизу, можна формувати рівномірно розподілені від "0" до "1" випадкові числа з необхідним числом знаків після коми. Так як цифри в таблиці не залежать одна від одної, то таблицю можна обходити різними способами: справа наліво, зверху донизу, обирати цифри, що знаходяться на парних позиціях тощо.

Таблиця 1.1

Приклад табличного генератора псевдовипадкових чисел

Випадкові цифри								Рівномірно розподілені від "0" до "1" випадкові числа
9	2	9	2	0	4	2	6	0.929
9	5	7	3	4	9	0	3	0.204
5	9	1	6	6	5	7	6	0.269
....								...

Перевагою використання цього є те, що він дає дійсно числа, близькі за статистичними властивостями до випадкових. Основні недоліки даного методу формування полягають у тому, що для зберігання великої кількості цифр потрібно багато пам'яті, виникають великі труднощі породження та перевірки такого роду таблиць, повтори при використанні таблиці не

гарантують числової випадковості послідовності, отже вони не гарантують надійності результату.

Алгоритмічні генератори найчастіше являють собою рекурентну процедуру наступного виду $x_{i+1} = f(x_{i\,k-+1}, x_{i\,k-+2}, ..., x)_i$, де i – час, а f – деяка функція перетворення до останніх членів послідовності чисел $x_{i\,k-+1}, x_{i\,k-+2}, ..., x$ у нове значення x_{i+1}. Послідовності, згенеровані такими методами рано чи пізно починають повторювати одну й ту саму послідовність чисел.

На даний момент відомо більше тисячі подібних генераторів псевдовипадкових послідовностей, які відрізняються функціями та параметрами, що використовуються. Істотно розрізняються і статистичні властивості генерованих ними числових послідовностей. Більшість алгоритмів, що використовуються, мають на увазі використання досить складної в обчислювальному відношенні функції f. Як показано на рисунку рисунку 1.3, алгоритмічні генератори можна розділяти на дві основні групи: некриптографічні і криптографічні.

Далі представлені окремі методи, що використовуються в некриптографічних: метод Фон-Неймана, лінійні конгруентні генератори, нелінійні конгруентні генератори, генератори на базі клітинного автомата, генератори, побудовані на основі регістрів зсуву з лінійним зворотним зв'язком. Останній вид вважається одним із найбільш ефективних методів для програмної та апаратної реалізації.

Метод Фон-Неймана. Цей метод був запропонований Джоном фон Нейманом у 1946 [19]. Нехай є деяке чотиризначне число R_0. Це число зводиться в квадраті і заноситься до R_1. Далі з R_1 береться середина (чотири середні цифри), які формують нове випадкове число, що записується в R_0 (рисунок 1.4). Далі процедура повторюється.

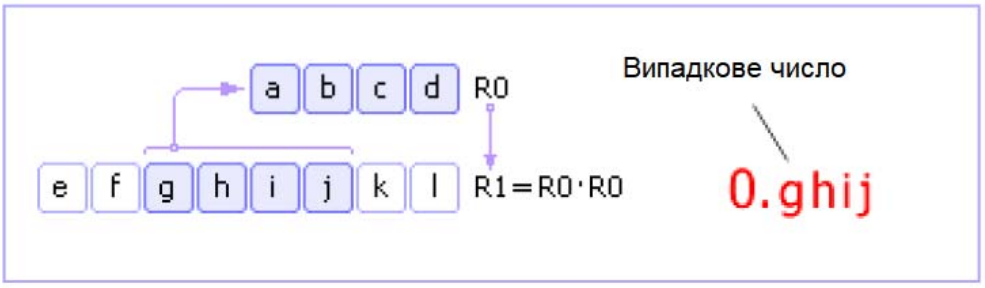

Рис. 1.4. Схема роботи методу Фон-Неймана

Таким чином, використання лінійних регістрів зсуву зі зворотним зв'язком дозволяє досить просто формувати двійкові псевдовипадкові послідовності. При цьому коефіцієнти характеристичного полінома $c_0, c_1, \dots, c_{n-1}$ – і початкові значення мають бути ретельно обрані та триматися у секреті. Однак псевдовипадкові послідовності, згенеровані з допомогою регістрів зсуву з лінійним зворотним зв'язком, не є достатньо надійними, у зв'язку з чим останнім часом багато уваги стали приділяти регістрам нелінійного зсуву. При цьому в різних роботах виробляються спроби об'єднання кількох підходів нелінійним способом (за допомогою нелінійної функції f), щоб отримати псевдовипадкові послідовності з добрими властивостями.

Існує ряд методів, що використовуються для підвищення якості генератора псевдовипадкових послідовностей, заснованих на регістрах зсуву з лінійним зворотним зв'язком. Найбільш відомими є:

- використання нелінійної функції фільтрації (фільтруючий генератор);

- використання виходу одного генератора псевдовипадкових послідовностей для управління синхросигналом інших генераторів псевдовипадкових послідовностей;

- використання нелінійної функції, що поєднує виходи кількох генераторі псевдовипадкових послідовностей (комбінуючий).

Будь-який генератор можна досить просто перетворити в генератор псевдовипадкович чисел, здійснюючи за допомогою відомих алгоритмів вибір заданих біт сформованої послідовності.

У зв'язку з цим, надалі у роботі ми говоритимемо про генератори, маючи на увазі формування у межах нього послідовностей як один з етапів формування чисел.

Методи теорії нечітких множин пропонують ефективний апарат запровадження нелінійності в комбінуючі генератори псевдовипадкових послідовностей, побудовані з урахуванням регістрів зсуву з лінійним зворотним зв'язком.

При цьому нелінійна комбінуюча функція може бути побудована на основі аналізу статистичних властивостей виходів використовуваних регістрів зсуву з лінійним зворотним зв'язком із застосуванням лінгвістичних змінних та нечітких продукційних правил. Їх використання для завдання нелінійної функції в архітектурі генератора дозволить побудувати адаптивну структуру генератора.

Це дасть можливість експерту на попередньому етапі визначити основні параметри генератора псевдовипадкових послідовностей, що використовуються для аналізу статистичних властивостей виходів використовуваних регістрів зсуву з лінійним зворотним зв'язком, та надалі виконати тюнінг даних параметрів, здійснивши пошук кращих, які забезпечують якість формованих псевдовипадкових послідовностей.

Таким чином, питання побудови архітектури та моделі генератора з нелінійною комбінуючою функцією, що базується на застосуванні апарату теорія нечітких множин є центральним питанням магістерського дослідження. Крім того, актуальним є:

- виділення основних параметрів даного генератора, що впливають на якість формуються псевдовипадкових послідовностей;

- дослідження даних параметрів, а також якості формованих при цьому псевдовипадкових послідовностей;

- вибір найбільш підходящих параметрів моделі.

Крім того, якість роботи розроблюваного генератора безпосередньо залежить від періоду послідовностей, що формуються характеристичними примітивними поліномами, використовуваними регістрами зсуву з лінійним зворотним зв'язком від обраних характеристичних поліномів.

Примітивні характеристичні поліноми, що вибираються, повинні забезпечувати максимальний період псевдовипадкових послідовностей. У зв'язку з цим потрібно розробити ефективний метод формування множини характеристичних примітивних поліномів для регістрів зсуву з лінійним зворотним зв'язком розробленого генератора псевдовипадкових чисел, заснованого на нечіткій логіці.

Висновки до першого розділу

1. Генератори псевдовипадкових послідовностей з хорошими статистичними властивостями застосовуються для вирішення багатьох прикладних завдань, таких як генерація криптографічних ключів, реалізація протоколів аутентифікації, створення імітаційних моделей і т.д. Для формування якісних послідовностей потрібні хороші статистичні властивості, непередбачуваність, довгий період, ефективність, відтворюваність.

2. Методи теорії нечітких множин пропонують ефективний апарат запровадження нелінійності в комбінуючі генератори псевдовипадкових послідовностей, побудовані з урахуванням регістрів зсуву з лінійним зворотним зв'язком.

3. Застосування методів теорії нечітких множин для задання нелінійної функції в архітектурі дозволить побудувати адаптивну структуру генератора. Це дасть змогу експерту на попередньому етапі визначити основні параметри генератора, що використовуються для аналізу статистичних властивостей виходів використовуваних регістрів, і надалі виконати тюнінг даних параметрів, здійснивши пошук кращих із них, які забезпечують якість формування послідовностей.

РОЗДІЛ 2
МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ ГЕНЕРАТОРА ПСЕВДОВИПАДКОВИХ ЧИСЕЛ НА ОСНОВІ НЕЧІТКОЇ ЛОГІКИ

2.1. Нечіткі множини, лінгвістичні змінні та нечіткий логічний висновок

Теорія нечітких множин ефективний апарат прийняття рішень в умовах неточності та неповноти вихідної інформації, є ефективним засобом обробки невизначеності при формалізації понять, виконуваних експертом.

Лінгвістична змінна – змінна, яка може приймати значення фраз із природної чи штучної мови. Наприклад, лінгвістична змінна "швидкість" може мати значення "висока", "середня", "дуже низька" і т.д. Математичне визначення лінгвістичної змінною називається п'ятірка $\{x, T(x), X, G, M\}$, де x – ім'я змінної; $T(x)$ −множина її значень (терм-множина), що являють собою найменування нечітких змінних, областю визначення кожної з яких є множина X. Множина Т називається базовою терм-множиною лінгвістичної змінної; G − синтаксична процедура, що дозволяє оперувати елементами терм-множини T, зокрема, генерувати нові терми (значення); М − семантична процедура, що дозволяє перетворити кожне нове значення лінгвістичної змінної, утворене процедурою G, нечітку змінну, тобто сформувати відповідну нечітку множину.

Характеристикою нечіткої множини A виступає функція належності $\mu_A(x)$ задане на універсальній множині U, що приймає значення в інтервалі [0,1]. Значення $\mu_A(x) = 0$ означає відсутність приналежності елемента x до множини A, а коли $\mu_A(x) = 1$, то це означає повну приналежність до множини A. Існує понад десяток типових форм кривих для завдання функцій

належності. Найвідомішими з них є: Z-подібні та S-подібні функції, П-подібні функції.

Поняття нечіткого висновку займає найважливіше місце у нечіткій логіці. На рисунку 2.1 – представлено загальну схему роботи системи нечіткого висновку

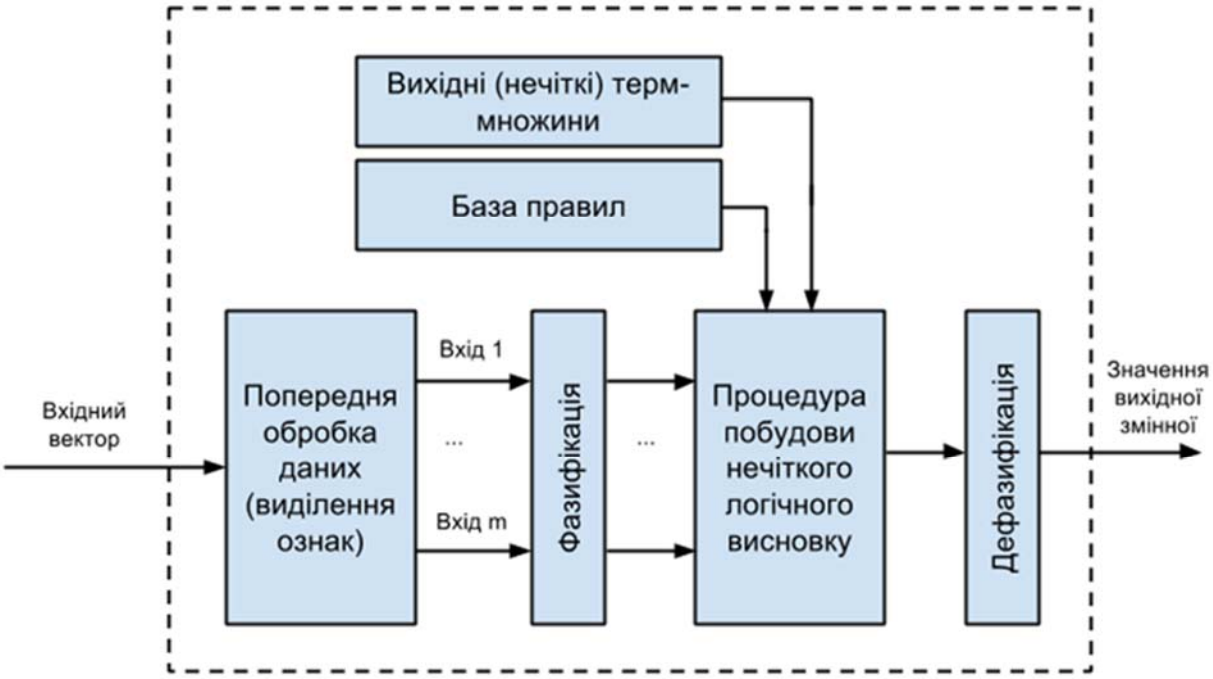

Рис. 2.1. Загальна схема роботи системи нечіткого висновку

Чіткий набір вхідних даних збирається і перетворюється на нечіткий вигляд з допомогою нечітких лінгвістичних змінних та функцій належності. Цей крок відомий як фазифікація. Після цього виконується нечіткий логічний висновок на основі набору нечітких правил. Зрештою, результат перетворюється на чіткий вигляд за допомогою функції дефазифікації. Сьогодні найбільшого практичного поширення набули наступні алгоритми нечіткого висновку: Мамдані, Сугено, Цукамото, Ларсена.

