

ДОДАТОК А

Графічний матеріал атестаційної роботи

**Харківський національний університет
радіоелектроніки**

АТЕСТАЦІЙНА РОБОТА

«Клітинно-автоматне моделювання динаміки планарних мережевих структур»

Виконав: Скрипник Максим Сергійович

Керівник: проф. каф. ЕОМ Міхаль Олег Пилипович

Аналіз предметної області:

2

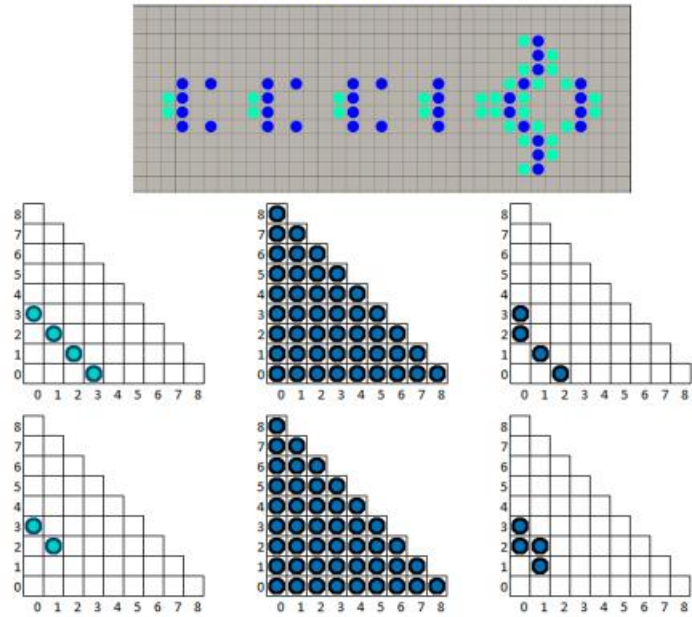
Метою атестаційної роботи є дослідження моделей і алгоритмів клітинних автоматів в контексті планарних мережевих структур.

Завдання:

- розглянути базові принципи організації клітинної структури об'єктів живої природи;
- розробити комп'ютерну модель ресурсних клітинних автоматів з використанням мови Python;
- провести аналіз планарного варіанту графів мережевої структури;
- отримати результати моделювання.

Клітинний автомат

3



Модель ресурсно-залежних клітинних автоматів. Вимоги до моделі.

