Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

АТЕСТАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Клітинно-автоматне моделювання динаміки планарних мережевих структур

(тема)

Виконав:
студент II курсу, групи СПм-19-1
Скрипник М.С.
(прізвище, ініціали)

Спеціальність
123 – Комп'ютерна інженерія
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма
Системне програмування
(повна назва освітньої програми)

Керівник: проф. Міхаль О.П.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 – Комп'ютерна інженерія
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
"____" _____ 20___ р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові Скрипнику Максиму Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Клітинно-автоматне моделювання динаміки планарних мережевих структур

затверджена наказом по університету від " 30 " жовтня 2020 р. № 1486 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 14 грудня 2020 р.

3. Вхідні дані до роботи
Клітинні автомати
Планарність
Мережева структура
Python

4. Перелік питань, що потрібно опрацювати в роботі
Моделі клітинних автоматів
Використання клітинних автоматів для побудови моделей
Планарні мережеві структури та клітинні автомати

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 14 арк. ф. А4

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз завдання	06.11.2020-16.11.2020	
2	Аналіз науково-технічної літератури	17.11.2020-23.11.2020	
3	Пошук існуючиих моделей	24.11.2020-28.11.2020	
4	Пошук аналогічних моделей комп'ютерних	29.11.2020-07.12.2020	
	мереж		
5	Вивчення концепції імітаційного моделювання	08.12.2020-15.12.2020	
6	Визначення планарності та дуальності	16.12.2020-20.12.2020	
7	Оформлення пояснювальної записки	08.12.2020-11.12.2020	
8	Оформлення графічної частини	08.12.2020-11.12.2020	
9	Представлення магістерської роботи	12.12.2020-13.12.2020	
	науковому керівнику		

Дата видачі завдання 3 листопада 2020 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Міхаль О.П.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка атестаційної роботи: 72 с., 17 рис., 3 дод., 7 джерела.

КЛІТИННІ АВТОМАТИ, ДИНАМІКА, ПЛАНАРНІСТЬ, МЕРЕЖЕВА СРУКТУРА, РЕСУРС, ПРОТОКОЛ.

Дана робота присвячена клітинно-автоматному моделюванню динаміки планарних мережевих структур.

Коротко проаналізована проблематика в цілому, яка пов'язана з моделюванням поведінки і взаємодії об'єктів живої природи в біологічних системах. Відзначено актуальність розробок в даному напрямі моделювання. Ескізно розглянуті базові принципи організації клітинної структури об'єктів живої природи. Виділено аспекти, які поставлені в розроблюваній комп'ютерній моделі. Відмічені особливості, що пріоритетно підлягають модельній реалізації. Конкретизовано коло питань, що підлягають моделюванню в рамках цієї роботи. Цікавим є для вивчення засобами комп'ютерного моделювання обмежень життєдіяльності багатоклітинного організму, пов'язаних з витрачанням даного ресурсу. Очікуване застосування результатів моделювання пов'язане з підтриманням оптимальних режимів функціонування систем, до складу яких входять об'єкти живої природи.

ABSTRACT

Master’s thesis: 72 pages, 17 figures, 3 appendices, 7 sources.

CELLULAR MACHINES, DYNAMICS, PLANNARITY, NETWORK STRUCTURE, RESOURCE, PROTOCOL.

This work is devoted to cell-automatic modeling of the dynamics of planar network structures. The issues in general, which are related to the modeling of behavior and interaction of living objects in biological systems, are briefly analyzed. The relevance of developments in this area of modeling is noted. The basic principles of the organization of cellular structure of objects of living nature are sketched. The aspects which are put in the developed computer model are allocated. Features that are primarily subject to model implementation are noted. The range of issues to be modeled in this work is specified. It is interesting to study by means of computer modeling the limitations of the vital activity of a multicellular organism associated with the consumption of this resource. The expected application of the simulation results is related to the maintenance of optimal modes of operation of systems that include wildlife objects.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 8
ВСТУП 9
1 МОДЕЛІ КЛІТИННИХ АВТОМАТІВ 11
1.1 Історія розвитку клітинних автоматів 11
1.2 Класифікація. Основні напрямки сучасного розвитку клітинних автоматів 12
1.3 Математична модель клітиннго автомата 14
1.4 Модифікації клітинних автоматів. Області застосування 18
1.5 Вимоги до моделей 20
1.5 Структурна схема запропонованої моделі 24
1.6 Висновки до розділу 1 26
2 ВИКОРИСТАННЯ КЛІТИННИХ АВТОМАТІВ ДЛЯ ПОБУДОВИ МОДЕЛЕЙ 28
2.1 Декорельовані клітинні перетворення 28
2.2 Створення моделі КА з використанням декорельованого перетворення 29
2.3 Алгоритми створення та вибору базисів декорельованих клітинних перетворень 29
2.3.1 Блочні клітинні автомати 29
2.3.2 Вибір коефіцієнтов для декорельованого перетворення 31
2.4 Алгоритми створення базисів декорельованих клітинних перетворень 32
2.4.1 Найпростіший алгоритм створення базисе декорельованого перетворення з історії розвитку клітинного автомату з заданим початковим станом 32

2.4.2 Покращений алгоритм створення базису декорельованого перетворення з історії розвитку клітинного автомату з заданим початковим станом ... 33
2.4.3 Алгоритм вибору базисів за кількістю низькочастотних складових ... 34
2.4.4 Алгоритм вибору базисів по розташуванню частотних складових ... 34
2.4.5 Алгоритм вибору базисів по розкиду значень частотних складових ... 35
2.5 Об'єднаний алгоритм вибору базисів ... 35
2.6 Алгоритм роботи програмного засобу ... 38
2.7 Висновки до розділу 2 ... 39
3 ПЛАНАРНІ МЕРЕЖЕВІ СТРУКТУР ТА КЛІТИННІ АВТОМАТИ ... 41
3.1 Основні поняття ... 41
3.2 Використання теорії графів ... 42
3.3 Планарність графу ... 45
3.4 Дуальні графи ... 47
3.5 Взаємозв'язок дуальності та планарності ... 48
3.6 Модельна реалізація ... 49
ВИСНОВКИ ... 60
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ... 61
ДОДАТОК А ... 62
Графічний матеріал атестаційної роботи ... 62
ДОДАТОК Б ... 70
ДОДАТОК В ... 72

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

КА – клітинні автомати

КАП – клітинно-автоматний підхід

ВСТУП

Теорія клітинних автоматів заснована на моделюванні складних систем, що складаються з простих підсистем, результатом колективного функціонування яких є опис складної поведінки всієї системи в цілому. Моделі клітинних автоматів явно зводять макроскопічні явища до точно визначених мікроскопічних процесах. Такий підхід враховує реальну фізику процесу, дозволяючи задавати складні граничні умови, розглядати складні фазові переходи з проміжними сполуками, висувати різні гіпотези щодо формування фаз і розподілу даних. Клітинні автомати є ефективним інструментом для дослідження і моделювання різноманітних процесів та явищ процесів ї. Завдяки клітинним автоматам можна описати безліч станів системи, виходячи з початкового стану її елементів і припускаючи деякі правила переходу з одного стану в інший. Таким чином, для моделювання системи можна застосувати метод клітинних автоматів, який при наявності інформації про початковий стан об'єкта дозволить простежити його розвиток – ітераційний процес, що відображає еволюцію системи. Застосувати ці методи можна також, наприклад, для вирішення завдань стиснення даних.

Клітинно-автоматний підхід, який базується на теоретико-графові (топологічному) формалізмі представлення мережевої структури, дозволяє відтворити будь-який із зазначених глобальних типів динаміки поведінки циклу розвитку співіснування-деградації системи, яка є мережевою структурою. Різниця між фазами розвитку, підтримкою функціонування і деградації, з точки зору клітинно-автоматної моделі, складаються лише в параметрах процедур псевдовипадкової вибірки і заповнення поля моделі. Для модельної реалізації у даній роботі була вибрана процедура підтримки функціоніванія мережевої структури за допомогою передачі елементів, які наближаються к своєму краху, резервних ресурсів сусідніх з ними елементів. Привабливість цього режиму функціонування – в максимальній підтримці працездатності системи. Система, як

ціле, при цьому повільно деградує, але максимально підтримує свою функціональність. Необхідно відзначити основні переваги використання клітинних автоматів для моделювання фізичних процесів:

- значення величин в клітинних автоматах представлені булевої алгеброю і тому не існує проблеми обробки дійсних чисел, всі обчислення є точними, немає помилок округлення;
- можливе вирішення для будь-якої форми кордонів (геометрія багатьох кордонів може бути ефективно описана в автоматах);
- спрощується аналіз розривних рішень (в порівнянні з традиційними кінцево-різницевими схемами);
- так як обчислення в клітинних автоматах є локальними (еволюція даної клітини визначається тільки нею самою та її найближчими сусідами), то моделі, засновані на клітинних автоматах, ідеально підходять до широко поширених зараз алгоритмів паралельних обчислень;
- зв'язок з фізичною реальністю: завдяки своїй "мікроскопічній природі", моделі клітинних автоматів можуть демонструвати явища, які відбуваються в реальних системах, але які не можуть бути отримані з макрорівнянь, – ті явища, які губляться в результаті апроксимації при переході від фізичної моделі до безперервного рівняння (наприклад, явища, пов'язані з мікроскопічними флуктуаціями);
- той факт, що для опису значень клітин в автоматах потрібна тільки булева алгебра, а не дійсні числа з великою кількістю знаків після коми, і локальність правил клітин, дозволяють створювати спеціальні комп'ютери для імітації автоматів з більш простими елементами, ніж у звичайних комп'ютерів, а також з паралельною обробкою інформації. По-перше, це збільшує швидкість обчислень. По-друге, програмування завдання стає набагато легшим (потрібно тільки задати правило автомата у вигляді таблиці входів-виходів). По-третє, топологія модельованого об'єкта в таких машинах клітинних автоматів відтворюється самим же модельним пристроєм, на відміну від традиційних комп'ютерів, що безсумнівно має свої переваги при вивченні модельованих явищ, роблячи цей процес більш наочним.

1 МОДЕЛІ КЛІТИННИХ АВТОМАТІВ

1.1 Історія розвитку клітинних автоматів

Ідея клітинно-автоматного моделювання була запропонована в 1940-х роках Джоном фон Нейманом, коли він працював над кінематичною ідеєю цифрових автоматів, здатних до самостійного виконання. Його первісна концепція була заснована на ідеї робота, що збирає іншого робота. Так з'явилася система у вигляді двомірної решетки, "всередині" якої алгоритмічно описується цей робот. За словами Джона фон Неймана, багато біологічних процесів естетичним чином описуються на мові клітинних автоматів (КА)та не диференційованих виразів. Більша частина роботи, яку Джон фон Нейман не зумів завершити при житті, була завершена і продовжена Бьорксом. Також виявилося, що КА дозволяють моделювати сложне введення ряду системи та поняття, ніж рівняння приватних похыдних. Наприклад, КА, запропонований Н. Винером і А. Розенблузом в 1946 р. для опису сердечних аритмій, дозволяє провести просте аналітичне дослідження. У той час, коли поправки в приватних виробництвах, які імітують таке ж саме введення, вимагають сложного комп'ютерного аналізу, їх оригінальні роботи продовжують цитуватися в сучасних дослідженнях аритмій та збуджучих середовищ. Це створення дослідницьких космічних апаратів, що розвиваються в 60-х роках завдяки актуальним дослідженням за побудовою, адаптацією та оптимізацією, а також вивчення чисто математичних властивостей КА. У 1970-х роках стала популярна вироблена Дж. Конвеєм двомірна автоматична модель клітин, відома як гра «Жизнь». Вона була створена як проста модель екології, що містить клітини, які живуть та змішуються відповідно до кількох простих правил. Цей найвідоміший приклад космічного корабля відображає безліч прикладів діяльності та здатність підтримувати безліч складних конструкцій. Крім

відмінного введення, популярність цієї моделі відчастин була пов'язана із збільшенням доступності комп'ютерів. Наступне волнове дослідження КА – і найбільш цінні для дослідження даних – стало застосування космічних апаратів у фізиці, і, зокрема, пошуку того, як важливі фізичні процеси можуть бути описані у обчислювальній формі. Великий інтерес до цього проявили Т. Тоффоли и И. Фридкин. Ряд работ С. Вольфрама в 80-х роках залучила увагу фізиків, в той час як інші були пов'язані з застосуванням кліточних автоматів для вирішення різних завдань в інших областях. Важливим технологічним досягненням того часу стало впровадження спеціального обладнання у машинобудівних машинах та широкомасштабному використанні паралельних комп'ютерів. Ці дослідження поклали початок розвитку решеточних газів, які стали окремою обласною дослідницькою роботою.

1.2 Класифікація. Основні напрямки сучасного розвитку клітинних автоматів

Вольфрам розробив класифікаційну схему, що складається з чотирьох якісних класів, на які всі клітинні автомати можуть бути розділені в залежності від типу їх еволюції. Еволюція автоматів першого класу призводить до просторово-однорідним станів, еволюція автоматів другого класу – до простих періодичних в часі структур. Автомати третього класу породжують неперіодичні конфігурації клітин. Клітинні автомати четвертого класу демонструють досить складну поведінку з різними локалізованими структурами і структурами, що переміщаються. Особливу цікавість представляють собою автомати четвертого класу. Було висловлено припущення про те, що автомати цього класу можуть демонструвати як завгодно складну поведінку. В принципі поведінку будь-якої системи можна змоделювати, імітуючи крок за кроком її еволюцію. Вольфрам широко аргументує, що досягнення в області клітинних автоматів не є ізольованими,

але досить стійкі і мають велике значення для всіх галузей науки. Такого роду визначення носять здебільшого якісний характер і їх можна по-різному інтерпретувати. Було визначено кілька якісних класів поведінки клітинних автоматів на основі різних статистичних показників і описана організація клітинно-автоматного простору, досліджено, як клітинний автомат може змінюватись шляхом зміни записів в його правилах. Ці різкі зміни мають характер біфуркацій в гладкі динамічні системи, або фазових переходів в статистичній механіці. Автори [100] перерахували шість класів поведінки, які уточнюють класи Вольфрама. Далі ця ідея набула популярності й багато дослідників працювали над її розвитком. Інші підходи до класифікації правил КА докладно описані в роботах Чаті і Менвіль, Гутовіц та МакІнтош. Клітинні автомати можна розглядати як дискретні динамічні системи. Їх глобальні властивості можуть бути отримані шляхом вивчення еволюції автоматів з безлічі всіх можливих початкових конфігурацій. Перші три класи автоматів мають аналоги серед атракторів безперервних динамічних систем (стійких особливих точок, граничних циклів, дивних атракторів). Автомати четвертого класу демонструють складну поведінку з довгими перехідними процесами і тому прямого аналога серед атракторів неперервних динамічних систем для них немає. Теорія динамічних систем дає перший підхід до кількісної класифікації поведінки клітинних автоматів. При їх дослідженні можна так само, як в теорії динамічних систем, вводити різноманітні кількісні характеристики.