Як показано раніше, методи теорії нечітких множин можуть запропонувати ефективний апарат введення нелінійності в комбінуючі генератору псевдовипадкових чисел, заснованого на нечіткій логіці,

побудовані з урахуванням регістрів зсуву з лінійним зворотним зв'язком. При цьому нелінійна комбінуюча функція може бути побудована на основі аналізу статистичних властивостей виходів використовуваних регістрів зсуву з лінійним зворотним зв'язком із застосуванням лінгвістичних змінних та нечітких продукційних правил.

2.2. Модель генератора псевдовипадкових чисел, заснованого на нечіткій логіці

У магістерській роботі пропонується архітектура та модель генератора псевдовипадкових чисел, заснованого на нечіткій логіці. Запропонований генератор відноситься до класу генераторів, робота яких будується на комбінуванні виходів кількох регістрів зсуву з лінійним зворотним зв'язком із застосуванням нелінійної функції. Нелінійність реалізована на основі аналізу статистичних якостей виходів використовуваних зсувів із застосуванням лінгвістичних змінних та системи нечіткого логічного висновку. Архітектура запропонованого генератора псевдовипадкових чисел, заснованого на нечіткій логіці представлена на рисунку 2.2.

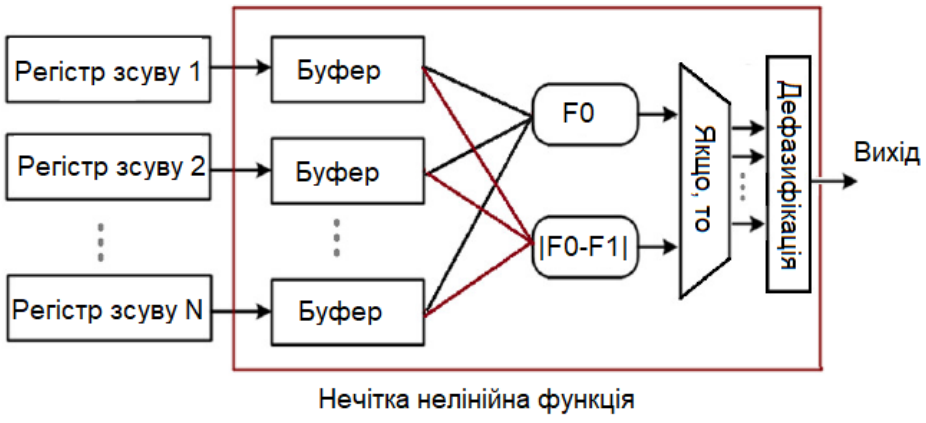

Рис. 2.2. Загальна архітектура генератора псевдовипадкових чисел, заснованого на нечіткій логіці

У цій архітектурі використовуються лінійні регістри зсуву зі зворотним зв'язком на формування початкових даних. Зібрані дані фазифікуються за допомогою введених лінгвістичних змінних та функцій належності. Далі реалізується нечіткий висновок за допомогою правил ЯКЩО-ТО. Наприкінці результат нечіткого висновку перетворюється на чітку форму за допомогою оператора дефазифікації.

Запропонована архітектура включає N регістрів зсуву з лінійним зворотним зв'язком, виходи яких надходять у пов'язані з ними N буферів розміром m біт. Для оцінки статистичних властивостей утримання буферів використовуються дві лінгвістичні змінні $(f_0, |f_1 - f_2|)$:

- кількість одиниць у буфері f_0;

- різниця між числом блоків f_1, що складаються з двох одиниць (0110), і кількості прогалин f_2, що складаються з двох нулів (1001) у буфері $|f_1 - f_2|$. Дані змінні були обрані на основі першого та другого постулатів Голомба, що є основними правилами для статистичних властивостей послідовностей.

Порівняння результатів здійснюється на основі нечітких ЯКЩО-ТО правил, визначальних регістрів зсуву з лінійним зворотним зв'язком, що має на даний момент часу найкраще значення на виході. Після дефазифікації за результатами порівняння визначається вихідне значення генератора псевдовипадкових чисел, заснованого на нечіткій логіці.

Формально запропонована модель генератора псевдовипадкових чисел, заснованого на нечіткій логіці представляється в наступному вигляді:

$$\langle \{P_i(x)\}_{i=1}^{n}, m, \{ЛЗ_{1,i}\}_{i=1}^{n}, \{ЛЗ_{2,i}\}_{i=1}^{n}, KB, A\rangle, \qquad (2.1)$$

де n − кількість використовуваних регістрів зсуву з лінійним зворотним зв'язком, $P_i(x)$ – примітивні характеристичні поліноми, m - розмір буфера (біт), $ЛЗ_{1,i}$ - лінгвістична змінна, що визначає функції належності для показника $f_0 0$ i-ого зсуву, $ЛЗ_{2,i}$ - лінгвістична змінна, що визначає функції

належності для показника $|f_1 - f_2|$ і-ого зсуву, KB – множина нечітких продукційних правил, що виконують аналіз результатів оцінки статистичних властивостей для кожного регістру зсуву з лінійним зворотним зв'язком, A – метод генерації псевдовипадкових чисел, заснований на формуванні генерації псевдовипадкових послідовностей.

Метод A дозволяє виконувати математичне моделювання процесу генарації псевдовипадкових чисел на базі розробленої моделі (2.1) та включає у собі такі основні етапи:

1) ініціалізація моделі генератора псевдовипадкових чисел, заснованого на нечіткій логіці із заданням параметрів моделі n, $P_i(x)$, m, $ЛЗ_{2,i}$, КВ;

2) ітеративна реалізація наступних кроків для формування псевдовипадкової послідовності:

2.1) формування виходів для кожного з регістрів зсуву з лінійним зворотним зв'язком та заповнення буферів;

2.2) оцінка статистичних властивостей буферів регістрів зсуву з лінійним зворотним зв'язком;

2.3) формування вихідного біта псевдовипадкових послідовностей.

3) формування псевдовипадкових чисел здійснюється на основі сформованої псевдовипадкової послідовності шляхом виділення з неї заданої кількості послідовних біт з подальшими зсувами вікна на один біт. Приклад формування 32-бітових послідовностей показано на наступному рисунку 2.3.

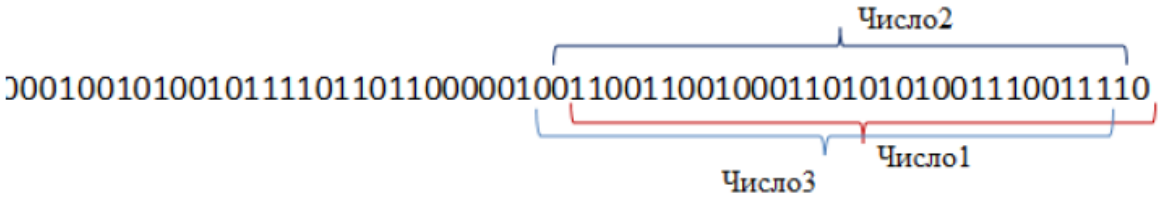

Рис. 2.3. Приклад формування 32-бітових послідовностей

Блок-схема алгоритму формування псевдовипадкових послідовностей (кроки 1-2 методу А) представлено на рисунку 2.4. Запропонований метод А дає можливість адаптивного налаштування генератора псевдовипадкових

чисел, заснованого на нечіткій логіці шляхом оцінки якості формованих псевдовипадкових послідовностей та подальшого підбору найбільш відповідних параметрів моделі.

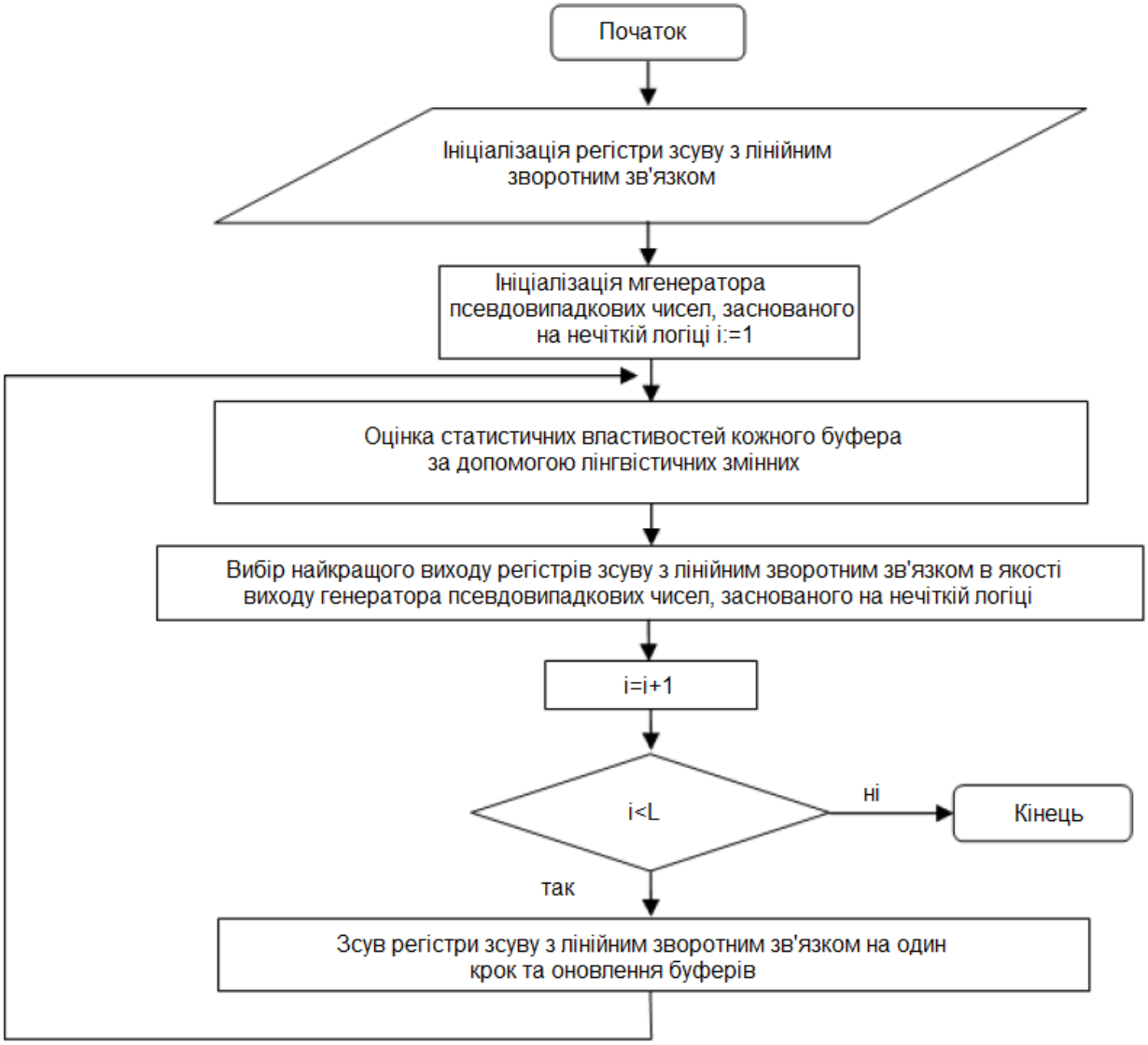

Рис. 2.4. Приклад формування 32-бітових послідовностей

В даному випадку якість буде значною мірою визначатися якістю псевдовипадкових послідовностей, що формуються всередині нього, яке має бути досліджено.

Лінгвістичні змінні псевдовипадкових послідовностей $ЛЗ_{1,i}$, $ЛЗ_{2,i}$ застосовуються для оцінки кожного з буферів, відповідних регістрів

зсуву з лінійним зворотним зв'язком, з метою оцінки їх статистичних властивостей.

Кожна лінгвістична змінна має кілька термів. Найбільш проста із запропонованих моделей буде включати два регістри зсуву з лінійним зворотним зв'язком і три терми в кожній з лінгвістичних змінних.

Застосування апарату теорії нечітких множин у вигляді лінгвістичних змінних та нечітких продукційних правил для завдання нелінійної функції в архітектурі генератора псевдовипадкових чисел, заснованого на нечіткій логіці та в моделі (2.1), дозволяє побудувати адаптивну структуру. Це дає можливість експерту на попередньому етапі визначити параметри моделі, що використовуються для аналізу статистичних властивостей виходів використовуваних регістри зсуву з лінійним зворотним зв'язком, і надалі виконати тюнінг даних параметрів, здійснивши пошук кращих із них, які забезпечують якість генерації псевдовипадкових послідовностей.