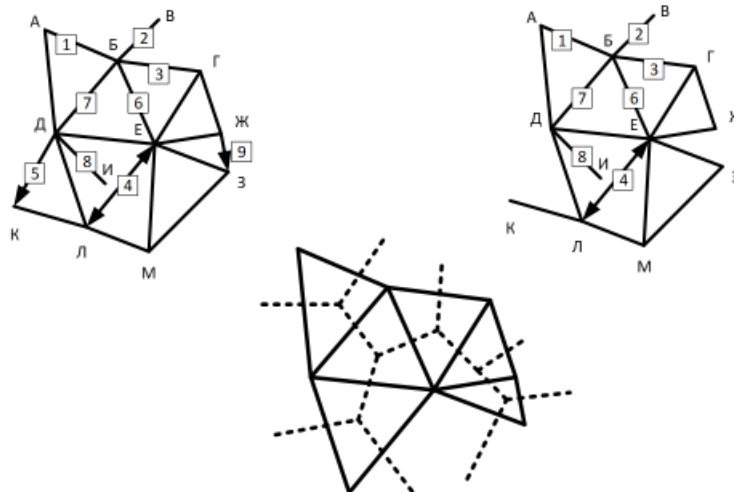
4



Модельні обмеження

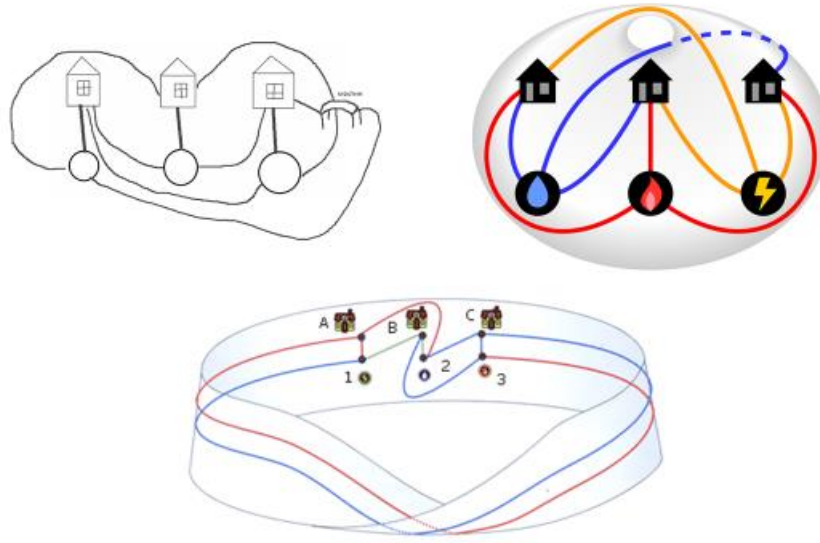
- Розглядається (моделюється) тільки двовимірний клітинний автомат. Якщо буде потрібно багатовимірне розширення - в принципі воно може бути завжди доопрацьовано при наявності двовимірного аналога. Але трудомісткість розробки, а так же самого процесу роботи з багатовимірною моделлю - зростає в квадратичній або навіть кубічній залежності. Тому, доцільністю диктується двовимірне обмеження.
- Розглядаються тільки квадратні поля КА. У відношенні реалізації, дане обмеження саме несуттєве. Але характерно, що їм задається конкретність. Якщо прийняти поле прямокутної форми, то принаймні це ще один додатковий параметр, за яким потрібно проводити дослідження.
- Розглядаються варіанти сусідства 4, 6 і 8. Вони є досить простими. В принципі, можуть бути вказані інші значення функції сусідства, але це знову призводить до збільшення параметрів моделі.
- Модель передбачається для дослідження на комп'ютерах загального призначення. Спеціальні багатопроцесорні розпаралелювальні системи використовувати не передбачається. Відповідно, в текст програми відповідні (зокрема багатопотокові) можливості не закладаються.
- Модель виконується в самодостатньому варіанті, на стандартній мові Python без будь-яких спеціальних мовних розширень.

Дуальність. Зв'язок з комп'ютерними мережами



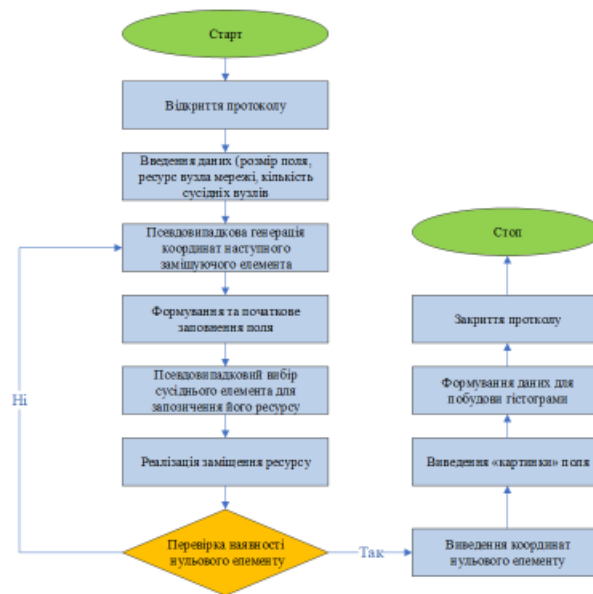
Поняття планарності

7



Блок-схема моделі

8



Розроблений алгоритм (покроковий опис)

9

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 кодї 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 змінюється. 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 використовувє 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.

Розроблений алгоритм (покроковий опис)

10

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 потічок.

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.