В даний час існує три основних напрямки, за якими розвиваються клітинні автомати. Об'єктом досліджень першого напрямку є самі клітинні автомати. Тут розглядаються питання, що стосуються того, як поводяться автомати різних класів; вводяться різні види ентропій, розмірностей, вивчається стійкість автоматів до малих збурень в початкових даних, автомати порівнюються з формальними мовами і т. д. Другий напрямок розвитку клітинних автоматів пов'язаний з розглядом їх як систем, здатних обробляти інформацію, що задається початковими конфігураціями. Існують

клітинні автомати, еквівалентні універсальним обчислювальним машинам. Основні компоненти обчислювального пристрою можуть бути сконструйовані в них за допомогою різних початкових конфігурацій. Приклад таких клітинних автоматів – гра "Життя".

Третій напрям розвитку клітинних автоматів пов'язаний із застосуванням останніх в математичному моделюванні різних фізичних явищ. Метод клітинних автоматів застосовано для моделювання різноманітних процесів просторової динаміки, в тому числі й теплопереносу та дифузії. Продемонстровано принципову можливість клітинно-автоматної апроксимації стаціонарного розподілу температури, яке описується рівнянням Лапласа. Достатньо успішне застосування клітинних автоматів для моделювання різноманітних динамічних систем, які характеризуються близькодією між складовими елементами, демонструє перспективність цього напрямку.

1.3 Математична модель клітиннго автомата

Клітинні автомати (КА) є дискретними динамічними системами. КА використовує множину змінних, які взаємодіють тільки локально і одноманітно. Вони відносяться до моделей, які явно зводять макроскопічні явища до точно визначених мікроскопічних процесів. Відповідно до парадигми КА, моделювання проводиться на n-мірній дискретній сітці. Кожна клітинка є частиною фізичного простору, який моделюється, і описується набором характеристик. Часова вісь розбита на рівні проміжки часу – ітерації.

Одиницею довжини є клітина, а одиницею часу – один ітераційний цикл. Стан кожного елемента сітки в методі КА змінюється послідовно крок за кроком відповідно до деяких локально визначених правил переходу.

Управляючі правила переходу записуються у вигляді:

$$\Omega(x,y,z,t+1) = \Theta(\Omega(x,y,z,t), \Lambda(x,y,z)) \ , \tag{1.1}$$

$\Omega(x,y,z,t+1)$ і $\Omega(x,y,z,t)$ – два послідовних в часі значення стану клітини;

$\Lambda(x,y,z)$ – деяка множина клітин, сусідніх з кліткою, звана околом;

$\Theta(\Omega(x,y,z,t), \Lambda(x,y,z))$ – правило клітинного автомата, за яким визначається нове значення визначеного параметра клітини.

Зміна стану клітини відбувається з урахуванням її попереднього стану і станів клітин околу із заданою вірогідністю. Результатом моделювання є матриця станів клітинного автомата. Поведінка клітинного автомата повністю визначається локальними залежностями.

Таким чином, поведінка КА повністю визначається локальними залежностями. Метод клітинних автоматів природним чином враховує дискретні граничні умови.

Окрім різної розмірності поля (одно-, дво- або тривимірне) існує певне різноманіття сусідства клітин. Наприклад, для двомірного поля існують такі структури сусідства як триангулярне, тетрагональне, гексагональне тощо.

Більше того, в межах окремих структур виділяють різні околи.

В якості схеми сусідства клітин існує декілька варіантів вибору.

Найчастіше застосовують окіл Мура або Неймана. В першому випадку схему сусідства для двовимірного випадку можна проілюструвати наступним чином.

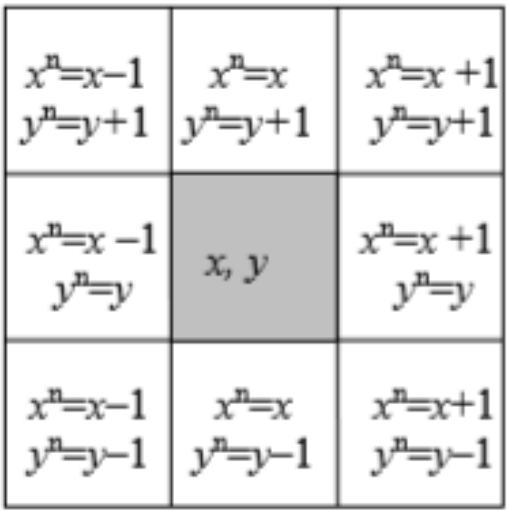

Рисунок 1.1 – Окіл Мура та координати клітин для двовимірної моделі

В другому випадку беруться до розгляду тільки основні сусіди. Ця схема є найпопулярнішою, тому розглянемо її для 3 варіантів вимірів клітинно- автоматного поля. Її можна представити в такий спосіб:

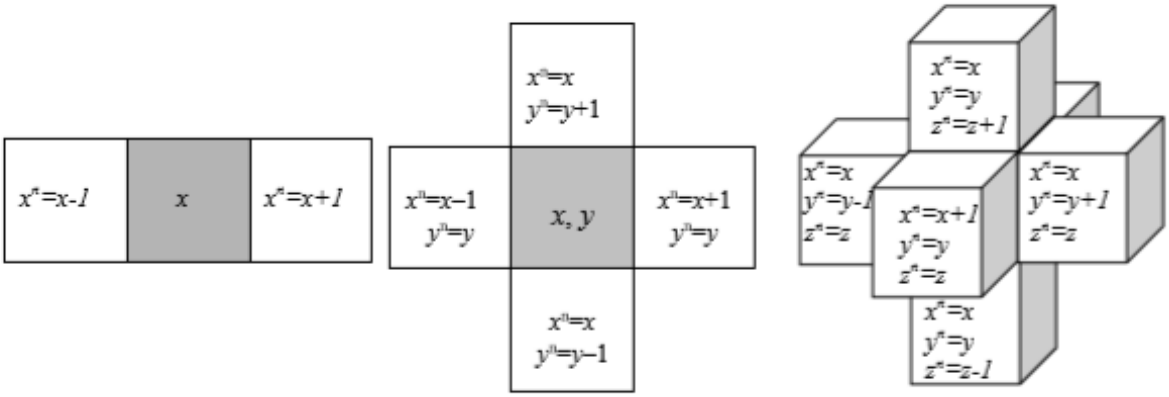

Рисунок 1.2 – Окіл Неймана та координати клітин для а) одно-, б) дво- та в) тривимірної моделі

Для рисунків 1.2. та 1.8. x, y, z – координати клітини, яка розглядається, x^n, y^n, z^n – координати сусідніх клітин, n={1,...N}, N – кількість сусідів.

Для кожного випадку кількість сусідів може бути різною. Так, наприклад для рисунку 1.1 N=8, для рисунку 1.82 а) N=2, для рисунку 2.2 б) N=4, для рисунку 1.2 в) N=6.

Існують й інші, менш застосовувані схеми сусідства, такі як окіл Марголуса, Некера, зірка Давида та інші.

Точки (або клітини) простору можуть поширюється на нескінченність, хоча на практиці передбачаються граничні умови.

Незважаючи на досить тривалий період досліджень клітинних автоматів, на сьогодні загальної теорії клітинних автоматів не сформовано. Основною проблемою клітинно-автоматного моделювання різних процесів є проблема побудови таких локальних правил мікровзаімодій, які під час функціонування певної кількості клітинних автоматів породжували ту чи іншу відповідну макродинаміку. Загальних рецептів щодо вирішення цієї проблеми немає, тому дослідники, що займаються клітинно-автоматним моделюванням, напівінтуітивним шляхом формулюють відповідні правила мікровзаємодій. Основний підхід, який використовується при цьому – побудова імітаційної моделі, яка описує близькодію між складовими елементами системи. Очевидно, що конкретна реалізація імітаційних моделей може бути досить різноманітна. Є успішні приклади побудови однієї і тієї ж моделі системи як в термінах дискретних кліткових автоматів, так і безперервних. Основна відмінність клітинних автоматів від звичайних диференціальних рівнянь полягає у локальності правил, за допомогою яких описується динаміка системи. У разі застосування ДР ми припускаємо існування деяких правил зміни усереднених по всій системі величин. При використанні КА існування таких макроправил необов'язково. Тут досить знати закони розвитку системи на мікрорівні в невеликих просторових областях, з яких складається макросистема.

Інша відмінність – це дискретність змінних. Для КА моделей завжди існує мінімальний дискретний крок, в той час, як для чисельного рішення ДР крок можна зменшувати до нескінченно малих величин.

1.4 Модифікації клітинних автоматів. Області застосування

Найбільш ефективно КА використовуються при описі поведінки системи, де її колективна поведінка визначається локальною поведінкою складових її елементів, коли система стає сильно неоднорідною і усереднення змінних по всій системі не здатне адекватно відображати її стан в цілому.

Для класичного клітинного автомату час прогресує з дискретним кроком, однаковим у всіх точках простору. Кожна точка має динамічні змінні стану, які пробігають кінцеве число значень. Еволюції кожної змінної визначається локальним законом (як правило, називається правилом переходу). І, нарешті, правило діє на всі точки одночасно, паралельно і однаково в усьому просторі для всіх моментів часу.

Пошуки нових нетрадиційних моделей привели до переосмислення ідеї клітинного автомата (КА) і появи безлічі його модифікацій. Одна з найпоширеніших ґрунтується на концепції методу рухомих клітинних автоматів. Вона пов'язує пару взаємодіючих автоматів. Це дозволяє перейти від статичної сіткової концепції до концепції сусідів. В результаті цього, автомати мають можливість змінювати своїх сусідів шляхом перемикання стану пар. Важливою перевагою цього методу є можливість моделювання руйнування матеріалу, включаючи генерацію ушкоджень, поширення тріщин, фрагментацію і перемішування речовини.

Ідея наступної модифікації полягає в тому, що всі дії представляються як випадкові процеси, оскільки в природних умовах для них немає спеціальної синхронізації. Відомо кілька методів моделювання випадкових процесів, що відбуваються в просторі. У різних предметних областях вони мають свої назви і відрізняються способами об'єднання елементарних дій в єдиний просторово-розподілений процес. У матеріалознавстві і мікроелектроніці – це «кінетичний метод Монте-Карло». У каталітичної хімії – це просто «метод Монте-Карло». Насправді вони всі є варіантами

асинхронного КА, які відрізняються способами побудови правил переходу.

Математичні моделі явищ, виражені в термінах асинхронних КА, а також дослідження методів композиції клітинних автоматів дозволили узагальнити накопичений досвід побудови складних КА-моделей.

Загальний підхід до моделювання методом неперервних асинхронних КА полягає у виборі випадковим чином деякої клітини. При цьому всі клітини є рівноімовірними відносно їх вибору. Далі випадковим рівноімовірним чином вибирається деяка сусідня клітина згідно до схеми сусідства та відбувається клітинно-автоматна взаємодія між двома клітинами.

Синхронний режим передбачає послідовну детерміновану схему взаємодії всіх клітин поля протягом одного ітераційного циклу. При цьому, для унеможливлення явища впорядкованої взаємодії клітин, визначених у момент часу t із клітинами, визначеними у момент часу t+1, синхронний підхід слід організувати або використовуючи два поля клітинних автоматів (один для t, другий для t+1), або розподіляючи порядок клітинних взаємодій (наприклад, у момент часу t взаємодіють клітини i+0 та i+1, i+3 та i+3, ..., а у момент t+1 взаємодіють i+1 та i+2, i+3 та i+4, ...). З іншого боку, асинхронний режим природним чином позбавлений можливості впорядкованої взаємодії клітин, значення яких визначені у різні моменти часу. Крім того, асинхронна схема взаємодії більш наближена до реальних процесів теплопереносу та дифузії, що на „мікрорівні” характеризуються стохастичною динамікою. Застосування цього методу вимагає менших у порівнянні з класичними клітинними автоматами обчислювальних потужностей у зв’язку з особливостями програмної реалізації асинхронного клітинно-автоматного алгоритму. Тому для моделювання процесів теплопровідності, що супроводжуються фазовими переходами першого роду, був обраний метод асинхронних клітинних автоматів.