2.3. Параметри нелінійної функції комбінування регістрів зсуву з лінійним зворотним зв'язком, побудованих на основі нечіткої логіки

До другої групи параметрів запропонованої моделі генератора псевдовипадкових чисел, заснованого на нечіткій логіці, відносяться параметри нелінійної функції, побудованої на базі нечіткої системи логічного висновку: обсяг буфера m, кількість термів кожної з лінгвістичних змінних, тип функцій належності, база знань, що включає в себе сукупність ЯКЩО-ТО правил, конфігурація функцій належності лінгвістичних змінних, що використовуються.

1) Об’єм буфера. При подальшому дослідженні параметра моделі генератора " Об’єм буфера", досліджувалося 5 можливих значень: (8 біт, 16 біт, 24 біт, 32 біт, 64 біт).

Дане значення у побудованій моделі є єдиним всім регістрам зсуву з лінійним зворотним зв'язком. У найпростішій моделі генератора псевдовипадкових чисел, заснованого на нечіткій логіці це значення прийнято рівним 8 біт.

2) Кількість термів кожної з лінгвістичних змінних. Кількість термів кожної з лінгвістичних змінних безпосередньо впливає на статистичні властивості згенерованої псевдовипадкової послідовності. Надалі досліджувалося три можливі значення даного параметра - (3,5,7) для $ЛЗ_{1,i}$ і два значення – (3,5) для $ЛЗ_{2,i}$. У ході подальшого дослідження моделі генератора псевдовипадкових чисел, заснованого на нечіткій логіці потрібно дослідити якість формованих послідовностей при різних комбінаціях значення даного параметра {(3,3),(3,5),(5,3),(5,5),(7,3),(7,5)}.

3) Тип функцій належності. У роботі будемо досліджувати такі типи функцій належності з позиції їх впливу на якість формованих послідовностей: трикутну, трапецієподібну.

4) Конфігурація функцій належності використовуваних лінгвістичних змінних. У найпростішій моделі використовувалась наступна початкова конфігурація функцій належності лінгвістичних змінних, заданих як трикутні числа.

Для першої лінгвістичної змінної f_0 введено три терми {Low, Medium, High}, що використовуються для фазифікації вхідних даних:

- значення {Low} призначається, коли $f_0 \in \{0,1,2\}$;
- значення {Medium} призначається, коли $f_0 \in \{3,4,5\}$;
- значення {High} призначається, коли $f_0 \in \{6,7,8\}$;

Також друга нечітка змінна $|f_1 - f_2|$ має три прості функції належності, зв'язані з трьома нечіткими лінгвістичними термами {Excellent, Good, Bad}, що використовуються для фазифікації вхідних даних, наступним чином:

- Значення {Excellent} призначається, коли $|f_1 - f_2| \in \{0\}$;
- значення {Good} призначається, коли $|f_1 - f_2| \in \{1,2\}$;

- Значення {Bad} призначається, коли $|f_1 - f_2| \in \{3\}$;

У ході подальшого дослідження моделі генератора псевдовипадкових чисел, заснованого на нечіткій логіці необхідно здійснювати підбір параметрів використовуваних функцій належності для пошуку найкращого їх положення, що формує якісні послідовності.

Ця конфігурація використовуваних функцій належності значень лінгвістичних змінних f_0 і $|f_1 - f_2|$представлено на рисунку 2.5.

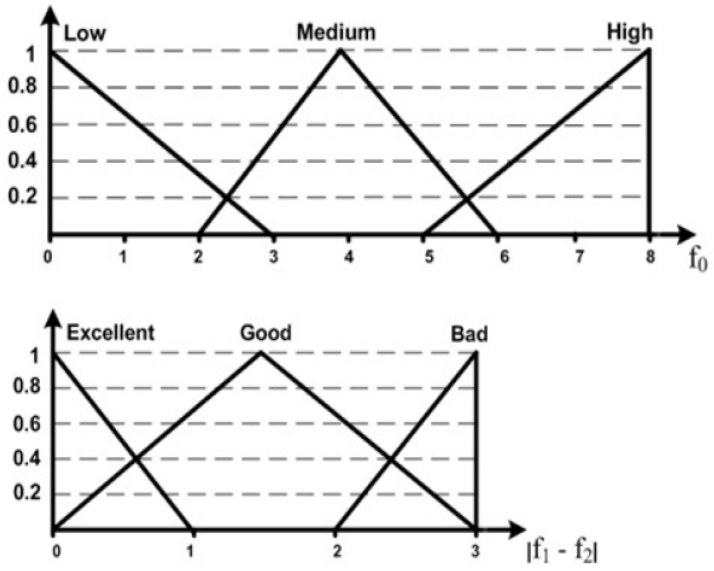

Рис. 2.5. Початкова конфігурація функцій належності f_0 і $|f_1 - f_2|$

Конфігурація функцій належності лінгвістичних, що використовуються досить сильно пов'язана з кількістю термів кожної з лінгвістичних змінних та обсягом буфера. Зі збільшенням кількості термів, зменшується кількість елементів f_0 і $|f_1 - f_2|$, зв'язаних із ним.

Наприклад, у тому випадку, коли кількість термів для лінгвістичної змінною f_0 дорівнює 5, а об'єм буфера дорівнює 8, можлива наступна конфігурація:

- значення {Very low} призначається, коли $f_0 \in \{0,1\}$;
- значення {Low} призначається, коли $f_0 \in \{2,3\}$;

- значення {Medium} призначається, коли $f_0 \in \{4\}$;

- значення {High} призначається, коли $f_0 \in \{5,6\}$;

- значення {Very high} призначається, коли $f_0 \in \{7,8\}$.

У тому випадку, коли кількість термів для лінгвістичної змінної f_0 дорівнює 3, а обсяг буфера дорівнює 8, можлива наступна конфігурація:

- значення {Low} призначається, коли $f_0 \in \{0,1,2\}$;

- значення {Medium} призначається, коли $f_0 \in \{3,4,5\}$;

- значення {High} призначається, коли $f_0 \in \{6,7,8\}$;.

5) База знань (сукупність нечітких продукційних правил). Сукупність нечітких продукційних правил визначає, який із буферів пов'язаних з використовуваними регістрами зсуву з лінійним зворотним зв'язком має найкращі оцінки щодо лінгвістичних змінних f_0 і $|f_1 - f_2|$.

У таблиці 2.1 представлена початкова сукупність використовуваних продукційних правил запропонованої моделі генератора псевдовипадкових чисел, заснованого на нечіткій логіці. При цьому розглядаються 3 терми лінгвістичних змінних.

Таблиця 2.1

Початкова сукупність нечітких продукційних правил

f_0 / $\lvert f_1 - f_2 \rvert$	Low	Medium	High
Excellent	Bad	Best	Bad
Good	Good	Good	Good
Bad	Bad	Good	Bad

При цьому можливе одержання 3 значень {Best, Good, Bad} з відповідними функціями належності. Отримане значення використовується для ухвалення рішення про те, який з регістрів зсуву має найкращі властивості.

Надалі цей набір правил коригувався з метою підвищення якості псевдовипадкових послідовностей, що формуються.

На заключному етапі здійснюється дефазифікація отриманого результату. Генератор псевдовипадкових чисел, заснованого на нечіткій логіці порівнює результати, отримані від різних регістрів зсуву з лінійним зворотним зв'язком та формує як вихід біт одного з них, що має найкращі статистичні показники.

Даний біт буде обраний як вихід моделі генератора в даний час. У таблиці 2.2 подано правила вибору виходу регістрів зсуву з лінійним зворотним зв'язком, які у найпростішому варіанті моделі генератора псевдовипадкових чисел.

Таблиця 2.2

Правила вибору виходу

Рег1 / Рег2	Best	Good	Bad
Best	Біт1	Біт2	Біт2
Good	Біт1	Біт1	Біт2
Bad	Біт1	Біт1	Біт1

У випадку, коли досліджувані регістри зсуву з лінійним зворотним зв'язком мають однакові статистичні характеристики, як результуючий вибирається з меншим степенем характеристичного полінома.

Для запобігання появі на виході замкнутого циклу, генеруючого псевдовипадкові послідовності, що складається з 0, ініціалізація регістрів зсуву з лінійним зворотним зв'язком здійснюється випадковим чином.

Висновки до другого розділу

1. Розроблено архітектуру та модель запропонованого генератора псевдовипадкових чисел, заснованого на нечіткій логіці. Введено нелінійну функцію, що здійснює комбінування регістрів зсуву з лінійним зворотним зв’язком в даному генераторі, заснована на введенні лінгвістичних змінних та нечітких продукційних правил.

2. Проаналізовано параметри запропонованої моделі, які поділені на дві групи: параметри використовуваних регістрів зсуву з лінійним зворотним зв'язком, а також параметри побудованої нелінійної функції з урахуванням нечіткої логіки. До першої групи віднесено один основний параметр – тип характеристичних примітивних поліномів використовуваних регістрами зсуву з лінійним зворотним зв'язком, а також степінь даних примітивних поліномів.

РОЗДІЛ 3
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ГЕНЕРАТОРА ПСЕВДОВИПАДКОВИХ ЧИСЕЛ НА ОСНОВІ НЕЧІТКОЇ ЛОГІКИ

3.1. Загальна архітектура системи

Для реалізації розробленої моделі генератора псевдовипадкових чисел, заснованого на нечіткій логіці та дослідження параметрів даної моделі, розроблено комплекс програм генерації псевдовипадкових послідовностей. Структура розробленого комплексу програм представлено на рисунку 3.1.

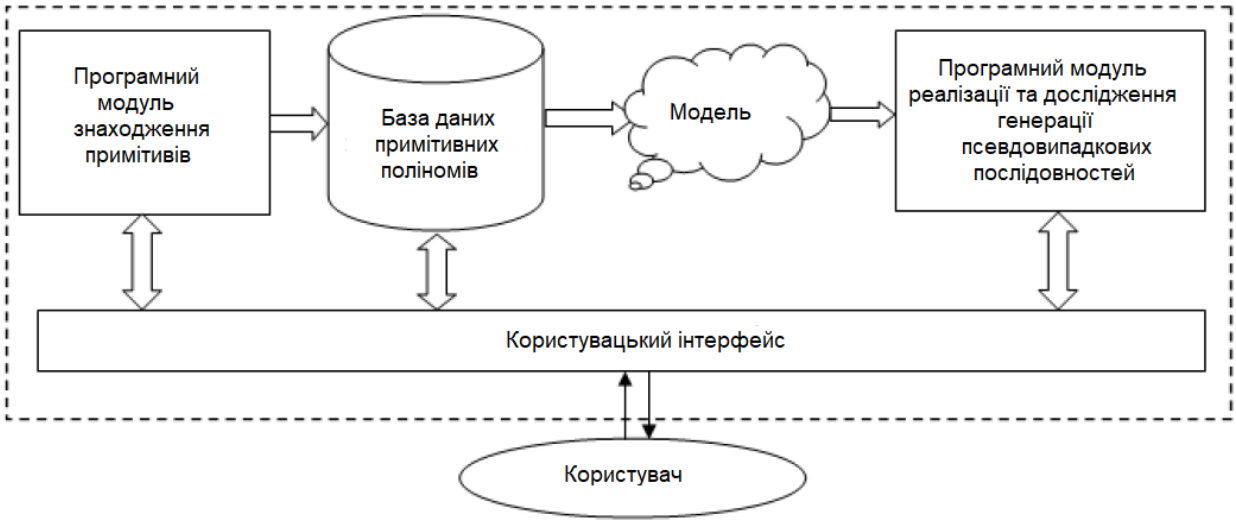

Рис. 3.1. Структура розробленого програмного комплексу

Даний комплекс програм дозволяє формувати модель генератора з вибраними параметрами, формувати примітивні характеристичні поліноми для використовуваних регістрів зсуву з лінійним зворотни зв'язком, а також дослідити параметри моделі генератора з погляду якості формованих у своїй генерації псевдовипадкових послідовностей. Оцінка якості генерації псевдовипадкових послідовностей здійснюється за допомогою тестів NIST, графічних тестів, методу Монте Карло. Для розробки комплексу програм

застосовувалися пакети Matlab та Mathematica (для формування примітивних поліномів).

Головне вікно програми містить дві основні панелі. Перша призначена для генерування псевдовипадкових послідовностей за допомогою генератора псевдовипадкових чисел, заснованого на нечіткій логіці з вибраними параметрами (Generate). Друга панель служить для оцінки якості генерації (Randomness tests). На рисунку 3.2 представлено головне вікно розробленого комплексу програм.

