

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
БУДІВНИЦТВА І АРХІТЕКТУРИ**

**О.О. ТЕРЕНТЬЄВ, О.А. БОНДАР, О.В. ДОЛЯ,  
О.І. БАЛІНА, Є.В. ГОРБАТЮК, О.Ю. КУЧАНСЬКИЙ**

**ДОСЛІДЖЕННЯ ОПЕРАЦІЙ**

*Навчальний посібник для студентів вищих технічних навчальних закладів,  
які навчаються за спеціальністю 122. «Комп'ютерні науки»,  
123. «Комп'ютерна інженерія», 125. «Кібербезпека»  
126. «Інформаційні системи та технології»,  
015.39 «Професійна освіта. Цифрові технології»*

Київ – 2021 р.

УДК 519.8(075)  
ББК 30.2-5-05

Рецензенти:

*С.В. Цюцюра*, д-р техн. наук, проф., завідувач кафедру інформаційних технологій (Київського національного університету будівництва і архітектури);

*А.О. Білощизький*, д-р техн. наук, проф., проректор (Астана ІТ університет);

*С.Д. Бушуєв*, д-р техн. наук, проф., завідувач кафедру управління проєктами (Київського національного університету будівництва і архітектури)

Рекомендовано Вченою радою Київського національного університету будівництва і архітектури як навчальний посібник для студентів вищих технічних навчальних закладів, які навчаються за спеціальністю 122. «Комп'ютерні науки», 123. «Комп'ютерна інженерія», 125. «Кібербезпека», 126. «Інформаційні системи та технології», 015.39 «Професійна освіта. Цифрові технології»

**Терентьєв О.О.**

**М 79** Дослідження операцій: навчальний посібник / О.О. Терентьєв, О.А. Бондар, О.В. Доля, О.І. Баліна, Є.В. Горбатюк, О.Ю. Кучанський. – К.: 2021. –116 с.:іл.

ISBN

Навчальний посібник складається з чотирьох розділів. Розділи охоплюють розгляд теоретичних і практичних питань: математичні методи в системах прийняття рішень, основи побудови математичних моделей, лінійне програмування, оптимізація на графах. У цих розділах розглядаються оптимізаційні алгоритми, приклади розв'язання задач за цими алгоритмами, наводяться контрольні запитання з теоретичного матеріалу.

**УДК 519.8(075)**  
**ББК 30.2-5-05**

ISBN

© Терентьєв О.О., Бондар О.А., Доля О.В.,  
Баліна О.І., Горбатюк Є.В., Кучанський О.Ю.

© КНУБА, 2021

# ЗМІСТ

	Стор.
<b>Вступ</b>	3
<b>Розділ 1. Математичні моделі та методи дослідження операцій</b>	5
1.1. Математичні моделі системи підтримки прийняття рішень в задачах діагностики технічного стану будівель	5
1.2. Основні поняття і визначення. Методика проведення дослідження операцій в задачах діагностики технічного стану будівель	9
1.3 Методи побудови математичних моделей в задачах діагностики технічного стану будівель	11
Висновки по розділу 1	13
<b>Розділ 2. Лінійне програмування</b>	14
2.1. Постановка задачі	14
2.2. Канонічна форма задач лінійного програмування	15
2.3. Графічне розв'язання задачі лінійного програмування	17
2.4. Симплекс-метод	18
2.5. Двоїстість у лінійному програмуванні	28
2.6. Двоїстий симплекс-метод	30
2.7. Лінійне цілочислове програмування	33
Висновки по розділу 2	38
<b>Розділ 3. Оптимізація на мережах</b>	39
3.1. Загальні положення	39
3.2. Задача про найкоротший ланцюг. Алгоритм Дейкстри	39
3.3. Задача про багатополіусну мережу. Алгоритм Флойда	41
3.4. Задача про оптимальне дерево-кістяк	45
3.5. Задача про призначення. Угорський алгоритм	47
3.6. Задача про максимальний потік	50
Висновки по розділу 3	52

<b>Розділ 4. Моделі і методи експертних оцінок в системі підтримки прийняття рішень</b>	53
4.1 Моделі та методи формування експертної оцінки системи підтримки прийняття рішень в задачах діагностики технічного стану будівель	53
4.2 Моделі структури властивостей експертної оцінки системи підтримки прийняття рішень в задачах діагностики технічного стану будівель	62
Висновки по розділу 4	73
<b>Розділ 5. Експериментальні дослідження та реалізація експертної системи підтримки прийняття рішень в задачах діагностики технічного стану будівель</b>	74
Висновки по розділу 5	89
<b>Розділ 6. Експериментальні дослідження та реалізація нейронної мережі системи підтримки прийняття рішень в задачах діагностики технічного стану будівель</b>	89
6.1 Архітектура інформаційної системи тестування нейронної мережі	89
6.2 Підсистема підготовки інформації тестування нейронної мережі	91
6.3 Модель тестування нейронної мережі	94
6.4 Модулі програмного забезпечення тестування нейронної мережі	100
6.5. План проведення експериментального дослідження тестування нейронної мережі	102
6.6 Результати дослідження тестування нейронної мережі	106
6.7. Тестовий приклад нейронної мережі	107
Висновки по розділу 6	111
<b>Список використаної літератури</b>	112

## ВСТУП

Дослідження операцій, яке включає в себе методи математичного моделювання та оптимізації, створює фундамент прикладної математичної підготовки студентів освітнього рівня бакалавр в галузі знань 12. «Інформаційні технології» В посібнику «Дослідження операцій» викладено основні положення теорії детермінованого лінійного та нелінійного програмування; методів чисельного пошуку екстремумів опуклих функцій на опуклих множинах, методів параметричного та цілочисельного програмування, термінологія та визначення основних понять теорії дослідження операцій в обсязі достатньому для отримання компетентностей у сфері практичного використання математичних моделей детермінованих явищ. Вивчення наведеного у посібнику матеріалу призводить до формування фундаментальних теоретичних знань з математичного моделювання та оптимізації, які використовуються при дослідженні операцій, а також прикладних практичних навиків із застосування системи комп'ютерної математики пакетів програм Matlab, Mathcad, Derive, Mathematica для побудови комп'ютерних математичних моделей та кількісного розв'язання оптимізаційних задач як на попередніх етапах проектування систем, пристроїв та засобів автоматизації та інформатизації будівельного виробництва, так і у реальному часі. Посібник містить значну кількість прикладів та завдань для самостійного вивчення матеріалу, може бути використаний для вивчення курсу дослідження операцій: математичне моделювання та оптимізація в галузі інформаційних технологій; курсового та дипломного проектування.

Посібник дозволяє студентам засвоїти: термінологію, визначення, основні поняття, символічні позначення основних операцій, що використовуються в детермінованих умовах, та їх зміст; постановки задач

математичного програмування, критерії та методи технічної та економічної оптимізації складних систем, що найбільш поширені в галузі інформаційних технологій; чисельні методи в задачах лінійного та нелінійного програмування та формувати прикладні практичні навички фізичної інтерпретації результатів обчислювальних експериментів та їх візуалізації; інформаційні технології розв'язання класичних задач лінійного, нелінійного, дискретного (у тому числі і цілочисельного), програмування та дослідження операцій; прийоми формалізації задач структурно-функціонального аналізу, загальної стратегії їх розв'язання та системної оптимізації складних конструктивних елементів в галузі інформаційних технологій.

Навчальний посібник підготовлено за матеріалами лекційних курсів, які автор протягом багатьох років викладає в Київському національному університеті будівництва і архітектури на факультеті автоматизації і інформаційних технологій. Також врахований досвід проведення практичних, лабораторних та індивідуальних занять, виконання курсових робіт.

# РОЗДІЛ 1

## МАТЕМАТИЧНІ МОДЕЛІ ТА МЕТОДИ ДОСЛІДЖЕННЯ ОПЕРАЦІЙ

### 1.1 Математичні моделі системи підтримки прийняття рішень в задачах діагностики технічного стану будівель

Підвищити точність формування дій, можливо при спільному використанні накопиченої інформації, результатів поточного контролю і моніторингу стану основних конструкцій, а також результатів експертного висновку щодо ефективності заходів по забезпеченню довговічності основних конструкцій і будівель в цілому. Узагальнення різномірної інформації і отримання логічних висновків можливо на основі застосування інтелектуальних систем підтримки прийняття рішень.

Розглянемо модель прийняття рішення з управління системи діагностики конструкцій і будівлі в цілому.

Припустимо, що стан конструкції будівлі характеризує один параметр (показник властивості). Позначимо його через  $x$ . Припустимо, що параметр стану  $x$  основної конструкції доступний для спостереження і виміру інструментальними та експертними методами.

Припустимо, що для параметра  $x$  встановлена множина його станів, які складають вектор  $X = x_1, x_2, \dots, x_n$ . Наприклад, для основних конструкцій згідно нормативних документів прийнято п'ять рівнів стану – від працездатного до аварійного (ВСН 57-88(р). Положення щодо технічного обстеження житлових будівель). Нехай заходи щодо дії на досліджену конструкцію є вектор  $Y = \{y_1, y_2, \dots, y_m\}$ . Кожному із станів  $x_i (i = 1(1)n)$  повинно бути зіставлені одно або декілька можливих дій  $y_j, y_j \in Y (j = 1(1)m)$ . Відповідність технічного стану конструкції заходам  $y_j (j = 1(1)m)$

має бути задане деяким правилом  $P$ .

На рис. 1.1 встановлені експертні правила відповідності дій на конструкцію

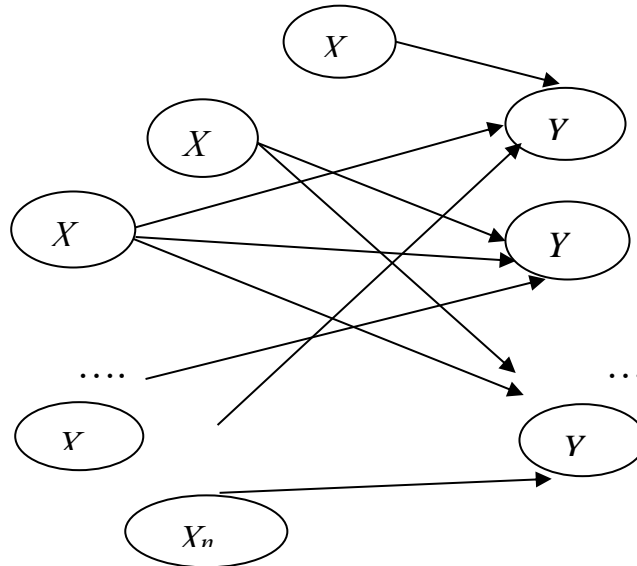


Рисунок 1.1 Експертні правила відповідності дій на конструкцію

Таким чином, можна сформулювати множину керованих дій  $U = \{u_1, u_2, \dots, u_k\}$ , до складу якої входять елементи структури вибору конкретної дії при зафіксованому в ході контролю стані  $x_f$  конструкції. Рішення системи управління  $u_i (i = 1(1)n)$  повинні формуватися системою підтримки прийняття рішень. Припустимо, що невідомість ситуації управління характеризується вектором  $\Omega = (W_1, W_2)$  умов невизначеності. Умова  $W_1$  характеризує невизначеність відповідності кожного  $u_i (i = 1(1)K)$  конкретному стану  $x$  через обмеженість початкових даних для опису зміни стану конструкції в умовах експлуатації будівлі і його реакції на можливі дії. Умова  $W_2$  характеризує невизначеність фактичного стану  $x_f$  параметру  $x$  при виконанні інструментального або експертного



контролю конструкції при моніторингу технічного стану будівлі експертами.

Врахуємо невизначеність типу  $W1$  введенням нечіткого опису технічного стану конструкції будівлі за допомогою лінгвістичної змінної  $\dot{X}$ .

Згідно виразу 1.1:

$$\dot{X} = \{X, \mu_x\}, \quad (1.1)$$

де  $\mu_x$  – функція приналежності лінгвістичної змінної  $\dot{X}$ .

У співвідношенні з виразом пропонується відповідність  $P$  знанням, що визначаються продукційними правилами типу “ЯКЩО ....., ТО”.

Таким чином, інтелектуальні системи підтримки прийняття рішень по управлінню за технічним станом конструкції представляє собою операцію імплікації:

$$u^* : (\dot{x} \rightarrow \dot{y}) / P \quad (1.2).$$

Розглянемо ситуацію, що пов’язана з фактором  $W2$ . Припустимо, що через особливості системи контролю отримати точну кількісну оцінку  $x_f$  поточного стану конструкції немає можливості. При цьому результат контролю  $q_f$  конструкції представляє собою один із термів  $q$  множини  $Q$  термів значення контрольованого параметру. Наприклад, якщо нечітке значення  $\dot{X}$  контрольованого параметру  $x$  має значення “високий”, то терми  $q$  вказаного значення можуть мати вигляд “помірно високий”, “порівняно високий”, “не дуже високий”, “дуже високий”. В цьому випадку параметр  $x$  конструкції може бути оцінений за непрямыми ознаками, наприклад, у вигляді: “значення параметру досить високе”, при цьому вказується рівень приналежності результату контролю до одного з термів. Тому для кожного терма  $q$  лінгвістичної змінної  $x$  необхідно за результатами ретроспективної інформації побудувати функцію

приналежності (ФП). При зроблених допущеннях, модель прийняття рішення, згідно використання заходу  $uf$  по управлінню за технічним станом конструкції будівлі можна представити таким чином:

$$u^*(uf) = W(qf, P, uf, / w1, w2), \quad (1.3)$$

де  $P$  – правило відповідності дії  $uf$  фактичному стану  $qf$  конструкції з урахуванням невизначеності двох типів.

В цьому випадку результат оцінки  $zf$  технічного стану конструкції по контрольованому параметру  $x$  з урахуванням двох видів невизначеності є результатом перетину двох функцій приналежності – ФП параметра  $xf$ , що характеризує продукційне правило (1.2) і ФП результату  $qf$  поточного контролю технічного стану конструкції.

Максимальне значення отриманого в ході цього об'єднання ФП нечіткої змінної  $zf$  буде представляти собою значення підсумкової ФП оцінки фактичного стану конструкції.

Цей висновок представлений співвідношенням:

$$\mu(\check{z}f) = \text{height}(\check{g}f \sqcap \check{x}f) = \sup \min(\mu(\check{g}f), \mu(\check{x}f)). \quad (1.4)$$

Значення показника, що представлено у виразі (1.4), будемо називати  $a$  – рівнем результату  $\check{z}f$  контролю стану конструкції. Після цього у відповідності з продукційним правилом (вираз 1.3) і управління  $u^*$  необхідно визначити вигляд керованої дії  $uf$  на конструкцію по параметру  $x$ . Управління  $u^*$  для конструкції повинно відповідати фактичному стану  $xf$  конструкції будівлі, що досліджена.

Таким чином, при отриманні рішення системи управління діагностики конструкції використовуються елементи штучного інтелекту, тому система що реалізовує цю процедуру може бути віднесена до класу інтелектуальних систем підтримки прийняття рішень.

## 1.2 Основні поняття і визначення. Методика проведення дослідження операцій в задачах діагностики технічного стану будівель

Термін *дослідження операцій* означає процес або послідовність операцій, які дозволяють отримати краще *розв'язання*.

Постановка задачі оптимізації передбачає визначення критерію оптимальності, параметрів оптимізації і обмежень. **Критерій оптимальності** визначає ознаки, на основі яких проводиться порівняльна оцінка допустимих розв'язків і вибір оптимального розв'язку. **Параметри оптимізації** являють собою незалежні змінні параметри, які повністю і однозначно визначають задачу, що розв'язується. **Обмеження** визначають залежності між параметрами, які повинні враховуватися при пошуку розв'язання. Критерієм оптимальності може бути вимога досягнення найбільшого або найменшого значення однією або декількома функціями параметрів оптимізації, які відображають кількісну міру досягнення мети оптимізації об'єкта, що розглядається. Кожна з таких функцій називається **цільовою**. Якщо цільова функція (ЦФ), єдина то задачу оптимізації називають **задачею математичного програмування**, а в протилежному випадку - **задачею багатокритеріальної (векторної) оптимізації**.

Задачі математичного програмування залежно від цільової функції і обмежень поділяють на такі класи:

- лінійне програмування - цільова функція і обмеження лінійні;
- нелінійне програмування - цільова функція або обмеження нелінійні;
- цілочисельне програмування - змінні можуть набувати тільки цілочислових значень;

- дискретне програмування - змінні можуть набувати тільки визначених дискретних значень із заданої множини;
- булеве програмування - змінні можуть набувати тільки два значення 0 або 1;
- параметричне програмування - коефіцієнти залежать від певного параметра;
- стохастичне програмування - цільова функція і обмеження містять випадкові параметри, які отримують певні значення із заданою імовірністю;
- геометричне програмування - цільова функція та обмеження являють собою функції спеціального вигляду-позиноми;
- динамічне програмування - багатокрокові процеси пошуку оптимального розв'язку;
- сепарабельне програмування - цільова функція являє собою суму функцій, різних для кожної змінної, обмеження можуть бути лінійні і нелінійні.

Математична постановка задачі оптимізації:

$$\begin{aligned}
 f(x_j) &\rightarrow \text{opt (min, max);} \\
 g_i(x_j) &\leq b_i; \\
 d_j &\leq x_j \leq D_j; \\
 i &= \overline{1, m}; \quad j = \overline{1, n}.
 \end{aligned}
 \tag{1.5}$$

Тут  $f(x_j)$  - цільова функція;  $g_i(x_j)$  - обмеження, що встановлюють залежності між змінними;  $x_j$  - змінні, які можуть набувати значень у певному діапазоні  $d_j$  і  $D_j$ .

Математичне програмування складається з таких етапів:

1. Вивчення системи; визначення мети, яка повинна бути досягнута внаслідок розв'язання задачі і умов, які визначають функціонування даного об'єкта або процесу; опис вхідної та вихідної інформації; визначення можливих підходів до розв'язання задачі.

2. Формалізація задачі; побудова математичної моделі (визначення критеріїв оптимальності, параметрів, обмежень).

3. Аналіз виду і структури математичної моделі; вибір методу розв'язання задачі.

4. Розв'язання задачі.

5. Аналіз результатів розв'язання.

6. Прийняття оптимального розв'язку.

### **1.3 Методи побудови математичних моделей в задачах діагностики технічного стану будівель**

*Математична модель* - система математичних співвідношень, яка описує об'єкт, що вивчається.

Для побудови математичних моделей вивчаються і аналізуються причино-наслідкові зв'язки, проводяться експерименти, використовуються аналогії. Побудова математичної моделі передбачає виконання наступних етапів:

- визначення змінних задачі;
- визначення цільової функції;
- визначення обмежень на параметри.

**Приклад.** Скласти математичну модель наступної задачі: при забудові мікрорайону використано проекти будинків трьох типів. Кількість

квартир у будівлі і вартість одного будинку для кожного з проектів задано в табл. 1.1.

Таблиця 1.1

Квартира	Тип будинку		
	1	2	3
Однокімнатна, шт.	50	50	40
Двокімнатна, шт.	80	40	60
Трикімнатна, шт.	20	30	50
Вартість будівлі, тис.грн	850	800	900

Скласти план забудівлі мікрорайону, що мінімізує витрати на будівництво, якщо потреба в одно-, дво- і трикімнатних квартирах відповідно 500, 700, 600 шт.

Визначимо змінні:

$x_j$  – кількість будівель  $j$ -го типу,  $j=1,2,3$ .

Визначимо обмеження:

$b_i$  – кількість необхідних квартир  $i$ -го типу,  $i=1,2,3$ .

Визначимо цільову функцію:

$c_j$  – вартість однієї будівлі  $j$ -го типу,  $j=1,2,3$ .

Визначимо норми використання будівель:

$a_{ij}$  – норма використання будівель  $j$ -го типу для отримання заданої кількості квартир  $i$ -го типу.

Одна будівля 1-го типу дозволяє забезпечити потребу в 50-ти однокімнатних квартирах, а  $x_1$  будівель забезпечать потребу в  $50x_1$  однокімнатних квартирах. З урахуванням того, що для інших будівель залежності аналогічні, а загальна потреба в однокімнатних квартирах повинна бути не менше 500 шт., по однокімнатних квартирах обмеження матиме вигляд:

$$50x_1 + 50x_2 + 40x_3 \geq 500.$$

Аналогічно можна скласти обмеження для інших квартир і записати залежність для цільової функції. Тоді математична модель задачі матиме вигляд:

$$850x_1 + 700x_2 + 400x_3 \rightarrow \min$$

$$50x_1 + 50x_2 + 40x_3 \geq 500$$

$$80x_1 + 40x_2 + 60x_3 \geq 700$$

$$20x_1 + 30x_2 + 50x_3 \geq 600.$$

Змінні повинні бути невід'ємні:  $x_j \geq 0$ ,  $j=1,2,3$ .

## **ВИСНОВКИ ПО РОЗДІЛУ 1**

1. Наведені математичні моделі системи підтримки прийняття рішень діагностики технічного стану конструкцій будівель.
2. Представлені основні поняття і визначення дослідження операцій.
3. Наведена методика проведення дослідження операцій.
4. Розглянуті методи побудови математичних моделей.

## РОЗДІЛ 2 ЛІНІЙНЕ ПРОГРАМУВАННЯ

### 2.1 Постановка задачі

Лінійне програмування являє собою найбільш часто використовуваний розділ оптимізації.

Задача лінійного програмування має такий вигляд:

$$\sum_{j=1}^n c_j x_j \rightarrow \max (\min); \quad (2.1)$$

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = \overline{1, r}; \quad (2.2)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{r+1, k}; \quad (2.3)$$

$$\sum_{j=1}^n a_{ij} x_j \geq b_i, \quad i = \overline{k+1, m}; \quad (2.4)$$

$$x_j \geq 0, \quad j = \overline{1, n}, \quad (2.5)$$

де  $x_j$  – змінні, які необхідно визначити і які впливають на значення ЦФ;  $b_i$  –  $i$ -й ресурс;  $a_{ij}$  – норма витрат ресурсу  $i$  на  $j$ -й виріб.

Задача містить обмеження на невід'ємність змінних, наявність якого визначається використанням симплекс-методу. Якщо при фізичній постановці задачі яка-небудь змінна, наприклад  $x_n$  є необмеженою по знаку, то її можна представити у вигляді  $x_n = x_n^{(1)} - x_n^{(2)}$ , де  $x_n^{(1)}, x_n^{(2)} \geq 0$ . Вважатимемо всі числа  $b_i \geq 0$ ,  $i = 1, \dots, m$ . Цього можна добитися множенням обмеження  $b_i \leq 0$ , на "–1".



Задачу лінійного програмування (ЗЛП) можна розв'язувати аналітичними і графічними методами. До аналітичних методів відносять метод Гаусса-Жордана і симплекс-метод. Графічне розв'язання ЗЛП достатньо наочне, але обмежене кількістю змінних. Спочатку ЗЛП зводять до канонічної форми, а потім розв'язують її.

## 2.2 Канонічна форма задач лінійного програмування

Канонічна форма задач лінійного програмування має такий вигляд:

$$\sum_{j=1}^n c_j x_j \rightarrow \max (\min);$$

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = \overline{1, m};$$

$$x_j \geq 0, \quad j = \overline{1, n}.$$

Задача лінійного програмування зводиться до канонічної форми шляхом перетворення всіх обмежень, які мають вигляд нерівностей, в рівності. Це досягається за такими правилами (для задачі максимізації):

- для кожного обмеження вигляду (2.3) додається нова базисна змінна;
- для кожного обмеження вигляду (2.2) додається нова базисна змінна і ця змінна вводиться у ЦФ з коефіцієнтом "-M" (M-задача);
- для кожного обмеження вигляду (2.4) додаються дві змінні  $x_j - x_i$  ( $x_i$  – для того щоб отримати рівність, а  $x_j$  – нова базисна змінна, яка вводиться у ЦФ з коефіцієнтом "-M" (M-задача).

Таким чином, для зведення ЗЛП до канонічного вигляду необхідно ввести в обмеження  $t$  базисних змінних. Задачу мінімізації можна або звести попередньо до задачі максимізації (шляхом множення цільової функції на "-1"), або змінні вводяться у цільову функцію з коефіцієнтом "+M".

**Приклад 1.** Звести до канонічного вигляду таку систему лінійних рівнянь:

$$4x_1 + 5x_2 \rightarrow \max$$

$$2x_1 + x_2 \leq 2$$

$$3x_1 + 4x_2 \geq 5$$

$$x_1 + x_2 = 3$$

$$x_1, x_2 \geq 0.$$

Задача лінійного програмування має канонічний вигляд:

$$4x_1 + 5x_2 - Mx_5 - Mx_6 \rightarrow \max$$

$$2x_1 - x_2 + x_3 = 2$$

$$3x_1 + 4x_2 - x_4 + x_5 = 5$$

$$x_1 + x_2 + x_6 = 3$$

$$x_i \geq 0, \quad i = \overline{1,6}.$$

**Приклад 2.** Звести до канонічного вигляду таку систему лінійних рівнянь:

$$4x_1 + 5x_2 \rightarrow \min$$

$$2x_1 - x_2 \leq 2$$

$$3x_1 + 4x_2 \geq 5$$

$$x_1 + x_2 = 3$$

$$x_1, x_2 \geq 0.$$

Задача лінійного програмування має канонічний вигляд:

$$4x_1 + 5x_2 + Mx_5 + Mx_6 \rightarrow \min$$

$$2x_1 - x_2 + x_3 = 2$$

$$3x_1 + 4x_2 - x_4 + x_5 = 5$$

$$x_1 + x_2 + x_6 = 3$$

$$x_i \geq 0, \quad i = \overline{1,6}.$$

### 2.3 Графічне розв'язання задачі лінійного програмування

**Постановка задачі.** Графічно можна розв'язувати задачі з обмеженнями типу рівностей, якщо кількість обмежень на одиницю або на два менша числа змінних ( $n - m \leq 2$ ). У випадку двох змінних ЗЛП має просту геометричну інтерпретацію і може бути розв'язана за допомогою такого алгоритму.

#### **Графічний метод розв'язування ЗЛП.**

1. Побудувати область допустимих розв'язків (ОДР). В загальному випадку вона являє собою опуклий багатокутник. Якщо обмеження в задачі несумісні, то множина допустимих розв'язків є пустою множиною, а задача пошуку екстремуму не має сенсу.

2. Знайти градієнт цільової функції. Градієнт може бути побудований у будь-якій точці координатної площини (як правило, його будують на початку координат).

3. Провести лінію рівня функції, перпендикулярну до градієнта.

4. Перемістити лінію рівня паралельно самій собі до дотику з множиною допустимих розв'язків. Точки дотику є точками екстремуму.

5. Класифікувати точки дотику з використанням властивостей градієнта.

При розв'язуванні ЗЛП можливі такі типові ситуації:

- задача має єдиний розв'язок;
- задача має нескінчену множину розв'язків;
- задача не має розв'язку.

**Приклад.** Розв'язати графічно задачу лінійного програмування.

$$2x_1 - x_2 \rightarrow \max$$

$$-x_1 + x_2 \leq 3$$

$$6x_1 + 7x_2 \leq 42$$

$$2x_1 - 3x_2 \leq 6$$

$$x_1, x_2 \geq 0.$$

Лінії ОДР будують для кожної нерівності по двох точках, шляхом прирівнювання однієї зі змінних до нуля і визначення положення іншої координати. Областю допустимих розв'язків є заштрихований багатокутник (рис. 2.1). Цільову функцію будують зведенням формули до рівності певному значенню (в даному випадку до 10). Градієнт означає напрямок найшвидшого зростання функції; пересування лінії ЦФ вказує на точку розв'язку – точку В.

Розв'язки задачі:  $x_1 = 5,25$ ;  $x_2 = 1,5$ ;  $z = 9$ .

## 2.4 Симплекс-метод

**Постановка задачі.** Головна ідея симплекс-методу полягає в цілеспрямованому послідовному знаходженні екстремальних точок за допомогою допустимих базисних розв'язків системи обмежень. Змінні  $x_1, x_2, \dots, x_m$  називаються **базисними**, якщо вони входять тільки в одне з рівнянь

системи обмежень з коефіцієнтом 1, а в усі інші рівняння з коефіцієнтами 0. Всі інші  $n-t$  змінних називаються *небазисними (вільними)*. *Базисним розв'язком* ЗЛП називається розв'язання:

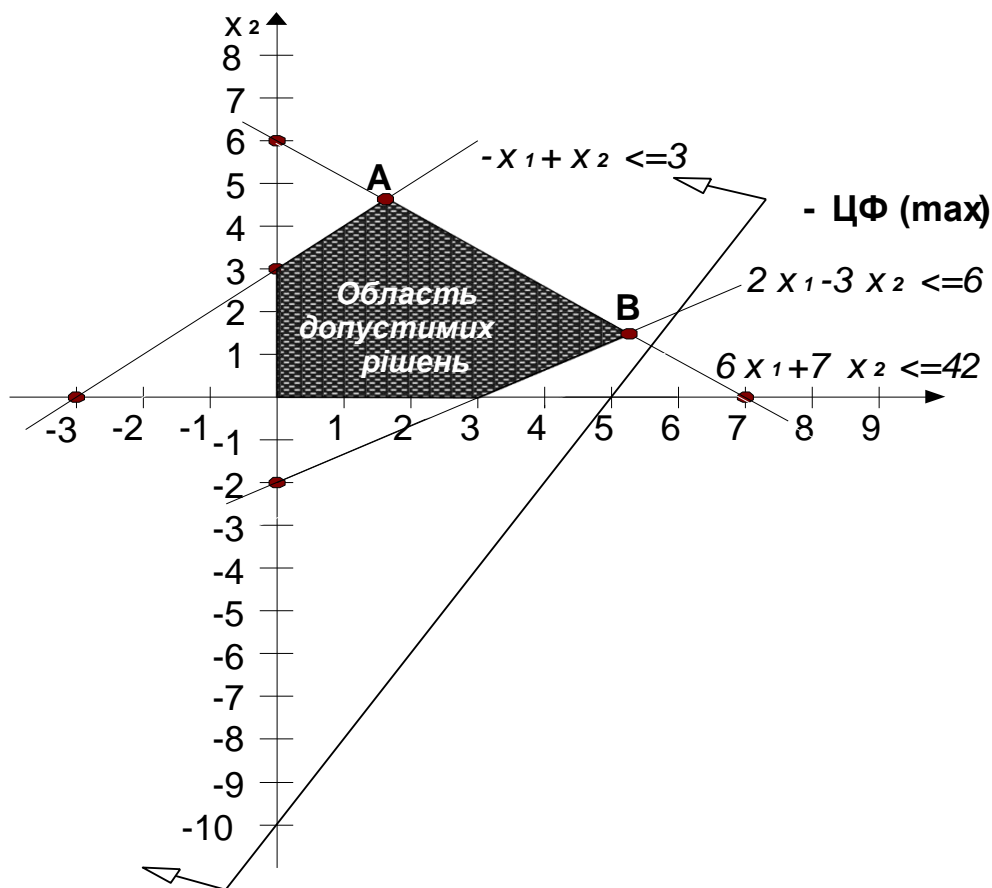


Рисунок 2.1 Графічне розв'язання задачі лінійного програмування

$$x_i = b_i, i = 1, \dots, m; \quad x_{m+1} = x_{m+2} = \dots = x_n = 0.$$

Базисний розв'язок називається *допустимим*, якщо  $x_i \geq 0, i = 1, \dots, m$ .

Базисний розв'язок називається *невиродженим*, якщо  $x_i > 0, i = 1, \dots, m$ .

Процес розв'язування починається з деякої початкової екстремальної точки (допустимий базисний розв'язок) і продовжується доти, поки не буде знайдена точка, яка відповідає оптимальному розв'язку. Як нову екстремальну точку обирають суміжну з попередньою екстремальну точку, в якій значення ЦФ щонайменше не погіршується. Максимальна кількість ітерацій не перевищує  $C_n^m$  (кількість сполучень із  $n$  по  $m$ ).

Суміжні екстремальні точки відрізняються тільки однією змінною. Кожна наступна екстремальна точка визначається заміною однієї з поточних базисних змінних новою з числа небазисних.