Відношення між типами КА, які відрізняються алфавітами, типами функцій переходів і режимами синхронізації представлені на рисунку 1.3

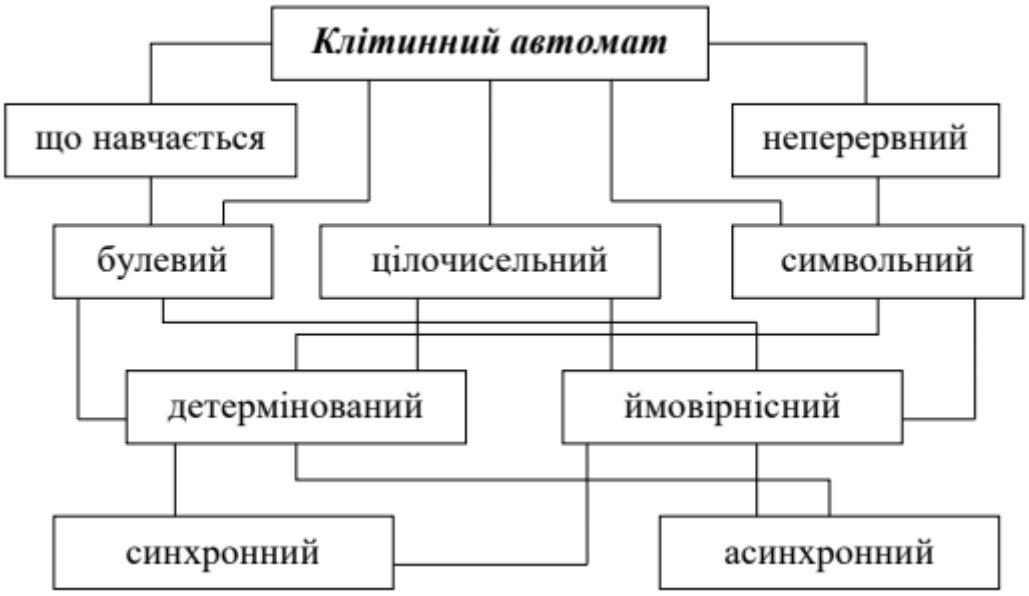

Рисунок 1.3 – Типи клітинних автоматів і відносини між ними

За допомогою клітинних автоматів, як показали роботи останніх років, можна представлять біологічні, фізичні, інформаційні процеси, а також використовувати їх у багатьох інших областях, де традиційні моделі являють собою рівняння в частинних похідних. Тема клітинних автоматів дуже актуальна, тому що область їх застосування практично безмежна, що може привести до розгадок багатьох питань в навколишньому світі.

1.5 Вимоги до моделей

Основні вимоги до розроблюваної моделі визначаються характером, обсягом і заходом інформативності моделювання відповідно до специфіки досліджуваного предмета. В даному випадку, предметом дослідження є еволюції динамічних мережевих структур. Динаміка мережевих структур полягає в їх реконфігурованих в процесі експлуатації, а еволюціонування проявляється при цьому як зростання нерегулярності зв'язків. У зв'язку з цим, модель відтворює клітинно-автоматну структуру в умовах зростання разупорядоченності. З огляду на це, доцільний ряд обмежень спрощень.

Спрощення – не означає скорочення функціоналу моделі. Навпаки: помірно (розумно) спрощена модель містить додатковий потенціал для конкретизації умов її роботи, введення додаткових параметрів і ін. Все це в сукупності – дозволяє краще адаптувати модель до конкретної предметної області, а отже розширює функціонал її застосування.

Перелічимо основні обмежують вимоги до моделі:

- Графічний інтерфейс для моделює програми не потрібен. Чи достатній висновок в Протокол з подальшою off-line обробкою.

- Режим on-line, зрозуміло, теж не потрібен. Модель генерує гістограми, які використовуються в подальшому формуванні критеріїв керування динамічною системою.

- Статистична оцінка результатів на даному етапі достатня з побудови гістограм розподілу числа клітин КА за значенням їх ресурсу, за кількома запусків моделі.

- Побудова графіків гістограм on-line – так само не потрібна. Моделююча програма повинна видати числові дані до протоколу, а графіки можна будувати в табличному процесорі (Excel, Libre Office ...).

- Розглядається (моделюється) тільки двовимірний КА. Багатовимірне розширення, якщо буде потрібно, може бути доопрацьовано за двовимірному аналогу.

- Розглядаються поля квадратної форми з варіантами сусідства 4, 6 і 8. Якщо буде потрібно, можуть бути реалізовані інші значення функції сусідства, але це веде до збільшення числа параметрів моделі.

Сформульовані обмеження визначають специфіку розроблюваної моделі. Розробка проводилася в два етапи. Мета такого поділу – взможнсть порівняти результати. Перший етап – спрощений варіант, другий – більш конкретизований, але більш витратний за часом виконання моделі. Зіставлення графіків (представлені нижче) в цілому демонструє пропорційність. Графіки другого етапу виглядають як систематично масштабовані графіки першого етапу. Це означає, що обидва етапи

представляють інтерес в аспекті використання для конкретних предметних областей. В умовах обмежених обчислювальних потужностей, перший етап – більш простий варіант – дозволить отримає прийнятні корисні результати при збільшеному масштабі моделі.

Коротко прокоментуємо представлений алгоритм

Крок 1. Пуск.

Крок 2. Введення даних: початковий ресурс клітини, розмір поля, число запусків моделі, і ін. Визначення допоміжних змінних. Допоміжні змінні не є безпосередньо параметрами моделі. На даному етапі моделювання їх вплив не регламентується.

Крок 3. Відкриття протоколу. Протокол відкривається на дозапис. Це означає, що програма в цілому може запускатися багаторазово, а в протокол будуть зібрані всі результати запусків згідно з планом машинного експерименту.

Крок 4. Черговий запуск моделі. Зовнішній цикл запусків моделі.

Крок 5. Нове заповнення поля моделі початковими значеннями ресурсу. Чистка поточної гістограми моделі.

Крок 6. Черговий запуск циклу до отримання першого нульового ресурсу. Тут скидаються всі необхідні змінні циклу та інше.

Крок 7. Вибір за допомогою ГВЧ елемента поля A і одного з його сусідів B з найменшим ресурсом. Як результат - буде ресурсозалежна зміна ресурсу.

Крок 8. Зміна ресурсу A відповідно до обраної стратегії.

Крок 9. Перевірка ненульового значення A

Крок 10. Зміна ресурсу B відповідно до обраної стратегії.

Крок 11. Перевірка ненульового значення B

При перевірках наявності нулів в кроках 9 і 11, в разі нуля – перехід на Крок 12.

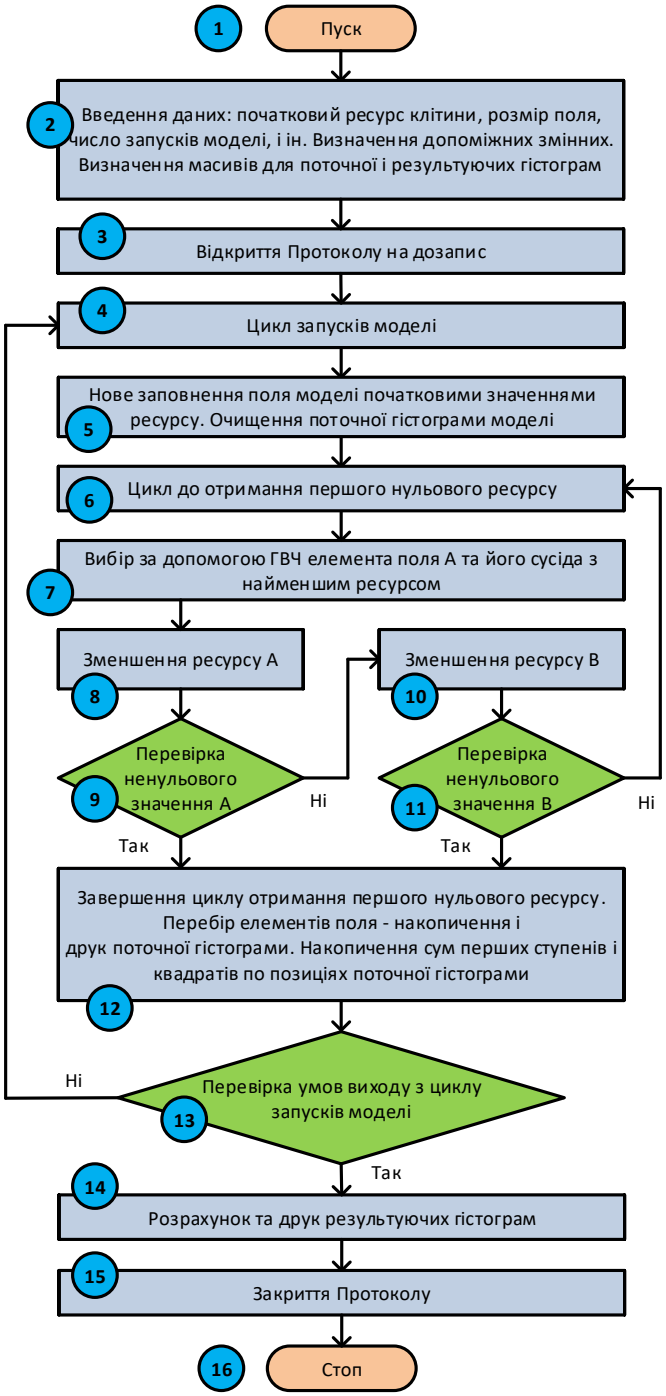

Рисунок 2.4 – Блок-схема програми

Інакше – повернення на Крок 6.

Крок 12. Завершення циклу по отриманню першого нульового ресурсу.

Крок 13. Перевірка виходу з циклу запусків моделі. Перевірка здійснюється порівнянням з фіксованим (введеним на Кроці 2) числом запусків моделі.

Крок 14. Розрахунок і друк результуючих (статистично усереднених) гістограм:

- маточікування,
- середнє відхилення,
- маточікування плюс сигма,
- маточікування мінус сигма.

Крок 15. Закриття Протоколу. Як зазначалося, Протокол буде відкритий на дозапис при наступному запуску програми.

Крок 16. Стоп

1.5 Структурна схема запропонованої моделі

Уявімо і прокоментуємо структурну схему моделі. Покажемо відповідність структурної схеми покрокового опису алгоритму моделі. Можливо, при цьому доведеться кілька повторитися.

Структурна схема представлена на рисунку 1.4. Нумерація блоків – в порядку проходження основного напрямку (потоку) алгоритму – дана в гуртках при блоках.

Легко бачити, що нумерація блоків на рисунку 1.4 відповідає нумерації кроків в покроковому описі. Далі – по блокам відповідно до їх нумерації.

- Пуск.

Запуск програми. Параметри запуску – вихідні дані – вводяться з окремого файлу – стартового Протоколу. Цей файл повинен бути доступний при запуску програми. При відсутності доступу – повідомлення про помилку і завершення роботи програми.

Призначення файлу стартового Протоколу – піти від коригування тексту програми щоразу перед запуском.

- Введення числа N реалізацій моделі та інших параметрів. Ініціалізація Протоколу.

Число N запусків реалізацій моделі всередині одного запуску програми

– зручний спосіб набору репрезентативних результатів. Важливими параметрами, що вводяться під час запуску програми, є конфігурація і розмір поля клітинного автомата. (Конфігурація – прямокутна. Вона задається вказівкою довжини і ширини поля.

Вся інформація, що вводиться з стартового Протоколу, – поміщається в ініційований вихідний Протокол. Останній – створюється параметром append. Тобто в нього дописують результати при кожному наступному запуску програми.

- Запуск черговий реалізації моделі. Установка початкових значень ресурсів клітин КА. Обнулення гістограми

Модель програється N-кратно. При цьому щоразу заново встановлюються значення ресурсів клітин клітинного автомата. Гістограма (масив для накопичення гістограми) всякий раз обнуляється, тому що корисною (результатом роботи моделі) є інформація від кожного запуску моделі.

- Ініціація чергового кроку КА.

Збільшення на одиницю лічильника циклу роботи з КА. При цьому в моделі відтворюється черговий поділ клітин. Для цього вибирається деяка клітка А з поля клітинного автомата і один з її сусідів (одна з клітин оточення) В.

- Вибір клітини А.

Вибір здійснюється псевдовипадковим чином. П позиція клітини вказується з використанням ГСЧ.

- Вибір В одного з сусідів А.

Вибір здійснюється так само за допомогою ГВЧ. Число сусідів - є параметр моделі, що вводиться з стартового Протоколу.

- Зменшення на одиницю ресурсів А і В.

В відповідних (обраних) клітинах А і В поля КА значення ресурсу зменшуються на одиницю, кожен

- Моніторинг поля КА.

Заповнення масиву гістограми Поле КА проходиться, тобто перебираються клітини КА. Підраховується число клітин кожного «віку». Масив гістограми заново перезаповнюють.

- Висновок гістограми в Протокол.

Після завершення просмтра поля КА, в масиві гістограми міститься поточна розкладка клітин КА по віковим групам. Масив скидається в Протокол.

- Перевірка досягнення порогового значення витрачання ресурсів КА.

В даному варіанті моделі граничним значенням є поява першої клітини з нульовим ресурсом.

- Запис в Протокол про завершення поточної реалізації моделі.

Проводиться запис до протоколу про завершення поточної реалізації моделі. Завершенням роботи моделі є фіксація фінального стану гістограми.

- Перевірка відпрацювання необхідного числа N реалізацій моделі.

При кожному запуску програми відпрацьовується послідовно N повних проходів моделі. Перевіряється факт відпрацювання всіх (необхідного числа) реалізацій.

- Закриття Протоколу.

На даному Кроці може реєструватися тривалість роботи програми. Фіксація тривалості – цікава якщо вивчаються можливості практичної роботи з моделлю, зокрема в залежності від розмірів поля КА і з урахуванням конкретної продуктивності обчислювальних коштів.

- Стоп.

Програма відпрацювала повністю. Результат роботи зберігається в окремому файлі Протокол.

1.6 Висновки до розділу 1

Наведена класифікація та основні напрямки сучасного розвитку клітинних автоматів. Описана математична модель клітинного автомата,

визначені шляхи побудови правил КА-взаємодії.

Проведений порівняльний аналіз КА-методу з сітковими методами показав, що незважаючи на доволі складний процес побудови таких локальних правил мікровзаімодій, які б під час функціонування певної кількості клітинних автоматів породжували ту чи іншу відповідну макродинаміку, існує ряд явищ, для яких застосування КА-моделювання виявляється прийнятнішим за використання чисельних методів розв’язку диференціальних рівнянь з частинними похідними.