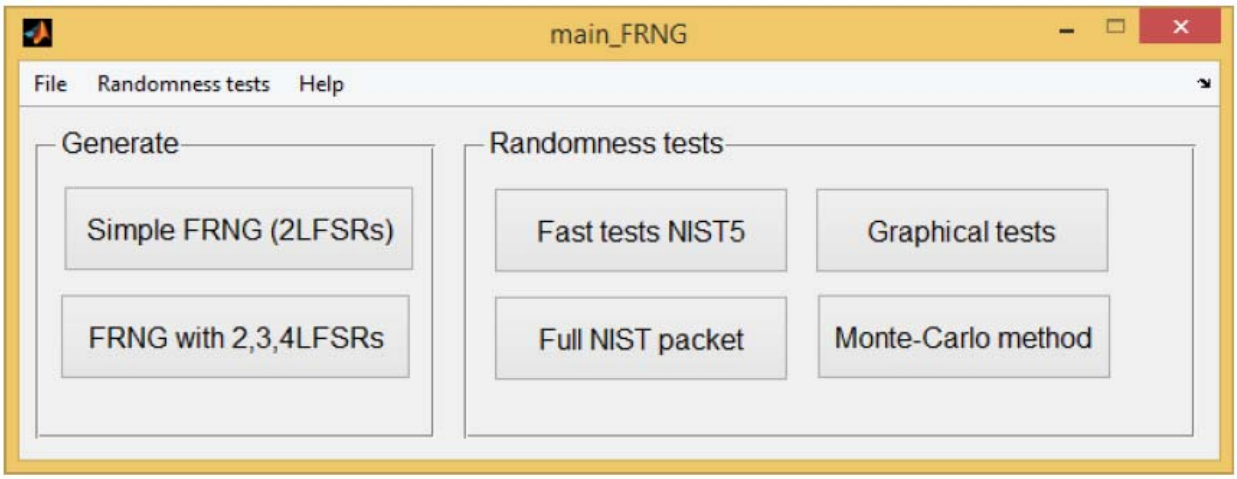

Рис. 3.2. Головне вікно розробленого програмного комплексу

На рисунку 3.3. представлено діалогове вікно для задання різних параметрів моделі генератора псевдовипадкових чисел, заснованого на нечіткій логіці. Можливе задання всіх параметрів моделі.

У ході генерації псевдовипадкових послідовностей комплекс програм показує час, який витрачається на генерацію, забезпечує можливість збереження згенерованої послідовності, дає інформацію про збаласованість вибраної моделі.

Для ініціалізації регістрів зсуву з лінійним зворотним зв'язком використовується функція Matlab (Randi) або ця ініціалізація здійснюється користувачем.

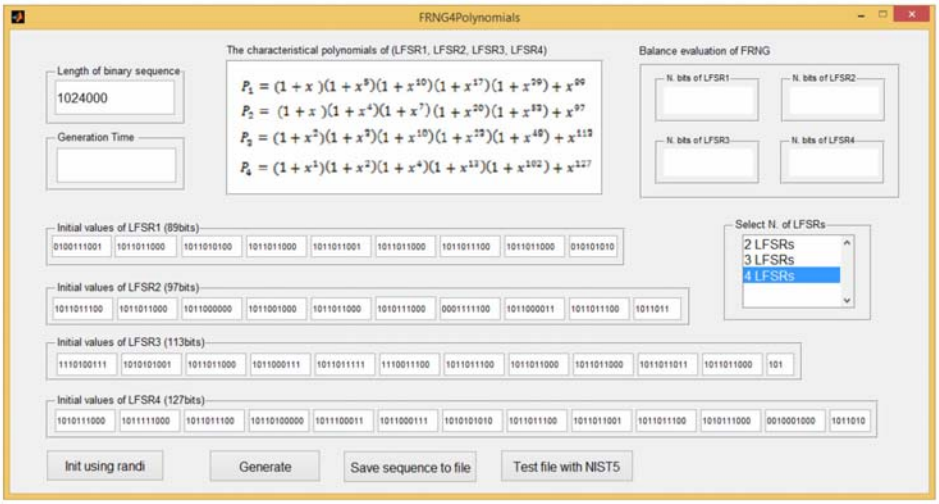

Рис. 3.3. Діалогове вікно для встановлення параметрів моделі

На рисунку 3.4 представлено діалогове вікно, що використовується для оцінки статистичних властивостей генератора псевдовипадкових чисел, заснованого на нечіткій логіці за допомогою 5 відібраних найважливіших тестів NIST.

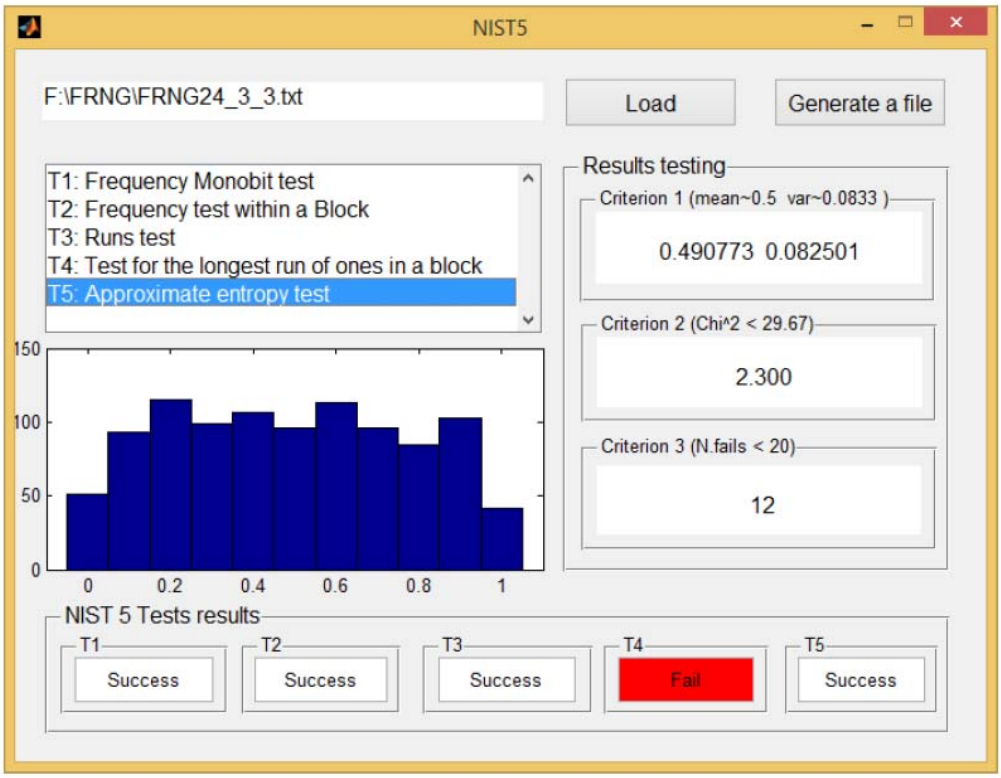

Рис. 3.4. Вікно оцінки статистичних властивостей за допомогою п'яти найважливіших тестів пакету NIST

Комплекс програм показує результати оцінки якості за трьома раніше розглянутими критеріями, гістограму обчислених P-values та результати, що свідчать про проходження/не проходження генератора псевдовипадкових чисел, заснованого на нечіткій логіці конкретного тесту.

На рисунку 3.5 представлено діалогове вікно оцінки генератора псевдовипадкових чисел, заснованого на нечіткій логіці за допомогою повного набору тестів NIST.

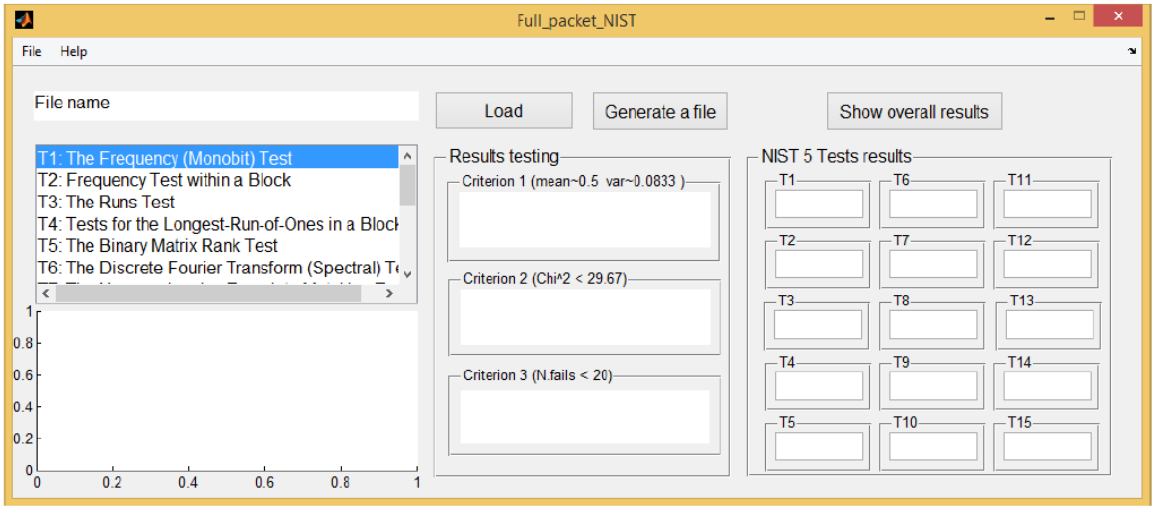

Рис. 3.5. Діалогове вікно оцінки генератора псевдовипадкових чисел, заснованого на нечіткій логіці за допомогою повного набору тестів NIST

На рисунку 3.6 представлено діалогове вікно оцінки генератора псевдовипадкових чисел, заснованого на нечіткій логіці за допомогою набору графічних тестів.

На рисунку 3.7 представлено діалогове вікно оцінки генератора псевдовипадкових чисел, заснованого на нечіткій логіці за допомогою методу Монте-Карло.

Надається можливість обчислення одного з інтегралів та завдання параметрів тестування. Даний модуль здійснюється також порівняння результатів, отриманих з допомогою генератора псевдовипадкових чисел,

заснованого на нечіткій логіці з результатами, отриманими функцією Randi (Matlab).

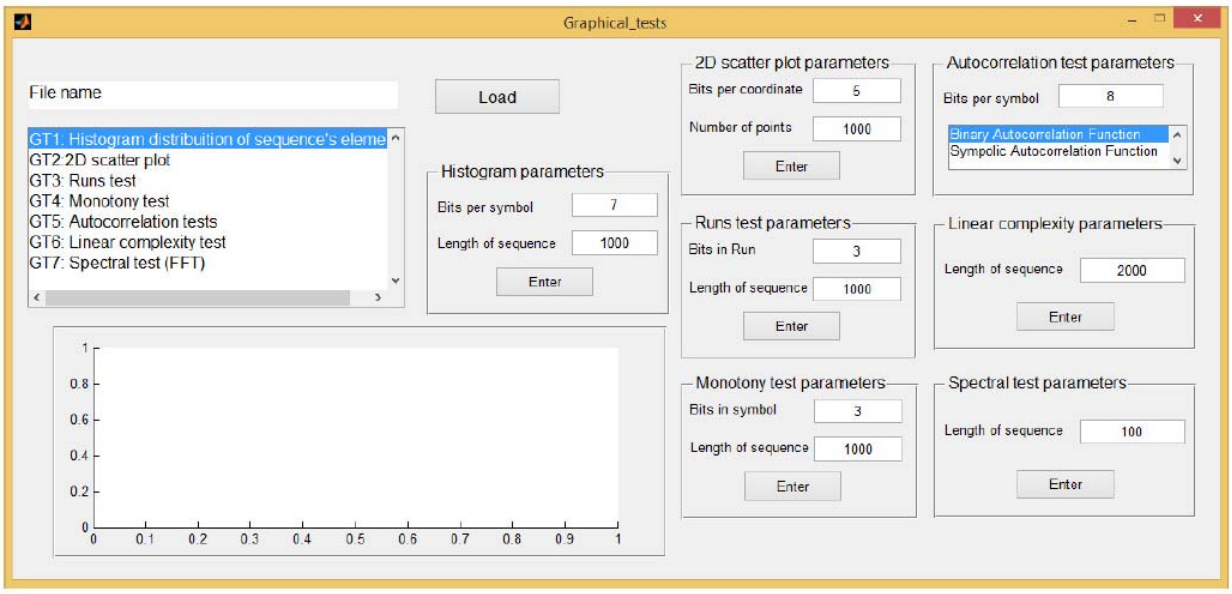

Рис. 3.6. Діалогове вікно оцінки генератора псевдовипадкових чисел, заснованого на нечіткій логіці за допомогою повного набору графічних тестів

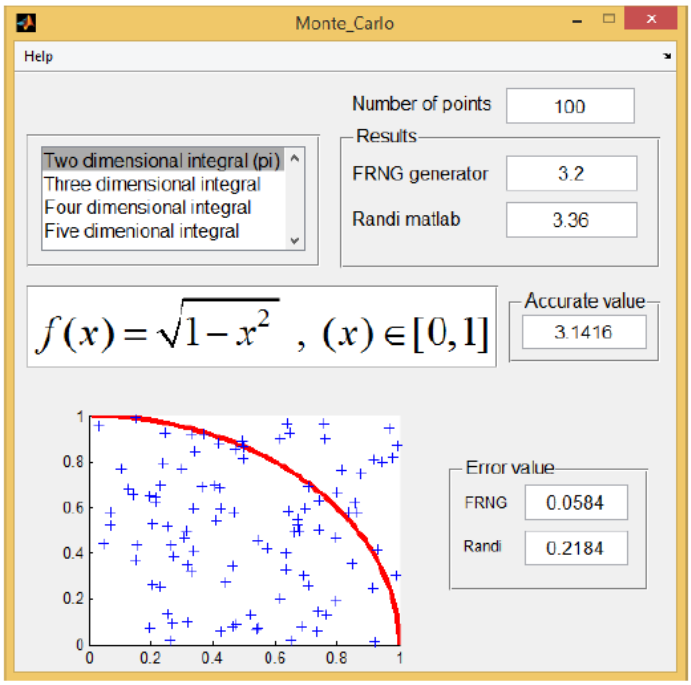

Рис. 3.7. Діалогове вікно оцінки генератора псевдовипадкових чисел, заснованого на нечіткій логіці за допомогою методу Монте-Карло

3.2. Дослідження параметрів моделі генератора псевдовипадкових чисел, заснованого на нечіткій логіці

Параметри моделі генератора псевдовипадкових чисел, заснованого на нечіткій логіці поділені на дві групи: параметри використовуваних регістрів зсуву, а також параметри побудованої нелінійної функції на основі нечіткої логіки.