Симплекс-метод передбачає виконання таких операцій:

- визначення початкового допустимого базисного розв'язку, при цьому  $(n-m)$  небазисних змінних прирівнюються нулю і з цільової функції виключають базисні змінні;

- визначення з числа небазисних змінних змінної, яку необхідно включити в число базисних змінних, ця нова базисна змінна повинна покращити значення цільової функції;

- визначення з числа базисних змінних змінної, яка повинна бути виключена з базисного розв'язку, тобто замінена новою базисною змінною.

Виконання розрахунків пов'язане з використанням симплекс-таблиць (табл. 2.1).

Таблиця 2.1

**Симплекс-таблиця**

Базис	$C_{\text{баз}}$	БР	$c_1$	$c_2$	...	$c_n$	$c_{n+1}$	...	$c_{n+m}$
			$A_1$	$A_2$	...	$A_n$	$A_{n+1}$	...	$A_{n+m}$
$A_{n+1}$	$c_{n+1}$	$b_1$	$a_{11}$	$a_{12}$	...	$a_{1n}$	$a_{1n+1}$	...	$a_{1n+m}$
$A_{n+2}$	$c_{n+2}$	$b_2$	$a_{21}$	$a_{22}$	...	$a_{2n}$	$a_{2n+1}$	...	$a_{2n+m}$
...	...	...	...	...	...	...	...	...	...
$A_{n+m}$	$c_{n+m}$	$b_m$	$a_{m1}$	$a_{m2}$	...	$a_{mn}$	$a_{mn+1}$	...	$a_{mn+m}$
$\Delta_j = z_j - c_j$		$Z_0$	$Z_1 - C_1$	$Z_2 - C_2$	...	$Z_n - C_n$	$Z_{n+1} - C_{n+1}$	...	$Z_{n+m} - C_{n+m}$

БР – базисні розв'язки.

Значення  $z_j$  визначаються так:

$$z_j = \sum_{i=1}^m c_{iB} a_{ij}, \quad j=1, m+n,$$

де  $c_{iB}$  – коефіцієнти ЦФ при поточних базисних змінних;  $i_B$  – індекс базисної змінної, що розташована в  $i$ -му рядку симплекс-таблиці;  $a_{ij}$  – елементи стовпця коефіцієнтів при змінній  $x_j$  в системі рівнянь, що відповідає поточному базису.

Алгоритм розв’язання задач симплекс-методом містить такі етапи (для задачі мінімізації):

**Крок 1.** Знайти початковий базисний розв’язок:

- записати вихідну канонічну задачу (простий симплексметод або  $M$ -задача);
- визначити базисні змінні, які входять тільки в одне з рівнянь системи з коефіцієнтом 1, а в усі інші - з коефіцієнтом, що дорівнює 0;
- виділити вільні змінні (всі інші, крім базисних);
- знайти початковий базисний розв’язок, вважаючи, що вільні змінні дорівнюють 0.

**Крок 2.** Заповнити симплекс-таблицю.

Обчислити відносні оцінки і записати їх у таблицю.

$$\Delta_j = \sum_{i=1}^m c_{iB} a_{ij} - c_j = z_j - c_j,$$

$$z_j = \sum_{i=1}^m c_{iB} a_{ij}; \quad j = 1, m+n.$$

Для базисних змінних оцінки дорівнюють 0. Це можна використовувати як для перевірки правильності заповнення таблиці, так і для скорочення обчислень.

**Крок 3.** Проаналізувати відносні оцінки.

1. Якщо всі оцінки  $\Delta_j$  недодатні, тобто

$$\Delta_j \leq 0, \quad j = \overline{1, m+n},$$

то розрахунок закінчений і необхідно знайти отриманий базисний розв'язок. Значення базисних змінних знаходяться в стовпці БР, а інші змінні вважаються рівними 0, як вільні.

Проаналізувати отриманий базисний розв'язок:

- якщо число нульових оцінок  $\Delta_j = 0$  дорівнює числу базисних змінних, то задача має **єдиний розв'язок**;
- якщо число нульових оцінок  $\Delta_j = 0$  перевищує число базисних змінних, то задача має **нескінченну безліч розв'язків**;
- якщо усі  $\Delta_j$  – недодатні, але базисний розв'язок містить хоча б одну штучну змінну, яка не дорівнює 0, то **обмеження задачі несумісні**.

2. Якщо серед оцінок є додатні, то необхідно знайти серед них максимальну:

$$\Delta_r = \max \Delta_j,$$

і проаналізувати коефіцієнти стовпця таблиці, якому відповідає максимальна додатна оцінка (якщо таких оцінок декілька, то прийнято вибирати оцінку з найменшим номером). Якщо цей стовпець містить хоча б один додатний коефіцієнт, то номер стовпця позначається через  $r$ , і змінна, яка йому відповідає, повинна бути введена в число базисних. Якщо серед коефіцієнтів цього стовпця немає жодного додатного коефіцієнта, то це означає, що множина допустимих розв'язків задачі є необмеженою, і **задача розв'язку не має**.

Стовпець, що відповідає обраній оцінці, називається **розв'язувальним**.



**Крок 4.** Поділити елементи стовпця базисних розв'язків на відповідні елементи розв'язувального стовпця і серед отриманих часток вибрати найменшу. Рядок, що відповідає обраному відношенню називається **розв'язувальним**.

Таким чином, нова змінна  $x_r$  вводиться на місце змінної  $x_{S_B}$ , що вилючається з числа базисних, номер змінної  $S_B$ , а також номер  $S$  відповідного рядка таблиці, визначаються з умови

$$\min_{1 \leq i \leq m} [X_{iS} / a_{ir}] = [X_{S_B} / a_{Sr}]$$

де  $X_{S_B}$  – значення координати поточного базисного розв'язку, що відповідає  $i$ -му рядку;  $a_{ir}$  – коефіцієнт при координаті  $x_r$  в  $i$ -му рядку. Якщо таких змінних виявиться більше однієї, то з базису виводиться та змінна, яка має більший номер. Розглядаються тільки невід'ємні відношення, тобто якщо коефіцієнт  $a_{ir}$  – від'ємний або дорівнює 0, то відношення не підраховується.

Елемент  $a_{Sr}$ , розташований на перетинанні розв'язувального рядка і розв'язувального стовпця, називається **розв'язувальним** і виділяється.

З числа базисних виводиться змінна, що відповідає розв'язувальному рядку, а на її місце вводиться змінна, що відповідає розв'язувальному стовпцю.

**Крок 5.** Обчислити новий базисний розв'язок, здійснивши перерахування таблиці:

1. Замість координати  $X_{S_B}$  до складу базисних увести координату  $x_r$ , значення якої визначається за формулою

$$x_r = X_{S_B} / a_{Sr}$$

і перерахувати  $s$ -й рядок, у якому відбулися зміни по базису:

$$a_{sj} = a_{sj} / a_{sr}, \quad \overline{j = 1, m+n.}$$

Таким чином, кожен елемент розв'язувального рядка ділиться на розв'язувальний елемент  $a_{sr}$ ;

2. Обчислити всі інші коефіцієнти:  $a_{sj}$

$$a_{ij} = a_{ij} - a_{sj} / a_{sr} a_{ir},$$

$$i=1, \dots, m; \quad i \neq s; \quad \overline{j=1, m+n.}$$

Новий базисний розв'язок визначити на основі поточного базисного розв'язку за формулами

$$x_{iB} = x_{iB} - a_{ir} x_r, \quad i_B \neq s_B.$$

Для спрощення обчислень за наведеними формулами використовується "правило прямокутника".

Нехай підраховується значення  $a_{ij}$ . Варто з'єднати елемент  $a_{ij}$  у попередній таблиці з розв'язувальним елементом  $a_{sr}$ . Отримано одну з діагоналей прямокутника. Другу діагональ утворить з'єднання елементів  $a_{ir}$  і  $a_{sj}$ . Далі з поточного значення  $a_{ij}$  віднімається добуток елементів  $a_{ir}$  і  $a_{sj}$ , поділений на розв'язувальний елемент  $a_{sr}$ .

$$a_{ij} = a_{ij} - \frac{a_{sj} a_{ir}}{a_{sr}}$$

Перейти до кроку 3.

**Приклад 1.** Розв'язати симплекс-методом ЗЛП.

$$2x_1 - x_2 \rightarrow \max$$

$$-x_1 + x_2 \leq 3;$$

$$6x_1 + 7x_2 \leq 42$$

$$2x_1 - 3x_2 \leq 6$$

$$x_i \geq 0, \quad i = 1, 2.$$

1. Зведемо до канонічного вигляду:

$$-x_1 + x_2 + x_3 = 3$$

$$6x_1 + 7x_2 + x_4 = 42$$

$$2x_1 - 3x_2 + x_5 = 6$$

$$x_i \geq 0, i = \overline{1,5}.$$

де  $x_3, x_4, x_5$  – додаткові змінні, додані в нерівність, щоб перетворити їх у рівність.

2. Результати розв'язування представлені в табл.2.2–2.4.

Таблиця 2.2

Базис	Сбаз	БР	2	-1	0	0	0
			A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>4</sub>	A <sub>5</sub>
A <sub>3</sub>	0	3	-1	1	1	0	0
A <sub>4</sub>	0	42	6	7	0	1	0
A <sub>5</sub>	0	6	2	-3	0	0	1
$\Delta_j = Z_j - C_j$		0	-2	1	0	0	0

Таблиця 2.3

Базис	Сбаз	БР	2	-1	0	0	0
			A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>4</sub>	A <sub>5</sub>
A <sub>3</sub>	0	6	0	-0,5	1	0	0,5
A <sub>4</sub>	0	24	0	16	0	1	-3
A <sub>1</sub>	-2	3	1	-1,5	0	0	0,5
$\Delta_j = Z_j - C_j$		6	0	-3	0	0	1

Таблиця 2.4

Базис	Сбаз	БР	2	-1	0	0	0
			A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>4</sub>	A <sub>5</sub>
A <sub>3</sub>	0	6,75	0	0	1	0,031	0,406

A <sub>2</sub>	-1	1,5	0	1	0	0,063	-0,19
A <sub>1</sub>	2	5,25	1	0	0	0,094	0,22
$\Delta_j = Z_j - C_j$		9	0	0	0	0,125	0,625

Згідно з алгоритмом розв'язування задачі знайдено:  $x_1 = 5,25$ ;  $x_2 = 1,5$ ;  $x_3 = 6,75$ ;  $x_4 = 0$ ;  $x_5 = 0$ . Значення цільової функції:  $z = 9$ .

**Приклад 2.** Розв'язати симплекс-методом  $M$ -задачу лінійного програмування.

$$5x_1 + 2x_2 - x_3 \rightarrow \max$$

$$- 2x_1 - x_2 - x_3 \geq -5$$

$$3x_1 + 2x_2 + x_3 = 6$$

$$5x_1 + 3x_2 + 4x_3 \geq 1$$

$$x_i \geq 0, \quad i = \overline{1,3}.$$

1. Зведемо до канонічного вигляду:

$$5x_1 + 2x_2 - x_3 - Mx_6 - Mx_7 \rightarrow \max$$

$$2x_1 + x_2 + x_3 + x_4 = 5$$

$$3x_1 + 2x_2 + x_3 + x_6 = 6$$

$$5x_1 + 3x_2 + 4x_3 - x_5 + x_7 = 1$$

$$x_i \geq 0, \quad i = \overline{1,3}.$$

де  $x_4, x_5, x_6, x_7$  – додаткові змінні, додані в нерівності, щоб перетворити їх у рівності:  $x_4, x_6, x_7$  – початкові, допустимі базисні розв'язки.

2. Результати розв'язування представлені в табл. 2.5–2.7.

Таблиця 2.5

Базис	$C_{\text{баз}}$	БР	5	2	-1	0	0	-M	-M
			$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	$A_7$
$A_4$	0	5	2	1	1	1	0	0	0
$A_6$	-M	6	3	2	1	0	0	1	0
$A_7$	-M	1	5	3	4	0	-1	0	1
$\Delta_j = Z_j - C_j$		-7M	-5 -8M	-2 -5M	1-5M	0	M	0	0

Таблиця 2.6

Базис	$C_{\text{баз}}$	БР	5	2	-1	0	0	-M	-M
			$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	$A_7$
$A_4$	0	23/5	0	-1/5	-3/5	1	2/5	0	-2/5
$A_6$	-M	27/5	0	1/5	-7/5	0	3/5	1	-3/5
$A_1$	5	1/5	1	3/5	4/5	0	-1/5	1	1/5
$\Delta_j = Z_j - C_j$		27M/5 +1	0	- M/5+1	7M/5 +5	0	1 - 3M/5	0	8M/5 +1

Таблиця 2.7

Базис	$C_{\text{баз}}$	БР	5	2	-1	0	0	-M	-M
			$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	$A_7$
$A_4$	0	1	0	-1/3	1/3	1	0	-2/3	0
$A_5$	0	9	0	1/3	-7/3	0	1	5/3	-1
$A_1$	5	2	1	2/3	1/3	0	0	1/3	0
$\Delta_j = Z_j - C_j$		10	0	4/3	8/3	0	0	-5/3 +M	2+M

Згідно з алгоритмом розв'язування задачі знайдено:  $x_1 = 2$ ;  $x_2 = 0$ ;  $x_3 = 0$ ;  $x_4 = 1$ ;  $x_5 = 9$ ;  $x_6 = 0$ ;  $x_7 = 0$ . Значення ЦФ:  $z = 10$ .

## 2.5 Двоїстість у лінійному програмуванні

Кожній вихідній (прямій) ЗЛП відповідає *двоїста задача*. Розв'язання однієї задачі є одночасно розв'язанням і іншої. Поняття двоїстості використовується при дослідженні ЗЛП на чутливість, дозволяє дати інтерпретацію отриманим розв'язкам, розкрити нові закономірності і співвідношення, які властиві проблемі, що досліджується. Змінні і обмеження двоїстої задачі формуються шляхом симетричних структурних перетворень прямої задачі за такими правилами.

1. Кожному  $i$ -му обмеженню прямої задачі відповідає змінна  $y_i$  двоїстої задачі і, навпаки, кожному  $j$ -му обмеженню двоїстої задачі відповідає змінна  $x_j$  прямої задачі.

2. Матриця коефіцієнтів при двоїстих змінних в обмеженнях двоїстої задачі є транспонованою матрицею коефіцієнтів при змінних в обмеженнях вихідної задачі.

3. Вільні члени обмежень однієї із задач є коефіцієнтами при відповідних змінних у цільовій функції іншої задачі; максимізація змінюється на мінімізацію і навпаки.

4. Якщо в прямій задачі обмеження мають знаки нерівностей  $\leq$ , то в двоїстій задачі вони будуть  $\geq$ .

Для прямої ЗЛП такого виду:

$$\sum_{j=1}^n c_j x_j \rightarrow \max;$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{1, m};$$

$$x_j \geq 0, \quad j = \overline{1, n}.$$

існує двоїста ЗЛП (ДЗЛП):

$$\sum_{i=1}^m b_i y_i \rightarrow \min;$$

$$\sum_{i=1}^m a_{ij} y_i \leq b_j, \quad j = \overline{1, n};$$

$$y_i \geq 0, \quad i = \overline{1, m}.$$

**Приклад.** Звести пряму ЗЛП до ДЗЛП:

$$x_1 - 4x_2 - 3x_3 \rightarrow \min$$

$$3x_1 + 4x_2 + x_3 \leq 7$$

$$x_1 + 2x_2 + x_3 = 6$$

$$x_3 \geq 4$$

$$x_j \geq 0, \quad j = \overline{1, 3}.$$

Виконаємо попередні перетворення задачі :

- задачу мінімізації перетворимо на задачу максимізації;
- знаки " $\geq$ " замінимо на знаки " $\leq$ ";
- знаки "=" замінимо на знаки " $\leq$ " і замість одного рівняння введемо двое.

$$-x_1 + 4x_2 + 3x_3 \rightarrow \max$$

$$3x_1 + 4x_2 + x_3 \leq 7$$

$$x_1 + 2x_2 + x_3 \leq 6$$

$$-x_1 - 2x_2 - x_3 \leq -6$$

$$-x_3 \leq -4$$

$$x_j \geq 0, \quad j = \overline{1, 3}.$$

Побудуємо двоїсту задачу до прямої за вищенаведеними правилами:

$$7y_1 + 6y_2 - 6y_3 - 4y_4 \rightarrow \min$$

$$3y_1 + y_2 - y_3 \geq -1$$

$$4y_1 + 2y_2 - 2y_3 \geq 4$$

$$y_1 + y_2 - y_3 - y_4 \geq 3$$

$$y_i \geq 0, \quad i = \overline{1, 4}.$$

## 2.6 Двоїстий симплекс-метод

В основі двоїстого симплекс-методу (ДСМ) лежить ідея побудови на кожній ітерації, за винятком останньої, розв'язку, який є недопустимим внаслідок невиконання умов невід'ємності змінних; при цьому відповідні розв'язки двоїстої задачі на кожній ітерації є допустимими. ДСМ - це застосування симплекс-методу до двоїстої задачі, доповнене побудовою на кожній ітерації вектора змінних, який є майже допустимим опорним розв'язком вихідної задачі. ДСМ використовується при розв'язуванні ЗЛП у таких випадках:

- початковий опорний допустимий базисний розв'язок не є очевидним (наприклад, метод штучних змінних);
- після отримання розв'язків необхідно додати до вихідних обмежень додаткові обмеження (наприклад, розв'язки задач лінійного цілочисельного програмування).

**Базисом опорного плану**  $Y = (y_1, y_2, \dots, y_m)$ , або **сполученим базисом**, називається система  $m$  незалежних векторів обмежень прямої задачі, для яких  $m$  обмежень двоїстої задачі виконуються як строгі рівності.

Вектор  $X = (x_1, x_2, \dots, x_n)$  компоненти якого дорівнюють коефіцієнтам розкладу змінних  $x$  по векторах сполученого базису називається **псевдопланом вихідної задачі**. Псевдоплан  $X$  буде оптимальним розв'язком вихідної задачі, якщо серед його базисних компонентів немає від'ємних. План  $Y$ , який визначає псевдоплан  $X$ , є розв'язком двоїстої задачі.

**Алгоритм двоїстого симплекс-методу.**

**Крок 1.** Представити вихідну задачу лінійного програмування у канонічному вигляді.

**Крок 2.** Записати двоїсту задачу до заданої задачі.

**Крок 3.** Визначити базис опорного плану для двоїстої задачі.

**Крок 4.** Визначити коефіцієнти розкладу небазисних векторів по векторах сполученого базису.

**Крок 5.** Записати псевдоплан  $X = (x_1, x_2, \dots, x_n)$ .



**Крок 6.** Перевірити, чи існують серед компонентів псевдоплану  $X = (x_1, x_2, \dots, x_n)$  від'ємні. Якщо їх не існує, то план є оптимальним, інакше переходимо до кроку 7.

**Крок 7.** Перевірити в рядках псевдоплану, які відповідають його від'ємним компонентам  $b_i$ , чи всі елементи  $a_{ij} \geq 0$ , для  $j=1, \dots, n$ . Якщо так, то задача не має розв'язку. В протилежному випадку переходимо до кроку 8.

**Крок 8.** Вибрати розв'язувальний рядок, тобто вектор, який буде виведено з базису. Якщо існує декілька  $b_i \leq 0$ , то обирається рядок з найменшим значенням  $b_i$ .

**Крок 9.** Вибрати розв'язувальний стовпець, тобто вектор який буде введено в базис. Номер цього стовпця визначається із співвідношення:

$$\min \{-\Delta_j / a_{ij} / a_{ij} \leq 0\}$$

**Крок 10.** Виконуються симплексні перетворення і перехід до кроку 6.

**Приклад.** Розв'язати задачу лінійного програмування двоїтим симплекс-методом.

$$2x_1 + x_2 \rightarrow \max$$

$$x_1 + x_2 \leq 6$$

$$2x_1 + 8x_2 \geq 16$$

$$15x_1 - 3x_2 \geq 15$$

$$x_i \geq 0, i = \overline{1, 2}.$$

Запишемо задачу в канонічному вигляді:

$$2x_1 + x_2 \rightarrow \max$$

$$x_1 + x_2 + x_3 = 6$$

$$- 2x_1 - 8x_2 + x_4 = - 16$$

$$- 15x_1 + 3x_2 + x_5 = - 15$$

$$x_i \geq 0, i = \overline{1, 5}.$$

Двоїста задача до заданої задачі:

$$6y_1 - 16y_2 - 15y_3 \rightarrow \min$$

$$y_1 - 2y_2 - 15y_3 \geq 2$$

$$y_1 - 8y_2 + 3y_3 \geq 1$$

$$y_j \geq 0, j = 1, 3.$$

Псевдоплан має вигляд  $X = (0, 0, 6, -16, -15)$ .

Заповнимо симплекс-таблицю і виконаємо розв'язання згідно з алгоритмом. Результати розв'язання наведені в табл. 2.8–2.10.

Таблиця 2.8

Базис	C <sub>баз</sub>	БР	2	1	0	0	0
			A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>4</sub>	A <sub>5</sub>
A <sub>3</sub>	0	6	1	1	1	0	0
A <sub>4</sub>	0	-16	-2	-8	0	1	0
A <sub>5</sub>	0	-15	-15	3	0	0	1
$\Delta_j = Z_j - C_j$		0	-2	-1	0	0	0

На першій ітерації задача розв'язується симплексметодом. Розв'язувальний елемент дорівнює 1.

Таблиця 2.9

Базис	C <sub>баз</sub>	БР	2	1	0	0	0
			A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>4</sub>	A <sub>5</sub>
A <sub>1</sub>	2	6	1	1	1	0	0
A <sub>4</sub>	0	-4	0	-6	2	1	0
A <sub>5</sub>	0	75	0	18	16	0	1
$\Delta_j = Z_j - C_j$		12	0	1	2	0	0

На другій ітерації задача розв'язується двоїстим симплекс-методом.

Таблиця 2.10

Базис	C <sub>баз</sub>	БР	2	1	0	0	0
			A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>4</sub>	A <sub>5</sub>
A <sub>1</sub>	2	32/6	1	0	4/3	1/6	0
A <sub>2</sub>	1	2/3	1	1	-1/3	-1/6	0
A <sub>5</sub>	0	63	0	0	22	3	1
$\Delta_j = Z_j - C_j$		34/3	0	0	7/3	1/6	0

Розв'язки:  $X = (32/6, 2/3, 0, 0, 63)$ ,  $Y = (7/3, 1/6, 0)$ ,  $z = 34/3$ .

## 2.7 Лінійне цілочислове програмування

Задача цілочислового лінійного програмування (ЦЛП) описується моделлю такого вигляду:

$$\sum_{j=1}^n c_j x_j \rightarrow \max (\min);$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{1, m};$$

$$x_j \geq 0, \quad j = \overline{1, n}.$$

$$x_j - \text{цілі числа}, \quad j = \overline{1, \dots, n}$$

Округлення результатів розв'язання задачі може привести або до недопустимого розв'язку, або до результатів, які суттєво відрізняються від оптимальних.

Всі методи розв'язування задач ЦЛП поділяються на дві групи: *методи відтинання і комбінаторні методи*. В методах відтинання на кожній ітерації додається лінійне обмеження, яке задовольняє цілочислового розв'язанню вихідної задачі і виключає поточне нецілочислове розв'язання попередньої ітерації. Збіжність досягається по

кінцевій кількості ітерацій, але це число може бути дуже великим. Серед методів відтинання найбільш відомі *перший алгоритм Гоморі* (для розв'язування повністю цілочислових задач) і *другий алгоритм Гоморі* (для розв'язування частково цілочислових задач). Всі *комбінаторні методи* базуються на ідеї перебору всіх допустимих цілочислових розв'язків. Серед комбінаторних методів найбільш відомим є *метод гілок і границь*.

*Перший алгоритм Гоморі* передбачає побудову відтинання за такими правилами. Рівняння

$$\sum_{j=1}^n a_{ij} x_j = b_i,$$

яке містить дробові значення базисних змінних  $x_i = b_i$ , перетворюється у додаткову нерівність:

$$\sum_{j=1}^n \alpha_{ij} x_j = \beta_i,$$

де  $\alpha_{ij} = a_{ij} - [a_{ij}]$ ;  $\beta_{ij} = b_{ij} - [b_{ij}]$ ;  $\alpha_{ij} \geq 0$ ;  $\beta_{ij} \geq 0$ .

Тут  $\alpha_{ij}$  – дробова частина числа  $a_{ij}$ ;  $[a_{ij}]$  – ціла частина числа  $a_{ij}$ ;

$\beta_{ij}$  – дробова частина числа  $b_{ij}$ ;  $[b_{ij}]$  – ціла частина числа  $b_{ij}$ .

### ***Перший алгоритм Гоморі.***

***Крок 1.*** Розв'язати ЗЛП без урахування вимог на цілочисловість змінних. Якщо розв'язання задачі цілочислове, то обчислення завершено. В противному випадку перейти до кроку 2.

***Крок 2.*** Вибрати в останній симплекс-таблиці нецілу змінну  $x_i$  з максимальною дробовою частиною.

***Крок 3.*** Записати використовуючи останню симплекс-таблицю рівняння відтинання в дробових частинах:

$$- \sum_{j=1}^n \alpha_{ij} x_j + x_i = -\beta_i,$$

**Крок 4.** Розширити останню симплекс-таблицю на один рядок і один стовпець, записати в цю таблицю додаткові обмеження.

**Крок 5.** Розв'язати псевдоплан двоїтим симплекс-методом. Перейти до кроку 1.

**Приклад.** Розв'язати цілочислову задачу лінійного програмування:

$$\begin{aligned} 3x_1 + 2x_2 &\rightarrow \max \\ x_1 + x_2 &= 11 \\ x_1 - x_2 &\leq 6 \\ x_1 &\geq 3 \\ x_j &\geq 0, \quad j = 1, 2 \\ x_j &\text{ – цілі числа, } j = 1, 2. \end{aligned}$$

Зведемо задачу до канонічного вигляду:

$$\begin{aligned} 3x_1 + 2x_2 - Mx_3 - Mx_5 &\rightarrow \max \\ x_1 + x_2 + x_3 &= 11 \\ x_1 - x_2 + x_4 &= 6 \\ x_1 - x_6 + x_5 &= 3 \\ x_j &\geq 0, \quad j = 1, \dots, 6 \\ x_j &\text{ – цілі числа, } j = 1, \dots, 6. \end{aligned}$$

Результати розв'язання представлені в табл. 2.14–2.19.

Таблиця 2.14

Базис	$C_{\text{баз}}$	БР	3	2	-M	0	-M	-M
			$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$
$A_3$	-M	11	1	1	1	0	0	0
$A_4$	0	6	1	-1	0	1	0	0
$A_5$	-M	3	1	0	0	0	1	-1
$\Delta_j = Z_j - C_j$		0 - 14M	-3 -2M	-2 - M	0 -M	0 0M	0 -M	0 M

Таблиця 2.15

Базис	$C_{\text{баз}}$	БР	3	2	-M	0	-M	-M
			$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$
$A_3$	-M	8	0	1	1	0	-1	1
$A_4$	0	3	0	-1	0	1	-1	1
$A_1$	3	3	1	0	0	0	1	-1
$\Delta_j = Z_j - C_j$		9 - 8M	3 0M	-2 -M	0 -M	0 0M	3 M	-3 -M

Таблиця 2.16

Базис	$C_{\text{баз}}$	БР	3	2	-M	0	-M	-M
			$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$
$A_2$	2	8	0	1	1	0	-1	1
$A_4$	0	11	0	0	1	1	-2	2
$A_1$	3	3	1	0	0	0	1	-1
$\Delta_j = Z_j - C_j$		25 0M	0 0M	0 0M	2 0M	0 0M	1 0M	-1 0M

Таблиця 2.17

Базис	C <sub>баз</sub>	БР	3	2	-M	0	-M	-M
			A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>4</sub>	A <sub>5</sub>	A <sub>6</sub>
A <sub>2</sub>	2	5/2	0	1	1/2	-1/2	0	0
A <sub>6</sub>	0	11/2	0	0	1/2	1/2	-1	1
A <sub>1</sub>	3	17/2	1	0	1/2	1/2	0	0
$\Delta_j = Z_j - C_j$		61/2	0	0	5/2	1/2	0	0

Розв'язки нецілочислової задачі:  $X = (17/2, 5/2, 0, 0, 0, 11/2)$ ,  $z = 61/2$ .

Відтинання для змінної  $x_1$  на основі останньої симплекстаблиці (для базису A<sub>1</sub>):

$$-1/2x_3 - 1/2x_4 + x_7 = -1/2$$

Додамо це обмеження до системи обмежень вихідної задачі і вирішуватимемо задачу із застосуванням двоїстого симплекс-методу.

Таблиця 2.18

Базис	C <sub>баз</sub>	БР	3	2	0	0	0	0	0
			A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>4</sub>	A <sub>5</sub>	A <sub>6</sub>	A <sub>7</sub>
A <sub>2</sub>	2	5/2	0	1	1/2	-1/2	0	0	0
A <sub>6</sub>	0	11/2	0	0	1/2	1/2	-1	1	0
A <sub>1</sub>	3	17/2	1	0	1/2	1/2	0	0	0
A <sub>7</sub>	0	-1/2	0	0	-1/2	-1/2	0	0	1
$\Delta_j = Z_j - C_j$		61/2	0	0	5/2	1/2	0	0	0

Таблиця 2.19

Базис	C <sub>баз</sub>	БР	3	2	0	0	0	0	0
			A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>4</sub>	A <sub>5</sub>	A <sub>6</sub>	A <sub>7</sub>
A <sub>2</sub>	2	3	0	1	1	0	0	0	-1
A <sub>6</sub>	0	5	0	0	0	0	-1	1	1
A <sub>1</sub>	3	8	1	0	0	0	0	0	1

$A_4$	0	1	0	0	1	1	0	0	-2
$\Delta_j = Z_j - C_j$	30	0	0	2	0	0	0	0	1

Розв'язки цілочислової задачі:  $X = (8, 3, 0, 1, 0, 5, 0)$ ,  $z = 30$ .

## ВИСНОВКИ ПО РОЗДІЛУ 2

1. Наведена канонічна форма задач лінійного програмування.
2. Представлене графічне розв'язання задачі лінійного програмування.
3. Розглянуто двоїстість у лінійному програмуванні.
4. Наведений двоїстий симплекс-метод.
5. Представлене лінійне цілочислове програмування.



## РОЗДІЛ 3

### ОПТИМІЗАЦІЯ НА ГРАФАХ

#### 3.1 Загальні положення

Мережі використовуються для розв'язування задач оптимізації в різних предметних областях. Ці задачі мають певну специфіку формалізації і розв'язання:

- значна частина задач цього типу може вирішуватися методами лінійного програмування, однак специфічні особливості цих задач дозволили розробити більш ефективні методи;

- ці задачі мають геометричне зображення у вигляді графа.

В цьому розділі розглядаються такі задачі:

- задача про найкоротший ланцюг;
- задача про багатополосну мережу;
- задача про оптимальне дерево-кістяк;
- транспортна задача;
- задача про призначення;
- задача про максимальний потік.

#### 3.2 Задача про найкоротший ланцюг. Алгоритм Дейкстри

**Постановка задачі.** Необхідно визначити в мережі найкоротший ланцюг між заданими вхідним пунктом і пунктом призначення.

**Алгоритм Дейкстри.**

Алгоритм будується на положенні, що якщо найкоротший ланцюг із вузла  $i$  у вузол  $j$  проходить через вузол  $k$ , то найкоротший ланцюг із вузла  $i$  у вузол  $k$  є частиною початкового найкоротшого ланцюга, який закінчується у вузлі  $k$ .

У процесі виконання цього алгоритму використовується процедура приписування міток вузлам. Мітки вузлів можуть бути тимчасові і постійні. Тимчасова мітка може бути замінена на іншу тимчасову, якщо буде

знайдено більш короткий шлях до даного вузла. Якщо такого шляху не існує, то тимчасова мітка змінюється на постійну.

Позначимо через  $u_i$  найкоротший шлях вихідного вузла 1 до вузла  $i$ , через  $d_{ij}$  - довжину дуги  $(i, j)$ . Для вузла  $j$  визначимо мітку  $[u_j, i]$  так:

$$[u_j, i] = [u_i + d_{ij}, i], d_{ij} \geq 0.$$

Ітеративна процедура алгоритму Дейкстри складається з таких кроків.