2 ВИКОРИСТАННЯ КЛІТИННИХ АВТОМАТІВ ДЛЯ ПОБУДОВИ МОДЕЛЕЙ

2.1 Декорельовані клітинні перетворення

Основу клітинного перетворення складає ортогональний базис $C = (c_{ij})_{i=1,j=1}^{NN}$, який є квадратною матрицею, в рядки якої записані вектори, які мають попарну взаємну ортогональність. Таким чином, декорельовані клітинні перетворення відносяться до ортогональних перетвореннь поряд з перетворенням Уолша- Адамара, дискретного косинусного перетворення, перетворенням Хаара (а також і іншими вейвлетного перетвореннями) і т.і.

Декорельоване клітинне перетворення відбувається по формулі:

$$G = C^T F C \qquad (2.1)$$

де F – матриця тієї ж розмірності, що і матриця С, у вигляді якої представлений блок перетворених елементів даних. Побудова окремо взятого ортогонального базису здійснюється наступним чином.

Вибирається клітинний автомат з алфавітом внутрішніх станів А, задається безліч В, таке, що | А | = | В |, і вводиться відображення к: А → В. Потім задається початковий стан решітки клітинного автомата і розглядається відповідна історія розвитку. Окремі стану в даній історії розвитку, загальним числом N, обрані таким чином, щоб виконувалося вимога попарной взаємної ортогональності, і складається базис декоррельованого клітинного перетворення після застосування введеного відображення к. В подальшому, при згадуванні базисів декоррельованих перетворень демо мати на увазі базиси декорельованих клітинних перетворень, якщо інше не обумовлено.

2.2 Створення моделі КА з використанням декорельованого перетворення

З урахуванням введеного поняття декорельованого клітинного перетворення представимо математичну модель стиснення цифрових зображень на основі клітинних автоматів k-го порядку.

Дана модель, зображена на рисунку 2.1, є розширенням розглянутої в в першому розділі узагальненої математичної моделі стиснення цифрових зображень. На вхід представленої моделі надходить цифрове зображення F, на вихід – двійковий вектор l, що представляє собою послідовність біт, що кодують стислий зображення. Вихідна матриця пікселів зображення F послідовно перетворюється в F', G і G' на етапах попередньої обробки, декореляції і квантування відповідно, як це вже було описано раніше. Розширення узагальненої математичної моделі стиснення цифрових зображень полягає в тому, що ортогональний базис декорельованого перетворення описується в термінах клітинних автоматів, що призводить до появи додаткового входу, на який подаються параметри алгоритмів побудови і вибору базисів декорельованих клітинних перетворень, в результаті роботи яких будується ортогональний базис, що визначає конкретне декорельоване перетворення.

2.3 Алгоритми створення та вибору базисів декорельованих клітинних перетворень

2.3.1 Блочні клітинні автомати

Математична модель блочного клітинного автомата виглядає наступним чином

$$CA = \langle Z_n, (N1,...,N_n), A, m, S, \psi \rangle \qquad (2.2)$$

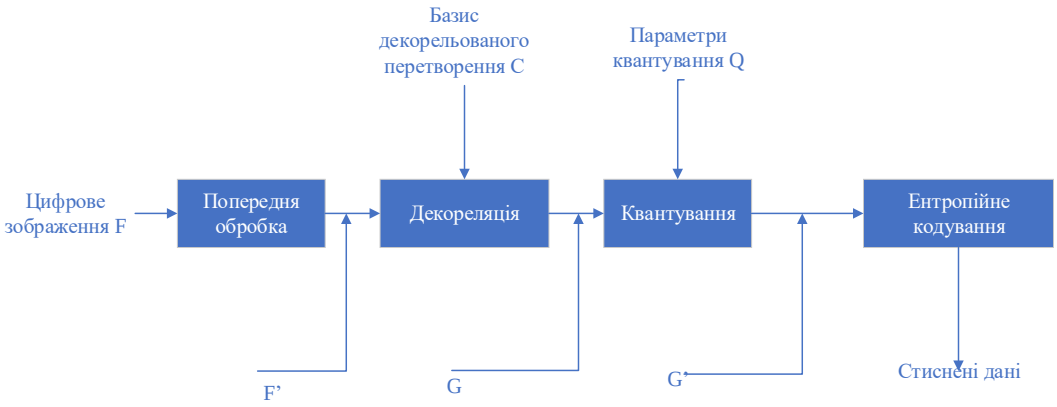

Рисунок 2.1 – Модель стиснення цифрових зображень

Головна відмінність від розглянутої класичної моделі полягає в тому, що в якості перетворення одиниці даних в блоковому клітинному автоматі виступає не окрема клітина, а деяка їх кількість, об'єднана в блок, розмірність якого відповідає розмірності простору координат Z^n. І властивість однорідності решітки клітинного автомата, що виражається в однаковості всіх клітин решітки з точки зору динаміки системи, також переноситься на блоки. Крім того, в описі блочного клітинного автомата важливим є поняття схеми розбиття, що визначає спосіб розбиття решітки на блоки, в загальному випадку, за допомогою завдання зміщення першого блоку від клітини с (0,0, ..., 0) по всіх n просторовим осях.

У наведеній вище п'ятірці компонент Z^n, $(N_1,...,N_n)$ и A відповідають аналогічним компонентам класичної моделі клітинного автомата, m задає розміри блоку, S являє собою вектор-рядок, що складається з схем розбиття і визначальну послідовність їх застосування, ψ є локальною функцією переходу так званою в даному випадку блокової функцією переходу.

Функція ψ зазвичай задається у вигляді таблиці, що складається з двох стовпців, в першому з яких перераховані всі можливі стани блоку розбиття, а в другому - відповідні їм стану в наступний момент часу.

Якщо другий стовпець таблиці являє собою деяку перестановку першого, то ψ, що є в такому випадку підстановкою, буде мати властивість

оборотності. Таким чином, всі можливі оборотні блокові функції переходу для блочного клітинного автомата з заданими розмірами блоку відповідають симетричної групі Sk ступеня k, і їх кількість є | Sk | = k!

Основною передумовою до вибору блокових клітинних автоматів для побудови на основі їх динаміки базисів декорреліруючіх перетворень послужив той факт, що клітинні автомати указанноготіпаобладают більшою гібкостьюсточкі зору управління їх поведінковими властивостями в порівнянні з класичними клітинними автоматами.

2.3.2 Вибір коефіцієнтов для декорельованого перетворення

Декоррельовані перетворення є частотними, так як здійснюють розкладання дискретної функції на частотні складові, виділяючи домінуючі низькочастотні складові і уточнюючі високочастотні. Якщо опуститися на рівень елементарних операцій над даними, то бачимо, що подібні перетворення зводяться до складання значень елементів даних з різними коефіцієнтами, складовими базис перетворення, значення яких зазвичай не перевищують одиниці. Тому має сенс вибирати елементи множини B з відрізка [-1; 1], або подібного йому відрізка [-a; a], 1 <a ∈ N, з малим коефіцієнтом подібності, причому кількість позитивних і негативних значень необхідно вибирати приблизно рівним, щоб забезпечити чергування знаків в базисних векторах.

При побудові множин базисних коефіцієнтів для різних блокових клітинних автоматів будемо оперувати малими цілочисельними і раціональними величинами. В якості знаменника раціональних величин має сенс вибирати ступеня двійки, щоб операція ділення в процесі обчислень могла бути замінена зрушенням на відповідну кількість біт.

2.4 Алгоритми створення базисів декорельованих клітинних перетворень

2.4.1 Найпростіший алгоритм створення базисе декорельованого перетворення з історії розвитку клітинного автомату з заданим початковим станом

Цей алгоритм розглядає як передбачувані елементи базису декорельованого перетворення послідовного стану історії розвитку клітинного автомата з деякого початкового стану. У цьому випадку можливість побудови базису визначається наявністю властивості попарной взаємної ортогональності у одержуваних векторів.

Для побудови N-векторного базису необхідно вибрати клітинний автомат з гратами, що складається з N клітин, і довжиною блоку розбиття m таким чином, щоб грати могла бути розділена на ціле число блоків. Крім того, необхідно вибрати спосіб обробки граничних клітин, які з'являються при будь-якій схемі розбиття з безлічі S відмінною від (0). Для цього можна використовувати так зване обгортання або склеювання кордонів, коли клітини, що знаходяться на протилежних краях решітки приймаються в якості сусідів один одного, або додавати до грат необхідну кількість додаткових клітин, які також братимуть участь у розвитку клітинного автомата.

У розглянутому алгоритмі стиснення зображень на основі клітинних автоматів був використаний другий спосіб, оскільки склеювання кордонів решітки блочного клітинного автомата може привести до появи короткої циклічної історії розвитку. Таким чином, алгоритм побудові N-мірного базису декоррелірующего клітинного перетворення оперує вектор-рядком, що складається з N + 2 (m-1) елементів, оскільки максимальне число клітин, необхідних для доповнення граничних клітин решітки до цілого блоку, становить m-1 при схемі розбиття (1). Далі необхідно задати блочну функцію

переходу Ψ і вибрати початковий стан решітки клітинного автомата C0, а також задати безліч B, що містить елементи базисних векторів.

2.4.2 Покращений алгоритм створення базису декорельованого перетворення з історії розвитку клітинного автомату з заданим початковим станом

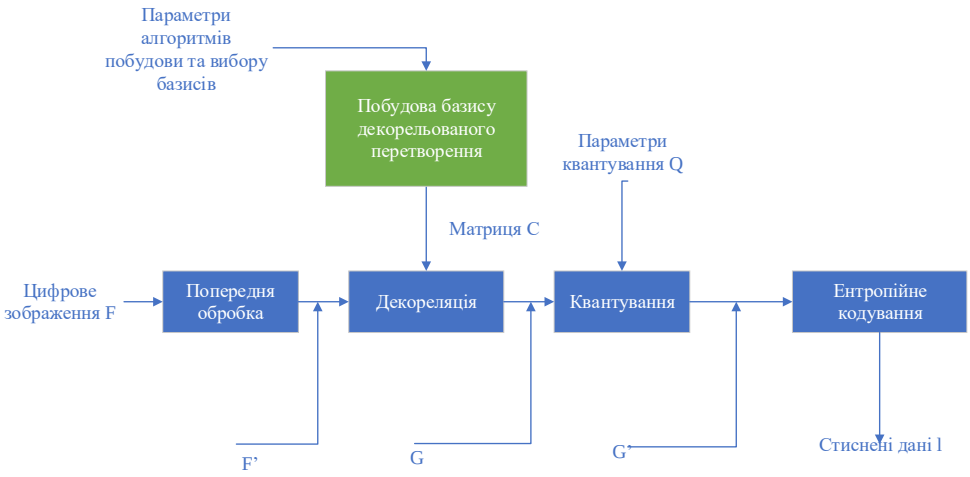

Рисунок 3.2 –Модель стиснення зображень

Алгоритм побудови базисів декорреліруючіх перетворень з станів розвитку клітинного автомата, придатний для практичного використання, повинен гарантовано повертати ортогональний базис, або повідомляти про неможливість його побудови при заданому наборі вхідних значень. Очевидно, що основний недолік запропонованого найпростішого алгоритму полягає в прийнятті наступних один за одним станів розвитку клітинного автомата в якості базисних векторів, оскільки ймовірність їх попарной взаємної ортогональності тим менше, чим більше значення N і k. Таким чином, розвиток запропонованої ідеї полягає в наступному: приймати початковий стан решітки блочного клітинного автомата за перший з N базисних векторів, після чого кожен новий вектор, відповідний черговому

стану розвитку клітинного автомата, перевіряти на взаємну ортогональность з уже знайденими векторами. Однак кількість станів в окремо взятій історії розвитку клітинного автомата з урахуванням наявності додаткових клітин в решітці може становити до $|A|^{N-2(m-1)}$ наслідок чого необхідно ввести обмеження на число переглядаються станів. Будемо вважати, що історія розвитку клітинного є «неперспективною», якщо до моменту часу $t_N < |A|^{N-2(m-1)}$,уло знайдено менше Nп попарно взаємно ортогональних векторів, $1 \leq N_n < N$ причому значення t_N і N_N для клітинного автомата k-го порядку з гратами з N клітин можуть бути підібрані експериментальним шляхом.

2.4.3 Алгоритм вибору базисів за кількістю низькочастотних складових

За допомогою представленого алгоритму можна розділити всі безліч одержуваних базисів декорреліруючіх перетворень на підмножини, які об'єднують базиси з однаковою кількістю низькочастотних складових, і потім працювати з тими подмножествами, котриепредставляютнаібольшійінтерес. Подібне розділення може бути вироблено і на етапі безпосередньої побудови базисів, з тим щоб відразу виключити базиси, що не володіють потрібними характеристиками, з подальшого дослідження.

2.4.4 Алгоритм вибору базисів по розташуванню частотних складових

Алгоритм вибору базисів декорреліруючіх перетворень по розташуванню низькочастотних складових серед перетворених елементів даних, призначивши одним з його параметрів двійкову маску - вектор-рядок, що складається з одиниць і нулів, відповідних бажаного розташуванню низькочастотних і високочастотних складових.

2.4.5 Алгоритм вибору базисів по розкиду значень частотних складових

Використання апарату математичної статистики та обчислення таких характеристик, як дисперсія, середньоквадратичне відхилення, коефіцієнт варіації і т.п., в даному випадку не дуже доречно через малого обсягу вибірки, тому скористаємося простішим підходом і будемо оцінювати розкид значень за допомогою обчислення відносини мінімальної і максимальної за модулем низькочастотних складових.