Якість роботи генератора псевдовипадкових чисел, заснованого на нечіткій логіці безпосередньо залежить від періоду генерації, що формуються характеристичними примітивними поліномами, що використовують регістри зсуву, від вибраних характеристичних поліномів. Обираються примітивні характеристичні поліноми, які повинні забезпечувати максимальний період формованої генерації.

На початку, для демонстрації можливостей генератора псевдовипадкових чисел, заснованого на нечіткій логіці було побудовано модель, що включає наступні поліноми:

$$P_1 = 1 + x^{83} + x^{84} + x^{86} + x^{89}, T_1 = 2^{89} - 1 = 6.1897 \times 10^{26}$$
$$P_2 = 1 + x^{91} + x^{97}, \qquad T_2 = 2^{97} - 1 = 1.5846 \times 10^{29}$$

Дані поліноми дозволяють забезпечити період $T \approx 1055$. Ця модель також включає 2 буфери розміром 8 біт, лінгвістичні змінні $ЛЗ_{1,i}$, $ЛЗ_{2,i}$ з трьома термами, формалізованими на рисунку 3.5, а також множиною нечітких продукційних правил, заданих у вигляді таблиці. Результати тестування даного генератора за допомогою тестів NIST представлені у таблиці 3.1.

Таким чином, генерації псевдовипадкових послідовностей, сформованого генератора псевдовипадкових чисел, заснованого на нечіткій логіці, пройшла всі тести NIST на випадковість. Однак, нам необхідно

докладніше досліджувати та здійснити вибір найбільш відповідних примітивних характеристичних поліномів для використання в генерації.

Таблиця 3.1.

Результати тестування генератора псевдовипадкових чисел, заснованого на нечіткій логіці за допомогою пакету статистичних тестів NIST

Тести пакету NIST	Параметри	P-value	Результат
$T1$	$n = 10^6$	0.424	успіх
$T2$	$n = 10^6, M = 10^5, N = 10$	0.850	успіх
$T3$	*n=1000*	0.790	успіх
$T5$	*n=750000, M=*10^4*, k=6, N=75*	0.480	успіх
$T5$	*M=Q=3, N=100*	0.209	успіх
$T6$	*n=1000*	0.977	успіх
$T8$	*M=100,m=10,N=10, B=0000000001*	0.620	успіх
$T8$	*M=100,m=10,N=10, B=0000000001*	0.298	успіх
$T9$	*n=387840, L=6, Q=640, K=64000*	0.129	успіх
$T10$	*n=*10^6*, M=1000, N=1000*	0.57	успіх
$T11$	*n=*10^4*, m=2*	0.534	успіх
$T12$	*n=1000, m=3*	0.840	успіх
$T13$	*n=*10^4	0.815	успіх

Основними вимогами до характеристичних поліномів, що вибираються є:

1) вибрані характеристичні поліноми генератора псевдовипадкових чисел, заснованого на нечіткій логіці повинні мати великий ступінь, достатній для задоволення постулатам, а також повинні забезпечувати хороші статистичні властивості для згенерованої послідовності;

2) характеристичні поліноми мають забезпечувати невеликі витрати на апаратну реалізацію, що визначаються кількістю задіяних елементів XOR для реалізації поліномів;

3) примітивні характеристичні поліноми повинні мати високі степені і велику вага Хеммінга для забезпечення високої дифузійної ємності.

Таблиця 3.2 містить список усіх ефективних ненаведених поліномів над $GF(2)$ зі степенем $n = 11$ та вагою Хеммінга $W_H = 5$.

Таблиця 3.2

Список ефективних поліномів зі степенем $n = 11$ та вагою Хеммінга $W_H = 5$

b_1	b_2	$f(x)$
1	5	$(1+x^1)(1+x^5)+x^{11}$
1	6	$(1+x^1)(1+x^6)+x^{11}$
1	7	$(1+x^1)(1+x^7)+x^{11}$
1	8	$(1+x^1)(1+x^8)+x^{11}$
2	3	$(1+x^2)(1+x^3)+x^{11}$
2	6	$(1+x^2)(1+x^6)+x^{11}$
2	7	$(1+x^2)(1+x^7)+x^{11}$
3	5	$(1+x^3)(1+x^5)+x^{11}$
3	6	$(1+x^3)(1+x^6)+x^{11}$
3	7	$(1+x^3)(1+x^7)+x^{11}$

Оскільки значення n мале, кількість таких поліномів невелика. Однак, у загальному випадку для великих n цей список вийде більшим. Його довжина пропорційна необхідного степеня та кількості параметрів (m).

Варто відзначити, що над $GF(2)$ поліноми, що не наводяться, не обов'язково є примітивними. Тому нам необхідно перевіряти умови примітивності.

Таблиця 3.3 містить окремі значення m, відповідні коефіцієнти W_H та відповідні діапазони степенів, які слідують використовувати із ними. З погляду безпеки, вага Хеммінга має бути $W_H > 10$.

Таблиця 3.3

Значення m, W_H та діапазони ступенів n

m	W_H	***n***
4	17	35-66
5	33	67-130
6	65	131-258
7	129	259-514
8	257	515-1026
9	513	1027-2050

Другим найважливішим кроком при виборі поліномів генератора псевдовипадкових чисел, заснованого на нечіткій логіці є їх перевірка на примітивність. Однак, пошук примітивних поліномів для використання в генерації псевдовипадкових послідовностей запропонованої моделі генератора псевдовипадкових чисел, заснованого на нечіткій логіці у значній степені ускладнюється обчислювальною трудомісткістю їх перевірки примітивність, особливо у разі використання великих степенів n.

З метою зменшення обчислювальної складності розв'язання даної задачі, у роботі було розроблено ефективний чисельний метод знаходження характеристичних примітивних поліномів для генерації псевдовипадкових послідовностей, побудованих на регістрах зсуву з лінійним зворотним зв'язком.

Даний метод включає наступні етапи:

Етап 1. Знаходження поліномів виду.

Етап 2. Перевірка знайдених поліномів на примітивність. При цьому здійснюється перевірка наступних умов, де f - досліджуваний поліном:

1) $f\,(0)\; =\; f\,(1)\; = 1$

2) $min\left\{k: f|x^{2^k} - x\right\} = n$

3) для всіх простих чисел $p|2^n - 1, x^{\frac{2^{n-1}}{p}} \neq 1 mod f$

Усі поліноми задовольняють першу вимогу. Найбільш обчислювально складним етапом є перевірка третьої вимоги. Загалом вона вимагає факторизації числа $2^n - 1$. Зменшення обчислювальної складності при виконанні цієї перевірки здійснювалося на основі результатів, отриманих у проекті Каннінгема (Cunningham project), в якому отримано перелік простих чисел від 1 до $2^{1200} - 1$. Застосування даної процедури дозволило знизити експоненційну складність факторизації до складності процедури $O(F * n^2)$ 2, де F - кількість факторів.

Для пошуку примітивних поліномів за допомогою розробленого чисельного методу було розроблено алгоритм і програмний код, що реалізує його модуль у середовищі Mathematica. Ця програма включає в себе реалізацію даних двох етапів. Реалізація першого етапу представлена на рисунку 3.9 в середовищі Mathematica

$$testpoly5[b_1\_, b_2\_, b_3\_, b_4\_, b_5\_, n\_] :=$$
$$If[(b_1 > 0)$$
$$\&\&(b_2 > b_1)$$
$$\&\&(b_3 > (b_2 + b_1))$$
$$\&\&(b_4 > (b_1 + b_2 + b_3))$$
$$\&\&(b_5 > (b_1 + b_2 + b_3 + b_4))$$
$$\&\&(n > (b_1 + b_2 + b_3 + b_4 + b_5)), True, False];$$

$$Poly5\left[b_{1\_}, b_{2\_}, b_{3\_}, b_{4\_}, b_{5\_}, n\_\right] := Expand[(1 + x^{b1})(1 + x^{b2})(1 + x^{b3})(1 + x^{b4})(1 + x^{b5})];$$

$$Poly5[1,5,10,17,39,89] = 1 + x + x^5$$
$$+ x^6 + x^{10} + x^{11} + x^{15} + x^{16}$$
$$+ x^{17} + x^{18} + x^{22} + x^{23} + x^{27}$$
$$+ x^{28} + x^{32} + x^{33} + x^{39} + x^{40}$$
$$+ x^{44} + x^{45} + x^{49} + x^{50} + x^{54}$$
$$+ x^{55} + x^{56} + x^{57} + x^{61} + x^{62}$$
$$+ x^{66} + x^{67} + x^{71} + x^{72} + x^{89}$$

Рис. 3.9. Скрипт реалізації першого етапу в середовищі Mathematica

Далі в нашій роботі коротко означатимемо поліном за допомогою його параметрів та степеня ($\boldsymbol{b_1, b_2, \dots, b_m, n}$).

Вибір лише ненаведених поліномів з отриманих результатів виконується в середовищі Mathematica за допомогою функції $\boldsymbol{IrreduciblePolynomialQ[\,polynomial, modulus \to 2]}$.

Далі здійснюється отримання списку всіх ефективних ненаведених поліномів над $\boldsymbol{GF(2)}$. Для $\boldsymbol{m = 5, n = 67}$.

```
coeffpoly5[n_] := {i = 0;
    Do[If[testpoly5[c1, c2, c3, c4, c5, n] == True,
    If[IrreduciblePolynomialQ[Poly5[c1, c2, c3, c4, c5, n], Modulus → 2] == True,
    {i++; Print[c1, "", c2, "", c3, "", c4, "", c5, "", n]}, Null], Null],
    {c1, 1, Quotient[n, 32] + 1},
    {c2, c1 + 1, 1 + Quotient[n, 16]},
    {c3, c2 + c1 + 1, 1 + Quotient[n, 8]},
    {c4, c3 + c2 + c1 + 1, 1 + Quotient[n, 4]},
    {c5, c4 + c3 + c2 + c1 + 1, n - (c4 + c3 + c2 + c1 + 1)}];}

Table[coeffpoly5[n], {n, 67, 67}];}]
```

Рис. 3.10. Скрипт реалізації процедури отримання ефективних поліномів

У таблиці 3.4 показано частину отриманого списку поліномів. В данному прикладі довжина повного списку дорівнює 146, в таблиці показані тільки останні 10 із них.

Розроблена програма була використана для отримання списків усіх ефективних поліномів з простими степенями n від 1 до 1200. Дані поліноми для використання в регістрах зсуву з лінійним зворотним зв'язком необхідно перевірити на примітивність.

Для реалізації другого етапу програма виконує перевірки умов:

1) 2a) 2b) 3)

Перевірка умов 2a) 2b) здійснюється за допомогою середовища Mathematica наступним чином:

2a) $IrreducibleFactorExponent[f\_] := PolynomialGCD[f, \partial_x f, Modulus \to 2]$

2b) $Do[Print[PolynomialGCD[f, x^{2^k} - x, Modulus \to 2]], \{k, 2, 12\}]$

Таблиця 3.4.

Частина отриманих ефективних поліномів, що не наводяться для $m = 5, n = 67$

$(b_1, b_2, b_3, b_4, b_5, n)$
2,3,8,15,33,67
2,3,8,16,32,67
2,3,8,17,31,67
2,3,9,15,32,67
2,4,7,15,29,67
2,4,7,15,37,67
2,4,7,16,36,67
2,4,8,15,35,67
2,4,8,16,31,67
2,4,8,16,34,67

Найбільш обчислювально складним є перевірка третьої вимоги. Як було сказано раніше, залучаючи проект Каннінгема (Cunningham project), нам вдалося знизити експоненційну складність факторизації до складності процедури $O(F \cdot n^2)$, де F - кількість факторів.