**Крок 0.** Вихідному вузлу (вузол 1) приписуємо мітку  $[0, -]$ . Вважаємо  $i = 0$ .

**Крок  $i$ .**

1. Обчислюємо тимчасові мітки  $[u_i + d_{ij}, i]$  для всіх вузлів  $j$ , які можна досягти безпосередньо із вузла  $i$  і які не мають постійних міток. Якщо вузол  $j$  вже має мітку  $[u_j, k]$ , яка отримана від іншого вузла  $k$ , і якщо  $u_i + d_{ij} < u_j$ , тоді мітка  $[u_j, k]$  замінюється на  $[u_i + d_{ij}, i]$ .

2. Якщо всі вузли мають постійні мітки, процес обчислень закінчується. У протилежному випадку обирається мітка  $[u_r, s]$  з найменшим значенням відстані  $u_r$  серед усіх тимчасових міток. Вважаємо  $i = r$  і повторюємо крок  $i$ .

**Приклад.** Знайти найкоротший ланцюг у графі (рис.3.1).

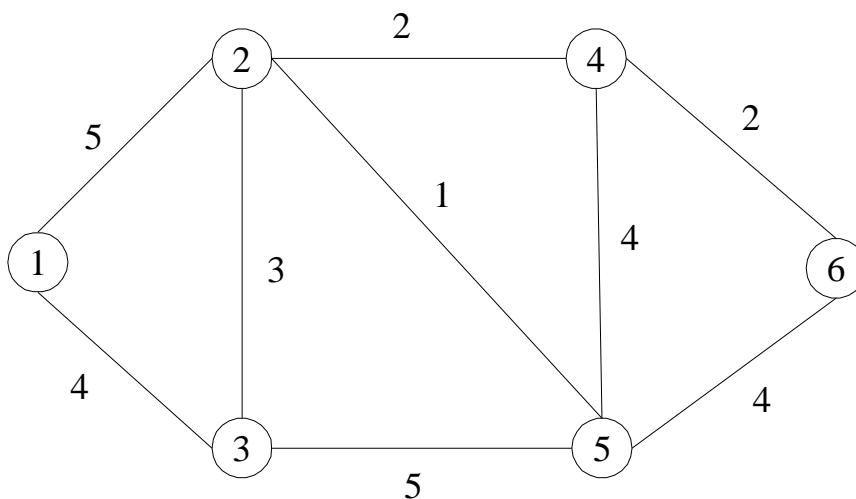


Рисунок 3.1 Приклад мережі для визначення найкоротшого шляху

Послідовність рішення задачі наведено в табл. 2.1.

Таблиця 2.1

№	1	2	3	4	5	6
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	0	5	4	$\infty$	$\infty$	$\infty$
2	0	5	4	$\infty$	9	$\infty$
3	0	5	4	7	6	$\infty$
4	0	5	4	7	6	10
5	0	5	4	7	6	9

Згідно з алгоритмом розв'язання задачі знайдено: найкоротший маршрут з вершини 1 у вершину 6 дорівнює 9.

### 3.3 Задача про багатополюсну мережу. Алгоритм Флойда - Ху

**Постановка задачі.** Необхідно визначити найкоротші ланцюги між усіма вузлами багатополюсної мережі.

**Алгоритм Флойда - Ху.**

Алгоритм будується на таких положеннях. Необхідно визначити найкоротший ланцюг із вузла  $i$  у вузол  $k$  (рис. 3.2).

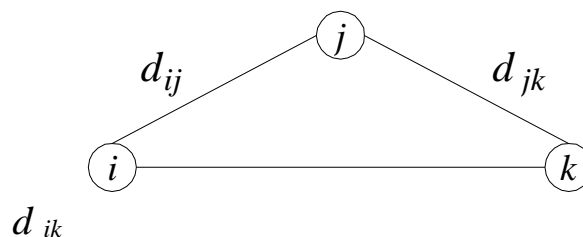


Рисунок 3.2 Фрагмент мережі для визначення відстані між вузлами  $i$  і  $k$

Спочатку ця довжина береться рівною довжині дуги  $(i, k)$  і дорівнює  $d_{ik}$ , або  $\infty$ , якщо з'єднання немає. Потім послідовно перевіряються всі можливі проміжні вузли, розташовані між  $i$  і  $k$ . Якщо довжина ланцюга, що проходить через деякий проміжний вузол  $j$ , менша за поточне значення, то

змінній  $d_{ik}$  надається нове значення  $d_{ik} = d_{ij} + d_{jk}$ . Дана процедура повторюється для всіх можливих пар вузлів. Задача для  $n$  вузлів розв'язується за допомогою  $n$  ітерацій.

На кожній ітерації  $j$  відстань обчислюється так:

$$d_{ik}^j = \min [d_{ik}^{j-1}, d_{ij}^{j-1} + d_{jk}^{j-1}], \quad i \neq j \neq k.$$

На кожній ітерації будуються дві матриці. Перша з них називається **матрицею довжин** найкоротших шляхів  $D^j = [d_{ik}^j]$  і означає найкоротші відстані між усіма парами вузлів. Друга матриця називається **матрицею маршрутів**  $R^j = [r_{ik}^j]$  і означає проміжні вузли найкоротших ланцюгів між усіма парами вузлів. Тут  $r_{ik}^j$  – проміжний вузол найкоротшого ланцюга з  $i$  в  $k$ , обраний на  $j$ -й ітерації:  $r_{ik}^j = j$ , якщо  $d_{ik}^{j-1} > d_{ij}^{j-1} + d_{jk}^{j-1}$ ;  $r_{ik}^j = r_{ik}^{j-1}$ , в протилежному випадку.

**Крок 0.**  $j=0$ . Будуємо початкову матрицю вартостей  $D^0$  і початкову матрицю маршрутів  $R^0$ . Елементи  $d_{ij}^0$  початкової матриці вартостей дорівнюють вартості дуги  $(i,k)$ , якщо вузли  $i$  та  $k$  безпосередньо зв'язані дугою  $(i,k)$ , і  $\infty$  в супротивному разі, всі діагональні елементи матриці  $D^0$  дорівнюють нулю. Елементи  $r_{ik}^0$  матриці маршрутів  $R^0$  дорівнюють  $k$ .

**Крок 1.** Починаючи з  $j = 1$ , виділяємо в матриці  $D^{j-1}$   $j$ -й стовпець та  $j$ -й рядок, оголошуємо їх **базовими** і виконуємо таку операцію:

$$d_{ik}^j = \min [d_{ik}^{j-1}, d_{ij}^{j-1} + d_{jk}^{j-1}].$$

Елементи  $r_{ik}^j$  матриці  $R^j$  знаходять через елементи  $r_{ik}^{j-1}$  матриці  $R^{j-1}$  за такою схемою:  $r_{ik}^j = j$ , якщо  $d_{ik}^{j-1} > d_{ij}^{j-1} + d_{jk}^{j-1}$ ;  $r_{ik}^j = r_{ik}^{j-1}$ , в протилежному випадку.

**Приклад.** Для прикладу графа, який наведено при розгляді алгоритму Дейкстри, визначити найкоротші відстані між усіма парами вузлів і відповідні їм маршрути (рис. 3.3).

Крок 0

$D^0$

$R^0$

	1	2	3	4	5	6
1	—	5	4	$\infty$	$\infty$	$\infty$
2	5	—	3	2	1	$\infty$
3	4	3	—	$\infty$	5	$\infty$
4	$\infty$	2	$\infty$	—	4	2
5	$\infty$	1	5	4	—	4
6	$\infty$	$\infty$	$\infty$	2	4	—

	1	2	3	4	5	6
1	—	2	3	4	5	6
2	1	—	3	4	5	6
3	1	2	—	4	5	6
4	1	2	3	—	5	6
5	1	2	3	4	—	6
6	1	2	3	4	5	—

Крок 1

$D^1$

$R^1$

	1	2	3	4	5	6
1	—	5	4	$\infty$	$\infty$	$\infty$
2	5	—	3	2	1	$\infty$
3	4	3	—	$\infty$	5	$\infty$
4	$\infty$	2	$\infty$	—	4	2
5	$\infty$	1	5	4	—	4
6	$\infty$	$\infty$	$\infty$	2	4	—

	1	2	3	4	5	6
1	—	2	3	4	5	6
2	1	—	3	4	5	6
3	1	2	—	4	5	6
4	1	2	3	—	5	6
5	1	2	3	4	—	6
6	1	2	3	4	5	—

Крок 2

$D^2$

$R^2$

	1	2	3	4	5	6
1	—	5	4	7	6	$\infty$
2	5	—	3	2	1	$\infty$
3	4	3	—	5	4	$\infty$
4	7	2	5	—	3	2
5	6	1	4	3	—	4
6	$\infty$	$\infty$	$\infty$	2	4	—

	1	2	3	4	5	6
1	—	2	3	2	2	6
2	1	—	3	4	5	6
3	1	2	—	2	2	6
4	2	2	2	—	2	6
5	2	2	2	2	—	6
6	1	2	3	4	5	—

Рисунок 3.3 Послідовність розв'язання задачі про багатополісну мережу

*Крок 3*  $D^3$

	1	2	3	4	5	6
1	—	5	4	7	6	$\infty$
2	5	—	3	2	1	$\infty$
3	4	3	—	5	4	$\infty$
4	7	2	5	—	3	2
5	6	1	4	3	—	4
6	$\infty$	$\infty$	$\infty$	2	4	—

$R^3$

	1	2	3	4	5	6
1	—	2	3	2	2	6
2	1	—	3	4	5	6
3	1	2	—	2	2	6
4	2	2	2	—	2	6
5	2	2	2	2	—	6
6	1	2	3	4	5	—

*Крок 4*  $D^4$

	1	2	3	4	5	6
1	—	5	4	7	6	9
2	5	—	3	2	1	4
3	4	3	—	5	4	7
4	7	2	5	—	3	2
5	6	1	4	3	—	4
6	9	4	7	2	4	—

$R^4$

	1	2	3	4	5	6
1	—	2	3	2	2	4
2	1	—	3	4	5	4
3	1	2	—	2	2	4
4	2	2	2	—	2	6
5	2	2	2	2	—	6
6	4	4	4	4	5	—

*Крок 5*  $D^5$

	1	2	3	4	5	6
1	—	5	4	7	6	9
2	5	—	3	2	1	4
3	4	3	—	5	4	7
4	7	2	5	—	3	2
5	6	1	4	3	—	4
6	9	4	7	2	4	—

$R^5$

	1	2	3	4	5	6
1	—	2	3	2	2	4
2	1	—	3	4	5	4
3	1	2	—	2	2	4
4	2	2	2	—	2	6
5	2	2	2	2	—	6
6	4	4	4	4	5	—

Рисунок 3.3 Послідовність розв'язання задачі про багатополісну мережу

	<i>D</i>					
	1	2	3	4	5	6
1	—	5	4	7	6	9
2	5	—	3	2	1	4
3	4	3	—	5	4	7
4	7	2	5	—	3	2
5	6	1	4	3	—	4
6	9	4	7	2	4	—

	<i>R</i>					
	1	2	3	4	5	6
1	—	2	3	2	2	4
2	1	—	3	4	5	4
3	1	2	—	2	2	4
4	2	2	2	—	2	6
5	2	2	2	2	—	6
6	4	4	4	4	5	—

Рисунок 3.3 Послідовність розв’язання задачі про багатополюсну мережу

Матриці  $D$  і  $R$  є розв’язками задачі.

### 3.4 Задача про оптимальне дерево-кістяк

**Постановка задачі.** Необхідно визначити дерево-кістяк, сумарна вартість ребер якого – оптимальна (максимальна або мінімальна).

**Алгоритм побудови оптимального дерева-кістяка.**

**Крок 1.** На множині вершин  $N$  мережі  $(N, S)$ , де  $S$  - множина ребер, вводимо дві множини:  $A$  - множина з'єднаних вузлів і  $\bar{A}$  - множина вузлів які не з'єднані. Припускається, що  $A \cup \bar{A} = N$  і  $A \cap \bar{A} = \emptyset$ . Спочатку всі вузли належать множині  $\bar{A}$ :  $\bar{A} = N$ ,  $A = \emptyset$ .

**Крок 2.** Вибираємо будь-який вузол із множини  $A$  і з'єднуємо його з іншим вузлом ребром мінімальної (максимальної) вартості і два одержані вузли вилучаємо з множини  $\bar{A}$  і включаємо їх у множину  $A$ . В кінці цього кроку множина  $A$  складається з двох вузлів.

**Крок 3.** Серед усіх ребер, які з'єднують вузли з множини  $\bar{A}$  з вузлами із множини  $A$ , вибираємо ребро з оптимальною вартістю і кінцевий вузол цього ребра, який належить множині  $\bar{A}$ , вилучаємо з множини  $\bar{A}$  і приєднуємо до множини  $A$ .

**Крок 4.** Якщо  $\bar{A} \neq 0$ , то перейти на виконання кроку 3. При  $\bar{A}=0$  алгоритм закінчує роботу.

**Приклад.** Для графа, який наведено при розгляді алгоритму Дейкстри, побудувати мінімальне дерево-кістяк (рис.3.4).

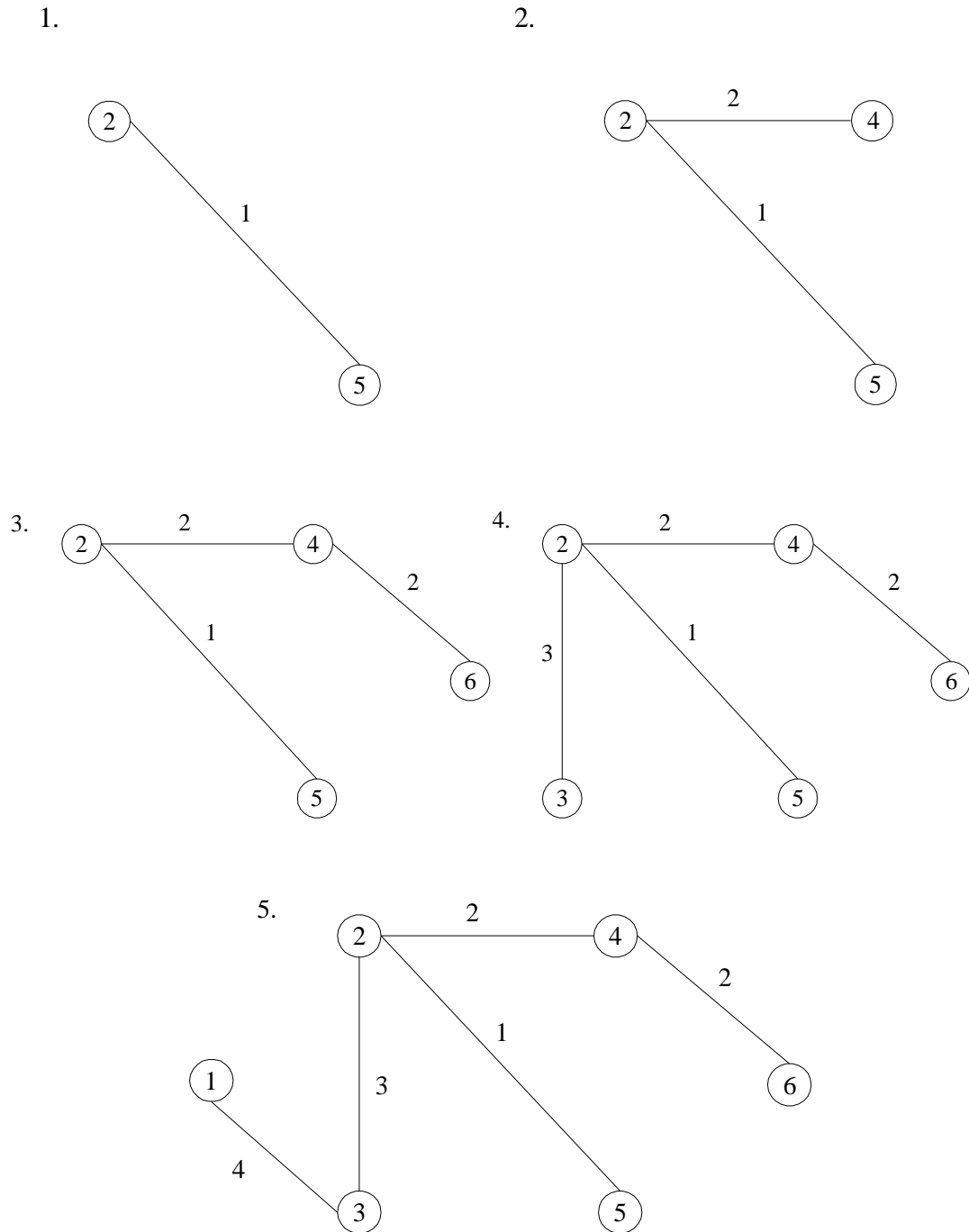


Рисунок 3.4 Розв'язання задачі про мінімальне дерево-кістяк



### 3.5 Задача про призначення. Угорський алгоритм

**Постановка задачі.** Необхідно  $n$ -ресурсів оптимальним чином призначити на  $n$ -об'єктів так, що кожен ресурс призначається тільки на один об'єкт і кожному об'єкту приписується тільки один ресурс.

Математична модель задачі про призначення має вигляд:

$$\sum_{j=1}^n \sum_{i=1}^n c_{ij} x_{ij} \rightarrow \max (\min);$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i = \overline{1, n};$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j = \overline{1, n};$$

$$x_j \geq 0, \quad j = \overline{1, n}.$$

де  $c_{ij}$  – ефективність ресурсу  $i$  на об'єкті  $j$ ;

$x_{ij} = \{1, \text{ якщо } i - \text{ресурс призначається на } j - \text{й об'єкт};$

0 у протилежному випадку.

**Угорський алгоритм розв'язання задачі про призначення.**

Метод заснований на редукції вихідної матриці і призначенні за нульовими елементами редукованої матриці. При неможливості призначення за нульовими елементами матриця перетворюється так, щоб допустимих нулів було більше, ніж в редукованій матриці.

Існує матриця  $C = \| c_{ij} \|$ .

Виконаємо перетворення цієї матриці:

$$a_{ij} = c_{ij} - u_{ij} - v_{ij},$$

де  $u_{ij}$  – мінімальний елемент для рядка  $i$ , а  $v_{ij}$  – мінімальний елемент для стовпця  $j$ .

$$a_{ij} x_{ij} = c_{ij} x_{ij} - u_{ij} x_{ij} - v_{ij} x_{ij},$$

$$\sum_{i=1}^n \sum_{j=1}^n a_{ij} x_{ij} = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} = \sum_{i=1}^n \sum_{j=1}^n u_{ij} x_{ij} = \sum_{i=1}^n \sum_{j=1}^n v_{ij} x_{ij}.$$

**Крок 1.** Перейти від початкової матриці до редукованої. Для цього від усіх елементів кожного рядка відняти мінімальний елемент цього рядка, а потім у редукованій по рядках матриці виконати таку саму операцію зі стовпцями. Після цього початкову матрицю замінити редукованою матрицею.

**Крок 2.** Виконати призначення. Якщо після редукції в кожному рядку і в кожному стовпці можна вибрати по одному нульовому елементу так, щоб виконувалась система обмежень, то таке призначення буде оптимальним. Якщо цього не можна зробити, то треба перейти на наступний крок.

**Крок 3.** Модифікація редукованої матриці. Закреслити в редукованій матриці всі стовпці і рядки, які містять хоча б один нуль, починаючи з тих рядків і стовпців, які містять максимальне число нулів, і продовжувати закреслення в такому порядку й далі. Із всіх невикреслених елементів знайти найменший, відняти його від кожного невикресленого елемента і додати його до кожного елемента, розташованого на перетині двох ліній, за допомогою яких були викреслені рядки та стовпці. Перейти на крок 2 і якщо не вдасться виконати повне призначення, асоційоване з нулями модифікованої матриці, то знову перейти на крок 3. І так доти, доки не одержимо оптимальне призначення, асоційоване з нулями модифікованої матриці.

Якщо необхідно дослідити задачу на максимум, то всі елементи матриці слід помножити на (-1) і потім скласти з найбільшим числом матриці, таким чином, щоб матриця не вміщувала від'ємних елементів. Потім задача розв'язується, як задача мінімізації.

**Приклад.** Розв'язати задачу про призначення при мінімізації ЦФ (рис. 3.5).

1.

$a_i \backslash b_j$	$S_1$	$S_2$	$S_3$	$S_4$
$W_1$	2	8	12	6
$W_2$	18	14	20	18
$W_3$	8	10	22	14
$W_4$	16	14	16	10

2.

$a_i \backslash b_j$	$S_1$	$S_2$	$S_3$	$S_4$	min
$W_1$	2	8	12	6	2
$W_2$	18	14	20	18	14
$W_3$	8	10	22	14	8
$W_4$	16	14	16	10	10

3.

$a_i \backslash b_j$	$S_1$	$S_2$	$S_3$	$S_4$
$W_1$	0	6	10	4
$W_2$	4	0	6	4
$W_3$	0	2	14	6
$W_4$	6	4	6	0
min	0	0	6	0

4.

$a_i \backslash b_j$	$S_1$	$S_2$	$S_3$	$S_4$
$W_1$	0	6	4	4
$W_2$	4	0	0	4
$W_3$	0	2	8	6
$W_4$	6	4	0	0

5.

$a_i \backslash b_j$	$S_1$	$S_2$	$S_3$	$S_4$
$W_1$	0	6	4	4
$W_2$	4	0	0	4
$W_3$	0	2	8	6
$W_4$	6	4	0	0

6.

$a_i \backslash b_j$	$S_1$	$S_2$	$S_3$	$S_4$
$W_1$	0	4	2	2
$W_2$	6	0	0	4
$W_3$	0	0	6	4
$W_4$	8	4	0	0

Рисунок 3.5 Послідовність розв'язання задачі про призначення

Згідно з алгоритмом розв'язування задачі знайдено оптимальні призначення:  $W_1-S_1$ ,  $W_2-S_3$ ,  $W_3-S_2$ ,  $W_4-S_4$ .

### 3.6 Задача про максимальний потік

**Постановка задачі.** Необхідно визначити максимальний потік з джерела в стік для мережі заданої конфігурації.

**Алгоритм знаходження максимального потоку.**

Алгоритм ґрунтується на тому, що починаючи з джерела  $S$ , обирається будь-який ланцюг у стік  $T$  і помічаються вузли цього ланцюга. Якщо стік можна помітити, то тоді на цьому ланцюгу можна збільшити потік. Після виконання цих операцій всі мітки стираються і робляться нові мітки. Значення, які отримуються в результаті послідовних кроків, додаються. Якщо стік не можна помітити, то отриманий потік і є оптимальним.

Кожна вершина  $j$  отримує мітку або  $(i^+, \varepsilon_j)$ , або  $(i^-, \varepsilon_j)$ , де  $i$  означає номер попередньої вершини ненасиченого ланцюга від  $S$  до  $T$ , а  $\varepsilon_j$  – ціле число, або  $\infty$ , яке означає величину можливого приросту потоку вздовж ланцюга від джерела  $S$  до даної вершини  $j$ .

**Крок 1.** Позначимо вершину  $S$  міткою  $(0, \infty)$ . Виберемо будьяку непомічену вершину  $i$  (спочатку це джерело  $S$ ). Розглянемо всі непомічені вершини  $j$ , які пов'язані дугами з вершиною  $i$ . Ті вершини  $j$ , для яких  $x_{ij} < c_{ij}$ , отримують мітку  $(i^+, \varepsilon_j)$ , де  $\varepsilon_j = \min(\varepsilon_i, c_{ij} - x_{ij})$ . Ті вершини  $j$ , для яких  $x_{ji} > 0$ , отримують мітку  $(i^-, \varepsilon_j)$ , де  $\varepsilon_j = \min(\varepsilon_i, c_{ij} - x_{ji})$ . Потім розглянемо наступну вершину і поступимо так само. Ці операції повторюються до тих пір, поки вершина  $T$  не отримує мітку, або поки не можна буде зробити нові мітки.

**Крок 2.** Вершина  $T$  отримує мітку або  $(j^+, \varepsilon_t)$  або  $(j^-, \varepsilon_t)$ . Якщо мітка буде  $(j^+, \varepsilon_t)$ , то змінимо  $x_{jt}$  на  $(x_{jt} + \varepsilon_t)$ , якщо ж мітка буде  $(j^-, \varepsilon_t)$ , то змінимо  $x_{ij}$  на  $(x_{ij} - \varepsilon_t)$ . Потім переходимо до вершини  $j$  і поступимо так само: якщо мітка буде  $(i^+, \varepsilon_j)$ , то змінимо  $x_{ij}$  на  $(x_{ij} + \varepsilon_t)$ , якщо ж мітка буде  $(i^-, \varepsilon_j)$ , то змінимо  $x_{ji}$  на  $(x_{ji} - \varepsilon_t)$ . Ці операції повторюються до тих пір поки не буде досягнена вершина  $S$ . Після цього всі мітки стираються і робиться перехід до кроку 1.

Процес розставлення міток являє собою пошук ланцюга з  $S$  в  $T$  і визначення на цьому ланцюгу значення максимального потоку, на який сумарний потік можна збільшити. Процедура завершується, коли вже неможливо продовжувати процес розставлення міток.

**Приклад.** Визначити максимальний потік для даної мережі (рис. 3.6).

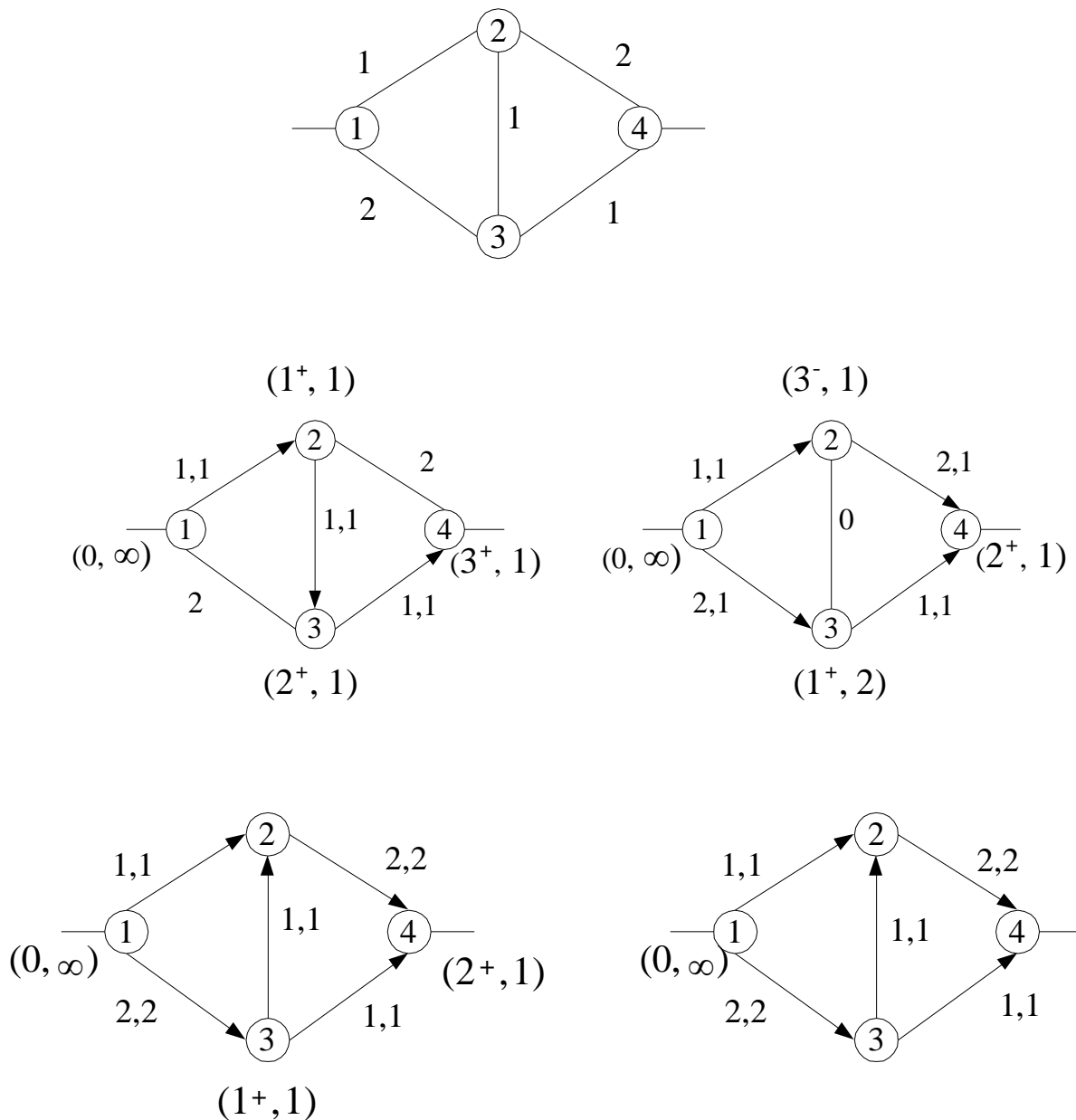


Рисунок 3.6 Послідовність розв'язання задачі про максимальний потік

Згідно з алгоритмом розв'язання задачі знайдено:  
максимальний потік дорівнює 3.

## **ВИСНОВКИ ПО РОЗДІЛУ 3**

1. Наведена задача про найкоротший ланцюг. Алгоритм Дейкстри.
2. Наведена задача про багатополюсну мережу. Алгоритм Флойда.
3. Наведена задача про оптимальне дерево-кістяк.
4. Наведена задача про призначення. Угорський алгоритм.
5. Наведена задача про максимальний потік.

## **РОЗДІЛ 4**

### **МОДЕЛІ І МЕТОДИ ЕКСПЕРТНИХ ОЦІНОК В СИСТЕМІ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ**

#### **4.1 Моделі та методи формування експертної оцінки системи підтримки прийняття рішень в задачах діагностики технічного стану будівель**

Існують декілька підходів до вирішення задач, які базуються на експертній обробці даних та побудові моделей та методів діагностики технічного стану будівель.

Одним з напрямків, що базуються на експертній обробці даних, в реалізації комплексу задач по безпечній експлуатації будівель є моделі та методи формування експертної оцінки при обстеженні технічного стану будівель. Перевага цього підходу полягає в задачі створення системи діагностики технічного стану будівель; дослідженні інтелектуальної технології при реалізації інформаційної системи управління обстеження та діагностики технічного стану будівель; проведення досліджень експертних систем.

В зв'язку з цим розглянемо підхід експертної оцінки обстеження технічного стану об'єктів будівництва. При такому підході з'являється можливість отримання результатів ознак пошкодження при різних варіантах і при різних ознаках та співставлення результатів заданими спочатку. Це дозволяє вести процес спостереження та своєчасно приймати необхідні рішення щодо безпечної та надійної експлуатації будівель і створення нормальних умов перебування обслуговуючого персоналу, враховуючи результати обстеження будівель.

При побудові експертної оцінки формується експертна група, яка після проведення попереднього огляду будівлі та визначення необхідних обсягів робіт, на другому етапі обстеження, визначають дефекти та пошкодження основних конструктивних елементів будівлі.