2.5 Об’єднаний алгоритм вибору базисів

Дослідження базисів декоррельованих перетворень, одержуваних з станів розвитку блочного клітинного автомата, передбачає спільне використання всіх запропонованих алгоритмів. Однак кожен з них здійснює перетворення деякої послідовності даних F, володіє просторової надмірністю, і потім оперує перетвореними елементами даних, внаслідок чого доречно звести всі запропоновані алгоритми в один, який назвемо об'єднаним алгоритмом вибору базисів декорреліруючіх клітинних перетворень, або просто алгоритмом вибору базисів декорреліруючіх клітинних перетворень.

Вхідні дані:

Матриця C = $(c_{ij})^{NN}{}_{i=,\ j=1}$

- Вектор-рядок F = $(f_i)^N{}_{i=1,j=1}$, $f_i \in \{0, \ldots, m-1\}$;
- Дійсне $0 < \lambda \leq 1$;
- Ціле $1 \leq r \leq N$;
- Вектор-рядок H = $(h_i)^N{}_{i=1}$, $h_i \in \{0,1\}$, $\sum_{i=1}^{N} h_i = r$;
- Дійсне $0 < v \leq 1$;
- Ціле $\varepsilon \in \{0,1\}$.

Вихідні дані: відповідь на питання: чи задовольняє базис C заданим умовам кількості, розташування, розкиду і знака низькочастотних складових?

Крок 1. Для і від 1 до N виконати:

$a_i \leftarrow \sum_{j=1}^{N} c_{ij}^2$;

Крок 2. Для і від 1 до N виконати:

$g_i \leftarrow \sum_{k=1}^{N} \frac{1}{a_i} f_i c_{ki}$;

Крок 3.

Обчислити $M \leftarrow \frac{1}{N} \sum_{i=1}^{N} f_i$;

Крок 4. Доки M > 0 виконати наступне:

Крок 4.1 Нехай $l \leftarrow 0$.

Крок 4.2 Для і від 1 до N виконати:

Крок 4.2.1 $x_i \leftarrow \frac{g_i}{M}$

Крок 4.2.2 Якщо $|x_i| \geq \lambda$, то обчислити $l \leftarrow l+1$ та $X_i \leftarrow 1$.

В іншому випадку $X_i \leftarrow 0$.

Крок 4.2.3 Якщо $\varepsilon = 1$ та $X_i = 1$ та $x_i < 0$ то повернути негативну відповідь, так як була виявлена негативна низькочастотна складова. В іншому випадку перейти до наступного кроку.

Крок 4.3 Якщо $l = 0$, то обчислити $M \leftarrow \frac{M}{2}$ та перейти до кроку 5. Інакше перейти на крок 6.

Крок 5. Якщо $l \neq r$ то повернути негативну відповідь, так як кількість низькочастотних складових не збігається з заданим. В іншому випадку перейти до наступного кроку.

Крок 6. Нехай $y_1 \leftarrow M$ та $y_2 \leftarrow 0$

Крок 7. Для і від 1 до N виконати:

Крок 7.1. Якщо $X_i \neq h_i$ то повернути негативну відповідь, так як було виявлено розбіжність з заданим маскою H розташуванням частотних складових. В іншому випадку перейти до наступного кроку.

Крок 7.2. Якщо $|x_i| < y_1$ та $X_i = 1$, будемо вважати, що $y_1 \leftarrow |x_i|$.

Крок 7.3. Якщо $|x_i| > y_2$, то вважаємо, що $y_2 \leftarrow |x_i|$.

Крок 8. Якщо $\frac{y_1}{y_2} \geq \nu$, то повернути позитивну відповідь. В іншому випадку повернути негативну відповідь, так як відношення мінімальної по модулю низькочастотної складової до максимальної менше заданого значення параметра розкиду ν.

Крім кількості низькочастотних складових, їх розташування серед перетворених елементів даних і розкиду значень представлений алгоритм також враховує і їх знаки, оскільки одночасне наявність позитивних і негативних низькочастотних складових зменшує ступінь залишкової просторової надмірності серед перетворених елементів даних.

Рішення поставлених завдань неможливе без створення програмного комплексу, що є науково-дослідним інструментом, призначеним для автоматизації процесів побудови та дослідження ортогональних базисів декорреліруючіх клітинних перетворень, і реалізує розроблений метод стиснення цифрових зображень на основі клітинних автоматів. Для аналізу був обраний програмний комплекс CATCompression.

До складу програмного комплексу CATCompression входять три самостійних програмних модуля, які забезпечують вирішення окремих завдань, що виникають при вирішенні загальної задачі стиснення цифрових зображень на основі клітинних автоматів: CATBasesCreating - програмний модуль, який реалізує побудова і вибір ортогональних базисів декорреліруючіх клітинних перетворень з станів розвитку блочного клітинного автомата; CATBasesResearch - програмний модуль, призначений для дослідження декорреліруючіх клітинних перетворень, одержуваних за допомогою програмного модуля CATBasesCreating; CATCodec - програмний модуль, який реалізує метод стиснення цифрових зображень на основі клітинних автоматів.

Призначення програмних модулів відповідає до трьом основним етапам:

- побудова сімейств базисів декорельованих клітинних перетворень із

заданими характеристиками для множин базисних коефіцієнтів.

- вивчення характеристик стиснення представників побудованих сімейств базисів і виділення таких базисів, декорельовані перетворення на основі яких призводять до найменших спотворень в відновлюванні після перетворення даних при рівних втратах інформації.

- розробка методу стиснення цифрових зображень на основі клітинних автоматів. На рисунку 2.3 представлена схема роботи перерахованих програмних модулів в складі програмних засобів (ПЗ) CATCompression.

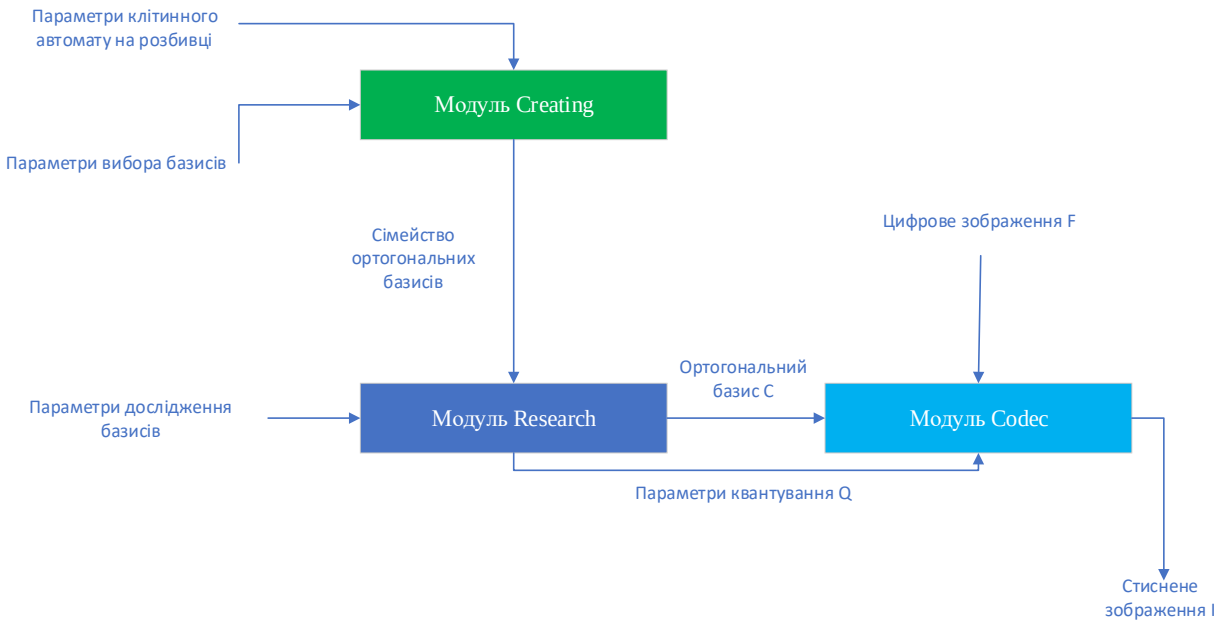

Рисунок 2.3 – Аналіз існуючих програмних засобів з реалізацією КА

2.6 Алгоритм роботи програмного засобу

Так як програмний засіб реалізує описані раніше алгоритми побудови і вибору базисів декорельованих клітинних перетворень, всі запрограмовані користувачем дані в точності відповідають вхідним параметрам даних алгоритмів. Відзначимо тільки, що елементи інтерфейсу програми CATBasesCreating можна розділити на три групи відповідно до груп задаються користувачем вхідних даних: параметри блочного клітинного

автомата і безліч базисних коефіцієнтів; параметри алгоритму вибору базисів; числопереглядаються історій розвитку клітинного автомата і глибина пошуку в окремо взятій історії розвитку.

На рисунку 4.5 представлений приклад перетворення одновимірної функції, що моделює рядок (стовпець) пікселів напівтонового зображення у цифровому форматі з плавним колірним переходом і двома відхиленнями від лінійного збільшення значень пікселів. Представлен приклад перетворення двовимірної функції, що представляє собою блок пікселів напівтонового зображення у цифровому форматі Lenna з координатами верхнього лівого пікселя блоку x = 128, y = 256. На рисунку 4.5 представлений даний блок полутонового цифрового зображення і результат його відновлення після перетворення. В обох випадках була задіяна апроксимація дискретного вейвлетного перетворення, заснована на ортогональному базисі, частотний спектр якого містить рівну кількість низько- і високочастотних складових. Квантування здійснювалося за простою схемою з двома коефіцієнтами SQ = , де елемент виступав в якості подільника низькочастотних складових, високочастотних. В результаті квантування було звернуто в нуль 25% високочастотних складових, однак відмінності між вихідним блоком і відновленим можуть бути помічені лише при значному збільшенні масштабу.

2.7 Висновки до розділу 2

Проаналізований метод стиснення цифрових зображень на основі клітинних автоматів необхідно порівняти з методами стиснення цифрових зображень JPEG та JPEG2000, які були прийняті в якості еталонних. Дане порівняння проводилося на стандартному тестовому корпусі напівтонових і безперервно-тонових цифрових зображень, що включає в себе різні елементи, наприклад Baboon, Barbara, Boats, Goldhill, Lenna, Peppers і т.д. Усереднені результати обчислення значення PSNR з даного корпусу зображень для різних ступенів стиснення представлені на рисунку 3.4.

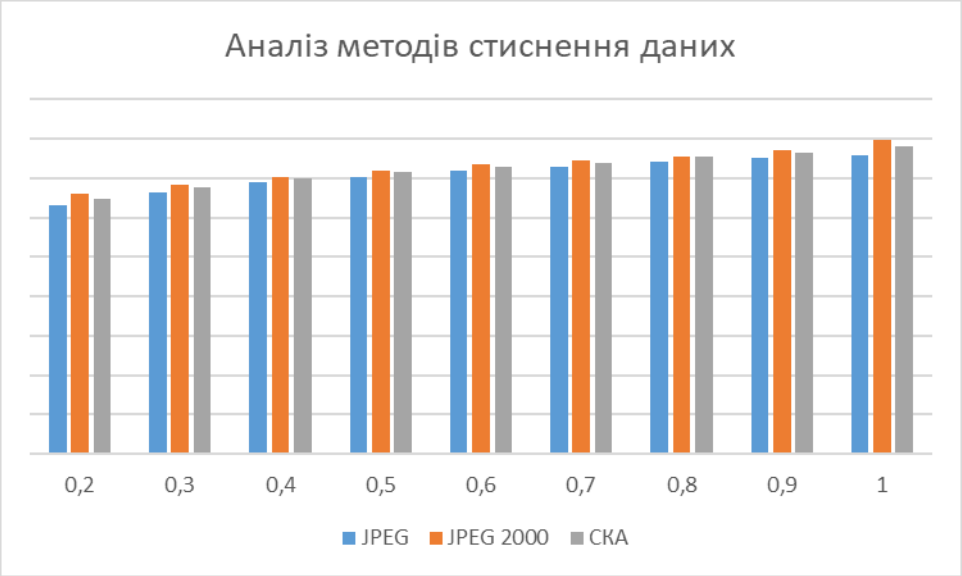

Рисунок 3.4 – Порівняльний аналіз роботи

Розглянуто поняття декорельованого клітинного перетворення, ортогональний базис якого будується з станів розвитку клітинного автомата k-го порядку.

Побудовано математичну модель стиснення цифрових зображень на основі клітинних автоматів.

Розглянуто розширення класичної моделі клітинних автоматів - блокові клітинні автомати, які дозволяють уникнути проблем неконструйованих станів і циклічних історій розвитку невизначеної довжини при побудові декорреліруючих клітинних перетворень в разі вибору оборотних блокових функцій переходу.

3 ПЛАНАРНІ МЕРЕЖЕВІ СТРУКТУР ТА КЛІТИННІ АВТОМАТИ

3.1 Основні поняття

Клітинно-автоматне моделювання - це створення комп'ютерних моделей, що базуються на парадигмі клітинних автоматів.

Моделювання динамікі (МД) є моделювання процесів, що відбуваються в часі. Кожен конкретний фіксований момент часу характеризується певним станом моделі.

Функціонування (зміна, видозміна, рух) моделі в часі є послідовний перехід з одного стану в інший. Характеристики моделі (числові значення, параметри) змінюються в часі. У інтерпретаційному сенсі це відображає певні аспекти моделюється предметної області. Отримання даних і вивчення цих змін, що відбуваються в часі є предметом моделювання. У цьому сенсі йдеться про моделювання динаміки. Тобто під динамікою розуміються зміни, що відбуваються в часі.

Мережева структура (в даному аспекті) - це об'єкт моделювання (сутність), який можливо представити (яка надається в моделі) у вигляді графа, доповненого (забезпеченого, навантаженого) деякими числовими (кількісними) характеристиками.

Числові характеристики можуть бути різними в залежності від предметної області, тобто від інтерпретації моделі і результатів моделювання.

Граф є математичний об'єкт, що відноситься до топології. В описовому плані (тобто якісно, без математичної формалізації), граф є сукупність вузлів і зв'язків між ними.