Для скорочення складності алгоритму ми також вибирали як n просте число. Було написано програму для отримання списку простих степенів n і всіх відповідних факторів числа $(2^n - 1)$ для n від 19 до 1193. Для перевірки третьої умови для певного полінома (з простим степенем) наша програма повинна брати відповідні фактори з файлу, і надалі здійснює всі необхідні перевірки.

Наприклад, наведемо приклад перевірки третьої умови в середовищі Mathematica для прикладу $(n = 89, f = (1,5,10,17,39,89))$. При застосуванні функції: $FactorInteger$ виходить: {{618970019642690137449562111,1}}

Це означає, що нам необхідно здійснити лише одну перевірку: $x^1 \neq 1, mod\ f$.

Це робиться за допомогою функції (PolynomialMod [f, {p,2}]). В результаті виконання цієї функції виходить: $PolynomialRemainder$. Можна, можливо сказати що досліджуваний поліном f задовольняє умову тесту на примітивність.

У результаті можна зробити висновок що досліджуваний поліном $f = (1,5,10,17,39,89)$ є примітивним. Програмна реалізація третьої умови в середовищі Mathematica виконані таким чином, рисунок 3.11.

```
IsPrimativePolynomial[p_,n_]:=
Module[{r, factors}, r = 2^n - 1;
factors = FactorInteger[r]; nof = Length[factors];
res = True;
For[i = 1, i ≤ nof, ++i,
      {px = factors[[i]][[1]]; qx = r / px;
      ax = PolynomialMod[x^qx - 1, {p, 2}];
      If[ax != 0, , res = False; Print[px, "doesn't pass the test"];
      Break[]];
      Print[px, "passed"];
      }];
If[res == True, Print["the polynomial is primitive"],
Print["the polynomial is not primitive"]]]
```

Рис. 3.11. Скрипт реалізації процедури отримання ефективних поліномів

Ряд прикладів примітивних поліномів, отриманих за допомогою розробленої програми у середовищі Matematica, наведено у таблиці 3.5.

Після проведеного дослідження як примітивні характеристики поліномів регістри зсуву з лінійним зворотним зв'язком розробленої моделі генератора псевдовипадкових чисел, заснованого на нечіткій логіці були обрані такі поліноми:

$$P_1(x) = (1+x\ )(1+x^5)(1+x^{10})(1+x^{17})(1+x^{39}) + x^{89}$$
$$P_2(x) = (1+x\ )(1+x^4)(1+x^7)(1+x^{20})(1+x^{53}) + x^{97}$$

Дані поліноми мають степені 89,97 і задовольняють усім вищепереліченим вимогам до даних поліномів.

Таблиця 3.5.

Приклад роботи запропонованого алгоритму

n	***m***	$W_H = 2^m + 1$	*f(x)*
37	4	17	(3, 4, 9, 17, 37)
61	4	17	(2, 3, 15, 39, 61)
67	5	33	(1, 2, 4, 9, 41, 67)
89	5	33	(1, 5, 10, 17, 39, 89)
149	6	65	(3, 5, 9, 19, 37, 75, 149)
251	6	65	(1, 2, 4, 14, 23, 141, 251)
313	7	129	(2, 3, 6, 12, 24, 65, 169, 313)
509	7	129	(3, 6, 13, 27, 54, 119, 266, 509)
983	8	257	(2, 5, 12, 28, 58, 118, 243, 512, 983)

У даному розділі було проведено оцінку якості формованих для різних значень досліджуваних параметрів моделі генератора псевдовипадкових чисел, заснованого на нечіткій логіці з метою вибору найбільш переважних значень. Оцінка якості генератора здійснювалася за допомогою п'яти найважливіших вибраних статистичних тестів пакету NIST із застосуванням трьох критеріїв вибору з оцінкою значень Pvalues.

Було досліджено такі параметри: обсяг буфера m, кількість термів кожної з лінгвістичних змінних, база знань, що включає в себе сукупність ЯКЩО правил, конфігурація функцій власності використовуваних лінгвістичних змінних, типів функцій належності.

Для того, щоб інтерпретувати отримані результати та прийняти рішення про успішне проходження статистичного тесту, використовувалися три основні статистичні критерії, що дозволяють оцінити ступінь відповідності рівномірному розподілу:

1) критерій середнього значення та дисперсії.

2) відношення кількості невдалих підпослідовностей до їх загального кількості.

2) Критерій хі-квадрат.

Для порівняння статистичних властивостей отриманих послідовностей було проведено чисельні експерименти. У кожному експерименті генерувалося множину послідовностей довжиною 1024000 біт, отриманих за зміни значень параметрів генератора. Дані послідовності поділялися на 1000 підпослідовностей завдовжки 1024 біта.

В результаті їх тестування за допомогою вибраних п'яти тестів пакету NIST виходить 1000 $P-values$ для кожного тесту. Результати тестування порівнювалися між собою для знаходження найкращих значень досліджуваних параметрів генератора псевдовипадкових чисел, заснованого на нечіткій логіці.

- Сукупність ЯКЩО, ТО правил. Цей параметр було досліджено паралельно з таким параметром, як об’єм буфера. З метою спрощення викладу, представимо отримані результати лише для обсягу буфера = 8.

При цьому, найкраща сукупність ЯКЩО, ТО правил визначалася для різних значень сукупності параметрів $(f_0, |f_1 - f_2|\ ) = \{(3,3), (3,5), (5,3), (5,5),\ (7,3), (7,5)\}$. Розглянемо різні комбінації значень даних параметрів та відповідні їм найкращі сукупності ЯКЩО, ТО правил.

Таблиця 3.6

Найкраща сукупність, ЯКЩО, ТО правил

f_0 / $\lvert f_1 - f_2 \rvert$	Low {0,1,2}	Medium {3,4,5}	High {6,7,8}
Excellent {0}	Bad	Best	Bad
Good {1}	Bad	Good	Bad
Bad {2}	Bad	Bad	Bad

Таблиця 3.7

Найкраща сукупність ЯКЩО, ТО правил

(Обсяг буфера =8,$(f_0, |f_1 - f_2| = (5,3)$

f_0 / $\lvert f_1 - f_2 \rvert$	**VeryLow {0,1}**	**Low {2}**	**Medium {3,4,5}**	**High {6}**	**Very High {7,8}**
Excellent {0}	Bad	Bad	Best	Bad	Bad
Good {1}	X	Bad	Good	Bad	X
Bad {2}	X	X	Good	X	X

Таблиця 3.8

Найкраща сукупність ЯКЩО, ТО правил

(Обсяг буфера =8,$(f_0, |f_1 - f_2| = (7,3)$)

f_0 / $\lvert f_1 - f_2 \rvert$	**Very Low {0,1}**	**Low {2}**	**L-M {3}**	**Medium {4}**	**H-M {5}**	**High {6}**	**Very High {7,8}**
Excellent {0}	Bad	Bad	Good	Best	Good	Bad	Bad
Good {1}	X	Bad	Good	Good	Good	Bad	X
Bad {2}	X	X	Bad	Good	Bad	X	X

З точки зору якості формованих генерації варіант, представлений у таблиці 4.9. є найкращим. Кількість термів кожної з лінгвістичних змінних $(f_0, |f_1 - f_2|$:

Результати обчислювальних експериментів з генератора псевдовипадкових чисел, заснованого на нечіткій логіці щодо даних параметрів, подані у таблиці 3.9.

Знак «+» у цій таблиці говорить про успішне проходження генератора псевдовипадкових чисел, заснованого на нечіткій логіці відповідного тесту з використанням критерію середнього значення та дисперсії. 7 моделей з параметрами, що пройшли перший критерій відбору, далі тестувалися за допомогою критерію хі-квадрат.

Таблиця 3.9

Результати обчислювальних експериментів з генератора псевдовипадкових чисел, заснованого на нечіткій логіці при зміні кількості термів та обсягу буфера (критерій середнього значення та дисперсії)

Кількість термів	Об'єм буфера	T1	T2	T3	T5	T5
3,3	8	+	–	–	+	–
	16	+	+	+	+	+
	24	+	+	+	+	+
	32	+	+	+	+	+
	64	+	+	+	+	+
5,3	8	+	–	–	+	–
	16	+	–	+	+	+
	24	+	–	+	+	+
	32	–	–	+	+	+
	64	+	–	+	+	+
7,3	8	+	–	–	+	–
	16	+	–	+	+	+
	24	+	–	+	+	+
	32	+	–	+	+	+
	64	+	–	+	+	+
3,5	16	+	–	+	+	+
	24	+	–	+	+	+
	32	+	+	+	+	+
	64	+	+	+	+	+
5,5	16	+	–	+	+	+
	24	+	–	+	+	+
	32	+	–	+	+	+
	64	+	+	+	+	+
7,5	16	+	–	+	+	+
	24	+	–	+	+	+
	32	+	–	+	+	+
	64	+	–	+	+	+

Типи функцій належності. Існує багато типових форм кривих для задання функцій належності. Найбільшого поширення практично отримали: кусочно-лінійні, Z –подібні та S –подібні, П –подібні функції. З метою знаходження найкращого варіанту функцій належності нами розглядалися: "трикутна функція ", "трапецієподібна функція", та "функція належності щільності нормального розподілу".

В результаті дослідження було виявлено, що трапецієподібна функція є найкращим варіантом. Даний параметр не має великого впливу на якість вихідної послідовності.

Конфігурація лінгвістичних змінних. Зі збільшенням кількості термів лінгвістичної змінної зменшується кількість елементів підмножини. Наприклад, коли кількість термів першої нечіткої змінної f_0 дорівнює 5 і обсяг буфера дорівнює 8, як приклад можуть бути обрані наступні 5 підмножин, відповідних 5 терм: $\{very\ low \rightarrow \{0,1\}, low \rightarrow \{2,3\}, medium \rightarrow \{4\}, high \rightarrow \{5,6\}, very\ high \rightarrow \{7,8\}\}$. Коли кількість термів дорівнює 3 ми можемо вибрати 3 підмножини: $\{low \rightarrow \{0,1,2\}, medium \rightarrow \{3,4,5\}, high \rightarrow \{6,7,8\}\}$.

Початкові конфігурації лінгвістичних змінних генератора псевдовипадкових чисел, заснованого на нечіткій логіці представлені на рисунку 3.12.

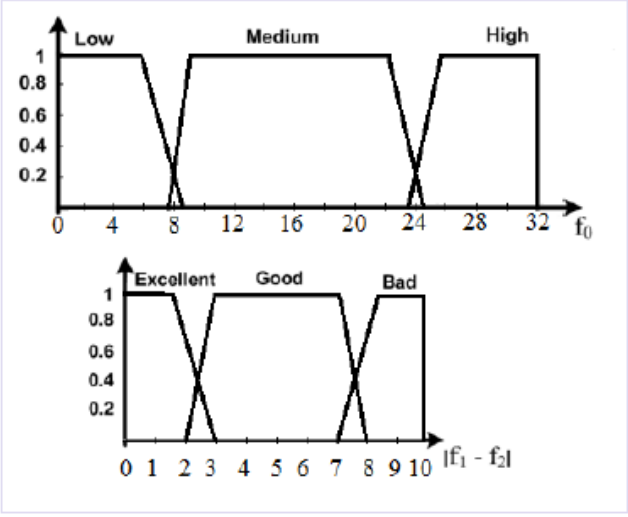

Рис. 3.13. Початкові конфігурацій функцій належності

Рисунок 3.14 (а,б) показує функції належності лінгвістичних змінних $(f_0, |f_1 - f_2|)$ для яких отримано збалансований вихід генератора.

(а)

(б)

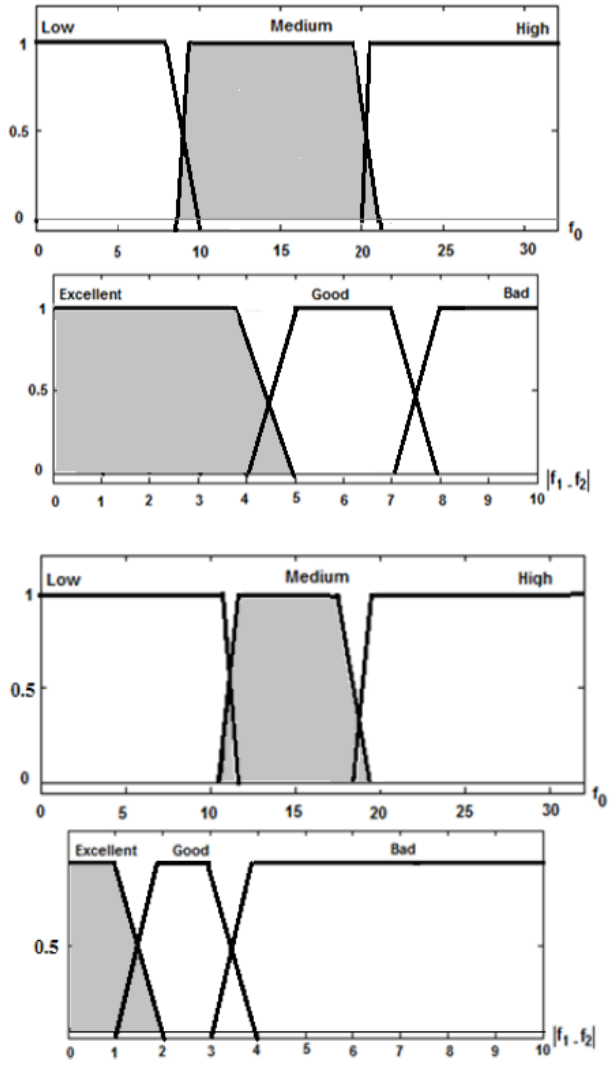

Рис. 3.14. Конфігурацій функцій належності для збалансованого генератора

Таким чином, ми можемо зробити висновок, що запропонований ггенератора псевдовипадкових чисел, заснованого на нечіткій логіці, став досить стійким проти відомих алгебраїчних атак.