*Метод формування експертної оцінки при обстеженні технічного стану будівель (етапи):*

1. Формування ознак пошкоджень (ранжування) – ( $m$ ).
2. Формування експертної групи:
  - призначається кількість експертів ( $h$ );
  - в залежності від відповідей експертів формується матриця-рядок по кожній  $j$ -тій ознаці пошкодження:

$$Y_j = |a_{1j}, a_{2j}, \dots, a_{hj}| \quad (4.1);$$

- знаходиться середнє значення оцінок групи по  $j$ -тій ознаці пошкодження:

$$A_j = \left( \sum_{t=1}^h a_{tj} \right) / h \quad (4.2);$$

- визначається відхилення оцінки кожного експерта від середнього значення оцінок групи по всім  $j$ -тим ознакам пошкодження  $\Delta_{ij} = |a_{ij} - A_j|$ , в результаті формується матриця відхилень:

$$D = ||D_j|| = \left\| \begin{array}{cccc} \Delta_{11} & \Delta_{21} & \dots & \Delta_{h1} \\ \Delta_{12} & \Delta_{22} & \dots & \Delta_{h2} \\ \dots & \dots & \dots & \dots \\ \Delta_{1m} & \Delta_{2m} & \dots & \Delta_{hm} \end{array} \right\| \quad (4.3);$$

- знаходиться середнє відхилення оцінок кожного експерта за всіма ознакам пошкодження від середнього значення оцінок групи:



$$\bar{\Delta} = \left( \sum_{j=1}^m \bar{\Delta} \right) / m \quad (4.4);$$

в результаті отримана матриця-рядок:

$$\bar{D} = \left| \bar{\Delta}_1, \bar{\Delta}_2, \dots, \bar{\Delta}_h \right| \quad (4.5);$$

- експерти нумеруються за ступенем віддаленості їх оцінок від середнього значення оцінок групи. В підсумку встановлюється кортеж компетентності експертів:

$$\bar{D}^* = \left| \bar{\Delta}_1^*, \bar{\Delta}_2^*, \dots, \bar{\Delta}_h^* \right| \quad (4.6);$$

- визначається середнє значення коефіцієнта впевненості. Коефіцієнт впевненості визначається в залежності від ознак фізичного зношення та правил оцінки технічного стану основних конструктивних елементів будівлі, як правило, приймається рівним 0,5 ( $F=0,5$ ). При  $F < 0,5$  експертну групу переформовують шляхом виключення із списку останніх номерів експертів, в яких спостерігається різке відхилення відповідей від середньої думки групи.

### 3. Формування правил роботи експертної групи (табл. 4.1):

- впорядковуються ознаки пошкодження, починаючи з найменшого важливого  $x_1 < x_2 < \dots < x_m$  (4.7);

- приписуються ранги  $a_i (a_1=1; a_2=1; \dots a_m=1)$  (4.8);

- визначаються вагові коефіцієнти ознак пошкодження  $a_j (j = \overline{1, m})$

за формулою:

$$a_j = \sum_{t=1}^h a_{jt} / \left( \sum_{j=1}^m \sum_{t=1}^h a_{jt} \right) \quad (4.9).$$

## Формування правил роботи експертної групи

Номер ознаку пошкодження	Умовні позначення ознаку пошкодження	Номер експерта					Вагові коефіцієнти
		1	2	...	...	j	
1	$m_1$	$h_{11}$	$h_{12}$	...	...	$h_{1j}$	$a_1$
2	$m_2$	$h_{21}$	$h_{22}$	...	...	$h_{2j}$	$a_2$
...	...	...	...	...	...	...	...
n	$m_i$	$h_{i1}$	$h_{i2}$	...	...	$h_{ij}$	$a_j$

4. Оцінка ступеню узгодженості думок експертів (табл. 4.2):

- результати ранжування представляються у вигляді матриці рангів;
- визначається відповідно сума рангів по кожній  $j$ -тій ознаці пошкодження і середня сума рангів:

$$Q_j = \sum_{t=1}^h a_{jt}, \quad (4.10),$$

де  $Q_j$  – сума рангів;

$$T = \sum_{j=1}^m \sum_{t=1}^h a_{jt} / m, \quad (4.11),$$

де  $T$  – середня сума рангів.

- розраховується сума квадратів відхилень:

$$S_E = \sum_{j=1}^m \delta_j^2 = \sum_{j=1}^m (Q_j - T)^2 \quad (4.12);$$

- визначається коефіцієнт впевненості:

$$F = 12S_E / \left\{ h^2 (m^3 - m) \right\} \quad (4.13).$$

Таблиця 4.2

## Оцінка ступеню узгодженості думок експертів

Номер ознаку пошкодження	Умовні позначення ознаку пошкодження	Номер експерта					Сума рангів	Відхилення суми рангів	Квадрат відхилення
		1	2	...	...	j			
1	$m_1$	$h_{11}$	$h_{12}$	...	...	$h_{1j}$	$Q_1$	$Q_1 - T$	$(Q_1 - T)^2$
2	$m_2$	$h_{21}$	$h_{22}$	...	...	$h_{2j}$	$Q_2$	$Q_2 - T$	$(Q_2 - T)^2$
...	...	...	...	...	...	...	...	...	...
n	$m_i$	$h_{i1}$	$h_{i2}$	...	...	$h_{ij}$	$Q_j$	$Q_j - T$	$(Q_j - T)^2$
Середня сума рангів							T		
Сума квадратів відхилень								$S_E$	

Якщо  $F > 0,5$ , то існує достатня ступінь узгодженості між думками експертів. Якщо  $F < 0,5$ , то група експертів корегується шляхом виключення останнього експерта в кортежі, перераховується коефіцієнт впевненості і таке інше, до отримання необхідного ступеню узгодженості.

Якщо експерт не може вказати порядок спадання двох або декількох ознак пошкодження, він приписує кожному із них однаковий ранг.

В цьому випадку коефіцієнт впевненості розраховують до залежності:

$$F = S_E / \left( \frac{1}{12} h^2 (m^3 - m) - h \sum_{t=1}^h T_t \right) \quad (4.14),$$

$$T_t = \frac{1}{12} \sum_{j=1}^m (Z_j^3 - Z_j) \quad (4.15),$$

де  $Z_j$  – число однакових рангів в  $t$ -м ранжуванні.

## Приклад експертної оцінки системи підтримки прийняття рішень в задачах діагностики технічного стану будівлі

При формуванні експертної оцінки обстеженню технічного стану будівель можуть підлягати об'єкти громадського (житлові, адміністративні та громадські організації) та виробничого призначення (промислові підприємства, виробничі цехи, заводи, фабрики, гаражі).

Основними ознаками пошкодження при обстеженні будь-якої будівлі є: тріщина в цоколі фундаменту ( $m_1$ ), тріщина в стіні ( $m_2$ ), тріщина в плиті перекриття ( $m_3$ ), тріщина в залізобетонних кроквяних балках даху ( $m_4$ ), тому приймемо ці ознаки пошкодження в якості основних при оцінці.

На рис. 4.1. представлений зовнішній вигляд будівлі та виявленні ознаки пошкодження при обстеженні.

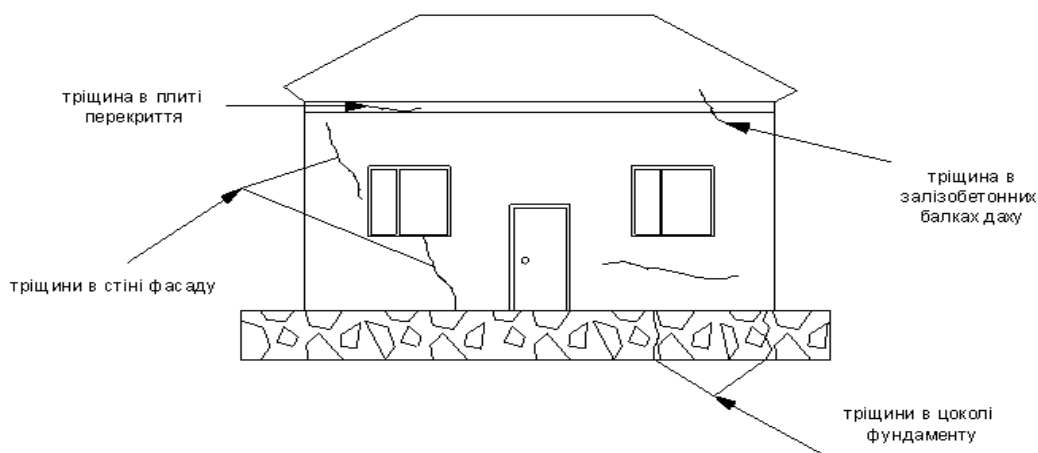


Рисунок 4.1 Зовнішній вигляд будівлі, виявленні ознаки пошкодження

При обстеженні технічного стану будівель в спеціалізованих організаціях, що займаються питаннями обстеження, існують підрозділи в яких, як правило формуються експертна група з 2 – 5 чоловік, тому на

основі проведеного опитування експертної групи (сформована із п'яти спеціалістів):

- I – експерт – завідуючий відділом;
- II – експерт – головний інженер;
- III – експерт – інженер I – ої категорії;
- IV – експерт – інженер;
- V – експерт – мол. наук. співробітник.

складені матриці рядку оцінок експертів по кожній  $j$ -тій ознаці пошкодження згідно виду (4.1):

$$Y_1 = / 0,41 \quad 0,37 \quad 0,05 \quad 0,44 \quad 0,33 / ;$$

$$Y_2 = / 0,21 \quad 0,33 \quad 0,11 \quad 0,23 \quad 0,30 / ;$$

$$Y_3 = / 0,10 \quad 0,08 \quad 0,33 \quad 0,13 \quad 0,05 / ;$$

$$Y_4 = / 0,16 \quad 0,17 \quad 0,21 \quad 0,06 \quad 0,21 / ;$$

Середнє значення оцінок групи по кожній ознаці пошкодження відповідно будуть становлять згідно виду (4.2):

$$A_1 = (0,41 + 0,37 + 0,05 + 0,44 + 0,33)/5 = 0,320;$$

$$A_2 = (0,21 + 0,33 + 0,11 + 0,23 + 0,30)/5 = 0,236;$$

$$A_3 = (0,10 + 0,08 + 0,33 + 0,13 + 0,05)/5 = 0,138;$$

$$A_4 = (0,16 + 0,17 + 0,21 + 0,06 + 0,21)/5 = 0,162.$$

Після визначення відхилень оцінок кожного експерта від середнього значення оцінок групи  $\Delta_{tj}$  по кожній ознаці пошкодження отримана матриця відхилень згідно виду (4.3):

$$\Delta_{11} = (0,41 - 0,32) = 0,09; \quad \Delta_{21} = (0,37 - 0,32) = 0,05; \quad \Delta_{31} = (0,05 - 0,32) = 0,27; \quad \Delta_{41} = (0,44 - 0,32) = 0,12; \quad \Delta_{51} = (0,33 - 0,32) = 0,01;$$

$$\Delta_{12} = 0,026; \quad \Delta_{22} = 0,094; \quad \Delta_{32} = 0,126; \quad \Delta_{42} = 0,006; \quad \Delta_{52} = 0,064;$$

$$\Delta_{13} = 0,038; \quad \Delta_{23} = 0,058; \quad \Delta_{33} = 0,192; \quad \Delta_{43} = 0,008; \quad \Delta_{53} = 0,088;$$

$$\Delta_{14} = 0,002; \quad \Delta_{24} = 0,008; \quad \Delta_{34} = 0,048; \quad \Delta_{44} = 0,102; \quad \Delta_{54} = 0,048.$$

$$D = \begin{vmatrix} 0,090 & 0,050 & 0,270 & 0,120 & 0,010 \\ 0,026 & 0,094 & 0,126 & 0,006 & 0,064 \\ 0,038 & 0,058 & 0,192 & 0,008 & 0,088 \\ 0,002 & 0,008 & 0,048 & 0,102 & 0,048 \end{vmatrix}$$

Обчислені середні відхилення оцінок кожного експерта по всім ознакам пошкодження від середнього значення оцінок групи мають наступні значення згідно виду (4.4):

$$\overline{\Delta_1} = (0,090 + 0,026 + 0,038 + 0,002) / 4 = 0,039;$$

$$\overline{\Delta_2} = (0,050 + 0,094 + 0,058 + 0,008) / 4 = 0,053;$$

$$\overline{\Delta_3} = (0,270 + 0,126 + 0,192 + 0,048) / 4 = 0,159;$$

$$\overline{\Delta_4} = (0,120 + 0,006 + 0,008 + 0,102) / 4 = 0,059;$$

$$\overline{\Delta_5} = (0,010 + 0,064 + 0,088 + 0,048) / 4 = 0,052.$$

В результаті встановлена матриця – рядок згідно виду (4.5):

$$\overline{D} = /0,039 \ 0,053 \ 0,159 \ 0,059 \ 0,052/.$$

Аналіз отриманих результатів дозволяє скласти кортеж компетентності експертів згідно виду (4.6):  $\overline{D}^* = \langle 1,5,2,4,3 \rangle$ .

Проведені експертами ранжування ознак пошкодження по важливості та значення вагових коефіцієнтів ознак пошкодження  $a_j$  представлені в табл. 4.3 згідно виду (4.8 – 4.10).

Таблиця 4.3

**Результати ранжування ознак пошкодження експертами та значення вагових коефіцієнтів**

Номер ознаку пошкодження	Умовні позначення ознаку пошкодження	Номер експерта					Вагові коефіцієнти
		1	5	2	4	3	
1	$m_1$	5	5	5	5	1	0,350
2	$m_2$	4	4	4	4	2	0,300
3	$m_3$	1	2	1	2	2	0,133
4	$m_4$	3	1	3	3	3	0,217

При оцінці ступеню узгодженості думок експертів були використані залежності четвертого етапу. Результати розрахунків зведені в табл. 4.4 згідно виду (4.11 – 4.14).

Таблиця 4.4

**Оцінка ступеню узгодженості експертів**

Номер ознаку пошкодження	Умовні позначення ознаку пошкодження	Номер експерта					Сума рангів	Відхилення суми рангів	Квадрат відхилення
		1	5	2	4	3			
1	$m_1$	5	5	5	5	1	21	9	81
2	$m_2$	4	4	4	4	2	18	6	36
3	$m_3$	1	2	1	2	2	8	- 4	16
4	$m_4$	3	1	3	3	3	13	1	1
Середня сума рангів						12			
Сума квадратів відхилень								134	

Оскільки ні один із експертів не поставив двом або більше ознакам пошкодження однакові ранги, то для знаходження коефіцієнта впевненості використовуємо вираз згідно виду (3.14):

$$F = 12 * 134 / 5^2 * (4^3 - 4) = 1608 / 1500 = 1,072.$$

Так як  $F > 0,5$ , то існує достатня узгодженість між експертами і отримані результати приймаються в якості кінцевих.

Основні ознаки пошкодження приймають кількісну оцінку:

- *тріщина в цоколі фундаменту = 0,350;*
- *тріщина в стіні = 0,300;*
- *тріщина в плиті перекриття = 0,133;*
- *тріщина в залізобетонних кроквяних балках даху = 0,217.*

Таким чином, при такому підході з'являється можливість отримання результатів ознак пошкодження при різних варіантах і при різних ознаках і співставлення результатів заданими спочатку. Це дозволяє вести процес спостереження та своєчасно приймати необхідні рішення щодо безпечної та надійної експлуатації будівель, враховуючи результати обстеження.

#### **4.2 Моделі структури властивостей експертної оцінки системи підтримки прийняття рішень в задачах діагностики технічного стану будівель**

Знос будівель прискорюється при прояві дефектів, що були допущені в ході дослідження та вибору ділянок для будівництва, при проектуванні і зведенні будівель, а також через порушення правил експлуатації.

Дефекти будівель в нормальних умовах є наслідком або недостатньої кваліфікації дослідників, проектувальників, працівників, що приймають будівлі в експлуатацію, або недбалості цих осіб. Дефекти можуть виникати також в процесі проектування та будівництві будівель



при здійсненні в них виробництва робіт по новій технології, зведенні в маловивчених в будівельному відношенні районах та в інших складних умовах.

Найбільш небезпечні дефекти в основах і фундаментах, в стінах, тобто в основних конструкціях, оскільки їх прояв веде до деформацій і руйнування всієї будівлі. Менш небезпечні дефекти в перегородках та інших ненесучих конструкціях.

При експлуатації будівель важливо оцінити характер і небезпеку пошкоджень. Причини, що викликають пошкодження наступні: дії зовнішніх чинників; дії внутрішніх чинників, обумовлених технологічним процесом; прояв дефектів, допущених при дослідженнях, проектуванні, зведенні будівлі; недоліки і порушення правил експлуатації будівель.

Тому побудова моделей структури властивостей в області проведення обстеження технічного стану будівель повинні привести до покращення всього спектру кількісних та якісних показників даної системи.

Існують декілька підходів до вирішення задач, які базуються на експертній обробці даних та побудові моделей структури властивостей при проведенні обстеження для вирішення задач діагностики.

Розглянемо функціонування даної системи, де  $h$  – кількість експертів. В залежності від ситуації система характеризується властивостями  $I_i = \{1, 2, \dots, n\}$  (категорія технічного стану) та  $V_i = \{v_{i1}, v_{i2}, \dots, v_{in}\}$  (ширина розкриття тріщин).

Маючи в якості вихідних даних вектори  $I_i$  (категорія технічного стану) і  $V_i$  (ширина розкриття тріщин) по всіх ситуаціям  $h_i$  (кількість

експертів), можна для кожної з них визначити вектор вагових коефіцієнтів  $\Delta_i = \{\lambda_{i1}, \lambda_{i2}, \dots, \lambda_{in}\}$ , який представляє собою  $n$  – мірний вектор, де компоненти його пов’язані відношеннями:

$$\left| \begin{array}{l} 0 \leq \lambda_{iq} \leq 1, \quad q = \overline{1, n}; \\ \sum_{q=1}^n \lambda_{iq} = 1. \end{array} \right. \quad (4.16)$$

Складові  $\lambda_{iq}$  вектора  $\Delta_i$  мають зміст вагових коефіцієнтів, що визначають відносну перевагу  $q$ -ї властивості над іншими. Знаходяться вони за допомогою залежності:

$$\lambda_{iq} = \frac{\prod_{l=1}^n v_{il}}{\sum_{q=1}^n \prod_{l=q}^n v_{il}}, \quad q = \overline{1, n}, \quad i = \overline{1, m}. \quad (4.17)$$

В результаті розрахунків значень вагових коефіцієнтів по кожній типовій ситуації знаходиться матриця виду:

$$A = \left\| \begin{array}{cccc} \lambda_{11} & \lambda_{12} & \dots & \lambda_{1n} \\ \lambda_{21} & \lambda_{22} & \dots & \lambda_{2n} \\ \dots & \dots & \dots & \dots \\ \lambda_{m1} & \lambda_{m2} & \dots & \lambda_{mn} \end{array} \right\|, \quad (4.18)$$

де  $\lambda_{kj}$  – вага  $j$ -ї властивості в  $k$ -ї типовій ситуації функціонування системи.

Для того, щоб сформувати структуру властивостей системи, враховуючи різні типові ситуації її функціонування, необхідно отримати узагальнені ваги  $\lambda_j^*$  ( $j = \overline{1, n}$ ) кожної властивості, що найбільш повно враховують інформацію, закладену в матрицю  $A$ .

На основі матриці  $A$  визначається рівень кожної властивості:

$$H_j = -\frac{1}{m} \sum_{i=1}^m \lambda_{ij} \ln \lambda_{ij}, \quad (i=\overline{1,m}, j=\overline{1,n}). \quad (4.19)$$

Оскільки  $0 \leq H_j \leq 1$  ( $j=\overline{1,n}$ ), то логічно знайти рівень змінюваності  $j$  – ої властивості в межах задачі, що розглядається:

$$d_j = 1 - H_j \quad (j=\overline{1,n}). \quad (4.20)$$

Тоді узагальнені ваги властивостей можна обчислити за формулою:

$$\alpha_j = \frac{d_j}{\sum_{j=1}^n d_j}, \quad j=\overline{1,n}, \quad (4.21)$$

яка справедлива при умові, що всі ситуації вважаються однаково значущими, тобто немає експертних оцінок їх пріоритету. Якщо відомі ці оцінки, то доцільно знайти комплексну вагомість:

$$\lambda_j^* = \frac{\overline{\alpha_j} \alpha_j}{\sum_{j=1}^n \overline{\alpha_j} \alpha_j}, \quad j=\overline{1,n}. \quad (4.22)$$

Тут вектор  $\overline{\alpha} = \{ \overline{\alpha}_1, \overline{\alpha}_2, \dots, \overline{\alpha}_n \}$  можна визначити за матрицею експертних оцінок кожної ситуації. Якщо провести групову експертизу, в якій кожний із  $s$  членів експертної комісії призначає свої значення коефіцієнтів, що задовольняють умовам:

$$\sum_{j=1}^n \overline{\alpha}_{kj} = 1, \quad \overline{\alpha}_{kj} \geq 0, \quad j=\overline{1,n}, \quad k=\overline{1,s}, \quad (4.23)$$

то в результаті буде отримана матриця експертних оцінок:

$$A = \left\| \begin{array}{cccc} \overline{\alpha}_{11} & \overline{\alpha}_{12} & \dots & \overline{\alpha}_{1n} \\ \overline{\alpha}_{21} & \overline{\alpha}_{22} & \dots & \overline{\alpha}_{2n} \\ \dots & \dots & \dots & \dots \\ \overline{\alpha}_{m1} & \overline{\alpha}_{m2} & \dots & \overline{\alpha}_{mn} \end{array} \right\|, \quad (4.24)$$

де  $\bar{\alpha}_{kj}$  – експертна оцінка відносної значимості  $j$ -ї ситуації, пропонована  $k$ -м екпертом.

Для визначення оптимально - компромісних вагових коефіцієнтів  $\bar{\alpha}^*$ , що виражають “колективну думку”, задається схема компромісу  $F(A, \bar{\alpha}^*)$  і вирішується екстремальна задача:

$$F(A, \bar{\alpha}^*) = \min_{\bar{\alpha} \in D} F(A, \bar{\alpha}), \quad (4.25)$$

де вектор  $D$  має наступний вигляд:

$$D = \left\{ \bar{\alpha} \mid \sum_{j=1}^n \bar{\alpha}_j = 1, \bar{\alpha}_j \geq 1, j = \overline{1, n} \right\}. \quad (4.26)$$

Дана схема компромісу  $F(A, \bar{\alpha})$  є деякою мірою зближення між довільним вектором  $\bar{\alpha} \in D$  та елементами матриці  $A$ . В якості міри зближення  $F(A, \bar{\alpha})$  може слугувати функція:

$$F(A, \bar{\alpha}) = \sum_{k=1}^s \sum_{j=1}^n (\bar{\alpha}_{kj} - \bar{\alpha}_j)^2. \quad (4.27)$$

Тоді оптимальним вирішенням задачі (4.27) буде вектор середніх значень по елементам стовбчиків матриці  $A$ :

$$\bar{\alpha}_j^* = \frac{1}{s} \sum_{k=1}^s \bar{\alpha}_{kj}, \quad j = \overline{1, n}. \quad (4.28)$$

Для врахування кваліфікації експертів доцільно ввести коефіцієнти їх компетентності:

$$h_k > 0, \quad k = \overline{1, s}, \quad \sum_{k=1}^s h_k = 1. \quad (4.29)$$

Значення  $\bar{\alpha}_j^*$  розраховуються за формулою:

$$\bar{\alpha}_j^* = \sum_{k=1}^s \bar{\alpha}_{kj} \cdot h_k. \quad (4.30)$$

Отриманий вектор  $A^* = \{\lambda^*_1, \lambda^*_2, \dots, \lambda^*_n\}$ , використовується в якості вихідної інформації для формування структури властивостей системи, оскільки вага  $\lambda^*_j$  вказує положення  $j$ -ї властивості в загальній структурі властивостей.

Процедуру побудови моделей структури властивостей для експертної системи можливо розглянути на прикладі обстеження технічного стану складської будівлі.

В організаціях, які займаються обстеженням, існують підрозділи в яких, як правило, формуються експертна група з 2 – 5 чоловік, що проводить обстеження технічного стану будівель (рис. 4.2).



Рисунок 4.2 Структурна схема експертної групи, що формується в спеціалізованих організаціях

На рис. 4.3 представлені п'ять суттєвих ознак пошкодження основних конструктивних елементів будівлі, що розглядається.



Рисунок 4.3 Схема ознак пошкодження конструктивних елементів будівлі

Кожен з чотирьох експертів при проведенні обстеженні п'яти конструктивних елементів визначає ширину розкриття тріщини елемента та ставить у відповідність категорію технічного стану даного обстежуемого конструктивного елемента (2, 2/3, 3, 3/4, 4).

Результати експертних оцінок ( $H$ ) категорії технічного стану ( $I$ ), ширини розкриття тріщин ( $V$ ) та ознак пошкодження ( $M$ ) зведені в табл. 4.5.

По залежності (4.17) знайдені компоненти вагового вектора  $\lambda_{ij}$  для кожного з чотирьох експертів, тобто:

Таблиця 4.5

### Значення компонент вагових векторів

Експерти		Ознаки пошкодження				
		$m_1$	$m_2$	$m_3$	$m_4$	$m_5$
$h_1$	$I$	2	3	3/4	2/3	2/3
	$V$	1.5	2.5	3.5	2.0	2.7
	$\lambda$	0.488	0.325	0.130	0.037	0.018
$h_2$	$I$	3	4	2/3	3	2
	$V$	3.0	3.9	2.2	2.8	2.0

	$\lambda$	0.679	0.226	0.058	0.026	0.009
$h_3$	$I$	4	3/4	2	2	3/4
	$V$	4.0	3.0	1.8	2.0	3.5
	$\lambda$	0.713	0.179	0.059	0.033	0.016
$h_4$	$I$	2/3	2	4	4	3
	$V$	2.0	1.5	4.0	3.5	2.7
	$\lambda$	0.515	0.258	0.172	0.043	0.012

### ***I – експерт***

$$\begin{aligned}\lambda_{11} &= V_1 V_2 V_3 V_4 V_5 / V_1 V_2 V_3 V_4 V_5 + V_2 V_3 V_4 V_5 + V_3 V_4 V_5 + V_4 V_5 + V_5 = \\ &= 1.5 * 2.5 * 3.5 * 2.0 * 2.7 / 1.5 * 2.5 * 3.5 * 2.0 * 2.7 + 2.5 * 3.5 * 2.0 * 2.7 + \\ &+ 3.5 * 2.0 * 2.7 + 2.0 * 2.7 + 2.7 = 70.88 / 145.13 = 0.488;\end{aligned}$$

$$\begin{aligned}\lambda_{12} &= V_2 V_3 V_4 V_5 / V_1 V_2 V_3 V_4 V_5 + V_2 V_3 V_4 V_5 + V_3 V_4 V_5 + V_4 V_5 + V_5 = \\ &= 2.5 * 3.5 * 2.0 * 2.7 / 145.13 = 0.325;\end{aligned}$$

$$\begin{aligned}\lambda_{13} &= V_3 V_4 V_5 / V_1 V_2 V_3 V_4 V_5 + V_2 V_3 V_4 V_5 + V_3 V_4 V_5 + V_4 V_5 + V_5 = \\ &= 3.5 * 2.0 * 2.7 / 145.13 = 0.130;\end{aligned}$$

$$\begin{aligned}\lambda_{14} &= V_4 V_5 / V_1 V_2 V_3 V_4 V_5 + V_2 V_3 V_4 V_5 + V_3 V_4 V_5 + V_4 V_5 + V_5 = \\ &= 2.0 * 2.7 / 145.13 = 0.037;\end{aligned}$$

$$\begin{aligned}\lambda_{15} &= V_5 / V_1 V_2 V_3 V_4 V_5 + V_2 V_3 V_4 V_5 + V_3 V_4 V_5 + V_4 V_5 + V_5 = \\ &= 2.7 / 145.13 = 0.018.\end{aligned}$$

### ***II – експерт***

$$\begin{aligned}\lambda_{21} &= V_1 V_2 V_3 V_4 V_5 / V_1 V_2 V_3 V_4 V_5 + V_2 V_3 V_4 V_5 + V_3 V_4 V_5 + V_4 V_5 + V_5 = \\ &= 3.0 * 3.9 * 2.2 * 2.8 * 2.0 / 3.0 * 3.9 * 2.2 * 2.8 * 2.0 + 3.9 * 2.2 * 2.8 * 2.0 + \\ &+ 2.2 * 2.8 * 2.0 + 2.8 * 2.0 + 2.0 = 144.14 / 212.11 = 0.679;\end{aligned}$$

$$\begin{aligned}\lambda_{22} &= V_2 V_3 V_4 V_5 / V_1 V_2 V_3 V_4 V_5 + V_2 V_3 V_4 V_5 + V_3 V_4 V_5 + V_4 V_5 + V_5 = \\ &= 3.9 * 2.2 * 2.8 * 2.0 / 212.11 = 0.226;\end{aligned}$$

$$\begin{aligned}\lambda_{23} &= V_3 V_4 V_5 / V_1 V_2 V_3 V_4 V_5 + V_2 V_3 V_4 V_5 + V_3 V_4 V_5 + V_4 V_5 + V_5 = \\ &= 2.2 * 2.8 * 2.0 / 212.11 = 0.058;\end{aligned}$$

$$\lambda_{24} = V_4 V_5 / V_1 V_2 V_3 V_4 V_5 + V_2 V_3 V_4 V_5 + V_3 V_4 V_5 + V_4 V_5 + V_5 =$$

$$= 2.8*2.0 / 212.11 = 0.026;$$

$$\lambda_{25} = V_5 / V_1 V_2 V_3 V_4 V_5 + V_2 V_3 V_4 V_5 + V_3 V_4 V_5 + V_4 V_5 + V_5 = \\ = 2.0 / 212.11 = 0.009.$$

### **III – експерт**

$$\lambda_{31} = V_1 V_2 V_3 V_4 V_5 / V_1 V_2 V_3 V_4 V_5 + V_2 V_3 V_4 V_5 + V_3 V_4 V_5 + V_4 V_5 + V_5 = \\ = 4.0 * 3.0 * 1.8 * 2.0 * 3.5 / 4.0 * 3.0 * 1.8 * 2.0 * 3.5 + 3.0 * 1.8 * 2.0 * 3.5 + \\ + 1.8 * 2.0 * 3.5 + 2.0 * 3.5 + 3.5 = 151.20 / 212.10 = 0.713;$$

$$\lambda_{32} = V_2 V_3 V_4 V_5 / V_1 V_2 V_3 V_4 V_5 + V_2 V_3 V_4 V_5 + V_3 V_4 V_5 + V_4 V_5 + V_5 = \\ = 3.0 * 1.8 * 2.0 * 3.5 / 212.10 = 0.179;$$

$$\lambda_{33} = V_3 V_4 V_5 / V_1 V_2 V_3 V_4 V_5 + V_2 V_3 V_4 V_5 + V_3 V_4 V_5 + V_4 V_5 + V_5 = \\ = 1.8 * 2.0 * 3.5 / 212.10 = 0.059;$$

$$\lambda_{34} = V_4 V_5 / V_1 V_2 V_3 V_4 V_5 + V_2 V_3 V_4 V_5 + V_3 V_4 V_5 + V_4 V_5 + V_5 = \\ = 2.0 * 3.5 / 212.10 = 0.033;$$

$$\lambda_{35} = V_5 / V_1 V_2 V_3 V_4 V_5 + V_2 V_3 V_4 V_5 + V_3 V_4 V_5 + V_4 V_5 + V_5 = \\ = 3.5 / 212.10 = 0.016.$$

### **IV – експерт**

$$\lambda_{41} = V_1 V_2 V_3 V_4 V_5 / V_1 V_2 V_3 V_4 V_5 + V_2 V_3 V_4 V_5 + V_3 V_4 V_5 + V_4 V_5 + V_5 = \\ = 2.0 * 1.5 * 4.0 * 3.5 * 2.7 / 2.0 * 1.5 * 4.0 * 3.5 * 2.7 + 1.5 * 4.0 * 3.5 * 2.7 + \\ + 4.0 * 3.5 * 2.7 + 3.5 * 2.7 + 2.7 = 113.40 / 220.05 = 0.515;$$

$$\lambda_{42} = V_2 V_3 V_4 V_5 / V_1 V_2 V_3 V_4 V_5 + V_2 V_3 V_4 V_5 + V_3 V_4 V_5 + V_4 V_5 + V_5 = \\ = 1.5 * 4.0 * 3.5 * 2.7 / 220.05 = 0.258;$$

$$\lambda_{43} = V_3 V_4 V_5 / V_1 V_2 V_3 V_4 V_5 + V_2 V_3 V_4 V_5 + V_3 V_4 V_5 + V_4 V_5 + V_5 = \\ = 4.0 * 3.5 * 2.7 / 220.05 = 0.172;$$

$$\lambda_{44} = V_4 V_5 / V_1 V_2 V_3 V_4 V_5 + V_2 V_3 V_4 V_5 + V_3 V_4 V_5 + V_4 V_5 + V_5 = \\ = 3.5 * 2.7 / 220.05 = 0.043;$$

$$\lambda_{45} = V_5 / V_1 V_2 V_3 V_4 V_5 + V_2 V_3 V_4 V_5 + V_3 V_4 V_5 + V_4 V_5 + V_5 = \\ = 2.7 / 220.05 = 0.012.$$