Мова Python. Фрагменти коду

11

```
# 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
G = [0]*(n+1)
A1 = [n]*m
A = [copy(A1) for i in range(m)]

A[yyy][xxx] -= 1
if A[yyy][xxx] == 0 :
    zz=0; x0=xxx; y0=yyy ## указание координат нуля

print >> file, "zz = "zz," Число шагов: "N, "координаты нуля: ",
x0+1, y0+1
for i in range(m):
    for j in range(m):
        for jj in range(n+1):
            if A[i][j]==jj: G[jj] += 1 ## помещене в
гистограмму
        print >> file, A[i][j],
    print >> file
print >> file, "G = "G," асоро: ",sum(G) ## выводит гистограмму

print >> file, "==== The END ====="
file.close() ## закрывает протокол
print " Protocol is generated"
```

Мова Python. Фрагменты протоколу

12

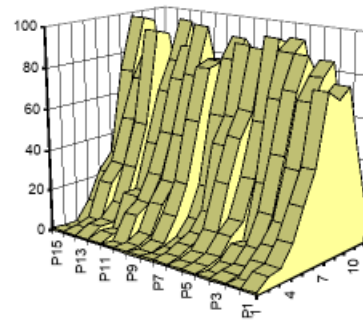
```
Протокол
Ресурс: 8 Поле: 20
zz = 0 Число шагов: 333 координаты нуля: 8 12
65665776456677867787
78787787478587447776
6757665565677555678
5747873477854467788
76557766878673346677
84347888568475887688
6675688746757476777
55557774565787658885
88877884885766667788
88766675684686357668
75677462266584475558
67787450578673665887
77878776667786878567
67756656688575775366
77678768778686877678
66766768877786568766
77886656776578447754
6666867776667766666
78776767766667887888
23576664734666778785
G = [1, 0, 3, 9, 26, 51, 106, 126, 78]
==== The END =====
```

Результати моделювання

13

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



Висновки

14

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.

ДОДАТОК Б

Программный код

```

# 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
G = [0]*(n+1)
A1 = [n]*m
A = [copy(A1) for i in range(m)]
zz = n
N = 0                                  ## число шагов до
получения первого НУЛЯ
while int (zz > 0):
    N += 1
    x=int(random.random() * m)         ## номер столбца
    y=int(random.random() * m)         ## номер строки
    x3=int(random.random() * 4)        ## функция соседства = 4
    A[y][x] -= 1                       ## [номер строки] [номер
столбца] отсчёт с нуля
    if A[y][x] == 0 :
        zz=0; x0=x; y0=y              ## указание координат нуля
    if x3 == 0:
        xx = 0; yy = -1
    elif x3 == 1:
        xx = 1; yy = 0
    elif x3 == 2:
        xx = 0; yy = 1
    else:
        xx = -1; yy = 0
    xxx = x + xx
    yyy = y + yy
    if xxx < 0 : xxx = m-1             ## РЕАЛИЗАЦИЯ функции
соседства = 4
    if yyy < 0 : yyy = m-1
    if xxx > m-1 : xxx = 0
    if yyy > m-1 : yyy = 0
    A[yyy][xxx] -= 1
    if A[yyy][xxx] == 0 :
        zz=0; x0=xxx; y0=yyy         ## указание координат нуля

print >> file, "zz = ", zz, " Число шагов: ", N, " координаты нуля: ", x0+1,
y0+1
for i in range(m):

```

```

    for j in range(m):
    for jj in range(n+1):
        if A[i][j]==jj: G[jj] += 1    ## помещение в гистограмму
    print >> file, A[i][j],
    print >> file
print >> file,"G = ",G," всего: ",sum(G)    ## ВЫВОДИТ гистограмму

print >> file, "==== The END ====="
file.close()    ##
закрывает протокол
print "        Protocol is generated"
z=0
while z != "":    ##
сигнал о завершении на дисплей
    print "======"
    z = input("        Press ENTER for exit ... ")

## ===== End of program =====