3.3. Експериментальні дослідження та тестування запропонованого генератора

За допомогою даних тестів статистичні властивості згенерованих псевдовипадкових послідовностей відображаються у вигляді графічних

залежностей, на вигляд яких можна робити висновок про випадковість генерованих послідовностей. Набір графічних тестів включає у собі 7 тестів.

Для реалізації графічних тестів було використано розроблений комплекс програм. Результати дослідження за допомогою аналізу гістограми розподілу елементів послідовності представлені на рисунку 3.15.

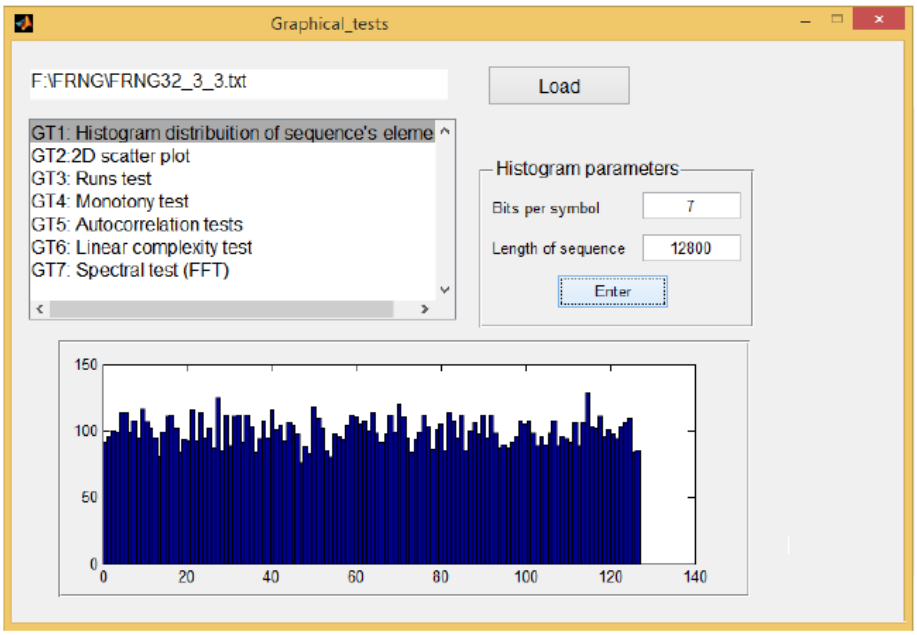

Рис. 3.15. Результати дослідження генератора за допомогою аналізу гістограми розподілу елементів послідовності

Довжина тестованої послідовності дорівнює (12800 символів), кожен символ складається із семи біт і представлений на гістограмі у числовій формі з 0 до 127. Також видно на рисунку 4.11, що частота появи кожного із символів коливається довкола середнього значення (100). Досліджувана послідовність успішно пройшла тест гістограми розподілу елементів послідовності.

Результати дослідження за допомогою аналізу розподілу точок на площині представлені на рисунках 3.16, 3.17.

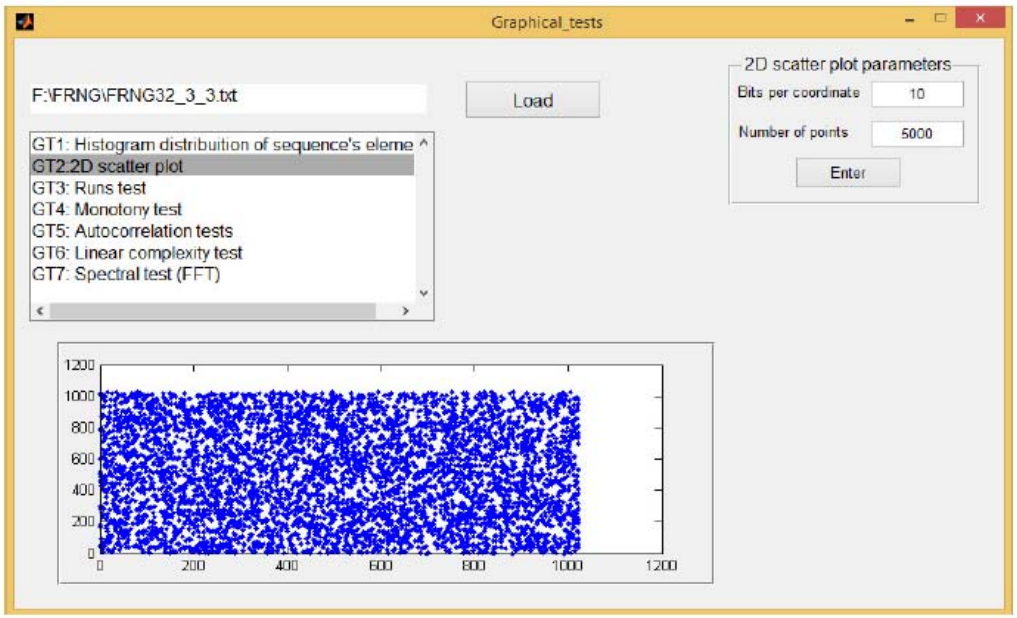

Рис. 3.16. Аналіз розподілу точок на площині (5000 точок)

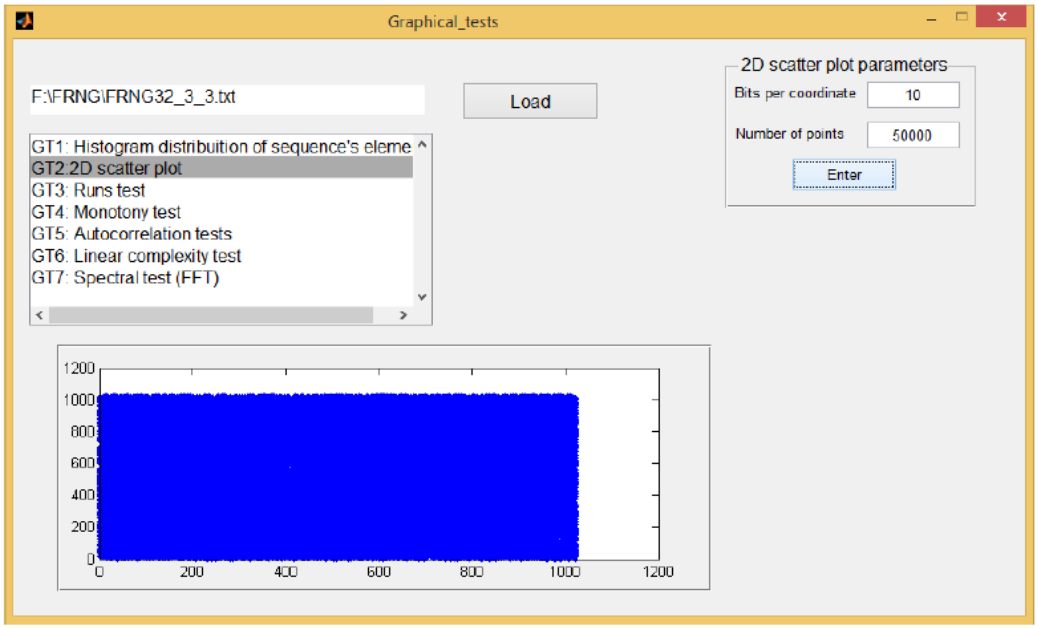

Рис. 3.17. Аналіз розподілу точок на площині (5000 точок)

Дані рисунки показують, що генератор не має регулярностей, згенерована послідовність за допомогою генератора псевдовипадкових чисел, заснованого на нечіткій логіці успішно пройшов цей тест. Результати дослідження за допомогою перевірки серій представлені на рисунку 3.18.

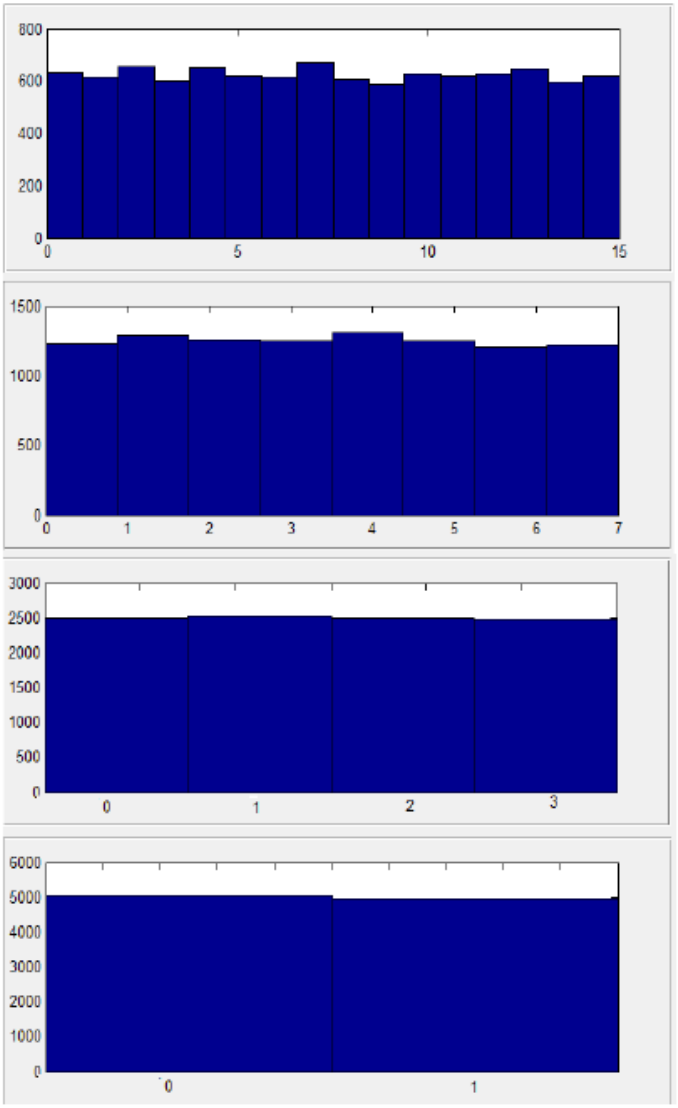

Рис. 3.18. Тест перевірки серій $k = 4,3,2,1$

Цей тест призначений для перевірки рівномірності розподілу символів у досліджуваній послідовності на основі аналізу частоти появи нулів і одиниць і серій, що з k біт. У послідовності, яка має статистичні властивості близькі до властивостей істинно випадкової послідовності, розкиди між числом поява нулів і одиниць, між числом поява серії-двійки кожного виду і між числом поява серій-трійки кожного виду повинні прагнути нуля. В іншому випадку досліджувана послідовність не є випадковою. Результати дослідження за допомогою перевірки монотонності представлені на рисунку 3.19.