На основі матриці  $L$  визначається рівень кожної властивості  $H_j$ , які зведені в табл. 4.6 згідно виду (4.19 – 4.23):

Таблиця 4.6

**Результати розрахунків вагових коефіцієнтів**

	$m_1$	$m_2$	$m_3$	$m_4$	$m_5$
$H_j$	0.300	0.340	0.225	0.116	0.058
$d_j$	0.700	0.660	0.775	0.884	0.942
$a_j$	0.177	0.167	0.196	0.223	0.238
$\overline{\alpha_j^*}$	0.410	0.266	0.078	0.166	0.123
$\overline{\alpha_j^0} = \lambda_j^*$	1.84	1,11	0.379	0.934	0.732

$$H_1 = - 1/4 [\lambda_{11} \ln \lambda_{11} + \lambda_{21} \ln \lambda_{21} + \lambda_{31} \ln \lambda_{31} + \lambda_{41} \ln \lambda_{41}] =$$

$$= - 1/4 [0.488*(-0.717) + 0.679*(-0.387) + 0.713*(-0.338) + 0.515*$$

$$* (-0.663)] = -1/4 [-0.350-0.263-0.241-0.341] = - 1/4 * (-1.200) = 0.300;$$

$$H_2 = - 1/4 [\lambda_{21} \ln \lambda_{21} + \lambda_{22} \ln \lambda_{22} + \lambda_{32} \ln \lambda_{32} + \lambda_{42} \ln \lambda_{42}] =$$

$$= - 1/4 [0.325*(-1.120) + 0.226*(-1.490) + 0.179*(-1.720) + 0.258*$$

$$* (-1.350)] = - 1/4 * (-1.360) = 0.340;$$

$$H_3 = - 1/4 [\lambda_{31} \ln \lambda_{31} + \lambda_{23} \ln \lambda_{23} + \lambda_{33} \ln \lambda_{33} + \lambda_{43} \ln \lambda_{43}] =$$

$$= - 1/4 [0.130*(-2.04) + 0.058*(-2.85) + 0.059*(-2.83) + 0.172*(-1.76)] =$$

$$- 1/4 * (-0.900) = 0.225;$$

$$H_4 = - 1/4 [\lambda_{41} \ln \lambda_{41} + \lambda_{24} \ln \lambda_{24} + \lambda_{34} \ln \lambda_{34} + \lambda_{44} \ln \lambda_{44}] =$$

$$= - 1/4 [0.037*(-3.30) + 0.026*(-3.65) + 0.033*(-3.41) + 0.043*(-3.15)] =$$

$$- 1/4 * (-0.464) = 0.116;$$

$$H_5 = - 1/4 [\lambda_{51} \ln \lambda_{51} + \lambda_{25} \ln \lambda_{25} + \lambda_{35} \ln \lambda_{35} + \lambda_{45} \ln \lambda_{45}] =$$

$$= - 1/4 [0.018*(-4.02) + 0.009*(-4.71) + 0.016*(-4.13) + 0.012*(-4.42)] =$$

$$- 1/4 * (-0.233) = 0.058.$$

$$d_1 = 1 - H_1 = 1 - 0.300 = 0.700; d_2 = 1 - H_2 = 1 - 0.340 = 0.660;$$

$$d_3 = 1 - H_3 = 1 - 0.225 = 0.775; d_4 = 1 - H_4 = 1 - 0.116 = 0.884;$$

$$d_5 = 1 - H_5 = 1 - 0.058 = 0.942.$$

Подальше розраховуємо узагальнені ваги, тобто:

$$a_1 = d_1/d_1 + d_2 + d_3 + d_4 + d_5 = 0.700/0.700 + 0.660 + 0.775 + 0.884 + 0.942 = 0.700/3.961 = 0.177;$$

$$a_2 = d_2/d_1 + d_2 + d_3 + d_4 + d_5 = 0.660/0.700 + 0.660 + 0.775 + 0.884 + 0.942 = 0.660/3.961 = 0.167;$$

$$a_3 = d_3/d_1 + d_2 + d_3 + d_4 + d_5 = 0.775/0.700 + 0.660 + 0.775 + 0.884 + 0.942 = 0.775/3.961 = 0.196;$$

$$a_4 = d_4/d_1 + d_2 + d_3 + d_4 + d_5 = 0.884/0.700 + 0.660 + 0.775 + 0.884 + 0.942 = 0.884/3.961 = 0.223;$$

$$a_5 = d_5/d_1 + d_2 + d_3 + d_4 + d_5 = 0.942/0.700 + 0.660 + 0.775 + 0.884 + 0.942 = 0.942/3.961 = 0.238.$$

Враховуючи експертну інформацію про значимість властивостей, визначається комплексна значимість  $\lambda_j^*$ , яка зведена в таблиці 3.6 згідно виду (4.24 – 4.30):

$$A = \begin{vmatrix} 0.41 & 0.21 & 0.10 & 0.16 & 0.12 \\ 0.37 & 0.33 & 0.08 & 0.17 & 0.282 \\ 0.50 & 0.26 & 0.03 & 0.15 & 0.04 \\ 0.44 & 0.23 & 0.13 & 0.14 & 0.06 \\ 0.33 & 0.30 & 0.05 & 0.21 & 0.11 \end{vmatrix};$$

$$\lambda_1^* = \bar{a}_1 a_1 / 1/5 [\bar{a}_1 a_1 + \bar{a}_2 a_2 + \bar{a}_3 a_3 + \bar{a}_4 a_4 + \bar{a}_5 a_5] = 0.177 * 0.41 / 0.2 * [0.177 * 0.41 + 0.167 * 0.266 + 0.196 * 0.078 + 0.223 * 0.166 + 0.238 * 0.123] = 1.84;$$

$$\lambda_2^* = \bar{a}_2 a_2 / 1/5 [\bar{a}_1 a_1 + \bar{a}_2 a_2 + \bar{a}_3 a_3 + \bar{a}_4 a_4 + \bar{a}_5 a_5] = 0.167 * 0.266 / 0.2 * [0.177 * 0.41 + 0.167 * 0.266 + 0.196 * 0.078 + 0.223 * 0.166 + 0.238 * 0.123] = 1.11;$$

$$\lambda_3^* = \bar{a}_3 a_3 / 1/5 [\bar{a}_1 a_1 + \bar{a}_2 a_2 + \bar{a}_3 a_3 + \bar{a}_4 a_4 + \bar{a}_5 a_5] = 0.196 * 0.078 / 0.2 * [0.177 * 0.41 + 0.167 * 0.266 + 0.196 * 0.078 + 0.223 * 0.166 + 0.238 * 0.123] = 0.379;$$

$$\lambda_4^* = \bar{a}_4 a_4 / 1/5 [\bar{a}_1 a_1 + \bar{a}_2 a_2 + \bar{a}_3 a_3 + \bar{a}_4 a_4 + \bar{a}_5 a_5] = 0.223 * 0.166 / 0.2 * [0.177 * 0.41 + 0.167 * 0.266 + 0.196 * 0.078 + 0.223 * 0.166 + 0.238 * 0.123] = 0.934;$$

$$\lambda_5^* = \bar{a}_5 a_5 / 1/5 [\bar{a}_1 a_1 + \bar{a}_2 a_2 + \bar{a}_3 a_3 + \bar{a}_4 a_4 + \bar{a}_5 a_5] = 0.238 * 0.123 /$$
$$/ 0.2 * [0.177 * 0.41 + 0.167 * 0.266 + 0.196 * 0.078 + 0.223 * 0.166 + 0.238 * 0.123] = 0.732.$$

## **ВИСНОВКИ ПО РОЗДІЛУ 4**

1. Запропоновані та досліджені моделі і методи експертної оцінки системи підтримки прийняття рішень в задачах діагностики технічного стану будівель.

2. Розглянуті моделі та методи, що були підтверджені на прикладах експертної оцінки системи підтримки прийняття рішень в задачах діагностики технічного стану будівель.

## РОЗДІЛ 5

### ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА РЕАЛІЗАЦІЯ ЕКСПЕРТНОЇ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ В ЗАДАЧАХ ДІАГНОСТИКИ ТЕХНІЧНОГО СТАНУ БУДІВЕЛЬ

Будівлі є системами, що складаються з великого числа елементів та працюють в умовах складних станів, що деформуються. Поведінка будівельних конструкцій характеризується рядом чинників, що носять випадковий характер. Це відноситься до міцнісних характеристик матеріалів, навантаженням, діючим на елементи будівлі. В процесі виготовлення окремих елементів, їх монтажу можливі відхилення параметрів конструкцій від заданих значень. Тому для обстеження технічного стану необхідно прогнозувати можливість їх подальшої експлуатації з урахуванням взаємозв'язків і характеру формування властивостей.

Обробка експериментальних даних (результатів) експертної системи, проведення обстеження технічного стану на основі аналізу нечітких систем, розробки бази знань і ланцюгів логічного виведення оцінки основних дефектів та пошкоджень конструктивних елементів будівлі дозволить проводити моделювання експертної системи, аналізувати отримані результати експерименту роботи нечіткої системи, які підтверджуються теоретичними даними.

На рис. 5.1 представлена схема організації проведення експериментального дослідження технічного стану об'єктів обстеження.

Всі етапи системи нечіткого виводу можуть бути реалізовані по різному, так як включають в себе окремі параметри, які повинні бути фіксованими або специфіковані. Вибір конкретних варіантів параметрів

кожного із етапів визначає алгоритм, який в повному об'ємі реалізує нечіткий вивід в системах правил нечітких продукції.

В даний час розроблені і широко використовуються наступні алгоритми системи нечіткого виводу: алгоритм Мамдані, алгоритм Цукамото, алгоритм Ларсена, алгоритм Сугено.

Найбільш практичне застосування в задачах нечіткого моделювання в області діагностування технічного стану будівель є алгоритм виводу Мамдані та алгоритм Сугено, що дають можливість обґрунтувати прийняте рішення, дозволяють працювати з великими об'ємами даних, показують більш високу точність. Тому ці алгоритми мають найбільш практичне застосування, та в цілому дозволяють проводити спостереження деформаційних процесів та приймати заходи щодо подальшої експлуатації як конструктивних елементів так і будівлі в цілому, в залежності від побудови та застосовування бази правил системи нечіткого виводу.

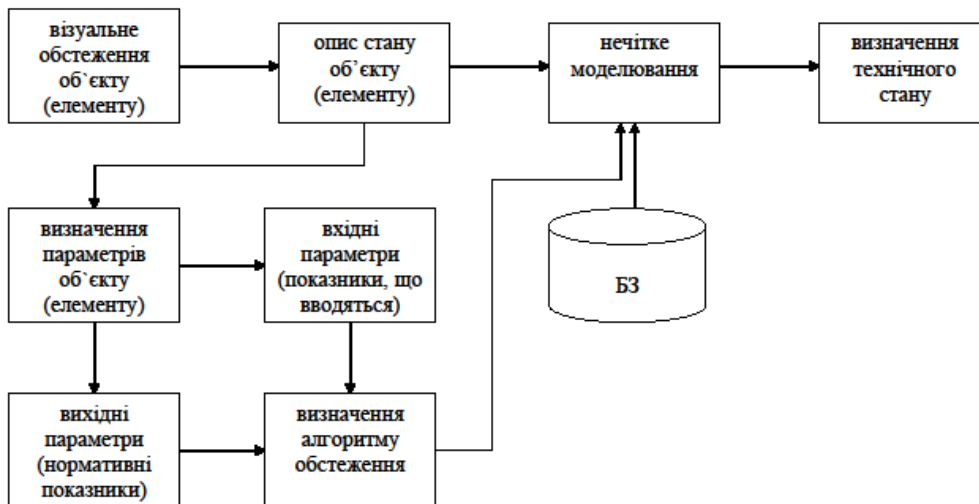


Рисунок 5.1 Схема організації проведення експериментального дослідження технічного стану об'єктів обстеження

Алгоритм Мамдані складається з наступних етапів:

1. Формування бази правил системи нечіткого виводу.
2. Фазифікація вхідних змінних.
3. Агрегація підумов в нечітких правилах продукцій. Для знаходження ступеню істиності умов кожного із правил нечітких продукцій використовуються парні нечітки логічні операції. Правила, ступінь істиності умов яких відмінна від нуля, вважається активними та застосовуються для розрахунків.
4. Активізація підвисновків в правилах нечітких продукцій.
5. Акумуляція висновків правил нечітких продукцій.
6. Дефазифікація вихідних змінних. Використовується метод центру тяжіння або метод центру площі.

На рис. 5.2 представлена загальна схема експертного дослідження технічного стану об'єктів методом центру тяжіння за алгоритмом Мамдані.

Розглянемо приклад експериментального дослідження при обстеженні технічного стану конструктивного елементу (тріщина в стіні).

Необхідно розробити модель діагностики технічного стану конструктивного елементу (тріщина в стіні) за алгоритмом Мамдані.

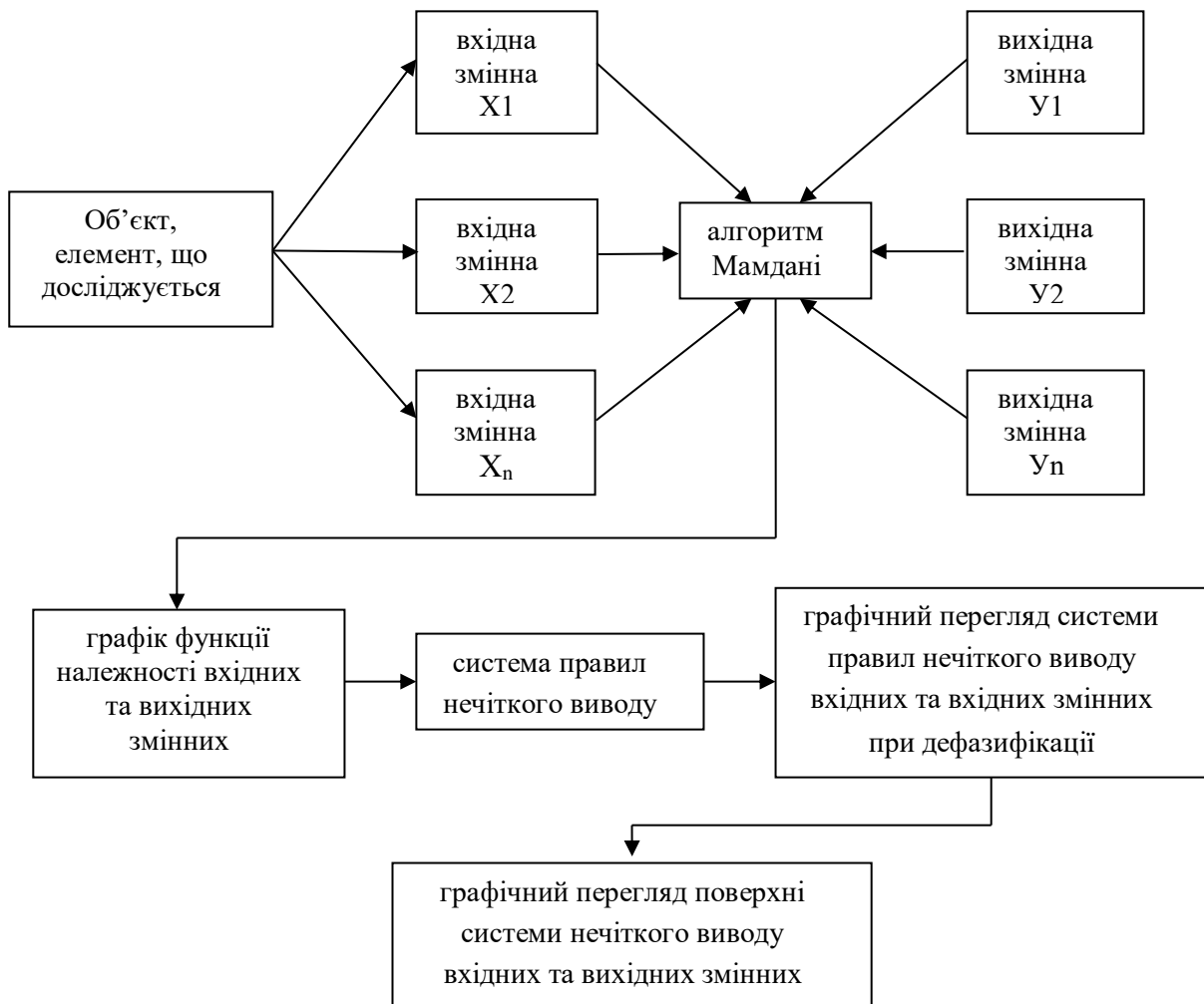


Рисунок 5.2 Загальна схема експертного дослідження технічного стану об'єкту методом центру тяжіння за алгоритмом Мамдані

На рис. 5.3 представлений приклад реалізації експертного дослідження технічного стану конструктивного елементу (тріщина в стіні). На цій структурній схемі запропонована методика дослідження технічного стану конструктивного елементу (тріщина в стіні) методом центру тяжіння за алгоритмом Мамдані при вхідних (ширина розкриття тріщин, що вводиться) змінних та вихідних (нормативні показники, показники фізичного зносу при визначенні категорії технічного стану конструкції) змінних.

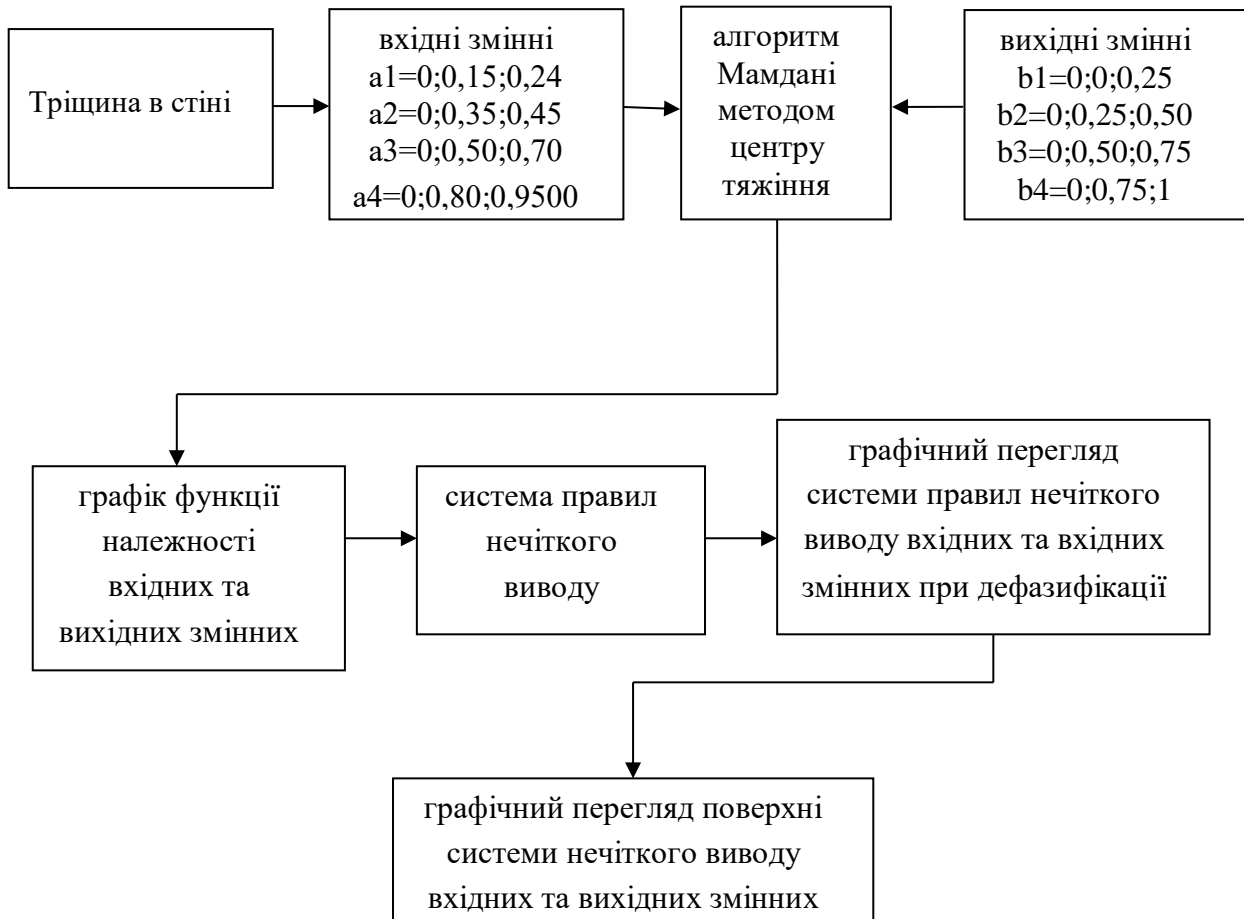


Рисунок 5.3 Приклад реалізації експертного дослідження технічного стану конструктивного елементу (тріщина в стіні) методом центру тяжіння за алгоритмом Мамдані

Для цього на початку обстеження (рис. 5.4) вводимо доволно по одній (можна декілька) вхідній лінгвістичній змінній (ширина розкриття тріщин, які розглянуті на прикладі та є типовими при обстеженні конструкції) та вихідній (можна декілька) лінгвістичній змінній (нормативні данні фізичного зносу ширини розкриття тріщин при визначенні категорії технічного стану конструкції керуючись положенням “Правила оцінки фізичного зношення будинків. ВСН 53-86(р)”.



На рис. 5.5 та рис. 5.6 представлений редактор функції належності, який в графічному режимі дозволяє аналізувати всі функції належності, змінювати ім'я, тип, параметри. Для кожної вхідної та вихідної лінгвістичної змінної водимо всі необхідні параметри, які необхідно аналізувати.

На рис. 5.7 представлений редактор правил системи нечіткого виводу, який дає можливість аналізувати та вводити правила продукції системи нечіткого виводу.

На рис. 5.8 представлений графічний перегляд системи нечіткого виводу, яка дозволяє можливість візуалізувати результати нечіткого виводу та отримувати значення вихідних змінних (значення 0,529) в залежності від початкових значень вхідних змінних (при середньому значенні 0,5 який встановлює експерт при дослідженні).

На рис. 5.9 представлений графічний перегляд поверхні системи нечіткого виводу, який візуалізувати графіки залежності вихідних змінних від окремих вхідних змінних.