Теорія графів це підрозділ топології, що вивчає (описує) сутності, представимо у вигляді вузлів (вершин, точок) і зв'язків (ребер, гілок, з'єднувальних ліній) між ними. Додатковими (смисловими, числовими)

характеристиками можуть бути деякі числові величини (набори числових величин), які ставляться у відповідність вузлів і (або) зв'язків графа.

Таким чином, мережева структура, в розглянутому сенсі, є сутність ширша (багата, інформаційно ємна), ніж чисто математичний топологічний граф. Збагачення (додаток, розширення) графа додатковими характеристиками - істотно розширює інтерпретаційні можливості (предметні області, можливості застосування) моделі.

3.2 Використання теорії графів

Основні уявлення, що стосуються графів, ілюструються на рисунку 3.1. Тут представлений граф, що містить 12 вершин (позначені заголовними буквами А, Б, В, ..., М). Деякі з вершин з'єднані з деякими іншими з вершин зв'язками (ребрами). Деякі з ребер (не всі) позначені цифрами 1, 2, ..., 9. Граф може бути зв'язковим і не зв'язковим. У зв'язкового графі кожна з вершин з'єднана з усім іншим тілом (сукупністю вершин) графа. На рисунку 3.1 вершини графа мають від 1 до 7 зв'язків. Зокрема, вершини В і І мають по одному зв'язку; вершини Л, Б, Д, Е мають, відповідно, по 4, 5, 6 і 7 зв'язків. Деякі з зв'язків (ребер) графа можуть бути спрямованими. На рисунку 3.1 ребра 5 і 9 є спрямованими. Вони позначені стрілками, що характеризують напрямок. Так, ребро 5 направлено з вершини Д в вершину К. Ребро 4 позначено двома протилежно спрямованими стрілками. Воно може бути названо двонаправленим (на відміну від односпрямованих ребер 5 і 9). Згідно з принципами, прийнятими в теорії графів, двунаправленное ребро вважається еквівалентним ненаправленому. Тобто ребро 4 - таке ж, як і, наприклад «нуль-спрямовані» 7 і 3. Якщо для конкретної прикладної області моделювання якось (на даному етапі не зрозуміло як) виявиться, що двунаправленность буде функціонально відрізнятися від «нуль-спрямованості», це властивість, мабуть, може бути додано (впроваджено, вставлено) за допомогою визначення спеціального параметра для цього

ребра. Однак, згідно із звичайними (класичними) уявленням теорії графів - це не передбачено.

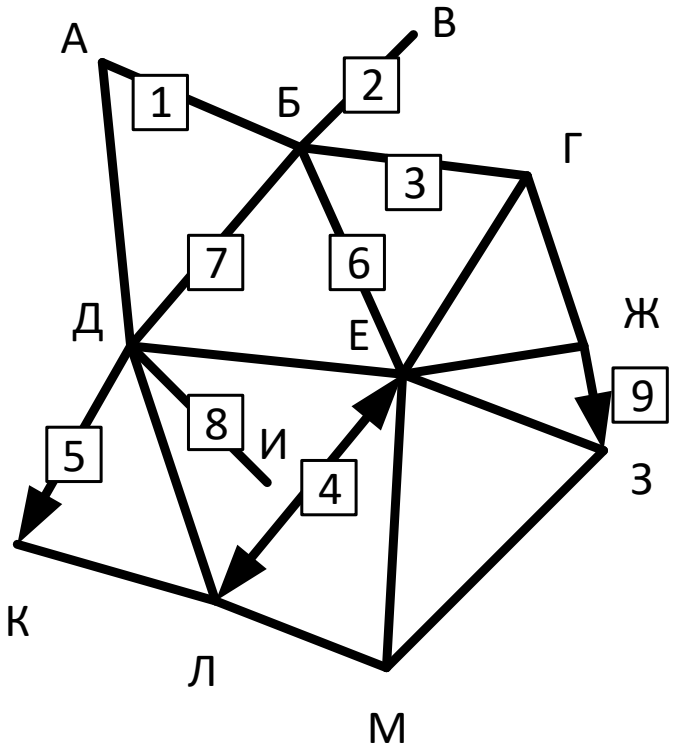

Рисунок 3.1 – Частково-орієнтований граф

Графи, всі ребра яких мають стрілки, називаються орієнтованими. Графи, у яких всі ребра - не спрямовані, називаються неорієнтованими. Графи у яких деякі ребра спрямовані, а інші не спрямовані, - називаються частково орієнтованими. Граф на рисунку 3.1 є частково орієнтованим. Якщо вилучити (видалити) зв'язки 5 і 9, цей граф стане неорієнтованим (рисунок 3.2).

Характерно, що після видалення зв'язків 5 і 9, на графі змінилося число вершин з єдиним зв'язком. З'явилася така вершина - К. Характерно також, що для отримання не орієнтовані графа нам не довелося видаляти зв'язок 4. Як зазначено вище, двонаправлений зв'язок еквівалентна ненаправленої. Важливою якісною характеристикою графа є зв'язність. Як зазначалося вище,

однозв'язний граф - описує деякий єдиний масив зв'язків між вершинами. З будь-якої з вершин можна вийти і через кілька ребер потрапити в будь-яку іншу вершину графа. Бувають так само не зв'язкові графи. У них є кілька підмножин загальної множини вершин, таких, що з вершин одного підмножини можна потрапити по соединітелним ребрах в вершини іншого підмножини. Відсутні з'єднувальні ребра між зазначеними окремими частинами (областями) графа. Відсутність (або порушення) зв'язності графа - важлива особливість мережевої структури. У багатьох випадках (в залежності від конкретної предметної області), порушення зв'язності графа істотно впливає на функціональність мережевої структури, аж до повного виходу з ладу. У зв'язку з цим, зокрема, моделювання, пов'язане зі зміною зв'язків в графах, конкретно интерпретационно прив'язане до предметної області і набуває велике прикладне значення.

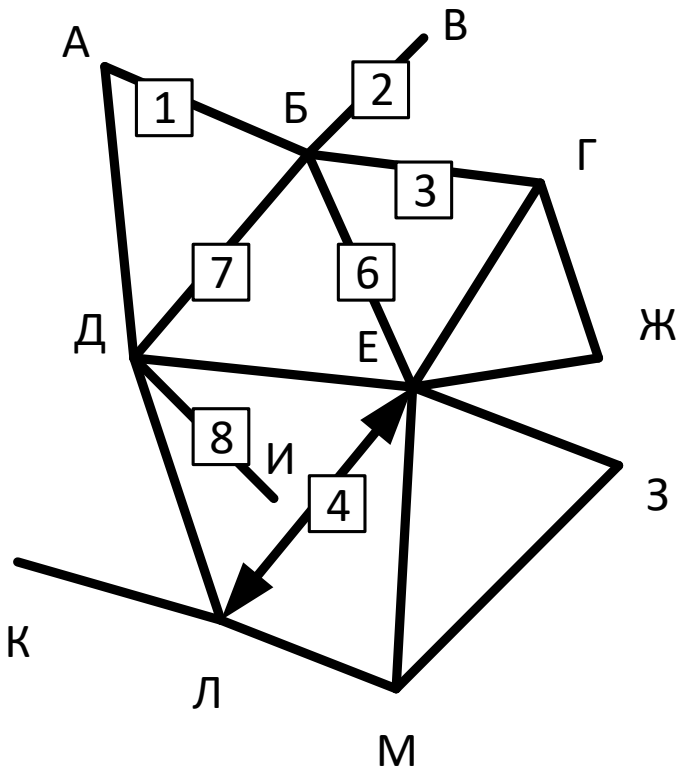

Рисунок 3.2 – Не орієнтований граф

3.3 Планарність графу

Цікавою особливістю (характеристикою) є планарність графа, яку зручно пояснити на прикладі класичної демонстраційної задачі (притчі) про три колодязя. «Умови завдання» (тобто опис предметної області) наступні. Троє сусідів беруть воду з трьох колодязів. Кожному з них потрібна вода з кожного з колодязів. Мабуть, у цих водних джерел різні корисні властивості, однаково необхідні кожному з сусідів. Проблема в тому, що сусіди між собою пересварилися. Тому потрібно прокласти маршрути від їх будинків до криниць так, щоб стежки (ребра графа) між собою не перетиналися. Дане завдання («три будиночки - три колодязя») не розв'язна на площині без застосування «містка» (рисунок 3.3), але може бути вирішена на поверхні з більш високою зв'язністю, наприклад, на тороідальній поверхні (рисунок 3.4), або, наприклад, на аркуші Мебіуса (рисунок 3.5).

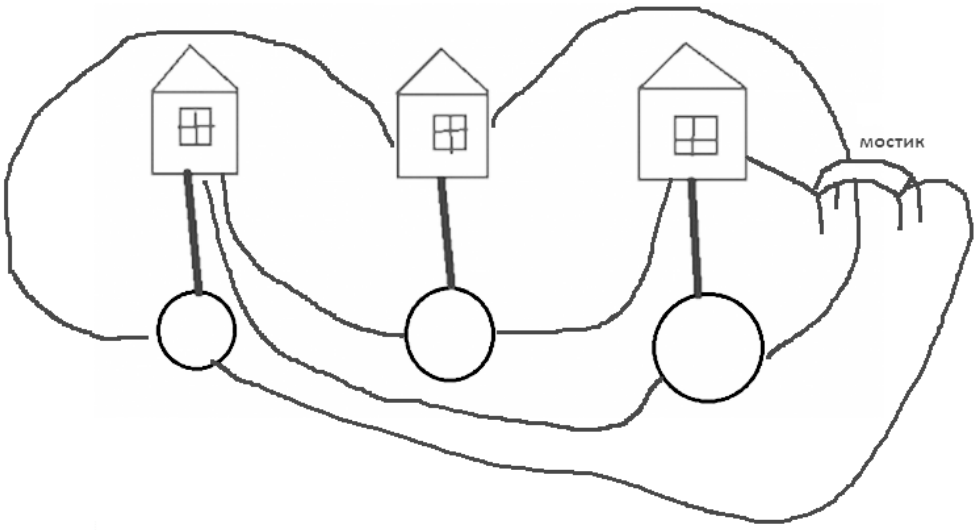

Рисунок 3.3 – Планарний варіант завдання

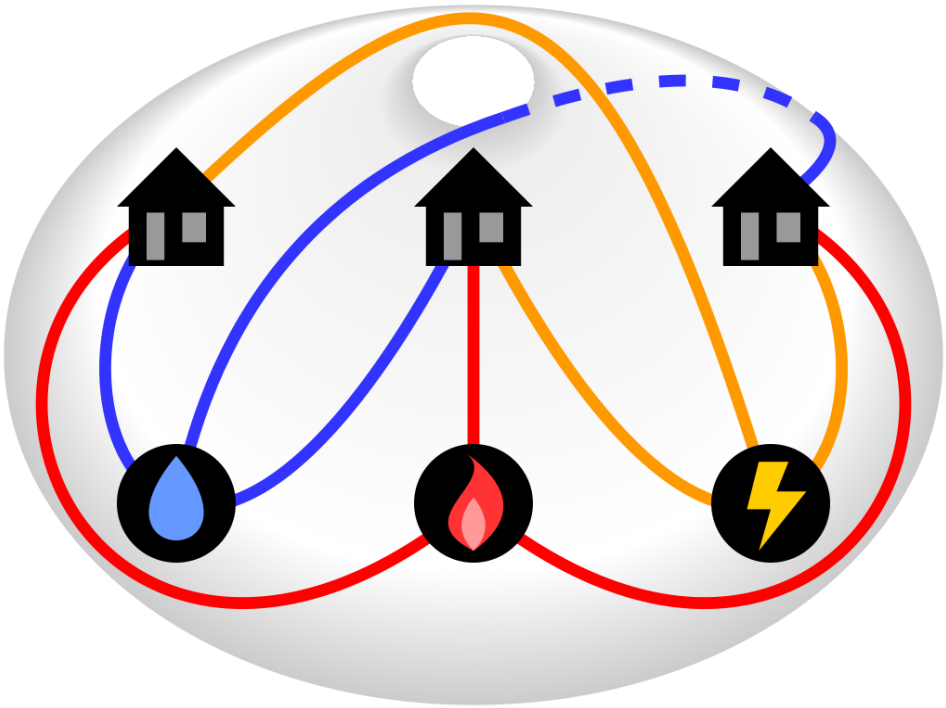

Рисунок 3.4 – Варіант рішення на тороідальній поверхні

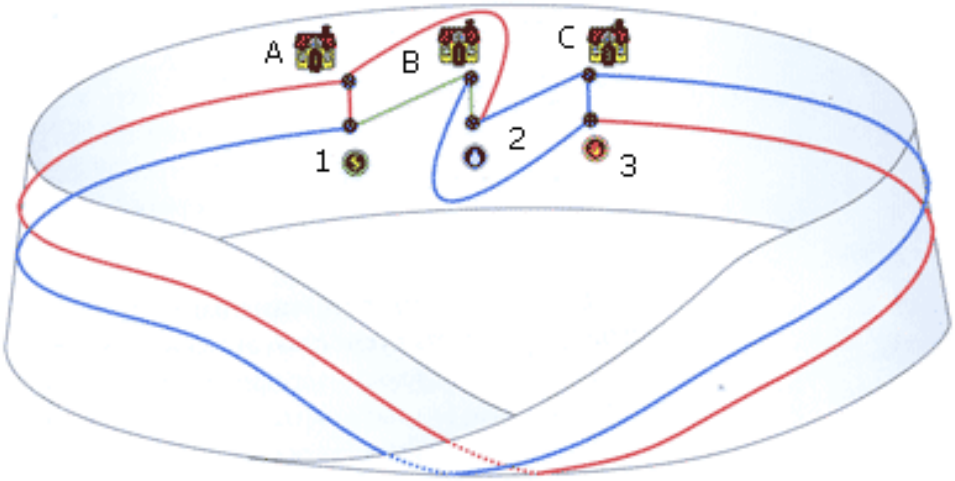

Рисунок 3.5 – Варіант рішення на листі Мьобіуса

У цій атестаційній роботі є обмеження (конкретизація): чисто планарний варіант графа мережевої структури. Слід зазначити, що в прикладному сенсі подібне обмеження не є фатальним. Площинність (розгорнутість на площині) мережевої структури, реалізована в математичній

моделі, у багатьох приватних конкретних випадках, мабуть (швидше за все), може забезпечити наближені рішення, досить близькі до реальності при «не дуже заплутаних» багатозв'язних варіантах. При цьому, зрозуміло, для конкретних точностних оцінок потрібні окремі дослідження, що виходять за межі цієї роботи.