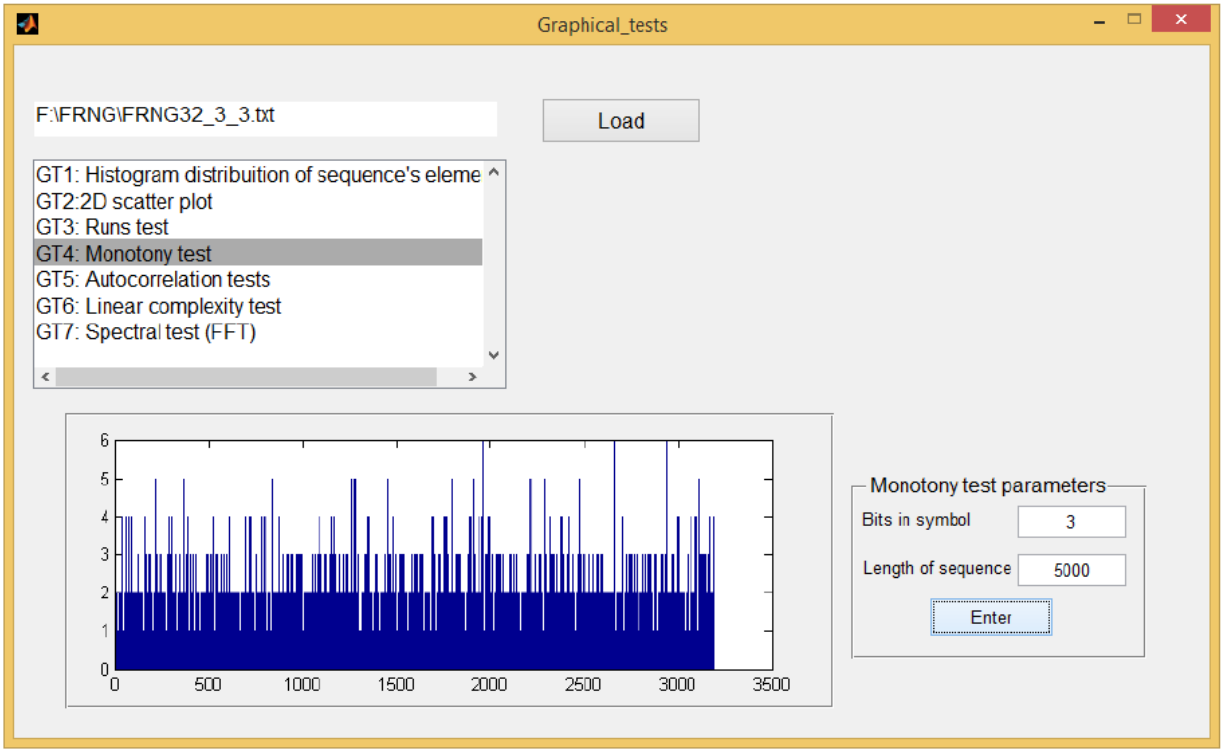

Рис. 3.19. Результат застосування тесту перевірки на монотонність

Цей тест призначений для оцінки рівномірності розподілу символів у згенерованій послідовності на основі аналізу довжин ділянок незростання та незменшення елементів досліджуваної послідовності. У послідовності, чиї статистичні характеристики близькі до характеристик істинно випадкової послідовності, ймовірність появи ділянки незростання (невипадання) певного розміру залежить від його довжини: що більше довжина, то менше ймовірність. В іншому випадку послідовність вважається не випадковою.

Результати дослідження за допомогою профілю лінійної складності представлені на рисунку 3.20. Лінійною складністю двійкової послідовності називається число, яке дорівнює довжині найкоротшого регістра зсуву з лінійним зворотним зв'язком, який генерує послідовність, що має як перші n членів значення цієї двійкової послідовності. Тест профілю лінійної складності є одним із найважливіших тестів. Він призначений для дослідження згенерованої послідовності генерування на випадковість,

аналізуючи залежність лінійної складності одержаної послідовності від її довжини.

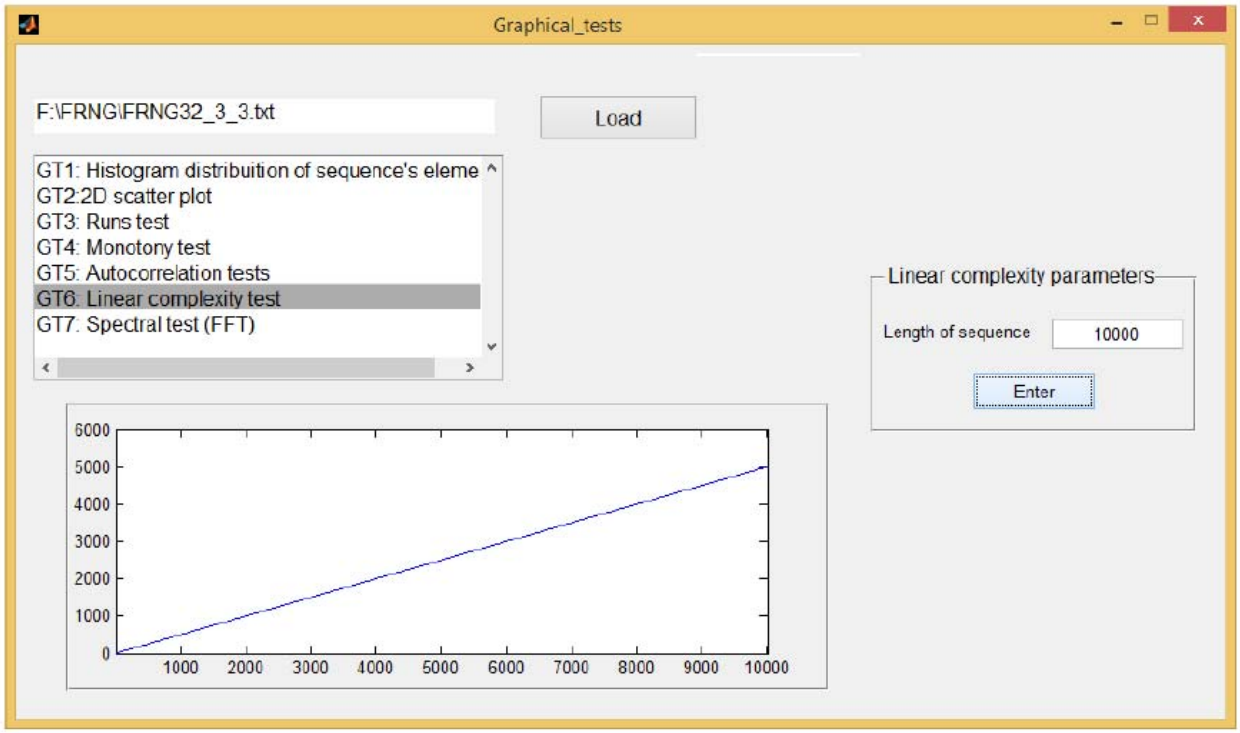

Рис. 3.20. Профіль лінійної складності

У генерації псевдовипадкових послідовностей повинні бути відсутні будь-які регулярності. Для демонстрації цього ми побудували лінійний профіль складності з невеликою кількістю бітів (рисунок 3.21). Цей рисунок показує відсутність регулярностей у досліджуваній послідовності.

При закінченні процесу тестування генерованих послідовностей за допомогою запропонованого генератора отримані наступні результати:

- гістограма розподілу символів послідовності підтверджує рівномірний розподіл символів, формується генератором послідовності;

- аналіз тесту розподілу на площині не показав будь-яких візерунків на отриманому зображенні;

- профіль лінійної складності показує лінійне збільшення складності послідовностей у міру збільшення розміру вибірки;

- графічний спектральний тест показує відсутність значних сплесків гармонік.

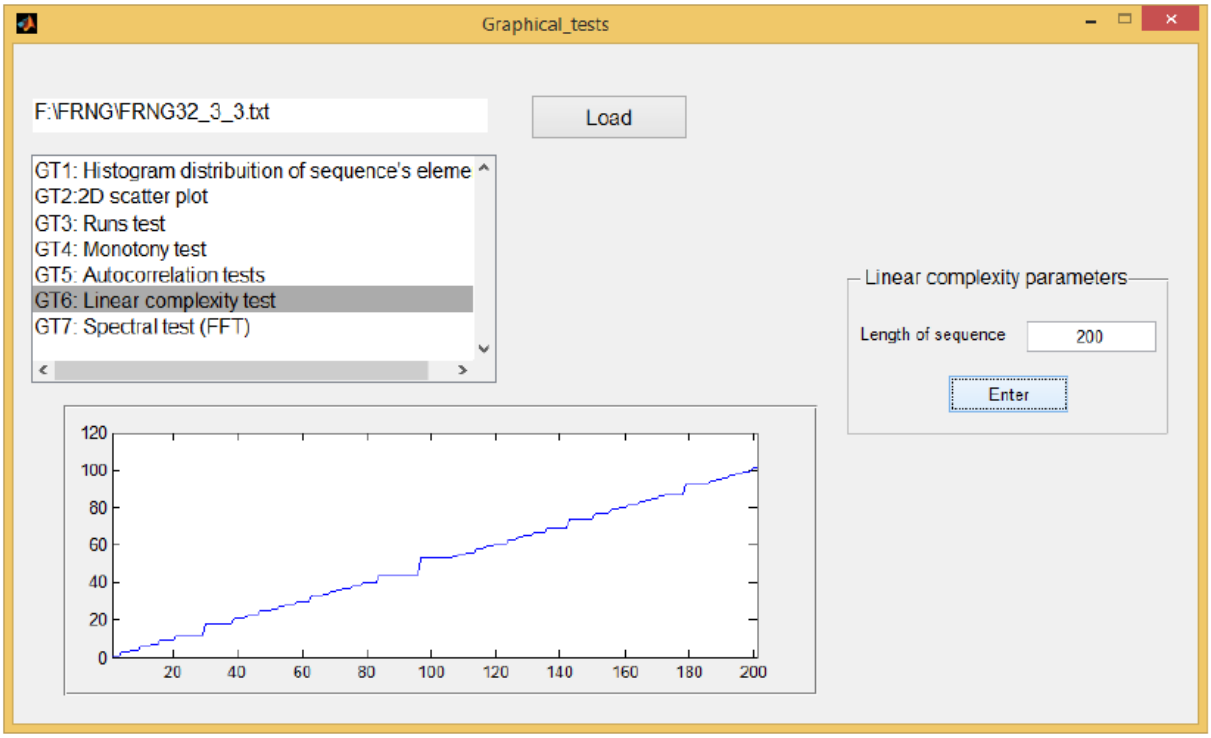

Рис. 3.21. Результати дослідження за допомогою профілю лінійної складності для невеликої кількості біт.

Отже ми дійшли висновку про те, що сформовані псевдовипадкові послідовності генератором псевдовипадкових чисел, на основі нечіткої логіки успішно проходили всі графічні тести на випадковість.

Висновки до третього розділу

1. У цьому розділі представлений комплекс програм генерації псевдовипадкових послідовностей, що реалізує розроблену модель генератора псевдовипадкових чисел, заснованого на нечіткій логіці, і навіть призначений на дослідження параметрів цієї моделі.

2. За допомогою цього програмного комплексу досліджено запропоновану модель, проведено обчислювальні експерименти з метою знаходження найкращих параметрів моделі з позиції якості генерації псевдовипадкових послідовностей.

ВИСНОВКИ

В результаті виконання магістерського дослідження одержано наступні наукові та практині результати:

1. Генератори псевдовипадкових послідовностей з хорошими статистичними властивостями застосовуються для вирішення багатьох прикладних завдань, таких як генерація криптографічних ключів, реалізація протоколів аутентифікації, створення імітаційних моделей і т.д. Для формування якісних послідовностей потрібні хороші статистичні властивості, непередбачуваність, довгий період, ефективність, відтворюваність.

2. Методи теорії нечітких множин пропонують ефективний апарат запровадження нелінійності в комбінуючі генератори псевдовипадкових послідовностей, побудовані з урахуванням регістрів зсуву з лінійним зворотним зв'язком.

3. Застосування методів теорії нечітких множин для задання нелінійної функції в архітектурі дозволить побудувати адаптивну структуру генератора. Це дасть змогу експерту на попередньому етапі визначити основні параметри генератора, що використовуються для аналізу статистичних властивостей виходів використовуваних регістрів, і надалі виконати тюнінг даних параметрів, здійснивши пошук кращих із них, які забезпечують якість формування послідовностей.

4. Розроблено архітектуру та модель запропонованого генератора псевдовипадкових чисел, заснованого на нечіткій логіці. Введено нелінійну функцію, що здійснює комбінування регістрів зсуву з лінійним зворотним зв’язком в даному генераторі, заснована на введенні лінгвістичних змінних та нечітких продукційних правил.

5. Проаналізовано параметри запропонованої моделі, які поділені на дві групи: параметри використовуваних регістрів зсуву з лінійним зворотним зв'язком, а також параметри побудованої нелінійної функції з урахуванням

нечіткої логіки. До першої групи віднесено один основний параметр – тип характеристичних примітивних поліномів використовуваних регістрами зсуву з лінійним зворотним зв'язком, а також степінь даних примітивних поліномів.

6. Представлений комплекс програм генерації псевдовипадкових послідовностей, що реалізує розроблену модель генератора псевдовипадкових чисел, заснованого на нечіткій логіці, і навіть призначений на дослідження параметрів цієї моделі.

7. За допомогою цього програмного комплексу досліджено запропоновану модель, проведено обчислювальні експерименти з метою знаходження найкращих параметрів моделі з позиції якості генерації псевдовипадкових послідовностей.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. A Statistical Test Suite for Random and Pseudorandom Number generators for Cryptocraphic Applications. місце видання невідоме : National Institute of Standards and Technology, 2010.
2. Gaithersburg, Revision 1a. National Institute of Standards and Technology. 2010. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. 2010 p.
3. Doganaksoy A., Sulak F., Uguz M., Seker O., Akcengiz Z. 2017. Mutual correlation of NIST statistical randomness tests and comparison of their sensitivities on transformed sequences. Turkish Journal of Electrical Engineering and Computer Sciences. 2017 p., №1.
4. Е.В. Фаур, А.І. Щерба, В.М. Рудницький. 2016. Метод та критерій оцінювання якості послідовностей випадкових чисел Кібернетика та системний аналіз. 2020 р., Т. 52 №2.
5. М.Б. Будько, М.Ю. Будько, А.В. Гірік, В.А. Грозів. 2019. Методи генерації та тестування випадкових послідовностей. Університет ІТМО. 2019 р