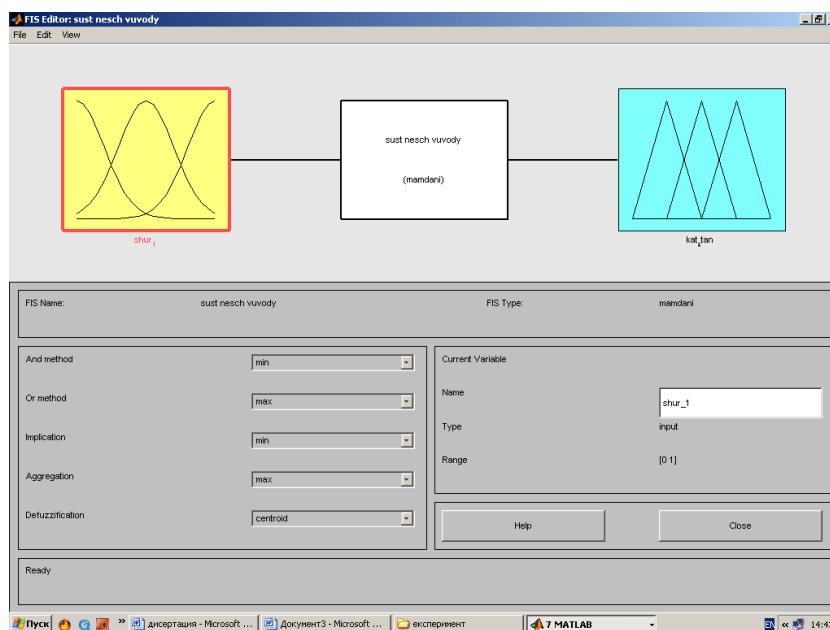


Рисунок 5.4 Графічний редактор системи нечіткого виводу

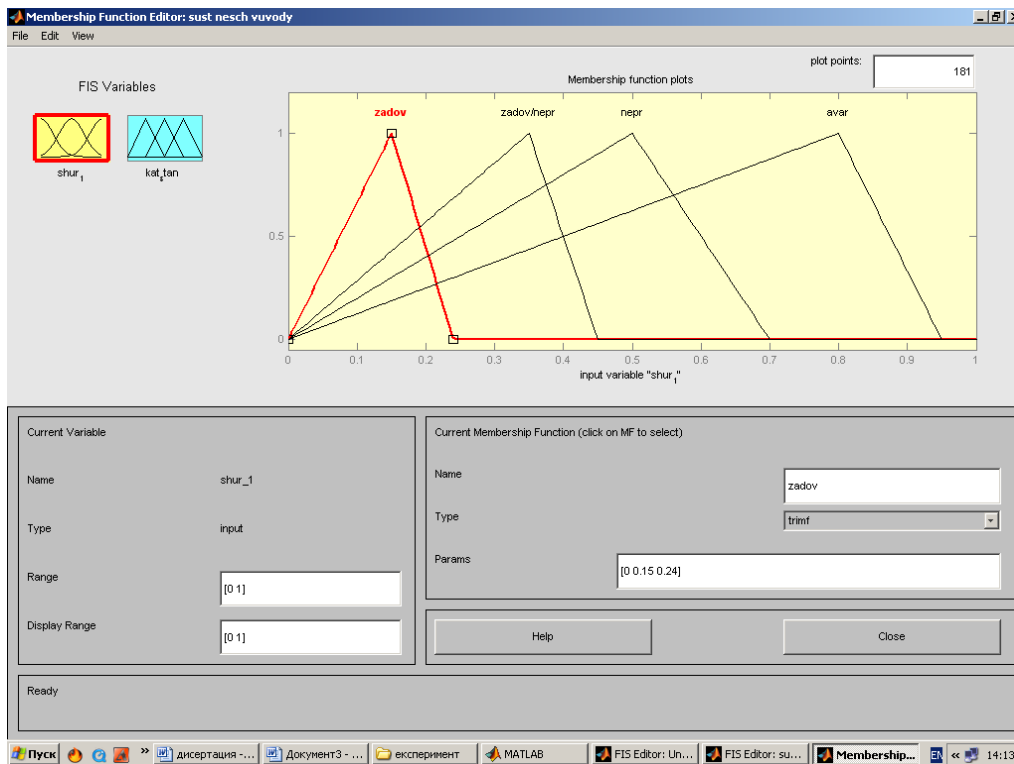


Рисунок 5.5 Редактор функції належності для вхідної змінної

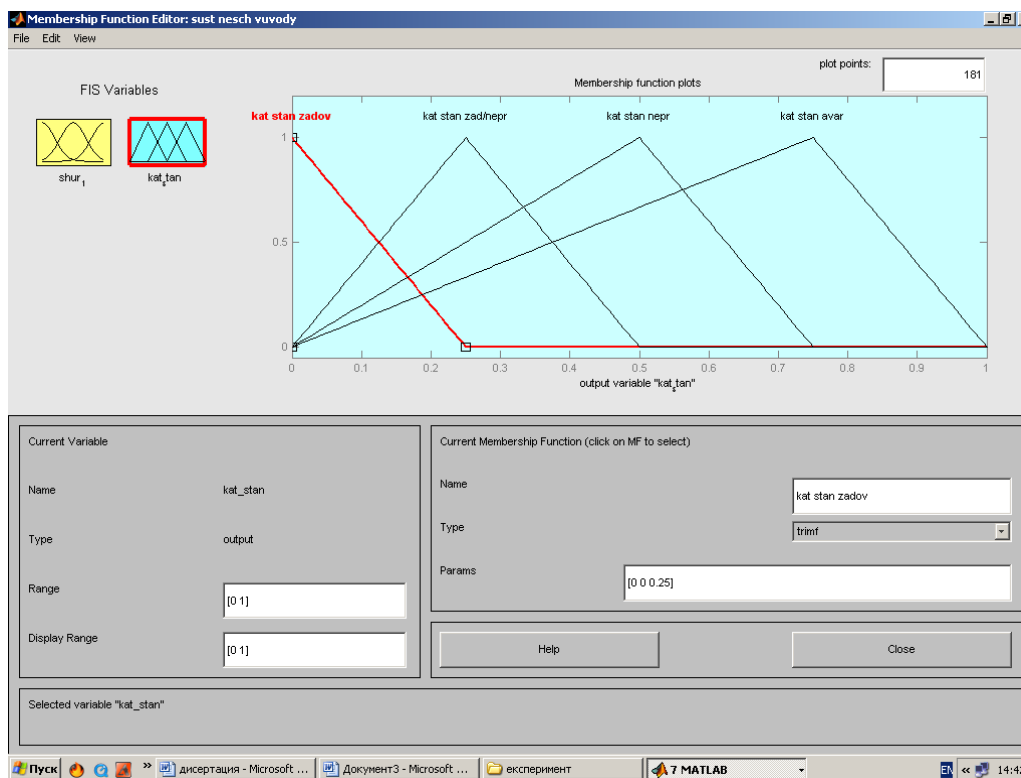


Рисунок 5.6 Редактор функції належності для вихідної змінної

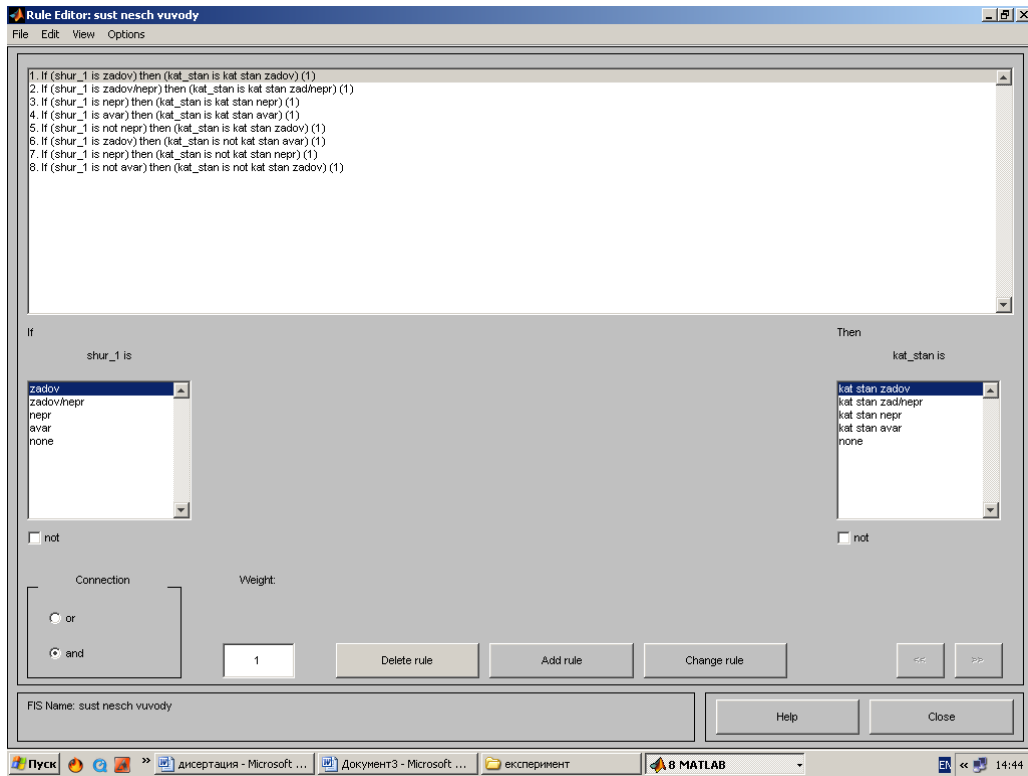


Рисунок 5.7 Редактор бази правил системи нечіткого виводу

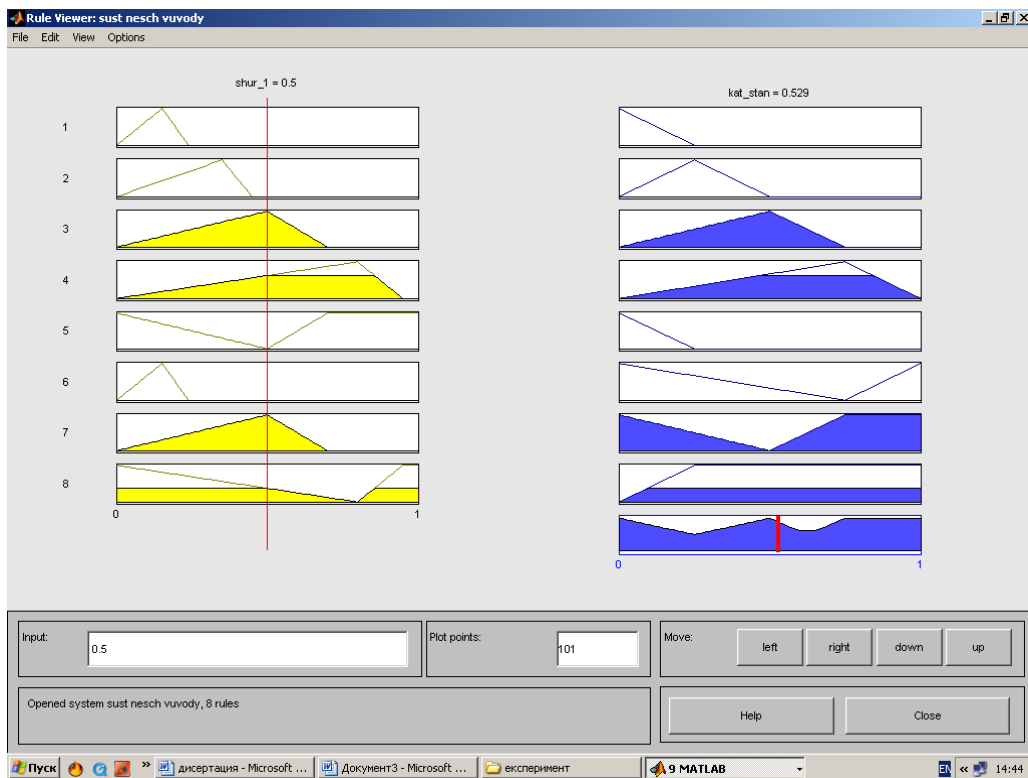


Рисунок 5.8 Графічний перегляд системи нечіткого виводу

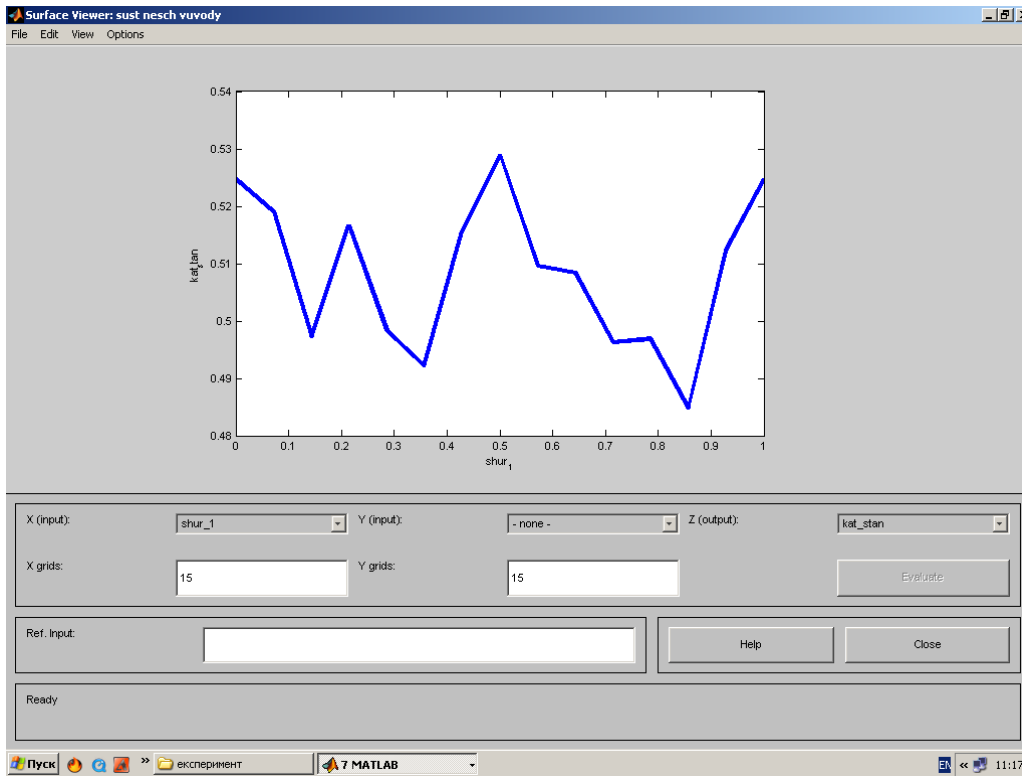


Рисунок 5.9 Графічний перегляд поверхні системи нечіткого виводу

В результаті проведеного експерименту, який був отриманий на прикладі дослідження технічного стану конструктивного елементу (тріщина в стіні) методом центру тяжіння за алгоритмом Мамдані при входних (ширина розкриття тріщин) змінних та вихідних (нормативні данні, показники фізичного зносу при визначенні категорії технічного стану елементу) змінних в подальшому дає можливість аналізувати та прогнозувати надалі як буде вести себе тріщина при різних входних данних в залежності від побудови бази правил системи нечіткого виводу та отримувати значення вихідних змінних в залежності від початкових значень входних змінних.

На рис. 5.9 отриманий графік поверхні системи нечіткого виводу, на якому відображені основні піки розвитку деформації пошкодженої конструкції в залежності від категорії технічного стану нормативних документів (*kat stan*) та ширини розкриття тріщин, що задається (*shur*).

На графіку можна проаналізувати, що найбільшим піком деформації тріщини в стіні є значення ширини розкриття тріщини 0,5 мм при категорії технічного стану 0,529 мм, яка означає, що тріщина стіни знаходиться в стані непридатної щодо нормальної та безпечної експлуатації.

Це все дає можливість в подальшому проводити процес спостереження деформаційних процесів та приймати заходи щодо подальшої експлуатації як конструктивних елементів так і будівлі в цілому, в залежності від побудови та застосовування бази правил системи нечіткого виводу.

Для порівняння на рис. 5.10 – 5.16 представлена система нечіткого виводу за алгоритмом Сугено. Для систем нечіткого виводу типу Сугено вибирається метод зваженого середнього.

Алгоритм Сугено складається з наступних етапів:

1. Формування бази правил системи нечіткого виводу. В базі правил застосовуються тільки правила нечітких продукцій і форми:

ПРАВИЛО <#>: ЯКЩО " $\beta_1 \in \alpha'$ "

І " $\beta_2 \in \alpha''$ ",

ТО " $w = \epsilon_1 \times a_1 + \epsilon_2 \times a_2$ ",

де  $\epsilon_1, \epsilon_2$  – вагові коефіцієнти. Значення вихідної змінної  $w$  в заключенні визначається як деяке дійсне число.

2. Фаззифікація вхідних змінних.

3. Агрегація підумов в нечітких правилах продукцій. Для знаходження ступеню істиності умов кожного із правил нечітких продукцій використовується логічна операція *min* – кон'юнкції. Правила, ступінь істиності умов яких відмінна від нуля, вважається активними та застосовуються для подальших розрахунків.

4. Активізація підвисновків в правилах нечітких продукцій. В результаті визначається множина значень та множина значень вихідних змінних.

5. Акумуляція висновків правил нечітких продукцій. Фактично відсутнє, так як розрахунки здійснюються з дійсними числами.

6. Дефазифікація вихідних змінних. Використовується модифікація метод центру тяжіння для одноточкових множин.

На рис. 5.10 представлений приклад реалізації експертного дослідження технічного стану конструктивного елементу (тріщина в стіні) методом зваженого середнього за алгоритмом Сугено.



Рисунок 5.10 Приклад реалізації експертного дослідження технічного стану конструктивного елементу (тріщина в стіні) методом зваженого середнього за алгоритмом Сугено

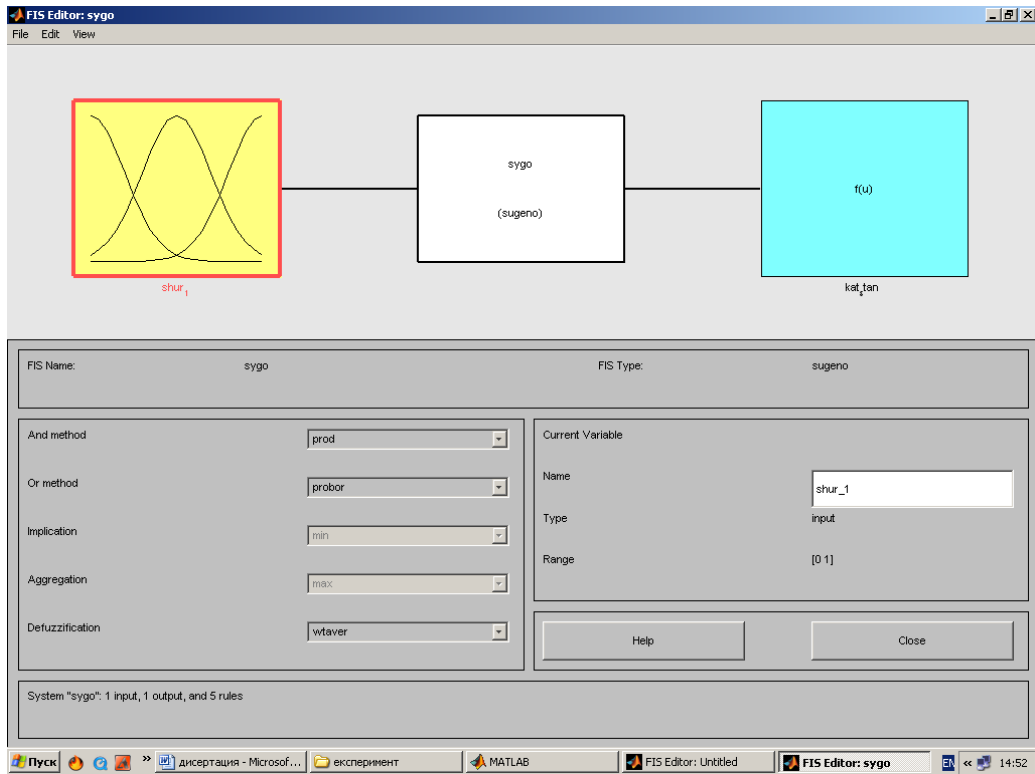


Рисунок 5.11 Графічний редактор системи нечіткого виводу

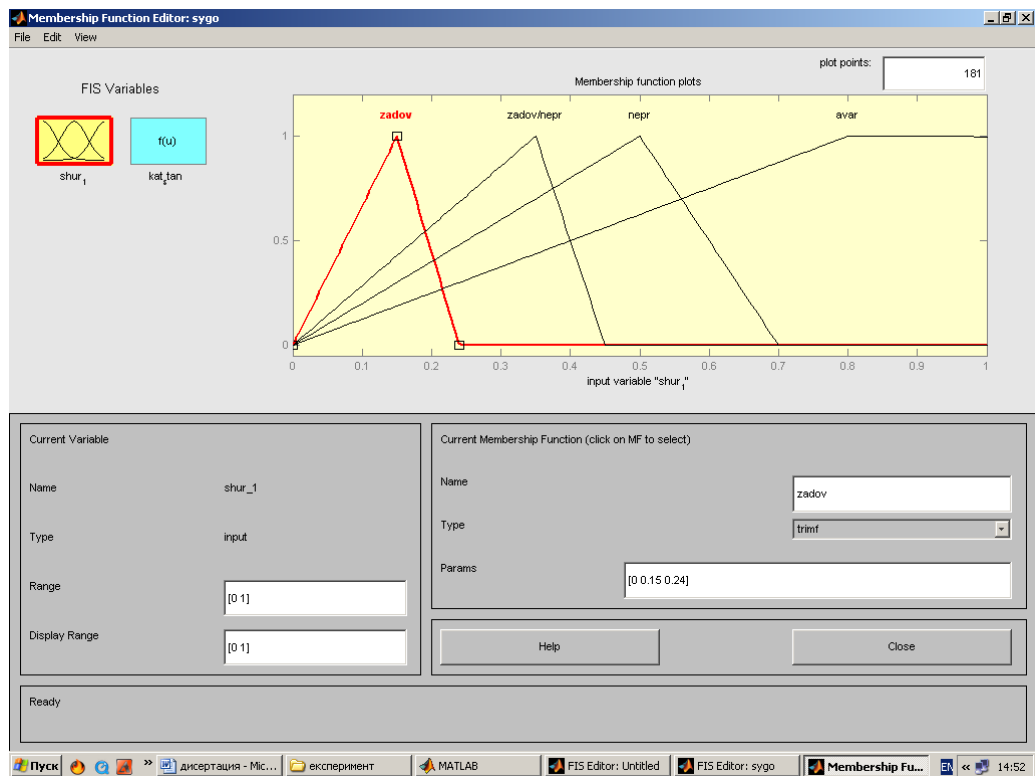


Рисунок 5.12 Редактор функції належності  
ширини розкриття тріщин

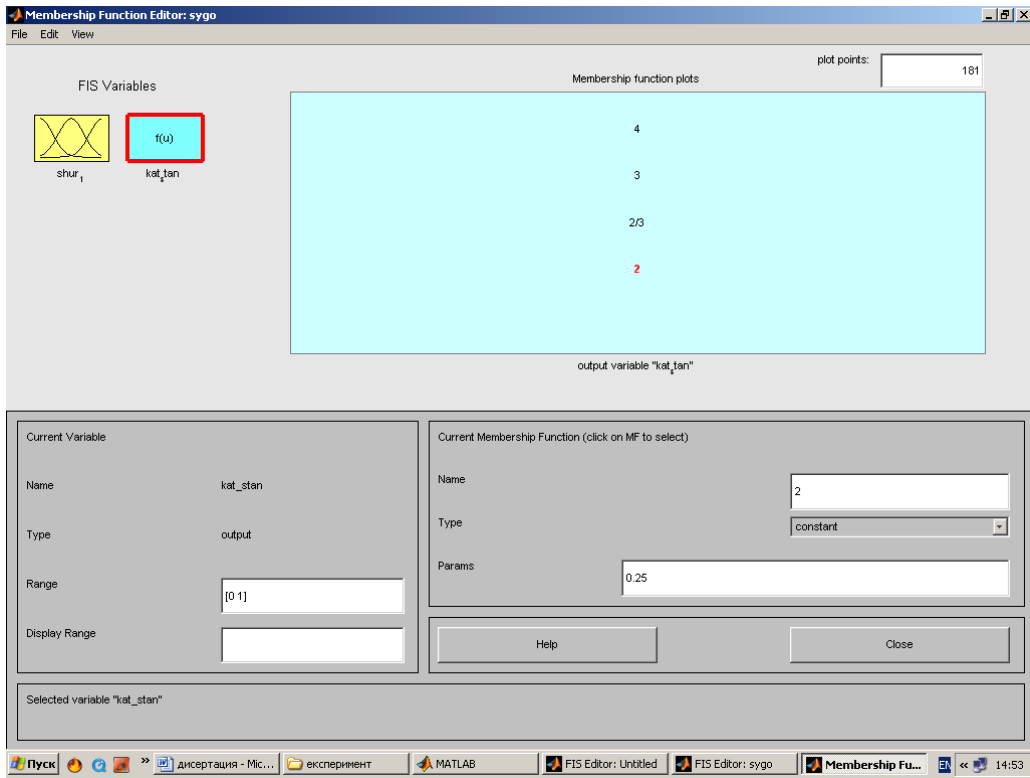


Рисунок 5.13 Редактор функції належності  
категорії технічного стану

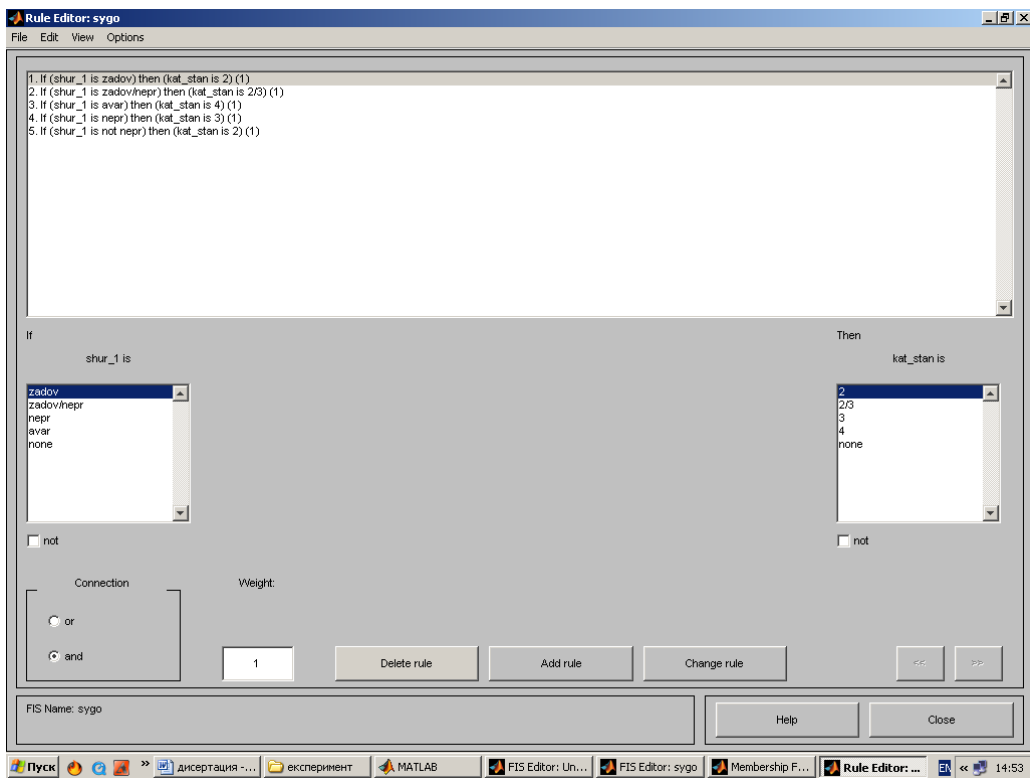


Рисунок 5.14 Редактор бази правил системи нечіткого виводу



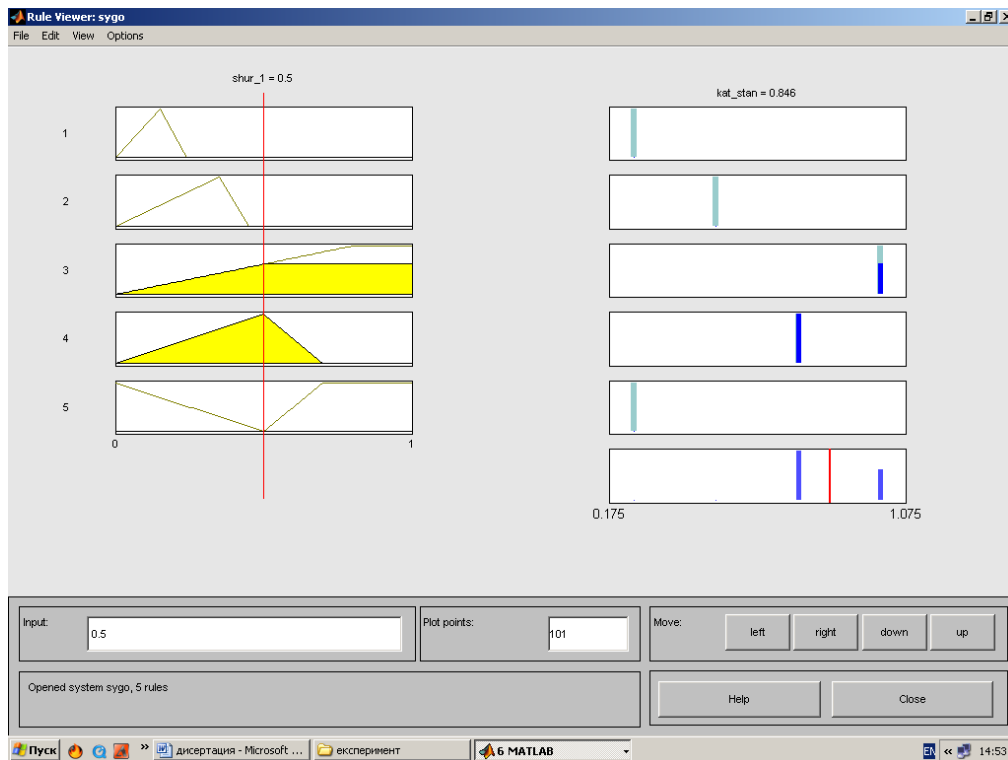


Рисунок 5.15 Графічний перегляд системи нечіткого виводу

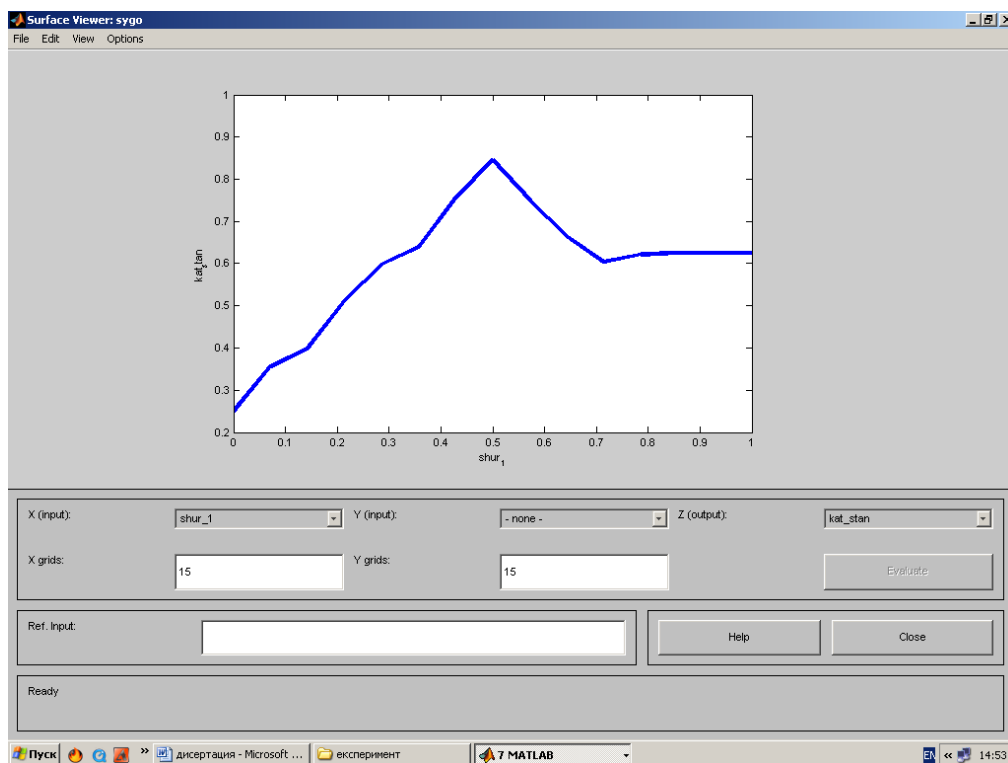


Рисунок 5.16 Графічний перегляд поверхні системи нечіткого виводу

В результаті проведеного експерименту на основі алгоритму Сугено (рис. 5.16) маємо, що на графіку поверхні системи нечіткого виводу, на якому відображені основні піки розвитку деформації пошкодженої конструкції в залежності від категорії технічного стану (*kat stan*) та ширини розкриття тріщин (*shur*), найбільшим піком деформації тріщини в стіні є значення ширини розкриття тріщини 0,5 мм при категорії технічного стану 0,846 мм, яка означає, що тріщина стіни знаходиться в стані аварійності.

Після порівняння алгоритмів Мамдані та Сугено можна сказати, що ці моделі відрізняються форматом бази знань та процедурою дефазифікації. Всі ці моделі є універсальними апроксиматорами, але при значних об'ємах виборки експериментальних даних ідентифікація за допомогою моделі типу Сугено забезпечує більшу точність. На прикладі дослідження видно, що найбільшим піком деформації є значення ширини розкриття тріщини 0,5 мм при категорії технічного стану 0,846 мм, яка означає, що тріщина стіни знаходиться в стані аварійності, але при цьому виникають труднощі зі змістовною інтерпретацією параметрів нечіткої моделі та поясненням логічного виводу. За допомогою моделі типу Мамдані таких труднощій не виникає, її параметри після навчання легко інтерпретируються змістовно. Тому, на прикладі дослідження видно, що найбільшим піком деформації є значення ширини розкриття тріщини 0,5 мм при категорії технічного стану 0,529 мм, яка означає, що тріщина стіни знаходиться в стані непридатної до нормальної експлуатації. Тому для задач, де більш важна точність ідентифікації, краще застосовувати нечіткі моделі типу Сугено, а для задач де більш важливим є пояснення, обґрунтування прийнятого рішення, будуть мати перевагу нечіткі моделі типу Мамдані.

Аналіз експерименту показав, що модель типу Сугено дозволяє працювати з великими об'ємами даних, показує більш високу точність, ніж модель типу Мамдані, при ідентифікації нелінійних залежностей.

## **ВИСНОВКИ ПО РОЗДІЛУ 5**

1. Проаналізовані результати експерименту з моделювання системи підтримки прийняття рішень в задачах діагностики технічного стану будівель.

2. Проведені експериментальні дослідження роботи нечіткої системи підтримки прийняття рішень в задачах діагностики технічного стану будівель та отримані результати, які підтверджені теоретичними даними.

## **РОЗДІЛ 6**

### **ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА РЕАЛІЗАЦІЯ НЕЙРОННОЇ МЕРЕЖІ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ В ЗАДАЧАХ ДІАГНОСТИКИ ТЕХНІЧНОГО СТАНУ БУДІВЕЛЬ**

#### **6.1 Архітектура інформаційної системи тестування нейронної мережі**

Для автоматизації процесів оціночної діяльності від збору інформації до прийняття рішення необхідна складна система збереження, підготовки, обробки даних. Архітектури інформаційної системи тестування нейронної мережі представлена на рис. 6.1.

Система передбачає наявність підсистеми збору статистичної інформації, обробки і розподілення даних для штучної нейронної мережі,

ядро нейромережевого апроксиматора та додатки експертів-оцінювачів для управління. СУБД інформаційної системи має клієнт-серверну архітектуру для забезпечення роботи декількох експертів одночасно, представлено на рис. 6.2.

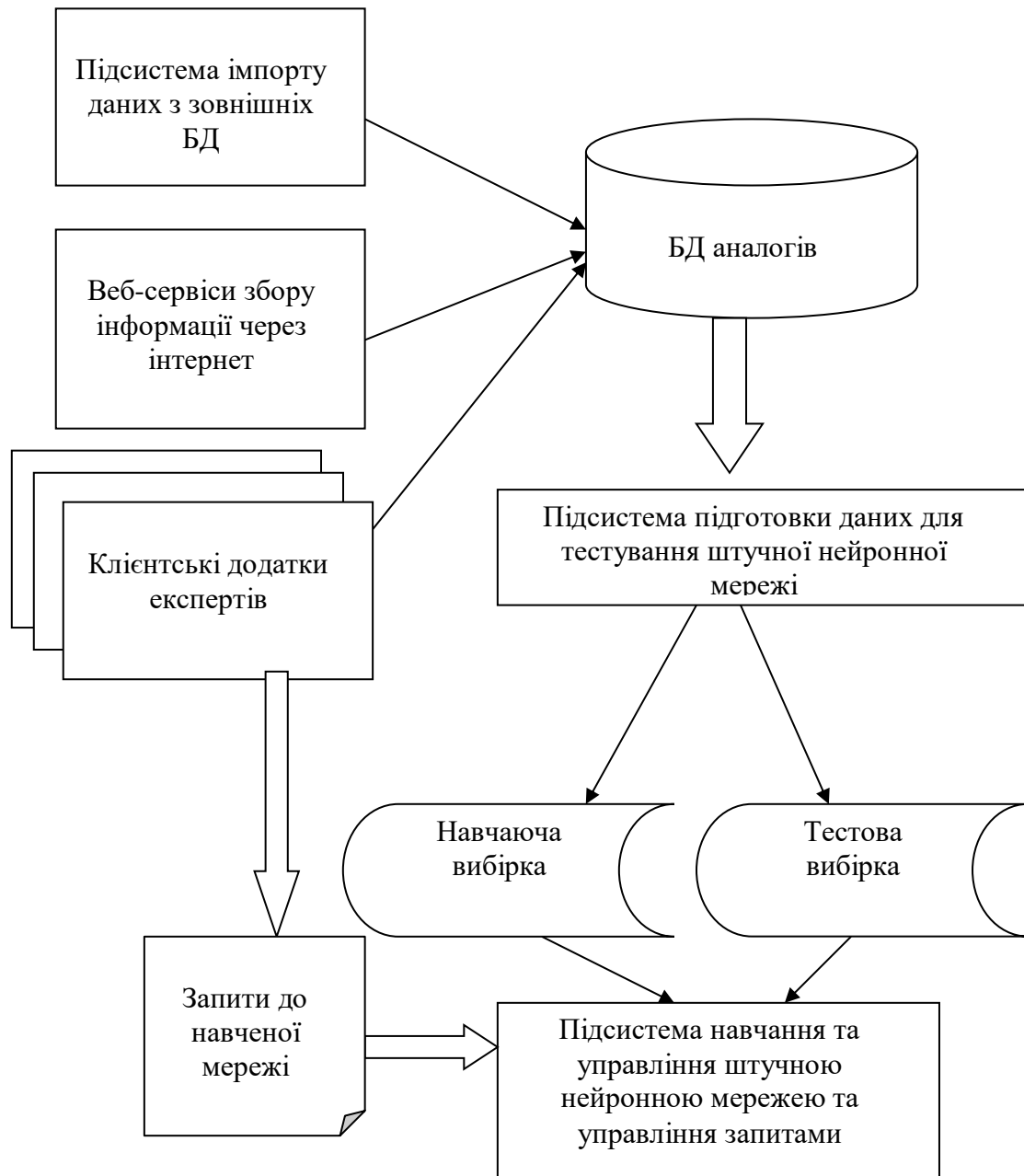


Рисунок 6.1 Архітектура інформаційної нейронно-мережевої системи тестування

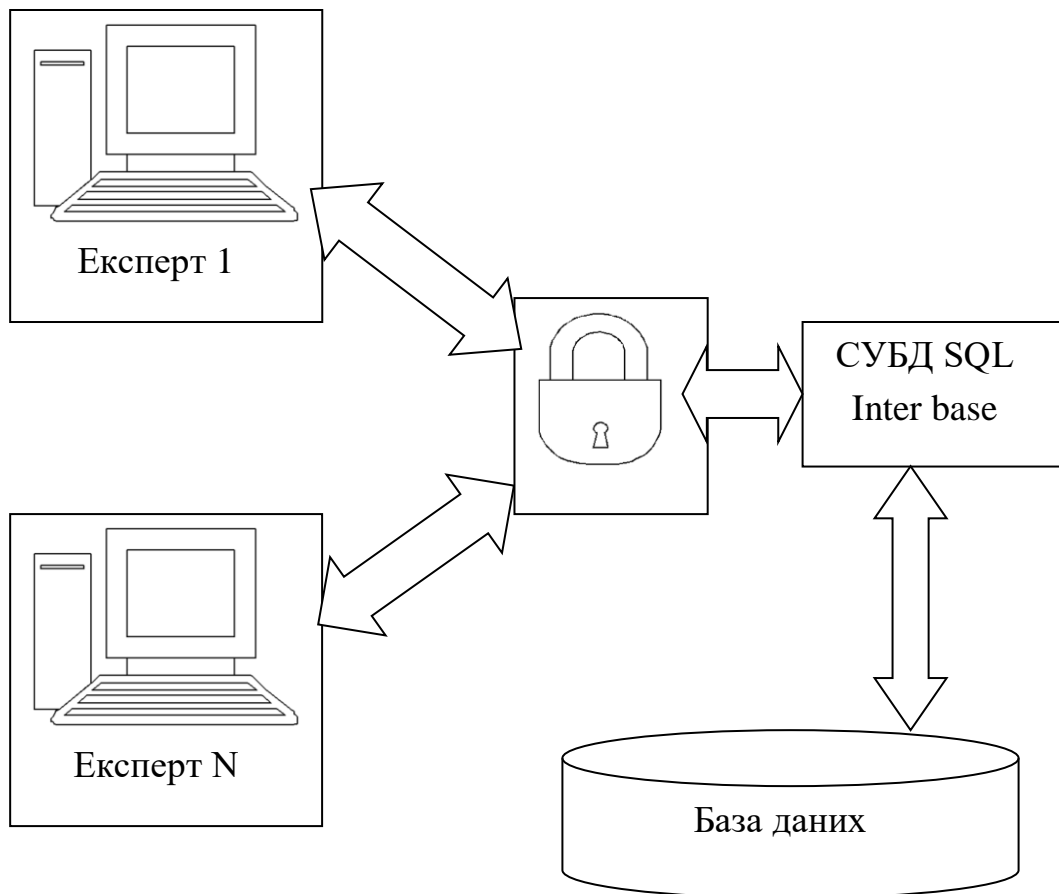


Рисунок 6.2 Клієнт-серверна архітектура підсистеми збору інформації

## 6.2. Підсистема підготовки інформації тестування нейронної мережі

Підсистема підготовки щодо тестування має свій конфігуратор, де користувач може визначити кількість даних, які необхідно виділити для тестування нейронної мережі і для навчання. Робота підсистеми починається з вибору даних з бази даних за допомогою запиту, що конфігурується.