3.4 Дуальні графи

Поняття дуального графа ілюструється рисунком 3.6. Цей же рисунок ілюструє процедуру побудови дуального графа. На рисунку графи, зображені суцільними і пунктирними лініями, є взаємно дуальними. Вершини і межі графів замінюють один одного. При побудові в центрах граней графа поставлені вершини дуального графа і з'єднані між собою так, щоб одне ребро дуального графа перетинало одне ребро вихідного графа.

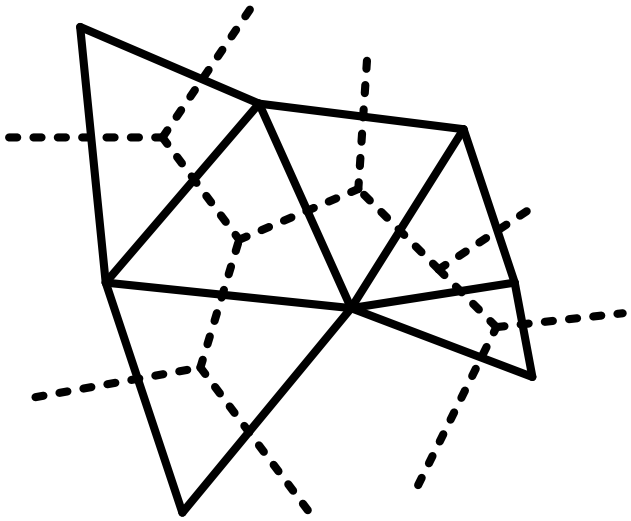

Рисунок 3.6 – Процедура створення дуального графу

Показані на дуальному графі (пунктирні лінії) ребра, що перетинають «зовнішній контур» вихідного графа - на рисунку 3.6 «повисають у повітрі». Насправді, ці лінії з'єднуються в єдиній вершині (умовно не показаній), що відповідає «зовнішній області» вихідного графа. Невідповідність, що

здається з рисунку 4.6, що ці лінії розходяться в різні боки, є умовною. Справа в тому, що граф - це не геометричний об'єкт, а топологічний. Тому ребра графа - це не відрізки прямих, а сполучні лінії довільної необхідної конфігурації, що зрозуміло, зокрема, з рисунків 3.3, 3.4, 3.5. Єдина вимога до них - не порушення топологічних властивостей зображуваного об'єкта. У випадку, показаному на рисунку 3.6 такою властивістю є сходження в одній точці (вершині) всіх ліній , які «повисають у повітрі». Ця точка, де вони зійшлися - є вершина дуального графа, поставлена (згідно з процедурою побудови) на зовнішній межі вихідного графа.

3.5 Взаємозв’язок дуальності та планарності

У зв'язку зі сказаним, цікаво згадати класичну формулу Ейлера для площинного (планарного) графа:

$$В - Р + Г = 2,$$

де: В – число вершин,

Р - число ребер,

Г - число граней графа.

Характерно, що В і Г входять в цю формулу симетрично. Даний факт по суті демонструє (доводить) те, що якщо вихідний граф планарний, то і дуальний граф так само планарний. Тобто вихідний і дуальний графи - взаємно планарні. Зазначений взаємозв'язкок між дуальністю і планарністю, стосовно до моделювання мережевих структур, цікавий в плані інтерпретації результатів моделювання. Розроблена модель може розумітися як мережева або як клітинна, в залежності від інтерпретації. Відповідно, результати моделювання можуть застосовуватися як для мережевих, так і для пористих структур, в залежності від того, як («на якій з цих мов») описується (або більш зручно описувати) відповідну предметну область. Слідство цієї

подвійності інтерпретації - універсальність, адаптивність і підвищена варіабельність моделі, а також, зокрема, подвоєння можливих областей застосування результатів.

3.6 Модельна реалізація

Для модельної реалізації була вибрана процедура підтримки функціонування мережевої структури за допомогою передачі елементів, близьких до знищення, резервних ресурсів сусідніх з ними елементів.

Привабливість цього режиму функціонування - в максимальній підтримці працездатності системи. Система, як ціле, при цьому повільно деградує, але максимально підтримує свою функціональність. На рисунку 3.1 представлена блок-схема програми моделі для вивчення динаміки поведінки (розвитку, підтримки, деградації) мережевої структури.

Підбір функціональності параметрів і відповідне алгоритмічне втілення в ній такі, що реалізується (може бути простежено) саме зазначений режим з максимальним уповільненням деградації системи. Метою моделювання може бути, зокрема, знаходження критеріїв і вказівка найбільш доцільного моменту проведення регламентного обслуговування або часткової функціональної реконструкції системи. Залежно від конкретної предметної області, цільові орієнтації можуть дещо різнитися, але в цілому вивчення поведінки системи в динаміці - найбільш доцільний шлях забезпечення оптимального обслуговування (супроводження, підтримки) мережевої структури. Як показано на рисунку 3.1, старт алгоритму роботи моделі супроводжується відкриттям протоколу. Даний момент вимагає пояснень. Було визнано недоцільним зайве ускладнення програмної частини, замикаючи роботу всієї моделі на фінальній генерації результатів. Існують програми, спеціалізовані і оптимізовані під обробку числових результатів. Обробка даних (безвідносно до конкретного типа та призначення) - окрема просунута і істотно розвинена програмно-інформаційна галузь.

У ній розвиваються, програмно реалізуються і вдосконалюються (з урахуванням конкретних практичних запитів) різні підходи, пов'язані зі статистичною обробкою, апроксимацією, інтерполяцією і екстраполяцією даних. А також розвиваються засоби графічного відображення результатів. Дублювання досягнень цієї окремої інформаційної галузі стало б марною суттєвої розтратою сил з малими шансами на успіх отримання навіть порівнянних результатів. Тому в процесі розробки моделі було визнано розумним зосередити зусилля на розрахунковій частині моделі, з висновком результатів (числових даних) в проміжний текстовий протокол. Подальша обробка даних і графічна репрезентація - проводиться з текстового протоколу в додатково обраному спеціалізованому засобі відображення. Були використані електронні таблиці (Excel) на яких і реалізувалася графічна частина проекту. Реалізація введення даних - так само без спеціалізованого інтерфейсу. Різноманітність (багатопараметрічность) реалізації системно-мережевої частини моделі передбачає безпосередню програмну модифікацію моделі. Поточні відомі елементи алгоритмічних рішень не вичерпують всіх можливих варіантів. Модель являє собою відкриту систему. Як випливає з попереднього викладу, структурно-мережева предметна область описана в досить загальному (узагальненому) вербальному плані, і навіть без функціональної формалізації. Це означає припустимо широкий інтерпретаційний контекст. При цьому контекст є відкритим, тобто таким, що модифікується і доповнюється практично в будь-якій частині. У зв'язку з цим, при поточній експлуатації та модифікації моделі, потрібно безпосередній доступ до програмного коду в окремих вузлах і блоках моделі. Тому не доцільно використання так само і окремого програмно реалізованого спеціалізованого інтерфейсу введення вихідних параметрів. Цей інтерфейс довелося б щоразу переробляти при функціональному модифікуванні моделі, оскільки вводяться параметри (число параметрів), що по суті є частинами функціоналу моделі.

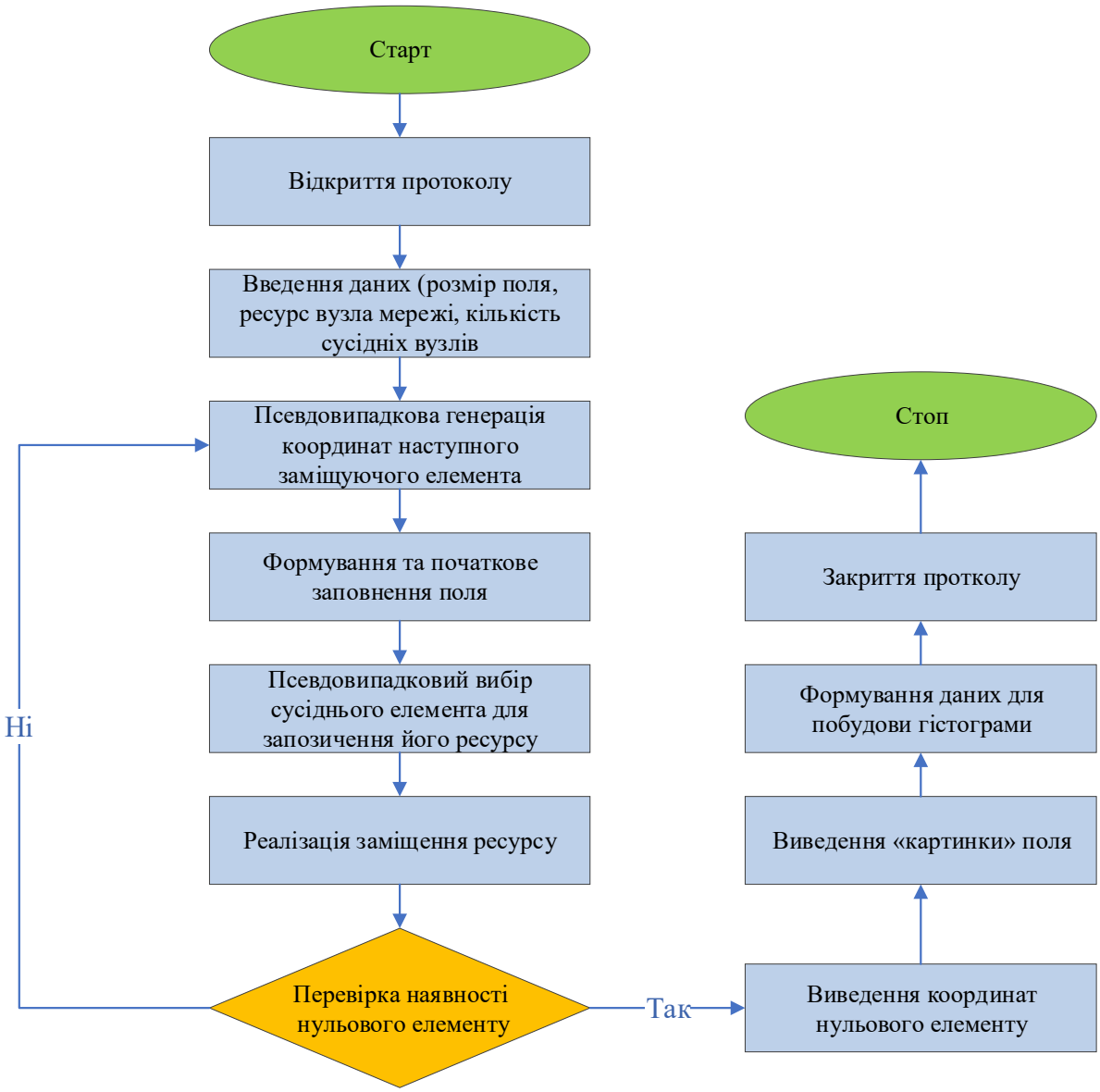

Рисунок 3.7 – Блок-схема моделі

3.7 Алгоритм

Покроковий опис розробленого алгоритму наступний:

Крок 1. Старт. Початок роботи моделі. Передбачається, що перед виконанням цього Кроку (тобто перед стартом програми) оператор ввів нові (змінив колишні) вихідні дані.

Крок 2. Відкриття протоколу. Програма відкриває протокол з прописаним в тексті (в коді програми) ім'ям на дописування. Це означає, що

колишні записи в протоколі (результати попередніх запусків програми) зберігаються, а нові результати дописують «в хвіст» тексту протоколу.

Крок 3. Введення даних (розмір поля, ресурс вузла мережі, число сусідніх вузлів). Дані, що стосуються параметрів моделі, «стають відомими» (прийняті до виконання).

Крок 4. Формування і початкове заповнення поля. Поле (масив поля) формується і заповнюється однаковими вихідними числами - значеннями ресурсу елемента поля (вузла мережевої структури)

Крок 5. Псевдовипадкова генерація координат чергового елемента, який замінюєтьс. Генерується пара псевдовипадкових чисел, які масштабуються як координати одного з елементів поля. Цей елемент буде модифікований з використанням ресурсу одного зі своїх сусідів. Це буде інтерпретовано в моделі як видалення (втрата) цього елемента і заміщення його іншим, частково використовує ресурс одного з сусідніх (оточуючих) елементів.

Крок 6. Псевдовипадковий вибір сусіднього елемента для запозичення його ресурсу. Буде обраний той сусідній елемент, який «поділиться» своїм ресурсом з заміщаються елементом.

Крок 7. Реалізація заміщення ресурсу. Здійснення дій по заміщенню, згідно значень, обраними на кроках 5 і 6.

Крок 8. Перевірка наявності нульового елемента. Після виконання Кроку 7 може виявитися так, що елемент цілком задіяв (використовував) ресурс свого сусіда. При цьому в полі з'являється елемент з нульовим ресурсом. Тому здійснюється повний перебір елементів поля з метою пошуку нульового елемента. Якщо нульовий елемент не виявлено, це означає, що функціональність мережевої структури зберігається. Перехід на Крок 5 для обробки чергового заміщає елемента. Якщо нульовий елемент виявлений, це означає, що мережева структура втрачає функціональність. Програма виходить з циклу обробки заміщаються елементів і йде на завершення - Крок 9.