```

ДОДАТОК В

Протоколи

```

Ресурс: 8 Поле: 20
zz = 0 Число шагов: 333 координаты нуля: 8 12
6 5 6 6 5 7 7 6 4 5 6 6 7 7 8 6 7 7 8 7
7 8 7 8 7 7 8 7 4 7 8 5 8 7 4 4 7 7 7 6
6 7 5 7 6 6 5 5 6 5 6 7 7 5 5 5 5 6 7 8
5 7 4 7 8 7 3 4 7 7 8 5 4 4 6 7 7 8 7 8
7 6 5 5 7 7 6 6 8 7 8 6 7 3 3 4 6 6 7 7
8 4 3 4 7 8 8 8 5 6 8 4 7 5 8 8 7 6 8 8
6 6 7 5 6 8 8 7 4 6 7 5 7 4 7 6 7 7 7 7
5 5 5 5 7 7 7 4 5 6 5 7 8 7 6 5 8 8 8 5
8 8 8 7 7 8 8 4 8 8 5 7 6 6 6 6 7 7 8 8
8 8 7 6 6 6 7 5 6 8 4 6 8 6 3 5 7 6 6 8
7 5 6 7 7 4 6 2 2 6 6 5 8 4 4 7 5 5 5 8
6 7 7 8 7 4 5 0 5 7 8 6 7 3 6 6 5 8 8 7
7 7 8 7 8 7 7 6 6 6 7 7 8 6 8 7 8 5 6 7
6 7 7 5 6 6 5 6 6 8 8 5 7 5 7 7 5 3 6 6
7 7 6 7 8 7 6 8 7 7 8 6 8 6 8 7 7 6 7 8
6 6 7 6 6 7 6 8 8 7 7 7 8 6 5 6 8 7 6 6
7 7 8 8 6 6 5 6 7 7 6 5 7 8 4 4 7 7 5 4
6 6 6 6 6 8 6 7 7 7 6 6 6 7 7 6 6 6 6 6
7 8 7 7 6 7 6 7 7 6 6 6 6 7 8 8 7 8 8 8
2 3 5 7 6 6 6 4 7 3 4 6 6 6 7 7 8 7 8 5
G = [1, 0, 3, 9, 26, 51, 106, 126, 78]
===== The END =====
Ресурс: 8 Поле: 20
zz = 0 Число шагов: 313 координаты нуля: 5 18
6 6 5 7 4 8 6 6 6 8 8 7 7 6 6 8 7 7 7 7
7 8 8 7 2 6 7 8 5 6 7 5 8 8 7 8 8 7 7 5
8 6 6 5 6 6 6 6 7 7 7 5 5 7 7 7 7 6 6 5
8 7 7 3 6 5 7 6 7 8 4 7 7 8 7 8 7 5 7 7
8 7 8 7 7 6 8 7 8 8 5 8 7 7 6 6 4 4 6 7
8 6 6 4 4 5 7 8 8 6 7 6 5 7 7 8 7 7 7 8
7 6 7 7 8 7 7 8 7 4 4 7 7 6 6 6 6 6 6 5
8 5 5 6 8 4 4 7 6 6 7 7 7 8 7 5 6 6 6 6
4 6 7 7 7 7 7 8 8 5 5 5 8 6 7 7 8 7 7 6
5 7 7 7 6 6 7 8 7 3 5 5 8 4 6 8 5 5 6 6
8 8 7 6 8 8 8 7 8 6 8 6 6 5 5 8 7 7 5 5
7 8 7 6 7 6 7 4 6 3 4 7 6 5 5 6 5 6 5 6
8 7 8 8 7 6 3 4 7 7 6 4 6 7 7 5 5 7 6 7
8 7 8 8 6 6 5 8 5 6 7 7 8 8 8 6 8 8 6 7
5 7 8 7 7 5 5 8 7 5 6 7 6 5 6 5 7 6 6 7
7 7 6 6 8 7 6 5 8 4 7 8 6 5 3 6 6 7 8 8
7 6 6 8 6 8 7 6 8 6 7 8 7 7 2 7 8 8 7 8
8 8 5 3 0 4 7 7 7 8 7 6 6 8 6 7 8 7 6 8
5 8 4 3 6 7 7 8 8 8 8 8 5 5 5 6 7 6 7 5
6 8 4 6 8 7 5 5 7 8 7 7 5 5 5 4 6 7 7 7
G = [1, 0, 2, 7, 21, 56, 95, 129, 89]
===== The END =====

```