Вся схема роботи підсистеми підготовки інформації щодо тестування нейронної мережі зображена на рис. 6.3.

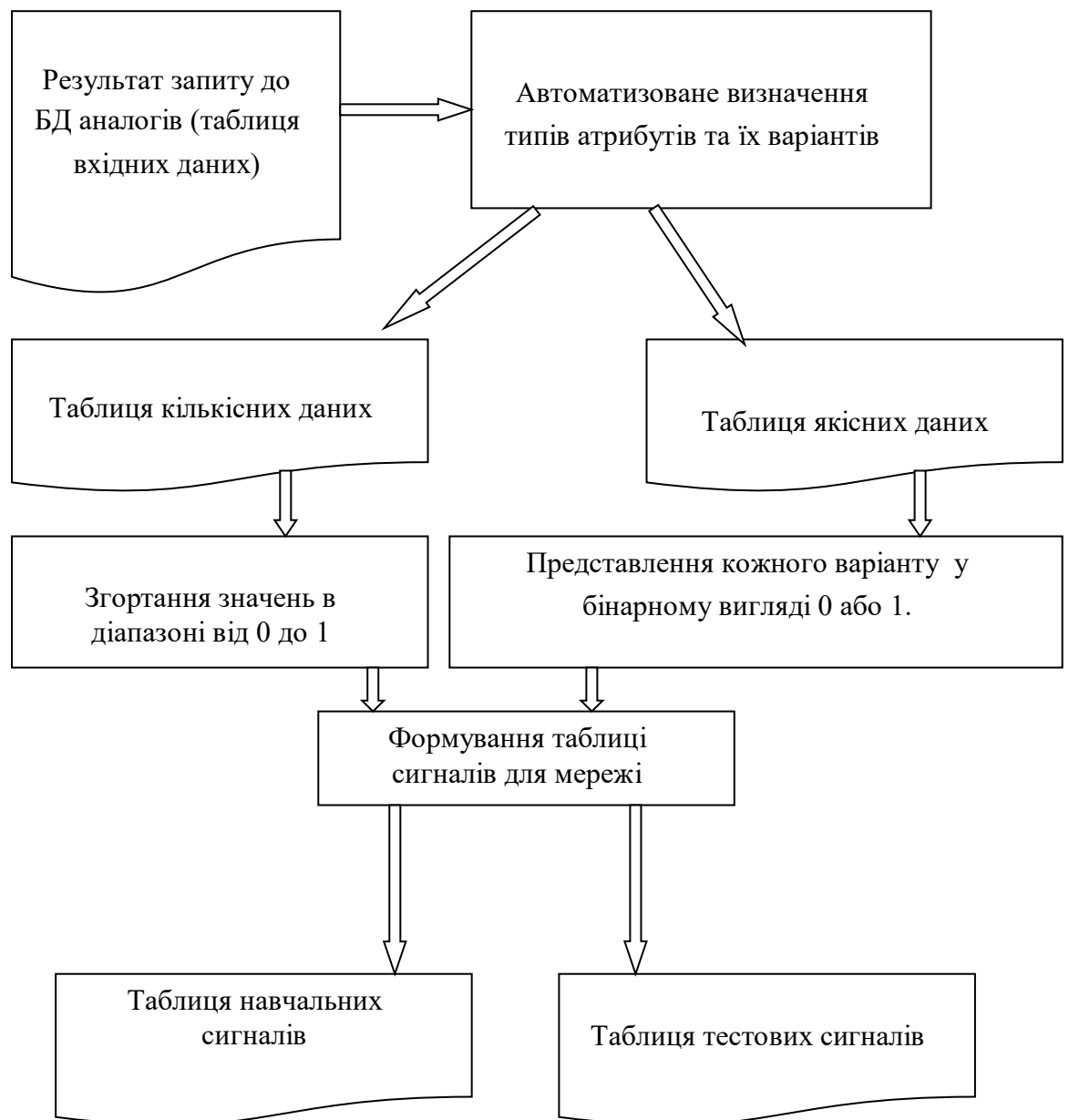


Рисунок 6.3 Підсистема підготовки інформації щодо тестування нейронної мережі

Після отримання результатів запиту дані надходять до модулю, що визначає типи атрибутів та їх варіанти даних. Потім дані, що мають числовий тип відокремлюються від даних, що мають текстовий тип. Оскільки мережа не може сприймати дані як текстовий тип, пропонується деталізувати таблицю нечислових даних, тобто кожен варіант представити як атрибут, що має значення тільки “0” або “1”.

Дані що є чисельно вираженими теж потребують обробки. Так як на вхід сигмоїдального нейрона подаються сигнали тільки в межах від 0 до 1, то значення таких параметрів як площа, ціна необхідно перемасштабувати. Приклад зведення значень таких факторів зображений на рис. 6.4.

### Перед обробкою

	S_TSH	S_TD	S_TV	S_TPVV	F_TP	F_CT	P_PV	P_PSH	P_PPP
	0	0	0,1	0	0	0	0,1	0,1	0
▶	0,05	0,05	1	0,05	0,05	0,05	1	0,05	0,05
	0,1	0,1	1,6	0,1	0,1	0,1	1,6	0,1	0,1
	0,125	0,125	1,9	0,125	0,125	0,125	1,9	0,125	0,125
	0,135	0,135	2	0,135	0,135	0,135	2	0,135	0,135

### Після обробки

	S_TSH	S_TD	S_TV	S_TPVV	F_TP	F_CT	P_PV	P_PSH	P_PPP
▶	0	0	0	0	0	0	0	0,016949152542...	0
	0,001666666666...	0,01	0,010011123470...	0,0005	0,0005	0,001666666666...	0,010011123470...	0	0,000625
	0,003333333333...	0,02	0,016685205784...	0,001	0,001	0,003333333333...	0,016685205784...	0,016949152542...	0,00125
	0,004166666666...	0,025	0,020022246941...	0,00125	0,00125	0,004166666666...	0,020022246941...	0,025423728813...	0,0015625
	0,004500000000...	0,027000000000...	0,021111450000...	0,00135	0,00135	0,004500000000...	0,021111450000...	0,020011550000...	0,001607500000...

Рисунок 6.4 Приклад зведення числових значень в діапазон від 0 до 1

### 6.3 Модель тестування нейронної мережі

Задача апроксимації функцій штучною нейронною мережею складність яких завідома невідома, потребує прикладного програмного забезпечення з широкими можливостями конфігурації. Для цього була розроблена об'єктно-орієнтована модель мережі, що дозволяє легко управляти її параметрами (кількістю прошарків, нейронів в прошарках). Модель дозволяє слідкувати за кожним нейроном окремо і в разі необхідності змінювати його конфігурацію (тип функції активації, швидкість навчання, вхідні сигнали та зв'язки з іншими нейронами). Модель мережі зображено на рис. 6.5.

Модель складається з трьох класів: нейрон, прошарок, мережа. Розглянемо детально кожен клас, його атрибути і методи.

**Клас Neuron** – моделює стан та поведінку штучного нейрона.

Властивості класу Neuron:

`public double requiredF` – має значення потрібного вихідного сигналу на виході нейрона;

`public int countOfSinaps` – кількість синапсів (вхідних сигналів). Якщо це прихований шар, кількість вхідних сигналів дорівнює кількості нейронів попереднього шару;

`public double speed` – швидкість навчання. Параметр визначає величину зміни приросту при корекції вагових коефіцієнтів;

`public double[] x` – вектор вхідного сигналу;

`public double[] w` – вектор вагових коефіцієнтів;

`public double f` – значення дійсного виходу;

`public string typeOfNeuron` – параметр визначає тип нейрона в залежності від його позиції в мережі (вхідний сигмоїдальний, прихований, вхідний “1 з N”, вихідний);



public double[] dw – вектор приросту для наступної ітерації навчання;

public double error – величина похибки.

Методи класу Neuron:

public double GetF() – функція, що повертає значення виходу нейрона при поточному значенні вхідного сигналу;

public void SetSignal() – процедура активації синапсів (визначення вхідного сигналу);

public void SetDw() – процедура підрахунку вектора dw;

public double GetError() – функція визначення похибки;

public void CorectWeigts() – процедура налаштування вагових коефіцієнтів нейрона при поточній помилці.

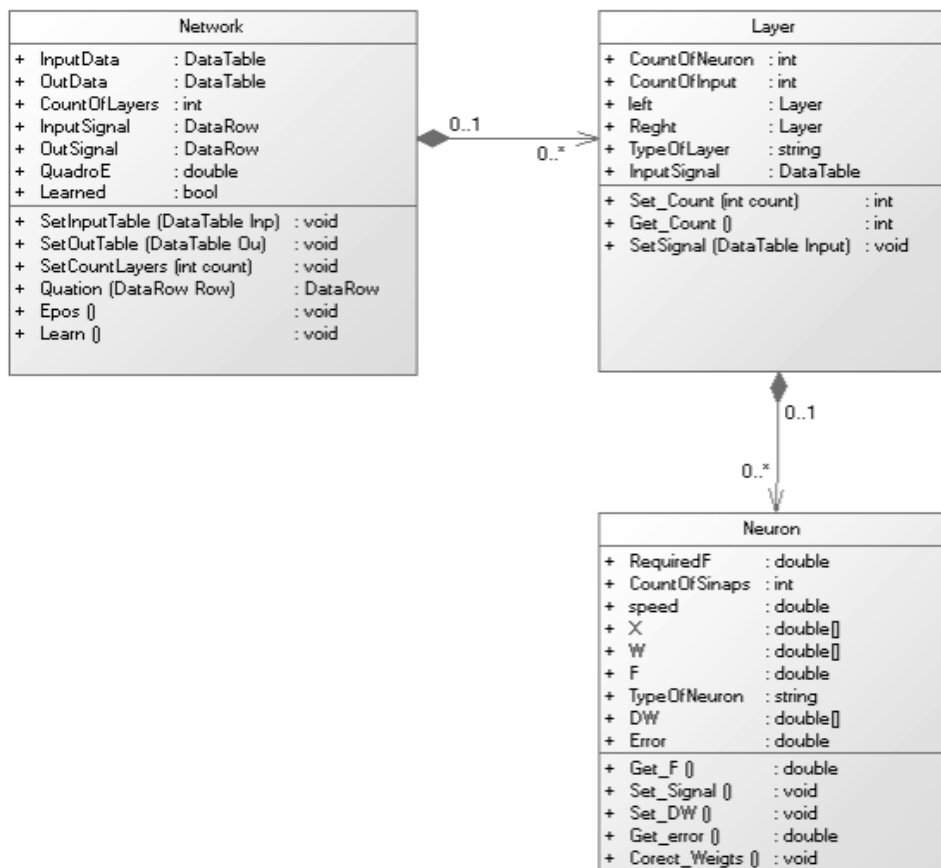


Рисунок 6.5 Діаграма класів реалізації нейронної мережі

**Клас Layer** – моделює стан та поведінку прошарку нейронів. Має в собі 1 або більше об'єктів класу Neuron.

Властивості класу Layer:

`public int countOfNeuron` – зберігає в собі інформацію про кількість нейронів у шарі.

`public int countOfInput` – кількість вхідних сигналів, що приймає прошарок. Це значення дорівнює кількості нейронів попереднього шару або кількості сенсорів, якщо це перший прошарок.

`public Layer left` – вказівник на попередній (лівий) прошарок, якщо поточний об'єкт це перший прошарок то вказівник вказує на пустий об'єкт.

`public Layer right` – вказівник на наступний (правий) прошарок, якщо поточний об'єкт – вихідний прошарок то вказівник вказує на пустий об'єкт.

`public string typeOfLayer` – тип прошарку, може приймати значення трьох типів: “input”, “output”, “hidden”, вхідний, вихідний і прихований, відповідно.

`public DataTable inputSignal` – таблиця вхідних сигналів. Тільки для вхідного прошарку.

Методи класу Layer:

`public int SetCount(int count)` – встановлює кількість нейронів в шарі.

`public int GetCount()` – повертає кількість нейронів в поточному шарі.

`public void SetSignal(DataTable input)` – ініціалізує вхідний сигнал. Для кожного синапса встановлює значення відповідне до поточного рядку таблиці сигналів.

**Клас Network** – моделює стан та поведінку прошарку нейронів. Включає в себе 1 або більше об'єктів класу Layer.

Властивості класу Network:

`public DataTable inputData` – основна таблиця вхідних сигналів в мережу. Вміщує в собі дані про с конвертовані значення факторів.

`public DataTable outData` – основна таблиця потрібних вихідних сигналів. Кількість така ж як і в таблиці `inputData`.

`public int countOfLayers` – несе в собі інформацію про кількість шарів, включаючи приховані, вхідний та вихідний прошарки.

`public DataRow inputSignal` – поточний сигнал для навчання мережі. Дорівнює рядку таблиці `inputData`, що має номер, який відповідає поточному значенню ітерації в епосі навчання.

`public DataRow outSignal` – поточний вихідний сигнал для навчання мережі. Дорівнює рядку таблиці `outData`, що має номер, який відповідає поточному значенню ітерації в епосі навчання.

`public double quadroE` – загальна квадратична похибка минулої епохи навчання.

`public bool learned` – має істинне значення, якщо середня квадратична похибка в межах допустимої.

Методи класу Network:

`public void SetInputTable(DataTable inp)` – завантажує таблицю вхідних сигналів. Та конвертує її в матрицю вхідних сигналів.

`public void SetOutTable(DataTable ou)` – завантажує таблицю вихідних сигналів. Та конвертує її в матрицю вихідних сигналів.

`public void SetCountLayers(int count)` – встановлює кількість прошарків мережі, включаючи прихований, вихідний та вхідний.

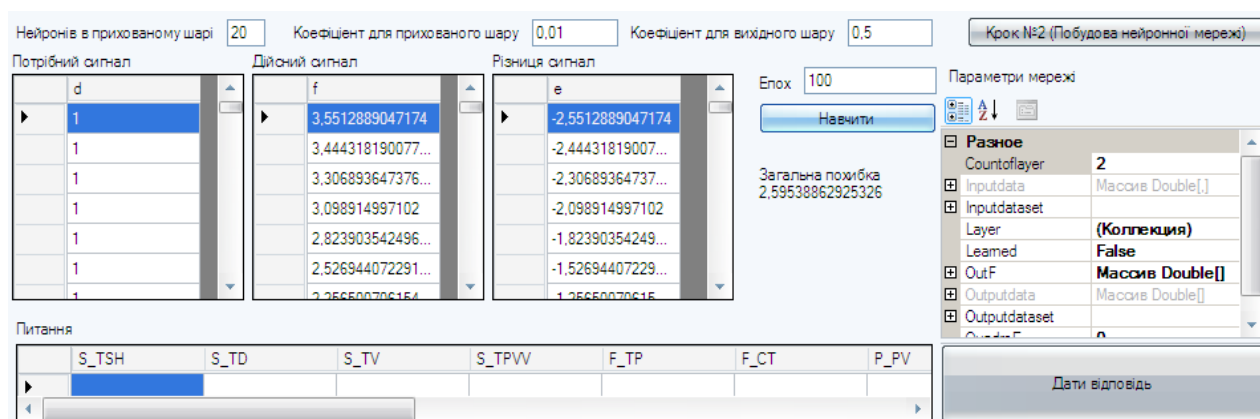
`public DataRow Quation(DataRow row)` – основний метод реалізуючи запитання до навченої мережі. Повертає рядок зі значеннями вихідного сигналу при вхідному сигналі, що є вхідним параметром `row`.

`public void Epos()` – навчає мережу періодом в одну епоху.

`public void Learn()` – навчає мережу доки значення загальної квадратичної похибки не лежатиме в межах допустимої.

Застосування об'єктно-орієнтованого підходу в проектуванні системи оцінки дозволило реалізувати прикладне застосування з широкими можливостями управління штучною нейронною мережею в реальному часі.

**Інтерфейс** управління і конфігурування штучною нейронною мережею. Користувач (експерт-оцінювач, дослідник) повинен мати можливість підбору аналогів з бази даних та інтерфейс, через який можна налаштовувати нейронну мережу для навчання. Інтерфейс включає в себе можливість управління запитом до нейронної мережі в будь який період навчання. На рис. 6.6 зображено основне вікно спостереження за мережею та її управління.



Дані якості навчання

Вікно управління

Рисунок 6.6 Інтерфейс управління і спостереження за штучною нейронною мережею

Інструментами управління і налаштуванням мережі є діалогові вікна для управління прошарками над нейронами в прошарках, а також відкритими для редагування властивостями. Вікно редагування властивостей прошарку зображено на рис. 6.7. Аналогічним є інструмент редагування властивостей штучних нейронів зображено на рис. 6.8.

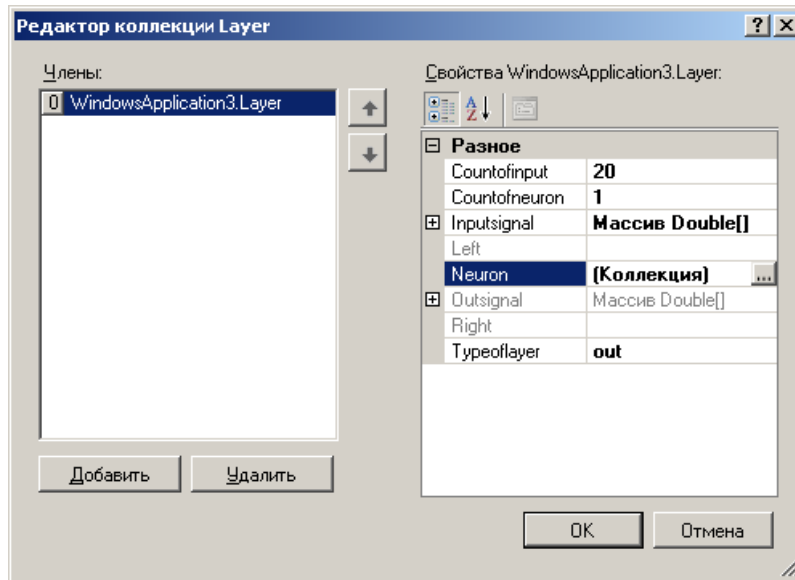


Рисунок 6.7 Редактор прошарків

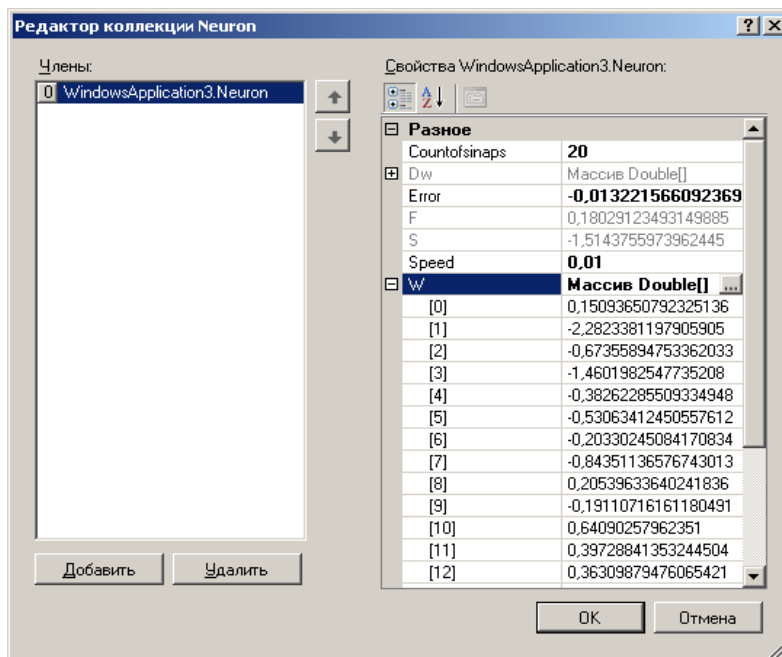


Рисунок 6.8 Редактор штучних нейронів

## 6.4 Модулі програмного забезпечення тестування нейронної мережі

Програмне забезпечення передбачає модульну структуру для підтримки можливості інтеграції окремих частин з іншими додатками. Також модульна структура полегшує подальше вдосконалення окремих частин без залежностей від інших. Схема модулів програмного забезпечення зображена на рис. 6.9.

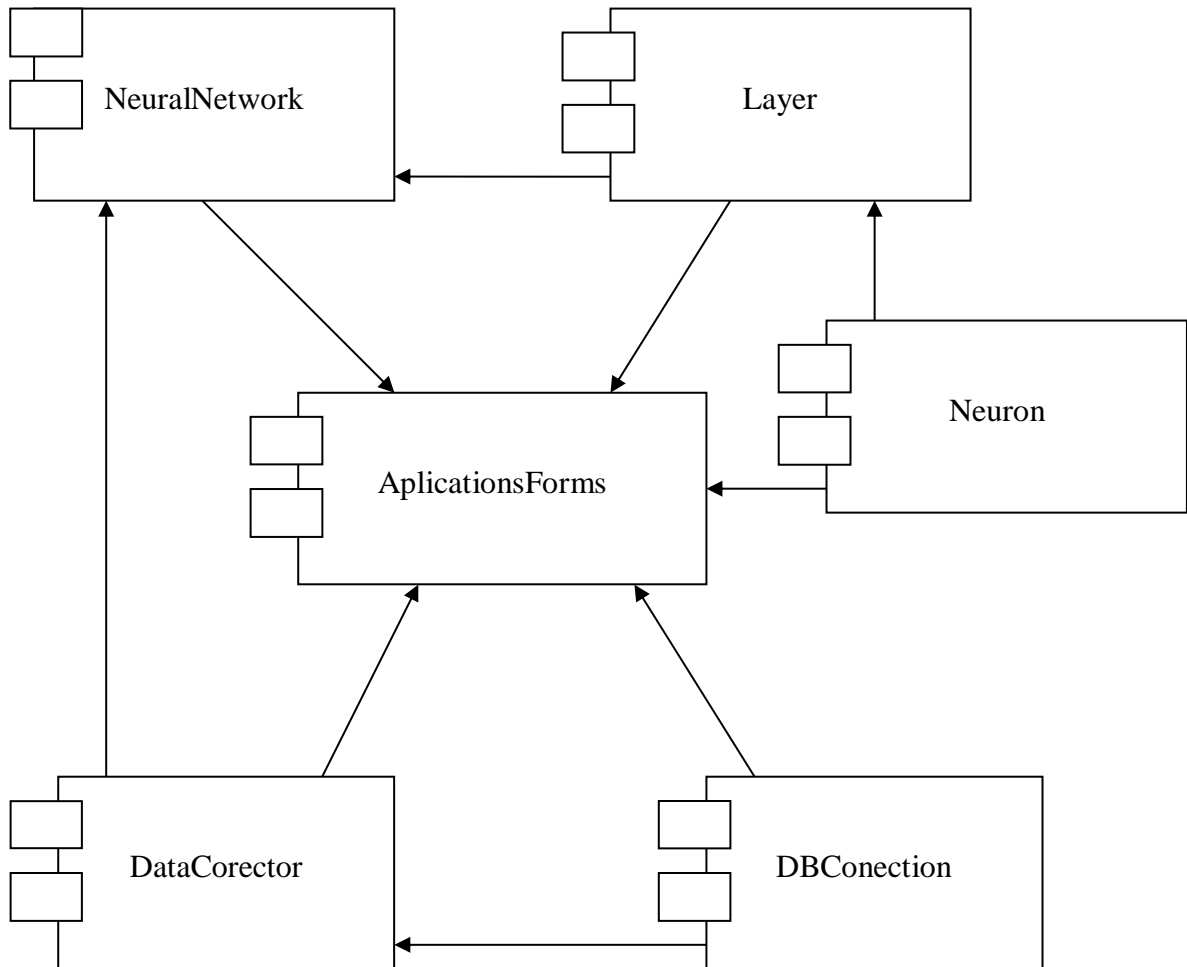


Рисунок 6.9 Модулі програмного забезпечення нейронної мережі

Neural Network – модуль реалізації штучної мережі. Містить в собі опис метаданих класу Network.

Layer – модуль реалізації колекції прошарків нейронів. Містить в собі опис метаданих класу Layer.

Neuron – модуль реалізації колекції штучних нейронів. Містить в собі опис метаданих класу Neuron. ApplicationsForms – модуль інтерфейсові частини програмного забезпечення.

DataCorector – модуль реалізації підсистеми підготовки даних для навчання та тестування мережі.

DBConnection – модуль підбору навчальної множини з'єднаний за базою даних. Має підтримку формування запитів до бази даних.

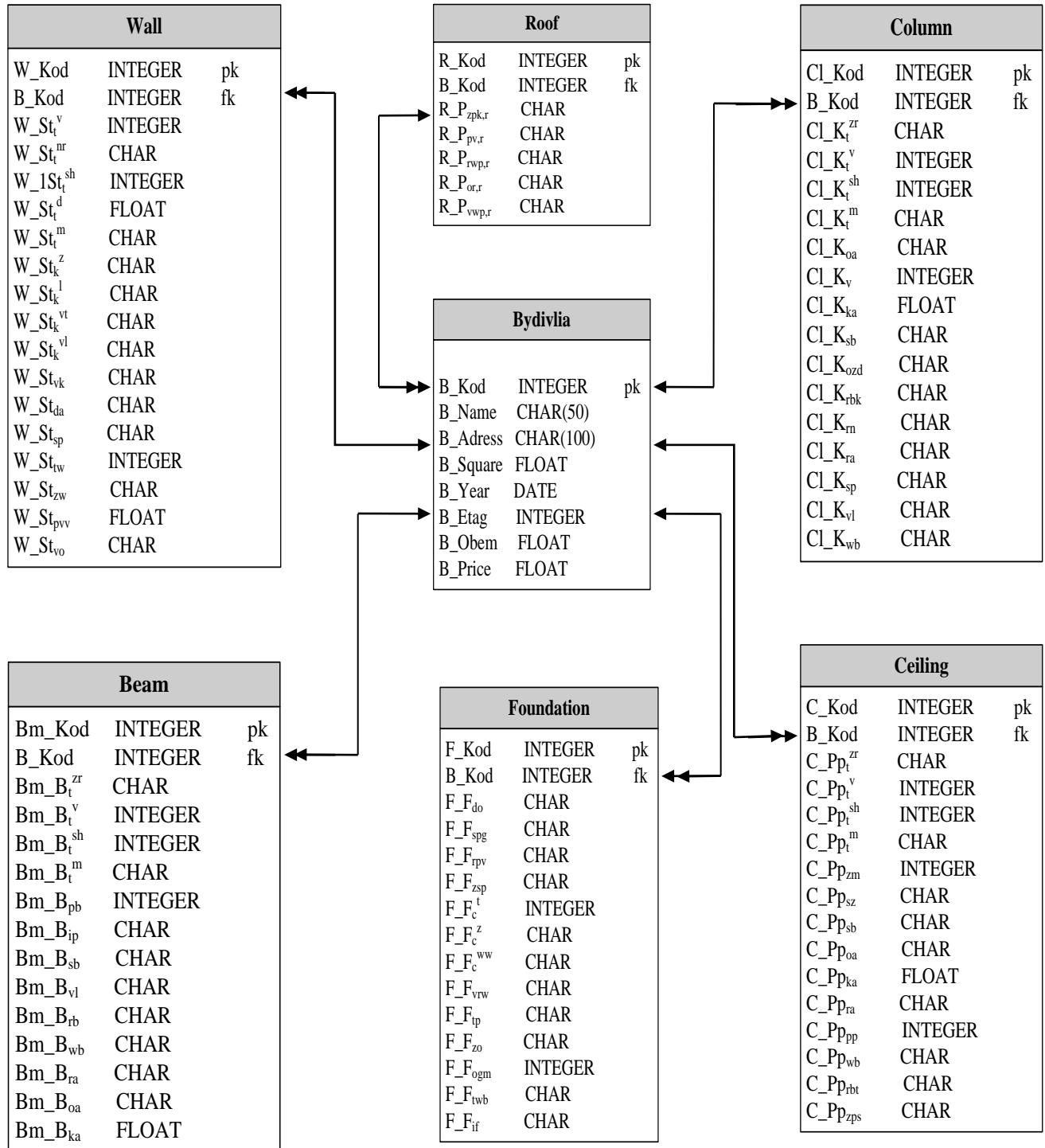


Рисунок 6.10 Даталогічна модель бази даних інформаційної системи

## 6.5 План проведення експериментального дослідження тестування нейронної мережі

На рис. 6.11 представлений план проведення експериментального дослідження тестування нейронної мережі.

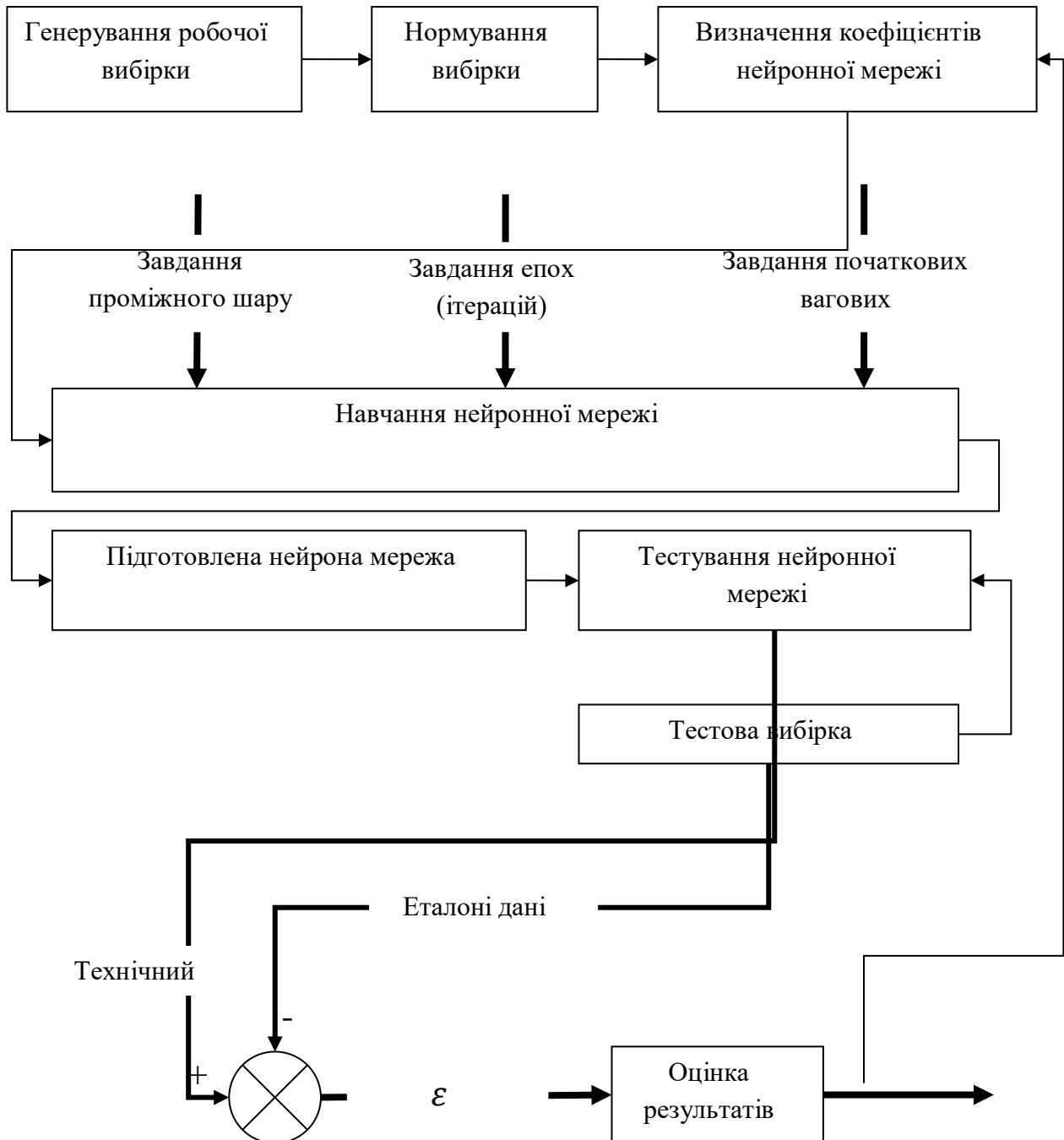


Рисунок 6.11 План експериментального дослідження тестування мережі



Для тестування мережі використовувались свіжі статистичні дані, що зібрані з доступних інтернет-ресурсів. Для навчання штучної нейронної мережі вибірка становила 220 прикладів, з яких 20 було використано для тестування мережі, а 200 – для навчання. Вибірка для тестування мережі не входила в набір даних, що корегують синапатичні коефіцієнти. Тому саме вона і була основним показником ефективності. Вибірка для навчання формувалася за допомогою критеріїв які описані в табл. 6.1.