Крок 9. Висновок координат нульового елемента. Координати нульового елемента заносяться в потокол.

Крок 10. Висновок "картинки" поля. В протокол заноситься поточний стан матриці поля. Елементи матриці (все крім одного - нульового) відображають свій невикористаний ресурс, що залишився.

Крок 11. Формування даних для побудови гістограми. Підраховується число елементів кожного із значень ресурсу. Лінійка цих чисел є гістограма розподілу залишкового ресурсу мережевої структури. Даний набір чисел - основне досягнення даного запуску програми, - основний результат моделювання.

Крок 12. Закриття протоколу. Файл протоколу закривається і стає доступним для оператора.

Крок 13. Стоп. Завершення роботи моделі. Програма зависає у вічному циклі в очікуванні натискання будь-якої клавіші комп'ютера. Це зроблено для того, щоб оператор побачив на екрані повідомлення про завершення роботи програми і продовжив дослідження. Зокрема, можливо, оператор введе нові дані і перезапустить модель.

3.8 Реалізація моделі

Python - об'єктно-орієнтована мова високого рівня. На відміну від Java, Python не вимагає виключно об'єктної орієнтованості, але класи в Python досить просто вивчити і зручно використовувати. Python підтримує множинне успадкування, перевизначення інфіксне операторів. В Python є винятки і механізм їх перехоплення. Таким чином, може бути побудована правильну обробку помилок і створена надійна програма. Вбудовані механізми інтроспекції дозволяють опитувати інтерфейси об'єктів під час виконання програми. Вибір Python в ролі мови програмування для опису властивостей моделі зумовлений низкою факторів: - Програмний код на мові Python читається легше, а значить, багато-кратне його використання і

обслуговування виконується набагато простіше, ніж використання програмного коду на інших мовах сценаріїв. Якість програмного забезпечення: для багатьох основна перевага мови Py-thon полягає в легкості читання, ясності і більш високій якості, від-Ліча його від інших інструментів в світі мов сценаріїв. Єдинооб-разіе оформлення програмного коду на мові Python полегшує його поні-гу навіть для тих, хто не брав участі у створенні цього коду. Крім того, Python підтримує найсучасніші механізми багаторазового використання програмного коду, яким є об'єктно-орієнтоване програмування (ООП). - Обсяг програмного коду на мові Python зазвичай становить 1/3 - 1/5 еквівалентного програмного коду на мові C ++ або Java. Це означає менший обсяг введення з клавіатури, менша кількість часу на налагодження і менший обсяг трудовитрат на супровід. Висока швидкість розроб-лення: в порівнянні з компілює або строго типізований язи-ками, такими як C, C ++ і Java, Python у багато разів підвищує виробляй-ність праці розробника. Крім того, програми на мові Python за-пускаються відразу ж, минаючи тривалі етапи компіляції і зв'язування, що не-обхідні в деяких інших мовах програмування, що ще більше збільшує продуктивність праці програміста.

Перенесення програмного коду з операційної системи Linux в Windows зазвичай полягає в простому копіюванні програмного коду сценарія з однієї машини на іншу. Переносимість програм: велика частина програм на мові Python виконується без змін на всіх основних платформах. Більш того, Python надає масу можливостей по со-будівлі переносяться графічних інтерфейсів, програм доступу до баз даних, веб-додатків і багатьох інших типів програм. Навіть інтерфейси операційних систем, включаючи спосіб запуску програм і обробку каталогів, в Python реалізовані стерпним способом. Стандартна бібліотека надає масу можливостей, догостро-Бова в прикладних програмах, починаючи від пошуку тексту по шаблону і закінчуючи мережевими функціями. Підтримка бібліотек коротко полягає в наступному. У складі Python поставляється велика кількість зібраних і переносяться

функціональних можливостей, відомих як стандартна бібліотека. Крім того, Python допускає розширення, як за рахунок власних бібліотек користувача, так і за рахунок бібліотек, створених сторонніми розробниками. Наприклад, розширення NumPy позиціонується як вільних і більш потужний еквівалент системи програмування математичних обчислень Mathlab. З числа сторонніх розробок можна назвати інструменти створення веб-сайтів, програмування математичних обчислень, доступ до послідовного порту, розробку ігрових програм і багато іншого.

Python - високорівнева мова програмування загального призначеня з акцентом на продуктивність розробника і читаність коду. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліо-тека включає великий обсяг корисних функцій. Python підтримує кілька парадигм програмування, в тому числі структурний, об'єктно-орієнтоване, функціональне, і аспектно-орієнтоване. Основні архітектурні риси - динамічних-чна типізація, автоматичне керування пам'яттю, повна інтроспек-ція, механізм обробки виключень, підтримка багатопоточних обчислюваны зручні високорівневі структури даних. Код в Python організовувати в функції і класи, які можуть об'єднуватися в модулі (які в свою чергу можуть бути об'єднані в пакети). У модулях містяться деякі функції, що дозволяють налаштовувати інтерпретатор. Ці модулі є стандартними для всіх реалізацій мови, тому використання стандартних функцій допомагає зробити код стерпним на інші платформи.

Якщо говорити більш виразно, з точки зору теорії модулі голок-рають як мінімум три ролі: - повторне використання програмного коду: модулі дозволяють зберегти програмний код у вигляді файлів. На відміну від програмного кода, який вводиться в інтерактивній оболонці інтерпретатора Python і зникає безповоротно після виходу з оболонки, програмний код в фай-лах модулів зберігається постійно - його можна повторно завантажувати і запускати стільки разів, скільки буде потрібно. Можна додати, що модулі - це місце, де визначаються імена, відомі як атрибути, на які може посилатися

безліч зовнішніх клієнтів; - поділ системи простору імен: модулі в мові Python також є самою високорівневою одиницею організації програм. По суті, вони - всього лише пакети імен. Модулі дозволяють ізолювати імена в замкнуті пакети, які дозволяють уникнути конфліктів імен. Фактично, все, що знаходиться в модулі - виконуваний програмний код і створювані об'єкти, - завжди неявно включається в модулі. Внаслідок це-го модулі є природними інструментами угруповання компонен-тів системи; - реалізація служб або даних для спільного користування: з функціональної точки зору модулі також можуть використовуватися для реалізації компонентів, що використовуються системою, внаслідок чого потрібно тільки одна копія такого компонента. Наприклад, якщо необхідний глобаль-ний об'єкт, який використовується більш ніж однією функцією або модулем, можна написати його у вигляді модуля, який потім може імпортуватися безліччю клієнтів.

3.8.1 Пояснення щодо роботи моделі

У Додатку Б представлено лістинг програми моделі мережевої структури. Програма реалізована на мові Python, версія 2.7. Вибір версії мови програмування диктувався розумної достатністю. Наступна версія (Python 3), щодо програмних можливостей, задіяних для вирішення даного завдання, не містить будь-яких проривних якісно нових можливостей. Відмінності стосуються, переважно, синтаксису записи окремих операторів. Подальші версії моделі, мабуть, доцільно буде перевести на Python 3 тільки в разі переходу до багатовимірним мережевим структурам. Аргументом обґрунтування доцільності може з'явитися більш висока продуктивність і наявність спеціалізованих графічних бібліотек. У Додатку Б представлений фрагмент протоколу роботи програми моделі. Показані два запуски програми.

«Картинки» поля різняться, тому що в програмі використовуються генератори випадкових чисел. На обох «картинках» нульові елементи знаходяться в позиціях згідно вказаними координатами, відлічуваним від лівого верхнього кута «картинки».

Приложение 1
Текст программы

```
# coding: cp1251

## Однослойная клеточно-автоматная модель повреждения
## фрагмента самовосстанавливающейся сетевой структуры.
## Случайный выбор замещения. Число соседей  -   4
## Работает до появления на поле ПЕРВОГО нуля
## Построение гистограммы.
## ================================================================

## ====================  Main program  ========================

import random
import math
from copy import copy

file  = open("protocol.txt", "a")      ##  открытие протокол
print >> file, "==== Протокол ===="
n = 8                                  ## ресурс узла сети
m = 20                                 ## размер поля
print >> file "Ресурс: " n " Поле: " m
```

Рисунок 3.8 – Фрагмент коду розробленої програми

Ресурс: 10 Поле: 20

1	6	7	16	30	45	78	73	78	48	18
1	1	7	12	20	43	72	82	88	52	22
1	2	3	9	23	48	76	91	80	46	21
1	0	2	8	7	33	71	91	96	70	21
1	2	4	11	16	34	59	97	93	66	17
1	1	3	11	32	52	63	91	86	43	17
1	2	5	8	16	47	59	92	86	67	17
1	0	1	3	11	25	58	84	93	84	40
1	0	2	13	27	56	80	82	81	47	11
1	1	4	17	26	47	75	98	70	49	12
1	0	12	16	28	47	72	87	77	46	14
1	3	2	18	18	40	65	79	98	56	20
1	5	5	12	29	49	94	92	60	38	15
1	2	4	21	29	50	69	81	80	52	11
1	0	0	9	23	31	72	98	96	54	16

Рисунок 3.9 – Результати роботи з 15 гістограм

Представлені два 9-елементних вектора для побудови гістограм розподілу залишкових ресурсів. Лівий елемент відповідає нулю. Тому в обох векторах він дорівнює 1. Початкове значення ресурсу при запуску програми було зазначено 8, тому вектор містить 9 елементів: від нульового до 8-го.

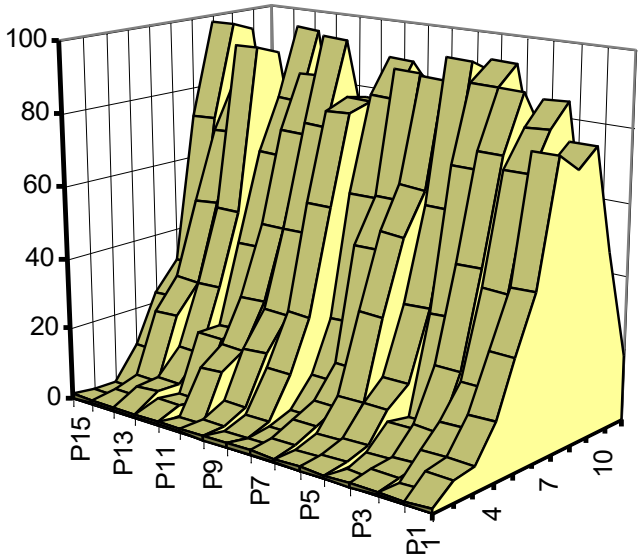

Рисунок 3.10 – Результати роботи

У Додатку Б наведено таблицю з 15 реалізацій програми і відповідний їй графік сімейства з 15 гістограм (рисунок 3.9). Рядки таблиці - вектора значень для побудови гістограм - аналогічні представленим в Додатку 2. Дані 15 запусків програми, подібні представленим в Додатку 2, були перегруповані в таблицю Додаток 3, за якою збудовано 3D-графік за допомогою Excel (рисунок 3.10). Різниця даних в Додатках Б в числі реалізованих ітерацій в кожному запуску програми, а також в початкових значеннях ресурсу (8 і 10).

ВИСНОВКИ

Найближчі можливі (і доцільні) напрямки досліджень в зв'язку з представленою моделлю можуть бути наступні:

- варіювання (збільшення) вихідного ресурсу системи;
- варіювання розміру поля;
- варіювання конфігурації (співвідношення довжина-ширина) поля.

У цих трьох напрямках може бути цікаво вивчення статистичної стійкості результатів. Очікується, що з ростом відповідних параметрів дисперсія значень буде скорочуватися. На підставі таких досліджень може бути обраний прийнятний масштаб дисперсії, в рамках якої модель можна буде більше не варіювати.

Інший можливий (цікавий) напрямок - варіювання числа допустимих нулів. Тобто можна виходити з того, що мережева структура буде зберігати функціональність при втраті $N > 1$ елементів. Відповідні залежності за розподіленням залишкових ресурсів можуть мати ту чи іншу цінність в залежності від інтерпретації результатів. По суті, це можуть бути різні предметні галузі використання моделі.

Ще одна група можливостей - комбінування втрат елементів (втрати ресурсів) із зовнішнім (регулярним або псевдовипадковим) впливом по додатковому нарощуванню (відновленню ресурсів). Подібний напрямок стосується вже не чисто деградуючих мережевих структур (з частковим самовідновленням тільки за рахунок внутрішніх резервів), а систем з частковим зовнішнім ремонтом і регламентним обслуговуванням. Це - якісно інша предметна область і інший (новий) напрямок застосування розробленої моделі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Аристов А. О. Теория квазиклеточных сетей: научная монография – М: МИСиС, 2014. –188 с.

2. Ванаг В. К. Исследование пространственно распределённых динамических систем методами вероятностного клеточного автомата (рус.) // Успехи физических наук. Обзоры актуальных проблем. : журнал. – май 1999. – Т. 169, № 5. – С. 481-505.

3. Фон Нейман Дж. Теория самовоспроизводящихся автоматов. М.: Мир, 1971.

4. Тоффоли Т., Марголус Н.Машины клеточных автоматов. М.: Мир, 1991. 280 с.

5. Бандман О.Л. Клеточно-автоматные модели пространственной динамики // Системная информатика.– 2005.– Вып. 10.– С. 57–113.

6. Бандман О.Л. Параллельная реализация клеточно-автоматных алгоритмов моделирования пространственной динамики // Сиб. журн. вычисл. математики. / РАН. Сиб. отд-ние.— Новосибирск, 2007.– Т. 10, No 4. – С. 335–348

7. Міхаль О.П., Гук А.С., Скрипник М.С. Кліттинно-автоматне моделювання динаміки мережевих структур // Проблеми інформатизації. Тези доповідей восьмої міжнародної науково-технічної конференції / Черкаси-Харків-Баку-Бельсько-Бяла, 2020 – Т1.– С.43