Таблиця 6.1

Вхідні дані нейромережі

Назва елемента будівлі	Назва пошкоджень	Стани			
		нормальний	задовільний	непридатний	аварійний
Стіна	$St_t^{sh}$ – ширина розкриття тріщин	0...0,125	0,125...0,25	0,25...1	1...3
	$St_t^d$ – довжина тріщин	0...0,2	0,2...0,3	0,3...1	1...5
	$St_t^v$ – вид тріщин	0...7,5	7,5...15	15...75	75...90
	$St_{pvv}$ – перевищення допустимого відхилення від вертикалі	0...1,5	1,5...3	3...10	10...100
Фундамент	$F_{tp}$ – тріщини в плитній частині	0...0,5	0,5...1	1...5	5...100
	$F_c^t$ – тріщини цокольної частини	0...0,25	0,25...0,5	0,5...5	5...30
Плити перекриття	$Pp_t^v$ – вид тріщин	0...7,5	7,5...15	15...75	75...90
	$Pp_t^{sh}$ – ширина розкриття тріщин	0...0,15	0,15...0,3	0,3...1	1...3

	$P_{pp}$ – прогин плит	0...2,5	2,5...0,5	5...30	30...80
Залізобетонні колони	$K_t^v$ – вид тріщин	0...7,5	7,5...15	15...75	75...90
	$K_t^{sh}$ – ширина розкриття тріщин	0...0,15	0,15...0,3	0,3...1	1...3
	$K_v$ – викривлення колони	0...2,5	2,5...0,5	5...10	10...50
Залізобетонні балки	$V_t^v$ – вид тріщин	0...7,5	7,5...15	15...75	75...90
	$V_t^{sh}$ – ширина розкриття тріщин	0...0,15	0,15...0,3	0,3...1	1...3
	$V_{pb}$ – прогин балки	0...2,5	2,5...0,5	5...10	10...40

В додатку №1 представлені дані для навчання нейронної мережі.

Так як на вхід сигмоїдального нейрона подаються сигнали тільки в межах від 0 до 1, то система перетворює занесені дані. Перетворені дані представлені на рис. 6.12.

Дані після обробки (нормалізовані дані для навчання)

	S_TSH	S_TD	S_TV	S_TPVV	F_TP	F_CT	P_PV	P_PSH	P_PPP
▶	0	0	0	0	0	0	0	0,016949152542...	0
	0,001666666666...	0,01	0,010011123470...	0,0005	0,0005	0,001666666666...	0,010011123470...	0	0,000625
	0,003333333333...	0,02	0,016685205784...	0,001	0,001	0,003333333333...	0,016685205784...	0,016949152542...	0,00125
	0,004166666666...	0,025	0,020022246941...	0,00125	0,00125	0,004166666666...	0,020022246941...	0,025423728813...	0,0015625

Рисунок 6.12 Нормовані дані

Після нормування даних проводиться корегування нейронної мережі, а саме кількість нейронів в прихованому шарі обирається 20. Далі встановлюються коефіцієнти вагів для внутрішнього-0,1 і 0,5-

зовнішнього шарів нейронної мережі, а також кількість епох (ітерацій) 100 представлено на рис. 6.13.

Після внесення коефіцієнтів починається навчання нейронної мережі. І чим більше кількість епох тим довше навчається система, що представлено на рис. 6.14.

Після навчання визначається загальна похибка ідентифікації. Перевіряється нейрона мережа контрольною вибіркою і отримується результат представлено на рис. 6.15.



Рисунок 6.13 Ведення коефіцієнтів

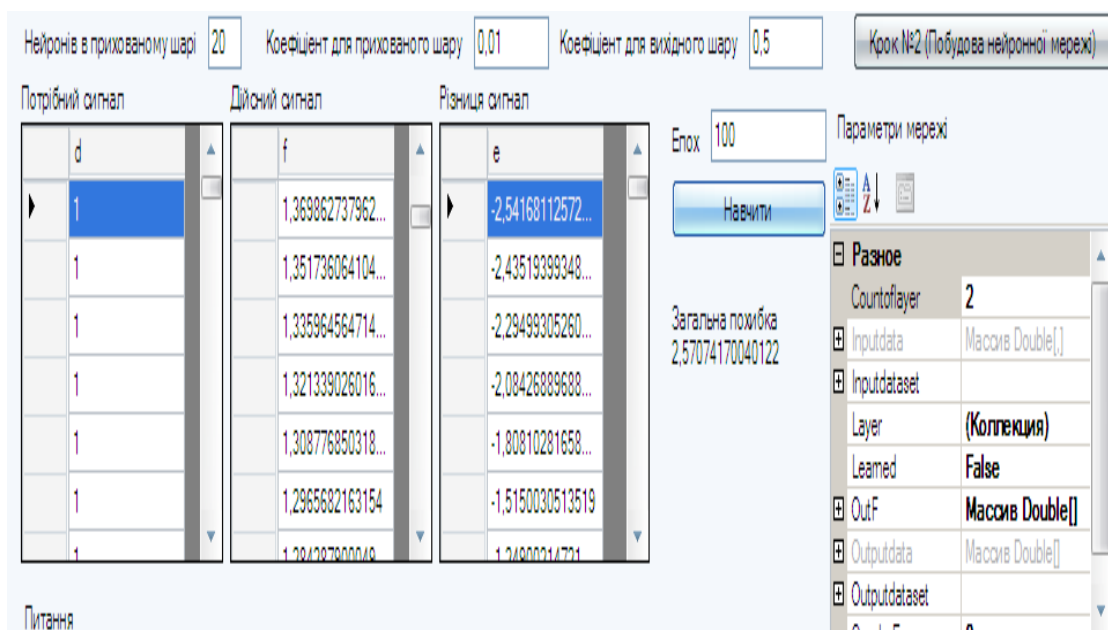


Рисунок 6.14 Навчання нейронної мережі

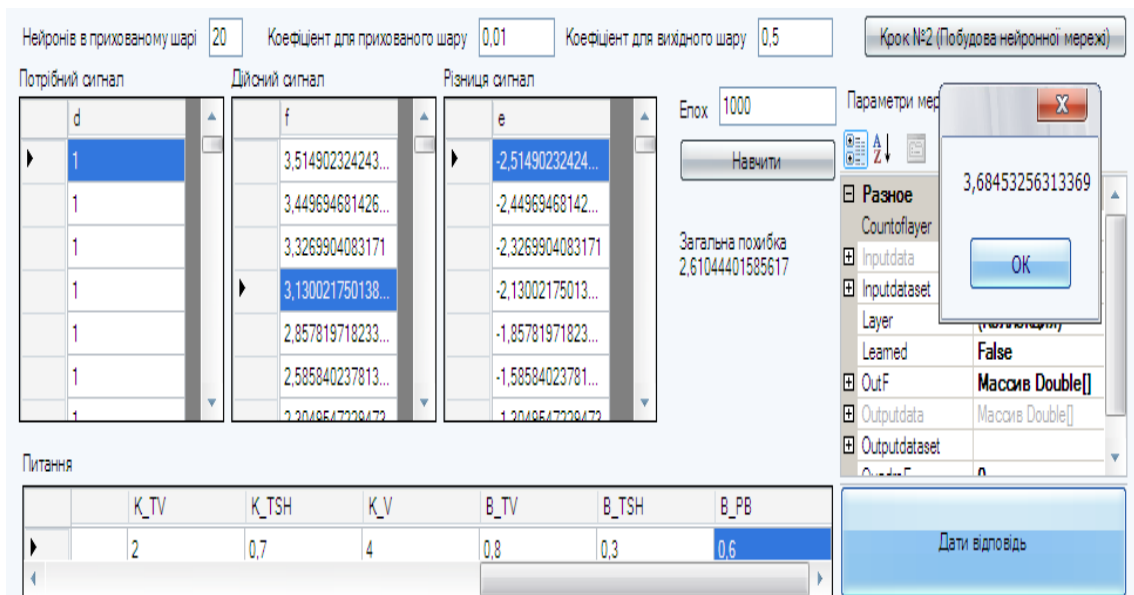


Рисунок 6.15. Виведення результатів нейронної мережі

## 6.6 Результати дослідження тестування нейронної мережі

На рисунку 6.16 представлений результат тестування нейронної мережі.

Навчання штучної нейронної мережі зайняло 10 хвилини 68000 епох. Графік залежності загальної середньої квадратичної похибки від номеру епохи навчання показано на рис. 6.17.

Проведене дослідження доводить потенційну ефективність використання нейронних мереж також для аналізу ринку нерухомості при проведенні оціночної діяльності.

Подальше збільшення прикладів для навчання мережі та кількості критеріїв зменшить загальну похибку мережі.

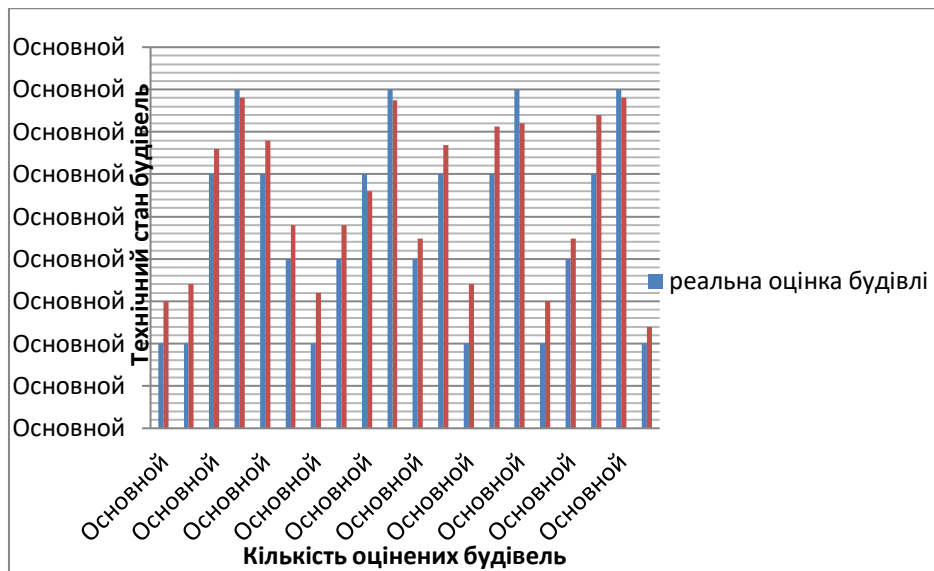


Рисунок 6.16 Гістограма тесту на реальних даних

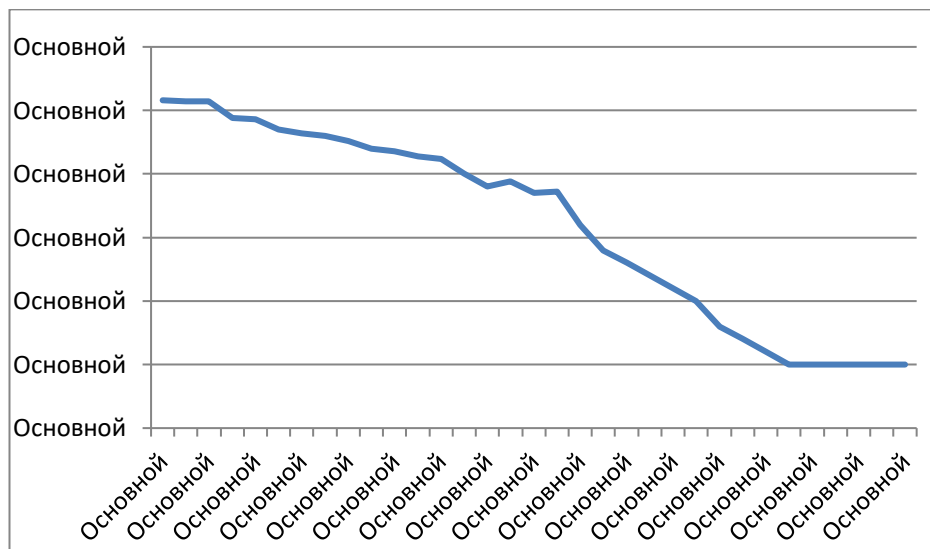


Рисунок 6.17. Зменшення значення похибки при навчанні

## 6.7 Тестовий приклад нейронної мережі

Спочатку будується і навчається нейронна мережа, що зображена на рис. 6.18 і 6.19.

Після навчання нейронної мережі вноситься данні про будівлі і натискається кнопка “Дати відповідь” і система видає результат, представлений на рис. 6.20.

Оцінка стану будівель

Дані для навчання нейронної мережі

	S_TSH	S_TD	S_TV	S_TPW	F_TP	F_CT	P_PV	P_PSH	P_PPP
	0	0	0,1	0	0	0	0,1	0,1	0
▶	0,05	0,05	1	0,05	0,05	0,05	1	0,05	0,05
	0,1	0,1	1,6	0,1	0,1	0,1	1,6	0,1	0,1
	0,125	0,125	1,9	0,125	0,125	0,125	1,9	0,125	0,125
	0,135	0,135	2	0,135	0,135	0,135	2	0,135	0,135

Крок №1 (Нормалізація даних для навчання мережі)

Дані після обробки (нормалізовані дані для навчання)

	S_TSH	S_TD	S_TV	S_TPW	F_TP	F_CT	P_PV	P_PSH	P_PPP
▶	0	0	0	0	0	0	0	0,016949152542...	0
	0,001666666666...	0,01	0,010011123470...	0,0005	0,0005	0,001666666666...	0,010011123470...	0	0,000625
	0,003333333333...	0,02	0,016685205784...	0,001	0,001	0,003333333333...	0,016685205784...	0,016949152542...	0,00125
	0,004166666666...	0,025	0,020022246941...	0,00125	0,00125	0,004166666666...	0,020022246941...	0,025423728813...	0,0015625
	0,004500000000...	0,027000000000...	0,021111150000...	0,00135	0,00135	0,004500000000...	0,021111150000...	0,020011550000...	0,0016750000...

Нейронів в прихованому шарі  Коefіцієнт для прихованого шару  Коefіцієнт для вихідного шару  Крок №2 (Побудова нейронної мережі)

Потрібний сигнал      Дійсний сигнал      Різниця сигнал

Epoch  Параметри мережі

Загальна помилка  
0

Рисунок 6.18 Побудова нейронної мережі

Оцінка стану будівель

Дані для навчання нейронної мережі

	S_TSH	S_TD	S_TV	S_TPWV	F_TP	F_CT	P_PV	P_PSH	P_PPP
▶	0	0	0,1	0	0	0	0,1	0,1	0
	0,05	0,05	1	0,05	0,05	0,05	1	0,05	0,05
	0,1	0,1	1,6	0,1	0,1	0,1	1,6	0,1	0,1
	0,125	0,125	1,9	0,125	0,125	0,125	1,9	0,125	0,125
	0,135	0,135	2	0,135	0,135	0,135	2	0,135	0,135

Крок №1 (Нормалізація даних для навчання мережі)

Дані після обробки (нормалізовані дані для навчання)

	S_TSH	S_TD	S_TV	S_TPWV	F_TP	F_CT	P_PV	P_PSH	P_PPP
▶	0	0	0	0	0	0	0	0,016949152542...	0
	0,001666666666...	0,01	0,010011123470...	0,0005	0,0005	0,001666666666...	0,010011123470...	0	0,000625
	0,003333333333...	0,02	0,016685205784...	0,001	0,001	0,003333333333...	0,016685205784...	0,016949152542...	0,00125
	0,004166666666...	0,025	0,020022246941...	0,00125	0,00125	0,004166666666...	0,020022246941...	0,025423728813...	0,0015625
	0,004500000000...	0,027000000000...	0,021124502000...	0,00135	0,00135	0,004500000000...	0,021124502000...	0,020013550000...	0,001075000000...

Нейронів в прихованому шарі  Коefіцієнт для прихованого шару  Коefіцієнт для вихідного шару  Крок №2 (Побудова нейронної мережі)

Потрібний сигнал      Дійсний сигнал      Різниця сигнал

d	f	e
▶ 1	3,553261589824...	-2,55326158982...
1	3,3462640646281	-2,3462640646281
1	2,977465076601...	-1,97746507660...
1	2,407934695903...	-1,40793469590...
1	1,948175100696...	-0,94817510069...
1	1,738811024684...	-0,73881102468...
1	1,602280000000...	-0,60228000000...

Епох  Параметри мережі

Навчити

Загальна помилка  
1,69719938712756

Параметри мережі

- Разное
- Countoflayer **2**
- Inputdata Массив Double[]
- Inputdataset
- Layer **(Коллекция)**
- Learned **False**
- OutF **Массив Double[]**
- Outputdata Массив Double[]
- Outputdataset
- OutF

Питання

	S_TSH	S_TD	S_TV	S_TPWV	F_TP	F_CT	P_PV
▶							

Дати відповідь

Рисунок 6.19 Навчання нейронної мережі

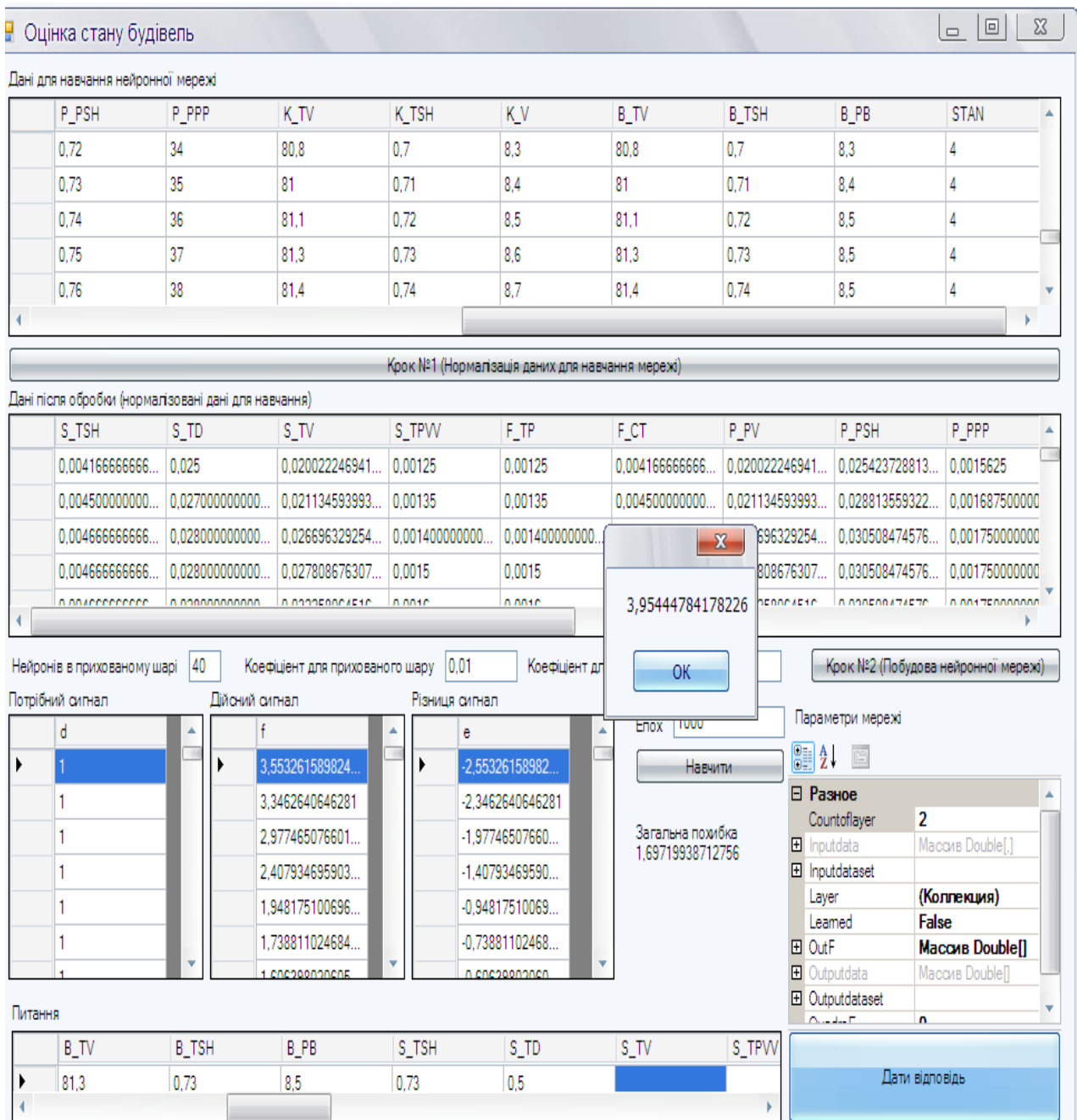


Рисунок 6.20 Результат нейронної мережі

*Висновок:* Проведене дослідження доводить потенційну ефективність використання нейронних мереж для побудови моделей і методів автоматизованої системи діагностики технічного стану будівель.



## ВИСНОВКИ ПО РОЗДІЛУ 6

1. Представлена архітектура інформаційної системи тестування нейронної мережі.
2. Наведені особливості моделі визначення тестування нейронної мережі.
3. Запропоновані основні модулі програмного забезпечення тестування нейронної мережі.
4. Представлений план проведення експериментального дослідження тестування нейронної мережі.
5. Запропоновані основні результати дослідження тестування нейронної мережі.

## СПИСОК ЛІТЕРАТУРИ

1. Ануфриев И.Е., Смирнов А.Б., Смирнова Е.Н. MATLAB 7.-СПб.: БХВПетербург, 2005.- 1104 с.
2. Бакулин М.Г., Варукина Л.А., Крейнделин В.Б. Технология МІМО: принципы и алгоритмы. – М.: Горячая линия – Телеком, 2014. – 244 с.
3. Барвінський А.Ф, Олексін І.Я, Крупка З.І. та ін. Математичне програмування. – Львів: —Інтелект – Захід, 2004. – 446 с.
4. Волков И.К., Загоруйко Е.А. Исследование операций: Учеб. для вузов.-М.: Изд-во МГТУ им. Н.Э. Баумана, 2002.- 436 с.
5. Вітлінський В.В., Наконечний С.І., Терещенко Т.О. Математичне програмування. – К.: КНЕУ, 2001. – 248 с.
6. Герман-Галкин С.Г. Компьютерное моделирование полупроводниковых систем в MATLAB 6.0: Учебное пособие. – СПб.: КОРОНА принт, 2001. -320 с.
7. Данциг Дж. Линейное программирование, его применения и обобщения /Пер. с англ. Г.Н. Андрианова и др. Общ. Ред. И предисл. Н.Н. Воробьева. – М.: Прогресс, 1966. –600 с.
8. Зуховицкий С.И. и др. Линейное и выпуклое программирование. М.: Наука, 1967. – 460 с.
9. Зайченко Ю.П. Исследование операций: Учебник. – 6 изд., перераб. и доп. – Киев: Издательский Дом «Слово», 2003.- 688 с.
10. Зайченко О.Ю., Зайченко Ю. П. Дослідження операцій. Збірник задач. – К.: Видавничий Дім “Слово”, 2007.- 472 с.
11. Костевич Л.С. Математическое программирование: Информ. Технологии оптимальных решений: Учеб. Пособие.- Минск.: Новое знание, 2003.-424 с.
12. Лавров Є.А., Перхун Л.П., Шендрик В.В., Кузнецов Е.Г., Парфененко Ю.В., Сергієнко В.А. Математичні методи дослідження операцій: підручник. - Суми: СумДУ, 2017-212с.

13. Ларіонов Ю.І., Левикін В.М., Хажмурадов М.А. Дослідження операцій в інформаційних системах.-Харків.: Компанія СМІТ, 2005.-364 с.
14. Мэтьюз Д.Г., Финк К.Д. Численные методы. Использование MATLAB.-М.: Издательский дом «Вильямс», 2001.- 720 с.
15. Самсонов В.В. Алгоритми розв'язання задач оптимізації: Навчальний посібник. К.: НУХТ, 2014.-300 с.
16. Таха Хемди А. Введение в исследование операций. - М.: издательский дом “Вильямс”, 2005.-912с.
17. Томашевський В.М. Моделювання систем. Підручник. -К.: Видавнича група ВНУ, 2007.- 352 с.
18. Цисарь И.Ф. MATLAB Simulink. Компьютерное моделирование в экономике . – М.: СОЛОН-ПРЕСС, 2008.- 256 .
19. [https://essuir.sumdu.edu.ua/bitstream-download/123456789/68212/1/Lavrov\\_matematychni\\_metody.pdf](https://essuir.sumdu.edu.ua/bitstream-download/123456789/68212/1/Lavrov_matematychni_metody.pdf)
20. <https://core.ac.uk/download/pdf/11316926.pdf>
21. Інтелектуальна інформаційна технологія діагностики технічного стану будівель [Текст] : монографія /В. М. Михайленко, О. О. Терентьєв, М. І. Цюцюра // – К: ЦП «Компринт», 2015. – С. 162.
22. Михайленко В. М. Організація та технологія проведення обстеження технічного стану будівель на основі інформаційних технологій /Яцик П. М., Терентьєв О. О., Полторак О. Б.// – К.: Будівельне виробництво, міжвідомчий науково-технічний журнал, випуск 50/2009, НДІБВ, 2009. – С. 23-29.
23. Михайленко В. М. Теоретико-множинна модель будівлі при обстеженні технічного стану /Яцик П. М., Терентьєв О. О., Полторак О. Б.// – К.: Будівельне виробництво, міжвідомчий науково-технічний журнал, випуск 50/2009, НДІБВ, 2009. – С. 16-23.
24. Михайленко В. М. Моделі структури властивостей будівлі при проведенні обстеження для вирішення задач діагностики /Терентьєв О. О.// –

К.: Будівельне виробництво, міжвідомчий науково-технічний журнал, випуск 50/2009, НДІБВ, 2009. – С. 30-34.

25. Михайленко В. М. Загальний підхід до моделювання надійності фундаментних конструкцій /Терентьев О. О., Корнієнко М. В.// – П.: Галузеве машинобудування, будівництво, випуск 3(28), Полтава, 2010. – С. 178-185.

26. Михайленко В. М. Дослідження ланцюгів логічного виведення роботи експертної системи при обстеженні технічного стану будівель /Терентьев О. О., Яцик П. М., Полтораки О. Б.// – К.: Будівельне виробництво, міжвідомчий науково-технічний журнал, випуск 52/2010, НДІБВ, 2010. – С. 37-44.

27. Михайленко В. М. Інформаційна технологія оцінки технічного стану елементів будівельних конструкцій із застосуванням нечітких моделей /Терентьев О. О., Єременко Б. М.// – Д.: Строительство, материаловедение, машиностроение, сб. науч. трудов Под общей редакцией профессора В. И. Большакова выпуск 70/2013, Днепропетровск, 2013. – С. 133-141.

28. Михайленко В. М. Обробка експериментальних результатів роботи експертної системи для задачі діагностики технічного стану будівель /Терентьев О. О., Єременко Б. М.// – Д.: Строительство, материаловедение, машиностроение, сб. науч. трудов Под общей редакцией профессора В. И. Большакова выпуск 78/2014, Днепропетровск, 2014. – С. 190-195.

29. Михайленко В. М. Експериментальні дослідження та реалізація інформаційної системи тестування нейронної мережі для задачі діагностики технічного стану будівель / Терентьев О. О., Шабала Є. Є., Турушев О. С.// – К.: Управління розвитком складних систем, збірник наукових праць, випуск 27/2016, КНУБА, 2016. – С. 139-144.

30. Михайленко В. М. Побудова діагностичних моделей основних конструкцій будівель / Терентьев О. О., Шабала Є. Є., Баліна О. І., Доля О. В.// – К.: Управління розвитком складних систем, збірник наукових праць, випуск 28/2016, КНУБА, 2016. – С. 155-159.

31. Моделі і методи системи діагностики технічного стану будівель [Текст] : монографія / А. О. Білощицький, П. Є. Григоровський, О. О. Терентьєв // – К: ЦП «Компринт», 2015. – С. 232.
32. Терентьєв О. О. Основи організації нечіткого виведення для задачі діагностики технічного стану будівель та споруд /Шабала Є. Є, Малина Б. С.// – К.:Управління розвитком складних систем, збірник наукових праць, випуск 22/2015, КНУБА, 2015. – С.138 – 143.
33. Терентьєв О. О. Експертна інформаційна система підтримки прийняття рішень для задачі діагностики технічного стану будівель / Шабала Є. Є, Гайдаржи І. О.// – К.: Управління розвитком складних систем, збірник наукових праць, випуск 24/2015, КНУБА, 2015. – С. 131–136.
34. Терентьєв О. О. Розробка інформаційної технології проектування та контролю місцеположення мобільних об'єктів / Баліна О. І., Шабала Є. Є., Турушев О. С.// – К.: Управління розвитком складних систем, збірник наукових праць, випуск 25/2016, КНУБА, 2016. – С. 133–139.
35. Терентьєв О. О. Моделі визначення фізичного зношення конструктивних елементів будівлі для задач діагностики технічного стану / Баліна О. І., Шабала Є. Є.// – К.: Управління розвитком складних систем, збірник наукових праць, випуск 26/2016, КНУБА, 2016. – С. 153-157.
36. Olexander Terentyev The Method of Direct Grading and the Generalized Method of Assessment of Buildings Technical Condition /Mykola Tsiutsiura// – International Journal of Science and Research (IJSR), Volume 4 Issue 7, July 2015. – P. 827-829.
37. Olexander Terentyev The Method of Prediction of Deformations of Buildings and Failure Analysis the Examination of Technical Condition of Buildings /Malyna Bohdan// – International Journal of Science and Research (IJSR), Volume 4 Issue 8, August 2015. – P. 280-282.

38. Olexander Terentyev Methodology a comprehensive survey and assessment of technical condition of staircases – Scientific Journal «ScienceRise», Volume 8/2(13), August 2015. – P. 41-46.

39. Olexander Terentyev Mathematical model of the system of decision support for problem diagnostics of technical condition of building constructions – Scientific Journal «ScienceRise» №9/2(14), September 2015. – P. 35-40.

40. Olexander Terentyev Expert information system for decision support for the problem of diagnostics of technical condition of buildings /Bohdan Malyna// – International Journal of Science and Research (IJSR), Volume 4 Issue 10, October 2015. – P. 652–654.

41. Olexander Terentyev Development of models and methods for determining the physical deterioration of items for the task of diagnostics of technical condition of buildings and structures /Olexander Poltorak// – Scientific Journal «ScienceRise» №8/2(25), August 2016. – P. 14-19.

Навчальне видання

**Терентьев** Олександр Олександрович

**Бондар** Олена Анатоліївна

**Доля** Олена Вікторівна

**Баліна** Олена Іванівна

**Горбатюк** Євгеній Володимирович

**Кучанський** Олександр Юрійович

## **ДОСЛІДЖЕННЯ ОПЕРАЦІЙ**

Навчальний посібник